

# Multispektral billedanalyse af fødevarer

Jesper R. Rasmussen og Lars H. Nikolajsen

Kongens Lyngby 2006

Technical University of Denmark  
Informatics and Mathematical Modelling  
Building 321, DK-2800 Kongens Lyngby, Denmark  
Phone +45 45253351, Fax +45 45882673  
[reception@imm.dtu.dk](mailto:reception@imm.dtu.dk)  
[www.imm.dtu.dk](http://www.imm.dtu.dk)

# Resumé

---

Dette projekt omhandler emnet ”multispektral billedanalyse af fødevarer”. De tre undersøgte områder er adskillelsen af fedt og kød samt bestemmelse af kødfarve for spegepølser, estimering af barrierelagstykkelsen for en kiks med et påført lag af vandbarriere og estimering af vandindhold i brød. Projektet er udarbejdet i samarbejde med Danisco A/S.

Et multispektralt billede består af billeder taget ved mange forskellige bølgelængder. Billederne brugt i dette projekt er alle taget med et VideometerLab system fra Videometer A/S og indeholder 18 spektra fra 395 – 970nm.

## **Kødfarven af spegepølser**

En spegepølseres kødfarve er, ud over smag og pris, en vigtig salgsparemeter. Fedtmængden, -fordelingen og -farven vil påvirke hvordan det menneskelige øje ser farven, og forskellige personer vil kunne vurdere den samme farve forskelligt. Derfor vil en entydig og reproducerbar angivelse af farven være et nyttigt værktøj i forbindelse med udvikling og salg af tilsætningsstoffer for spegepølser til industrien.

Der er udviklet en metode til at estimere farven af kødet i en spegepølse ved at anvende PCA (principal komponent analyse) eller MAF (minimum autokorrelationsfaktorer). Derefter er fedtet og kødet skilt fra hinanden med en adaptiv thresholding, og der beregnes en gennemsnitsværdi af farven fra midten og ud mod kanten. Fedtets gennemsnitsfarve bestemmes på tilsvarende vis. Kød- og fedtfarven angives med to tal, et der angiver hvor lys/mørk den er og et der angiver rødheden. Derudover er der fremstillet en metode til at skille fedtområderne fra hinanden og angive antallet og størrelserne af dem.

### Estimering af lagtykkelsen for vandbarrieren til kiks

Vandbarriere på kiks har til formål at øge holdbarheden for fødevarer hvor et vandholdigt produkt ligger på en sprød kiks, ved at forhindre vandet i at trænge ind i kiksken. En tilstrækkelig tykkelse af barrierelaget og en påførsel uden huller eller revner er nødvendig for at barrierelaget virker efter hensigten. For vandbarrieren på kiks er der opstillet en lineær model ud fra en serie af kiks hvor barrierelagstykkelsen er kendt på forhånd. Modellen beskriver højden af det påførte barrierelag i hver pixel og standardafvigelsen for det samlede billede angiver hvor jævnt barrierelaget er påført. Den gennemsnitlige standardafvigelse for jævnheden er fundet til hhv.  $30,61\mu m$  og  $52,84\mu m$  for de to dataserier. Derudover er barrierelagstykkelsen præsenteret som et højdekort hvor tykkelsen i de enkelte områder af kiksken kan aflæses ud fra farvespektret. Den beregnede standardafvigelse på testdata for den bedste model er på  $93,53\mu m$  i forhold til de målte tykkelser. Selvom modellen har givet lovende resultater, har datasættet ikke været godt nok til at vurdere om resultaterne er brugbare.

### Estimering af vandindholdet i brød

Vandindholdet i brød er en vigtig parameter i forhold til brødets friskhed, og de nuværende metoder (HB43 og LP16) til måling af vandindhold er relativt tidskrævende. Der er udviklet et system til at estimere vandindholdet i en brødske ud fra statistiske features af et multispektralt billede. Den bedste model er fundet vha. LARS-EN regressionsanalyse. Modellen kan ud fra features beregnet på hele billedet estimere vandindholdet i procent målt med HB43-metoden med en standardafvigelse på 0,71 procentpoint, og 1,18 procentpoint for målinger med LP16-metoden. Estimeringen kan forbedres ved kun at benytte den del af billedet der ligger tættest på brødets snitflade til beregning af features. Med 38% af billedet opnås en standardafvigelse for HB43-metoden på 0,68 procentpoint, og med 40% opnås en standardafvigelse for LP16-metoden på 1,14. Denne forbedring opnås formentlig fordi variationskilden, i form af større luftpiller i brødet, elimineres.

For alle tre emner er der udviklet en softwarepakke til at udføre analyserne så automatiseret som muligt. Derudover er der blevet udviklet en softwarepakke der udnytter den multispektrale information i billedet til at lave en robust opdeling mellem for- og baggrund. Denne pakke har vist sig anvendelig overfor alle de billeder vi har haft til rådighed i vores datasæt, men har dog visse svagheder overfor billeder med meget lidt baggrund.

Der er ikke blevet stillet konkrete krav til hvor præcise de tre systemer skal være, så det er op til Danisco at vurdere om de fundne resultater gode nok til at systemerne kan tages i brug.



# Abstract

---

This project deals with the topic "Multispectral Image Analysis in the Food Industry". The three analysed subjects are: the separation of meat and fat plus determining the meat colour of salami; estimation of the barrier thickness of a cracker with applied water barrier system; and estimating the water content in bread. The project is developed in collaboration with Danisco A/S.

A multispectral image consists of images taken at many different wavelengths. The images used in this project are all taken using VideometerLab from Videometer A/S and contains 18 spectra from 395 – 970nm.

## **Meat colour of salami**

Besides taste and price, the meat colour of the salami is an important sales parameter. The amount, spread and colour of the fat will affect how the human eye sees the colour and different people will evaluate the same colour differently. An unambiguous and reproducible measure of colour will therefore be a useful tool in connection with development and sale of food additives for the industry.

A method for estimating the meat colour of salami has been developed with the use of PCA (Principal Component Analysis) or MAF (Minimum Autocorrelation Factors). The meat and fat is separated with adaptive thresholding and the mean meat colour value is calculated in rings from the centre to the surface of the salami. The same is done for the fat. The meat and fat colour is noted by two numbers. The first one denotes how light/dark the salami is and the second denotes the redness. Besides this, a method of separating the fat lumps from each other and providing the number of lumps and their sizes has been developed.

### **Estimation of water barrier thicknesses on crackers**

The purpose of the water barrier is to extend the storage life of snacks consisting of crackers with low water content and food products with high water content, by preventing the water from entering the cracker. A certain layer thickness and a layer without holes or cracks are essential for the water barrier to have the wanted effect. A linear regression model has been made on the basis of a series of crackers with a known applied barrier thickness. The model describes the height of the barrier layer for each pixel as a height map, and the standard deviation for the whole image describes how evenly the layer has been applied. The average standard deviation found of layer uniformity is  $30,61\mu m$  and  $52,84\mu m$  for the two data series respectively. From the colours of the height map it is possible to see the thickness of the barrier layer in different regions of the cracker. The calculated mean standard deviation on test data for the best model is  $93.53\mu m$ . The dataset was not welldefined enough to evaluate the usability of the method, although the results where promising.

### **Estimation of water content in bread**

The water content in bread is an important factor considering the freshness of the bread. The current methods (HB43 and LP16) for measuring water content are relatively time-consuming. A system for estimating water content in a bread slice from statistical features of a multispectral image is developed. The best model is found using LARS-EN regression analysis. The model is able to estimate the HB43 measured water content percentage with a standard deviation of 0,71 percentage points from features calculated from the whole image, and 1,81 percentage points for the LP16 method. By only considering the part of the image that is closest to the cut surface of the bread it is possible to improve the estimation. Using 38% of the image a standard deviation of 0,68 percentage points is obtained for the HB43 method and using 40% of the image for the LP16 method we get a standard deviation of 1,14 percentage points. This improvement is probably caused be the elimination of a variation source of larger air holes in the bread.

For all three subjects a software package for performing the analysis as automated as possible has been developed. Besides this a software package that takes advantage of the multispectral information in the image to separate foreground from background in a robust manner. This package has proved itself useful for all images of the presented dataset, but has some weaknesses in regard to images with smaller amounts of background.

There has not been made definite requirements in regards to the precision of the three systems, meaning that Danisco must evaluate if the results are good enough to start using the developed systems.

# Forord

---

Dette afgangsprøveprojekt er lavet ved instituttet Informatik og Matematisk Modellering (IMM), Danmarks Tekniske Universitet (DTU) i Lyngby, København. Projektet er udarbejdet af Lars H. Nikolajsen og Jesper R. Rasmussen og omfatter et værk på 35 ECTS point pr. person, som afsluttende eksamensprojekt på civil ingeniør uddannelsen og giver adgang til titlen Master of Science and Engineering, M.Sc.Eng. Projektet er udarbejdet i foråret og ”den varmeste sommer nogensinde”, fra 1. februar 2006 frem til 1. september 2006.

Projektet er udarbejdet i samarbejde med Danisco A/S der har stillet al billedmateriale og forsøgsudstyr til rådighed. Samtlige multispektrale billeder er optaget med et VidemeterLab system. Projektet omhandler emnet ”multispektral billedanalyse af fødevarer”. De tre undersøgte områder er adskillelsen af fedt og kød samt kødfarvebestemmelse for spegepølser, kiks med et påført lag af vandbarriere og estimering af vandindhold i brød.

Det forventes af læseren at denne som minimum er bekendt med grundlæggende billedbehandling og statistik.

Kgs. Lyngby, 31. august 2006

---

Lars H. Nikolajsen (s001954)

Jesper R. Rasmussen (s001942)



# Tak

---

Vi vil gerne sende en stor tak til vores vejleder Jens Michael Carstensen for sine mange og gode idéer til fremførelse og fremmelse af projektet.

Derudover vil vi gerne sende en speciel tak til vores eksterne vejleder fra Danisco, Flemming Møller. Udover et brændende engagement i projektet viste Flemming sig at have styr på alting indenfor Danisco's del af branchen for tilsætningsstoffer i fødevarer, om det så gjaldt spegepølser, brød eller barriersystem, og hvis der ikke lige var svar på rede hånd, gik der ikke mange øjeblikke før han havde indhentet svar fra brødspecialisten Rie, sprøjteeksperten Casper eller spegepølsefolkene Lionel og Troels, så også en stor tak til dem.



# Indhold

---

<b>1</b>	<b>Indledning</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Disposition . . . . .	3
<b>I</b>	<b>Domænebeskrivelse</b>	<b>5</b>
<b>2</b>	<b>Terminologiliste</b>	<b>9</b>
<b>3</b>	<b>Facetter</b>	<b>11</b>
3.1	Interessenter . . . . .	11
3.2	Konceptkarakteristik . . . . .	14
<b>4</b>	<b>Business processes</b>	<b>19</b>
4.1	Spegepølser . . . . .	19
4.2	Barriersystem på kiks . . . . .	20

4.3	Vandindhold i brød . . . . .	21
<b>5</b>	<b>Benyttet udstyr</b>	<b>23</b>
5.1	VideometerLab . . . . .	23
5.2	Matlab . . . . .	26
<b>6</b>	<b>Dataindsamling</b>	<b>27</b>
6.1	Processen til påføring af vandbarriere . . . . .	27
<b>II</b>	<b>Kravspecifikation</b>	<b>31</b>
<b>7</b>	<b>Business process re-engineering</b>	<b>35</b>
7.1	Spegepølser . . . . .	35
7.2	Barriersystem på kiks . . . . .	36
7.3	Vandindhold i brød . . . . .	36
<b>8</b>	<b>Domænekrav</b>	<b>37</b>
8.1	Spegepølser . . . . .	37
8.2	Barriersystem på kiks . . . . .	37
8.3	Vandindhold i brød . . . . .	38
<b>9</b>	<b>Brugergrænsefladekrav</b>	<b>39</b>
9.1	Menneske-maskin dialog . . . . .	39
9.2	Menneske-maskin fysiologisk interaktion . . . . .	40
9.3	Datainitialisering . . . . .	40



<b>10 Maskinkrav</b>	<b>41</b>
10.1 Generelt . . . . .	41
10.2 Ydeevne . . . . .	42
10.3 Platform . . . . .	42
<b>11 Projektfokus og -afgrænsning</b>	<b>43</b>
<b>III Teori</b>	<b>45</b>
<b>12 Multispektral billedteori</b>	<b>49</b>
12.1 Multispektrale billeder . . . . .	49
12.2 Eigenverdier og -vektorer og Principal Komponent Analyse . . .	50
12.3 Principal Komponent Analyse på multispektrale billeder . . . . .	53
12.4 Minimum autokorrelations faktorer . . . . .	55
12.5 Falsk farvesammensætning . . . . .	58
12.6 Densitets billeder, score billeder og plots . . . . .	58
12.7 Residual billeder . . . . .	60
<b>13 Regressionsmetoder</b>	<b>67</b>
13.1 Krydsvalidering . . . . .	67
13.2 Least Angle Regression - Elastic Net . . . . .	68
13.3 Multivariat lineær billed-regression . . . . .	74
13.4 Multivariat logistisk billed-regression . . . . .	76

---

<b>IV</b>	<b>Dataanalyse</b>	<b>79</b>
<b>14</b>	<b>Preprocessering</b>	<b>83</b>
14.1	Fjern ”døde” pixels . . . . .	84
14.2	Find forgrund/baggrund . . . . .	84
14.3	Generér ROI . . . . .	87
14.4	Fjern kant på ROI . . . . .	89
14.5	Fjern baggrund . . . . .	89
14.6	Beskær billedet . . . . .	90
14.7	Resultat . . . . .	90
<b>15</b>	<b>Spegepølser</b>	<b>93</b>
15.1	Analyse af de rå billeder . . . . .	93
15.2	Forbehandling af spegepølsebilleder . . . . .	94
15.3	Kød/fedt adskillelse . . . . .	103
15.4	Farveskala for spegepølse . . . . .	114
15.5	Segmentering af fedt . . . . .	119
15.6	Præsentation af resultater . . . . .	124
15.7	Diskussion af resultater . . . . .	130
<b>16</b>	<b>Barriersystem på kiks</b>	<b>135</b>
16.1	Barrirelagets målte tykkelser . . . . .	136
16.2	Eksplorativ datasætanalyse . . . . .	138
16.3	Opstilling af model for barrirelagstykkelse . . . . .	139

---

16.4	Resultater af modelgenereringen . . . . .	143
16.5	Diskussion af resultater . . . . .	149
<b>17</b>	<b>Vandindhold i brød</b>	<b>151</b>
17.1	Datasætbeskrivelse . . . . .	151
17.2	Log-transformation af responset . . . . .	155
17.3	Featureudvælgelse . . . . .	156
17.4	Variation af modelparametre . . . . .	156
17.5	Brødets snitflade . . . . .	161
17.6	Resultater . . . . .	163
<b>V</b>	<b>Implementering</b>	<b>175</b>
<b>18</b>	<b>Preanalyse og preprocessering</b>	<b>179</b>
<b>19</b>	<b>Spegepølser</b>	<b>183</b>
<b>20</b>	<b>Barriersystem på kiks</b>	<b>187</b>
20.1	Modelgenerering . . . . .	188
20.2	Modelanvendelse . . . . .	190
<b>21</b>	<b>Vandindhold i brød</b>	<b>193</b>
<b>VI</b>	<b>Brugervejledning</b>	<b>197</b>
<b>22</b>	<b>Installationsvejledning</b>	<b>201</b>

<b>23 Spegepølser</b>	<b>203</b>
<b>24 Barriersystem på kiks</b>	<b>205</b>
<b>25 Vandindhold i brød</b>	<b>207</b>
<b>VII Opsummering</b>	<b>211</b>
<b>26 Konklusion</b>	<b>213</b>
26.1 Spegepølser . . . . .	214
26.2 Barriersystem på kiks . . . . .	215
26.3 Vandindhold i brød . . . . .	215
<b>27 Fremtidige forbedringer</b>	<b>217</b>
27.1 Generelt . . . . .	217
27.2 Spegepølser . . . . .	218
27.3 Barriersystem på kiks . . . . .	219
27.4 Vandindhold i brød . . . . .	220
<b>Litteratur</b>	<b>221</b>
<b>Figurer</b>	<b>225</b>
<b>Tabeller</b>	<b>237</b>

---

<b>VIII Bilag</b>	<b>241</b>
<b>A VideometerLab</b>	<b>243</b>
<b>B Spegepølser - rådata</b>	<b>245</b>
<b>C Spegepølser - PCA</b>	<b>251</b>
<b>D Spegepølser - MAF</b>	<b>253</b>
<b>E Spegepølser - Resultater</b>	<b>255</b>
E.1 Sample 1 . . . . .	255
E.2 Sample 2 . . . . .	259
E.3 Sample 3 . . . . .	262
E.4 Sample 4 . . . . .	265
E.5 Sample 5 . . . . .	268
<b>F Spegepølser - Test af adaptive thresholding</b>	<b>271</b>
F.1 Variation af vinduesstørrelsen . . . . .	271
F.2 Variation af tolerance-konstanten . . . . .	274
<b>G Barrieresystem rapport - Casper Lund</b>	<b>277</b>
<b>H Kiks - rådata</b>	<b>281</b>
<b>I LARS-EN analyseresultater af brøddata</b>	<b>291</b>
<b>J Kildekode</b>	<b>301</b>

Indeks

303

# Indledning

---

Dette projekt omhandler emnet multispektral billedbehandling af fødevarer og er udarbejdet i samarbejde med Danisco A/S, der bl.a. fremstiller ingredienser til fødevarer. De tre emner der vil blive behandlet i projektet er:

- Kødfarven af spegepølser.
- Estimering af lagtykkelsen for vandbarrieren til kiks.
- Estimering af vandindholdet i brød.

Det første datasæt bestod i nogle forskellige billeder af spegepølser og nogle få kiks med vandbarriere. Efterfølgende var vi på besøg hos Danisco A/S i Brabrand, hvor et komplet sæt kiks med vandbarriere i forskellige lagtykkelser blev fremstillet. Undervejs modtog vi så det endelige spegepølse datasæt, og til sidst har vi modtaget et datasæt med billeder af brødskiver og deres tilhørende målte vandindhold.

Ved starten af projektet har vi fået udleveret en Matlab toolbox med grundlæggende funktioner til billedbehandling af multispektrale billeder, deriblandt funktioner til at indlæse, skrive og vise de multispektrale billeder. Toolboxen er udviklet af Videometer A/S. Derudover benytter vi også en anden toolbox

udviklet af Karl Sjöstrand [Sjöstrand, 2006] der indeholder funktioner til regressionsanalyse.

## 1.1 Motivation

Dette afsnit giver en kort introduktion til hvad der ønskes udviklet, og hvorfor, indenfor de tre nævnte fødevarer emner.

**Spegepølser.** En spegepølser farve varierer fra kanten og ind mod midten af pølsen. Det er denne farveudvikling og farven generelt Danisco bl.a. styrer med deres tilsætningsstoffer, og de vil derfor gerne have en robust metode til at kvantificere farven, da spegepølsefarven er en af de vigtigste salgsparmetre ud over smagen. Der skal derfor fremstilles en metode til at give et kvantitativt mål for den farve en given spegepølse har i midten og ud mod kanten. For at gøre dette skal fedtet skilles fra kødet, også de steder hvor fedt og kød overlapper.

**Barrieresystem på kiks.** Danisco A/S har udviklet et vandbarrieresystem (Grindsted 1000 og 2000) til bl.a. kiks. Vandbarrieren skal adskille fødevarer med højt vandindhold fra fødevarer med meget lavt indhold af vand. Dette skal sikre at vandet ikke trænger ned i det påførte emne, og dermed gør det blødt og uappetitligt. Vandbarrieresystemet forlænger ved den rette barrieretykkelse typisk holdbarheden til cirka det dobbelte. I dette tilfælde kigger vi på kiks, der jo som bekendt er meget lidt vandholdige. Vandbarrieren påføres kiksen ved en sprøjteprocess hvorved der indføres en række usikkerhedsfaktorer. Man kunne f.eks. forestille sig at laget ikke var jævnt påført og derved ikke havde den samme høje effekt over hele kiksen, eller at laget simpelthen ikke er lige så tykt som beregnet. Der skal derfor udvikles en metode til at vise hvor jævnt laget er påført, og vise hvor på kiksen der er huller i laget.

**Vandindhold i brød.** For at et brød kan holde sig friskt så længe som muligt tilsættes en række tilsætningsstoffer. Nogle af disse stoffer er med til at hæmme mug dannelse m.m. Lige så vigtigt er det at brødet føles friskt og ikke tørrer ud. Til dette formål tilsættes en række enzymer og emulgatorer. For at finde frem til den mest optimale kombination af disse, måles vandindholdet af brødet over tid. Der findes i dag flere metoder til at finde frem til indholdet af vand i brødet, men de er både langsomme og omstændige. Derfor ønskes en metode til at udføre denne analyse vha. billedbehandling.



## 1.2 Disposition

Dette afsnit beskriver den overordnede dokumentstruktur, og giver en introduktion til hver af de enkelte dele.

**Del I, Domænebeskrivelse.** Domænebeskrivelsen beskriver domænet. Her ved forstås en nærmere beskrivelse af de forskellige aktører og deres interesse i projektet samt konceptkarakteristikker for de tre fødevareemner. Derudover gives en beskrivelse af de procedurer der anvendes i dag til bestemmelse af udviklingen for de ønskede egenskaber.

I afsnittet vil der også blive givet en beskrivelse af det benyttede udstyr, samt den procedure der blev anvendt til fremstilling af datasættet for kiks med barriere.

**Del II, Kravspecifikation.** I denne del vil kravene til den udviklede software og den nødvendige hardware blive beskrevet. Derudover beskrives projektets fokus og afgrænsning.

**Del III, Teori.** Denne del beskriver den teori der er blevet anvendt under analysen. Delen er overordnet delt i to kapitler, hvor det første beskriver den multisppektrale billedteori, og det andet er en beskrivelse af de anvendte regressionsmetoder.

**Del IV, Dataanalyse.** Dataanalysen indeholder beskrivelsen af hvordan de enkelte fødevarer grupper er blevet behandlet, fra de rå input billeder fra VideometerLab'en til de færdige analyseresultater.

**Del V, Implementering.** Denne del beskriver de overordnede kodestrukturer for den udviklede software. Kodestrukturene er tænkt som introduktion for en eventuel videreudvikling af softwaren.

**Del VI, Brugervejledning.** Denne del giver en praktisk anvendelig introduktion til den udviklede software, med eksempler der er til at benytte direkte i Matlab.

**Del VII, Opsummering.** I opsummeringen gives konklusionen for det samlede projekt og forslag til forbedringer der kunne tilføjes projektet i fremtiden.



Del I

# Domænebeskrivelse



Denne del beskriver forskellige aspekter af det valgte domæne. Først gives en terminologiliste der lister forskellige domæne begreber og vigtige termer brugt gennem rapporten. Derefter vil forskellige facetter af domænet blive behandlet.



# Terminologiliste

---

Dette kapitel indeholder en liste over begreber, forkortelser og vigtige termer brugt igennem rapporten. På denne måde sikres uniformitet i de efterfølgende kapitler og en fælles forståelse udviklere og andre interessenter imellem.

- **Barrierelag** - Vandbarrierelag af Grindsted Barrieresystem 2000, påført en kiks. Barrierelaget består af lipider og voks.
- **Bånd** - En bølgelængde af lys. Bruges om de 18 bølgelængder, eller bånd, som VideometerLab systemet optager billeder med. Ifm. billederne bruges ordet *bånd* om alle pixel-værdierne for en bølgelængde, der tilsammen udgør et 2D gråtone billede.
- **Egenskaber** - Vi anvender begrebet ”egenskaber” og komponenter for det som i litteraturen indenfor PCA og MAF også kaldes for signaler.
- **Eigenverdier og -vektor** - bruges som det fordanskede eigenvalue og -vector.
- **Features** - Fælles betegnelse for de statistiske værdier, gennemsnit, standardafvigelse og fraktiler udregnet på baggrund af hvert enkelt bånd eller forhold mellem bånd i et billede.

- **KAS toolbox** - Toolbox udviklet af Karl Sjöstrand [Sjöstrand, 2006]. Toolboxen indeholder Matlab implementeringer af LARS og LARS-EN algoritmerne.
- **Komponent** - Ved PCA og MAF analyse fås et multispektralt billedes komponenter. Ifm. billeder bruges ordet *komponent* om det 2D gråtone billede der fremkommer ved at gange hver enkelt komponent på det tilhørende bånd i det multispektrale billede.
- **LOO-krydsvalidering** - *Leave-One-Out-krydsvalidering* er den simpleste form for krydsvalidering, hvor hver enkelt sample udelades fra træningsdata for modelgenereringen en af gangen, og modellen med den mindste fejl-estimering vælges.
- **MAF** - *Minimum/Maximum Autocorrelation Factors*.
- **MSI** - *MultiSpectral Imaging* er den engelske oversættelse af multispektral billedanalyse.
- **PCA** - *Principal Component Analysis* (Principal Komponent Analyse). I teksten vil det ofte blive skrevet "PCA analyse" fordi det sprogmæssigt er pænere og mere læsbart.
- **ROI** - *Region Of Interest*, det engelske udtryk for de områder/pixels af et billede, der er relevant i en given sammenhæng. Typisk vil dette være et binært billede bestående af 1'taller for det område man er interesseret i, og 0'er for det der skal skæres væk.
- **Salami** - Salami er brugt som et andet ord for *spegepølse*. Ordet *Salami* har den evne at det staves og forstås ens på dansk og engelsk, og er derfor anvendeligt ifm. f.eks. kodebeskrivelserne.



## KAPITEL 3

# Facetter

---

I dette kapitel kigger vi på forskellige facetter<sup>1</sup> af projektet. Først præsenteres de forskellige interessenter der har indvirken på projektet, hvad deres interesser består i samt hvad de ønsker at få ud af projektet. Derefter præsenteres de koncepter der behandles igennem projektet, herunder en præsentation af de fødevarer der vil blive analyseret.

### 3.1 Interessenter

Dette afsnit beskriver de eksterne parter der har interesse i projektet, dvs. hvad de forskellige parter ønsker at få ud af projektet. De to hovedaktører er Danisco A/S der fremstiller tilsætningsstoffer, ønsker at få suppleret de eksisterende analysemetoder med billedbehandlings metoder, og Videometer A/S der har udviklet VideometerLab systemet og den tilhørende Matlab MSI-toolbox.

---

<sup>1</sup>Ordet *facet* kommer oprindeligt fra det engelske *face*, ansigt. Det bruges f.eks. om fladerne på en slebet diamant, hvor hver flade er en *facet*. I denne kontekst er det derfor forskellige sider eller ansigter af projektet.

### 3.1.1 Danisco A/S

Danisco er for den almindelige forbruger nok mest kendt for at producere sukker og andre sødemidler, men deres produktsortiment inkluderer bl.a. også ingredienser og tilsætningsstoffer til en lang række fødevarer indenfor så godt som alle grene af fødevarerindustrien. For at kunne sælge disse produkter skal Danisco have bedst mulig dokumentation for at produkterne har den virkning de påstår og i forbindelse med udvikling og videreudvikling er objektive, reproducerbare og sammenlignelige målinger yderst vigtige.

For vandbarrieren til kiks ønsker Danisco at kunne dokumentere kvaliteten af påføringsprocessen og dermed bestemme, hvor meget barrierelag, der er nødvendigt for at opnå den ønskede effekt da prisen naturligvis stiger for kunden, jo mere barrierelag der skal tilføjes.

Den måde man vurderer spegepølser på er meget omkostningsfuld pga. at de udelukkende skal vurderes af professionelle slagtere og andre med speciale indenfor kød. Ved at benytte multispektral billedanalyse er det ønsket at få en billigere, mere objektiv og reproducerbar metode til at analysere og vurdere spegepølserne.

Når Danisco idag skal måle vandindholdet i brød foregår det med målemetoder der typisk er relativt tidskrævende (>1 time pr prøve). Med en målemetode baseret på billedanalyse ønskes det at gøre den enkelte måling langt hurtigere og gøre processen ved mange målinger simplere. De anvendte analysemetoder ødelægger desuden prøven, så det ikke er muligt at foretage flere målinger af samme skive, hvilket er en væsentlig variationskilde.

Danisco har forskellige andre målemetoder til analyse af fødevarer, som f.eks Storm P. agtige løsninger som f.eks. den kunstige tommelfinger der måler elasticitet af bl.a. brød og kiks, og mikroskopiske analyser til f.eks. analyse af den kemiske sammensætning. De har ikke tidligere benyttet analyse af billeder i det synlige spektrum og NIR-området<sup>2</sup> som det gøres i dette projekt.

### 3.1.2 Videometer A/S

Videometer udvikler bl.a. systemer til multispektral billedanalyse i industrien, både direkte i forbindelse med produktion og til analyse og videreudvikling af produkter, og det er dem der har udviklet VideometerLab systemet. Områderne for anvendelse af deres systemer spænder bredt fra plastindustrien, medicin,

<sup>2</sup>Nær Infrarød området betegner bølgelængder fra ca. 750-1400nm

sortering af minkpelse og *visuel vejning* af grise. Analyse af fødevarer med multispektral billedanalyse er et relativt nyt område med et stort potentiale der i øjeblikket er i høj vækst. Der er mange årsager til at det går godt på området, bl.a. fordi almindelige computere er kommet så højt op i hastighed at man ikke skal have specielt udstyr til at behandle billederne, så omkostningerne på den måde bliver holdt nede.

Videometer ønsker at undersøge mulighederne for at benytte multispektral billedanalyse af fødevarer til at finde generelle modeller til bestemmelse af forskellige fødevars egenskaber. Som forbruger er synssansen den vigtigste kvalitetsparameter vi har at gå efter når vi handler ind.<sup>3</sup> Derfor er det nærliggende at benytte billedbehandlings teknikker til at vurdere og udlede de egenskaber vi er interesseret i.

Fødevarer består hovedsagligt af forskellige typer af organisk kemi, der har den egenskab at absorbere lys forskelligt ved bølgelængderne 400-1000nm, og kan på den måde skilles ad med multispektral billedanalyse af disse bølgelængder. Derudover baner den multispektrale billedanalyse af fødevarer vejen for at man kan analysere en type fødevarer uden at påvirke varen, og på den måde undgå at man får fortyndet evt. urenheder, da man direkte kan se disse i hver pixel.

### 3.1.3 Fødevarerindustrien

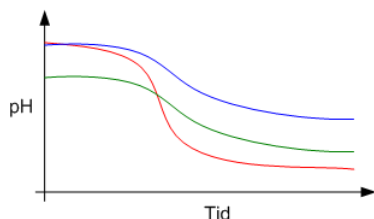
De ønsker at kunne sammenligne Danisco's produktudbud, ud fra nogle konsistente, sammenlignelige og troværdige tal. De ønsker også at kunne sammenligne et produkt de allerede køber af Danisco med et tilsvarende produkt med ændrede og/eller forbedrede karakteristikker.

For spegepølserne ønsker de at blive præsenteret for tal der viser hvor rød kød-farven vil blive hvis de benytter én bakteriekultur frem for en anden, da det har stor betydning for forbrugernes oplevelse af spegepølserne.

For vandbarrierelaget er de interesserede i at få de systemopsætningsparametre fra Danisco, der sikrer den bedst mulige påførsel af barrierelaget, så det undgås at der kommer huller eller fordybninger i barrierelaget der kunne gøre det lettere for vand at trænge igennem.

---

<sup>3</sup>Vi benytter naturligvis også vores føle og lugtesans, men en vare skal trods alt først fange og fastholde vores blik før vi tager den op og benytter de andre sanser i kombination til at vurdere om vi vil købe den.



**Figur 3.1:** Udvikling af spegepølsens pH-værdi som funktion af tid. pH-værdien falder som funktion af tiden fordi de tilsatte bakteriekulturer udvikler så meget mælkesyre at de efterhånden selv dør ud. De tre farvede linjer på figuren er eksempler på karakteristikkene for forskellige bakteriekulturer.

For brødet er de interesseret i at få brød der holder på vandet i længere tid, sådan at brødet holder sig friskt længere. Derfor er de interesserede i at Danisco udvikler tilsætningsstoffer til deres brød så det holder på vandet.

## 3.2 Konceptkarakteristik

Dette afsnit beskriver de fødevareremner som vil blive analyseret, hvad vi overordnet vil analysere ved hver enkelt og hvorfor.

### 3.2.1 Spegepølser

For at spegepølsen får den rigtige smag, farve og konsistens tilføres en specifik bakteriekulturer og en smule sukker. Når bakterierne spiser af sukkeret dannes der mælkesyre som får pH-værdien til at falde. Ved lagringen danner bakterierne samtidig enzymer og der sker en enzymatisk nedbrydning af proteinet i kødet, hvorved pH-værdien stiger svagt. Samlet set falder pH-værdien dog så meget at den tilsatte bakteriekultur efterhånden dør hen. Dette er illustreret på figur 3.1, hvor hver farve viser karakteristikkene for tre forskellige bakteriekulturer. Ved denne process får spegepølsen den rigtige farve inderst og yderst, og kød og fedt bliver bundet sammen, sådan at den korrekte konsistens opnås og pølsen ikke smuldrer og falder fra hinanden når man tager skindet af den. Valget af hvilken bakteriekultur der vælges og dermed hvilken endelig pH værdi, konsistens og farve der opnås er hovedsageligt pølseproducenternes. Det er samtidig meget geografisk relateret, f.eks. har de franske pølser en generelt højere pH-værdi end de danske.

Pølsen får ved denne proces en mørkere brunlig farve yderst, og en lysere mere rødlig farve inderst. Den vigtigste egenskab ved pølsens udseende er netop denne inderste nuance af rød. Fedtet er i denne sammenhæng ikke interessant at kigge på, og man ønsker derfor en separat vurdering af kødfarven.

### 3.2.2 Barrieresystem på kiks

Kager, chokoladebarer og lignende består ofte af et eller flere lag kiks med en blødere substans imellem, det kunne f.eks. være chokolade, nougat, syltetøj, creme, ost, is, karamel osv. Disse ting indeholder vand, der kan trænge ind i kiksken som indeholder meget lidt vand, og derved gøre den blød og uappetitlig. Kan man forhindre vandet i at trænge ind i kiksken vil den bevare sin sprødhed og varen vil kunne have en længere holdbarhedsdato. Dette kan opnås ved at påføre kiksken en vandbarriere.

#### Grindsted BarrierSystem 2000

Vandbarrieren - Grindsted BarrierSystem 2000 (BS2000) - består af den fødevarer godkendte emulgator Grindsted ACETEM<sup>4</sup> og bivoks der kan påføres kiksken<sup>5</sup> i et lag på ned til  $50\mu\text{m}$  i tykkelsen<sup>6</sup>.

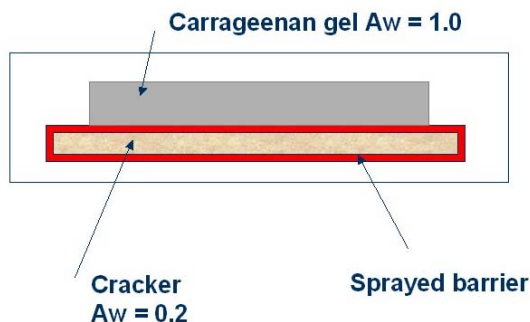
Det er meget vigtigt at barrierelaget er intakt uden huller og jævnt påført, da eventuelle huller eller lokale områder med et markant tyndere barrierelag vil forringe barrierelagets effekt dramatisk. BS2000 er designet til at være mekanisk fleksibel sådan at sprækker og huller ikke opstår i barrierelaget efter påførsel. Til gengæld er der stadig ved selve påførslen en risiko for fejl eller ujævn påføring. Bivoks og chokolade med højt smeltepunkt er i andre sammenhænge benyttet som vandbarriere og anvendes stadig, men har desværre vist sig at have den uheldige effekt at de ikke er særligt mekanisk stabile og derfor har tendens til at danne sprækker og huller [IFI-Online, 2004].

I hurtige snacks, der i stigende grad erstatter vores måltider, er fedtindholdet en meget vigtig parameter. Selvom BS2000 mest af alt består af fedtstof og derved hæver det generelle fedtindhold i snacken kan man ved at påføre vandbarrieren

<sup>4</sup>ACETEM (Acetic acid esters) er en film-dannende fleksibelt molekyle der primært benyttes til barrierer af forskellig art [Danisco, 2005]

<sup>5</sup>BS2000 kan også benyttes til andre typer af fødevarer, men i denne sammenhæng vil vi kun kigge på kiks.

<sup>6</sup>Vores forsøg viste dog at vi kun kunne opnå lag på ned til ca  $130\mu\text{m}$ . Hvis det blev tyndere end dette, blev laget meget ujævnt. Danisco har dog tidligere haft held med lagtykkelser ned til  $50\mu\text{m}$



**Figur 3.2:** Nuværende testmetode af vandbarriere til kiks.

på kiksene nedsætte fedtindholdet i det lag man lægger ovenpå og derved sænke fedtindholdet af den samlede snack.

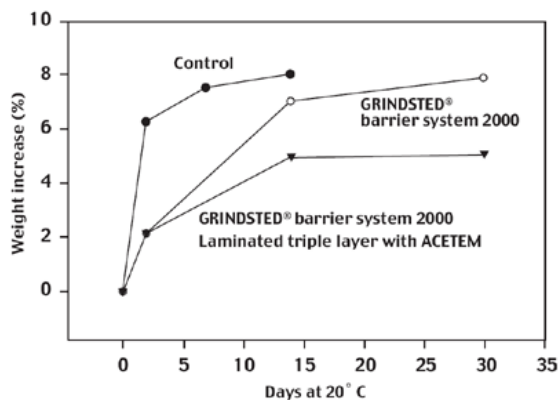
Måden man tester hvor tykt laget skal være på kiksene før det holder tæt, er ved at lægge en vandholdig gel ovenpå en kiks påført barrieresystemet som illustreret på figur 3.2, og derefter veje vandindtaget i kiksene over tid.

På figur 3.3 ses resultatet af en forsøg hvor barrieresystemet er blevet anvendt som vist på figur 3.2. Gelen havde en vandaktivitet<sup>7</sup> på 0,54. Kiksene med en vandaktivitet på 0,2 og den pålagte gel, blev opbevaret ved 20°C over en periode, og man målte vandoptaget i kiksene undervejs ved at veje den. Uden vandbarriere havde kiksene en levetid på ca 10 dage, mens den blev fordoblet ved brug af et 400µm vandbarrierelag [FPI, 2006]. Udover at veje kiksene, og på den måde få et tal for hvor meget af vandet kiksene har indtaget, kan man også måle kiksens struktur med en struktur analysator der trykker på kiksene for at finde ud af hvor sprød den er.

### 3.2.3 Vandindhold i brød

En af de helt store udfordringer indenfor brødfremstilling er at få brødet til at virke friskt selv efter flere dage. Derfor anvendes der konserveringsmidler som hæmmer dannelse af mug m.m., men lige så vigtigt er at brødet så vidt muligt bevarer fornemmelsen af friskt brød og ikke tørrer ud. For at opnå dette tilsættes forskellige enzymer og emulgatorer i brødet. Derefter er det nødvendigt

<sup>7</sup>En vandaktivitet på 1 er noget af det tætteste man kan komme på rent vand når det samtidig skal opføre sig mekanisk som en gel. En vandaktivitet på 0,54 er et mere realistisk eksempel på et reelt produkt.



**Figur 3.3:** Resultatet af et forsøg med barrierelag som vist på figur 3.2. Barrierelaget fordoblede holdbarheden for kiksken med et barrierelag på  $400\mu m$ .

at analysere brødene på forskellig vis for at kunne fastslå hvilke kombinationer af enzymer og emulgatorer og i hvilke mængder, der giver det bedste resultat.

Ældningsprocessen af brødet sker hovedsageligt som følge af vandfordampningen og retrograderingen. Retrograderingen beskriver hvordan det vand der er bundet i stivelsesstrukturene bliver trukket over i proteinstrukturene i brødet.





## KAPITEL 4

# Business processes

---

Business Processes, eller forretningsprocesser på dansk, er de metoder der benyttes i øjeblikket. Dette kapitel beskriver de forretningsprocesser der benyttes i øjeblikket til test af de ønskede egenskaber for de tre ovenfor nævnte fødevarergrupper.

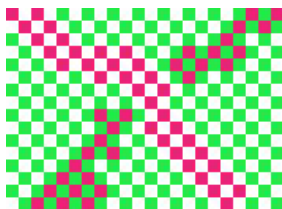
### 4.1 Spegepølser

I dag kan man måle farveintensiteten i et område med en Minolta-farvemåler. En Minolta-farvemåler er et instrument der kan måle den gennemsnitlige farveintensitet i et område på ca  $1\text{cm}^2$ . Dette er dog problematisk med spegepølser, da man med et sådant apparat utvivlsomt vil få en del af det lysere fedt med i analysen af det valgte område.

Derudover bruger man ofte subjektive sensorikdata<sup>8</sup> som *pink*, *vinrød*, *mat*, *blank* til beskrivelse af farven. Disse sensorikdata er ikke udpræget reproducerbare, da de afhænger af den enkelte slagter, hvorved man introducerer en masse skjulte

---

<sup>8</sup>Sensorikdata genereres af et dommer panel bestående af eksperter der bedømmer fødevareren ud fra en række listede kriterier.



**Figur 4.1:** Det menneskelige syn snyder når forskellige farver ligger tæt op af hinanden.

menneskelige faktorer, da den enkelte slagters bedømmelse af kødfarven kan være under indflydelse af bl.a. slagterens momentane humørtilstand.

Det menneskelige syn er også en betydende faktor. Som mennesker opfatter vi farver meget forskelligt, alt efter hvilke farver der ligger tæt op af den farve man forsøger at vurdere. Dette er illustreret på figur 4.1 hvor øjet snydes til at se de røde tern som enten lysere eller mørkere alt efter om de ligger op af de grønne eller de hvide tern. Dette er et ekstremt tilfælde, men selvom fedtet i en pølse ikke er grønt vil øjet på samme måde opfatte kødfarven som værende lysere jo mere fedt der er i pølsen.

Desuden kan det være svært at skelne mellem fedt og kød, da disse to klasser kan overlape en del, forstået på den måde at man nogen steder kan se en klump fedt igennem et tyndt lag kød som næsten er gennemsigtigt, eller omvendt kan se kød under et tyndt lag fedt. Alle disse ting spiller altså ind når det menneskelige øje skal vurdere farven af kødet.

## 4.2 Barriersystem på kiks

Som beskrevet ovenfor findes der metoder til at teste om barrierelaget er tykt nok til at holde vand og emne adskilt i den ønskede periode. Tilgængelig findes der ikke på nuværende tidspunkt metoder til at vurdere om det på emnet påførte barrierelag er jævnt, eller indeholder huller eller fordybninger i laget. Som det vil blive beskrevet i afsnit 6.1 er der mange faktorer der spiller ind når barrieren påføres et emne. Selv om mange af disse vil kunne fjernes under en permanent opstilling vil der stadig være en del parametre tilbage.

Måden man idag estimerer hvor tykt et barrierelag der er påført kiks, er ved at veje den før og efter påførelsen. Differencen mellem de to, er derved mængden af den påførte barriere. Derefter kan man estimere højden af barrierelaget med

formlen

$$(4.1) \quad \text{TYKKELSE} = \frac{\Delta\text{VÆGT}}{\text{AREAL} * \text{DENSITET}}$$

Hvor TYKKELSE er tykkelsen, eller højden, af barrierelaget i  $cm$ ,  $\Delta\text{VÆGT}$  er vægten af det påførte barrierelag i gram, AREAL er arealet for det område der er påført barrierelag i  $cm^2$  og DENSITET er barrierelagets densitet. Densiteten for BS2000 er på  $0.9 \frac{g}{cm^3}$ .

### 4.3 Vandindhold i brød

Vandindholdet i en brødskeve bestemmes som en procentdel af den samlede vægt. For at måle vandindholdet udstandses det stykke af brødskeven man vil måle, typisk en cirkel med en diameter på 5cm i midten af brødet eller en del af kanterne, og prøven anbringes i en specielt varmeovn med indbygget vægt, der så vejer prøven før og efter en udtørring hvor alt vand fordampes. Udtørringen tager i de anvendte maskiner (HB43 og LP16) typisk 1-2 timer. Vægtforskellen på før og efter svarer til det fordampede vand. Vandfordelingen er svær at bestemme i brød, dels fordi usikkerheden stiger med faldende prøvestørrelse og dels fordi det rent praktisk vil være et meget stort arbejde og/eller kræve meget udstyr at skulle foretage mange målinger af 1-2 timers varighed for hver enkelt skive brød.

Retrogradering beskriver den process der sker når brød ældes og vandet trækkes fra stivelsesfasen over i proteinfasen og får brødet til at virke mere tørt, selvom vandindholdet ikke er ændret voldsomt. Samtidig med retrograderingen vil der dog også ske en fordampning. Retrograderingen kan bestemmes ved en metode kaldet DSC (*Differential Scanning Calorimetry*), der måler den mængde energi der skal til for at smelte retrograderet amylopektin<sup>9</sup>, jo mere retrograderet amylopektin et stykke brød indeholder, jo mere energi skal der tilføres for at smelte det. Lidt mere besværligt kan retrograderingen også bestemmes ved LF-NMR (*Lav-Felts Nuklear Magnetisk Resonans*), hvor det udnyttes at faste og flydende stoffer kan skelnes fra hinanden når de udsættes for nogle veldefinerede magnetfelter.

---

<sup>9</sup>Det komplekse kulhydrat stivelse indeholder to typer molekyler; Amylose (ca. 15-30%) og Amylopektin (ca. 70-85%). Amylopektin er opbygget af et stort antal glukosemolekyler. I modsætning til amylose er amylopektin et forgrenet molekyle.



## KAPITEL 5

# Benyttet udstyr

---

Dette kapitel beskriver det udstyr der er anvendt til data indsamling og behandling.

### 5.1 VideometerLab

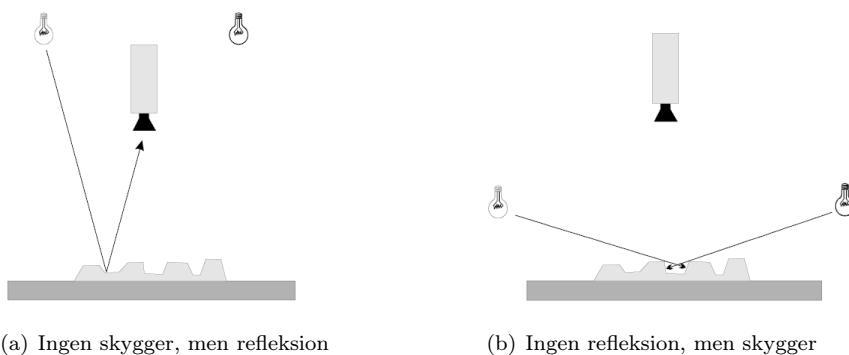
Til at optage billederne har vi benyttet et *VideometerLab* system som vist på figur 5.1.

*VideometerLab* system benyttes til at tage multispektrale billeder til brug i laboratorier. Kameraet måler lysintensiteten fra det fotograferede objekt ved forskellige bølgelængder af lys, fra det nær infrarøde område (NIR) og igennem hele det synlige spektrum. Det system vi har benyttet tager billeder ved 18 forskellige bølgelængder af lys. Se bilag A for den komplette liste over bølgelængder og andre data for *VideometerLab* systemet.

Systemet er specielt designet til at sikre reproducerbare billeder med total diffus belysning, uden skygger og uden spejlende refleksioner. Ifølge [Hansen, 1999] er problemet med belysningen ved normale omstændigheder at hvis belysningen kommer fra oven for at undgå skygger, resulterer det i spejlende refleksion af



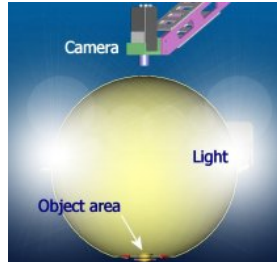
**Figur 5.1:** VideometerLab kameraet.



**Figur 5.2:** Skygger og refleksion ved forskellig belysning. Enten er der ingen skygger, men tilgængelig reflekteres lyset op i kameraet, eller også er det omvendt.

lyset direkte op i kameraet, og hvis det kommer fra siden for at undgå refleksion dannes der skygger på det belyste emne som illustreret på figur 5.2.

Den bedste løsning på dette problem er hvis lyset kommer fra alle retninger på samme tid. Dette er gjort ved at benytte en *Ulbricht kugle*. Kuglen er malet indvendigt med en mat titanium maling der sikrer en diffus og homogen belysning af emnet [Gomez et al., 2006]. I toppen er der et hul til kameraet, i bunden er der hul til emnet og i siderne er der huller til lyskilderne. Hele opstillingen med det sort/hvide kamera i toppen af kuglen er illustreret på figur 5.3. Når systemet tager billeder slutter det helt tæt til underlaget, sådan at der ikke kommer falsk lys ind på emnet. Kun i de tilfælde at emnet er for stort til at kunne ligge inde for systemets åbning, vil der kunne komme falsk lys ind på emnet og forstyrre optagelsen, hvilket derfor så vidt muligt skal undgås.



**Figur 5.3:** Tegning af et tværsnit af VideometerLab med kamera og lyssætning.

For at styre billedoptagelsen benyttes den medfølgende software der styrer både kalibrering og lyssopsætning af systemet. De digitale billeder der returneres fra systemet bliver gemt i *HIPS* format der direkte kan læses ind i Matlab via den medfølgende software pakke. Repræsentationen af billedet i Matlab er en tredimensionel matrice, hvor billedet  $\underline{G}$  med  $I$  rækker,  $J$  søjler og  $K$  bølgelængder repræsenteres ved

$$\underline{G} = I \times J \times K$$

Det system der er benyttet her, har en opløsning på  $960 \times 1280 \times 18$  hvor værdien af hvert element  $g_{ijk}$  i  $\underline{G}$  kan betragtes som en pixelværdi.

### 5.1.1 Kalibrering af kameraet

For at opnå den højest mulige reproducerbarhed, skal systemet kalibreres. Til systemet hører der tre farvede skiver med, en mørk, en lys og en lys med mørke prikker. Da systemet lukker helt til ned over skiverne, vil disse reference skiver altid give den samme kontrast reference, og dermed opnås den høje reproducerbarhed.

Systemet prøver så vidt muligt at undgå ”overeksponering” af billederne og man skal derfor inden en optagelse starter af en emne gruppe, opsætte lyssætningen vha. det lyseste emne man har. På den måde forsøges det at holde de resulterende digitale værdier mellem 0 og 100. Da der *kan* forekommer emner der visse steder er enten lysere eller mørkere end lysreferencen kan der altså forekomme værdier både over 100 og under 0 - selvom negative værdier er sjældent set.

## 5.2 Matlab

Til behandling af billederne benyttes programmet Matlab. I Matlab repræsenteres det multispektrale billede som en matrice af doubles, hvor den første dimension er rækkerne, den anden er kolonnerne, og den tredje er bølgelængden.

De billeder der kommer fra VideometerLab'en har en størrelse på

$$960 \times 1280 \times 18 = 22118400 \text{ pixels}$$

Når billedet læses ind i matlab fylder det derfor

$$22118400 \text{ pixels} \times 8 \frac{\text{Byte}}{\text{pixel}} = 176947200 \text{ Bytes} \approx 176,95 \text{ MB}$$

Ifølge [MathWorks, 2006] allokerer Matlab hukommelse på heap'en til et helt billede af gangen. Hvilket vil sige at billederne bliver lagt som en kontinuerlig mængde data. Der er ikke nogen automatisk *garbage collector* der sættes igang hvis hukommelsen er fragmenteret, og hvis man flere gange har allokeret et så stort billede, kan hukommelsen derfor være fuld af mindre huller, hvor det samlede antal frie Bytes er stort, men hvor der ikke er plads til et nyt billede i sin fulde længde. Derfor skal det undgås at allokere plads til for mange fulde billeder af gangen.

Den samlede mængde data Matlab kan benytte sig af, afhænger af størrelsen på dels computerens indbyggede hukommelse, i form af RAM, og dels størrelsen på swap-filen. På et 32bit system, er der dog en naturlig grænse på 4GB samlet hukommelse, da den maksimale hukommelses adresse er på  $2^{32}\text{bit} = 4294967296\text{bit} = 4\text{GB}$ . I Windows XP er hver enkelt process begrænset til 2GB hukommelse hvor de sidste 2GB er dedikeret til kernen.

Matlab har en indbygget funktion til at kunne lægge f.eks. en ekstra kolonne til en eksisterende matrice ved at man skriver hvilke data denne skal indeholde. Matlab gør så det at den allokerer et nyt område i hukommelsen til den nye matrice, og lægger de gamle og de nye data derover i hvorefter det gamle område bliver frigivet. Dette kan være et problem hvis det bliver gjort for mange gange i træk, da det meget hurtigt fører til den ovenfor beskrevne defragmentering af hukommelsen.



# Dataindsamling

---

Dette kapitel beskriver hvordan *kiks med vandbarriere* datasættet er blevet produceret. Billeder af brød og spegepølser har Danisco stået for.

## 6.1 Processen til påføring af vandbarriere

BS2000 opbevares i forbindelse med påføring i en beholder med dobbelt kappe, således at den kan holdes ved en konstant temperatur bestemt af vandet der ledes igennem yderkappen. Beholderen kan sættes under tryk eller holdes ved atmosfærisk tryk. BS2000 føres igennem en ventil ned til en dyse hvor det blandes med trykluft og sprøjtes ud som et gardin. Emnet føres forbi dette gardin af BS2000 på et transportbånd så hele overflades påføres en ensartet mængde. BS2000 påføres emnet flydende ved en temperatur på omkring 65-70 grader Celcius og størkner på få sekunder. Ændres hastigheden af transportbåndet ændres naturligvis også den tid emnet er under dysen og dermed mængden af påført BS2000. Forsøgsopstillingen ses på figur 6.1, og på figur 6.2 ses hvordan barrierelaget påføres Mariekiksen der kører forbi på transportbåndet under sprøjtedysen.



**Figur 6.1:** Forsøgsopstilling med forsøgsleder Casper Lund fra Danisco og Jesper Rasmussen.



**Figur 6.2:** Påførelse af barrieresystem på Mariekiks.

Det er en relativt følsom proces, hvor mange variable påvirker kvaliteten af det endelige resultat. Temperaturen af BS2000 og temperaturen af emnet det påføres påvirker hvor langt ind i emnet barrierelaget trænger. For at barrierelaget hæfter godt på emnet skal det trænge lidt ind, men det er mængden af barrierelag på overfladen af emnet, der afgør hvor godt barrieren vil virke. Temperaturen i rørene, der fører BS2000 fra beholderen til sprøjtedysen er i den forbindelse også vigtig, fordi laget bliver ujævnt hvis det når at køle for meget af igennem rørene, men det skal samtidig ikke varmes ekstra op, da det derved vil trænge længere ind i emnet og mindske effekten.

Kort opsummeret har processen følgende hovedparametre:

- Trykket i beholderen med BS2000
- Trykket på blandingsluften
- Temperaturen af emnet
- Temperaturen af BS2000
- Temperaturen af blandingsluften
- Temperaturen af rørene
- Rumtemperaturen
- Transportbåndets hastighed
- Dysehovedets udformning



(a) Jesper Rasmussen ved notesblokken og de forskellige kiksetyper



(b) Lars Nikolajsen viser nogle af de færdige resultater frem

**Figur 6.3:** De to forsøgsassistenter i arbejde og godt pakket ind.

Del II

# Kravspekifikation



Denne del indeholder de krav der bliver stillet til det udviklede software. Først beskrives de omarbejdede forretnings processer (Business Process re-engineering)<sup>10</sup>. Så beskrives de overordnede krav for hver af de tre emner, kiks med barriere, spegepølse og vandindhold i brød. Hvorefter de krav der stilles til softwarens brugergrænseflade listes. Dernæst listes de krav der stilles til den hardware som softwaren skal køre på. Til sidst beskrives opgavens fokusområde og afgrænsning.

---

<sup>10</sup>Den engelske term *Business process re-engineering* er efterhånden relativt almindeligt kendt, og vil blive brugt i resten af teksten om de omarbejdede forretnings processer.





## KAPITEL 7

# Business process re-engineering

---

Ved begrebet Business process re-engineering forstås de ændringer der indføres til de eksisterende forretnings processer. Dette afsnit beskriver hvordan de eksisterende forretnings processer beskrevet i kapitel 4 bliver lavet om med de ændringer der er foreskrevet nedenfor. I det følgende er ordet "system" brugt om den software pakke der implementerer disse ændringer.

### 7.1 Spegepølser

Systemet skal tage et vilkårligt multispektralt billede af en spegepølse som input og vise en opdeling af kød og fedt som to billeder. Derudover skal systemet vise kødfarven som en funktion af afstanden til midten, og fedtfarven skal vises som en enkelt værdi. Kødfarve-værdien skal være direkte sammenlignelig med farven fra andre spegepølser.

Dette skal medføre at man helt slipper for de begrænsede subjektive sensorik udtryk som kødspecialisterne bruger, og istedet får nogle robuste, sammenlignelige og reproducerbare informationer om spegepølsen.

## 7.2 Barrieresystem på kiks

For barrierelaget til kiks gælder det at der ikke på nuværende tidspunkt findes instrumenter til at kontrollere hvor jævnt barrierelaget er påført et emne, hvilket er et problem fordi der som beskrevet ovenfor er mange parametre der spiller ind når barrieren påføres et emne.

Der skal derfor laves et system der ved hjælp af et multispektralt billede kan bestemme lagtykkelsen på en kiks og vise den som et højdekort samt give et tal for hvor jævnt barrierelaget er påført og dermed give en robust måde at vurdere kvaliteten af påføringen af barrierelaget.

## 7.3 Vandindhold i brød

Ved at brød kan testes for vandindhold lettere og hurtigere vha. billedanalyse og flere gange for samme skive kan der spares ressourcer til de enkelte analyser og der kan analyseres større datasæt uden det vil tage længere tid. Desuden vil den ikke-destruktive billedanalyse give mulighed for at minimere den variationskilde der kommer af at skulle benytte forskellige skiver til forskellige tider, sådan at man kan se ældningsforløbet af den samme skive.

# Domænekrav

---

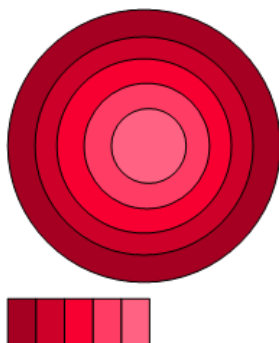
## 8.1 Spegepølser

Systemet skal ud fra et givent multispektralt billede af en spegepølse kunne beregne følgende parametre

- **Farveændring** - Vise med et billede og tal der angiver spegepølsens farveændring fra yderst til inderst separat for kødet i pølsen, som illustreret på figur 8.1.
- **Kød- og Fedt-farve** - Give et tal for hhv. kødets og fedtets gennemsnitsfarve.
- **Fedtstørrelser** - Vise størrelsesfordelingen af fedtområderne i spegepølsen som et histogram.

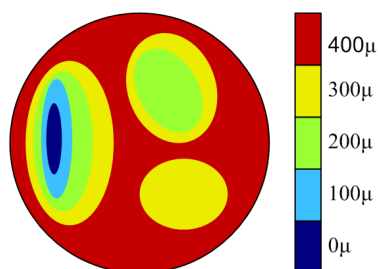
## 8.2 Barrieresystem på kiks

Systemet skal være i stand til ud fra et givent multispektralt billede af en kiks med barrierelag at beregne følgende parametre



**Figur 8.1:** Skala der viser en spegepølseres kødfarveintensitet fra yderst til inderst.

- **Barrirelagstykkelse** - Give et tal for hvor tykt det påførte barrirelag er.
- **Barrirelagets jævnhed** - Give et tal for hvor jævnt barrirelaget er påført kiksen - undersøge om der er steder på kiksen hvor der er påført mere barrirelag end på andre. Samtidigt skal barrirelagets tykkelse vises som et billede hvor intensiteten af farven i hver pixel udtrykker lagets tykkelse som illustreret på figur 8.2.



**Figur 8.2:** Skala der viser højdeforskelle for barrirelag på en kiks.

### 8.3 Vandindhold i brød

Systemet skal ud fra et givent multispektralt billede af en brødskive kunne beregne følgende parametre

- **Vandindhold** - Give et estimat af vandindholdet i et udvalgt område af brødskiven.

## KAPITEL 9

# Brugergrænsefladekrav

---

Dette kapitel beskriver de krav der stilles til brugergrænsefladen.

### 9.1 Menneske-maskin dialog

- Alle tekster fra systemet skal være på engelsk.
- Al koden skal være på engelsk.
- Der skal eksistere en "hjælp-tekst" for alle kodefiler.
- Brugeren skal guides med en hjælpende tekst hvis denne har angivet for få eller for mange parametre til en funktion, eller parametrene ikke har det korrekte format.
- Brugeren skal kunne indstille alle parametre der er væsentlige for den givne analyse, uden at skulle ændre i selve scriptet.

## 9.2 Menneske-maskin fysiologisk interaktion

- Alle funktioner skal kunne køres ved indtastning på tastatur eller klik med en mus. Hvis musen skal bruges skal brugeren guides ved en hjælpende tekst på skærmen som forklarer hvad der skal gøres.

## 9.3 Datainitialisering

- De indstillinger der anses for at være de mest anvendelige i de fleste situationer skal være valgt på forhånd sådan at brugeren kan køre funktionerne med mindst mulig viden om den samlede funktionalitet.

## KAPITEL 10

# Maskinkrav

---

I dette kapitel beskrives de krav der stilles til hardware og software.

### 10.1 Generelt

- Den eksekverende software skal være Matlab version 7, Release 14 eller nyere.
- Systemet skal have installeret den modificerede *MSI-toolbox* og *KAS-toolboxen*.
- Al kildekode leveres med denne rapport, som en "addon" værktøjskasse til den modificerede *MSI toolbox* og *KAS toolbox*.
- Al kildekoden består af Matlab scripts.
- De billeder der bruges som input skal være i *HIPS* formatet.

## 10.2 Ydeevne

- Det eksekverende system skal som minimum være en Pentium4, 2,4GHz eller tilsvarende med minimum 2GB ram. Hvis systemet har en *swap*-fil tæller denne med i de 2GB ram, dog nedsættes hastigheden markant og ligeledes hvis der anvendes en langsommere CPU.

## 10.3 Platform

- Platformen skal kunne være i stand til at eksekvere Matlab i den ovenfor nævnte version.



# Projektfokus og -afgrænsning

---

Dette kapitel beskriver rapportens fokusområde og afgrænser opgaven til kun at indeholde dette.

Vi vil benytte de multispektrale billeder vi får fra VideometerLab. Et af VideometerLabs helt store fordele er at systemet er i stand til at tage total reproducerbare billeder ved at have en uniform metode til kalibrering og lyssætning af systemet. Selvom dataindsamling og kamerakalibrering er en vigtig del af ethvert billedbehandlingsprojekt der behandler rigtige<sup>11</sup> billeder, vil vi ikke beskæftige os med denne del af dataindsamlingen. Dog vil der være en kort introduktion af VideometerLab systemet og en beskrivelse af de nævnte fordele.

Hovedvægten af opgaven er lagt på funktionaliteten af den udarbejdede software, da der eksisterer et akut behov hos Danisco for at få metoder til de ovenfor beskrevne analyser. Da det endelige system ikke skal eksistere i et *live*<sup>12</sup> miljø, og kun er tænkt til forskning og analyse, er der ikke nogen hårde krav om systemets ydeevne. Det er derfor ikke vigtigt for projektet at sørge for at hver enkelt metode er optimeret mest muligt. Selvfølgelig skal det alligevel tilstræbes

---

<sup>11</sup>I modsætning til syntetisk fremstillede computergenererede billeder

<sup>12</sup>Med live menes et produktionsmiljø hvor systemet f.eks. skulle være en del af et kvalitetskontrols system med 100% dækning.

at gøre systemet så smidigt som muligt sådan at en enkelt analyse ikke tager flere timer, men måske kun få minutter eller sekunder.

Da de personer der kommer til at arbejde med systemet ikke forventes at være eksperter i hverken programmering eller generel softwareudvikling, skal det tilstræbes at gøre systemet så nemt at betjene og så automatiseret som muligt.

En acceptabel hastighed er bestemt af hvor længe brugeren vil vente på resultaterne. Er man sikker på at resultatet vil være tilfredsstillende er tiden af mindre betydning i forhold til hvis det er nødvendigt at justere på forskellige parametre inden det ønskede resultat nås.

Det er heller ikke projektets mål at udarbejde software der optimerer brugen af PC'ens hukommelse, da dette generelt ikke er noget problem med moderne maskiner. Dog er der visse begrænsninger, se afsnit 5.2.

Der er ikke blevet stillet konkrete krav til hvor gode resultater der skulle opnås indenfor de forskellige analyser, og det må derfor være op til Danisco A/S at vurdere om den udarbejdede funktionalitet opfylder deres interne kvalitetskrav.

**Del III**

**Teori**



I denne del vil de metoder der benyttes til den efterfølgende dataanalyse blive beskrevet.

De fleste metoder vil blive beskrevet med eksempler af en af de tre fødevarer: kiks med barrierelag, brød eller spegepølser. Det er så vidt muligt holdt konsekvent hvilken en fødevarer og hvilken sample der er benyttet igennem eksemplerne for de enkelte metoder, med mindre andet er fundet nødvendigt. For alle metoder er det forsøgt at beskrive deres muligheder såvel som begrænsninger.

Nedenfor vil flere eksplorative analysemetoder blive beskrevet. Disse udmærker sig ved at hjælpe til med at få datasættet ind under neglene, men er ikke allesammen direkte anvendelige i den efterfølgende dataanalyse, og vil derfor kun blive beskrevet i dette kapitel. Det drejer sig om metoderne *falsk farve sammensætning*, *densitetsbilleder*, *score billeder* og *plots* og *residual billeder*.



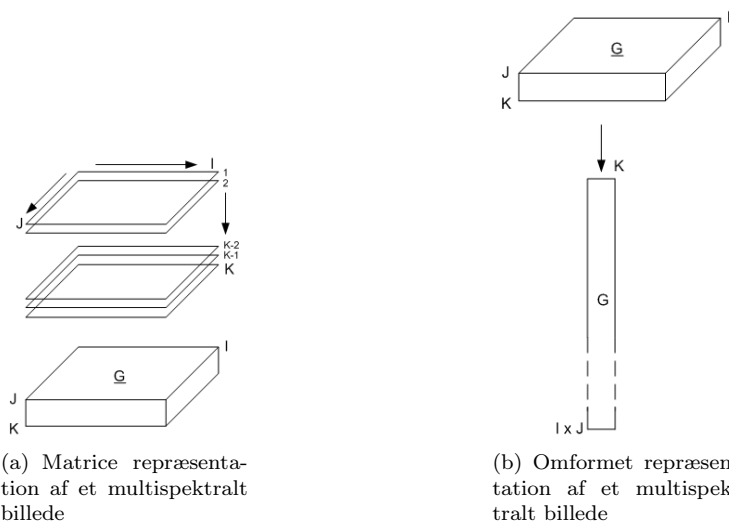
# Multispektral billedteori

---

## 12.1 Multispektrale billeder

Et multispektralt billede kan betragtes som et array af data i 3 dimensioner, hvor de to første beskriver billedets geometriske koordinater. Hvert ”lag” i den tredje dimension beskriver så et enkelt billede for en given variabel. I vores tilfælde er denne variabel en bestemt bølgelængde af lys. Alle værdier fra de forskellige bølgelængder for en given pixel kan samlet ses som en ”signatur”, der udtrykker kemierne i den pågældende pixel.

$\underline{G}$  repræsenterer hele matricen,  $I$  og  $J$  er billed-akserne og  $K$  definerer hvilket lag, der er tale om.  $g_{ijk}$  er et element i  $\underline{G}$ . I mange tilfælde er det lettere at behandle billedet matematisk, ved at omforme det 3-dimensionelle til et 2-dimensionelt billede med dimensionerne  $I \times J$  og  $K$ . Derved mistes umiddelbart den spatielle information, men ofte er denne ikke vigtig, kun den enkelte pixels tilhørsforhold til de øvrige bånd er vigtig. Disse to repræsentationer er illustreret på figur 12.1. Det normale 3D billede vil blive angivet med en understregning af det bogstav der repræsenterer det. Det oprindelige billede kan genskabes ved at lave den modsatte transformation. Derved mistes den spatielle information kun midlertidigt.



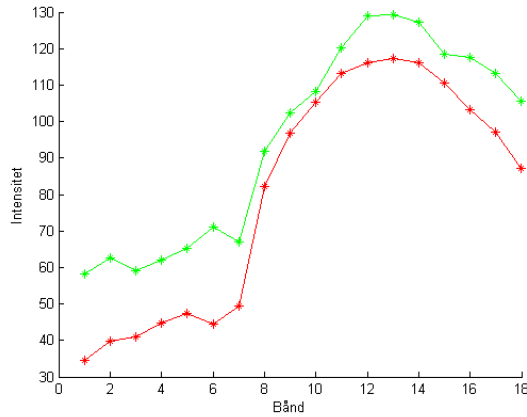
**Figur 12.1:** Figuren viser den normale 3-dimensionelle matrix-repræsentation ( $\underline{G}$ ), og den omformede 2-dimensionelle repræsentation ( $G$ ) af et multispektralt billede [Geladi og Grahn, 1996].

I multispektral billedanalyse af fødevarer udnytter vi at fødevarerne i forskellige punkter, eller pixels, indeholder informationer om hvordan kemien i de enkelte pixels absorberer lys forskelligt ved forskellige bølgelængder. På den måde adskiller multispektral sig fra de 2-dimensionale billeder, da vi ofte ikke kigger på forholdet mellem tætliggende pixels, dvs. i 2D akserne X og Y, men i den ekstra Z akse. På figur 12.2 ses intensiteterne over de 18 bånd for to udvalgte pixels i en pølse. De to pixels er udvalgt ved at se hvad der helt sikkert er fedt og hvad der helt sikkert er kød. Den grønne farve er signaturen for fedt, og den røde er signaturen for kød. Det kan ses hvordan de to grafers form følges pænt ad over de 18 bånd, men det kan også ses at der i nogen bånd er større forskel i intensiteterne end ved andre bånd.

## 12.2 Eigenverdier og -vektorer og Principal Komponent Analyse

Et af de umiddelbare problemer med multispektrale billeder er at data er meget korrelerede, sådan at det kan være svært at se signaturen for de egenskaber man er interesseret i. Derudover er datamængderne relativt store, hvilket gør det sværere at se det essentielle og det interessante i billederne. En af metoder-





**Figur 12.2:** Kød- og fedtspektre for to udvalgte pixels. Den grønne linie viser intensiteter for en fedt pixel, og den røde for en kød pixel.

ne til at skære datamængden ned, og samtidig maksimere variansen i billedet er Principal Komponent Analyse (PCA)<sup>13</sup>. Data for et multispektralt billede med  $N$  lag kan repræsenteres i et  $N$ -dimensionelt rum som en sky af punkter beskrevet af en ellipsoide (en  $N$ -dimensionel ellipse). I PCA benyttes Eigenvalue Decomposition (EVD - Eigenvalue Decomposition) eller Singular Value Decomposition (SVD) til at bestemme eigenverdier og eigenvektorer for de  $N$  dimensioner [Wikipedia, 2006b]. På denne måde skabes  $N$  nye lag, der hver beskriver én feature i billedet. I implementeringen af PCA funktionen er EVD benyttet til at finde eigen-vektorerne.

### 12.2.1 Eigenverdier og -vektorer

I det følgende bliver eigenverdier og -vektorer beskrevet, og det vil blive forklaret hvordan disse kan bruges i forbindelse med emnet multispektral billedanalyse. Det vil ikke blive beskrevet hvordan man beregner eigenvektorerne og -værdierne, da det er udenfor projektets rammer.

Eigen-vektor/-værdi ligningen er som beskrevet i [Geladi og Grahn, 1996, 81]:

$$(12.1) \quad Zp = \lambda p$$

<sup>13</sup>Principal komponent analyse er den danske oversættelse af principal component analysis (PCA).

Dette betyder at der eksisterer en vektor  $p$  med længden  $K$  og en konstant  $\lambda$  for den kvadratiske matrice  $Z$  med størrelsen  $K \times K$ <sup>14</sup> af reelle tal, sådan at  $p$  gange  $Z$  bliver til  $p$  skaleret op eller ned med en faktor  $\lambda$ . Man siger at  $p$  er en egenvektor for  $Z$  med egen-værdien  $\lambda$ . For kvadratiske matricer af størrelsen  $K$  vil det maksimale antal egen-vektorer være  $K$ . I modsætning til egen-værdier er egen-vektorer ufølsomme overfor skalering. Dvs. at

$$(12.2) \quad (\mathbf{k}Z)p = (\mathbf{k}\lambda)p$$

### 12.2.2 Principal Komponent Analyse

Principal Komponent Analyse (PCA) kan benyttes i mange og forskellige sammenhænge. Man kan dels se PCA som ren matematik eller statistik, men man kan også se det som en geometrisk beskrivende funktion. I dette afsnit ser vi på den rent matematiske form. I det efterfølgende afsnit ser vi på hvordan PCA kan benyttes sammen med multispektrale billeder til at finde geometriske strukturer, og senere vil PCA blive benyttet i en modificeret udgave til at frembringe *Minimum Autocorrelation Factors* (MAF).

PCA er dekomponeringen af matricen  $X$  til summen af en række simple matricer  $M_a$  (med samme størrelse som  $X$ ).

$$(12.3) \quad X = \sum_{a=1}^A M_a$$

Hvor  $A$  er den matematiske rang af  $X$ . For dekomponeringen ved PCA har matricerne  $M_a$  den specielle egenskab at deres rang er 1.

En matrices rang er et udtryk for det mindste antal ydre produkter, der skal til for at repræsentere matricen  $X$  eksakt. I forbindelse med egen-værdier og egenvektorer er rang det antal egen-værdier af  $X'X$  som er forskellig fra 0.

Hver  $M_a$  kan altså udtrykkes som det ydre produkt mellem to vektorer,  $t$  og  $p$ .

$$(12.4) \quad M_a = t_a p_a'$$

---

<sup>14</sup>I det følgende vil det være underforstået at  $Z$  matricen vil være kvadratisk, da vi primært vil benytte krydsproduktmatricen (evt. covariance- eller correlationsmatricen) i forbindelse med egen-vektor bestemmelsen.

Den samlede ligning kommer derfor til at se sådan ud:

$$(12.5) \quad X = \sum_{a=1}^A t_a p_a'$$

$t_a$  vektorerne kaldes *score-vektorer* og  $p_a$  vektorerne kaldes *loading-vektorer*, og loading-vektorerne er netop egenvektorerne til  $X'X$ .

*Eigenværdier*, *scores* og *loadings* kan udregnes ved en Eigenvalue Decomposition (EVD). EVD vil ikke blive beskrevet yderligere her, da det er uden for rapportens fokusområde. I Matlab implementeringen benyttes den indbyggede EVD-funktion (*eig*) til formålet.

## 12.3 Principal Komponent Analyse på multispektrale billeder

Som beskrevet i afsnit 12.1 repræsenteres multispektrale billeder ofte i et 3-dimensionelt array, hvor to af dimensionerne er selve billedkoordinaten, og den sidste dimension er variabelen for det multispektrale billede, i vores tilfælde bølgelængden. Ved PCA kan det være en fordel at reorganisere dette array til 2 dimensioner som vist på figur 12.1, for dermed at simplificere udregningerne. Strukturen i billedet går i første omgang tabt, men dette har ingen betydning for den beregningsmæssige del af analysen, og tabet er kun midlertidigt, da det oprindelige billede kan genskabes efterfølgende.

Følgende punkter beskriver fremgangsmåden for udførsel af PCA på et multispektralt billede:

1. Kovarians-matricen bestemmes for det multispektrale billede reorganiseret til 2D.
2. Eigenvektorer og eigenværdier bestemmes ud fra kovariansmatricen.
3. PCA-billedet udregnes ved at multiplicere det oprindelige billede med egenvektor matricen. Lagene ordnes efter betydning, så i praksis vil de første lag typisk være reel information og de sidste være mere eller mindre ren støj, såfremt der er større varians i signal end i støj, hvilket ikke altid er tilfældet for sekundære egenskaber.
4. PCA-billedet reorganiseres tilbage igen fra 2D repræsentationen til den oprindelige 3D repræsentation sådan at den spatielle information reetableres.

### 12.3.1 Loadings

Loadings er som nævnt et element i PCA analysen som en del af resultatet fra Singular Value Decomposition (SVD).

$$(12.6) \quad Z = PDP'$$

$Z$  er krydsprodukt matricen,  $P$  er loading matricen og  $D$  er en diagonal matrice med eigenverdier. Loadings i sig selv kan vurderes i form af f.eks. bar plots til at give et indtryk af hvilke bånd der udtrykker nogle af de samme egenskaber og hvilke der skiller sig ud. To eller eventuelt tre loading-vektorer kan også sammenlignes vha. scatter plots (i dette tilfælde ofte kaldes loading plots). Hver loading-vektor beskriver korrelationen med billedets bånd i den pågældende PCA-komponent. Hovedsageligt kan loading plots bruges til at få tal på egenskaber og sammenhænge der allerede er mere eller mindre tydelige i det oprindelige billede.

### 12.3.2 Score-billeder

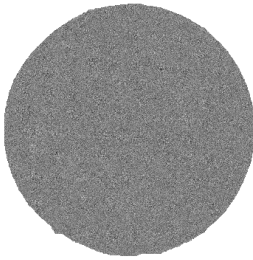
Score-billeder kan beregnes ud fra det omformede originalbillede<sup>15</sup>  $G$  og de fundne loading-vektorer  $p_a$ :

$$(12.7) \quad t_a = Gp_a \quad a = 1, \dots, A$$

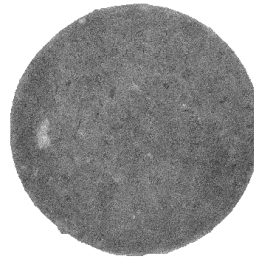
Resultatet fra ligning (12.7) bliver en score-vektor for hver loading-vektor, der kan reorganiseres til et billede med samme dimensioner som originalbilledet. Hver komponent i det samlede score-billede udtrykker en variation i det oprindelige billede, uafhængigt af alle andre komponenter. Den største variation er altid udtrykt i første komponent og med faldende variation derefter. Med mange bånd i originalbilledet vil de sidste bånd af score-billedet ofte kun indeholde støj, støj der dermed er adskilt fra de interessante data. Principalkomponenterne er orthogonale, hvilket i score-billederne er vist som en optimal visuel kontrast mellem hver komponent. I det følgende vil score-billeder også blive betegnet som PCA-billeder.

En svaghed ved PCA er at metoden ikke tager højde for den spatielle information i billedet. Selvom PCA sorterer komponenterne sådan at komponenter med højest varians kommer først og komponenten med lavest varians kommer sidst, kan komponenter med lav varians sagtens afspejle strukturer i billedet. Dette er et problem hvis det ønskes at mindske datamængden til kun at omfatte det

<sup>15</sup>Se afsnit 12.1



(a) Komponent 7



(b) Komponent 8

**Figur 12.3:** PCA ordner ikke nødvendigvis komponenterne så signal kommer før støj. Figuren viser PCA komponent 7 og 8 for et spegepølsebillede. I komponent 8 optræder tydelige strukturer i billedet, som ikke kan ses for komponent 7, der kun ser ud til at indeholde støj.

Komponent	Varians
7	2.2102
8	2.0812

**Tabel 12.1:** Varians for PCA komponent 7 og 8.

mest relevante. På figur 12.3 er vist et eksempel hvor den lavere komponent 8 tydeligt indeholder strukturer, hvor den højere komponent 7 kun indeholder støj. Variansen for komponent 7 er naturligvis højere end den for komponent 8 som det ses i tabel 12.1, og det er derfor den er rangeret højere. Strukturene fundet på figur 12.3(b) kunne iøvrigt være f.eks. et stykke brusk, eller noget andet der skiller sig ud fra de øvrige fedtstrukturer.

## 12.4 Minimum autokorrelations faktorer

Minimum autokorrelations<sup>16</sup> faktor (MAF) analyse er en generel metode fra multivariat statistik, der også kan benyttes til at minimere autokorrelationen i multispektrale billeder<sup>17</sup> [Nielsen, 1999], [Larsen, 2002]. I modsætning til PCA tager MAF hensyn til den spatielle natur i billeder, og er derfor på papiret mere velegnet til komponent analyse i multispektrale billeder end PCA.

<sup>16</sup>Autokorrelation er kryds-korrelationen af et signal med sig selv. Det er særligt nyttigt til at finde mønstre i et signal der er meget begravet i støj.

<sup>17</sup>Funktionen kan også bruges til at maximere autokorrelationen, hvorved rækkefølgen af komponenterne bare bliver vendt om. Denne metode kaldes simpelt for Maximum autokorrelations faktorer

Det multispektrale billede  $\underline{G}$  omformes til 2-d repræsentationen  $G_k$ , hvor  $k$  angiver den spatielle position (pixel) og  $\Delta$  angiver et spatielt skift. Den spatielle kovarians-funktion defineres som  $\Gamma(\Delta) = Cov(G_k, G_{k+\Delta})$ , hvor  $\Gamma^T(\Delta) = \Gamma(-\Delta)$ . Kovarians-matricen af  $G_k$  (signal-matricen) kaldes  $\Sigma$  ( $\Sigma = Cov(G)$ ) og kovarians-matricen for kovarians-matricen af  $G_k - G_{k+\Delta}$  kaldes  $\Sigma_\Delta$ , så

$$(12.8) \quad \Sigma_\Delta = 2\Sigma - \Gamma(\Delta) - \Gamma(-\Delta)$$

Kovariansen mellem en lineær kombination af de originale variable og de skiftede kan så bestemmes med

$$(12.9) \quad \begin{aligned} Cov\{\omega_i^T G_k, \omega_i^T G_{k+\Delta}\} &= \omega_i^T \Gamma(\Delta) \omega_i = \omega_i^T \Gamma^T(\Delta) \omega_i \\ &= \frac{1}{2} \omega_i^T (\Gamma(\Delta) + \Gamma(-\Delta)) \omega_i \\ &= \omega_i^T \left( \Sigma - \frac{1}{2} \Sigma_\Delta \right) \omega_i \end{aligned}$$

hvor  $\omega_i$  er egenvektoren til det enkelte bånd. Autokorrelationen i skiftet  $\Delta$  af en lineær kombination af de centrerede original-variable  $G_k$  er så

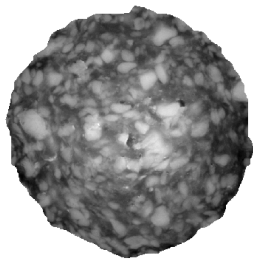
$$(12.10) \quad Corr\{\omega_i^T G_k, \omega_i^T G_{k+\Delta}\} = 1 - \frac{1}{2} \frac{\omega_i^T \Sigma_\Delta \omega_i}{\omega_i^T \Sigma \omega_i}$$

For at minimere korrelationen skal Rayleigh-koefficienten maksimeres

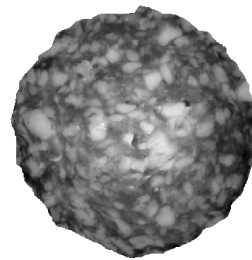
$$(12.11) \quad R(\omega) = \frac{\omega^T \Sigma_\Delta \omega}{\omega^T \Sigma \omega}$$

MAF-transformationen findes ved at gange egenvektorerne af  $\Sigma_\Delta$  i forhold til  $\Sigma$  med det oprindelige billede ( $eig(\Sigma, \Sigma_\Delta)$ )

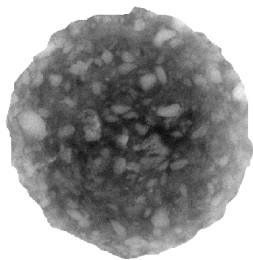
Et problem med MAF er at uønskede strukturer i billedet bliver fundet som havende større betydning end de egenskaber man ønsker at undersøge. Det ses tydeligt på figur 12.4(c) og 12.4(d), at fedtkanten har "forstyrret" MAF analysen



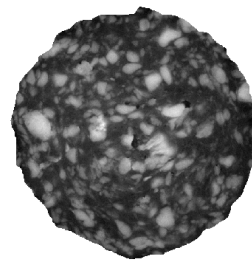
(a) Med fedtkant - PCA



(b) Uden fedtkant - PCA



(c) Med fedtkant - MAF



(d) Uden fedtkant - MAF

**Figur 12.4:** Figuren viser komponent 1 for *sample 2*, med og uden fedtkant ved en PCA og MAF analyse. Analyse er udført på et billede med og uden fedtkant, og derefter er fedtkanten blevet fjernet på billedet med, for at kunne sammenligne visuelt.

sådan at billedet på figur 12.4(c) udtrykker forskellen mellem fedt og kød dårligt. For PCA analysen er der ikke nogen nævneværdig forskel på om fedtkanten er med, eller ej. Denne konklusion understøttes også af variationskoefficienterne ( $cv = \frac{\sigma}{\mu}$ ), som et udtryk for kontrasten, i tabel 12.2. Hvor værdierne for PCA er meget ens, er der en stor forskel på MAF værdierne.

Da vi er mest interesseret i fedt/kød opdelingen og ikke i fedtkanten, skal denne fjernes for at sørge for at fedt/kød opdelingen bliver fundet ved den første komponent. Som det ses på figur 12.4(d) er der langt større kontrast mellem fedt og kød på dette billede, i forhold til figur 12.4(c) hvor fedtkanten ikke var fjernet før MAF analysen.

	Med fedtkant	Uden fedtkant
PCA	0.0958	0.0944
MAF	-0.1421	0.8869

**Tabel 12.2:** Variationskoefficienter (CV) for spegepølse *sample 2*, med og uden fedtkant.

## 12.5 Falsk farvesammensætning

Med denne metode kan man sammensætte bånd fra f.eks. et PCA-komponent billede til et RGB billede sådan at hver komponent er udtrykt med hver sin farve. Metoden er illustreret på figur 12.5(a) hvor tre komponenter er kombineret til at danne ét billede. På figuren ses tre billeder der hver bidrager til det samlede billede med spatielle strukturer. Nogle steder overlapper de forskellige strukturer, hvilket bliver udtrykt ved at intensiteten af de enkelte overlappende pixels i output billedet stiger i intensitet. De steder hvor der ikke umiddelbart er dominerende strukturer i de andre billeder træder strukturen fra den enkelte komponent i karakter og udtrykker sig med sin tildelte farve.

På figur 12.5(b) er der vist et eksempel på hvordan 3 komponenter for et *PCA*-komponent billede af en spegepølse, er kombineret i et enkelt billede og lader farveintensiteten udtrykke hvilke komponenter der har de højeste værdier i de enkelte pixels, og dermed udtrykker en specifik egenskab.

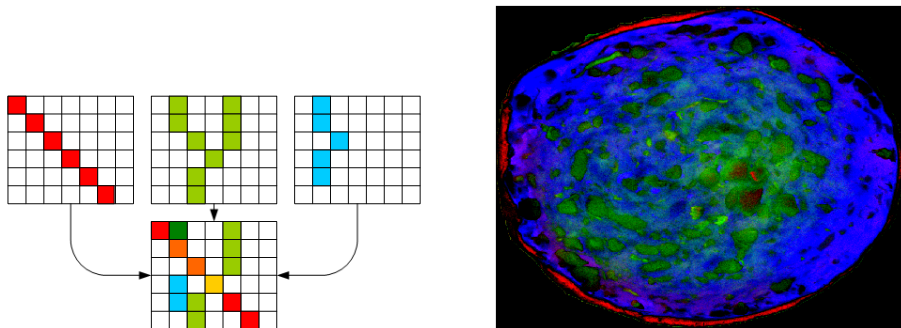
Når man benytter komponenterne i f.eks. et *PCA* billede, skal man være opmærksom på at der er stor forskel på værdierne ved de enkelte komponenter. Dvs. at værdierne for komponent 1 typisk er meget højere end værdierne for bare komponent 2. Derved vil komponent 1 stort set altid være den altdominerende faktor i billedet, og det er kun en meget lille andel de resterende komponenter vil bidrage med. Eventuelt kan de enkelte komponenter skaleres ens for at undgå at én komponent bliver for dominerende.

## 12.6 Densitets billeder, score billeder og plots

Score-billeder og plots kan bruges til at danne sig et visuelt overblik over variabelrummet.

Densitet billeder og score plots kan i 2 eller 3 dimensioner vise hvordan farveværdier i forskellige bånd er fordelt i forhold til hinanden. Et densitets billede eller score plot genereres som illustreret på figur 12.6(a). To eller tre billeder (alt efter om man ønsker et 2 eller 3 dimensionelt billede) løbes igennem pixelvis





(a) Tre komponenter af f.eks. et *PCA*-billede tildeles hver sin RGB farve.

(b) Her er falsk farvesammensætningsmetoden anvendt på et *PCA*-billede hvor rød er blevet tildelt til komponent 4, grøn er tildelt til komponent 3 og blå er tildelt til komponent 2.

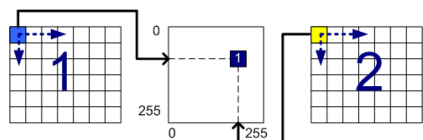
**Figur 12.5:** Falsk farvesammensætningsbilleder. På figur (a) vises hvordan tre forskellige komponenter kan sættes sammen til at udtrykke sig med visuelt adskillelige farver. På figur (b) er vist et praktisk eksempel hvor man kan se de forskellige komponenter af det samme *PCA*-billede udtrykke sig i det samme billede vha. forskellige farver.

og værdierne i hver pixel fra billede 1 plottes som X-værdi og billede 2's værdi i samme pixel bliver så Y-værdien. Har man tre billeder til et densitetsbillede eller score-plot vil pixel-værdierne i billede 3 repræsentere Z-værdien.

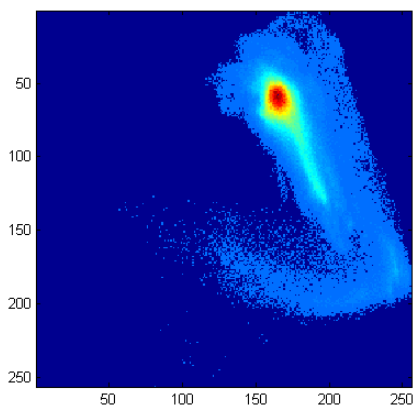
Et score-plot er et binært billede og derfor sættes pixel-værdierne i output billedet til 1 hvor værdierne fra de to input billeder mødes. I et densitets billede lægges der 1 til den eksisterende værdi i mødepunktet så man på den måde får et billede der indeholder større værdier der hvor der er stort sammenfald mellem de to input billeder. Derfor vil der typisk opstå "skyer" med svage farver hvor der optræder strukturer i den ene komponent, men ikke i den anden. Der hvor der er stort sammenfald vil der typisk fremkomme høje værdier og dermed intense farver.

På figur 12.6(b) ses et eksempel på hvordan et densitets billede kan se ud. Da de skyer eller "haler" der optræder i billedet tit er det mest interessante, kan det være en fordel at inddele billedet i flere regioner og undersøge hver region for sig. På den måde vil strukturer der eventuelt er gemt i skyerne træde tydeligere frem og på den måde kunne undergå yderligere undersøgelser.

Ved at vælge en specifik region ud af scoreplottet kan man generere en ROI der kun indeholder den feature man er interesseret i at undersøge.



(a) Metode til at fremstille et densitets eller score-plot billede



(b) Densitets billede af et PCA-komponent billede af en spejlepølse ved komponent 1 og 2. X-aksen viser komponent 1 og y-aksen viser komponent 2

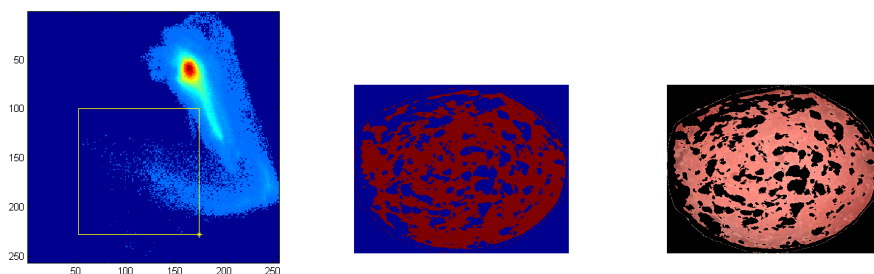
**Figur 12.6:** De to billeder illustrerer henholdsvis selve metoden til at lave et densitets eller score-plot billede og et eksempel på hvordan et sådant kan se ud.

På figur 12.7 er vist hvordan en region er valgt ud på det genererede densitets billede for komponent 1 og 2 i et PCA-billede, og en ROI er genereret ud fra de valgte grænser for komponent 1, dvs. på x-aksen. Ved på samme måde at generere en ROI for komponent 2, dvs. y-aksen og kombinere de to ROI opnås effekten med rammen som vist på figur 12.7(a). Denne ROI kan derefter overføres til det oprindelige billede, sådan at man får masket lige præcis den feature ud man er interesseret i at kigge på.

Da dette er en eksplorativ analyse af datasættet, hvor man manuelt skal finde frem til grænserne, vil det i mange tilfælde resultere i at det der bliver masket ud, ikke er en feature man er interesseret i. På figur 12.8 ses en udvalgt score region fra et PCA billedes komponent 1 og 2. Det kan ses ud fra billedet at den region der er valgt ud i begge komponenter er fedtet, men det kan også ses at der stadig er valgt en del kød ud i billedet, som kun kan fjernes ved manuelt at vælge en anden region ud.

## 12.7 Residual billeder

Rigtige billeder, i modsætning til kunstigt fremstillede, indeholder naturlig støj. Denne støj er som regel ikke interessant i forbindelse med en given analyse, men

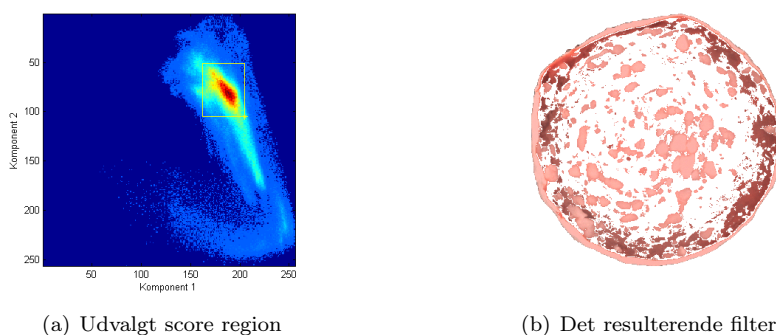


(a) Regions udvælgelse. Den gule ramme viser det område der er valgt ud. X-aksen viser komponent 1 og y-aksen viser komponent 2

(b) Genereret ROI ud fra de valgte grænser på figur (a) for komponent 1 og 2. Værdierne 52-175 for komponent 1 og 100-228 for komponent 2 er valgt.

(c) ROI'en overført til det oprindelige billede og vist med RGB farver.

**Figur 12.7:** Figuren viser hvordan en ROI kan genereres ud fra et densitetsplot. Først vælges regionen der repræsenterer den feature man er interesseret i, og dernæst genereres en ROI for den valgte region.



(a) Udvalgt score region

(b) Det resulterende filter

**Figur 12.8:** Et eksempel på hvordan en eksplorativ scoreplotanalyse af data kan resultere i uønskede resultater som man manuelt skal udbedre.

man *kan* komme ud for at støjen indeholder interessante data. Residualbilleder viser afstanden mellem et PCA-billede og billedets naturlige støj. Disse billeder viser afstanden til PCA modellen som intensiteter, sådan at lyse pixels er langt fra den givne model, og mørke pixels er tæt på.

Som beskrevet i afsnit 12.2 kan en PCA transformation beskrives ved ligningen (12.5). Da PCA analyse i sig selv er en matematisk transformation, tages der ikke højde for den støj der utvivlsomt vil optræde i rigtige billeder. Der vil derfor meget ofte være PCA-komponenter der ikke er brugbare. En mere ideel løsning ville være hvis man kunne sortere al støjen fra, sådan at ligning 12.2 kunne omskrives til

$$(12.12) \quad X = \sum_{a=1}^A t_a p'_a + E$$

hvor  $E$  er residualerne i  $I \times J \times K$  dimensioner, svarende til dimensionerne for det originale billede. I teorien indeholder  $E$  udelukkende tilfældig støj, og er derfor ikke interessant at kigge på som individuelle billeder, men beregner man i stedet for distancen som kvadratsummen til modellen i hver enkelt pixel, får man et  $I \times J \times K$  dimmensionalt billede der viser de beregnede afstande som strukturer i model-afstands billedet. Ligning (12.12) omskrives til

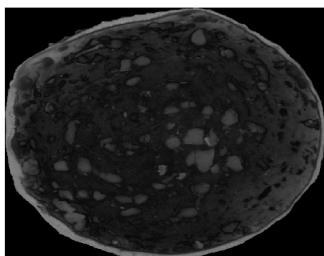
$$(12.13) \quad E = X - \sum_{a=1}^A t_a p'_a$$

for at finde afstanden til modellen benyttes denne ligning, hvor  $e_{ijkA}$  er et element i  $E$ .

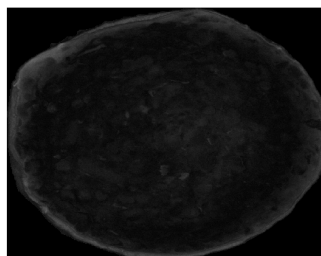
$$(12.14) \quad d_{ijA} = [e'_{ijkA} e_{ijkA}]^{0.5} = \left( \sum_{k=1}^K e_{ijkA} e_{ijkA} \right)^{0.5}$$

hvor  $d_{ijA}$  er afstands vektoren til modellen i punktet med koordinatet  $(i, j)$  med størrelsen  $(K \times 1)$ . Alle  $d_{ijA}$  pixelværdierne former tilsammen intensitets billedet  $D_A$ . På figur 12.9 ses model-afstands billederne af en PCA-transformeret spegepølse efter 1-4 PCA-komponenter. På billedet illustrerer de pixels med de lyseste farver (højeste intensiteter) den højeste korrelation til PCA modellen, og de mørke (med lavest intensitet) viser den største afstand til PCA modellen.

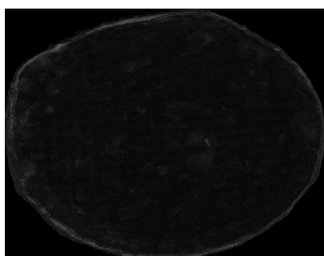
Det menneskelige øje har svært ved at skelne mellem gråtonerne i billederne på figur 12.9, og det kan derfor være en fordel at farvekode billedet, sådan at kontrasten mellem de lave værdier træder tydeligere frem. Farvekoden der er benyttet for billederne på figur 12.10 kan ses i tabel 12.3. Det ses tydeligt at den strukturelle information i billedet gradvist forsvinder fra PCA-modellen



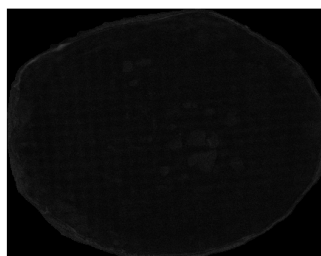
(a) Efter 1 PCA-komponenter



(b) Efter 2 PCA-komponenter



(c) Efter 3 PCA-komponenter

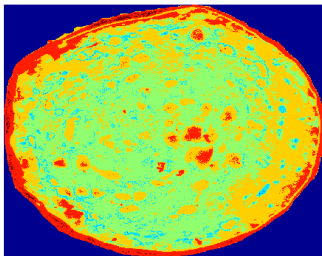


(d) Efter 4 PCA-komponenter

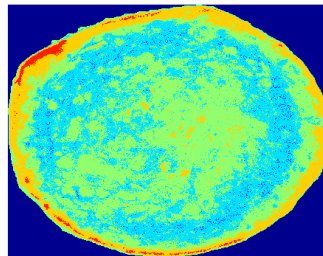
**Figur 12.9:** Billederne viser model-afstands billederne efter 1-4 PCA-komponenter af en transformeret spegepølse.

Intensitet	Farve
0-2	Mørkeblå
2-4	Blå
4-8	Lyseblå
8-16	Grøn
16-32	Orange
32-64	Rød
64-128	Mørkerød

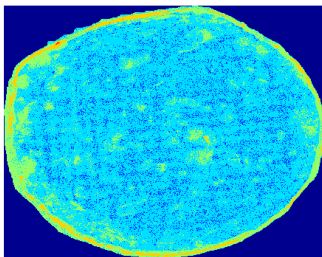
**Tabel 12.3:** Farvekodningsskema for residualbilledet på figur 12.10.



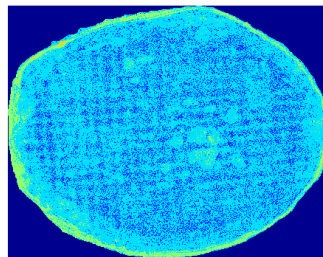
(a) Efter 1 PCA-komponenter



(b) Efter 2 PCA-komponenter



(c) Efter 3 PCA-komponenter



(d) Efter 4 PCA-komponenter

**Figur 12.10:** Billederne viser model-afstands billederne efter 1-4 PCA-komponenter af en transformeret spegepølse. Her er benyttet farvekodningsskemaet fra tabel 12.3.

efterhånden som man bevæger sig op igennem komponenterne, og at nogle regioner/egenskaber i billedet er godt definerede i de første par komponenter hvor andre er bedre repræsenteret i de senere.

Det kan også ses på billederne 12.10(c) og 12.10(d) at der fremkommer en gitter lignende struktur. Det er ikke helt til at sige hvad denne skulle komme af. Gitteret ligger tydeligvis ikke parrallelet med de to pixel akser, og der er ikke noget i algoritmen der tyder på at det skulle være en matematisk effekt. Derfor kunne det tyde på at det er noget fysisk støj vi ser, f.eks. fra den måde pølser er blevet skåret på, eller hvis den har ligget på et gitterformet underlag på et tidspunkt undervejs i processen.





# Regressionsmetoder

---

I dette kapitel vil forskellige metoder til regressionsanalyse blive beskrevet.

For at kunne opdele et billede i forskellige klasser som f.eks. fedtet og kødet i en spegepølse, estimere barrierelagstykkelsen for en kiks eller estimere vandindholdet i brød, benyttes regressionsanalyse. Regressionen opstiller en model ved at returnere en række parametre der giver en sammenhæng mellem et givent billede og den kendte klasse, kaldet responset.

Der findes potentielt uendeligt mange modeller lavet med regression og for at vælge den bedste model ud af mange, benyttes krydsvalidering. Ved krydsvalidering opstilles  $K$  forskellige modeller ud fra det samme datasæt, og den bedste kan så efterfølgende vælges ud.

## 13.1 Krydsvalidering

Krydsvalidering er en metode man benytter for at gøre en model mere generel og undgå overtilpasning (overfitting). Ideen er at udelukke en del af data (test-data), og bygge en model på resten (træningsdata). Når modellen er bygget, anvendes den på de resterende testdata.

K:	1	2	3	4	5
Data:	Træ- ning	Træ- ning	Træ- ning	Test	Træ- ning

**Figur 13.1:** K-fold krydsvalidering. Data er delt op i  $K=5$  nogenlunde lige store stykker.

I K-fold krydsvalidering deles data op i  $K$  nogenlunde lige store dele sådan at hver del indeholder ca.  $\frac{K}{N}$  observationer. Opdelingen er illustreret på figur 13.1.

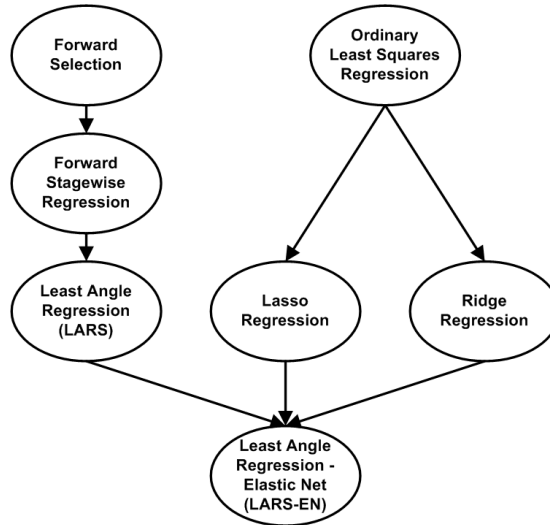
Vi fitter en model for den  $k$ 'te del, tester på de resterende  $K - 1$  data, og udregner fejlen for det beregnede. Dette gøres for  $K = 1, 2, \dots, N$  og den bedste model vælges ud fra den mindste fejl. Tilfældet hvor  $K = N$  kendes som *Leave One Out* (LOO) krydsvalidering. Ifølge [Hastie et al., 2001][Clemmensen, 2006] er krydsvalidering ved  $K = N$  unbiased<sup>18</sup>, tilgængæld kan variansen være høj hvis de  $N$  træningssæt er meget ens, og omvendt ved lave værdier af  $K$ . Ved små datasæt som vores hvor vi har 5 til 6 billeder af hver serie kiks eller spegepølse er LOO krydsvalidering at foretrække [Clemmensen, 2006].

Den helt store fordel ved at anvende krydsvalidering er som nævnt at den resulterende model ikke bliver over tilpasset. Tilgængæld kan det modsatte problem også opstå. Hvis data er lidt for forskellige, kan modellen pludselig blive for almen og er måske god generelt set, men er ikke længere god i det enkelte tilfælde.

## 13.2 Least Angle Regression - Elastic Net

I dette afsnit beskrives kort de regressionsalgoritmer, der ligger til grund for *Least Angle Regression - Elastic Net* (LARS-EN), og LARS-EN algoritmen beskrives. Disse regressionsmetoder inkluderer *Ordinary Least Squares*, *Ridge*, *Lasso*, *Forward Selection*, *Forward Stagewise* og *Least Angle regression*. LARS-EN er blevet benyttet med succes til at estimere vandindhold i sandprøver [Clemmensen, 2006] baseret på multispektrale billeder og sammenlignet med en række andre metoder (OLS, Forward Selection, Ridge, Lasso m.fl.). Konklusionen var at LARS-EN sammen med Ridge og Lasso gav de laveste standardafvigelser og at LARS-EN både var markant hurtigere og mere fleksibel i forhold til de andre to. I forbindelse med estimering af vandindhold i brød på baggrund af multispektrale billeder benyttes derfor LARS-EN algoritmen.

<sup>18</sup>Der findes ikke nogen fyldestgørende oversættelse af dette ord, men det dækker over at de resulterende modeller ikke har indbygget for mange forudsætninger i forhold til de givne data.



**Figur 13.2:** Sammenhæng mellem regressionsmetoder.

Som det beskrives nærmere i de enkelte afsnit udvider flere af metoderne hinanden i forsøget på at give en bedre og ”smartere” løsning. Denne sammenhæng er illustreret på figur 13.2.

### 13.2.1 Anvendt notation

- $X$  er det samlede datasæt af variable/prediktorer.
- $x_j$  er en given variabel/feature.
- $n$  er antal samples og  $p$  er antal variable.
- $y$  betegner responset - den afhængige variabel.
- $\beta$  er parameter-vektoren for alle variable.

### 13.2.2 Ordinary Least Squares regression

Den generelle model for multipel lineær regression er

$$(13.1) \quad y = X\beta + \epsilon, \text{ hvor } \epsilon \in N(0, \sigma^2)$$

*Ordinary Least Squares* (OLS) finder estimater for  $\beta$  ved at minimere residualernes *Sum of Squares* (RSS).

$$(13.2) \quad \text{Minimér } \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2$$

OLS-løsningen kan for en X-matrice af fuld rang<sup>19</sup> findes ved

$$(13.3) \quad \beta_{OLS} = (X^T X)^{-1} X^T y$$

OLS er en simpel regressionsmetode, der ofte giver et estimat med stor varians og resultatet kan være svært at tolke for modeller med et højt antal variable, da de alle medtages i OLS-løsningen. Ofte vil det være ønskeligt kun at have de variable med der udtrykker de vigtigste sammenhænge.

### 13.2.3 Ridge regression

Ridge regression er først beskrevet i [Hoerl og Kennard, 1970] - en metode der ligesom OLS minimere RSS (Residual Sum of Squares), dog med et ekstra "straf"-led. Strafleddet skal sørge for at parametrene ikke bliver for store så modellen overfitter til træningsdataene.

Vi har følgende forudsætninger for datasættet

$$\bar{y} = 0, \quad \bar{x}_j = 0, \quad \text{var}(x_j) = 1 \text{ for alle } j$$

Responset er centreret og de uafhængige variable er centreret og standardiseret.

Ridge regression løsningen bestemmes ved

$$(13.4) \quad \text{Minimér } \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

Hvor  $\lambda$  er den valgte "straf"-værdi for Ridge-betingelsen. Størrelsen af  $\lambda$  er yderst vigtig for et godt resultat, idet en for lille værdi betyder at parameterestimatet

<sup>19</sup>Alle variable er lineært uafhængige. Der kan dog stadig være høj korrelation imellem de variable, og det vil ofte være tilfældet med mange variable.

ikke bliver bedre end OLS-løsningen, og en for høj værdi vil straffe parametrene for meget så de bliver tæt på 0.

### 13.2.4 Lasso regression

Least Absolute Shrinkage and Selection Operator (Lasso) [Tibshirani, 1994], er som navnet antyder en regressionsmetode til at mindske og udvælge parameterestimer. Dette opnås ved at tilføje et led til OLS-løsningen der begrænser summen af de absolutte værdier af  $\beta$ . Med den begrænsning betyder det at en del af parametrene vil få værdien 0, og dermed mindskes antallet af beskrivende variable i modellen.

$$\bar{y} = 0, \bar{x}_j = 0, \text{var}(x_j) = 1 \text{ for alle } j$$

Lasso regression løsningen bestemmes ved

$$(13.5) \quad \text{Minimér } \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 \text{ med betingelsen } \sum_{j=1}^p |\beta_j| \leq s$$

Hvor  $s$  er den valgte grænse for Lasso-betingelsen.

### 13.2.5 Forward Selection regression

Forward Selection er en iterativ metode, der søger at løse det problem med OLS at alle variable medtages i modellen. Dette gøres ved at medtage én variabel mere i modellen hver iteration baseret på korrelationen til responset, så de mest signifikante variable tages ind først.

Fremgangsmåden for algoritmen er

1. Find prediktoren  $x_j$  med størst korrelation til den afhængige variabel  $y$
2. Udfør simpel lineær regression af  $y$  på  $x_j$
3. Dette giver en residual-vektor ortogonal på  $x_j$  og de resterende prediktorer projiceres med denne
4. Gentag indtil det ønskede antal parametre er nået eller når  $\text{corr}(y, x_j) = 0$  (eller under en tolerance-værdi)

### 13.2.6 Forward Stagewise regression

*Forward Stagewise* regression er en noget mere "forsigtig" udgave af *Forward Selection*, der med små skridt opbygger de enkelte parameterverdier.

Fremgangsmåden for algoritmen er

1. Start med  $\beta_1, \beta_2, \dots, \beta_p = 0$
2. Find den prediktor  $x_j$  med størst korrelation til  $y$
3. Opdater  $\beta_j = \beta_j + \delta_j$  hvor  $\delta_j = \epsilon \cdot \text{sign}(y \cdot x_j)$  og  $\text{sign}(y \cdot x_j)$  returnerer fortegnet for produktet af  $y$  og  $x_j$ .
4. Sæt  $y = y - \delta_j \cdot x_j$  og gentag punkt 2-4 indtil det ønskede antal parametre er nået

Fastsættelsen af størrelsen for  $\epsilon$  bestemmer skridtstørrelsen for algoritmen.

### 13.2.7 Least Angle Regression

*Least Angle Regression* (LARS) beskrevet i [Efron et al., 2003] er enklere og hurtigere end Lasso og Forward Selection, og kan beskrives som en mere "demokratisk" metode end Forward Selection mht. valg af parameterstørrelser.

Fremgangsmåden for algoritmen er

1. Start med  $\beta_j = 0$  for alle  $j$ .  $x_j$  formodes standardiseret,  $\text{var}(x_j) = 1$
2. Find den prediktor  $x_j$  med størst korrelation til  $y$
3. Forøg  $\beta_j$  i retning af  $\text{sign}(\text{corr}(y, x_j))$  indtil en anden prediktor  $x_k$  har lige så høj korrelation med det nuværende residual som  $x_j$
4. Flyt  $(\beta_j, \beta_k)$  i retning af den fælles *Least Squares*-retning for  $(x_j, x_k)$  indtil en anden prediktor  $x_l$  har lige så høj korrelation med det nuværende residual
5. Gentag indtil alle prediktorer er inkluderet i modellen eller  $\text{corr}(y, x_j) = 0$

LARS minder om Forward Stagewise og Forward Selection algoritmen på den måde at de er iterative, men den er meget hurtigere at udføre end Stagewise og

bedre end Selection. Desuden kan LARS modificeres til også at opfylde Lasso-betingelsen [Efron et al., 2003].

### 13.2.8 Least Angle Regression - Elastic Net

*Least Angle Regression - Elastic Net* (LARS-EN) udvider LARS med Lasso-modifikationen til også at kunne medtage Ridge-betingelsen. Det generelle regressionsproblem angivet i formel 13.1 kan udvides til et ridge-problem, og denne kan løses af LARS med Lasso-modifikationen for at give Elastic Net løsningen.

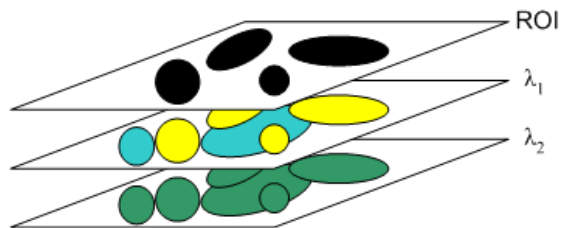
Det forudsættes at respons har middelværdi 0 og prediktorerne er normaliseret til gennemsnit 0 og ligger mellem  $\pm 1$ , derved undgås et konstantled i modellen.

$$(13.6) \quad \text{Minimér} \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda_1 \sum_{j=1}^p \beta_j + \lambda_2 \sum_{j=1}^p \beta_j^2$$

Vælges  $\lambda_1 = 0$  findes alle Ridge-løsninger og vælges  $\lambda_2 = 0$  findes alle Lasso-løsninger. Defineres  $\alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2}$  som foreslået af [Zou og Hastie, 2005] kan problemet omformes til

$$(13.7) \quad \text{Minimér} \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 \quad \text{med betingelsen} \quad (1 - \alpha) \sum_{j=1}^p |\beta_j| + \alpha \sum_{j=1}^p \beta_j^2 \leq s$$

Lasso-betingelsen styres af antal iterationer, der vil ligge tæt på. En høj lasso-betingelse (få iterationer) betyder få aktive variable. Ridge-betingelsen styres af  $\lambda$  og bestemmer hvor meget højere parameterverdier skal straffes. For lav straf overfitter modellen og høj straf giver en "glattere" model. For høj straf går parametrene mod 0.



Figur 13.3: Multivariat regression.

### 13.3 Multivariat lineær billed-regression

I forrige afsnit blev det beskrevet hvordan regressionsanalyse benyttes rent statistisk. I dette afsnit ser på hvordan regressionsanalyse bruges på multispektrale billeder.

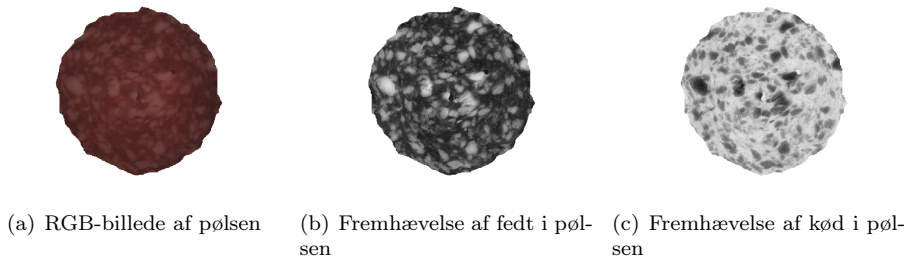
Når man har fundet de ønskede egenskaber i et billede og har trukket dem ud med en ROI, er det tit for at kunne overføre den information man har trukket ud af billedet til et vilkårligt billede. En af de metoder der kan benyttes til at opstille en model ud fra træningsdata er multivariat lineær billed-regression. Det ønskes at finde en funktion af det der udtrykker den ønskede egenskab i det ene billede, som man kan bruge til at finde den samme egenskab i et andet vilkårligt billede af samme type. Et billede med en sammensætning der ligger tæt op af billedet brugt til generering af modellen vil naturligvis kunne give et bedre resultat end et der har større forskelle. Det kan vise sig relevant med flere modeller under analysen af spegepølser, hvor der findes mange forskellige typer. Den generelle regressionsform for billeder er ifølge [Geladi og Grahn, 1996]

$$G_2 = F(G_1)$$

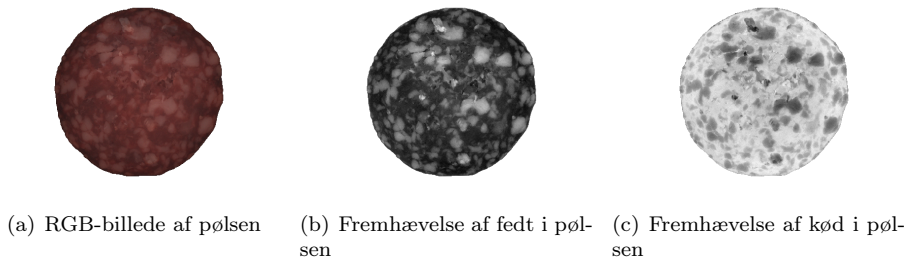
Hvor  $G_1$  er det billede man er interesseret i at trække egenskaber ud af,  $G_2$  er det billede hvori man ønsker at finde den valgte egenskab og  $F$  er funktionen der udtrykker relationen mellem de to billeder.

Princippet bag multivariat regression er illustreret på figur 13.3, hvor en ROI benyttes til at maske den egenskab ud man er interesseret i for alle de forskellige bånd. Derefter opstilles en parameter for hvert bånd der beskriver hvor godt den udvalgte egenskab er repræsenteret i det enkelte bånd, så størrelsen af parametren angiver forstærkningen af det enkelte bånd. Fortegnet kan invertere billedet hvis den udvalgte egenskab har lav intensitet i båndet. Figur 13.3 illustrerer hvordan en egenskab er tydeligt markeret i  $\lambda_1$  og dermed vil blive forstærket i dette bånd, mens den forsvinder blandt sine omgivelser i  $\lambda_2$  og dermed omvendt vil blive tonet ned ved dette bånd.





**Figur 13.4:** Regressionsanalyse af spegepølse. På figur (b) og (c) ses den fremhævede egenskab ved at lysintensiteten i de markerede pixel er høj dvs. med lyse farver.



**Figur 13.5:** Overførsel af regressions parametre til en anden pølse. På figur (b) og (c) ses den fremhævede egenskab ved at lysintensiteten i de markerede pixel er høj dvs. med lyse farver.

Regression på de enkelte bånd er én måde at gøre det på, en anden er at benytte features i form af f.eks. gennemsnit, standardafvigelser og fraktiler for de rå bånd, differencer, forholdet eller multiplikationen af to bånd.

Sammenhængen mellem det multispektrale billede  $G_1$  og regressionsparametrene  $\theta$  er vist i formel (13.8). Understregningen repræsenterer antallet af dimensioner for matricerne.

$$(13.8) \quad \underline{\underline{G_2}} = \underline{\underline{\theta}} * \underline{\underline{G_1}}$$

De tre billeder på figur 13.4 viser tre versioner af det samme spegepølse billede. Det første viser det rene RGB-billede, og de to andre viser regressionsanalysen for billedet. På 13.4(b) ses regressionsanalysen udført ud fra en ROI, der angiver hvilke dele af billedet som er fedt, og 13.4(c) viser den tilsvarende analyse udført på den inverterede ROI, afgrænset til selve pølsen naturligvis, sådan at det er kødet der er masket ud.

Figur 13.5 viser hvordan kødet og fedtet er fundet på et billede af en anden spegepølse ud fra de parametre der blev fundet ved analysen af spegepølsen fra figur 13.4. Det ses tydeligt at selvom det ikke er den samme pølse analysen er lavet på, kan modellen bruges til at finde fedt og kød i billedet.

Hvis man som i dette tilfælde udfører regressionen ud fra en enkelt sample, og dermed ikke har den naturlige variation bygget ind i modellen, risikerer man at modellen bliver overtilpasset (overfittet) til disse specifikke data. For at undgå dette, kan man benytte krydsvalidering. Den store fordel ved krydsvalideringen er at modellen typisk bliver mere generel. Til gengæld risikerer man samtidig at den ikke bliver god nok til formålet. Når vi begynder at kigge på en generel model opstilling til spegepølser, skal vi se på hvordan dette fungerer med vores data.

## 13.4 Multivariat logistisk billed-regression

Princippet for logistisk regression er basalt set det samme som for linear regression beskrevet ovenfor. Den store forskel er at man benytter en logaritme funktion til at finde sandsynligheden for at en given pixel tilhører en given klasse. I vores tilfælde har vi kun 2 klasser som vi vil fitte til, hvilket gør beregningerne en del mere simple end hvis vi havde tre eller flere. Vi benytter et 0/1 responsbillede, hvor 0 er de udvalgte kødpixels og 1 er de udvalgte fedtpixels. Sandsynligheden for at pixel  $x$  tilhører klasse  $g_1$  eller  $g_2$  er udtrykt ved

$$g_1 : p_1(x; \theta) = p(x; \beta) \quad \text{og} \quad g_2 : p_2(x; \beta) = 1 - p(x; \beta)$$

Ifølge [Hastie et al., 2001, eq. 4.20] kan den logaritmiske sandsynlighed skrives som

$$(13.9) \quad l(\beta) = \sum_{i=1}^N y_i \beta^T x_i + \log(1 + e^{\beta^T x_i})$$

hvor  $x_i$  er en vector af prediktorpixels og  $y_i$  er den tilsvarende responsvektor. For at finde de maksimale sandsynligheder differentieres ligning (13.9) og sættes lig nul

$$(13.10) \quad \frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^N x_i (y_i - p(x_i; \beta)) = 0$$

For at løse ligning (13.10) benyttes Newton-Raphson algoritmen. Algoritmen kan omformuleres til matricelnotation som en *iteratively reweighted least-square* [Hastie et al., 2001, eq. 4.26]

$$\begin{aligned}
 \beta^{new} &= \beta^{old} + (X^T W X)^{-1} X^T (y - p) \\
 &= (X^T W X)^{-1} X^T W (X \beta^{old} + W^{-1} (y - p)) \\
 (13.11) \quad &= (X^T W X)^{-1} X^T W z
 \end{aligned}$$

hvor responsen  $z$  er udtrykt ved

$$(13.12) \quad z = X \beta^{old} + W^{-1} (y - p)$$

Algoritmen løses iterativt hvor  $\beta$  starter med at blive sat til 0, og for hver iteration bliver  $\beta^{old}$  sat til den nyligt udregnede  $\beta^{new}$ . Den absolutte forskel mellem  $\beta$  parametrene beregnes for hver iteration og der fortsættes indtil forskellen er mindre end en predefineret tolerance værdi. Vi har valgt en fast toleranceværdi på  $10^{-4}$  da dette virker som en værdi der generelt har været mulig at opnå indenfor 15-20 iterationer, og samtidig giver et godt resultat.

### 13.4.1 Halvt-skridt metoden

Da den logistiske regression kan komme ud for ikke at konvergere og man derfor opnår en cyklisk effekt, benyttes en metode hvor man for hver iteration halverer en parameter *stepsize* sådan at

$$stepsize^{new} = stepsize^{old} * \frac{1}{2}$$

hvor *stepsize* ganges på  $y - p$  leddet i formel (13.12) for hver iteration.



Del IV

Dataanalyse



Denne del beskriver hvordan teorien bliver anvendt for de tre emner *fedt/kød segmentering af spegepølser, barrierelags tykkelse for kiks og vandindhold i brød*.

De fleste af de billeder der er brugt til analysen er genereret af Danisco. Det drejer sig om alle spegepølse og brød billeder. Billederne af kiksene har vi selv optaget efter at have udarbejdet materialet som beskrevet i afsnit 6.1.

Alle billeder er taget på den grå baggrund som VideometerLab systemet har som standard. Det er derfor vigtigt at få fjernet denne fra billederne. Dette gøres under preprocesseringen. For spegepølser og kiks er der ikke konstateret huller i de fotograferede emner, eller steder hvor man kan se helt eller delvist igennem. I disse tilfælde er det underordnet hvilken baggrund billederne iøvrigt er taget på. For brødene er der konstateret huller i nogle af skiverne, og der findes områder hvor man kan ane baggrunden igennem brødstrukturene. I denne del ser vi nærmere på betydningen af disse huller og hvad der kan gøres ved det.

Ved henvisning til *sample X* eller *Marie 1* menes de spegepølse og kiks samples der er angivet i bilag B og H.





# Preprocessering

---

De multispektrale billeder vi benytter har som udgangspunkt en opløsning på 1280 x 960 pixels og 18 bånd. Gemt som doubles (8 byte/pixel) i ukomprimeret tilstand fylder hvert billede derfor godt 177mb. Mange af beregningerne er relativt krævende, så det er i høj grad relevant at sørge for at beregningerne kun foretages på den eller de dele af billedet, der er interessante. Dette vil gøre hukommelsesforbruget mindre, hvilket er en stor fordel da Matlab, der benyttes til beregningerne, benytter en hukommeshåndtering der gør at alle variable bliver allokeret som en kontinuerlig blok hukommelse og Matlab har ikke nogen automatisk funktion indbygget til at reorganisere hukommelsen [MathWorks, 2006].

For at klargøre og begrænse dataene til kun at omfatte det område vi er interesseret i at kigge på, udføres en beskæring af billedet. Preprocesseringen udføres fortrinsvist på billederne af kiks med barrierelag og spegepølserne. For billederne af brød er disse skridt ikke nødvendige, da brødene fylder langt det meste af billedet og det kun er den midterste del der bruges, dog indeholder nogle få af brøddende huller der bør fjernes før selve analysen.

Fremgangsmåde for billed-beskæring består af følgende trin:

1. Fjern ”døde” pixels.
2. Find det bedst egnede bånd og intensitet til at adskille forgrund og baggrund.
3. Generér *Region of Interest* (ROI) ud fra den fundne threshold og bånd.
4. Erodér ROI n antal gange, hvis det er specificeret.
5. Sæt alle værdier udenfor ROI til 0.
6. Beskær billedet så meget som muligt ind til ROI'en.

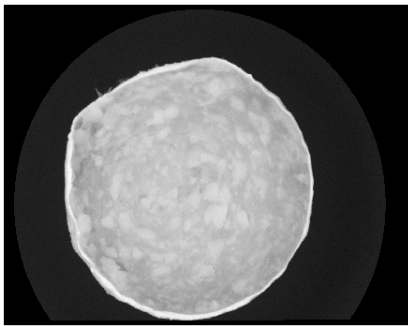
## 14.1 Fjern ”døde” pixels

Uvist af hvilken årsag har vi konstateret, at der i nogle billeder optræder NaN-værdier (Not a Number) enkelte steder, hvilket er problematisk i forhold til mange beregninger. Vi har set dem i visse serier af billederne af kiks med barrierlag, hvor det er konsistent på alle billeder, men kun i bånd 1 og 18. De ligger dog helt ude i kanten af billedet i baggrunden, så i praksis får det ingen betydning. Værdien af disse ”døde” pixels sættes til 0.

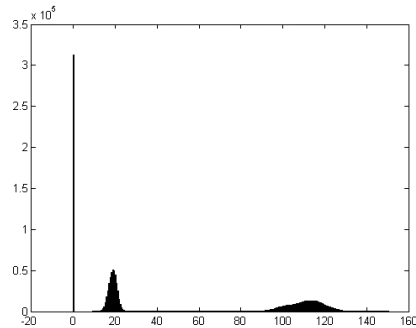
## 14.2 Find forgrund/baggrund

Funktionaliteten til automatisk at finde det bedste bånd til adskillelse af forgrund og baggrund i billedet bygger på den forudsætning, at der findes mindst ét bånd hvor forgrund og baggrund danner hver sin top i et histogram af båndet. Vores billedmateriale viser at dette er en ganske rimelig antagelse at gøre, da der for alle de forskellige fotograferede objekter findes bølgelængder, hvor der er markant forskel på forgrund og baggrund i histogrammet. Der er en risiko for at to forskellige områder i forgrunden bliver dem algoritmen finder, men i praksis har vi ikke konstateret at dette skulle være et problem, formentlig fordi baggrunden fylder relativt meget af billedet og toppen i histogrammet vil derfor være dominerende. Skulle det imod forventning ikke kunne lade sig gøre er det muligt selv at skabe en ROI i et billedbehandlingsprogram og benytte denne. Proceduren til dette er så dog ikke automatisk længere.

På figur 14.1(b) er vist histogrammet for bånd 13 for en spegepølse. Dette bånd viste sig at være bedst til formålet i det konkrete tilfælde med *sample 1* ifølge metoden beskrevet nedenfor. Billedet kan inddeles i tre hovedgrupper: Baggrund der er helt udenfor kameraets område har værdien 0 og udgør klart det højeste maksimum. Baggrund indenfor kameraets område og hovedsageligt har intensiteter på 20 plus/minus 10 i dette bånd. Til sidst er der forgrunden (spegepølsen i dette eksempel) hvis intensitet har maksimum omkring 110-115 i histogrammet og spreder sig plus/minus ca 30.



(a) Det bedste bånd til bestemmelse af for- og baggrund vha. thresholding (bånd 13)



(b) Histogram for bånd 13 af spegepølse (u-behandlet)

**Figur 14.1:** Figuren viser det rå billede af bånd 13 og det tilhørende histogram. Baggrunden kan svagt anes som en mørk cirkel rundt om spegepølsen.

Metoden til automatisk bestemmelse af det bedste bånd og grænseværdi for at adskille forgrund og baggrund midler hvert bånd i billedet indtil der er to maksima tilbage. Midlingen udføres med filteret fra formel 14.1.

$$(14.1) \quad data(i) = 0.25 * data(i - 1) + 0.50 * data(i) + 0.25 * data(i + 1)$$

Filteroperationen er *zero-padded*, dvs. at der for endeværdiernes midling bruges 0 til de indeks der er uden for billedområdet, derfor vil baggrunds-toppen med værdien 0 meget hurtigt forsvinde ved denne midling<sup>20</sup> så der kun er to toppe i billedet tilbage med maksimum større end 0.

For hvert af de midlede histogrammer af hvert bånd udregnes en score baseret på nedenstående kriterier. Et *score*-kriterie angiver at jo **større** tal der findes, jo bedre er båndet til at adskille de to fundne populationer. Et *penalty*-kriterie er modsat, dvs. jo **lavere** tal der findes, jo bedre er båndet til at adskille de to fundne populationer.

<sup>20</sup>Mindst 25% forsvinder ved hver iteration

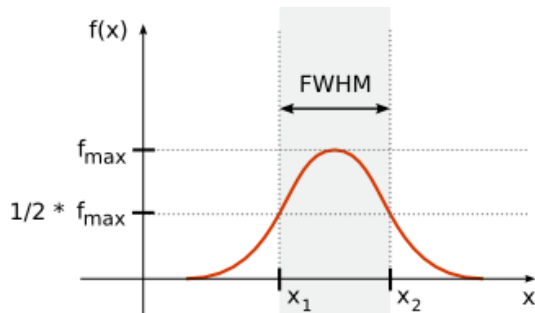
- **"PeakDistanceScore"** - Afstanden mellem de to fundne maksima.
- **"FullWidthHalfMaximumPenalty"** - Summen af de to toppes *Full Width at Half Maximum* (FWHM).
- **"PeakRatioScore"** - Forholdet mellem den mindste og den største tops højde (altid mellem 0 og 1). Er forskellen meget stor (og scoren lille) tyder det på at den lille top måske ikke repræsenterer baggrund med en anden feature i båndet.
- **"MinMaxRatioScore"** - Forholdet mellem højden af den mindste top og højden af den fundne minimumsværdi imellem de to toppe. Skulle minimum være 0, sættes scoren til højden på den mindste top. En stor forskel antyder en god separation imellem toppene på en anden måde end FWHM.

Disse faktorer har langt fra samme betydning i forhold til at finde det bedste bånd, så igennem flere tests med vores data er vi kommet frem til formel 14.2 for den samlede score. Under testen blev det afprøvet hvordan scoren artede sig i forhold til om billedet allerede var rensat eller ej. Ud fra histogrammerne kunne så bestemmes om funktionen skulle straffe relativt mere eller mindre for de forskellige parametre.

(14.2)

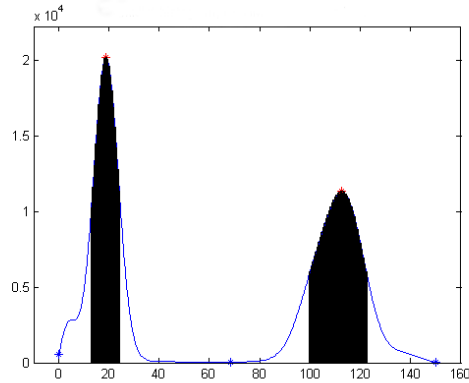
$$score = \frac{PeakDistanceScore^3 * PeakRatioScore * MinMaxRatioScore}{FullWidthHalfMaximumPenalty^2}$$

FWHM er udtrykt ved forskellen mellem de to punkter (på x-aksen), hvor værdien er faldet til halvdelen af det fundne maksimum. FWHM er en anerkendt metode til at give et estimat af spredningen af en fordeling der ikke umiddelbart kan beskrives af en normal-fordeling eller lignende. For normalfordelingen gælder sammenhængen  $FWHM \approx 2,35\sigma$ , hvor  $\sigma$  er standardafvigelsen. Metoden er illustreret på figur 14.2.



Figur 14.2: Full width at half maximum [Wikipedia, 2006a].

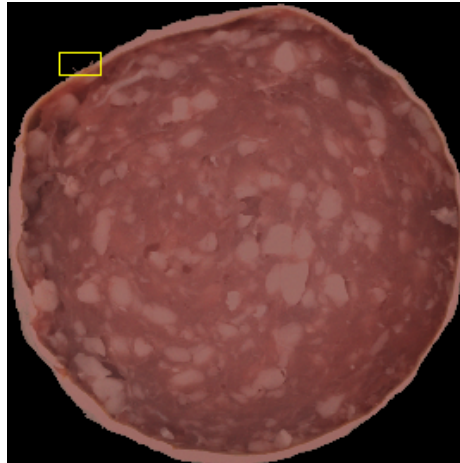
”Score” bliver altså størst når de to toppe ligger langt fra hinanden og bredden af deres fordelinger er lille. Båndet med den højeste score vælges som det bedste til formålet. Grænseværdien (threshold) der skal adskille forgrund og baggrund findes derefter som minimum imellem de to maksima i det fundne bånd, som illustreret på figur 14.3.



**Figur 14.3:** Histogram med minima, maksima og FWHM indtegnat. FWHM er markeret med sort, maksima med en rød stjerne og minima med en blå stjerne, hvor kun det midterste er interessant.

## 14.3 Generér ROI

Ved hjælp af simpel thresholding genereres den binære ROI ud fra den fundne threshold værdi og bånd. ROI'en renses derefter med *closing* metoden for mindre huller i både forgrund og baggrund. Det struktur element der benyttes til at lukke hullerne med, skal være tilpas stor til at mindre urenheder i form af enkelte pixels eller mindre klaser af pixels bliver fundet og fjernet, og samtidig tilpas lille for at undgå at fjerne betydende egenskaber i billedet. Her er valgt en ”skive” med en diameter på ca. 20 pixels, svarende til 1,5mm jvf. tabel A.2. På figur 14.4 ses en udvalgt region af en spegepølse, før og efter oprensningen af ROI'en. Det kan ses hvordan den lille klump af pixels der svæver uden for spegepølsen bliver fjernet, og samtidig bliver kanten ”glattet” ud, sådan at der fremkommer bløde pukler i stedet for de spidse ”kant-tråde”.



(a) Det valgte område før oprensning. Den gule firkant markerer det valgte område.



(b) Før oprensning



(c) Efter oprensning

**Figur 14.4:** Figuren viser et udvalgt område af det thresholdede billede før og efter oprensning af ROI'en, med den resulterende ROI anvendt på det oprindelige RGB billede.

## 14.4 Fjern kant på ROI

Hvis man ikke ønsker at kanten af det fotograferede objekt skal medtages i de videre beregninger kan den fundne ROI eroderes et antal gange. Groft sagt fjerner eller eroderer hver iteration 1 pixel af kanten hele vejen rundt.

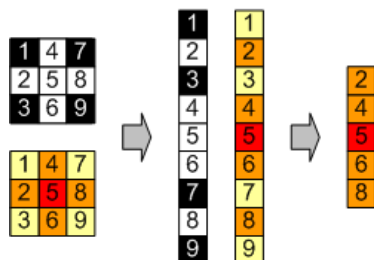
Dette er f.eks. nyttigt i tilfælde med kiksen, da det vi er interesserede i at kigge på er det jævne lag på midten af kiksen, og ikke inddrage forstyrrende elementer og ekstra varians i form af kanten på kiksen, der dels er mørkere end resten af kiksen, og dels ikke er så ensartet dækket af barrierelaget, og i denne sammenhæng derfor er irrelevant at kigge på.

For spegepølserne er vi interesserede i at se farveændringen helt ude fra kanten, så det kan det være en god ide ikke at erodere kanten væk. Kun i specielle tilfælde hvor pølsen f.eks. har en tydelig fedtkant, der ikke er interessant for analysen og derfor vil tilføre mere varians end ønsket, kan det være nødvendigt at fjerne denne. Dette specialtilfælde er yderligere beskrevet nedenfor.

## 14.5 Fjern baggrund

Hvis man under de videre beregninger alligevel kun benytter pixels indenfor ROI'en, har pixels udenfor ROI'en ikke nogen betydning og kan derfor sættes til 0. I forbindelse med visning af billeder opnås på denne måde en klart defineret baggrund.

Ved beregninger på hele billedet kan baggrunden fjernes fuldkommen ved at foretage en omstrukturering af pixelværdierne sådan at de kommer til at ligge i én lang stribe. Ved samtidig at foretage den samme omstrukturering af ROI'en og fjerne alle de pixels der i ROI'en er 0 opnås at beregningerne kun bliver udført på pixel værdier indenfor ROI'en. Denne mekanisme er illustreret på figur 14.5 hvor den sort/hvide figur repræsenterer ROI'en hvor sort er 0, dvs. det vi vil fjerne og hvid er 1, dvs. det vi er interesserede i.



**Figur 14.5:** Fjernelse af pixels uden for ROI ved omstrukturering.

## 14.6 Beskær billedet

Det er nu muligt at skære billedet ned til det mindst mulige rektangel<sup>21</sup>, der stadig indeholder hele ROI'en. I de fleste af de tilfælde vi beskæftiger os med vil beskæringen give et billede der fylder 40-60% af det oprindelige billede.

## 14.7 Resultat

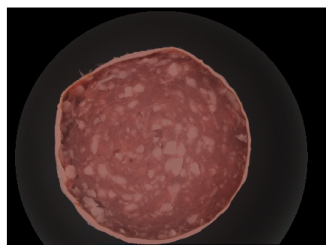
Figur 14.6 viser RGB-billeder af en pølse, en Mariekiks og et stykke brød før og efter den automatiske preprocessing. Det ses på figuren hvordan hele baggrunden er blevet skåret væk, sådan at kun emnet står tilbage.

Da oprensningen af billederne er noget af det der tager længest tid ved f.eks. spegepølseanalysen, er det ønsket at gemme de allerede rensede billeder, så det undgås at det skal gøres flere gange end højst nødvendigt. Derfor skal oprensningen også fungere med billeder der allerede er rensede.

I tabel 14.1 ses gennemsnit, minimum og maximum værdier for den udregnede threshold-score for billederne fra spegepølse, Mariekiks og brød dag 1 serierne, før og efter at de er blevet rensede. Ud fra disse tal ses det at gennemsnitsværdien for de ikke-rensede billeder er meget højere end for de billeder der allerede er rensede. Selvom gennemsnitsværdien for de ikke-rensede billeder er så høj, er minimumsværdien stadig meget lav, det ses også at der er et godt skel mellem minimum værdien for de ikke-rensede billeder og maximum for de rensede. Da disse to værdier anses som ekstremer inden for vores datasæt, vælges en grænseværdi der ligger midt mellem de to, på 10, sådan at hvis billedet har en

<sup>21</sup>Vi benytter et rektangel, da dette er nemt at beregne på, er let at repræsentere med matricer og det bevare den spatielle information i billedet

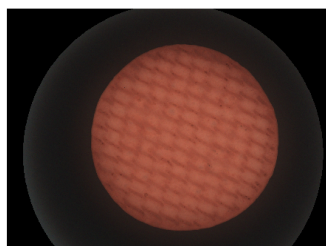




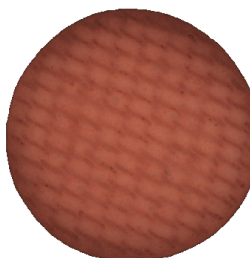
(a) Spegepølse - før



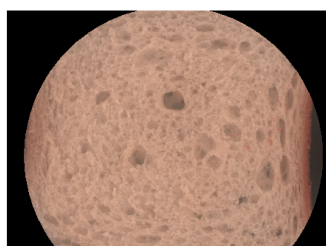
(b) Spegepølse - efter



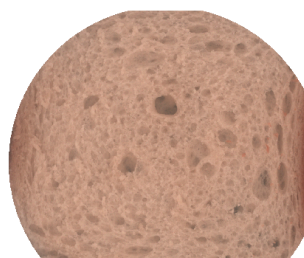
(c) Mariekiks - før



(d) Mariekiks - efter



(e) Brød - før



(f) Brød - efter

**Figur 14.6:** Figuren viser resultatet af den beskrevne preprocessering for de tre emner, spegepølse, Mariekiks og brød.

	Før	Efter
Gennemsnit	$1.0382 * 10^6$	0.6062
Minimum	20.5232	0.0037
Maximum	$5.1724 * 10^6$	3.8307

**Tabel 14.1:** Gennemsnit, minimum og maximum for scoreværdier før og efter preprocessing.

thresholdscore på mindre end denne værdi thresholdes billedet ved bånd 1 og ved en værdi større end nul.

Fra de spegepølse- og kiksedatasæt vi har fået stillet til rådighed er der ikke eksempler på at algoritmen fejler. For brødet er der tilgængelig visse problemer, da der som nævnt ovenfor er konstateret huller i brødskiverne hvor man helt eller delvist kan skimte baggrunden igennem. Figur 14.7 viser oprensningen af et stykke brød, hvor der er zoomet ind på et område af brødet. På figur 14.7(b) og 14.7(c) kan det ses at baggrunden i hullet i brødet bliver fundet, men at en del af den bliver taget med ved oprensningen af ROI'en. I kapitel 17 vil vi se på hvad der kan gøres for at forbedre problemet.



(a) Udvalgt region før oprensning (b) Før oprensning af ROI (c) Efter oprensning af ROI

**Figur 14.7:** Figuren viser oprensningen af et brød. Figur (a) viser en udvalgt region før oprensning af billedet, og figur (b) og (c) viser regionen før og efter oprensningen af ROI'en.

# Spegepølser

---

Analysen af spegepølserne har til formål at finde en robust metode til adskillelse af kød og fedt, og derfra udtrykke farven for hver af disse i en eller flere ringe, bestemt ud fra afstanden til kanten.

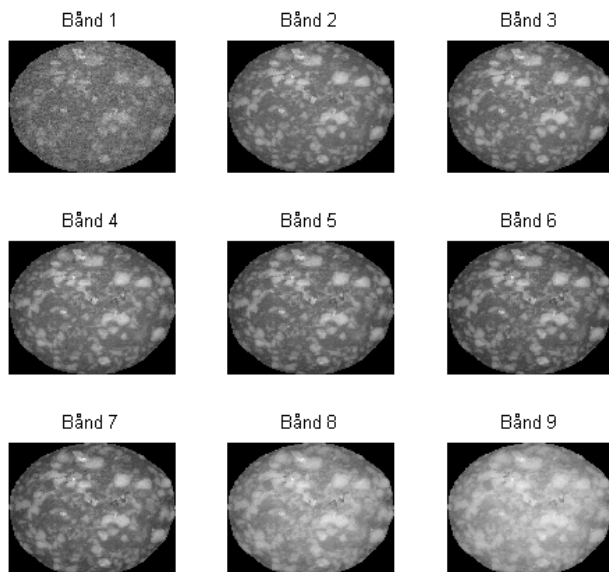
Datasættet består af billeder fra fem forskellige spegepølsetyper der farvemæssigt betegnes som ekstremerne indenfor Danisco's sortiment. Først analyseres hvad det er muligt at se på billederne direkte, derefter udføres en række forberedende operationer inden den egentlige analyse for at forbedre resultatet af den metode der skal kunne bestemme kød fra fedt. Til sidst i kapitlet præsenteres resultaterne.

## 15.1 Analyse af de rå billeder

På de rå billeder på figur 15.1 og 15.2 kan det ses, at fedtet i pølsen træder relativt tydeligt frem som lyse områder især på bånd 2-7 (fra blå til kortbølget rød bølgelængde). På figur 15.3(a) ses histogrammet for bånd 5, hvor der ses en hale til højre der repræsenterer det lyse fedt. Forskellen mellem kød og fedt er ikke stor nok til at der direkte kan ses to toppe. Vi vil senere se på hvorfor.

Fedt og kød kan i nogen grad også adskilles visuelt på nogle af de øvrige bånd. På nær-infrarød båndene (11-18) kommer kød og fedt intensiteterne dog tættere på hinanden og kun meget tydelige forskelle træder frem, se figur 15.3(b).

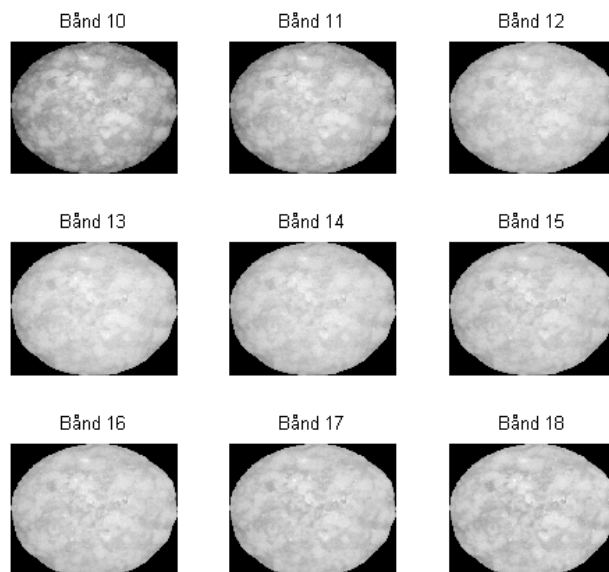
Spegepølse *sample 2* har en betydelig fedtkant (se figur 15.4), som især på de lave bånd er meget tydelig med højere intensitet end både kød og det øvrige af fedtet i pølsen.



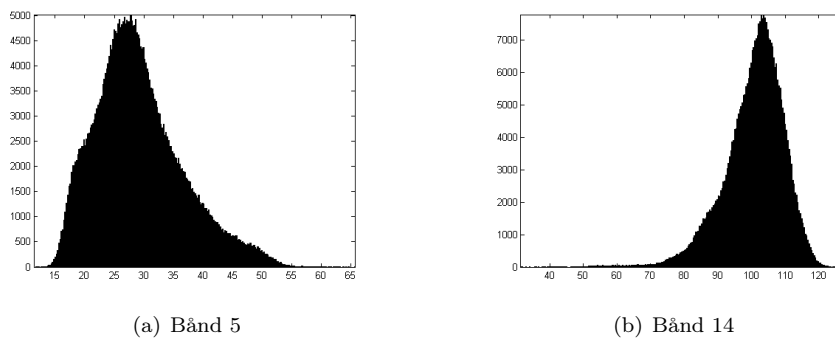
Figur 15.1: Bånd 1-9 for spegepølse *sample 1*.

## 15.2 Forbehandling af spegepølsebilleder

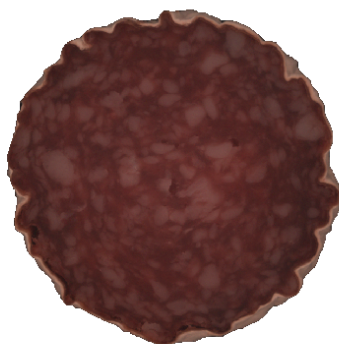
Som beskrevet i kapitel 14 renses billederne op sådan at det kun er selve pølsen der står tilbage og baggrunden er fjernet eller sat til 0.



Figur 15.2: Bånd 10-18 for spegepølse *sample 1*.



Figur 15.3: Histogrammer for udvalgte bånd af spegepølse *sample 1*.



**Figur 15.4:** RGB billede af *sample 2* med fedtkant.



(a) Fedtkanten der er eroderet væk

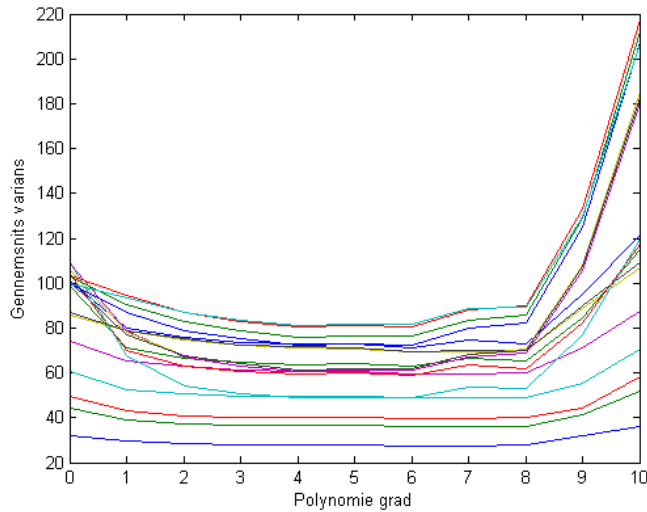


(b) Resultatet af eroderingen

**Figur 15.5:** Pølse ROI'en eroderes så fedtkanten fjernes.

### 15.2.1 Fjern fedt-kant

Spegepølse *Sample 2* har en tydelig fedt-kant der ikke er interessant i denne sammenhæng da vi primært interesserer os for kødfarven inderst, og i ringe ud af mod kanten af pølsen. Derfor kan denne med fordel fjernes før de forskellige beregninger påbegyndes, ellers risikeres det at f.eks. MAF analysen fejler ved at genkende kanten som den vigtigste egenskab, hvilket ikke er ønskeligt. Kanten fjernes ved at erodere den fundne ROI et antal gange, typisk et sted mellem 20 og 30 iterationer. På figur 15.5 er vist hvordan fedt-kanten af pølsen fjernes ved erodering med 20 iterationer.



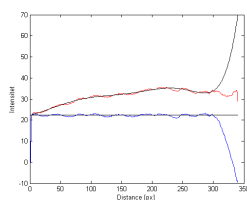
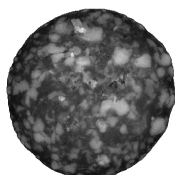
**Figur 15.6:** Figuren viser gennemsnitsvariansen for alle samples i hvert bånd overfor graden af polynomiet der er brugt til at fitte med.

### 15.2.2 Intensitetsniveau udligning

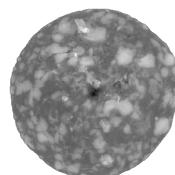
Det kan tydeligt ses på pølsebillederne hvordan farve intensiteten stiger ind mod midten af pølsen. Dette kan gøre det sværere at skelne fedt fra kød, eftersom intensiteten af det lyseste kød godt kan være lysere end farven af det yderste fedt.

For at forbedre resultatet af den videre analyse er det derfor ønskeligt at justere intensiteten af farven som en funktion af afstanden til kanten, da vi som hovedregel kan gå ud fra at pølsen er modnet homogent, og derfor vil have den samme ændring i farveintensitet hele vejen rundt om pølsen i en given afstand til kanten. Vi kan også se på figur 15.8 at stigningen ikke er lineær, da farveintensiteten hurtigt stiger ved kanten, men lægger sig så for at stige mere jævnt ind mod midten.

Hovedidéen med intensitetsudligningen er at få intensiteterne i kødet og fedtet til at være ens ved kanten og inde i midten. Derfor er det forventeligt at variansen over det samlede billede vil blive mindre jo bedre fittet er. På figur 15.6 ses gennemsnitsvariansen for alle samples i hvert bånd overfor graden af polynomiet der er brugt til at fitte med. Det kan ses at kurverne for variansen er lavest omkring 3. til 6. grads polynomier.

(a) Intensitets niveau udlig-  
ningen

(b) Før udligningen



(c) Efter udligningen

**Figur 15.7:** Intensitets niveau udligning med et 7. grads polynomie vist ved bånd 7 i spegepølser.

Et 2. grads polynomie synes lidt for ufleksibelt og følger ikke data i samme grad som et polynomie med højere rang. Et polynomie af rang på 5 eller mere har tendens til at blive for fintfølede, og kan i "enderne" finde på at lave meget store udsving fordi polynomiet bliver overfittet til data. Derimod fungerer polynomier af 3. og 4. grad godt med de givne data, hvor 5. og 6. grads polynomier har tendens til at overfytte. Da et 4. grads polynomie er mere fleksibelt end et 3. grads polynomie er dette valgt som fitting kurve. På figur 15.7 ses et eksempel på hvordan et 7. grads polynomie har resulteret i overfitting til data. Den mørke plet i midten af spegepølser er det direkte resultat af det meget store udsving den fittede kurve udfører ved de inderste distancer.

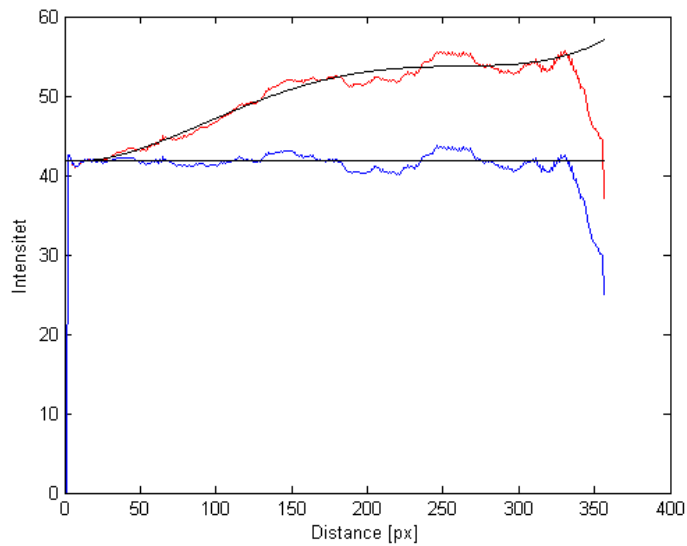
På figur 15.8 ses grafen for de oprindelige data, det fittede 4. grads polynomie og de fittede data. Til dette eksempel er valgt bånd 7 fra billedet, fordi det viser relativt stor intensitetsforskel fra inderst til yderst. På grafen ses det tydeligt hvordan den inderste del af pølser, dvs. ved de høje distancer omkring 370-400px pludselig laver et stort udsving nedaf. Dette skyldes at antallet af pixels for en given distance falder mod midten, og de enkelte pixels vil derfor have stor indflydelse på gennemsnits værdien.

På et billede med så grove strukturer som f.eks. fedtklumper i pølser, er det ikke usandsynligt at der inde omkring midten af billedet vil være en høj koncentration af f.eks. fedt eller kød pixels. Da vi kigger på gennemsnitsintensiteter er det tydeligt at dette vil få grafen til at svinge meget kraftigt enten op eller ned.

Det er derfor nærliggende at vælge at skære de inderste distancer fra ved beregning og fitting af polynomiet. Ved at skære 10% fra distancen inderst opnås kurven som er vist på figur 15.8. Det ses at det udsving der sker i slutningen af grafen ikke synes at have nogen indvirkning på den fittede linies forløb.

I tabel 15.1 ses *first order gray-level* (FOGL) statistikkerne for billedet før og efter intensitetsudligningen med et fittet 4. gradspolynomie. Mange af parametre-





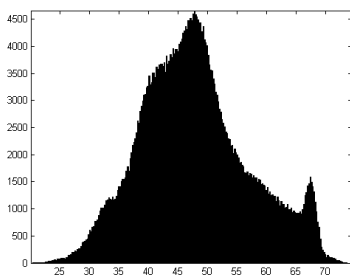
**Figur 15.8:** Figuren viser hvordan en linie er fittet til data som funktion af afstanden til midten. Den røde graf viser de oprindelige data, og den blå viser de værdier dataene er blevet fittet til. På figuren er brugt et 4. grads polynomie til at fitte med.

trene ændres stort set ikke, men det ses at gennemsnitsværdien falder med ca 9% som udtryk for at intensiteten i midten af spegepølser falder, sådan at den ligger på niveau med den ydre del. Variansen falder markant med 20, hvilket kan ses ved at fordelingen bliver meget smallere. Derudover er der to andre parametre der ændrer sig, nemlig *Skewness* og *Kurtosis*.

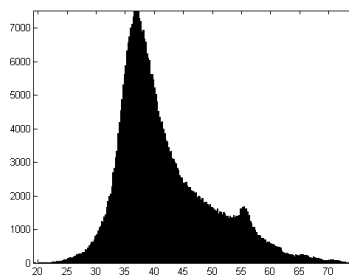
På figur 15.9 ses histogrammerne for bånd 7, før og efter intensitetsopretningen. Det ses tydeligt hvordan massemidtpunktet flyttes til venstre og dermed trækker en hale ud til højre, som afspejlet i den forhøjede positive *Skewness*. Det kan også ses at selve toppen bliver smallere og bliver mindre symmetrisk, som afspejlet i *Kurtosis* der skifter fortegn ved at blive næsten 1 større. Alt dette betyder at det nu er en mere ensartet pølse hvor det er nemmere at skille fedtet fra køddet, da man undgår at algoritmerne forveksler intensiteterne for det inderste kød med intensiteterne af det yderste fedt.

	Før	Efter
Mean	47.8325	41.7785
Variance	84.0488	64.3809
Std. deviation	9.1678	8.0238
Covariance	0.1917	0.1921
Skewness	0.3079	0.9611
Kurtosis	-0.2656	0.6257
Energy	0.0318	0.0447
Entropy	-3.5924	-3.3631

**Tabel 15.1:** FOGL statistikker før og efter intensitetsniveauudligningen af et billede.

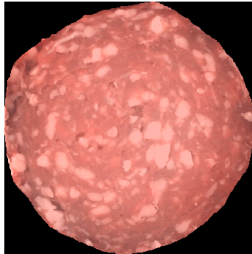


(a) Oprindeligt bånd 7

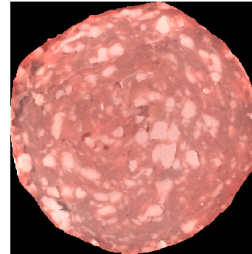


(b) Korrigeret bånd 7

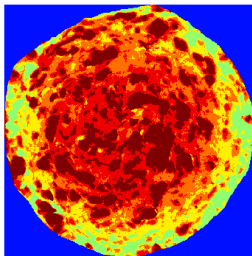
**Figur 15.9:** De to histogrammer viser farveintensitetshistogrammerne for bånd 7 i en spegepølse, før og efter at korrigeringen er foretaget.



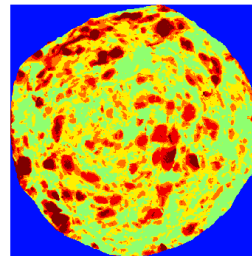
(a) Oprindeligt RGB billede af spegepølse



(b) Korrigeret RGB billede af spegepølse



(c) Farvekodet uden intensitets opretning



(d) Farvekodet med intensitets opretning

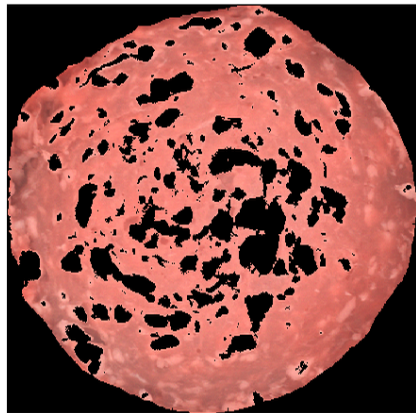
**Figur 15.10:** Farveintensitets oprettet RGB billede af spegepølse. På (c) og (d) ses det farvekodede 1. komponent billede af en PCA analyse på billedet.

Værdi	Farve
0-10	Mørkeblå
10-20	Blå
20-30	Lyseblå
30-60	Grøn
60-65	Gul
65-70	Orange
70-75	Rød
75-100	Mørkerød

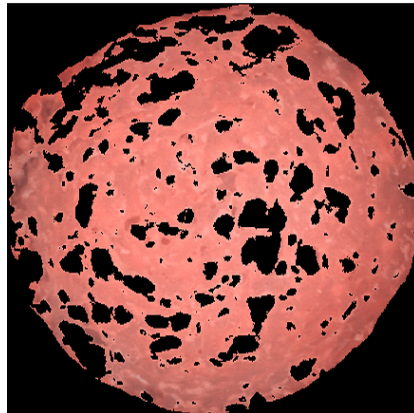
**Tabel 15.2:** Farvekodningsskema benyttet til PCA-komponenterne vist på figur 15.10.

På figur 15.10 ses resultatet af det oprettede billede vist som RGB billede. Det ses tydeligt hvordan stigningen i farveintensitet er forsvundet i det oprettede billede sådan at intensiteten af farven af kød og fedt er den samme ved kanten og inde i midten. På de farvekodede billeder af den 1. PCA komponent ses det endnu tydeligere hvordan store plamager af kød inderst i det ukorrigerede billede har den samme farve som noget af det yderste fedt og i nogle tilfælde endda højere. Hvor det på det korrigerede billede har den samme farve yderst og inderst.

På figur 15.11 ses hvordan en ROI er genereret ud fra, og derefter overført til, det oprindelige billede og det intensitets oprettede billede. ROI'en er genereret ud fra en scoreplot analyse som beskrevet i afsnit 12.6. Det ses hvordan fedtet i midten af den ukorrigerede pølse er fjernet, men der er samtidig taget for meget kød med. Samtidig er fedtet i kanten ikke fundet i særlig overbevisende grad. På det korrigerede billede kan det ses at fedtet i midten er fjernet, uden at tage for meget kød med og fedtet i kanten er samtidig også fundet. På det intensitetsoprettede billede ses det dog også at der er et lidt større hul i øverste venstre hjørne hvor detekteringen ikke er blevet helt vellykket da der er fjernet for meget kød. På ingen af de to billeder er fedtet nederst i pølsen detekteret. Den overordnede detektering er dog tydeligt forbedret.



(a) Uden intensitetsopretning



(b) Med intensitetsopretning

**Figur 15.11:** Figuren viser hvordan fedt og kød er skilt fra hinanden med og uden farveintensitetsopretning.

## 15.3 Kød/fedt adskillelse

I dette afsnit beskrives de metoder der er anvendt til adskillelse af kød og fedt i et spegepølsebillede. Disse metoder er

- PCA til manuel klasseadskillelse og derefter lineær regressionsanalyse
- PCA eller MAF kombineret med adaptiv thresholding

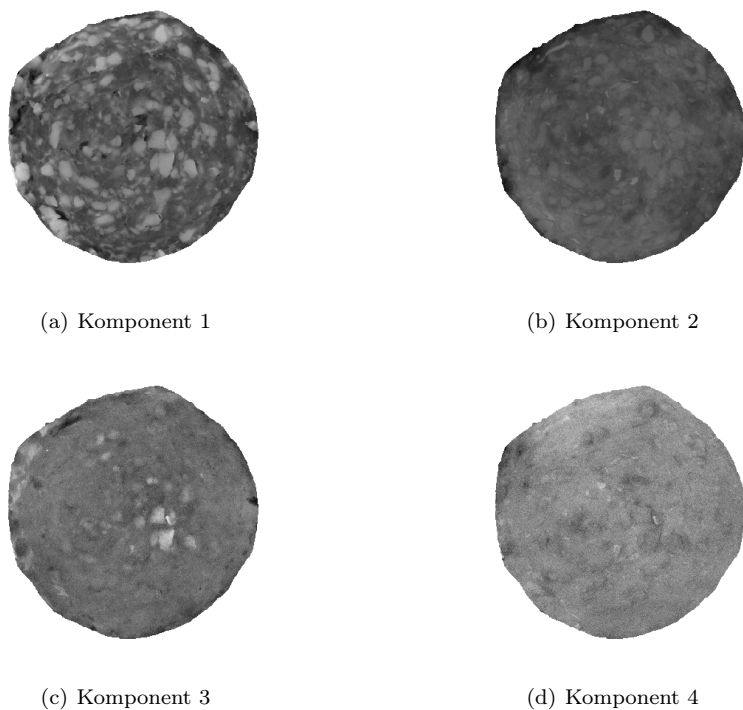
### 15.3.1 Principal komponent analyse

For at finde fedt- og kød-strukturer i billedet udføres en PCA analyse på billedet. Resultatet kan ses på figur 15.12 hvor de fire mest betydende komponenter er valgt ud. Komponent-billederne fra 5 og opefter indeholder primært støj som ikke er interessant for den videre analyse af kød/fedt adskillelsen.

Ud fra de fire billeder er det tydeligt at se at komponent 1 er den der adskiller fedt og kød bedst. I komponent 2 billedet kan også ses forskellen mellem kød og fedt, men ikke i samme grad som komponent 1. I tabel 15.3 er vist gennemsnitsværdier for kød og fedt adskillelse for *sample 1*, baseret på en simpel thresholding i PCA billedet, hvor værdierne på forhånd er blevet skaleret til at ligge mellem 0 og 1. I kolonnen der viser den absolutte forskel, og dermed kontrasten mellem de to klasser, ses det at forskellen mellem kød og fedt er ca. 9 gange større for komponent 1 end for komponent 2.

Det er ikke en selvfølge at det forholder sig sådan, og det er vigtigt altid at sikre sig visuelt at det gør, så man ikke får masket de forkerte egenskaber ud under model-generering. På figur 15.13 er vist et eksempel på dette, hvor det ser ud til at komponent 2 muligvis er bedre til kød/fedt opdelingen, eller i hvert fald er lige så god. Tabel 15.4 viser gennemsnitsværdierne for det skalerede fundne kød og fedt i *sample 2*. Disse tal bekræfter at forskellen i gennemsnitsværdierne mellem kød og fedt er meget lille for de to komponenter, hvor komponent 3 er så godt som ubrugelig, hvilket score-billedet også viser.

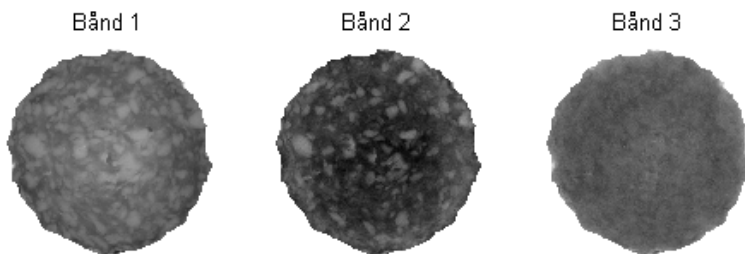
På figur 15.14 ses de farvekodede PCA komponenter, kødet med værdierne fra tabel 15.2. De enkelte komponent billeder er på forhånd skaleret sådan at pølsens værdier ligger mellem 0 og 100.



**Figur 15.12:** Figuren viser de fire mest betydende komponenter fra PCA analysen vist som score-billeder.

	Kød	Fedt	Absolutte forskel
<b>Komponent 1</b>	0.6891	0.9400	0.2509
<b>Komponent 2</b>	0.3206	0.2894	0.0312

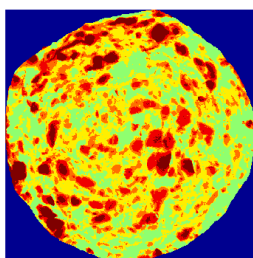
**Tabel 15.3:** Gennemsnitsintensiteter fedt/kød fordelinger i et PCA score-billede af *sample 1*. PCA billedet er forinden blevet skaleret til at ligge mellem 0 og 1.



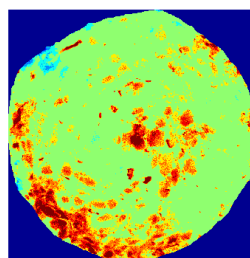
**Figur 15.13:** PCA komponent 1-3 for spegepølse *sample 2* uden fedtkant.

	Kød	Fedt	Absolutte forskel
<b>Komponent 1</b>	0.3784	0.4571	0.0788
<b>Komponent 2</b>	0.2257	0.3061	0.0804
<b>Komponent 3</b>	0.3795	0.3620	0.0175

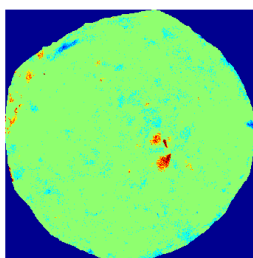
**Tabel 15.4:** Gennemsnitsintensiteter af normaliserede fedt/kød fordelinger i et PCA score-billede for *sample 2*.



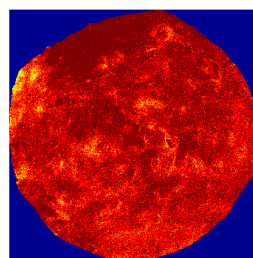
(a) Komponent 1



(b) Komponent 2



(c) Komponent 3



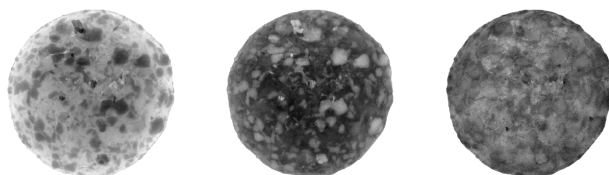
(d) Komponent 4

**Figur 15.14:** Figuren viser de fire mest betydende komponenter fra figur 15.12 farvekodet med skemaet fra tabel 15.2. Billederne er på forhånd skaleret sådan at billedet har værdier mellem 0 og 100.

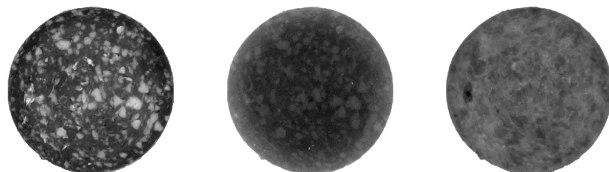
### 15.3.2 Minimum autokorrelationsfaktorer (MAF)

På samme måde som for PCA vil vi her se på muligheden for at kunne kende forskel på kød og fedt i en spegepølse vha. MAF. På bilag D er vist de tre mest betydende komponenter fra MAF analysen for hver af de fem samples. *Sample 2* indeholder en betydelig fedtkant, og i billedet hvor denne stadig er med bliver dette klart den mest betydende egenskab og dermed udtrykt på komponent 1. Hvis spegepølsen indeholder en fedtkant, skal denne eroderes væk, som beskrevet i afsnit 12.4.

Figur 15.15 og 15.16 viser hhv MAF billeder af *sample 1* og *3* og de tre mest betydende komponenter for hver. For *sample 1* kan vi se at både komponent 1 og 2 udtrykker forskellen på fedt og kød relativt tydeligt og at de er inverteret i forhold til hinanden. For *sample 3* er fedt/kød forskellen mere tydeligt koncentreret på komponent 1, dog kan strukturen stadig fornemmes i komponent 2 og 3. Det er værd at bemærke at kødet i *sample 1*, komponent 1 har de højeste intensiteter (de lyseste farver) imens kødet i komponent 1 for *sample 3* og de øvrige samples har de laveste intensiteter. Dette giver en ekstra udfordring når fedt og kød skal adskilles.



Figur 15.15: MAF bånd 1-3 for spegepølse *sample 1*.



Figur 15.16: MAF komponent 1-3 for spegepølse *sample 3*.

### 15.3.3 Manuel udvælgelse af egenskabsregioner

Vi kan se på spegepølserne, at kødet generelt set er mørkere end fedtet, og at det på både PCA og MAF billederne er netop kød/fedt opdelingen der typisk bliver



fundet ved de første komponenter. Derved får fedtet typisk højere værdier end kødet i det resulterende billede, og det er derfor muligt at benytte thresholding til adskillelse af fedt og kød. Da vi til modelgenerering ikke er interesseret i de områder af pølsen som vi ikke er helt sikre på er enten fedt eller kød, definerer vi tre områder. Et kød-, et fedt- og et enten-eller-område.

For at finde disse tre områder er man nødt til at prøve sig lidt frem, og se på resultaterne om opdelingen ser fornuftig ud. Alternativt kan man tage et billede som males med tre farver, som derefter defineres som forgrund, baggrund og enten-eller områder. Denne metode kan vise sig at være relativt langsommelig at udføre, samt at den ikke er reproducerbar i samme grad som thresholding-metoden.

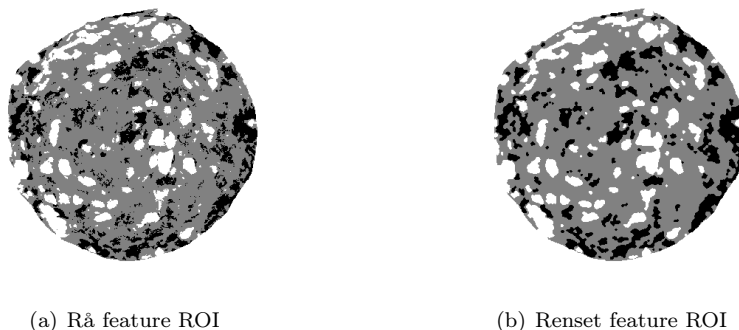
På figur 15.17(a) ses en pølse der er delt op i disse tre områder. En simpel thresholding tager ikke hensyn til de omkringliggende pixels, og med stor sandsynlighed kan man sige at de pixels der f.eks. ligger omringet af en specifik klasse i en specifik diameter, højst sandsynligt også tilhører den omkringliggende klasse. Derfor udføres først en lukning af huller i forgrunden, dvs. de to egenskabsklasser, og derefter en lukning af baggrunden. Ved lukning af forgrunden, opnås at mindre huller bliver lukket, sådan at områder med meget kød eller fedt ikke indeholder små områder af pixels som efter al sandsynlighed tilhører den omkringliggende klasse, som beskrevet ovenfor. Lukning af baggrunden udføres for at der ikke optræder enkelte pixels eller mindre klumper af pixels som mere eller mindre tilfældigt er valgt ud.

Med en strukturelementstørrelse til åbning og lukning på 7 pixels<sup>22</sup> i diameter opnås at kun objekter eller huller med en diameter på ca. 0,5mm fremgår af den endelige ROI som vist på figur 15.17(b).

### 15.3.4 Automatisk udvælgelse af egenskabsregioner

Som beskrevet er det muligt med en simpel thresholding på et PCA eller MAF billede til at skabe en ROI, der opdeler fedt og kød. Problemet med denne metode er at den er manuel så brugeren selv skal vælge grænserne der opdeles efter. Metoden er altså relativt omstændig, samtidig med at den bygger på en subjektiv vurdering som kan svinge fra person til person alt efter hvem der foretager analysen. Desuden er den globale thresholding ikke specielt fleksibel overfor regionale intensitetsforskelle.

<sup>22</sup>Pixelstørrelsen er på ca.  $0,077 \frac{mm}{px}$  jvf. tabel A.2. De 0,5mm er valgt ud fra at det virker som en passende størrelse hvor de endeligt viste elementer tydeligt kan ses og genkendes i det oprindelige billede.



**Figur 15.17:** Udvalgelse af egenskabs-ROI. De sorte pletter indikerer udvælgelsen af fedt, og de grå er udvælgelsen af kød. Figur (b) viser den rensede ROI hvor kun elementer med en diameter større end 0,5mm er vist.

For at løse disse problemer har vi derfor kigget på en adaptiv thresholding i stedet, hvor fordelene er tydelige.

- Den kræver ikke en kendt eller opgivet grænse at thresholde efter.
- Den er automatisk.
- Den kan tilpasse sig efter lokale intensitetsforhold i billedet.

Der findes flere algoritmer til at udføre adaptiv thresholding [Fisher et al., 2003], så de primære faktorer i valget af algoritme var at den er hurtig og let at implementere. Den konkrete algoritme vi har valgt benytter sig af følgende formel:

$$(15.1) \quad \text{FEATUREROI} = (IM - mIM) > C$$

$IM$  er det gråtonebillede man ønsker thresholdet.  $mIM$  er et billede med de lokale gennemsnitsintensiteter bestemt ud fra en fast defineret vinduesstørrelse. Der er også mulighed for at benytte medianen med tilsvarende resultater. Medianen vil være mere robust overfor outliers, men mindre anvendelig hvis data er opdelt i to populationer. Vi vælger at benytte gennemsnitsværdien i det følgende da vi ikke har konstateret nogle voldsomme outliers i datasættet, og da spegepølserne per definition er opdelt i de to populationer, fedt og kød, der dog ligger tæt med en flydende overgang. I det ekstreme tilfælde er vinduesstørrelsen valgt svarende til hele billedet, hvor man vil få samme resultat som ved global

thresholding. For at opnå det bedste resultat skal størrelsen af vinduet vælges således at de lokale intensitetsforhold i vinduet er relativt ensartede samtidig med at der skal være både den ene og den anden egenskab (i vores tilfælde fedt og kød) med i vinduet. På den måde bliver den ene egenskab mørkere end det lokale gennemsnit og den anden egenskab lysere. Faktoren  $C$  er en tolerance-konstant, der sørger for at den enkelte pixelværdi skal være en vis procentdel større end det lokale gennemsnit for at blive klassificeret som "forgrund". Forskellige resultater ved variation af vinduesstørrelse og tolerance-konstanten er vedlagt i bilag F.

Ud fra testene i bilag F kan vi konkludere at algoritmen ikke er specielt følsom overfor vinduesstørrelsen, så en størrelse på ca. 50-150 pixels giver gode og meget ens resultater. Sættes vinduesstørrelsen for lavt skaber det problemer i større områder med forgrund, hvor man risikerer at midten af området ikke har en højere værdi end gennemsnittet og dermed bliver vurderet til at være baggrund. Tolerance-konstanten vælges bedst ved 0 eller meget tæt på som beskrevet i bilag F.

Den adaptive thresholding udnytter billedets spatielle information, så vi er nødt til at bevare billedet i sin oprindelige form selvom dette inkluderer områder udenfor vores generelle ROI. Dette betyder at når det lokale gennemsnit beregnes i kantområderne vil vinduet indeholde pixels udenfor ROI. Inden de lokale gennemsnit beregnes sættes alle pixelværdier udenfor ROI derfor til det fundne gennemsnit indenfor ROI. På denne måde sikre vi at der ikke opstår problemer ved kantområder. Problemet kunne også være løst ved at udvide funktionen til beregning af det lokale gennemsnit til at tage ROI'en med så pixels udenfor ROI helt udelades, men da den anvendte funktion til beregning af det lokale gennemsnit er en indbygget Matlab-funktion, var det simple blot at sætte området udenfor ROI til gennemsnitsværdien. Som det ses i afsnit 15.6 giver denne metode heller ingen problemer.

Som beskrevet i afsnit 15.3.1 kan vi ikke være sikre på om kødet er det mørke og fedtet det lyse eller omvendt i et MAF eller PCA billede, og vi kan derfor heller ikke umiddelbart være sikre på hvilken egenskab den genererede feature-ROI udtrykker. Heldigvis kan vi udnytte vores kendskab til hvordan en spegepølse typisk ser ud, nemlig en relativt sammenhængende kødmasse med fedtet fordelt indimellem i mindre områder. Eulertallet for et binært billede, som vores feature-ROI, udtrykker antallet af separate objekter i billedet minus antallet af "huller" i objekterne. En feature-ROI der udtrykker kød vil derfor have et negativt eulernummer pga. alle "hullerne" fra fedtet imens en feature-ROI der udtrykker fedt vil have et positivt eulernummer nu da fedtet er forgrund og det kun er meget få steder at kød er helt omkranset af fedt. Hvis det var omvendt eller der var for mange steder hvor kødet i spegepølsen er omkranset af fedt, ville algoritmen fejle og kategorisere kød som fedt, og omvendt.

Det er altså muligt for os både at adskille fedt fra kød uden nogen aktivitet fra brugeren og vi kan styre om den fundne feature-ROI udtrykker fedt eller kød til den videre analyse.

### 15.3.5 Opstil generel model til kød/fedt-adskillelse

For at kunne opstille en generel model for adskillelse af kød og fedt, er det nødvendigt at have nogle træningsdata, hvor spegepølserne er delt op i de to klasser *kød* og *fedt*. Til formålet benyttes billederne fra bilag E, som yderligere manuelt er blevet checket igennem for at kontrollere en korrekt adskillelse. Kun de inderste 300x300 pixels er benyttet i model genereringerne fra hver sample. Når det kun er de inderste pixels der bliver brugt, er det dels for at holde hukommelsesforbruget på et minimum<sup>23</sup>, og for at trække et udsnit ud af pølsen hvor der ikke er for store variationskilder i form af kant eller stigende/faldende farveintensiteter. Efter at have opdelt spegepølsen i kød- og fedtregioner er det ønskeligt at overføre den viden man har om den enkelte spegepølse til andre spegepølser. En måde at opstille en sådan generel model på er at benytte regression. Vi benytter den genererede ROI som respons, hvor kød er markeret som 0 og fedt som 1. I afsnit 13.3 og 13.4 blev to metoder til regression beskrevet, den lineære og den logistiske. I dette afsnit opstilles og evalueres to modeller for kød/fedt-opdeling, genereret med lineær og en med logistisk regression.

I tabel 15.5 ses MSE- og korrelationsværdier for LOO<sup>24</sup> krydsvalideringen af den lineære regression af fedt/kød opdelingen for *sample 1-5*. MSE er den gennemsnitlige kvadrerede fejl pr. pixel. Det kan ses at den bedste model er fundet hvor  $K = 1$ . I tabel 15.7 ses MSE- og korrelationsværdierne for  $K = 1$  modellen, overført på alle fem samples. Generelt er det ikke tilstrækkeligt kun at overføre modellen til de data den er lavet ud fra. Her benyttes det for at etablere en reference til hvad der må være det bedst mulige resultat når der anvendes testdata.

Ud fra tallene ses det at det bedste fit for den lineære model fåes for *sample 1* og det dårligste for *sample 4*. På figur 15.18 ses den fundne lineære model overført på de to *samples 1* og *4*. For både det bedste fit ved *sample 1* og det dårligste ved *sample 4* er der stadig betydelige mængder kød der er blevet detekteret som fedt.

I tabel 15.6 ses MSE- og korrelationsværdier for LOO krydsvalideringen af den logistiske regression af fedt- /kød-opdelingen for *sample 1-5*. Det kan ses på MSE-værdierne at de generelt ligger meget lavere end for den lineære model med

<sup>23</sup>Se diskussionen om hukommelsesforbrug og Matlab i afsnit 5.2

<sup>24</sup>Her er LOO brugt da den er bedst anvendelig ved små datasæt, jvf. afsnit 13.1

et gennemsnitsværdi fald på ca 48%, og tilsvarende ligger korrelations-værdierne generelt meget højere med en gennemsnitsværdi stigning på ca. 66% i forhold til den lineære model. Også K-fold valideringen for den logistiske regression vælger  $K=1$  som den bedste model. Værdierne i tabel 15.8 viser modellen for  $K=1$  overført på alle fem samples. Det ses at også her er modellen overført på *sample 4* valgt som det dårligste fit, tilgængæld er fittet til *sample 2* valgt som det bedste her. Tilgængæld er der ikke de helt store udsving på parametrene, og det er først på anden decimal de skiller sig ud fra hinanden. Resultatet af den logistiske regression er vist i figur 15.19, hvor det ses at det gode fit nu er rigtig godt, og det dårlige fit ikke er så dårligt endda.

	<b>K=1</b>	<b>K=2</b>	<b>K=3</b>	<b>K=4</b>	<b>K=5</b>	<b>Gennemsnit</b>
<b>MSE</b>	0.2328	0.3932	0.3708	0.4163	0.3854	0.3597
<b>Corr.</b>	0.6066	0.4078	0.4437	0.3917	0.4290	0.4558

**Tabel 15.5:** MSE og korrelation af K-fold valideringen af lineær regression for *sample 1-5*.

	<b>K=1</b>	<b>K=2</b>	<b>K=3</b>	<b>K=4</b>	<b>K=5</b>	<b>Gennemsnit</b>
<b>MSE</b>	0.1240	0.2987	0.1323	0.1514	0.1494	0.1712
<b>Corr.</b>	0.7520	0.5271	0.7451	0.7127	0.7114	0.6897

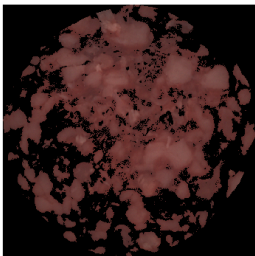
**Tabel 15.6:** MSE og korrelation af K-fold valideringen af logistisk regression for *sample 1-5*.

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Gennemsnit</b>
<b>MSE</b>	0.2328	0.2415	0.3525	0.3813	0.3469	0.3110
<b>Corr.</b>	0.6066	0.5982	0.4656	0.4318	0.4721	0.5149

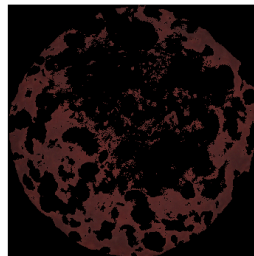
**Tabel 15.7:** MSE og korrelation af den lineære model for  $K=1$  overført på *sample 1-5*.

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Gennemsnit</b>
<b>MSE</b>	0.1240	0.1017	0.1388	0.1517	0.1485	0.1329
<b>Corr.</b>	0.7520	0.7930	0.7353	0.7122	0.7128	0.7411

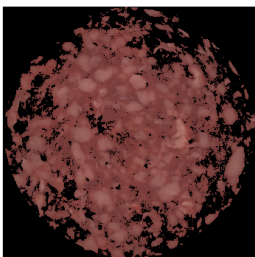
**Tabel 15.8:** MSE og korrelation af den logistiske model for  $K=1$  overført på *sample 1-5*.



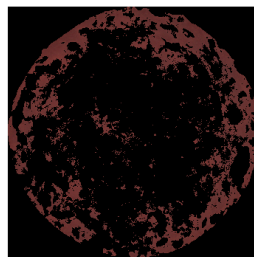
(a) *Sample 1 - Fett*



(b) *Sample 1 - Kød*

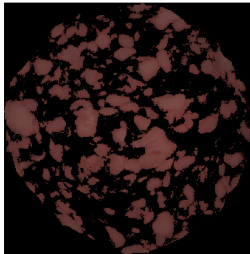
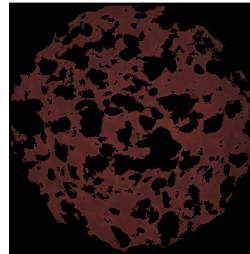
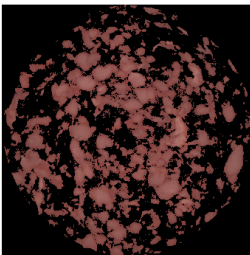
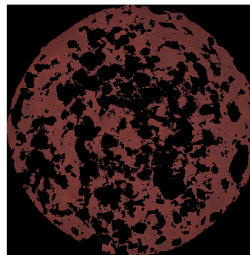


(c) *Sample 4 - Fett*



(d) *Sample 4 - Kød*

**Figur 15.18:** Den fundne lineære regressions model  $K=1$ , overført på de to pølser der scorede henholdsvis højeste og laveste MSE- og korrelationsværdier for modellen.

(a) *Sample 2* - Fett(b) *Sample 2* - Kød(c) *Sample 4* - Fett(d) *Sample 4* - Kød

**Figur 15.19:** Den fundne lineære regressions model  $K=1$ , overført på de to pølser der scorede henholdsvis højeste og laveste MSE- og korrelationsværdier for modellen.

## Evaluering af modellen

Selvom modellen for den logistiske regression giver et pænt resultat for de fem spegepølsebilleder vi har, og et resultat der er langt mere brugbart end det for den lineære model, er resultatet ikke perfekt. Der er stadig en del kød der bliver klassificeret forkert. Man kunne måske argumentere med at det var godt nok, set ud fra at det er farven af køddet vi er interesserede i at kigge på, og at der som man kan se ikke er noget fedt i kød-billedet, men det kød der er tilbage på fedt-billedet er helt klart også kød, og skal derfor være med i den endelige bestemmelse af kød-farven, ellers bliver disse værdier ikke robuste og sammenlignelige.

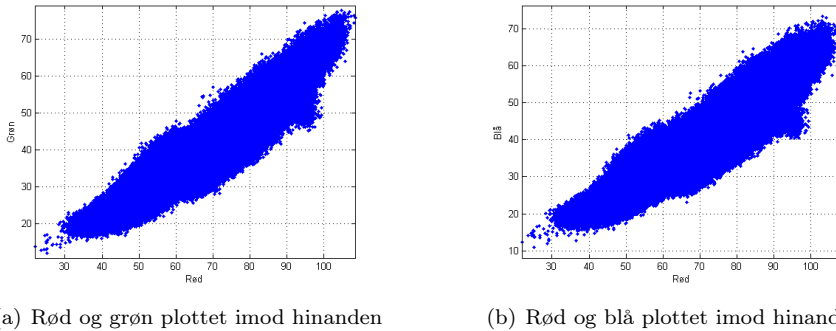
De spegepølser der er anvendt her blev af Danisco betegnet som ekstremterne inden for de spegepølser de arbejder med. Modellen kunne derfor vise sig at give brugbare resultater hvis de spegepølser den bliver brugt på er af samme, eller næsten samme, type. For at det virkelige skulle give brugbare resultater ville det være nødvendigt at have et ekstra sæt spegepølsebilleder til at teste den genererede model på. Det er generelt ikke brugbart at teste modellen på de samme spegepølsebilleder som den er genereret ud fra. Når det alligevel er gjort ovenfor, er det for at fastlægge et reference punkt for det der må betegnes som det mest optimale resultat. Da disse resultater ikke er overbevisende er det svært at tro på at det ville kunne give brugbare resultater selvom datasættet blev udvidet med separate test-billeder.

## 15.4 Farveskala for spegepølse

På nuværende tidspunkt bedømmes spegepølsers farver hos Danisco hovedsageligt subjektivt af eksperter på området, hvor spegepølsernes farve beskrives med tillægsord og ofte i sammenligning med andre spegepølser. De subjektive vurderinger indeholder begreber som pink, rød, mørkerød, overprocesseret, skinnende, mat og dækker også over farveforskellen mellem midten og kanten af pølsen. I dette afsnit undersøges muligheden for at udtrykke spegepølsernes farvespektrum objektivt og entydigt på tal-form. Tal-formen skal muliggøre en måde hvorpå man direkte kan sammenligne forskellige spegepølser, samt gøre farvevurderingen reproducerbar.

Farven i enhver pixel, eller et gennemsnit af mange pixels, kan udtrykkes som værdier i farverne rød, grøn og blå. Værdierne i disse tre farver vil være dækkende for dét som det menneskelige øje er i stand til at opfatte. Disse værdier kan dog være svære at forholde sig til fordi de ikke umiddelbart kan overføres til den





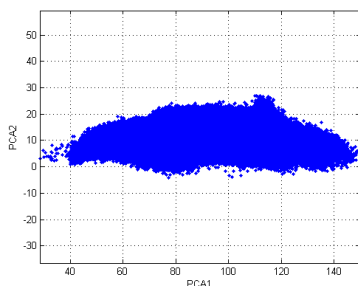
**Figur 15.20:** RGB plot af *sample 1-5*. Figurene viser en tæt lineær sammenhæng mellem intensiteterne.

subjektive vurderings form der i øjeblikket anvendes, som beskrevet ovenfor. På figur 15.20 er intensiteterne for rødt, grønt og blåt bånd plottet fra alle pixels i *sample 1-5*. Båndene der er valgt for rødt, grønt og blåt er 8, 5 og 4 svarende til 630nm, 505nm og 470nm, defineret som RGB-båndene i tabel A.1 for VideometerLabs 18 bånd. Vi har valgt at medtage alle pixels og ikke kun pixels defineret som kød ud fra den betragtning, at fedtfarven skal kunne beskrives på samme måde som kødfarven. Kødets farve er i analysesammenhæng den mest interessante at få vurderet, men fedtfarven vil også være relevant. Figur 15.20(a) viser rødt plottet overfor grønt og figur 15.20(b) viser rødt plottet over blå. Der ses en tydelig lineær sammenhæng mellem værdierne i begge grafer dog med en vis variation, så det tyder på at der er god mulighed for at udtrykke farven simplere end ved RGB-værdier.

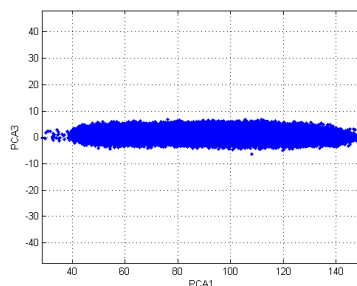
### 15.4.1 Farvekonverteringen

Ved at udføre en PCA analyse på dataene for rødt, grønt og blå svarer det til at vi roterer akserne i det 3-dimensionelle koordinatsystem med datasættet plottet, således at første akse/komponent efter analysen udtrykker den største farvevariation i billedet, anden akse/komponent udtrykker den næst-største farvevariation og tredje akse udtrykker den mindste farvevariation i billedet.

På figur 15.21 ses resultatet af PCA analysen af RGB-værdierne for de 5 samples. Det ses tydeligt at komponent 3 udtrykker en meget lille variation sammenlignet med komponent 1 og 2. De procentvise egenverdier for de tre komponenter er hhv 96.57, 3.09 og 0.34. Det betyder at 99,64% af datasættets variation (udtrykt ved *Sum of Squared Errors* SSE) er dækket ved de to første komponenter.



(a) PCA komponent 1 og 2



(b) PCA komponent 1 og 3

**Figur 15.21:** PCA plot af RGB værdier for *sample 1-5*. Udfra figurene ses det at langt det meste variation er indeholdt i komponent 1 og 2.

Konverteringen fra RGB-værdi til denne spegepølse-farveskala sker ganske enkelt ved at multiplicere det røde, grønne og blå bånd med egenvektorerne fra PCA analysen. Tabel 15.9 viser de fundne egenvektorer.

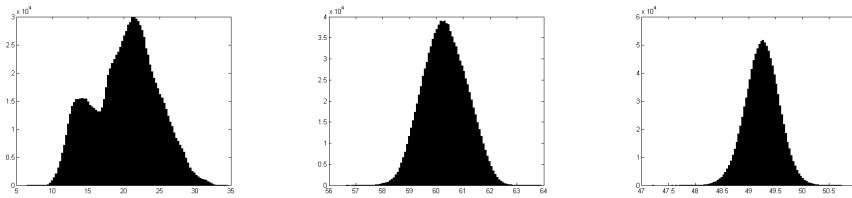
	Komponent 1	Komponent 2	Komponent 3
Rød	0.6923	0.7199	0.0493
Grøn	0.5278	-0.4586	-0.7149
Blå	0.4921	-0.5209	0.6975

**Tabel 15.9:** Egenvektorer for farveskalakonvertering.

For den fundne konvertering fra RGB til komponent 1,2 og 3 er minima og maksima bestemt for de tre komponenter og vist i tabel 15.10. De værdier som reelt vil være minimum og maksimum i vores datasæt vil være langt mindre ekstreme, men vi kunne sagtens komme ud for at finde værdier i nye prøver der ligger udenfor området for vores datasæt. Da det rent forståelsesmæssigt vil være mest naturligt kun at snakke om positive farvewærdier og at skalaerne for de tre komponenter er forskellige vælger vi at skalere værdierne efter konverteringen, så de ligger mellem 0 og 100.

Komponent	Minimum	Maksimum
1	0	437
2	-249	183
3	-182	190

**Tabel 15.10:** Minima og maksima for farveskalakonverteringen før skalering. Tallene er afrundet til nærmeste heltal.



(a) Histogram for PCA komponent 1 (b) Histogram for PCA komponent 2 (c) Histogram for PCA komponent 3

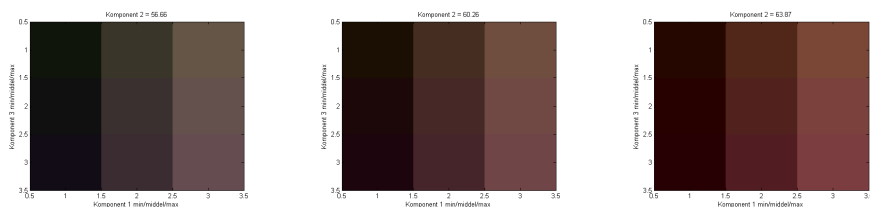
**Figur 15.22:** Histogrammer for PCA analysen af RGB værdierne for sample 1-5.

### 15.4.2 Simplificering af skalaen

Figur 15.22 viser histogrammerne for hver af de tre PCA komponenter. Komponent 3 er særligt interessant, da vi vil undersøge hvilken betydning det har for spegepølse farveskalaen at udelade denne værdi. Ved at udelade denne værdi vil de præsenterede tal naturligvis blive nemmere at overskue, men dette kan kun gøres hvis de resterende tal udtrykker pøsefarven tilstrækkeligt.

For komponent 3 er gennemsnitsværdien 48.9574, mindste værdi er fundet ved 46,5317 og største ved 50,4890. Histogrammet har, efter 3 iterationers midling med filteret fra formel (14.1) side 85, ét toppunkt fundet ved 49.0101, altså meget tæt på gennemsnittet. 90% af værdierne for komponent 3 ligger indenfor intervallet 48.3705-49.4897 for samtlige data fra *sample 1-5*. Det er derfor meget usandsynligt at værdier udenfor dette område i væsentlig grad bidrager til den farve som det menneskelige øje er i stand til at opfatte i et givent område af en spegepølse. Ved at se på den farveforskel der kan udtrykkes indenfor den fundne 90% variation af komponent 3 vil vi om muligt konkludere om den med rimelighed kan udelades.

På figur 15.23 vises betydningen af variation på den tredje komponent for 9 værdisæt af komponent 1 og 2 udvalgt som minimum, middel og maksimum i det spektre, og komponent 3 er sat til de fundne 90%-grænser og middelværdien. På de lyse af farverne (dvs. højre kolonne for hver graf, maximum for komponent 1) er der en antydning af forskel fra maximum til minimum af komponent 3, imens det er ret svært at se forskel ved de mørke af farverne på figur 15.22(b) og 15.22(c) (dvs. minimum for komponent 1). På figur 15.22(a) ses at lave værdier af komponent 3 giver en mørk grønlig farve og høje værdier er mere over imod lilla. På dette grundlag kan vi ikke endnu udelade komponent 3, da den trods alt giver synlig farveforskel i visse tilfælde.

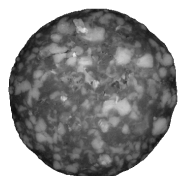


(a) Komponent 1 og 3 varieres, komponent 2 holdes konstant på **minimum**

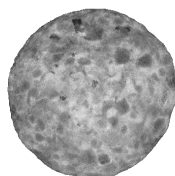
(b) Komponent 1 og 3 varieres, komponent 2 holdes konstant på **middel**

(c) Komponent 1 og 3 varieres, komponent 2 holdes konstant på **maksimum**

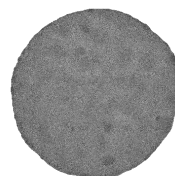
**Figur 15.23:** Farvevariation ved komponent 3.



(a) Komponent 1



(b) Komponent 2



(c) Komponent 3

**Figur 15.24:** Farveskalakomponenter for *sample 1*.

I stedet undersøges det hvordan de enkelte komponenter udtrykkes for en af spegepølserne i datasættet. På figur 15.24 er vist hver af de tre komponenter for *sample 1* fra datasættet. Det ses at komponent 1 udtrykker en meget stor del af spegepølsens struktur og den er også tydeligt mørkere rundt i kanten. Komponent 2 er ikke helt så tydelig mht. hvad den udtrykker men der ses dog alligevel en relativt tydelig struktur. Komponent 3 udtrykker derimod tæt på ingen information, kun i enkelte mindre områder ses en lille forskel i gennemsnitsintensiteten. Variationen af værdierne for komponent 3 ligger så tilfældigt, at det ikke vil resultere i et synligt tab af information, såfremt den altid sættes til den fundne gennemsnitsværdi.

### 15.4.3 Farvekort for spegepølsernes spektrum

Figur 15.25 viser hvordan de faktiske farveværdier fordeler sig på skalaen fra 0-100 for komponent 1 og 2 uden hensyntagen til komponent 3. Da spegepølsfarverne indenfor RGB-spektret naturligvis kun dækker et lille område er det også tilfældet på denne skala. På figur 15.26(a) er der zoomet ind på det relevante område hvor spegepølsfarverne ligger indenfor. På figur 15.26(b) er vist

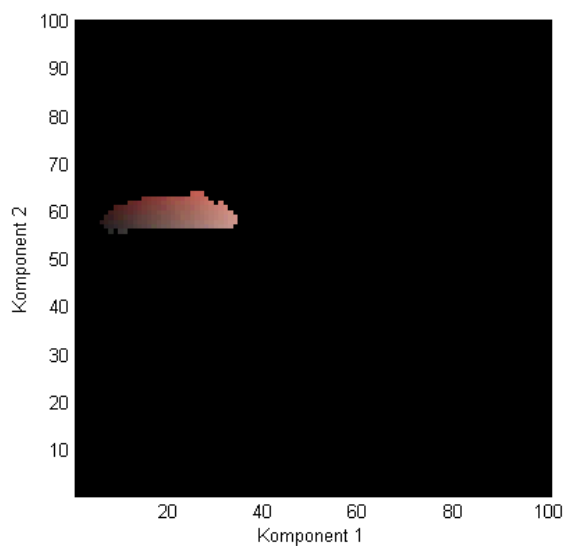
det tilsvarende område, blot hvor alle farverne er genereret med værdien for komponent 1 og 2 svarende til koordinaterne og den fundne middelværdi for komponent 3. Ud fra figuren ses det at komponent 1 udtrykker lysstyrken fra mørk til lys og komponent 2 udtrykker rødhed. Det samme kan vi også se på tabel 15.9 med egenvektorerne for farvekonverteringen, hvor komponent 1 vægter rød, grøn og blå meget ens, komponent 2 vægter rød positivt og de to andre negativt. For komponent 3 har rød en meget lav faktor, hvilket betyder at den er næsten uden betydning imens grøn og blå vægtes højt hhv. negativt og positivt, så komponenten som helhed udtrykker forholdet mellem grøn og blå. Som vist i forrige afsnit er der dog ikke en signifikant variation i denne komponent til at gøre den nødvendig til farvegengivelsen.

Et bedre indtryk af komponenternes betydning for farven ses på figur 15.27 hvor komponent 3 er valgt til en konstant på begge figurer. På figur 15.27(a) er komponent 2 holdt konstant og komponent 1 varieret, og modsat på figur 15.27(b). Ved variation af komponent 1 ses det at alle tre intensiteter stiger svarende til at komponenten udtrykker lysstyrken. Rød stiger en smule mere end grøn og blå men ikke voldsomt. Ved variation af komponent 2 ses det at intensiteten for rød stiger for stigende værdier af komponenten imens grøn og blå falder med stort set samme hældning (se tabel 15.9), svarende til at farven bliver mere rød. De komponenter der holdes konstant, er sat til en middelværdi indenfor det fundne spektrum og har kun betydning for offsettet af graferne.

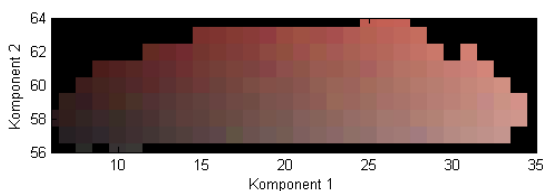
## 15.5 Segmentering af fedt

Som en del af spegepølseanalysen ønskes et tal for antallet af fedtklatter, gennemsnitsstørrelsen og størrelsesfordelingen. Den rensede egenskabs-ROI for fedt indeholder som det ses på figur 15.28 mange områder udgjort af flere enkelte fedtklatter. En ofte anvendt metode til segmentering af overlappende objekter er *Distance Mapping* (euklidisk, chessboard m.fl.) kombineret med *Watershed Segmentation* beskrevet i bl.a. [Vincent, 1991], [Breu et al., 1995], og [Chen et al., 2004], hvor det udnyttes at distancen fra baggrund til centrum af et objekt i forgrunden som regel vil være større end afstanden indtil midten af overlappet. Metoden virker klart bedst når de enkelte objekter er tilnærmelsesvis cirkulære. Et klassisk eksempel er ovale kaffebønner. Fedtet i en spegepølse kan godt have mange forskellige former, og overlappe en del, dog er en acceptabel antagelse at fedtklumperne oftest opfylder ovenstående.

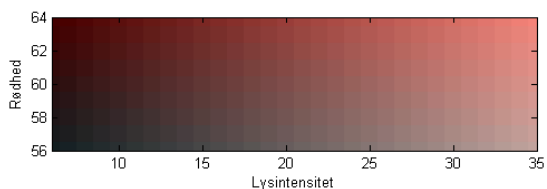
Watershed Segmentering baseret på euklidisk distance mapping (EDM) giver som det ses på figur 15.30(a) ikke noget godt resultat, da objekternes form ofte er meget ureel. Resultatet er at der for hver fedtklat eller ved to fedtklatters sam-



**Figur 15.25:** Totalt farvekort for spegepølser. Figuren viser de faktiske værdier af *sample 1-5* for komponent 1 af x-aksen og komponent 2 af y-aksen. Værdierne holder sig på et relativt lille område.

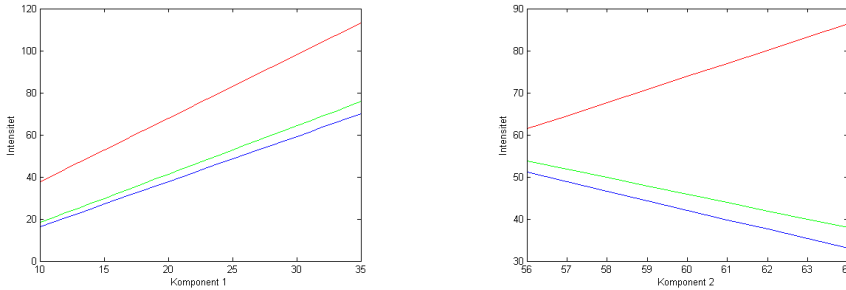


(a) Zoom af området med spegepølsefarver



(b) Et genereret farvekort

**Figur 15.26:** Figurene viser zoomede farvekort lavet ud fra rigtige og genererede data.



(a) RGB intensitet ved variation af komponent 1

(b) RGB intensitet ved variation af komponent 2

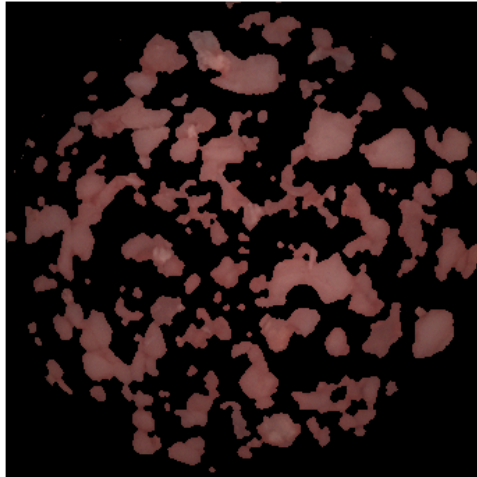
**Figur 15.27:** RGB intensiteter ved variation af komponent 1 og 2.

ling kan være flere lokale maksima, hvilket giver den overdrevne opdeling. Andre former for distance mapping såsom *Chessboard*, *cityblock*, og *quasi-euclidean* giver i nogen tilfælde bedre resultater, men stadig ikke overbevisende. Den valgte løsning til problemet er *grayscale reconstruction* og *h-dome transformation*, der gør flere nærtliggende lokale maksima til regionale maksima-plateauer.

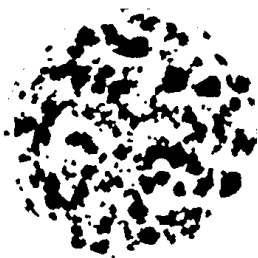
En højde  $h$  trækkes fra distance-billedet og *reconstruction* udføres på det resulterende billede. *Reconstruction* svarer til en gråtone-dilation i to dimensioner med den indbyggede logik, at en pixelværdi ikke kan blive højere end i original billedet og at algoritmen kører indtil der ikke sker flere ændringer i billedet. Algoritmen er ikke specielt følsom overfor størrelsen af  $h$ , men vælges en høj  $h$ -værdi vil flere områder blive betragtet som tilhørende samme fedtklat. Vi har fundet at en  $h$ -værdi svarende til radius af den mindste tilladte fedtstørrelse giver gode resultater. Der er ikke noget entydigt facit for fedtsegmenteringen og når den benyttes på et farvebillede af en spegepølse kan det også ses at nogle separate fedtklatter stadig hænger sammen, og andre der er blevet delt ikke er blevet delt lige hvor de rent faktisk sidder sammen. I forhold til optællingen og gennemsnitsstørrelsen er dette uden betydning og har kun lille betydning ift. størrelsesfordelingen. Metoden med *reconstruction* er illustreret på figur 15.31 taget fra [Vincent, 1991].

En rå optælling af antal fedtområder giver 86 for *sample 1* og et gennemsnitsareal<sup>25</sup> på  $7,9\text{mm}^2$ . Efter segmenteringen er der 112 fedtområder med et gennemsnitsareal på  $6,1\text{mm}^2$  og det endelige resultat kan ses på figur 15.30(b).

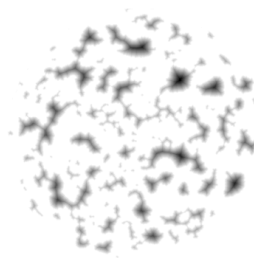
<sup>25</sup>Arealet i  $\text{mm}^2$  bestemmes ud fra den opgivne pixelstørrelse. Som standard er valgt pixelstørrelsen fra bilag A



**Figur 15.28:** Figuren viser fedtet fundet fra *sample 1* uden yderligere forsøg på at afskille de enkelte fedtklatter.



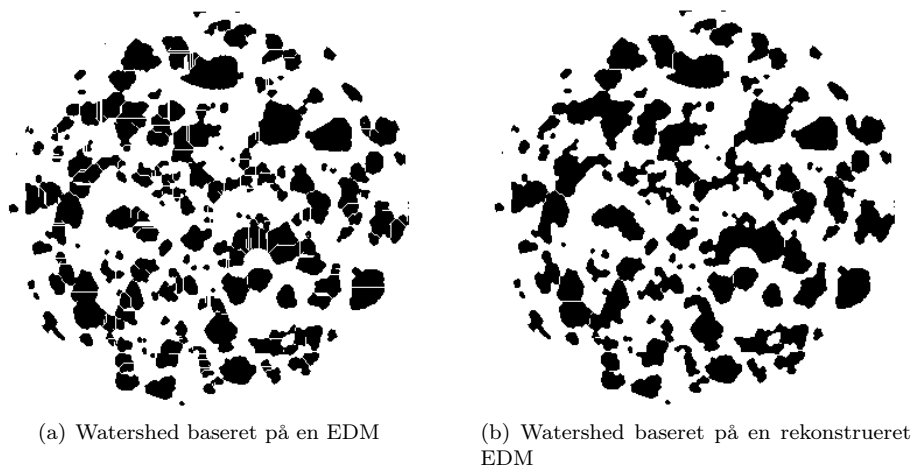
(a) ROI for fedt i *sample 1*



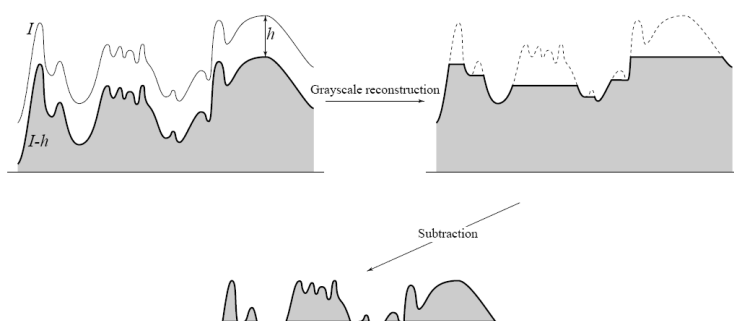
(b) EDM af fedt-ROI

**Figur 15.29:** Fedt-ROI og Euclidean Distance Map.





**Figur 15.30:** Watershed med og uden rekonstruktion.



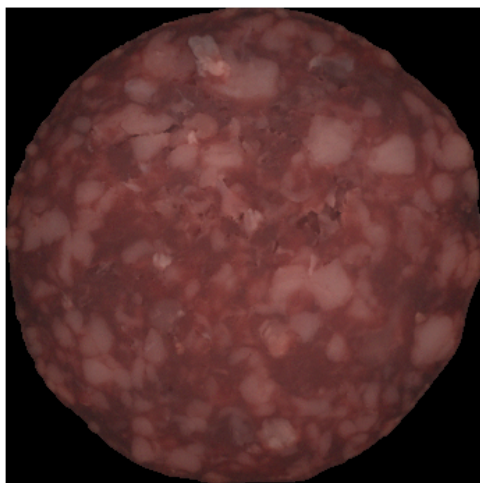
**Figur 15.31:** Figuren viser en principtegning af H-domes og reconstruction-metoden.

## 15.6 Præsentation af resultater

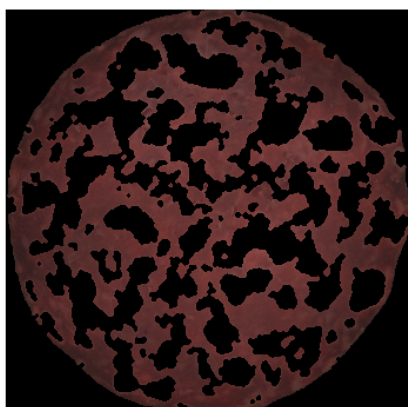
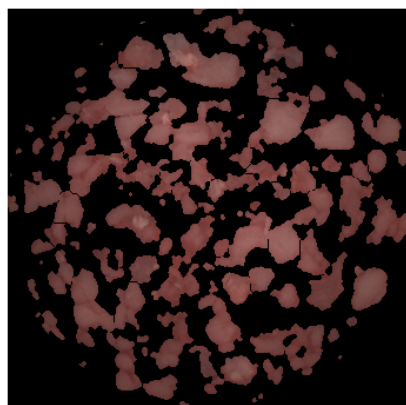
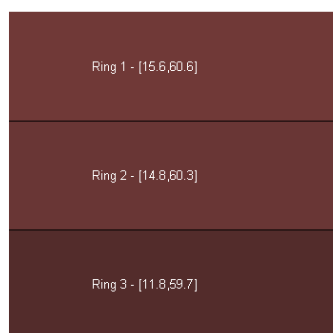
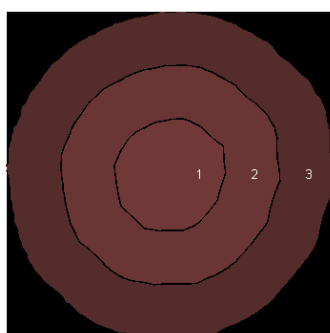
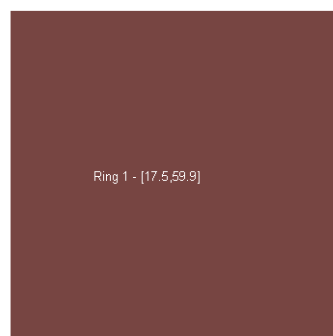
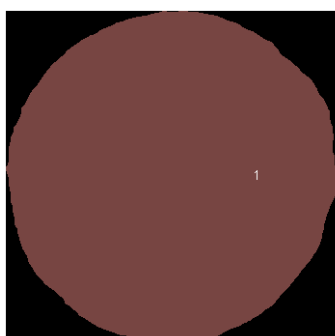
Dette afsnit beskriver resultaterne for *sample 1* i form af de billeder analysen giver. De talmæssige resultater for alle fem samples er samlet i to tabeller. Samtlige resultat-billeder fra analyserne af de fem samples er vedlagt i bilag E.

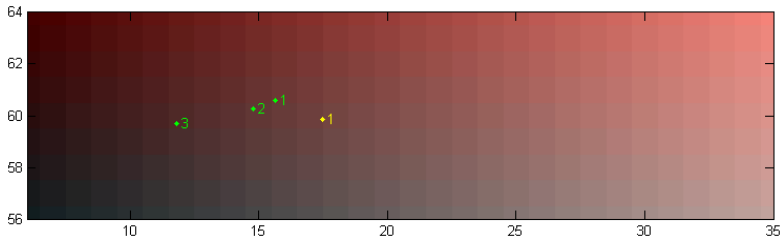
Ved analysen er benyttet analysemetoden MAF, og fedt/kød adskillelsen er lavet med adaptiv thresholding.

Figur 15.32 viser et farvebillede af spegepølserne. Det bruges dels som reference til de øvrige billeder og dels verificere det at ROI'en er blevet bestemt korrekt. De næste to billeder på figur 15.33 illustrerer den fundne opdeling i kød og fedt. Sammenholdt med figur 15.32 kan brugeren vurdere hvor godt de to klasser, kød og fedt, er blevet separeret. De fundne farver er vist som de ringe de er udregnet for på billedet til venstre og på billedet til højre vises værdierne sammen med farverne i rækkefølge. Tilsvarende viser figur 15.35 farve og værdi for fedtet, hvor det normalt kun er interessant at se på den samlede gennemsnitsværdi. Figur 15.36 angiver hvor de fundne farver for kød og fedtringe ligger på spegepølse-spektret. Til sidst vises den fundne fordeling af fedtområdernes størrelse i  $mm^2$  på figur 15.37.

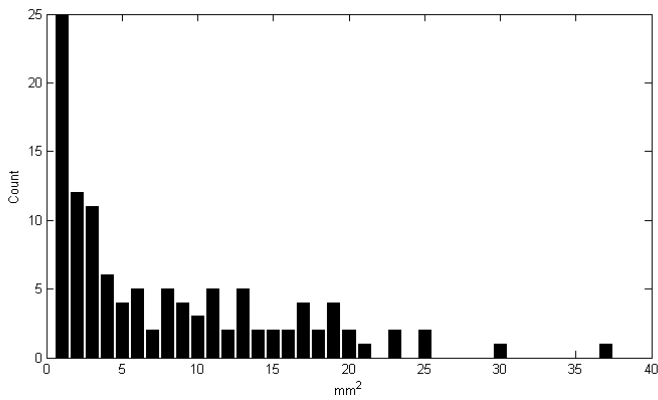


**Figur 15.32:** Farvebillede af *sample 1*.

(a) Kødets fundet i *sample 1*(b) Fedtet fundet i *sample 1***Figur 15.33:** Kød- og fedtopdelingen af *sample 1*.**Figur 15.34:** Gennemsnitsintensitet for kødet fundet i *sample 1*.**Figur 15.35:** Gennemsnitsintensitet for fedtet fundet i *sample 1*.



**Figur 15.36:** Farvekort med gennemsnitsintensiteter for *sample 1*. Kødringenes farver er angivet med grønt og fedtringenes farver med gult.



**Figur 15.37:** Størrelsesfordelingen af fedtområderne i *sample 1*.

I tabel 15.11 og 15.12 er samlet resultaterne for hhv. kødfarver og fedtfarve mm. Kødfarverne varierer tydeligvis mest på lysintensitet, som det også ses på farvekortene hvor værdierne er vist på spektret, og på forskellen i de forskellige ringe. Kun i mindre grad varierer rødheden. I tabellen for fedtet ser vi at især *sample 1* og *2* skiller sig ud fra de tre andre ved at være markant mørkere i farven, og vi kan også se at fedtområderne i de to er større og langt færre end for *sample 3-5*.

Sample	Kødfarve		
	Ring 1	Ring 2	Ring 3
1	15.6 / 60.6	14.8 / 60.3	11.8 / 59.7
2	15.4 / 60.7	13.9 / 60.5	12.7 / 59.8
3	20.2 / 60.9	19.5 / 60.9	17.5 / 60.9
4	20.6 / 61.0	20.0 / 60.9	17.6 / 60.9
5	21.0 / 61.3	20.0 / 61.3	17.9 / 61.2

**Tabel 15.11:** Farvевærdier for kødet i de fem spegepølsesamples. Det første tal for hver ring repræsenterer lysintensitet og det andet rødheden.

Sample	Fedt		
	Farve	Antal	Gns. str i mm <sup>2</sup>
1	17.5 / 59.9	114	7.1
2	15.8 / 59.6	93	9.4
3	22.8 / 60.0	217	4.3
4	22.7 / 60.1	188	5.1
5	23.6 / 60.6	238	3.9

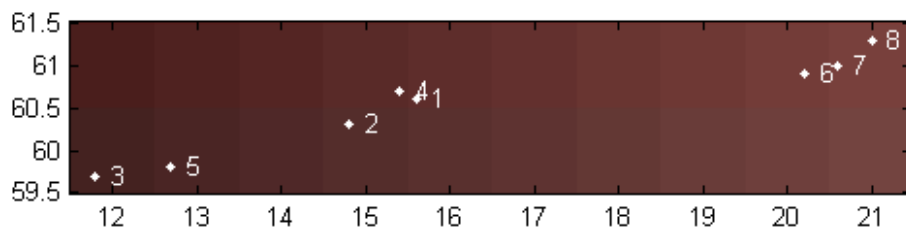
**Tabel 15.12:** Farvевærdier for fedtet i de fem spegepølsesamples. Det første tal for farven repræsenterer lysintensitet og det andet rødheden.

I tabel 15.13 ses sensorikdataene for de fem spegepølsesamples. Figur nummeret i tabellen henviser til numrene på figur 15.38. Det kan ses at de to figurnumre 2 og 4 der begge har fået betegnelsen *rød* ligger tæt på hinanden, punkterne 3 og 5 der begge er sat til *mørkerød* ligger også pænt samlet. For *sample 3* og *4* er det interessant at se at *sample 3* bliver vurderet ”mørkere og mere rød end *sample 4*”, hvor punkterne 6 og 7 rent faktisk ligger meget tæt. En lille nuance forskel er der dog og punkt 7 ligger da også en anelse lysere (til højre) end punkt 6, hvor rødheden tilgængelig er meget den samme. *Sample 3* er den der skiller sig mest ud fra resten i den ene ende af sensorik skalaen, og den ligger da også som et yderpunkt på skalaen.

Punkt 1 og 7 har fået de nært beslægtede betegnelser pink rød og lyserød pink. Punkterne ligger dog meget langt fra hinanden på skalaen. Når man kigger på

Sample	Ring nr.	Farve	Figur nr.
Sample 1	1	Pink rød. Mat.	1
	2	Rød	2
	3	Mørk rød	3
Sample 2	1	Rød. Rødvinsfarvet. Skinnende.	4
	2		
	3	Moden rød (overprocesseret)	5
Sample 3	1	Mørkere og mere rød end <i>sample 4</i>	6
	2		
	3		
Sample 4	1	Lyserød-Pink	7
	2		
	3		
Sample 5	1	Gullig - brunlig rød. Mat.	8
	2		
	3		

**Tabel 15.13:** Subjektive betegnelser for de 5 spegepølseresamples med tilhørende figurnummerreference til figur 15.38.

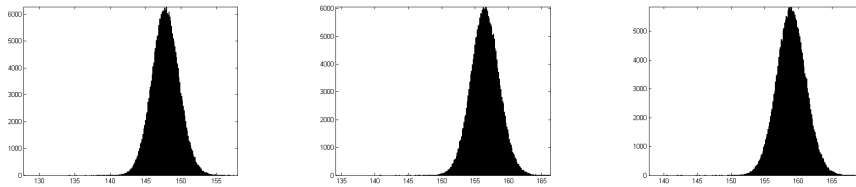


**Figur 15.38:** Farveskala med resultat værdier fra tabel 15.11 for alle spegepølser. På figuren er kun vist de figurnumre hvor der i tabel 15.13 er tilknyttet en sensorisk kommentar til.

resultatdataene for de to samples i bilag B, kan man måske ane at øjet nok har snydt ved bedømmelsen af *sample 1*, da det kan være svært at bedømme farven inderst i pølsen før fedtet er skilt fra. Efter adskillelsen er det lettere at se den tydelige nuanceforskel der er på de to pølser.

### 15.6.1 Skalering af farvebilleder

Fra VideometerLab'en er de målte intensiteter skaleret efter en sort-reference sat til 0 og en hvid-reference sat til 100. En prøve kan derfor godt have en højere intensitet på et givent bånd end hvid-referencen og dermed få en værdi højere



(a) Histogram for hvidt papir 630nm (rød)    (b) Histogram for hvidt papir 505nm (grøn)    (c) Histogram for hvidt papir 470nm (blå)

**Figur 15.39:** Histogrammer for RGB af hvidt papir.

end 100, eller den kan have en lavere intensitet end sort-referencen og dermed få værdier lavere end 0. I det synlige spektrum vil det ikke ofte være tilfældet med mindre der er tale om testbilleder, men i NIR-området vil f.eks. organiske prøver ofte have et højere respons end referencen, hvilket vi også kan konstatere i vores data.

I forhold til visningen af farvebilleder betyder det at der ikke er en oplagt og entydig maksimum intensitet, som de målte intensiteter kan skaleres efter. En prøve vil også i den virkelige verden se meget forskellig ud afhængig af lysforholdene, om det er indenfor eller udenfor, i et lyst eller mørkt rum, med eller uden direkte belysning eller om lyset kommer fra en almindelig glødepærer, diodelampe eller et lysstofrør som de gør hos Danisco, blot for at give nogle eksempler.

Det kan derfor ikke forventes at vi kan lave farverne sådan at de ligner 100% de farver som eksperterne ser på når de vurderer spegepølserne. I vores samlede datasæt har vi fundet maksimumværdier på cirka 110, 86 og 77 for hhv. rød, grøn og blå taget direkte fra de respektive bånd (8, 5 og 4). Som beskrevet i afsnit 5.1 sikrer VideometerLab'en reproducerbarhed og ensartet lyssætning, så selvom vi ikke kan udelukke maksimumværdier højere end de målte, så kan vi være relativt sikre på ikke at støde på værdier der er betydeligt højere. Derfor har vi valgt at skalere alle RGB-billeder efter en maksimumintensitet svarende til gennemsnitsintensiteten fundet på billedet af et blankt stykke papir. På figur 15.39 er vist histogrammerne og det ses at værdierne på hvert histogram ligger meget tæt med gennemsnit på 148, 157 og 159 for hhv. rød, grøn og blå og de har alle en standardafvigelse på cirka 2. Der er ikke konstateret intensiteter lavere end 0 hverken i test-billeder eller i nogle af prøverne, så nulpunktet ændres ikke.

De beregnede tal for farvевærdierne beskrevet tidligere sikre en helt entydig og sammenlignelig score og den ensartede skalering af billederne gør det muligt også at sammenligne visuelt om end kun billeder taget med VideometerLab.

Sample	Var. koef. MAF	Var. koef. PCA
1	0.4301	0.1217
2	0.6709	0.0951
3	2.5440	0.1002
4	1.3784	0.0964
5	16.4500	0.1074

**Tabel 15.14:** Variationskoefficienter for hhv. PCA og MAF.

## 15.7 Diskussion af resultater

Lineær- og logistisk regressionsanalyse kunne ikke give en robust generel model til at adskille kød og fedt på en anden spegepølse. I modelgenereringen har vi benyttet LOO K-fold krydsvalidering med de fem samples og både forsøgt med at klassificere alle pixels i billedet til enten kød eller fedt<sup>26</sup> som basis for modellen og vi har forsøgt med kun de pixels der mest tydeligt var enten kød eller fedt. Resultatet blev dog uanset metode, at opdelingen af kød og fedt blev meget ringe og ubrugelig i praksis.

PCA og MAF var i sig selv bedre at bruge, og med kombinationen af *Adaptive thresholding* og PCA eller MAF blev det muligt at adskille kød og fedt på et vilkårligt billede af en spegepølse. PCA og MAF minder meget om hinanden, hvor MAF dog kan fjerne en del støj i billedet, og vi ser i tabel 15.14 at MAF generelt giver langt større variationskoefficient ( $cv = \frac{\sigma}{\mu}$ ) på den komponent der benyttes til adskillelsen af kød og fedt og derfor er bedre at anvende. I praksis ser der dog ikke ud til at være den helt store forskel på den endelige opdeling med de to metoder. Intensitetsniveauudligning gav ikke nogen markant forbedring af resultatet med MAF, men den havde en positiv effekt for PCA-analysen.

I tidligere afsnit blev det nævnt at det ikke umiddelbart var til at se to toppe i histogrammet, hvorved en simpel thresholding kunne skille kød og fedt i spegepølsen. Efter adskillelsen af kød og fedt er det muligt at se på hvordan de to fordelinger rent faktisk ligger i forhold til hinanden.

I tabel 15.15 er vist de procentvise overlap mellem histogrammerne for kød og fedt fordelingerne. Det kan tydeligt ses at der ikke findes nogen bånd hvor det er muligt at adskille kød og fedt med en simpel thresholding. Heller ikke med intensitetsopretningen kan der findes et bånd der overbevisende viser at der kan vælges et bånd hvor fordelingerne ligger hver for sig. Det kan iøvrigt ses ud fra tabellen, at værdierne med intensitetsopretning i 4 af 18 tilfælde ligger højere end værdien uden, men at der generelt er sket en forbedring.

<sup>26</sup>Ved at thresholde et PCA billede med efterfølgende visuel verificering af resultatet.



Bånd	Procentvist overlap uden intensitetsopretning	Procentvist overlap med intensitetsopretning
1	25.77%	27.42%
2	17.79%	17.32%
3	18.25%	17.38%
4	18.03%	16.91%
5	16.64%	15.92%
6	15.83%	15.53%
7	15.84%	15.67%
8	23.28%	18.28%
9	26.35%	19.41%
10	28.21%	20.40%
11	29.84%	22.33%
12	30.30%	26.76%
13	30.68%	27.78%
14	30.38%	28.06%
15	27.45%	27.25%
16	26.28%	26.87%
17	24.62%	27.09%
18	22.45%	26.56%

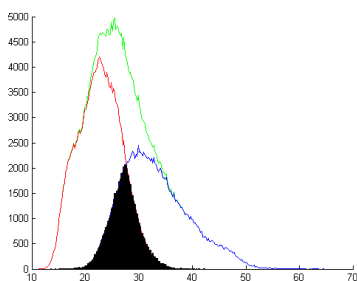
**Tabel 15.15:** Procentvist overlap for histogrammerne af de to populationer af kød og fedt i det oprindelige billede. Tallene er vist med og uden intensitetsopretningen af billedet.

Figur 15.40 viser histogrammerne for det mindste og største overlap af kød- og fedt-fordelingerne.

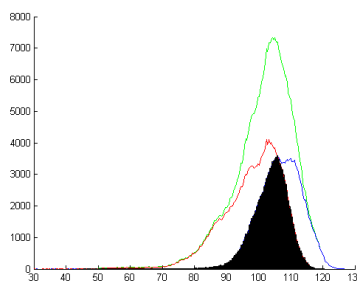
### 15.7.1 Robusthed

Analysen af spegepølser virker tilfredsstillende med standardindstillinger på alle spegepølser uden fedtkant. *Sample 2* har en fedtkant som skal fjernes før den giver et godt resultat.

Grunden til at det er nødvendigt at fjerne fedtkanten er, at det er en så markant egenskab i billedet at den ofte vil blive udtrykt i den mest betydende MAF komponent, og fedtet kanten består af er anderledes end fedtet i selve spegepølser og reflekterer derfor anderledes end resten af fedtet. For PCA analysen har det dog vist sig at være underordnet om fedtkanten er fjernet eller ej, før analysen. I bilag C og D viser figur C.2, C.3, D.2 og D.3 effekten på hhv. PCA- og MAF-analysen om spegepølser har en fedtkant eller ej. Ved PCA er der ikke den store forskel på komponent 1, der dog ikke ser ud til at have særlig stor kontrast på



(a) Bånd 6 - minimum overlap



(b) Bånd 13 - maximum overlap

**Figur 15.40:** De to grafer viser henholdsvis det mindste og det største overlap for de to populationer af kød og fedt i det oprindelige billede. På graferne er kød-populationen vist med rød, fedt med blå og det samlede histogram er vist med grønt. Den overlappende del er vist med sort. Histogrammerne er for spegepølse *sample 1* uden intensitetsopretning.

kød i forhold til fedt, men dette skyldes nok i høj grad skaleringen af billederne. Komponent 2 viser sig at være noget bedre, når kanten fjernes. Ved MAF er komponent 1 tydeligvis ikke god at bruge når spegepølsen har fedtkant, men komponent 2 ser ud til at være udmærket. Fjernes fedtkanten er det komponent 1 som bedst udtrykker forskellen mellem kød og fedt og tydeligvis bedre end når fedtkanten var med.

## 15.7.2 Hastighed

Som beskrevet i kapitel 11 har fokus primært været på funktionaliteten i udviklingsfasen, så hastighed har ikke været et punkt der var højt prioriteret.

Maskinerne anvendt til udvikling og test har været med 3+ GHz Intel processor og 1-2 GB ram, nogenlunde svarende til hvad der er maskinkravene for anvendelse af VideometerLab'en. På disse maskiner tager en komplet analyse fra det rå og ubehandlede billede til det endelige resultat et sted mellem 30 sekunder og 2 minutter med alle indstillinger valgt til standard. Cirka halvdelen af denne tid bruges på forbehandlingen af billedet, dvs. bestemmelse af ROI, beskæring af billedet og eventuel fjernelse af fedtkant.

Nogle ganske få af de parametre der kan stilles på vil få analysen til at tage længere tid, f.eks. hvis intensitets niveau udligning slås til. De resterende vil kun have meget lille eller ingen betydning for analysens hastighed. Forbehandlingen

kan desuden udføres separat og billedet gemmes, således at man sparer denne del hvis der skal køres flere analyser på samme billeder.



## Barrieresystem på kiks

---

Analysen af kiksene med barrierelag har til formål at opstille en model til at kunne bestemme barrieretykkelsen for en Mariekiks påført det Danisco udviklede vandbarrieresystem BS2000<sup>27</sup>. Kiksene er påført barrierelaget som beskrevet i afsnit 6.1. De benyttede datasæt består af 2 serier, kaldet *serie 1* og *serie 2*. De to serier er lavet ud fra to forskellige opsætninger af udstyret, og har derfor forskellige lagtykkelser. Nedenfor vil de to datasæt blive beskrevet yderligere.

Ud over disse to datasæt er der lavet en serie med nogle firkantede vaffelkiks af mærket "Bisquits & Care - Vaniljesmag vafler". Disse vafler er taget med for at se på mulighederne for at overføre en model til en anden type kiks. Vaflerne er sprayet ved samme kørsel som *Mariekiks serie 2*.

Derudover er der lavet en serie med nogle "fejlbehæftede" Mariekiks. Disse kiks er påført barriere samtidig med at der har ligget en mønt ovenpå enten en helt "tom" kiks, eller ovenpå en kiks der allerede er påført et lag barriere. Mønten er efterfølgende blevet fjernet, sådan at der er kommet en niveauforskel i barrierelaget.

Billederne for de fire datasæt for *Mariekiks serie 1*, *Mariekiks serie 2*, *vafler* og *Mariekiks med fejl* er vist i bilag H.

---

<sup>27</sup>Se afsnit 3.2.2

## 16.1 Barrierelagets målte tykkelser

Der blev i alt lavet tre gennemkørsler. Den første hvor målet var at have en barriere tykkelse pr. lag på  $130\mu\text{m}$ , denne blev benyttet til *Mariekiks serie 1*. Den anden kørsel blev indstillet til en barriere tykkelse på  $152\mu\text{m}$  pr. lag, denne blev benyttet til *Mariekiks serie 2* og *vafterne*. Den tredje kørsel blev indstillet til en barriere tykkelse på  $300\mu\text{m}$  pr. lag, denne blev benyttet til *Mariekiks med fejl*. Se iøvrigt afsnit 3.2.2 for en uddybende beskrivelse af forsøgs opstillingen og bilag G for en beskrivelse af forsøgets udførelse og indstilling af de forskellige parametre.

Gennemsnitstykkelserne af barrierelaget for de to serier *Mariekiks 1* og *2* er fundet ved at veje kiksene før og efter påføring af barrierelaget. Referencen er en *Mariekiks* påført  $400\mu\text{m}$  barrierelag der vejer  $0,92\text{g}$  mere dvs.  $434,78\frac{\mu\text{m}}{\text{g}}$  jvf. formel (4.1) side 21 og formel (16.1).

$$(16.1) \quad BL = v * 434,78 \frac{\mu\text{m}}{\text{g}}$$

Hvor  $BL$  er tykkelsen på barrierelaget målt i  $\mu\text{m}$  og  $v$  er vægten af den påførte barrierelagsmasse i gram. Den påførte barrierelagsmasse er målt ved at veje kiksene før og efter påførelsen. Hverken *vafterne* eller *Mariekiks med fejl* blev vejjet, og vi har derfor kun de estimerede barrierelagstykkelser at referere til.

Metoden med at veje barrierelagstykkelserne har desværre den begrænsning at man herved får et totalt mål for barrierelaget over hele kiksken, hvor der måske i virkeligheden er små fordybninger eller toppe, hvor barrierelaget kan samle sig eller vige væk fra. Derudover er der også en rund kant på *Mariekiksen*, hvor der ikke vil være lige så meget barrierelag som på resten af kiksken. Da kanten i sig selv er uvæsentlig for disse analyser, er der derfor udvalgt et område midt på kiksken med en størrelse på  $300\text{px} \times 300\text{px}$  til opstilling af modellerne. På denne måde undgås det at kanten har nogen indflydelse på resultaterne, men toppene og fordybningerne kan der ikke gøres noget ved. Derudover kan der nok med rimelighed argumenteres for at det netop er variationen og disse toppe og dale der gør analysen interessant.

De målte tykkelser kan ses i tabel 16.1 og 16.2 for *Mariekiks serie 1* og *2*, og tabel 16.3 og 16.4 for henholdsvis *vaffel* og *Mariekiks med fejl*. Det kan ses i begge tabeller for *Mariekiks serie 1* og *2*, at de målte gennemsnitstykkelser pr. lag ligger et stykke fra den estimerede på henholdsvis  $130\mu\text{m}$  for *serie 1* og  $152\mu\text{m}$  for *serie 2*.

Prøve nr	Vægt før	Vægt efter	BL vægt	Målt BL	Pr. lag
1	4,01g	4,34g	0,33g	143 $\mu$ m	143 $\mu$ m
2	3,96g	4,59g	0,63g	274 $\mu$ m	131 $\mu$ m
3	3,96g	4,91g	0,95g	413 $\mu$ m	139 $\mu$ m
4	3,97g	5,20g	1,23g	535 $\mu$ m	122 $\mu$ m
5	4,00g	5,53g	4,53g	665 $\mu$ m	130 $\mu$ m

**Tabel 16.1:** Barrierelag (BL) for Mariekiks model-datasæt for *serie 1*.

Prøve nr	Vægt før	Vægt efter	BL vægt	Målt BL	Pr. lag
1	4,17g	4,58g	0,41g	178 $\mu$ m	178 $\mu$ m
2	4,03g	4,79g	0,76g	330 $\mu$ m	152 $\mu$ m
3	4,10g	5,20g	1,10g	478 $\mu$ m	148 $\mu$ m
4	4,12g	5,56g	1,44g	626 $\mu$ m	148 $\mu$ m
5	4,13g	5,97g	1,84g	800 $\mu$ m	174 $\mu$ m

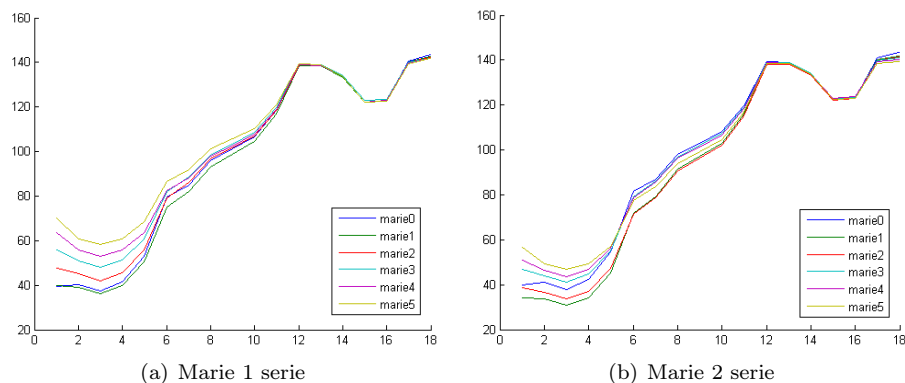
**Tabel 16.2:** Barrierelag (BL) for Mariekiks model-datasæt for *serie 2*.

Prøve nr	Målt BL	Pr. lag
1	152	152
2	304	152
3	456	152
4	608	152
5	760	152

**Tabel 16.3:** Barrierelag (BL) for vaffel kiks.

Prøve navn	Udenom mønt	Under mønt
ref_2x300	600 $\mu$ m	(ingen mønt)
coin_0_2x300	600 $\mu$ m	0 $\mu$ m
coin_1x300_2x300	600 $\mu$ m	300 $\mu$ m
coin_2x300_3x300	900 $\mu$ m	600 $\mu$ m

**Tabel 16.4:** Barrierelag (BL) for Mariekiks med pålagt mønt.



**Figur 16.1:** Gennemsnitsintensiteter af alle pixels henover alle 18 bånd for de to serier *Marie 1* og *2*.

## 16.2 Eksplorativ datasætanalyse

Dataene er i første omgang analyseret rent visuelt, for at se om de egenskaber vi ønsker at udtrykke, her barrierelagets tykkelse, er til at se mere eller mindre umiddelbart ud fra billederne.

På figur 16.1 ses gennemsnits intensiteter for de to Mariekiks serier. Der ses en tydelig sammenhæng mellem barrierelagets målte tykkelser, og gennemsnitsintensiteten op gennem båndene. Dog skiller et af billederne sig tydeligt ud, især for serie 2, hvor det ses at kurven for kontrolkiksen (*Marie 0*) uden barrierelag på har gennemsnitsværdier der ligger mellem *Marie 2* og *Marie 3* der har et påført barrierelag på henholdsvis  $330\mu\text{m}$  og  $478\mu\text{m}$ . Også kontrolkiksen (*Marie 0*) for serie 1 ligger ovenfor *Marie 1* der har et barrierelag på  $143\mu\text{m}$ . Da de øvrige kiks iøvrigt ligger pænt i rækkefølge, og da en kiks uden barriere ikke er interessant i forbindelse med at estimere kiksens barrierelagstykkelse, er disse fjernet fra alle model beregninger nedenfor.

Det kan derudover ses på billederne fra bilag H.5 til H.8 at kiksens struktur forsvinder hurtigere jo mere barrierelag der er påført og efterhånden som man går op igennem båndene.



## 16.3 Opstilling af model for barrierelagstykkelse

For at finde barrierelagets tykkelse på en kiks kan flere metoder benyttes, hver med deres fordele og ulemper.

Dette afsnit beskriver hvordan en generel model er blevet sat op med lineær regression. Det har været forsøgt at benytte logistisk regression som model, men det lykkedes ikke at få algoritmen til at konvergere, og dermed kom der ikke nogen tilnærmelsesvist brugbare resultater ud af det.

### 16.3.1 Lineær regressionsmodel

Som vi så i afsnit 16.2, viser der sig en tydelig lineær sammenhæng mellem gennemsnitsintensiteterne og den tilhørende barrierelagstykkelse for alle Mariekiks undtagen reference kiksen. Derfor er det nærliggende at tro at der må kunne findes en lineær regressions model der gælder for alle Mariekiks med barrierelag.

Modellen benytter den målte barrierelagstykkelse som respons for hver kiks. Regressionsmodellen er valgt på baggrund af en LOO krydsvalidering. Der er lavet en model ud fra både Marie *serie 1*, og *serie 2* kiksene, hvorefter modellerne er blevet overført på den modsatte serie. For begge serie gælder det at reference kiksen ikke er taget med i modelgenereringen, da den tydeligt afspejler en helt anden signatur end resten, som beskrevet ovenfor. Dog er reference kiksen medtaget når modellen bliver overført til kiksene, for at kunne se hvordan den opfører sig i forhold til modellen. Når modellen også er afprøvet på de kiks den er genereret ud fra er det for at have en nogenlunde sammenlignelig reference.

Resultatet af krydsvalideringen kan ses i tabel 16.5, 16.6 og 16.7 der viser henholdsvis MSE værdier, standardafvigelser og de beregnede gennemsnits tykkelser.

Standardafvigelserne i tabel 16.6 er rimeligt høje, men er tilgængelig nogenlunde stabile indenfor den enkelte serie. Tilgængelig er det interessant at se at for *serie 2* ligger standardafvigelserne for *model 1* lavere end for *model 2*. Dette er både modsat standardafvigelserne for *serie 1*, men også tendensen for MSE værdierne i tabel 16.5.

Ud fra MSE værdierne i tabel 16.5 ses det at de er meget lave for den serie de er genereret ud fra. Tilgængelig er de meget højere for den modsatte serie. Gennemsnitstykkelserne i tabel 16.7 viser det samme mønster, hvor de estimerede gennemsnitstykkelser for den serie modellen er genereret ud fra ligger sig meget

tæt op af de målte tykkelser, og for de fleste af kiksene er det først på 2. ciffer at de estimerede værdier skiller sig ud fra de målte. Det ses tilgængæld også tydeligt at de estimerede gennemsnitsværdier for den modsatte serie er meget langt fra de målte værdier. Dette kan også ses på figur 16.2 hvor de estimerede gennemsnitsværdier er plottet overfor de målte værdier. Det er helt tydeligt at se at der er en tydelig lineær sammenhæng mellem barrierelagstykkelserne, både for *serie 1* og *serie 2*, men at *serie 2* bliver estimeret under det målte med *model 1*, og omvendt bliver *serie 1* estimeret over det målte ved *model 2*. Det ses også at reference kiksene, vist med et 1-tal på figuren, skiller sig tydeligt ud fra resten og i halvdelen af tilfældene bliver estimeret til at være påført mere barrierelag end kiksene med det mindste lag, vist med et 2-tal på figuren.

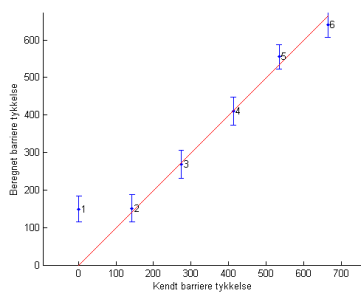
		Prøve Nr.					
		0	1	2	3	4	5
Model 1	Serie 1	23.13	1.39	1.37	1.40	1.42	1.67
	Serie 2	20.32	19.61	41.97	56.76	96.82	143.65
Model 2	Serie 1	121.51	49.84	72.21	120.05	210.88	205.87
	Serie 2	108.33	1.86	2.03	3.01	2.84	2.09

**Tabel 16.5:** MSE-værdier. *Serie 1* og *serie 2* billeder med to modeller genereret fra henholdsvis *serie 1* og *serie 2*. Alle værdier er i faktor  $10^3$ .

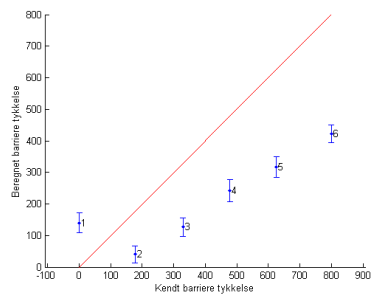
		Prøve Nr.					
		0	1	2	3	4	5
Model 1	Serie 1	34.46	36.42	36.64	37.29	31.85	32.61
	Serie 2	31.13	27.01	29.36	35.02	33.43	27.68
Model 2	Serie 1	49.66	53.91	55.15	57.26	49.72	51.36
	Serie 2	45.42	41.15	45.05	52.91	50.77	44.31

**Tabel 16.6:** Standard afvigelser. *Serie 1* og *serie 2* billeder med to modeller genereret fra henholdsvis *serie 1* og *serie 2*.

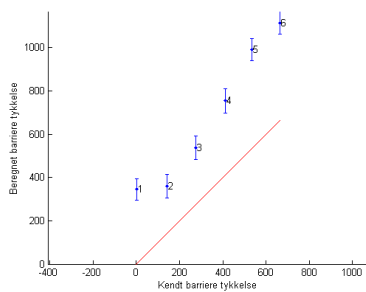
I tabel 16.8 ses gennemsnitsværdierne for standardafvigelserne for lagets jævnhed, samt standardafvigelserne mellem de estimerede og de beregnede tykkelser. Det ses tydeligt at model 1 er den der ud fra standardafvigelserne for de estimerede tykkelser er den bedste model. Dels passer modellen selvfølgelig godt på den serie den er genereret ud fra, men den passer også bedre på den modsatte serie end det er tilfældet med det omvendte.



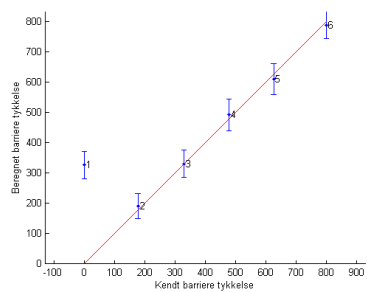
(a) Model 1 overført til serie 1



(b) Model 1 overført til serie 2



(c) Model 2 overført til serie 1



(d) Model 2 overført til serie 2

**Figur 16.2:** To modeller, en lavet på baggrund af serie 1, og en lavet på baggrund af serie 2, overført til henholdsvis serie 1 og serie 2. Plottene viser forholdet mellem de målte og de estimerede gennemsnitstykkelser og de tilhørende standardafvigelser.

			Prøve Nr.					
			0	1	2	3	4	5
Model 1	Målt		0	143	274	413	535	665
	Est.	Serie 1	149	151	269	410	555	640
		Serie 2	140	41	127	242	317	422
Model 2	Målt		0	178	330	478	626	800
	Est.	Serie 1	346	360	537	755	992	1116
		Serie 2	327	191	330	493	610	789

**Tabel 16.7:** Gennemsnitsværdier for estimerede og målte tykkelser. *Serie 1* og *serie 2* billeder med to modeller genereret fra henholdsvis *serie 1* og *serie 2*. Alle værdier er angivet i ( $\mu\text{m}$ ).

		Std. dev. på estimerede tykkelser	Std. dev. på lagets jævnhed
Model 1	Serie 1	16.64	34.88
	Serie 2	93.53	30.61
Model 2	Serie 1	108.19	52.84
	Serie 2	13.82	46.60

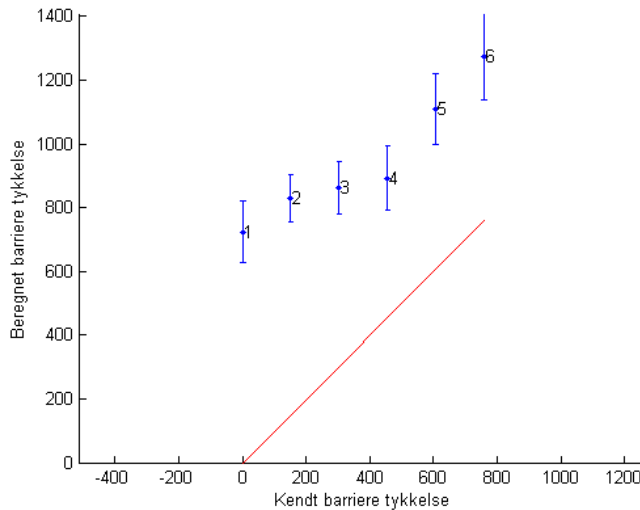
**Tabel 16.8:** Standardafvigelser for de to modeller. I beregningen af værdierne er *Marie 0* (Prøve nr. 0) ikke medtaget.

### 16.3.2 Kontrol af modellen

For at estimere barrierelaget på en anden vilkårlig kiks er den genererede model for *Marie kiks serie 2* anvendt på *vaffel* serien, der er lavet med samme gennemkørsel. I tabel 16.9 ses de estimerede tykkelser overfor de målte tykkelser, samt MSE-værdier og standardafvigelser. De estimerede gennemsnits tykkelser ligger meget over de målte og både standardafvigelser og MSE-værdierne er meget høje. På figur 16.3 ses der dog igen en nogenlunde lineær sammenhæng mellem de estimerede tykkelser og den målte påførte mængde barrierelag.

Prøve nr.	0	1	2	3	4	5
Målt tykkelse	0	152	304	456	608	760
Est. tykkelse	691	783	839	876	1083	1238
MSE ( $10^5$ )	5.19	4.90	3.21	2.03	2.69	2.88
Standardafvigelser	89.81	74.14	85.99	107.27	116.59	137.68

**Tabel 16.9:** Målte og estimerede værdier for *vaffler* med model genereret ud fra *Marie kiks serie 2*.



Figur 16.3: Model lavet fra *serie 2*, overført på *vaffel* kiks.

## 16.4 Resultater af modelgenereringen

Dette afsnit beskriver resultaterne af de to opstillede modeller for *Marieikiks serie 1* og *2*.

Først er modellerne anvendt på de samme kiks de er genereret ud fra, og efterfølgende dels på *vaffler* og *Marieikiks med fejl*. Når man anvender de data modellen er baseret på er det naturligvis ikke mærkeligt at resultaterne bliver gode, og de er derfor kun taget med for at vise den meget tydelige lineære sammenhæng. Det kan tydeligt ses at der er en meget tæt sammenhæng mellem den målte og den estimerede barrierelagstykkelse, både for den serie modellen er genereret ud fra, men også for den modsatte. Derudover kan det også ses at model 1 underestimerer barrierelagstykkelsen for *serie 2* og omvendt overestimerer model 2 barrierelagstykkelsen for *serie 1*. Det kan dog ses at MSE værdierne for model 1 anvendt på *serie 2* er næsten halvt så store som den omvendte, og at standardafvigelseerne for samme model er tilsvarende lavere, og endda lavere end model 2 anvendt på samme *serie 2*.

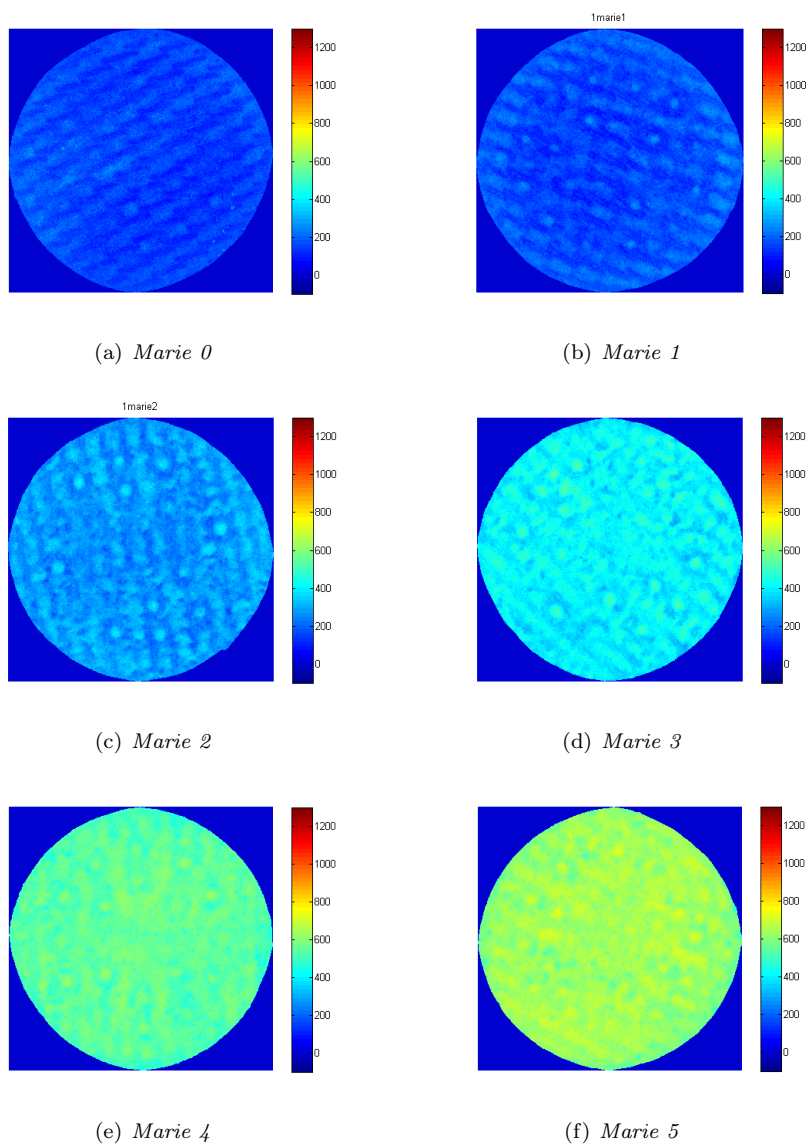
På figur 16.4 ses model 1 anvendt på *serie 1* kiksene. Figuren viser et højdekort over barrierelagstykkelsen over hele kiksene. Igen er denne figur kun taget med for at vise idealet for modelgenereringen.

På figur 16.5 ses modellen for *serie 1* anvendt på *serie 2*. Det ses at intensiteterne for hele kiksen stiger som funktion af barrierelagstykkelsen, men at gennemsnitsværdierne som man også kunne se i tabel 16.7 er estimeret for lavt. Tilgængæld kan man se at højdekort effekten træder tydeligt frem, og at man kan se de steder hvor der er fordybninger i kiksen, og der derfor har samlet sig mere barrierelag der end andre steder på kiksen. Jo tyndere barrierelaget er jo lavere lysintensiteter og omvendt jo tykkere det er jo højere lysintensitet.

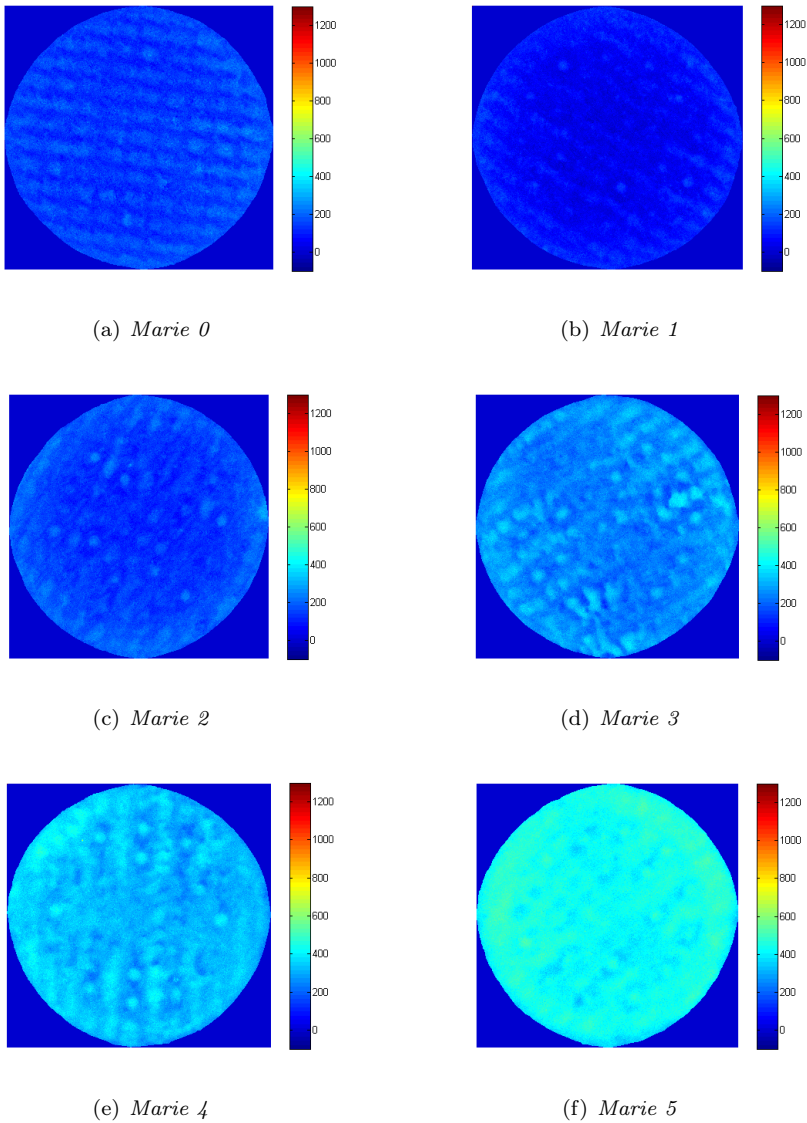
På figur 16.6 ses modellen for *serie 2* overført til vafferne. Igen kan der ses en stigning i lysintensiteterne jo mere barrierelag der er påført kiksen. På disse billeder ses der tilgængæld også en anden effekt, hvor nogle af kiksene har højere værdier i den ene ende af kiksen end den anden. Der er ikke umiddelbart noget under selve påføringen der kan have forårsaget denne effekt, og den må derfor være opstået ved selve billedoptagelsen. Vafferne var for store til at kunne ligge helt inde under VideometerLab kuglen ved optagelsen, og kuglen var derfor hævet op over vaffen. Der kan derfor være kommet "falsk" lys ind ved optagelsen af billederne. Derudover er vafferne lavet af et meget anderledes materiale i forhold til Mariekiksene, og kan have suget mere af barrierelaget nogen steder og ikke andre. Der kan dermed også være en forklaring på fænomenet der egentlig var tiltænkt forsøget fra starten af, da det netop var disse forskelligheder der var interessant at kigge på ved at kigge på forskellige typer kiks. Hvilken af disse to muligheder der eventuelt er den rigtige ville man kun kunne finde frem til ved at lave yderligere forsøg.

På figur 16.7 ses modellen for *serie 1* overført på de fire kiks fra serien af Mariekiks med pålagt mønt. Figur 16.7(a) viser referencen på  $600\mu\text{m}$  barrierelagstykkelse, og det er tydeligt på de andre kiks at se hvor denne lagtykkelse går igen. De to figurer 16.7(b) og 16.7(c) viser henholdsvis en barrierelags tykkelse på  $0\mu\text{m}$  og  $300\mu\text{m}$  der hvor mønten har ligget, begge med en lagtykkelse på  $600\mu\text{m}$  udenom. Det ses at der ikke er den helt store forskel at spore i intensiteterne. Denne problematik er den samme som med referencekiksene som beskrevet ovenfor.

Barrierelagets jævnhed kan beskrives med standardafvigelser. Vi så i tabel 16.6 at standardafvigelserne for de to Mariekiks serier var meget stabile, og uafhængig af barrierelags tykkelsen, hvilket var forventeligt, da der ved påførslen af barrierelaget ikke har været stillet på nogen af de i afsnit 6.1 side 27 omtalte parametre undervejs ved en serie gennemkørsel. Tilgængæld er afvigelserne relativt høje, hvilket formodentlig skyldes kiksens bølgede struktur. I tabel 16.8 så vi at den bedste model, dvs. modellen lavet ud fra *serie 1* giver en standardafvigelse mellem de målte og de estimerede barrierelagstykkelser på  $93,53\mu\text{m}$  og en standardafvigelse for samme model som udtryk for jævnt barrieren er påført på  $30,61\mu\text{m}$ .

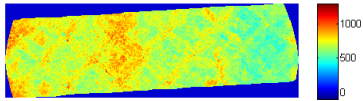
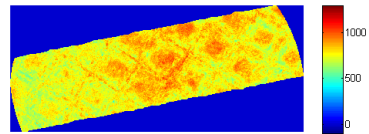
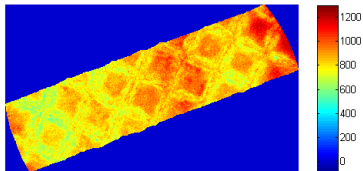
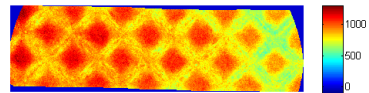
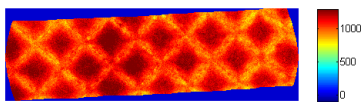
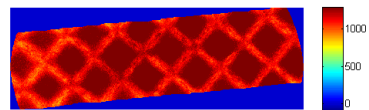


**Figur 16.4:** Højdekort for barrierelag af *Mariekiks serie 1*. Modellen er genereret ud fra *serie 1*.

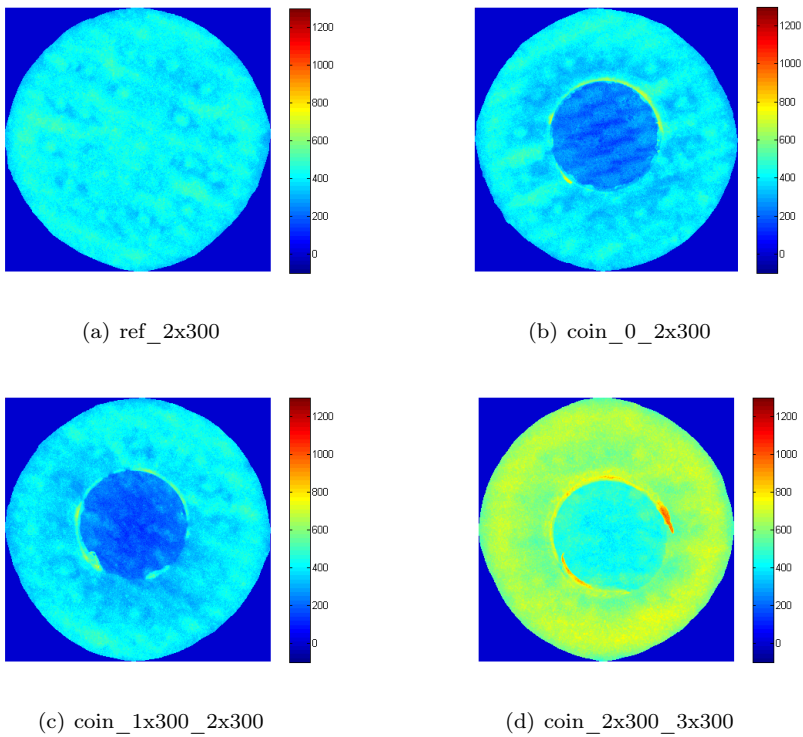


**Figur 16.5:** Højdekort for barrierelag af *Mariekiks serie 2*. Modellen er genereret ud fra *serie 1*.



(a) *Vaffel 0*(b) *Vaffel 1*(c) *Vaffel 2*(d) *Vaffel 3*(e) *Vaffel 4*(f) *Vaffel 5*

**Figur 16.6:** Højdekort for barrierelag af *Vaffel* kiks. Modellen er genereret ud fra *serie 2*.



**Figur 16.7:** Højdekort for barrierelag af *Mariekiks med fejl*. Modellen er genereret ud fra *serie 1*.

## 16.5 Diskussion af resultater

Der ses en tydeligt lineær sammenhæng mellem den påførte barrierelagstykkelse og det estimerede lag for gennemsnitsintensiteterne overfor mængden af påført barrierelag når man ser bort fra kontrolkiksen (*Marie 0*).

Men som det kan læses ud af ovenstående, er det ikke umiddelbart muligt at opstille en god model med lineær regression til at estimere barrierelagstykkelser for Mariekiks. Dog kunne der ses en tydelig lineær sammenhæng mellem barrierelagets tykkelse og det målte for begge serier, tilgængæld sås der ikke nogen sammenhæng mellem disse to linjer.

Ud fra denne konstatering er det nærliggende at tro at der kan have sneget sig en ukendt faktor ind ved selve påføringen af barrierelaget. Yderligere forsøg med et større antal kontrol-kiks lavet som samme serie, sådan at man har en serie til test og en til validering, vil muligvis kunne give bedre resultater.

En anden mulighed der undervejs har været afprøvet, er at lave en model ud fra et billede af ren barrierelag, og et for en "tom" kiks, eller med meget lidt barrierelag, og benytte disse to som de to klasser. Placeringen imellem de to klasser for en given sample angiver dermed tykkelsen. Dette forsøg gav ikke nogen resultater der var bedre end de ovenstående og er derfor ikke beskrevet nærmere her.

Det blev desuden forsøgt at opstille en model med logistisk regression, men dette gav heller ikke brugbare resultater, da model-genereringen ikke konvergerede mod et brugbart resultat.

Det kunne ses på standardafvigelserne at de var stabile inden for den samme serie, hvilket tyder på at der ved gennemkørslen af de enkelte serier ikke har været stillet på nogen af parametrene omtalt i afsnit 6.1 side 27. Tilgængæld var de relativt høje hvilket formodentlig skyldes kiksens struktur. Ved efterfølgende forsøg bør det overvejes at benytte en type kiks uden denne struktur, for at fjerne unødige fejlkilder under modelgenereringen. Dette kunne muligvis også opnås ved at benytte andre materialer, som f.eks. papir eller metal plader. Hvilken type materiale der er det bedste til formålet må komme an på en test.

Det må konstateres at sikkerheden og pålideligheden af datasættet desværre ikke er høj. Ud fra denne konstatering skal der nok opstilles et andet mere kontrolleret forsøg. En vigtig faktor i et sådant forsøg, vil være at have en serie kontrol kiks til validering af modellen, lavet med præcis de samme opstillingsparametre.

Temperaturen hvormed barrierelaget påføres kiks, spiller en stor rolle i forhold til hvilke intensiteter VideometerLab systemet opfatter. Dette skyldes at temperaturen af barrierelaget når det påføres kiks er medvirkende til i hvor høj grad barrieren trænger ind i kiks, og dermed ud af kameraets "synsfelt". En lille ændring i forsøgsopstillingen vil kunne påvirke varmen hvormed barrieren påføres kiks. Selvom den påførte mængde barrierelag er konstant, og vægten derfor er den samme, vil mængden på overfladen altså kunne variere i samme grad.

Rent billedteknisk skal det sikres at de høje kvaliteter ved VideometerLab systemet bliver bibeholdt, ved at der ikke skal være mulighed for falsk belysning. Dette kan sikres ved at benytte emner der er mindre end dem systemet rent fysisk kan gabe over.

# Vandindhold i brød

---

Analysen af brøddatasættet har til formål at opstille en model der gør det muligt at estimere vandindholdet i en skive brød ud fra et multispektralt billede taget med VideometerLab.

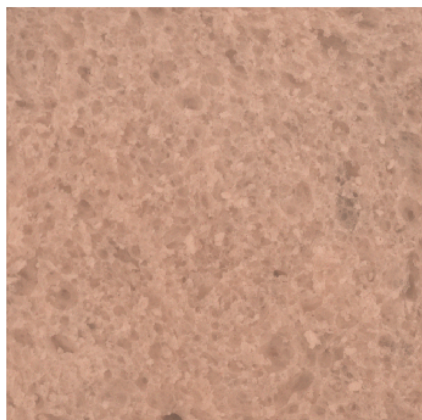
## 17.1 Datasætbeskrivelse

Brøddataene benyttet her er en del af datasættet fra et eksamensprojekt skrevet ved KVL [Jensen, 2006], der havde til formål at undersøge enzymer og emulgators effekt på brødets friskhed og vandindhold. Der blev bagt 6 forskellige typer brød og 2 af hver. Typerne er listet i tabel 17.1. Denne forskel i tilsætningsstoffer benyttes ikke direkte i udarbejdelsen af modellen til bestemmelse af vandindhold og giver yderligere variation i datasættet. Dog er det en variationskilde, der må anses som værende ganske rimelig at finde i typiske testdata, og en variationskilde modellen bør kunne tage højde for ved at estimere vandindholdet uden at skulle forklare hvilke faktorer der har været afgørende for vandindholdet.

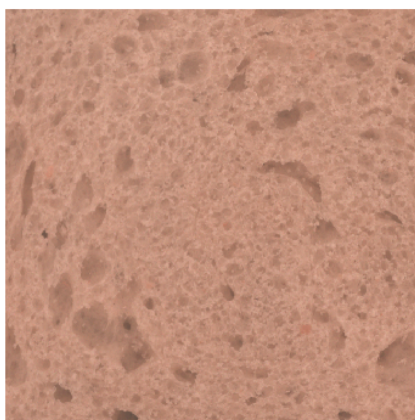
For hvert brød er der målt vandindhold med en metode kaldet HB43 af én skive, og med en metode kaldet LP16 af én skive og der er taget multispektrale billeder af begge de to skiver. Dette er foretaget på dag 1, dag 7 og dag 15. Yderligere

Enzym	Emulgator	Brød nr
Kontrol (ingen enzym)	uden	1 og 7
Kontrol (ingen enzym)	med	2 og 8
Enzym EDS132	uden	3 og 9
Enzym EDS132	med	4 og 10
Enzym EDS133	uden	5 og 11
Enzym EDS133	med	6 og 12

**Tabel 17.1:** Brøddataenes undergrupper.



(a) Brød 1, dag 1



(b) Brød 1, dag 15

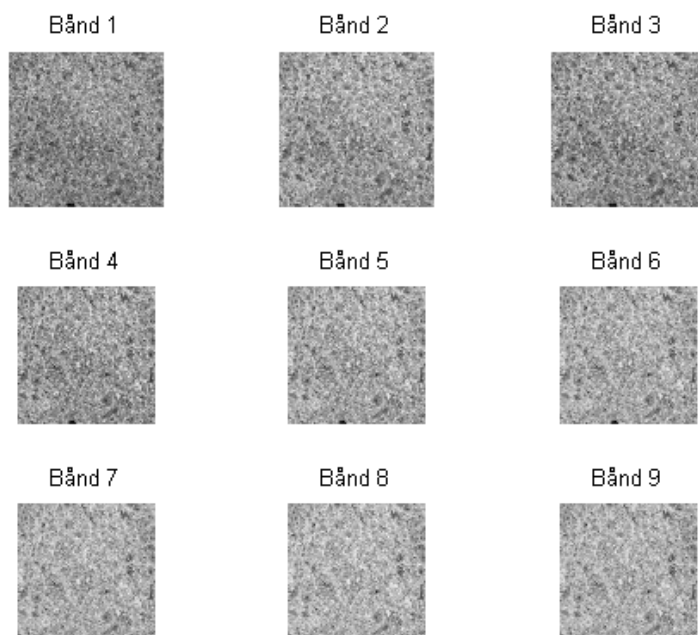
**Figur 17.1:** Brød 1, dag 1 og 15, forskellige skiver.

er der foretaget en Tekstur Profil Analyse (TPA) efter 3 og 9 timer og efter 1, 2, 4 og 7 dage, men disse data er ikke medtaget i analyserne her.

I alt har vi 72 billeder der to og to er relateret til de samme målinger af vandindhold. Den ene halvdel af billederne er anvendt som træningsdata til modelgenereringen og den anden halvdel er testdata til modellen.

Figur 17.1 viser billeder af skiver fra brød nr 1 ved hhv. 1 og 15 dage, hvor det ser ud til at udtørringen skaber skarpere overgange ved luftpuller i krummen, og at skiven efter 15 dage er en smule mørkere. De udregnede gennemsnitsværdier viser at de synlige spektra typisk ligger 0-6% højere ved dag 1 i forhold til dag 15 imens NIR-båndene ligger 0-4% højere ved dag 15 i forhold til dag 1.

Undersøges de enkelte bånd som vist på figur 17.2 og 17.3 ses det at de synlige spektra har et lavere respons (fremstår mørkere) end NIR-båndene. Krummestrukturen er tydelig i alle bånd.



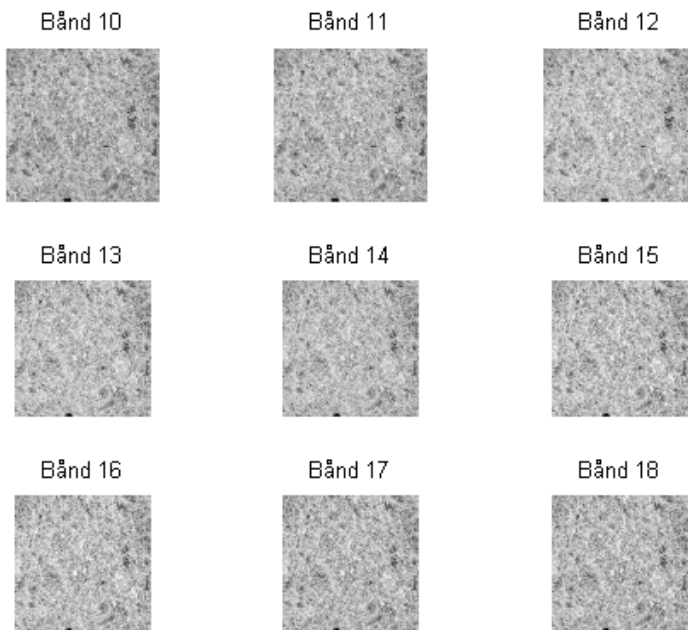
**Figur 17.2:** Bånd 1-9 for brød 1, dag 1.

<b>HB43</b>	<b>Dag 1</b>	<b>Dag 7</b>	<b>Dag 15</b>
Gennemsnit	43,9108	43,2842	38.3742
Standardafvigelse	0,3074	0,4082	0,7393

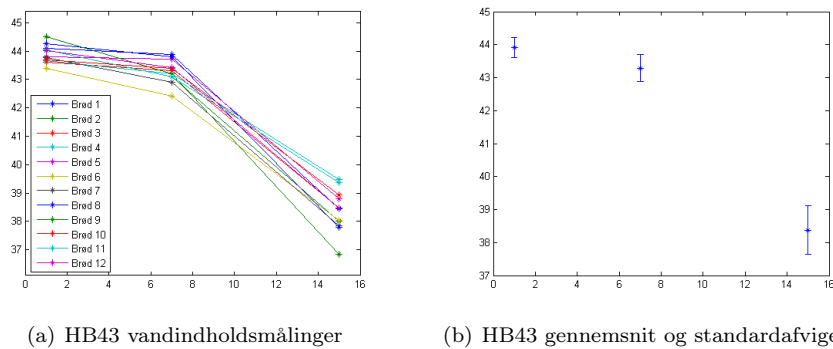
**Tabel 17.2:** Gennemsnit og standardafvigelse for HB43-målinger af brøddata.

<b>LP16</b>	<b>Dag 1</b>	<b>Dag 7</b>	<b>Dag 15</b>
Gennemsnit	43,7450	42.8708	38.2408
Standardafvigelse	0,3503	0,3910	1,9435

**Tabel 17.3:** Gennemsnit og standardafvigelse for LP16-målinger af brøddata.

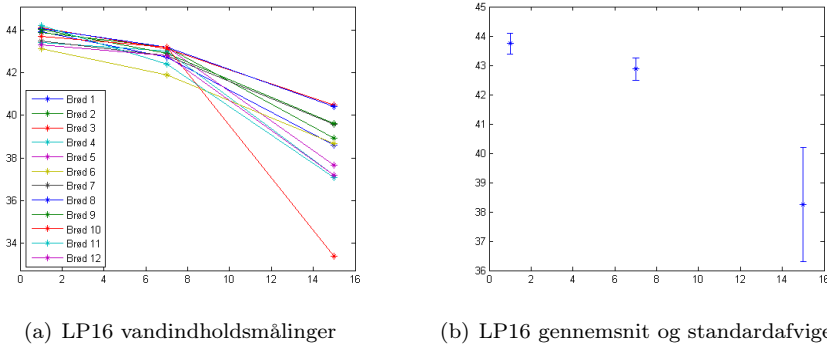


Figur 17.3: Bånd 10-18 for brød 1, dag 1.



Figur 17.4: HB43 vandindholdsmålinger, gennemsnit og standardafvigelse.





**Figur 17.5:** LP16 vandindholdsmålinger, gennemsnit og standardafvigelse.

De tilgængelige data giver ikke mulighed for at vurdere hvor stor variationen i vandindhold er for forskellige brød med samme tilsætning af enzymer og emulgator, dog kan vi se af tabel 17.2 og 17.3 at standardafvigelsen på målingerne med HB43 og LP16 ligger mellem 0,3 og 0,4 for dag 1 og 7, imens den er markant højere for dag 15 - for LP16-målingerne til dels pga. en tydelig outlier i datasættet. Af figur 17.4 og 17.5 ses det at prøver der har højt vandindhold ved målingen på dag 1 typisk også ligger i den høje ende efter hhv. 7 og 15 dage og vice versa.

## 17.2 Log-transformation af responset

Generelt vil et højere vandindhold i en prøve betyde en højere varians. Fordi datasættet kun har en enkelt måling for hver brødske er der ikke mulighed for at undersøge om det er tilfældet, dog ses det at standardafvigelsen for alle prøverne er markant større for målinger på dag 15 end på dag 7, og tilsvarende for dag 7 i forhold til dag 1. I [Clemmensen, 2006] er resultaterne til estimering af vandindhold i sandprøver forbedret ved at udføre en log-transformation af responset. På denne baggrund er der både lavet modeller til estimering af det målte vandindhold og log-transformationen af det målte vandindhold med begge målemetoder (HB43 og LP16). For at kunne sammenligne de log-transformerede med de øvrige værdier regnes disse tilbage til den normale skala før udregning af residualer, standardafvigelser m.m.

## 17.3 Featureudvælgelse

De fysiske målinger for vandindhold i centrum af brødskiverne er foretaget på en cirkel med 5 cm i diameter skåret ud fra midten. For at få de bedst mulige forudsætninger for en god model er der derfor udvalgt en ROI svarende til de inderste  $5 \times 5$  cm af brødet, hvilket var ca. 650 pixels på hver led. Vandindholdet i kanten af brødet er også målt, men da hele brødskiven ikke har kunne komme med i billederne har det ikke været muligt at udtrække den tilsvarende del af brødet, og vi vil derfor kun kigge på vandindhold målt i midten.

For hvert af de 18 bånd i hvert billede er der beregnet en række *features* på baggrund af histogrammet for det enkelte bånd. Disse features er: gennemsnit, standardafvigelse og den 1., 5., 10., 30., 50., 70., 90., 95., 99. procentil - i alt 11 features pr bånd. De samme features er fundet på den pixelvise log-transformation af hvert enkelt bånd og på alle unikke parvise differencer, multiplikationer og forhold mellem bånd<sup>28</sup>. Samlet giver dette 5445 features for hvert enkelt billede<sup>29</sup>. Dette featuresæt er valgt på baggrund af at et tilsvarende har vist sig velegnet til estimering af vandindhold i sandprøver [Clemmensen, 2006]. Yderligere førsteordens gråtone-statistikker (*skewness/skævhed*, *kurtosis/topstejlhed*, *energi*, *entropi*) er ikke medtaget i feature-sættet, da de vil være væsentlig mere påvirkelige af outliers i billedet.

## 17.4 Variation af modelparametre

Dette afsnit beskriver resultaterne for LARS-EN algoritmen med forskellige valg af det maksimale antal variable der skal medtages i modellen og  $\lambda$ .

### 17.4.1 Maksimum antal variable

Maksimum antal variable (*max-var*) er en stop-betingelse for hvor mange variable LARS-EN algoritmen skal medtage, og modelparametrene med koefficienter for alle feature gemmes for hver enkelt iteration på vej til dette antal. Efterfølgende udføres en krydsvalidering for at finde de parametre, der giver det bedste resultat uden at overfitte modellen til træningsdataene. Færre aktive<sup>30</sup> parametre vil også give en mere robust model. Samtidig er det interessant at se om et

<sup>28</sup>Med 18 bånd giver det  $(18 \cdot 17) / 2 = 153$  kombinationsmuligheder.

<sup>29</sup>198 features for de rå bånd, 198 for de log-transformerede og 1683 for hver af de tre parvise operationer.

<sup>30</sup>Parametre med værdi forskellig fra 0.

større antal variable vil gøre det muligt at give en mere præcis model. På denne baggrund er det valgt at beregne modeller for *max-var* sat til hhv. 10, 20, 40, 100, 200 og 400.

### 17.4.2 Lambda

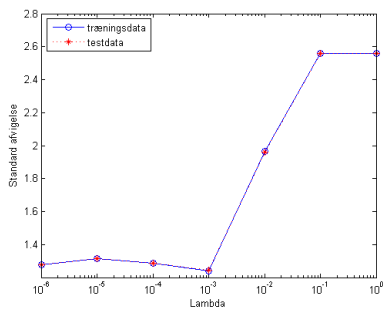
Fastsættelsen af en værdi for  $\lambda$  styrer i hvor høj grad de fundne parametre "straffes" baseret på deres størrelse. Lavere parametre vil typisk ikke fitte helt så godt til træningsdata som højere parametre vil, til gengæld vil estimeringen være mere robust. For hver af de ovenstående værdier for *max-var* er  $\lambda$  sat til  $10^{-6}$ ,  $10^{-5}$ ,  $10^{-4}$ ,  $10^{-3}$ ,  $10^{-2}$ ,  $10^{-1}$  og  $10^0$ .

### 17.4.3 Delresultater

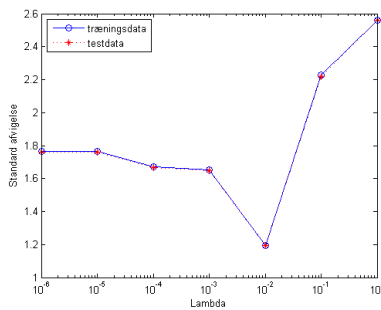
I dette afsnit er vist resultater for modeller til estimat af HB43 med den beskrevne variation af max antal variable og  $\lambda$ . De tilsvarende grafer for modeller til estimat af log(HB43), LP16 og log(LP16) er vist i bilag I.

Figur 17.6 viser for HB43 de fundne standardafvigelser for både trænings- og testdata for alle kombinationer af *max-var* og  $\lambda$ . For en given værdi af *max-var* ligger typisk meget tæt for  $\lambda$ -værdier mellem  $10^{-6}$  og  $10^{-3}$  (begge inkl.), hvor højere værdier af  $\lambda$  giver højere standardafvigelser og dermed en ringere model. Modellerne for hhv. 10 og 20 som *max-var* giver markant dårligere resultater end modellerne for 40-400, der for de laveste  $\lambda$ -værdier alle giver en standardafvigelse for træningsdata omkring 0,65 og for testdata 0,7.

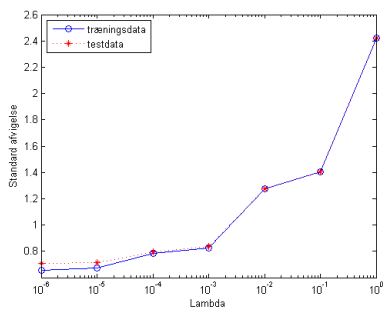
Disse resultater er opsummeret for hver af de fire modelsæt (HB43, log(HB43), LP16 og log(LP16)) på figur 17.7, 17.8, 17.9 og 17.10, der viser de bedste standardafvigelser og korrelationskoefficienter for hver *max-var*. For alle modellerne ses det at *max-var* på 40 eller derover i alle tilfælde giver bedre resultater end 10 og 20. Modellerne for det log-transformerede respons er stort set identiske med de direkte modeller, så der er intet vundet ved log-transformationen. Den bedste model til estimering af LP16 er markant dårligere end den bedste model til estimering af HB43, laveste standardafvigelser for testdata er hhv. 1,18 og 0,71.



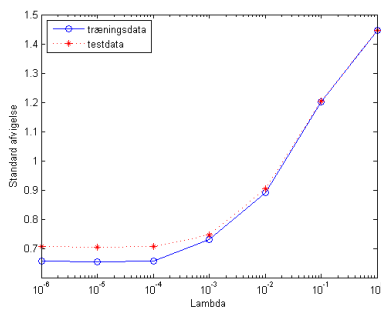
(a) Max 10 variable



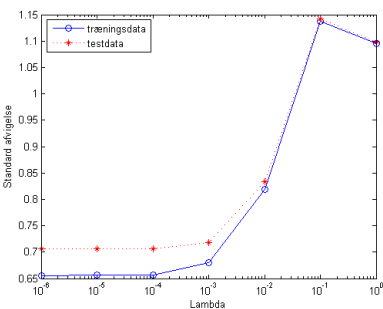
(b) Max 20 variable



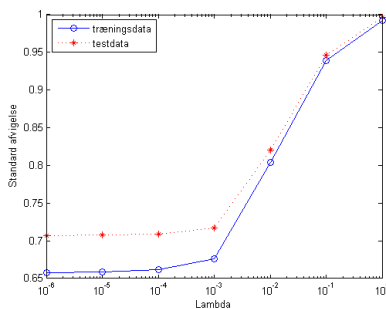
(c) Max 40 variable



(d) Max 100 variable

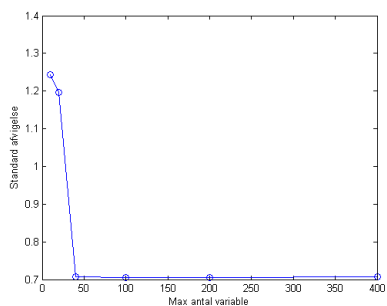


(e) Max 200 variable

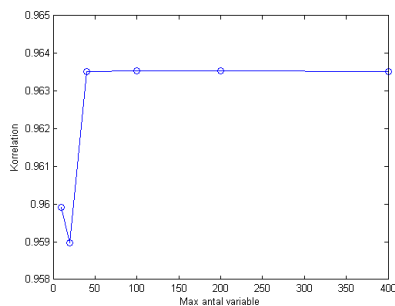


(f) Max 400 variable

**Figur 17.6:** Standardafvigelser for forskellige kombinationer af stop-kriteriet og  $\lambda$ .

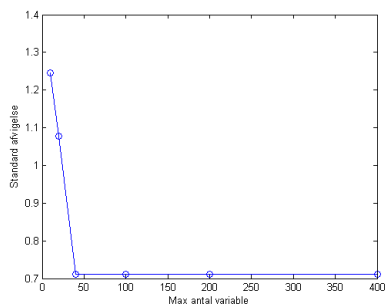


(a) Standardafvigelser for HB43

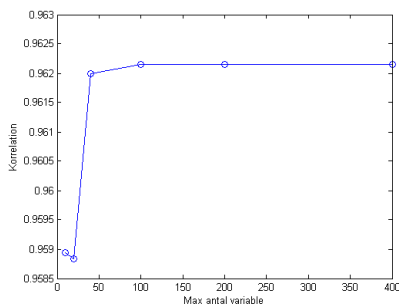


(b) Korrelationer HB43

**Figur 17.7:** Standardafvigelser og korrelationer for de bedste modeller til estimat af HB43 ved forskellige stop-kriterier.

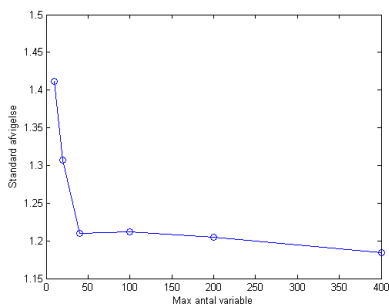


(a) Standardafvigelser log(HB43)

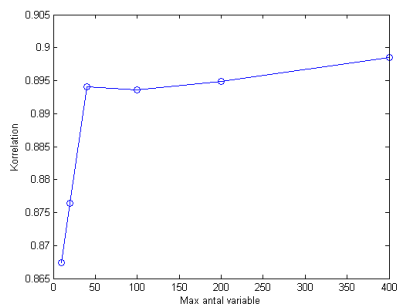


(b) Korrelationer for log(HB43)

**Figur 17.8:** Standardafvigelser og korrelationer for de bedste modeller til estimat af log(HB43) ved forskellige stop-kriterier.

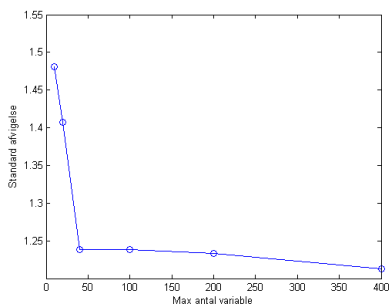


(a) Standardafvigelser for LP16

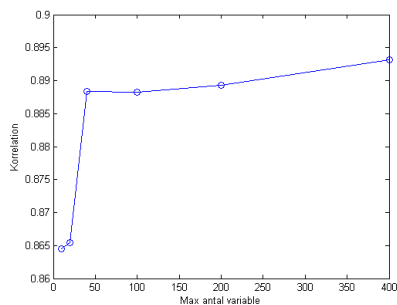


(b) Korrelationer for LP16

**Figur 17.9:** Standardafvigelser og korrelationer for de bedste modeller til estimat af LP16 ved forskellige stop-kriterier.



(a) Standardafvigelser for log(LP16)



(b) Korrelationer for log(LP16)

**Figur 17.10:** Standardafvigelser og korrelationer for de bedste modeller til estimat af log(LP16) ved forskellige stop-kriterier.

## 17.5 Brødets snitflade

Selve brødets struktur er en stor kilde til variationen af refleksion fra en skive. VideometerLab skaber et meget ensartet og diffust lys (se bilag A), men punkter på snitfladen af brødet vil stadig reflektere markant mere lys end et punkt i bunden af et luftlomme i skiven, selvom disse to områder begge fra centrum af skiven må formodes at have relativt ens vandindhold. I dette afsnit undersøges om det er muligt at få bedre modeller ved kun at medtage den del af billedet der ligger nærmest snitfladen.

### 17.5.1 Brødskivens negative krumning

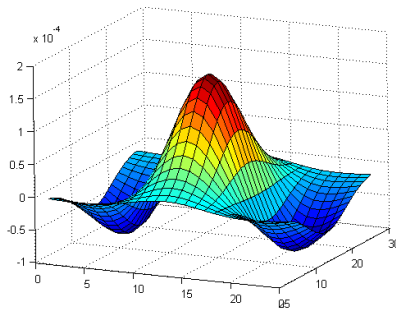
Som beskrevet vil punkter på eller tæt på snitfladen reflektere mere lys end punkter længere fra snitfladen lokalt set, men da der kan være større variationer over hele billedet er det ikke tilstrækkeligt blot at benytte intensiteten i et enkelt bånd og så udvælge fraktilen<sup>31</sup> med den højeste intensitet.

Ved at udtrykke intensitetens negative krumning skabes et billede hvor værdien i hver enkelt pixel er fastsat efter hvor stor en forskel der er på de omkringliggende pixels, dvs. pixels på en intensitetstop vil få høje værdier og pixels i en intensitetsdal vil få lave værdier.

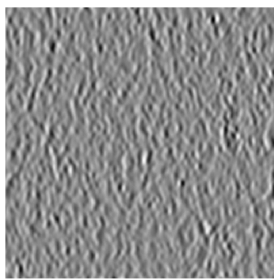
I afsnit 17.1 er det beskrevet at krummestrukturen er synlig i alle bånd, så ud fra den betragtning kan alle bånd bruges til at beregne krumningsbilledet. For at få flest mulige nuancer med vælges det bånd, der har den højeste standardafvigelse i histogrammet. Filteret vist på figur 17.11 forstærker værdier omkring centerpixelen og "straffer" værdier på hver side af centeret i filtermatricen er der 25x25. Derved får pixels med høj værdi og lavere værdier omkring sig, svarende til responset omkring snitfladen, de højeste værdier i resultatbilledet. Filteret anvendes både række- og søjlevis på det valgte bånd, hvilket for en brødskive kan give et resultat som vist på figur 17.12(a) og 17.12(b). Filteroperationen udføres både på rækker og søjler for ikke at fremhæve krummestruktur mere i den ene end den anden retning. De to resulterende billeder samles ved at tage den højeste værdi for hver pixel hvilket giver billedet vist på figur 17.12(c). Der er dog antydningen af en gitterstruktur i det samlede billede som må være kommet af filtreringen. Filteroperationen gør desuden at det er nødvendigt at fjerne en kant på cirka halvdelen af filterets bredde hele vejen rundt om billedet, så det endelige billede der kan beregnes features ud fra er 620x620 pixels.

---

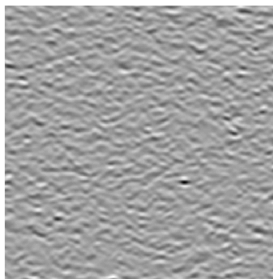
<sup>31</sup>Fraktil bruges som synonym for kvantil eller brøkdæl.



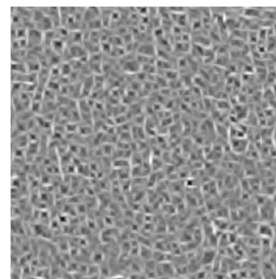
**Figur 17.11:** Filter til bestemmelse af en brødskives negative krumning.



(a) Søjlevis negativ krumning



(b) Rækkevis negativ krumning



(c) Række- og søjlevis negativ krumning af billedet kombineret så maksimum-værdien for hver pixel er valgt

**Figur 17.12:** Række- og søjlevis negativ krumning og et kombineret maksimum for en brødskive.



## 17.6 Resultater

Dette afsnit beskriver de fundne resultater med LARS-EN regression for features beregnet på hele billedet og et underafsnit beskriver resultaterne for features fundet ud fra fraktiler af billedet.

Der er opstillet modeller til estimering af vandindhold målt med HB43 og LP16, og det logtransformerede respons af de to. Modellerne er vurderet på baggrund af den standardafvigelse estimererne havde på testdataene og standardafvigelsen for træningsdataene kommenteres. Estimerer og residualer vurderes samlet og sammenholdes med prøvens alder.

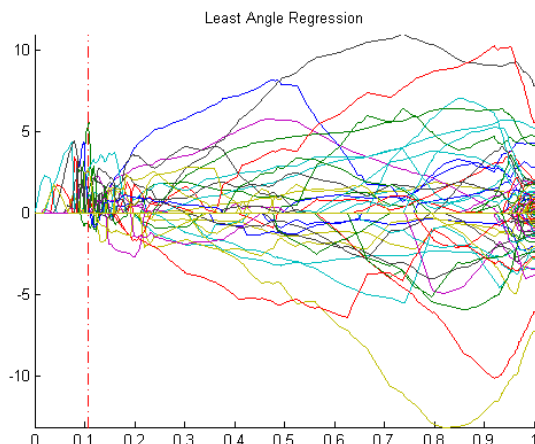
Model	Std. træning	Std. test	$\lambda$	Max. var.	Antal var.
HB43	0,65	0,71	$10^{-5}$	100	9
log(HB43)	0,66	0,71	$10^{-4}$	100	13
LP16	0,99	1,18	$10^{-2}$	400	23
log(LP16)	1,00	1,21	$10^{-2}$	400	23

**Tabel 17.4:** Modelresultater for estimering af vandindhold ud fra features af hele billedet.

Ud af tabel 17.4 ses det, at HB43 vandindhold kan estimeres med en standardafvigelse på 0,71 og LP16 ligger noget højere med en standardafvigelse på 1,21. Den variation som datasættet viser, især ved dag 15, burde ikke nødvendigvis gøre modellen dårligere, for såfremt de udregnede features tilsammen kan korreleres godt til vandindholdet, må det gælde for hvert enkelt billede. Antallet af dage prøven har ligget eller oplysningerne om tilsætning af enzymer eller emulgator er ikke medtaget i modellen, fordi de hver især er betydende faktorer der ville tage fokus væk fra selve billedanalysen og de udregnede features.

Figur 17.13 illustrerer parameterudviklingen for kørslen af LARS-EN regressionen der gav den bedste model for HB43 op til det ønskede maksimale antal variable. Den stiplede lodrette linje angiver hvor krydsvalideringen har fundet den optimale model, der også angivet i tabel 17.4 har medtaget væsentligt færre aktive parametre end det maksimale antal.

Log-transformationen har kun minimal betydning for resultatet, og tallene tyder ikke på at det bliver bedre. Det er interessant at bemærke at antallet af aktive variable i den foreslåede model er markant færre end det maksimale antal variable. Det er krydsvalideringen der ligger en kraftig begrænsning på antallet af variable, for at modellen ikke er overfittet til træningsdataene. Selvom det høje max-antal variable ikke direkte bruges har det dog alligevel den betydning at modelgenereringen har flere iterationer til at få optimeret parametrene for



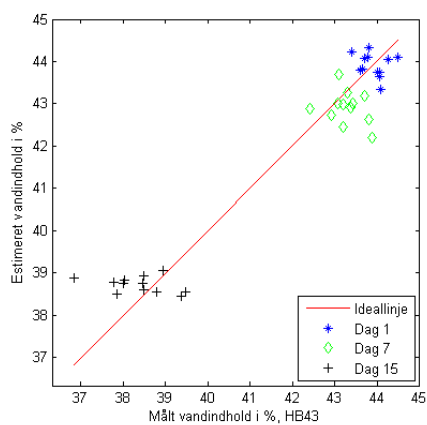
**Figur 17.13:** Parameterudviklingen for LARS-EN regressionen for HB43 med  $\lambda = 10^{-5}$  og  $max - var = 100$ .

de mest betydende features, så standardafvigelsen er markant lavere end ved modellerne med lavere max-antal variable.

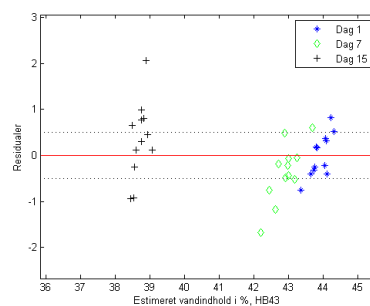
Estimaterne for HB43 vandindhold er stort set identiske for modellen baseret på de rå tal og for log-transformationen, som det ses på figur 17.14 og 17.15. Variationen på det målte og estimerede vandindhold for dag 1 og 7 ser ud til at være i samme størrelsesorden, imens der for dag 15 er en tydelig større variation på det målte vandindhold i forhold til det estimerede.

Sammenlignet med HB43 ligger estimaterne for LP16 og  $\log(LP16)$  på figur 17.16 og 17.17 overvejende på samme måde, dog ser der ud til at være flere outliers, særligt ved dag 15. Estimaterne for det log-transformerede respons er ligesom med HB43 stort set identisk.

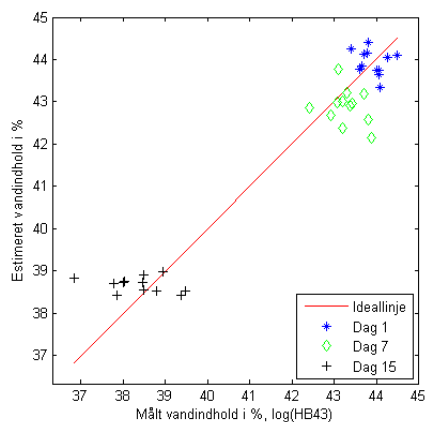
Tabel 17.5 samler resultaterne for de fire modeller og viser standardafvigelserne fordelt på de forskellige dage. Det ses tydeligt at log-transformationen ikke har nogen større betydning, at standardafvigelsen er tydeligt stigende med stigende alder og at HB43 estimeres markant bedre end LP16. Figur 17.18 illustrerer dette grafisk.



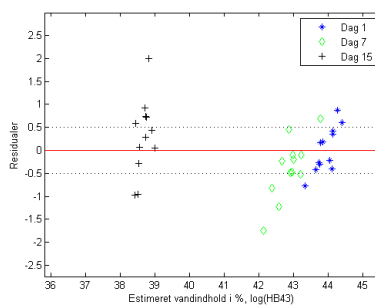
(a) HB43 estimerer af den bedste model



(b) HB43 residualer af den bedste model

**Figur 17.14:** HB43 estimerer og residualer af den bedste model.

(a) log(HB43) estimerer af den bedste model

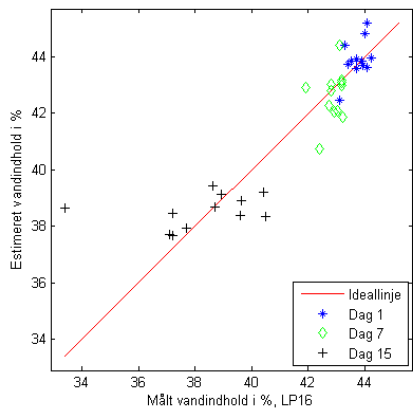


(b) log(HB43) residualer af den bedste model

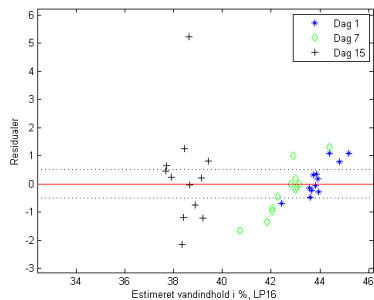
**Figur 17.15:** log(HB43) estimerer og residualer af den bedste model.

Model	Dag 1	Dag 7	Dag 15
HB43	0,4586	0,6361	0,8284
log(HB43)	0,4856	0,6614	0,8184
LP16	0,5842	0,8748	1,8411
log(LP16)	0,5103	0,8006	1,8378

**Tabel 17.5:** Standardafvigelserne for modellerne fordelt på brødets alder.

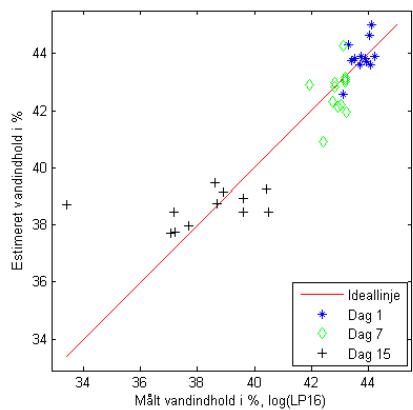


(a) LP16 estimer af den bedste model

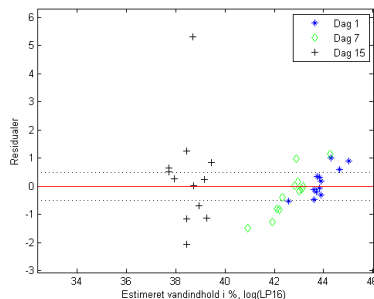


(b) LP16 residualer af den bedste model

**Figur 17.16:** LP16 estimer og residualer af den bedste model.

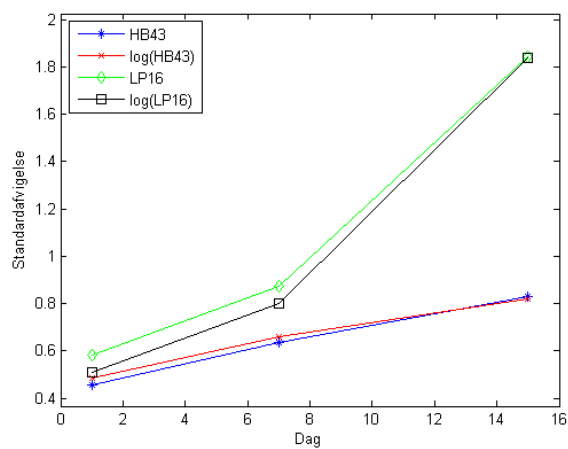


(a) log(LP16) estimer af den bedste model



(b) log(LP16) residualer af den bedste model

**Figur 17.17:** log(LP16) estimer og residualer af den bedste model.



**Figur 17.18:** Standardafvigelser for de forskellige modeller sorteret efter antal dage.

### 17.6.1 Fundne features

Dette afsnit beskriver hvilke features de fundne modeller er kommet frem til som værende bedst til at estimere vandindholdet.

Til estimering af vandindhold målt med HB43 metoden er det overvejende fraktiler af forhold mellem bånd der er fundet som de bedste features samt enkelte standardafvigelser og fraktiler for differencer mellem bånd. For LP16 er det hovedsageligt standardafvigelser og fraktiler for differencer mellem bånd og kun 3 fraktiler af forhold. Fraktilerne er typisk fra ekstremerne, dvs. 1, 5, 10, 90, 95 eller 99%.

Featureindex	Beskrivelse
635	70% fraktil af differencen mellem bånd 2 og 7
1082	5% fraktil af differencen mellem bånd 5 og 6
1083	10% fraktil af differencen mellem bånd 5 og 6
1938	Standardafvigelsen af differencen mellem bånd 13 og 16
1949	Standardafvigelsen af differencen mellem bånd 13 og 17
3782	90% fraktil af forholdet mellem bånd 1 og 3
3788	5% fraktil af forholdet mellem bånd 1 og 4
4003	95% fraktil af forholdet mellem bånd 2 og 7
4475	90% fraktil af forholdet mellem bånd 5 og 8
4476	95% fraktil af forholdet mellem bånd 5 og 8
4608	95% fraktil af forholdet mellem bånd 6 og 8
4744	1% fraktil af forholdet mellem bånd 7 og 10
5211	70% fraktil af forholdet mellem bånd 11 og 18
5228	1% fraktil af forholdet mellem bånd 12 og 14
5401	99% fraktil af forholdet mellem bånd 15 og 17
5409	70% fraktil af forholdet mellem bånd 15 og 18

**Tabel 17.6:** De udvalgte features til estimat af vandindhold (HB43).

Det er interessant at bemærke at der udelukkende er benyttet features der udtrykker noget om forskellen eller forholdet imellem to bånd. I disse modeller er der ikke benyttet features beregnet på baggrund af de rå bånd, de log-transformerede bånd eller multiplikationer mellem bånd.

### 17.6.2 Fraktil resultater

Vi udregner de samme features på baggrund af fraktiler af det fundne krumningsbillede, således at den første fraktil svarer til den 1% af billedet der er nærmest snitfladen, fraktil 2 er 2% osv. Fra 1-22% er fraktilerne beregnet med

Featureindex	Beskrivelse
399	Standardafvigelsen af differencen mellem bånd 1 og 2
400	5% fraktil af differencen mellem bånd 1 og 2
585	Standardafvigelsen af differencen mellem bånd 2 og 3
596	Standardafvigelsen af differencen mellem bånd 2 og 4
655	30% fraktil af differencen mellem bånd 2 og 9
756	70% fraktil af differencen mellem bånd 2 og 18
770	99% fraktil af differencen mellem bånd 3 og 4
1355	Standardafvigelsen af differencen mellem bånd 7 og 8
1366	Standardafvigelsen af differencen mellem bånd 7 og 9
1839	Standardafvigelsen af differencen mellem bånd 11 og 18
1846	90% fraktil af differencen mellem bånd 11 og 18
1938	Standardafvigelsen af differencen mellem bånd 13 og 16
1971	Standardafvigelsen af differencen mellem bånd 14 og 15
2013	99% fraktil af differencen mellem bånd 14 og 18
2015	Standardafvigelsen af differencen mellem bånd 15 og 16
2027	1% fraktil af differencen mellem bånd 15 og 17
2029	10% fraktil af differencen mellem bånd 15 og 17
2048	Standardafvigelsen af differencen mellem bånd 16 og 17
3798	1% fraktil af forholdet mellem bånd 1 og 5
4140	5% fraktil af forholdet mellem bånd 3 og 5
5395	10% fraktil af forholdet mellem bånd 15 og 17

**Tabel 17.7:** De udvalgte features til estimat af vandindhold (LP16).

1% interval, fra 24-50% med 2% interval og fra 51-91% med 10% interval. Fraktilerne er valgt sådan for at få en god indikation af hvilke resultater de mindre fraktiler giver, og med den formodning at resultaterne varierer mindre og mindre efterhånden som en større del af data medtages. Beregningen af features på baggrund af alle disse fraktiler på samtlige billeder er desuden relativt tidskrævende, så det var ønsket at begrænse antallet af beregninger.

Det maksimale antal variable og  $\lambda$  er sat til de værdier, der blev fundet bedst under modelgenereringen baseret på hele billedet. Efter konstateringen af at log-transformationen af respons ikke gav bedre resultater er der kun beregnet modeller baseret på fraktiler for de rå HB43 og LP16 vandindholdsmålinger.

Testdataenes standardafvigelser for modeller baseret på første til cirka femte/sjette fraktil er markant højere end de øvrige for både HB43 og LP16, som det ses på figur 17.19(b) og 17.20(b), og årsagen kunne være at datamængden her i form af antal pixels ganske enkelt er for lille til at give tilstrækkelig variation og statistisk sikkerhed i de udregnede features<sup>32</sup>. For træningsdata opnås det bedste resultat lige omkring den tiende fraktil i begge tilfælde, det er dog først omkring den fyrretyvende fraktil at den laveste standardafvigelse for testdata findes. For HB43 er der bortset fra en mindre top ved fraktil 15, en relativt lille forskel på de fundne standardafvigelser, der fra fraktil 10 og opefter ligger omkring  $0,7 \pm 0,02-0,03$ , se figur 17.19(b). For LP16, der som beskrevet i afsnit 17.6 ligger højere end HB43 i standardafvigelser, er der noget større variation i resultaterne for de forskellige fraktiler. Op til omkring fraktil 30 ligger standardafvigelsen og svinger ved 1,25, hvorefter den ved fraktilerne fra ca. 35-60 ligger omkring 1,15 for at stige en smule for de sidste fraktiler.

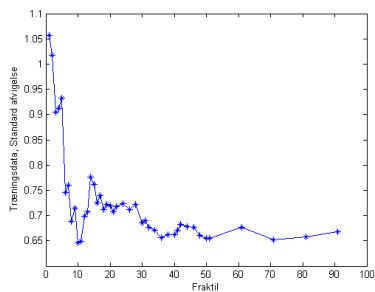
Begge kurvernes forløb på figur 17.19(b) og 17.20(b) giver ikke et indtryk af, at de fraktiler fundet som værende de bedste, er et meget sikkert valg. Havde figurerne i stedet vist et tydeligt minimum ét sted med jævnt stigende standardafvigelser ud fra denne fraktil havde konklusionen været mere oplagt, men det er ikke tilfældet her. For HB43 giver fraktiler fra ca. 30 og opefter et mindst ligeså godt resultat som ved at benytte hele billedet til generering af features og lige omkring fraktil 40 er resultatet 4% bedre. For LP16 giver fraktilerne fra omkring 40 til 60 et ca. 3-4% bedre resultat i forhold til at benytte hele billedet.

Model	Fraktil	Std. træning	Std. test	Antal variable
HB43	38	0,66	0,68	16
LP16	40	0,92	1,14	21

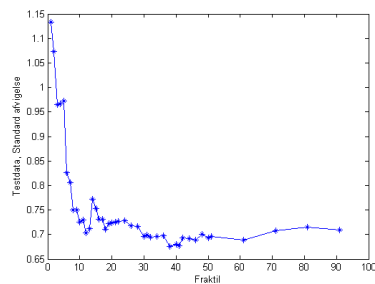
**Tabel 17.8:** Modelresultater for estimering af vandindhold ud fra features af en fraktil af billedet.

<sup>32</sup>Det benyttede billedareal er 620x620 pixels, så 1% er 3844 pixels



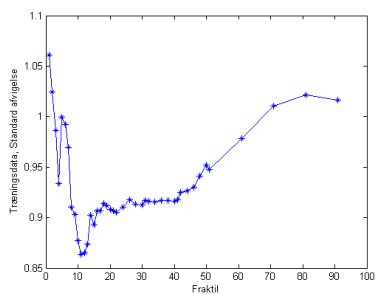


(a) HB43 standardafvigelser på træningsdata for forskellige fraktiler

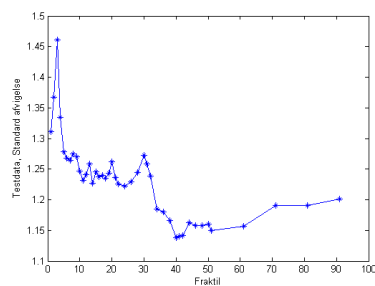


(b) HB43 standardafvigelser på testdata for forskellige fraktiler

**Figur 17.19:** HB43 standardafvigelser for vandindholdsestimat på trænings- og testdata for forskellige fraktiler.

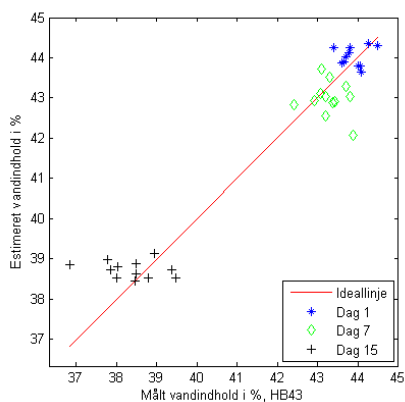


(a) LP16 standardafvigelser på træningsdata for forskellige fraktiler

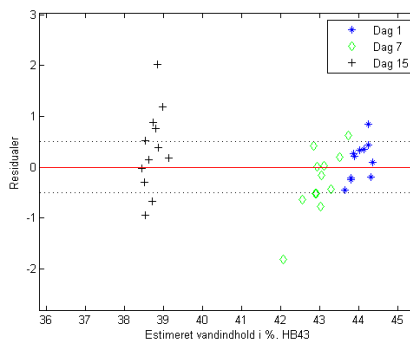


(b) LP16 standardafvigelser på testdata for forskellige fraktiler

**Figur 17.20:** LP16 standardafvigelser for vandindholdsestimat på trænings- og testdata for forskellige fraktiler.



(a) HB43 estimater

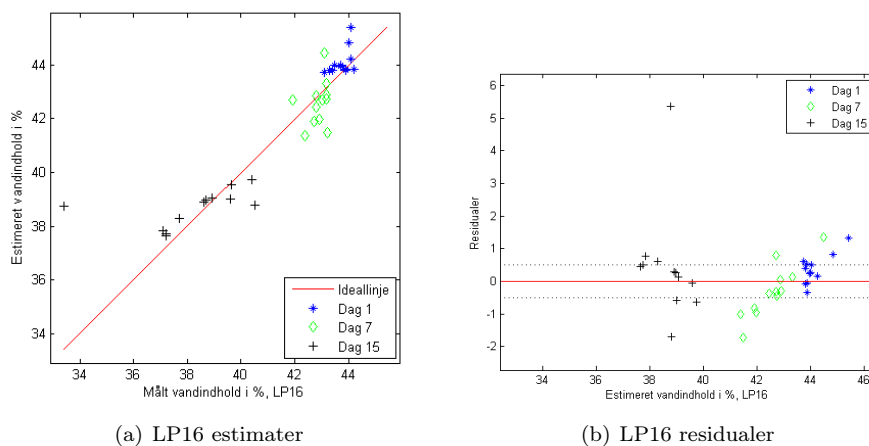


(b) HB43 residualer

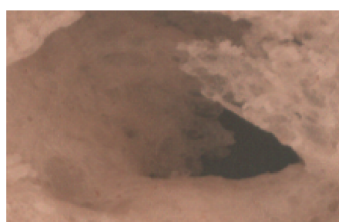
**Figur 17.21:** HB43 estimater og residualer af den bedste fraktil-model.

Figur 17.21 og 17.22 viser estimater og residualer for de bedste modeller for hhv. HB43 og LP16 baseret på fraktiler. Igen ses det at estimaterne for HB43 på dag 15 ligger meget tæt i forhold til variationen af de faktiske målinger.

I kapitel 14 er det fremhævet at huller i den fotograferede brødskeive kan være en ekstra kilde til variation, der dermed gør model-estimaterne ringere. På figur 17.23 er vist et hul fra den samme brødskeive med og uden en ROI svarende til den 38. fraktil af det negative krumningsbillede, der blev fundet som værende bedst til estimering af HB43. Det ses at selve hullet og store dele af de mørke områder er blevet masket ud og derfor ikke medtages i beregningerne af features.



**Figur 17.22:** LP16 estimater og residualer af den bedste fraktil-model.



(a) Zoom af et hul i en brødskive



(b) Zoom af et hul i en brødskive med en ROI svarende til den 38. fraktil

**Figur 17.23:** Et hul i en brødskive med og uden ROI svarende til den 38. fraktil af krumningsbilledet.



Del V

Implementering



Denne del beskriver de vigtigste koderestrukturer for analysefunktionerne. Al koden er lavet som Matlab funktions-filer og følger med denne rapport på CD i bilag J. Det medfølgende bibliotek er en udvidelse til den modificerede MSI-toolbox<sup>33</sup>, og kan ligge som et underkatalog til denne<sup>34</sup>. Visse eksisterende funktioner i MSI-toolboxen er blevet lettere modificeret, for at understøtte nye funktionaliteter. Den modificerede toolbox følger med på CD'en.

På figurene herunder er koderestrukturerne forklaret med deres logiske opbygning. De faktiske funktionsnavne er angivet i kursiv og firkantede parenteser, hvis procesbeskrivelsen dækker over en specifik funktion. Der hvor et valg er betinget af en brugerangivet parameter er denne angivet på samme måde i "valgdiamanterne".

Koderestrukturerne er vist som træstrukturer med flow-diagrammer, og vil blive forklaret oppefra og ned. De steder hvor en proces-box er markeret med fed, findes der en anden figur der beskriver understrukturene for denne proces.

---

<sup>33</sup>Udleveret af Jens Michael Carstensen

<sup>34</sup>Se installationsvejledningen i kapitel 22 for yderligere oplysning om hvordan softwaren installeres.





# Preanalyse og preprocessering

---

Analysen af spegepølser og kiks med barrieresystem kan deles ind i 2 blokke som vist på figur 18.1. De to funktioner til at udføre salamianalysen og ”kiks med barrieresystem”-analysen er henholdsvis *msisalamianalysis* og *msibsanalysis*.

Preanalysen er meget ens for både salami og barrieresystem analyserne, og den generelle flowstruktur på figur 18.2 gælder for begge preanalyser, eneste forskel er parameternavnene og forretningsreglerne<sup>35</sup>.

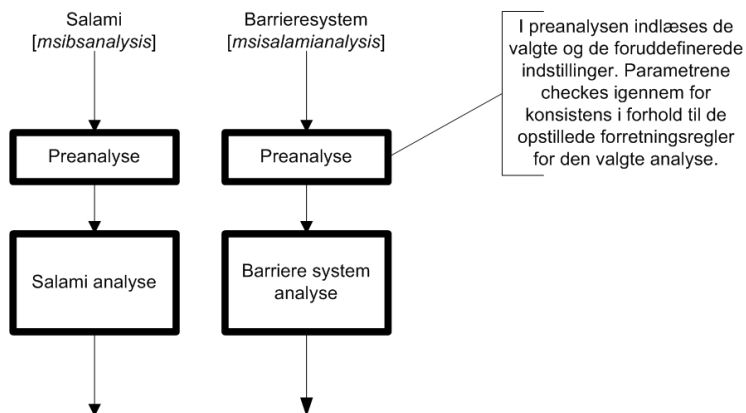
Funktionerne tager et variabelt antal parametre som sætter de ønskede forretningsregler, variabel værdier og resultat ønsker. Som standard er de anbefalede indstillinger valgt på forhånd, sådan at funktionerne kan køre med mindst mulige brugervalgte input.

Efter indlæsning af de angivne parametre checkes det igennem at der ikke er angivet modstridende parameterværdier, eller parameter værdi-sæt der ikke er anbefalet.

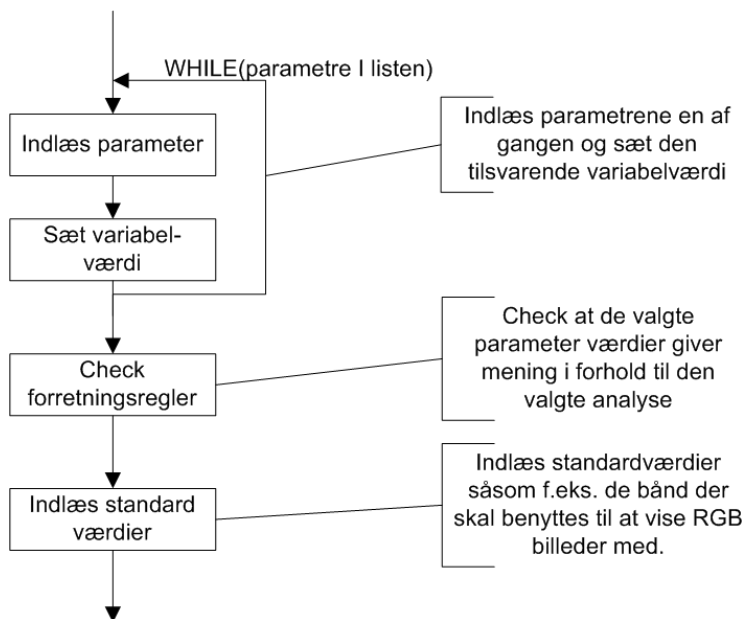
Til sidst indlæses standardværdier fra eksterne funktioner. Disse funktioner fungerer som datalagre, sådan at der ikke står vigtige konstanter i mange filer.

---

<sup>35</sup>Bemærk at disse forretningsregler ikke er de samme som beskrevet i kapitel 4.



**Figur 18.1:** For både salami og barrieresystem analyserne udføres først en række procedurer der strukturmæssigt er ens for begge analyser.



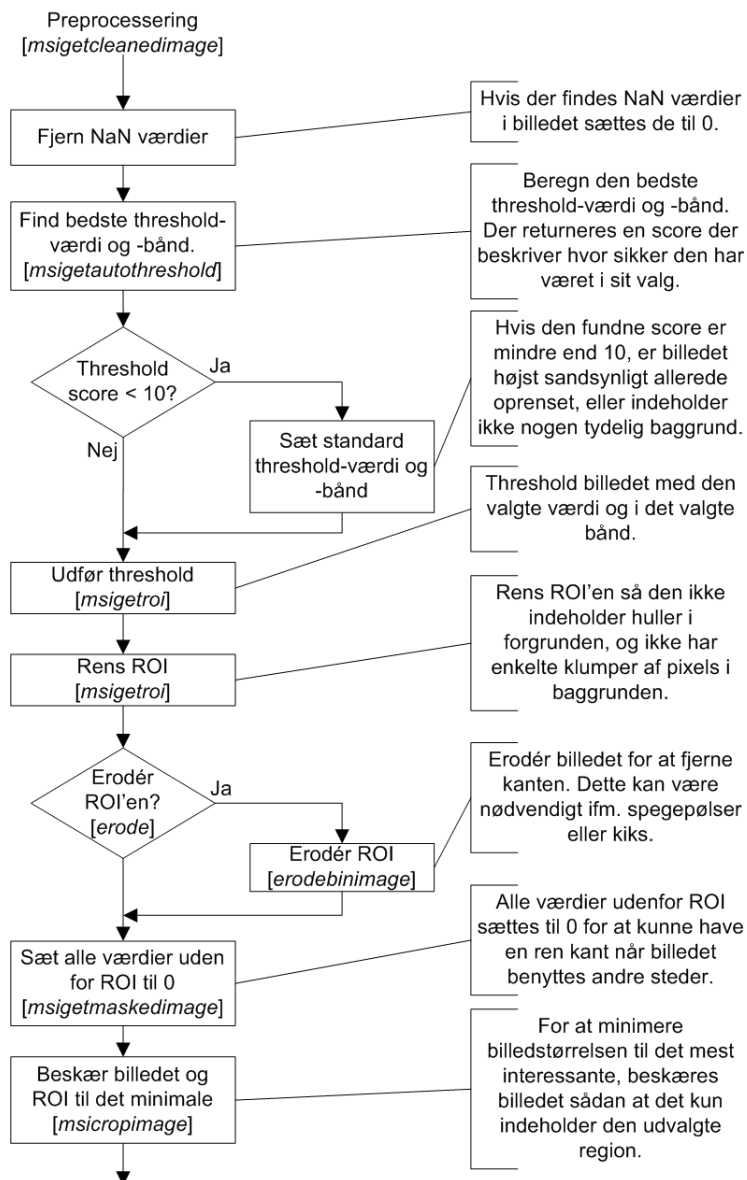
**Figur 18.2:** Flowchart for pre-analysen. Pre-analyse procedurer der udføres for både salami og barrieresystem analyserne før den egentlige analyse startes.

---

F.eks. bliver de bånd der anvendes til RGB-visning læst ind på denne måde, sådan at hvis de benyttede billeder kommer fra et VideometerLab system der benytter andre bølgelængde opsætninger end dem der er angivet i bilag A skal disse oplysninger kun ændres dette ene sted.

Figur 18.3 viser funktionsflowet for preprocesseringsfunktionen *msigetcleaned-image*. Først fjernes eventuelle NaN værdier, ved at sætte dem til 0. Efterfølgende beregnes det bedste threshold bånd og værdi med funktionen *msigetautothreshold*. Samtidig udregnes en score der beskriver hvor sikker algoritmen er på at den har fundet for- og baggrund. Hvis denne score er lavere end 10, sættes thresholdbåndet og -værdien til deres standardværdier. Efterfølgende benyttes funktionen *msigetroi* til at udføre en rå threshold på billedet med den fundne værdi og bånd. Thresholden resulterer i en ROI der renses så den ikke indeholder mindre huller i forgrunden, eller har små klaser af pixels i baggrunden. Derefter kan den evt. eroderes et specifikt antal gange med funktionen *erodebinimage*.

Når ROI'en er genereret og renses, sættes alle værdier uden for ROI'en til 0 med funktionen *msigetmaskedimage*, sådan at der fremkommer en pæn kant på det objekt der analyseres. Når kanten er sat til 0, skæres billedet til med funktionen *msicropimage*, sådan at den første og sidste række og kolonne pixels i billedet er kanten af objektet. Det samme gøres med ROI'en før begge returneres.



**Figur 18.3:** Preprocessingflowet. Billederne renses, den bedste threshold værdi og bånd bestemmes, og billederne thresholdes og beskæres.

# Spegepølser

---

Figur 19.1 og 19.2 viser funktions flowet for salami analyse funktionen *msisalamicanalysis*. Figuren er delt op på to figurer, men skal ses som en enkelt lang sammenhængende proces.

Funktionen tager filnavnet på det billede der ønskes analyseret som første argument, og derefter et variabelt antal parametre der benyttes i preanalysen som beskrevet ovenfor.

Det specificerede billede læses ind og renses som beskrevet ovenfor med funktionen *msigetcleanedimage*. Hvis der benyttes et billede der allerede er rensset, kan dette slås fra med parameteren *clean*.

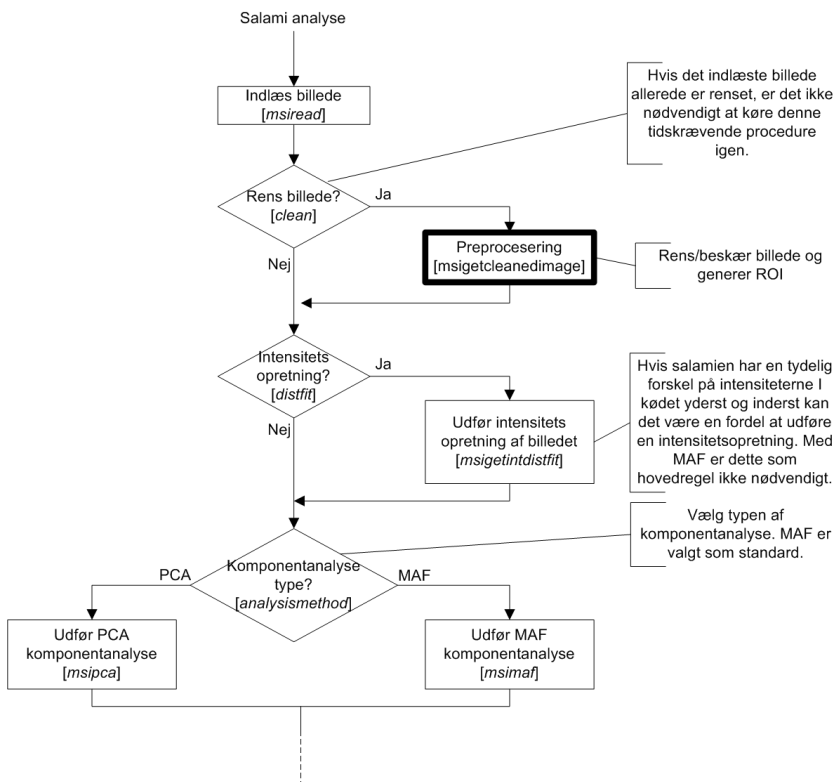
Hvis intensitetsopretning er valgt med parameteren *distfit* udføres en intensitetsopretning af billedet med funktionen *msigetintdistfit*. Da det typisk ikke er nødvendigt at benytte intensitetsopretning hvis analyse typen er sat til MAF vil der blive genereret en advarsel når funktionen kører hvis begge parametre er slået til.

Derefter udføres enten en PCA- eller MAF-analyse på billedet med funktionen *msipca* eller *msimaf*. Analysetypen er som standard sat til MAF, men kan ændres med parameteren *analysismethod*.

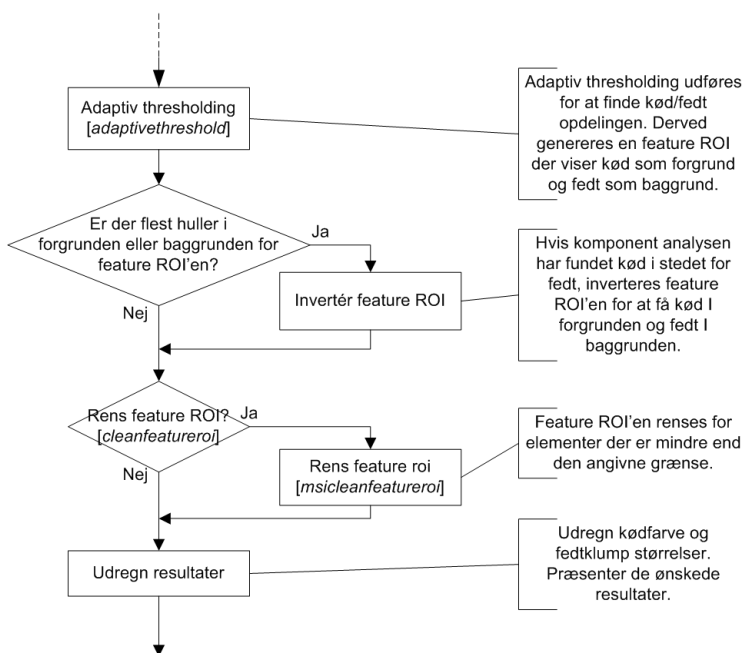
For at skille fedtet fra kødet og lave en feature-ROI udføres en adaptiv thresholding med funktionen *adaptivethreshold* på resultatet af PCA- eller MAF-analysen. Efter denne opdeling tælles antallet af huller i for og baggrunden for feature-ROI'en. Dette gøres fordi PCA og MAF-analyserne ikke altid fremhæver fedtet som mest betydende egenskab, men måske istedet for vælger kødet. For derfor at kunne bestemme kødfarven bliver vi nødt til at sikre os at det rent faktisk er kødet og ikke fedtet vi kigger på. Da vi går ud fra at en spegepølse består af kød med fedtklumper i, der derfor danner huller i kød-ROI'en kan vi tælle hullerne og på den måde bestemme om det er fedt eller kød. Hvis ROI'en er vendt om, inverteres den.

Da vi går ud fra at de mindste dele i en salami trods alt må have en vis størrelse, giver det ikke mening at kigge på fedt og kød klatter under en vis diameter, og feature-ROI'en renses derfor for disse med funktionen *cleanfeatureroi*. Hvis dette ikke ønskes, kan det slås fra med parameteren *cleanfeatureroi*.

Når kød/fedt adskillelsen er færdig, udregnes værdier for kød- og fedtfarve og fedt-klumpernes størrelse.



Figur 19.1: Flowchart for salamianalysen, del 1. Fortsættes på figur 19.2.



**Figur 19.2:** Fortsat fra figur 19.1. Flowchart for salamianalysen, del 2. Salami analysen udføres i skridt med de valgte analyse og procedure metoder. Undervejs vises de færdige resultater så snart de er tilgængelige, og kun hvis der er bedt om dem.



# Barriersystem på kiks

---

Figur 20.3 viser funktionsflowet for barriersystem analysefunktionen *msibsanalysis*. Funktionen er delt op i to dele, den første er modelgenereringen og den anden er modelanvendelsen. Den ene eller begge dele kan slås fra ved at sætte en af de bool'ske parametre *generatemodel* eller *applymodel* til falsk.

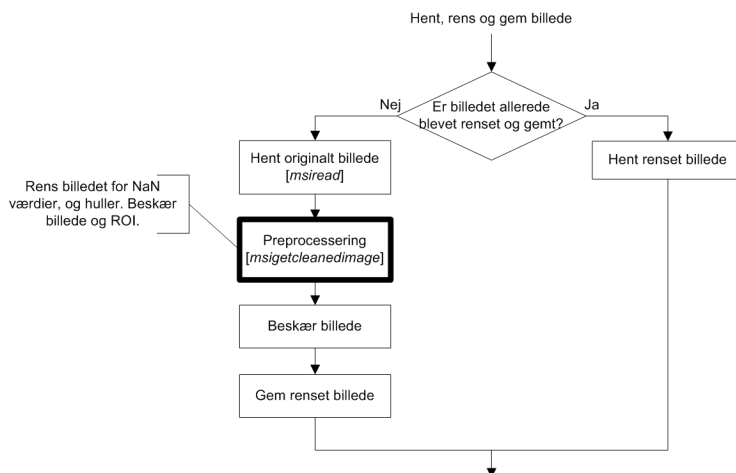
Funktionen tager en liste af billednavne og tilhørende responsværdier som første input og derefter et variabelt antal parametre der benyttes i preanalysen som beskrevet ovenfor. På figur 20.1 ses formatet for responslisten. Se iøvrigt brugervejledningen i kapitel 24 for et eksempel på hvordan sådan en respons liste kan se ud.

Responselist
- filename
- response

**Figur 20.1:** Strukturformat for *Responselist*.

Oprensningen af billederne er typisk det der tager længst tid ved denne procedure. Derfor sørges det for at når billederne én gang er rensset, gemmes de som *mat*-filer<sup>36</sup> der så kan hentes ind næste gang funktionen har brug for billedet. Proceduren er vist på figur 20.2. Først undersøges det om *mat*-filen allerede

<sup>36</sup>Matlabs eget filformat til variable.



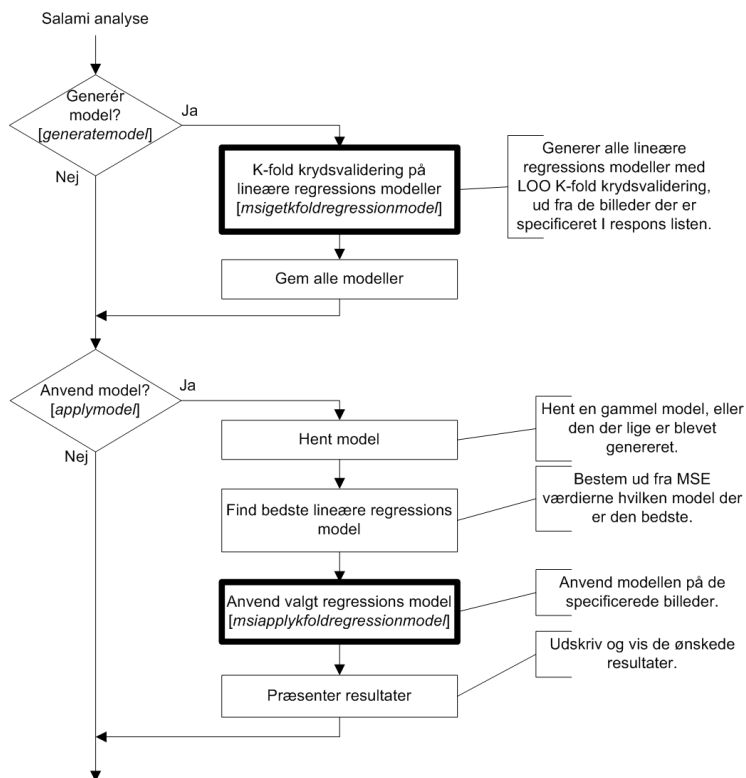
**Figur 20.2:** Funktionalitet til at indlæse et rått billede, rense og gemme det eller indlæse et allerede rensset billede.

findes. Hvis den gør, læses billedet ind direkte fra filen. Hvis filen ikke findes, læses billedet ind og renses. Før billedet gemmes, skæres det til så det har den ønskede vindues størrelse, angivet med parameteren *windowradius* der angiver radius på vinduet.

## 20.1 Modelgenerering

For at generere modellen benyttes funktionen *msigetkfoldregressionmodel* til at udføre en LOO krydsvalidering med lineær regression på de specificerede billeder. På figur 20.4 ses funktions flowet for K-fold krydsvalideringen af en lineær model. Funktionen er delt ind i to dele. Den første løber igennem de  $K$  billeder fra responslisten, henter eller genererer rensede billeder, samt sætter alle billederne sammen i en række. Det samme gøres med responsdataene, som vist på figur 20.5 hvor respons dataene er vist overfor RGB repræsentationen af testdataene.

Efter at der er genereret en stribe testdata, beregnes modellen for den lineære regression med funktionen *msipredlogregpar*, af de  $K - 1$  testdata overfor de tilhørende respons data. Modellen overføres derefter til valideringsdataene  $k_2$  med funktionen *msiapplyregpar*, som der derefter beregnes MSE, gennemsnit og standardafvigelser for med funktionen *msifogls*. Dette gøres igen for alle  $K$  billeder.

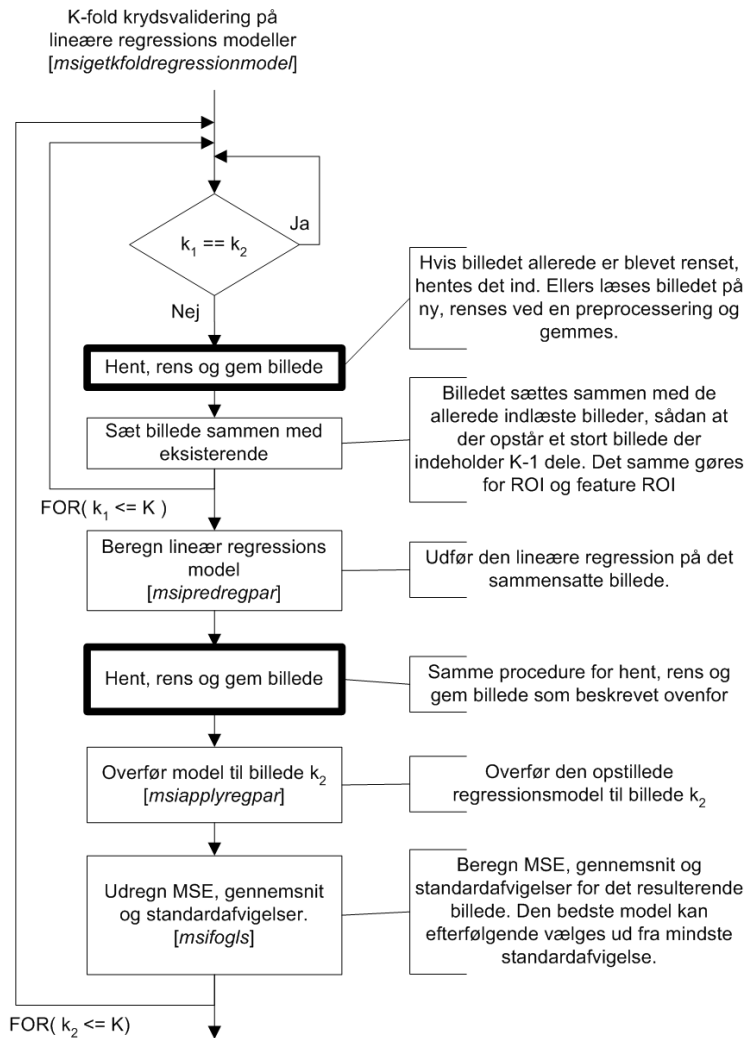


Figur 20.3: Overordnet flowstruktur for barrieresystem analysen.

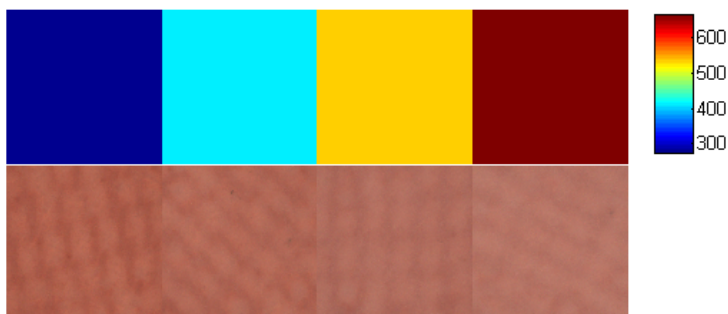
Når krydsvalideringen afsluttes, gemmes alle modeller og MSE-værdier i de tre filer *opmaalteresultater\_s1\_offsets.mat*, *opmaalteresultater\_s1\_regcoefs.mat* og *opmaalteresultater\_s1\_mses.mat* der henholdsvis indeholder offset, regressionsparametre og de beregnede MSE-værdier for alle  $K$  opstillede modeller.

## 20.2 Modelanvendelse

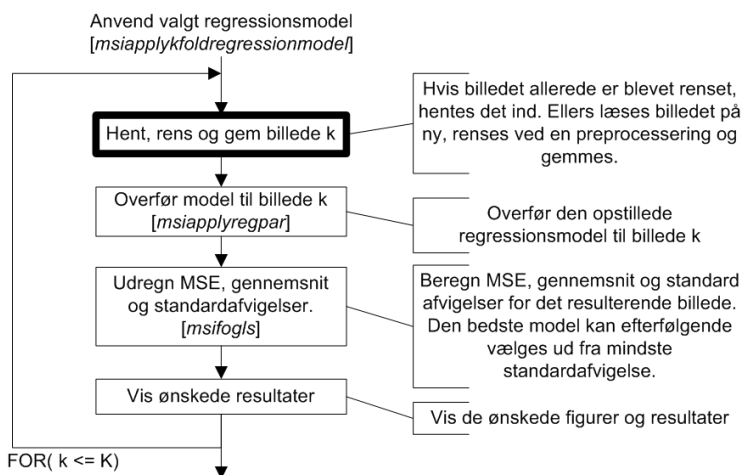
Den anden del af barriersystem analysen består i at anvende den opstillede model på listen af billede med tilhørende respons. Figur 20.6 viser flowet for funktionen *msiapplykfoldregressionmodel*. De  $K$  billeder køres igennem og renses før modellen bliver overført til dem med funktionen *msiapplyregpar*. Derefter udregnes MSE, gennemsnit og standardafvigelse for billedet, i forhold til respons værdien, med funktionen *msifogls*, og de ønskede resultater udskrives til skærmen og vises på de ønskede grafer.



**Figur 20.4:** Generering af lineær regressionsmodel ud fra en K-fold krydsvalidering.



Figur 20.5: Sammensat respons- og testdata for vandbarriere på kiks.



Figur 20.6: Anvendelse af den lineære regressionsmodel fra figur 20.4.

# Vandindhold i brød

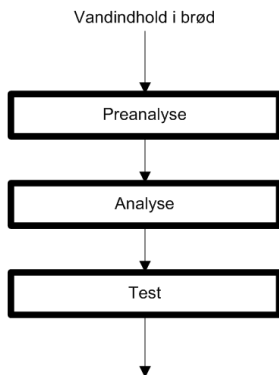
---

Analysen af vandindhold i brød er delt i tre hoveddele som vist på figur 21.1. Som udgangspunkt laves et *matlab-struct*-array, der kan indeholde filnavne, målt vandindhold og alle de øvrige tilgængelige data for brødene, dagen billedet er taget, brød-nummer, tilsat enzym og tilsat emulgator.

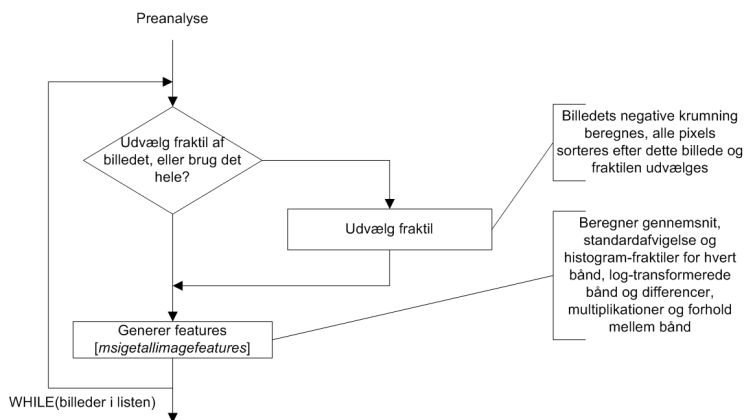
Med denne data-liste kan alle de ønskede features beregnes og gemmes, både for hele billedet, og for de fraktiler af snitfladen man ønsker, som vist på figur 21.2. Til den samlede test er der genereret features for i alt 43 forskellige fraktiler og for hele billedet.

Selve analysen og modelgenereringen på figur 21.3 starter med at den ønskede del af datasættet udvælges med funktionen *msigenerateregmodelldata*, f.eks. hvis der skal laves en model kun for brød uden tilsat emulgator. Funktionen er lavet så den kan tage et vilkårligt antal betingelser som streng- eller talvariable, der skal opfyldes for at den pågældende sample medtages i modellen eller testen.

Dernæst klagøres datasættet til LARS-EN analysen hvor det valgte featuresæt og respons trækkes ud af data-strukturen hvorefter det centreres og normaliseres, inden selve LARS-EN regressionen og krydsvalideringen udføres for alle de valgte  $\lambda$ - og *max-var*-værdier. Når den bedste model skal findes vil de to parametre typisk varieres over et relativt bredt spektrum som i den udførte analyse i afsnit 17.4, hvor *max-var* blev sat til hhv. 10, 20, 40, 100, 200 og 400 og for hver af

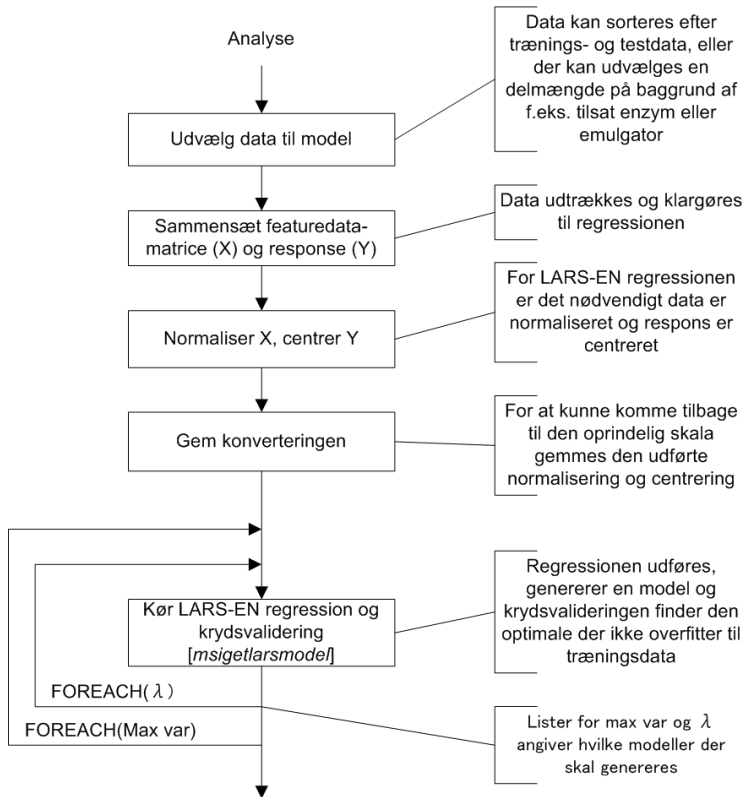


Figur 21.1: Overordnet flowstruktur for analysen af vandindhold i brød.



Figur 21.2: Flowstruktur for preanalysen af vandindhold i brød.





**Figur 21.3:** Flowstruktur for modelgenereringen af vandindhold i brød.

disse blev  $\lambda$  sat til  $10^{-6}$ ,  $10^{-5}$ ,  $10^{-4}$ ,  $10^{-3}$ ,  $10^{-2}$ ,  $10^{-1}$  og  $10^0$ . De genererede modeller gemmes for derefter at blive testet.

Flowet for testudførelsen er vist på figur 21.4 hvor der, efter den model der skal testes er indlæst, kan foretages en udvælgelse af datasættet, ligesom ved modelgenereringen, inden modellen benyttes på de valgte data, estimer udregnes og konverteres til den faktiske skala, så estimat og det oprindelige respons kan sammenlignes. Efter beregning af korrelation og standardafvigelse for alle modeller kan den bedste model findes.



Del VI

# Brugervejledning



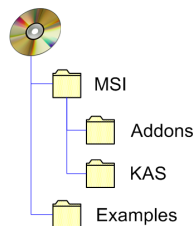
Denne del giver en praktisk anvendelig introduktion til den udarbejdede software. Først gives en vejledning i hvordan softwaren installeres, og derefter hvordan den benyttes for de tre emner, spegepølse, kiks med barriere og vandindhold i brød. Vejledningen forudsætter et vist kendskab til BNF-notation og Matlab, især hvordan funktioner kaldes, sti-navigering m.m.



# Installationsvejledning

---

Softwareen medfølger på den vedlagte CD i bilag J. På CD'en ligger koden i filstrukturen vist på figur 22.1. I roden af CD'en ligger dels selve kode-biblioteket i folderen *MSI* og et eksempel katalog, *Examples*, med kode der viser eksempler på kørsel af de implementerede funktioner. Folderen *MSI* indeholder underkatalogerne *Addons* og *KAS*. I selve folderen *MSI* findes den modificerede MSI toolbox. Folderen *KAS* er kildekoden hentet fra [Sjöstrand, 2006] og indeholder implementeringen af LARS og LARS-EN algoritmen og en funktion til krydsvalidering. Folderen *Addons* indeholder al den kode der er implementeret i dette projekt.



**Figur 22.1:** CD filstruktur.

For at kunne køre de implementerede funktioner skal software pakken først installeres i Matlab. For at installere pakken følges disse skridt:

1. Kopier folderen *MSI* med underkataloger til harddisken.
2. Start Matlab sæt *path*'en til at pege på den kopierede folder. Dette gøres ved at åbne menuen *File*→*Set Path*.
3. Klik på knappen *Add with Subfolders* og vælg stien på den kopierede mappe.
4. Klik på knappen *Save*, og derefter på *Close*.



# Spegepølser

---

For at køre en analyse af en spegepølse, køres funktionen *msisalamianalysis* med BNF notationen:

```
msisalamianalysis ('<filnavn> ['<parameternavn>' '<parameterværdi>']* ')
```

Hvor *filnavn* er navnet på det pølsebillede man ønsker analyseret, *parameternavn* er navnet på den parameter man ønsker at sætte, og *parameterværdi* er værdien af parameteren. Stjernen (\*) angiver at det indenfor de firkantede parenteser [] kan gentages et vilkårligt antal gange eller helt udelades. Funktionen kan kaldes med et variabelt antal sæt af parameternavn/parameterværdi. Her er vist en række eksempler på kørsel med funktionen:

```
% Den simpleste form, ubehandlet billede, skal kun bruge filnavn
msisalamianalysis('sample 1.hips');

% Hvis billedet allerede er rensat, så baggrund er sat til 0 og
  størrelsen beskåret.
msisalamianalysis('sample 1.hips', 'clean', false);

% Viser ubehandlet og behandlet billede, kun 2 cirkel-formede
  ringe (modsat adaptiv)
msisalamianalysis('sample 1.hips', 'showraw', true, 'showrgb',
  true, 'meatrings', 2, 'ringmethod', 'circle');

% benytter pca og komponent 2 i stedet for standard MAF komponent
  1.
msisalamianalysis('sample 2.hips', 'showmethod', true, '
  analysemethod', 'pca', 'component', 2);
```



## Barriersystem på kiks

---

For at køre en analyse af barrierelaget på en kiks, skal der først oprettes en responsfil der indeholder navnene på alle de filer der ønskes at køre en analyse på, samt de tilhørende respons værdier. Respons værdierne er her tykkelsen på barrierelaget. Denne fil kan oprettes med funktionen *generateresponselist* med følgende BNF notation:

```
generateresponselist '( <modelnavn> [',' <filnavn> ',' <responsværdi>]* )'
```

Hvor *modelnavn* er navnet på responslisten, *filnavn* er navnet på billedfilen og *responsværdi* er responsværdien for billedet. Funktionen kan kaldes med et variabelt antal filnavn/responsværdi sæt. Her er vist et eksempel der opstiller en respons liste med navnet *reslst1* og en række kiks med forskellige barrierelagstykkelser.

```
generateresponselist('reslst1', ...
    'kiks1.hips',0, ...
    'kiks2.hips',100, ...
    'kiks3.hips',200, ...
    'kiks4.hips',300, ...
    'kiks5.hips',400 ...
);
```

For at køre selve analysen benyttes funktionen *msibsanalysis*. Også denne funktion har en enkelt obligatorisk parameter, nemlig navnet på den genererede responsliste. BNF notationen ser sådan ud:

```
msibsanalysis ('<modelnavn> ['<parameternavn> ',  
                    <parameterværdi>]*')
```

Hvor *parameternavn* er navnet på den parameter man ønsker at sætte, og *parameterværdi* er værdien af parameteren. Her er vist en række eksempler på kørsel med funktionen:

```
%Benyt alle standard indstillinger for responslisten 'reslst1'  
msibsanalysis('reslst1');  
  
%Benyt modelnavnet 'modell' og filtrer filen 'kiks1.hips' fra.  
msibsanalysis('reslst1', 'modelname', 'modell', 'filter', 'kiks1.  
    hips');  
  
%Benyt den genererede model 'modell'  
msibsanalysis('reslst1', 'modelname', 'modell', 'generatemodel',  
    false);
```

# Vandindhold i brød

---

Først skal datasættet gøres tilgængeligt for funktionerne i Matlab. Til dette anvendes funktionen *struct*, som har følgende opbygning

```
data(indeks) = struct ('[ <parameternavn> ', ' <parameterværdi>]* ')'
```

Parameternavn og parameterværdi angives i par og der opbygges et array, hvor hvert indeks er en sample.

Koden her viser et eksempel på hvordan data for et enkelt brødbillede er indsat

```
bread(1) = struct('filename', '060289 Bread1_dgl.hips', 'bread',  
    1, 'day', 1, 'enzyme', 'Control', 'emulsifier', 0  
    , 'hb43', 44.1, 'lp16', 43.91)
```

Koden herunder genererer features for alle filer i den gemte datastruktur *bread*

```
% Katalog der indeholder billedfilerne  
directory = 'C:\Projektdata\Bread\train';  
  
% Angiver navnet på det felt i strukturen, der indeholder  
% filnavnet  
filenamefield = 'filename';  
  
% Genererer features for hele billedet og få returneret  
% datastrukturen indeholdende features  
datafeatures = msibatchgeneratefeatures(bread, directory,  
    filenamefield)
```

De underfunktioner, der styrer hvilke kombinationer af bånd der skal genereres features for, og hvilke features der skal udregnes for hver enkelt, kan ikke umiddelbart styres. Dog gør opbygningen at der med relativt få ændringer i koden kan laves andre kombinationer af bånd eller andre features for hver enkelt, og så længe de benyttede features er det samme for modelgenerering og modeltest har det ingen betydning for selve kørslen, hvilke features der er tale om.

Af praktiske årsager er genereringen af features for fraktiler af billedet holdt i en separat funktion, der dog benytter sig af de samme underfunktioner som featuregenereringen for det komplette billede.

Koden her genererer fraktil-features for fraktilerne fra 1-99% med 2% spring

```
% Genererer features for fraktilerne fra 1% til 99%
% med 2% spring
fractilelist = 0.01:0.02:0.99;

% Fjern kant efterladt af filteroperationen
edgeremoved = 15;

% Angiver navnet på det felt i strukturen,
% der indeholder filnavnet
filenamefield = 'filename';

% Angiver navnet på det felt i strukturen, der skal
% indeholde fraktilfeatures
fracfeaturesfield = 'fracfeatures';

msigeneratefracfeatures(datafeatures, filenamefield,
    fracfeaturesfield, edgeremoved, fractilelist)

% Den færdige datastruktur med både features for hele
% billedet og fraktiler gemmes automatisk
```

Nu er alle data klar til selve analysen. Hvis det ønskes kan et sub-sæt af datastrukturen vælges ud til modelgenereringen ved at benytte funktionen *msigenerateregmodelldata* som vist herunder med BNF-notation.

```
msigenerateregmodelldata '( <datasæt> [',' <parameternavn> ','
    <parameterværdi>]* )'
```

Et eksempel på udvælgelsen er vist her

```
% Datasæt indlæses
load 'data_features_fracfeatures_combined.mat'

% Datasættet er indlæst i variabelen 'data' og indeholder i alt 36
% samples
selecteddata = msigenerateregmodelldata(data, 'enzyme', 'Control',
    'emulsifier', 0);

% Variablen 'selecteddata' indeholder nu de 6 samples der
% opfylder sorteringskriteriet
```

Modelgenerering og -test er anbragt i samme funktion *msilarsenmodelsript*, hvor hver af de genererede modeller umiddelbart testes både med træningsdata og testdata. Alle de angivene parametre er nødvendige for funktionen

```

% Listen der angiver hvilke max-var værdier der skal
% laves modeller med
stop_list = [-10, -20, -40, -100, -200, -400];

% Listen der angiver hvilke lambda værdier der skal
% laves modeller med
lambda_list = [10^-6 10^-5 10^-4 10^-3 10^-2 10^-1 10^-0];

% Navnet på responsvariablen i datastrukturen,
% f.eks. 'hb43' eller 'lp16'
responseset = 'hb43';

% Angiver om responsvariablen er log-transformeret,
% 'log' hvis den er, 'normal' hvis den ikke er.
responsetype = 'normal'; % eller 'log'

% Kataloget med træningsdata
modeldatadir = 'C:\Projektdata\Bread\train';

% Kataloget med testdata
testdatadir = 'C:\Projektdata\Bread\test';

% Kataloget hvor de genererede modeller skal gemmes
modelsavedir = 'C:\Projektdata\Bread\models';

% Kataloget hvor de genererede modeltest skal gemmes
modeltestdir = 'C:\Projektdata\Bread\modeltests';

% Angiver navnet på den fil der indeholder data-strukturen
featurefile = 'breaddataandfeatures';

% Pre-fix, der sættes foran alle de genererede filer
fileprefix = 'broed_larsen_all';

% Variabel-navnet på det felt i strukturen, der indeholder
% de features som skal anvendes
featurefieldname = 'features';

% Modellen køres
msilarsenmodelsript(stop_list, lambda_list, responseset,
    responsetype, modeldatadir, testdatadir, modelsavedir,
    modeltestdir, featurefile, fileprefix, featurefieldname)

```

Modellerne af fraktiler af billedet genereres og testet uafhængigt af ovenstående test der varierer på *max-var* (variablen *stop* i koden) og  $\lambda$ . Her vælges et bestemt sæt af de to og en model for hver af fraktilerne med disse parametre genereres, testes og den bedste findes. I koden her er der både kørt en modelgenerering og -test for HB43-responset og LP16-responset.

```
% Fraktiler fra 1-99 med spring på 2
fraclist = 1:2:99;

% Værdier for max-var og lambda angives
stop = [-100];
lambda = [10^-5];

% Kataloger defineres
modeldatadir = 'C:\Projektdata\Bread\train';
testdatadir = 'C:\Projektdata\Bread\test';
modelsavedir = 'C:\Projektdata\Bread\fracmodels';
modeltestdir = 'C:\Projektdata\Bread\fracmodeltests';

% Navn på filen der indeholder data-strukturen
featurefile = 'data_features_fracfeatures_combined.mat';

% Prefix til de genererede resultat-filer
fileprefix = 'broed_larsen_all';

% Navn på den variabel i data-strukturen der indeholder fraktil-
    features
featurefieldname = 'fracfeatures';

% Modelgenerering og test med 'HB43'-responset
responseset = 'hb43';
responsetype = 'normal';

% Modellen køres
msilarsenmodelfracscript(fraclist, stop, lambda, responseset,
    responsetype, modeldatadir, testdatadir, modelsavedir,
    modeltestdir, featurefile, fileprefix, featurefieldname)

% Modelgenerering og test med 'LP16'-responset
responseset = 'lp16';
responsetype = 'normal';
stop = [-400];
lambda = [10^-2];

% Modellen køres
msilarsenmodelfracscript(fraclist, stop, lambda, responseset,
    responsetype, modeldatadir, testdatadir, modelsavedir,
    modeltestdir, featurefile, fileprefix, featurefieldname)
```



Del VII

Opsummering



## Konklusion

---

For alle tre emner er kravene opstillet i kravspecifikationen nået. For spegepølserne er vi nået frem til en løsning der er meget tæt på at være fuldautomatisk, og i langt de fleste tilfælde kan den da også benyttes uden assistance eller yderligere viden om systemet. For barrieresystemet har vi nået de mål der blev opstillet i kravspecifikationen. Vi må dog konstatere at selvom modellen har givet lovende resultater, har datasættet ikke været godt nok til at vurdere om resultaterne er brugbare. For brødet har vi ligeledes opnået målene fra kravspecifikationen, og der blev givet et tal for hvor nøjagtig målingen er. Ud fra disse resultater er det Danisco's job at beslutte om resultaterne er gode nok, eller om metoder og resultater skal forbedres inden de kan tages i brug.

Derudover er der blevet udviklet en softwarepakke der udnytter informationen i et multispektralt billed til at lave en robust opdeling mellem for- og baggrund billedet. Denne pakke har vist sig anvendelig overfor alle de billeder vi har haft til rådighed i vores datasæt, men har dog visse svagheder overfor billeder med meget lidt baggrund.

## 26.1 Spegepølser

Opgaven med spegepølserne var at udvikle en metode til at estimere farven af kødet i en spegepølse på en sammenlignelig og reproducerbar måde.

Dette er gjort ved at anvende PCA (principal komponent analyse) eller MAF (minimum autokorrelationsfactor). Derefter er fedtet og kødet skilt fra hinanden med en adaptiv thresholding, og der beregnes så en gennemsnitsværdi af farven fra midten og ud mod kanten, i det antal ringe man er interesseret i at få præsenteret resultatet i.

Derudover er der fremstillet en metode til at skille fedtklumperne fra hinanden og angive antallet og størrelserne af dem.

Undervejs er det blevet forsøgt at opstille en generel model baseret på lineær eller logistisk regression til adskillelse af fedt og kød. Denne viste sig dog ikke praktisk anvendelig, da den ikke kunne gøres generel *nok* til at virke godt for alle billeder i datasættet uden at blive *for* generel, og dermed lave for mange fejlkategoriseringer. Til gengæld viste den adaptive thresholding sig yderst anvendelig til formålet, og da den ikke krævede nogen form for forudgående kendskab til en opstillet model er denne metode blevet anvendt i den endelige software.

En del af opgaven lød på at udvikle en metode til at kunne sammenligne farverne på forskellige spegepølser. Derfor er der udarbejdet en generel spegepølsefarveskala, hvor man med to tal placerer spegepølserne på skalaen og kan betragte dem relativt i forhold til hinanden. De to tal udtrykker henholdsvis hvor lys eller mørk spegepølsen er, samt hvor rød den er.

De udleverede sensorikdata med spegepølseeksperternes subjektive betegnelser for kødfarven er sammenholdt med de beregnede resultater. De spegepølser med "ekstreme" betegnelser lå også i ekstremerne af farveskalaen og spegepølser der var givet samme farvebetegnelse lå typisk relativt tæt. Der var dog også tilfælde, hvor to spegepølser med samme farvebetegnelse ikke lå specielt tæt på farveskalaen. Dette kunne indikere at fedtindholdet, fedtklumpernes størrelser og fedtets farve kan have været med til at give indtrykket af en anden kødfarve end den faktiske.

## 26.2 Barrieresystem på kiks

Opgaven for barrieresystem på kiks gik ud på at estimere gennemsnitstykkelsen af barrieresystemet og at bestemme hvor jævnt et lag den er påført, samt vise et højdekort for barrierelaget.

Der er opstillet en model, baseret på lineær regression, ud fra en serie af kiks hvor barrierelagstykkelsen er kendt på forhånd. Barrierelagstykkelserne er fundet ved at veje kiksene før og efter påføring af barrierelaget. Modellen beskriver højden af det påførte barrierelag og ud fra denne model kan standardafvigelsen på estimaterne alene fortælle hvor jævnt barrierelaget var påført. Den gennemsnitlige standardafvigelse for jævnheden er fundet til hhv.  $30,61\mu\text{m}$  og  $52,84\mu\text{m}$  for de to dataserier. Derudover er barrierelagstykkelsen præsenteret som et højdekort hvor tykkelsen i de enkelte områder af kiksken kan aflæses ud fra farvespektret. Den beregnede standardafvigelse på testdata for den bedste model (baseret på *serie 1*) er på  $93.53\mu\text{m}$  i forhold til de målte tykkelser.

Modellen gav en tydelig lineær sammenhæng mellem de målte og de estimerede barrierelagstykkelser. Dog sås en markant afvigelse mellem de målte og de estimerede tykkelser. Problemet skyldes formentlig den indirekte måde lagets tykkelse er målt på, og da vi for de enkelte kiks kun har gennemsnittet at forholde os til er det problematisk at opstille en model der skal give et estimat af højden i de enkelte pixels i billedet.

Temperaturen af den påførte barriere har en væsentlig betydning, da denne influerer på i hvor høj grad barrieren trænger ned i kiksken, og dermed bliver ”usynlig” for kameraet. For høj indtrængen i kiksken betyder desuden at vandbarrieren mister sin effekt. Derfor vil en lille ændring i opstillingen der er skyld i en temperaturændring mellem de to seriers gennemkørsler kunne foresage at der ville være væsentlig forskel på det respons vi får fra kameraet, og det vi kan måle os frem til ved en vægtforøgelse, da vægtforøgelsen vil være den samme.

Et nyt forsøg med de ændringsforslag der gives i afsnittet om fremtidige forbedringer, afsnit 27.3, vil derfor være anbefalelsesværdigt, da man derved i højere grad vil kunne afgøre om modelopstillingen er god eller ej.

## 26.3 Vandindhold i brød

Det udviklede system er i stand til at beregne en række statistiske features for hvert enkelt billede og med LARS-EN regressionsanalyse finde de features der

bedst beskriver den lineære sammenhæng med vandindholdet målt med enten HB43- eller LP16-metoden.

Datasættet indeholder vandindholdsmålinger fra 35 til 44,5% af vægten for HB43 og fra 33,4 til 44,2% for LP16.

Vandindholdet kan estimeres for HB43 med en standardafvigelse på 0,71 procentpoint for testdata baseret på features af hele billedet. De bedste parametre for LARS-EN regressionsanalysen var maksimalt 100 variable, skåret ned til 9 aktive efter krydsvalidering og en  $\lambda$  på  $10^{-5}$ .

Den tilsvarende standardafvigelse for estimering af LP16 vandindhold er 1,18 med regressionsparametrene maksimalt 400 variable, 23 efter krydsvalidering og en  $\lambda$  på  $10^{-2}$ .

For de modeller baseret på features beregnet for hele billedet er større lufthuller i brødet formentlig en væsentlig variationsfaktor, både i tilfælde hvor det blot betyder et mørkere område og i nogle tilfælde endda synlig baggrund. Fjernes sidstnævnte fra området der benyttes til at udregne features kan det forbedre modellen.

Standardafvigelsen på estimering af HB43 kunne forbedres til 0,68 procentpoint ved at benytte den del af billedet svarende til de 38% nærmest snitfladen. Denne model havde 16 aktive variable efter krydsvalidering. For LP16 kunne standardafvigelsen på estimering af vandindholdet forbedres til 1,14 procentpoint ved at benytte features på den del af billedet svarende til de 40% nærmest snitfladen og med 21 aktive variable.

## KAPITEL 27

# Fremtidige forbedringer

---

I dette kapitel beskrives først de overordnede fremtidige forbedringer af hele systemet, og derefter vil det blive beskrevet hvilke forbedringer der er tiltænkt de enkelte fødevarer emner.

### 27.1 Generelt

Generelt ville det være en forbedring af softwaren at ligge et præsentationslag ovenpå det eksisterende i form af en grafisk brugerflade (GUI). Ved hjælp af et sådant lag ville det f.eks. være muligt at justere mere flydende på de enkelte parametre, og se ændringerne umiddelbart efter.

Oprensningen er lavet så den burde kunne bruges generelt på de fleste typer emner hvor der er en væsentlig mængde baggrund der skal fjernes, uanset om der er tale om fødevarer eller andre ting.

Systemet anvendt til kød- og fedtadskillelse i spegepølser indeholder en række funktionaliteter specifikt for denne fødevarer, men metoden til at benytte PCA eller MAF sammen med adaptiv thresholding til adskillelse af forskellige "klasser" i en fødevarer kan meget vel benyttes på andre produkter også.

Systemet anvendt til estimering af barrierelagstykkelse og systemet til estimering af vandindhold i brød benytter sig af hhv. lineær regression og LARS-EN regression til at opstille en model. Disse metoder bør kunne anvendes i tilfælde hvor man har et kendt respons for et datasæt og ønsker at opstille en model til estimering af denne. Resultatet vil afhænge af i hvor høj grad den ønskede egenskab er udtrykt i de bølgelængder der er tilgængelige i billederne, og hvilke features der benyttes.

Der er desuden opbygget et større funktionsbibliotek til generel eksplorativ analyse og præsentation af multispektrale billeder. Biblioteket kan anvendes i alle sammenhænge med multispektrale billeder, uafhængigt af det fotograferede emne.

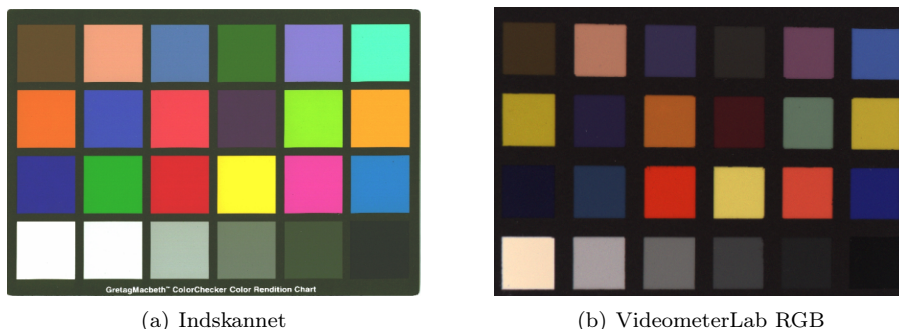
## 27.2 Spegepølser

Metoden til opdeling af kød og fedt virker robust, så længe pølsen ikke har en fedtkant. Hvis der er en fedtkant, skal denne fjernes først, ved at man skal angive hvor mange gange billedet skal eroderes. Det kunne derfor være nyttigt at udarbejde funktionalitet til at detektere og fjerne en eventuel fedtkant uden at tage for meget med.

Den metode der benyttes til at vise RGB-billeder, vælger tre bånd ud af billedet og viser dem som komponenterne rød, grøn og blå. Dette er en simplificeret og ikke særlig farveægte måde at vise RGB billeder på. På figur 27.1 kan ses et indskannet billede og et RGB-billede taget med VideometerLab'en af et *Gre-tagMacbeth ColorChecker* farvekort. Det indskannede kort ligner, med mindre afvigelser, farverne på det fysiske kort, mens RGB billedet fra VideometerLab'en i visse tilfælde er langt fra originalen. Man kan se en tydelig forskel på mange af farverne, og især på primær farverne i næstnederste række er der store forskelle at spore, og den grønne bliver nærmest blå.

Hvis en spegepølse har en meget brat overgang fra mørke til lyse intensiteter gående mod midten, kan det når spegepølsen deles op i ringe ud fra midten, resultere i at en ring kommer til at dække over en eventuel markant farveforskel indenfor et lille område af pølsen og dermed give et resultat der vil være gennemsnits intensiteten over dette stykke. En måde at omgå dette problem med den eksisterende software, ville være at opdele pølsen i flere ringe, og så efterfølgende summere op i passende inddelinger. En anden måde kunne være at udvide funktionaliteten til at brugeren selv kunne vælge hvor grænserne for ringene skulle ligge, f.eks. via museklik på billedet.





**Figur 27.1:** De to figurer viser henholdsvis et indskannet RGB billede og et RGB billede taget med VideometerLab'en af et Macbeth farvekort.

Eftersom vurderingen af om noget er fedt eller kød, langt hen ad vejen er en subjektiv vurdering for de områder hvor de to klasser overlapper, kunne man derfor forestille sig at man kunne indføre en margin for et måske-område, sådan at kantområderne omkring de enkelte fedtklatter ikke blev taget med i beregning af farven.

## 27.3 Barriersystem på kiks

Det kunne tænkes at give gode resultater at påføre barrierelag på andre emner end kiks. Dette kunne f.eks. være stykker af ensfarvet papir/pap eller metal skiver. På denne måde ville man undgå den naturlige variation ved kiksens "rist" mønster. Ud fra disse billeder kunne skabes en model der udtrykker hvordan responset fra barrierelaget alene ændrer sig med tykkelsen, og denne model kan så overføres til kiks eller lignende.

Man kunne også tænkes at få nogle brugbare resultater ved at fjerne kiksens struktur med f.eks. en snitflade analyse. Finde strukturen i kiks og fjerne dem fra billedet før analysen. Denne metode ville dog muligvis resultere i at man fjernede barrierelagets struktur, dvs. eventuelle huller, områder med mindre barrierelag eller lignende.

## 27.4 Vandindhold i brød

Med Tekstur Profil Analyse (TPA) bestemmes en række parametre baseret på en tryktest af brødet. Analyser svarende til dem der er benyttet i dette projekt til at estimere HB43 og LP16, kan forsøges benyttet til at generere modeller til estimering af TPA-parametre.

Det kan undersøges nærmere hvilke features der teoretisk burde have bedst korrelation til vandindholdet, så der benyttes et mere selektivt udvalgt featuresæt, evt. udvidet i forhold til det der er anvendt her.

Yderligere 1. ordens og 2. ordens gråtone statistikker kan undersøges for korrelation til vandindholdet.

For de fundne features kan det undersøges om sammenhængen med responset bedre kan beskrives som ikke-lineær.

# Litteratur

---

- [Breu et al., 1995] Breu, H., Gil, J., Kirkpatrick, D. og Werman, M. (1995). Linear time euclidean distance transform algorithms. *IEEE TPAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence*. URL [citeseer.ifi.unizh.ch/breu95linear.html](http://citeseer.ifi.unizh.ch/breu95linear.html).
- [Chen et al., 2004] Chen, Q., Yang, X. og Petriu, E. M. (2004). Watershed segmentation for binary images with different distance transforms. URL [http://www.discover.uottawa.ca/~qchen/my\\_papers/have04\\_published\\_version.pdf](http://www.discover.uottawa.ca/~qchen/my_papers/have04_published_version.pdf).
- [Clemmensen, 2006] Clemmensen, L. H. (2006). Estimation and classification through regression with variable selection amongst features extracted from multi-spectral images - estimation of moisture content in sand & identification of penicillium fungi. Specialeafhandling, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby. Supervised by Assoc. Prof. Bjarne Ersbøll, IMM. URL <http://www2.imm.dtu.dk/pubdb/p.php?4488>.
- [Danisco, 2005] Danisco (2005). Acetic acid esters (acetem). URL [http://www.danisco.com/cms/connect/corporate/products%20and%20services/product%20range/emulsifiers/acetic%20acid%20esters/acetic\\_acid\\_esters\\_en.htm](http://www.danisco.com/cms/connect/corporate/products%20and%20services/product%20range/emulsifiers/acetic%20acid%20esters/acetic_acid_esters_en.htm).
- [Efron et al., 2003] Efron, B., Hastie, T., Johnstone, I. og Tibshirani, R. (2003). Least angle regression. URL [http://www-stat.stanford.edu/~hastie/Papers/LARS/LeastAngle\\_2002.pdf](http://www-stat.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf).

- [Fisher et al., 2003] Fisher, R., Perkins, S., Walker, A. og Wolfart., E. (2003). Adaptive thresholding. URL <http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm>.
- [FPI, 2006] FPI (2006). The best possible taste. URL [http://www.fpi-international.com/articles/confectionary/FT1008\\_16\\_Danisco.pdf](http://www.fpi-international.com/articles/confectionary/FT1008_16_Danisco.pdf).
- [Geladi og Grahn, 1996] Geladi, P. og Grahn, H. (1996). *Multivariate image analysis*. Wiley, 1. udgave. ISBN 0-471-93001-6.
- [Gomez et al., 2006] Gomez, D. D., Clemmensen, L. H., Ersbøll, B. K. og Carstensen, J. M. (2006). Precise acquisition and unsupervised segmentation of multi-spectral images. Denne artikel er fremsendt til 'Elsevier Computer Vision and Image Understanding' men ikke publiceret endnu.
- [Hansen, 1999] Hansen, J. F. (1999). *On chromatic and geometrical calibration*. Ph.d.-afhandling, Department of Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby. URL <http://www2.imm.dtu.dk/pubdb/p.php?1210>.
- [Hastie et al., 2001] Hastie, T., Tibshirani, R. og Friedman, J. (2001). *The elements of Statistical Learning*. Springer, 1. udgave. ISBN 0-387-95284-5.
- [Hoerl og Kennard, 1970] Hoerl, A. E. og Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems.
- [IFI-Online, 2004] IFI-Online (2004). Fi awards: the winners in detail. URL [http://www.ifi-online.com/Tmpl\\_Article.asp?ContentID=295&ContentType=3](http://www.ifi-online.com/Tmpl_Article.asp?ContentID=295&ContentType=3).
- [Jensen, 2006] Jensen, A. T. (2006). Instrumental analysis of staling in bread, effects of enzymes and emulsifier. Specialeafhandling, KVL, Department of Food Science. Supervised by Prof. Søren Balling Engelsen (KVL), As. Prof. Flemming Hofmann Larsen (KVL) and Dr. Niels Barfod (Danisco).
- [Larsen, 2002] Larsen, R. (2002). Decomposition using maximum autocorrelation factors. *Journal of Chemometrics*, 16(8-10):427-435. URL <http://www2.imm.dtu.dk/pubdb/p.php?209>.
- [MathWorks, 2006] MathWorks, T. (2006). Matlab memory management guide. URL <http://www.mathworks.com/support/tech-notes/1100/1106.html>.
- [Nielsen, 1999] Nielsen, A. A. (1999). Orthogonal transformation.
- [Sjöstrand, 2006] Sjöstrand, K. (2006). Karl sjöstrand - matlab implementering af lars-en. URL <http://www2.imm.dtu.dk/~kas/>.

- [Tibshirani, 1994] Tibshirani, R. (1994). Regression shrinkage and selection via the lasso. URL [citeseer.ist.psu.edu/tibshirani94regression.html](http://citeseer.ist.psu.edu/tibshirani94regression.html).
- [Vincent, 1991] Vincent, L. (1991). Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, Vol. 2, No. 2, pp. 176-201, April 1993.
- [Wikipedia, 2006a] Wikipedia (2006a). Wikipedia - image:full width half maximum. URL [http://en.wikipedia.org/wiki/Image:Full\\_width\\_at\\_half\\_maximum.png](http://en.wikipedia.org/wiki/Image:Full_width_at_half_maximum.png).
- [Wikipedia, 2006b] Wikipedia (2006b). Wikipedia - singular value decomposition. URL [http://en.wikipedia.org/wiki/Singular\\_value\\_decomposition](http://en.wikipedia.org/wiki/Singular_value_decomposition).
- [Zou og Hastie, 2005] Zou, H. og Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(5):768-768. ISSN 1369-7412. URL <http://www-stat.stanford.edu/~hastie/Papers/>.



# Figurer

---

3.1	Udvikling af spegepølsens pH-værdi som funktion af tid . . . . .	14
3.2	Nuværende testmetode af vandbarriere til kiks . . . . .	16
3.3	Resultatet af et forsøg med barrierelag som vist på figur 3.2 . . .	17
4.1	Det menneskelige syn snyder når forskellige farver ligger tæt op af hinanden . . . . .	20
5.1	VideometerLab kameraet . . . . .	24
5.2	Skygger og refleksion ved forskellig belysning . . . . .	24
5.3	Tegning af et tværsnit af VideometerLab med kamera og lyssætning . . . . .	25
6.1	Forsøgsopstilling med forsøgsleder Casper Lund fra Danisco og Jesper Rasmussen . . . . .	28
6.2	Påførelse af barrieresystem på Mariekiks . . . . .	28
6.3	De to forsøgsassistenter i arbejde og godt pakket ind . . . . .	30

---

8.1	Principtegning der viser en spegepølses kødfarveintensitet fra yderst til inderst . . . . .	38
8.2	Principtegning der viser højdeforskelle for barriereag på en kiks	38
12.1	Repræsentationer af et multispektralt billede . . . . .	50
12.2	Kød- og fedtspektre for to udvalgte pixels . . . . .	51
12.3	PCA ordner ikke nødvendigvis komponenterne så signal kommer før støj . . . . .	55
12.4	PCA og MAF analyse, med og uden spegepølsens fedtkant . . .	57
12.5	Falsk farvesammensætning . . . . .	59
12.6	Densitetsbillede metode . . . . .	60
12.7	ROI-generering ud fra densitetsplot . . . . .	61
12.8	Eksplorativ scoreplotanalyse . . . . .	61
12.9	Residualbilleder for 4 PCA-komponenter . . . . .	63
12.10	Farvekodede residualbilleder . . . . .	64
13.1	K-fold krydsvalidering . . . . .	68
13.2	Sammenhæng mellem regressionsmetoder . . . . .	69
13.3	Multivariat regression . . . . .	74
13.4	Regressionsanalyse af spegepølse . . . . .	75
13.5	Overførsel af regressionsparametre til anden pølse . . . . .	75
14.1	Histogram for spegepølsebillede bånd 13 . . . . .	85
14.2	Full width at half maximum . . . . .	86
14.3	Histogram med minima, maksima og FWHM indtegnet . . . . .	87



---

14.4	Orensning af ROI . . . . .	88
14.5	Fjerne pixels uden for ROI . . . . .	90
14.6	Resultat af preprocessering . . . . .	91
14.7	Orensningen af et brødbillede . . . . .	92
15.1	Bånd 1-9 for en spegepølse . . . . .	94
15.2	Bånd 10-18 for en spegepølse . . . . .	95
15.3	Histogrammer for udvalgte bånd af en spegepølse . . . . .	95
15.4	RGB billede af <i>sample 2</i> med fedtkant . . . . .	96
15.5	Fjernelse af fedtkant . . . . .	96
15.6	Gennemsnitsvarians for alle samples i hvert bånd overfor graden af polynomiet der er brugt til at fitte med . . . . .	97
15.7	Intensitets niveau udligning med et 7. grads polynomie vist ved bånd 7 i spegepølsen . . . . .	98
15.8	Intensitets niveau udligning af spegepølse . . . . .	99
15.9	Farveintensitets opretning for bånd 7 . . . . .	100
15.10	Farveintensitets opretning af pølse billede . . . . .	101
15.11	Kød detektering med og uden farveintensitets opretning . . . . .	102
15.12	PCA komponenter for en spegepølse . . . . .	104
15.13	PCA komponent 1-3 for spegepølse <i>sample 2</i> uden fedtkant . . . . .	104
15.14	Farvekodede PCA-komponenter . . . . .	105
15.15	MAF bånd 1-3 for spegepølse <i>sample 1</i> . . . . .	106
15.16	MAF komponent 1-3 for spegepølse <i>sample 3</i> . . . . .	106
15.17	Udvælgelse af egenskabs-ROI . . . . .	108

---

15.18	Lineær regression på <i>sample 1</i> og <i>sample 4</i> . . . . .	112
15.19	Lineær regression på <i>sample 2</i> og <i>sample 4</i> . . . . .	113
15.20	RGB plot af <i>sample 1-5</i> . . . . .	115
15.21	PCA plot af RGB, <i>sample 1-5</i> . . . . .	116
15.22	Histogrammer for PCA analysen af RGB-bånd . . . . .	117
15.23	Farvevariation ved komponent 3 . . . . .	118
15.24	Farveskalakomponenter for <i>sample 1</i> . . . . .	118
15.25	Totalt farvekort for spegepølser . . . . .	120
15.26	Zoomet farvekort for spegepølser . . . . .	120
15.27	RGB intensiteter ved variation af komponent 1 og 2 . . . . .	121
15.28	Fedt fra <i>sample 1</i> , ikke segmenteret . . . . .	122
15.29	Fedt-ROI og Euclidean Distance Map . . . . .	122
15.30	Watershed med og uden rekonstruktion . . . . .	123
15.31	H-domes reconstruction . . . . .	123
15.32	Farvebillede af <i>sample 1</i> . . . . .	124
15.33	Kød- og fedtopdelingen af <i>sample 1</i> . . . . .	125
15.34	Gennemsnitsintensitet for kødet fundet i <i>sample 1</i> . . . . .	125
15.35	Gennemsnitsintensitet for fedtet fundet i <i>sample 1</i> . . . . .	125
15.36	Farvekort med gennemsnitsintensiteter for <i>sample 1</i> . . . . .	126
15.37	Størrelsesfordelingen af fedtområderne i <i>sample 1</i> . . . . .	126
15.38	Farveskala for spegepølse med sensorik-kommentarer . . . . .	128
15.39	Histogrammer for RGB af hvidt papir . . . . .	129

---

15.40	Største og mindste overlap af fedt- og kødfarver for <i>sample 1</i> . . .	132
16.1	Gennemsnitsintensiteter for <i>Mariekiks</i> . . . . .	138
16.2	Lineær regressionsmodel til barrierelagstykkelse . . . . .	141
16.3	Model lavet fra <i>serie 2</i> , overført på vaffel kiks . . . . .	143
16.4	Højdekort for barrierelag af <i>Mariekiks serie 1</i> . Modellen er genereret ud fra <i>serie 1</i> . . . . .	145
16.5	Højdekort for barrierelag af <i>Mariekiks serie 2</i> . Modellen er genereret ud fra <i>serie 1</i> . . . . .	146
16.6	Højdekort for barrierelag af <i>Vaffel kiks</i> . Modellen er genereret ud fra <i>serie 2</i> . . . . .	147
16.7	Højdekort for barrierelag af <i>Mariekiks med fejl</i> . Modellen er genereret ud fra <i>serie 1</i> . . . . .	148
17.1	Brød 1, dag 1 og 15, forskellige skiver . . . . .	152
17.2	Bånd 1-9 for brød 1, dag 1 . . . . .	153
17.3	Bånd 10-18 for brød 1, dag 1 . . . . .	154
17.4	HB43 vandindholdsmålinger, gennemsnit og standardafvigelse .	154
17.5	LP16 vandindholdsmålinger, gennemsnit og standardafvigelse .	155
17.6	Standardafvigelser for forskellige kombinationer af stop-kriteriet og $\lambda$ . . . . .	158
17.7	Standardafvigelser og korrelationer for de bedste modeller til estimat af HB43 ved forskellige stop-kriterier . . . . .	159
17.8	Standardafvigelser og korrelationer for de bedste modeller til estimat af log(HB43) ved forskellige stop-kriterier . . . . .	159
17.9	Standardafvigelser og korrelationer for de bedste modeller til estimat af LP16 ved forskellige stop-kriterier . . . . .	160

---

17.10	Standardafvigelser og korrelationer for de bedste modeller til estimat af log(LP16) ved forskellige stop-kriterier . . . . .	160
17.11	Filter til bestemmelse af en brødskives negative krumning . . . . .	162
17.12	Række- og søljevis negativ krumning og et kombineret maksimum for en brødskive . . . . .	162
17.13	Eksempel på parameterudvikling for en LARS-EN regression . . . . .	164
17.14	HB43 estimater og residualer af den bedste model . . . . .	165
17.15	log(HB43) estimater og residualer af den bedste model . . . . .	165
17.16	LP16 estimater og residualer af den bedste model . . . . .	166
17.17	log(LP16) estimater og residualer af den bedste model . . . . .	166
17.18	Standardafvigelser for de forskellige modeller sorteret efter antal dage . . . . .	167
17.19	HB43 standardafvigelser for vandindholdsestimat på trænings- og testdata for forskellige fraktiler . . . . .	171
17.20	LP16 standardafvigelser for vandindholdsestimat på trænings- og testdata for forskellige fraktiler . . . . .	171
17.21	HB43 estimater og residualer af den bedste fraktil-model . . . . .	172
17.22	LP16 estimater og residualer af den bedste fraktil-model . . . . .	173
17.23	Et hul i en brødskive med og uden ROI svarende til den 38. fraktil af krumningsbilledet . . . . .	173
18.1	Overordnet flowchart for salami og barrieresystem . . . . .	180
18.2	Flowchart for pre-analysen . . . . .	180
18.3	Flowchart for preprocesseringen . . . . .	182
19.1	Flowchart for salamianalysen, del 1 . . . . .	185

---

19.2	Flowchart for salamianalysen, del 2 . . . . .	186
20.1	Strukturformat for <i>Responselist</i> . . . . .	187
20.2	Funktionalitet til at indlæse et rått billede, rense og gemme det eller indlæse et allerede rensset billede . . . . .	188
20.3	Overordnet flowstruktur for barrieresystem analysen . . . . .	189
20.4	Generering af lineær regressionsmodel ud fra en K-fold krydsvalidering . . . . .	191
20.5	Sammensat respons- og testdata for vandbarriere på kiks . . . . .	192
20.6	Anvendelse af den lineære regressionsmodel . . . . .	192
21.1	Overordnet flowstruktur for analysen af vandindhold i brød . . . . .	194
21.2	Flowstruktur for preanalysen af vandindhold i brød . . . . .	194
21.3	Flowstruktur for modelgenereringen af vandindhold i brød . . . . .	195
21.4	Flowstruktur for modeltest til estimering vandindhold i brød . . . . .	196
22.1	CD filstruktur . . . . .	201
27.1	Farveægthed for RGB-billeder fra VideometerLab . . . . .	219
B.1	Bånd 1-9 for spegepølse <i>sample 1</i> . . . . .	246
B.2	Bånd 10-18 for spegepølse <i>sample 1</i> . . . . .	246
B.3	Bånd 1-9 for spegepølse <i>sample 2</i> . . . . .	247
B.4	Bånd 10-18 for spegepølse <i>sample 2</i> . . . . .	247
B.5	Bånd 1-9 for spegepølse <i>sample 3</i> . . . . .	248
B.6	Bånd 10-18 for spegepølse <i>sample 3</i> . . . . .	248

---

B.7	Bånd 1-9 for spegepølse <i>sample 4</i> . . . . .	249
B.8	Bånd 10-18 for spegepølse <i>sample 4</i> . . . . .	249
B.9	Bånd 1-9 for spegepølse <i>sample 5</i> . . . . .	250
B.10	Bånd 10-18 for spegepølse <i>sample 5</i> . . . . .	250
C.1	PCA komponent 1-3 for spegepølse <i>sample 1</i> . . . . .	251
C.2	PCA komponent 1-3 for spegepølse <i>sample 2</i> med fedtkant . . .	251
C.3	PCA komponent 1-3 for spegepølse <i>sample 2</i> uden fedtkant . .	252
C.4	PCA komponent 1-3 for spegepølse <i>sample 3</i> . . . . .	252
C.5	PCA komponent 1-3 for spegepølse <i>sample 4</i> . . . . .	252
C.6	PCA komponent 1-3 for spegepølse <i>sample 5</i> . . . . .	252
D.1	MAF komponent 1-3 for spegepølse <i>sample 1</i> . . . . .	253
D.2	MAF komponent 1-3 for spegepølse <i>sample 2</i> med fedtkant . . .	253
D.3	MAF komponent 1-3 for spegepølse <i>sample 2</i> uden fedtkant . .	254
D.4	MAF komponent 1-3 for spegepølse <i>sample 3</i> . . . . .	254
D.5	MAF komponent 1-3 for spegepølse <i>sample 4</i> . . . . .	254
D.6	MAF komponent 1-3 for spegepølse <i>sample 5</i> . . . . .	254
E.1	Farvebillede af <i>sample 1</i> . . . . .	256
E.2	Kød- og fedtopdelingen af <i>sample 1</i> . . . . .	256
E.3	Gennemsnitsintensitet for kødet fundet i <i>sample 1</i> . . . . .	257
E.4	Gennemsnitsintensitet for fedtet fundet i <i>sample 1</i> . . . . .	257
E.5	Farvekort med gennemsnitsintensiteter for <i>sample 1</i> . . . . .	257

---

E.6	Størrelsesfordelingen af fedtområderne i <i>sample 1</i> . . . . .	258
E.7	Farvebillede af <i>sample 2</i> . . . . .	259
E.8	Kød- og fedtopdelingen af <i>sample 2</i> . . . . .	260
E.9	Gennemsnitsintensitet for kødet fundet i <i>sample 2</i> . . . . .	260
E.10	Gennemsnitsintensitet for fedtet fundet i <i>sample 2</i> . . . . .	260
E.11	Farvekort med gennemsnitsintensiteter for <i>sample 2</i> . . . . .	261
E.12	Størrelsesfordelingen af fedtområderne i <i>sample 2</i> . . . . .	261
E.13	Farvebillede af <i>sample 3</i> . . . . .	262
E.14	Kød- og fedtopdelingen af <i>sample 3</i> . . . . .	263
E.15	Gennemsnitsintensitet for kødet fundet i <i>sample 3</i> . . . . .	263
E.16	Gennemsnitsintensitet for fedtet fundet i <i>sample 3</i> . . . . .	263
E.17	Farvekort med gennemsnitsintensiteter for <i>sample 3</i> . . . . .	264
E.18	Størrelsesfordelingen af fedtområderne i <i>sample 3</i> . . . . .	264
E.19	Farvebillede af <i>sample 4</i> . . . . .	265
E.20	Kød- og fedtopdelingen af <i>sample 4</i> . . . . .	266
E.21	Gennemsnitsintensitet for kødet fundet i <i>sample 4</i> . . . . .	266
E.22	Gennemsnitsintensitet for fedtet fundet i <i>sample 4</i> . . . . .	266
E.23	Farvekort med gennemsnitsintensiteter for <i>sample 4</i> . . . . .	267
E.24	Størrelsesfordelingen af fedtområderne i <i>sample 4</i> . . . . .	267
E.25	Farvebillede af <i>sample 5</i> . . . . .	268
E.26	Kød- og fedtopdelingen af <i>sample 5</i> . . . . .	269
E.27	Gennemsnitsintensitet for kødet fundet i <i>sample 5</i> . . . . .	269

---

E.28	Gennemsnitsintensitet for fedtet fundet i <i>sample 5</i> . . . . .	269
E.29	Farvekort med gennemsnitsintensiteter for <i>sample 5</i> . . . . .	270
E.30	Størrelsesfordelingen af fedtområderne i <i>sample 5</i> . . . . .	270
F.1	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv vinduesstørrelse 25 . . .	272
F.2	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv vinduesstørrelse 50 . . .	272
F.3	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv vinduesstørrelse 75 . . .	272
F.4	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv vinduesstørrelse 100 . .	272
F.5	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv vinduesstørrelse 125 . .	273
F.6	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv vinduesstørrelse 150 . .	273
F.7	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv vinduesstørrelse 175 . .	273
F.8	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv vinduesstørrelse 200 . .	273
F.9	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv konstant -0,04 . . . . .	274
F.10	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv konstant -0,02 . . . . .	274
F.11	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv konstant 0,00 . . . . .	275
F.12	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv konstant 0,02 . . . . .	275
F.13	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv konstant 0,04 . . . . .	275
F.14	<i>Sample 1</i> , kød/fedt adskillelse, adaptiv konstant 0,06 . . . . .	275
H.1	<i>Mariekiks serie 1</i> . . . . .	282
H.2	<i>Mariekiks serie 2</i> . . . . .	283
H.3	<i>Vaffel kiks</i> . . . . .	284
H.4	<i>Mariekiks med fejl</i> . . . . .	285



---

H.5	<i>Mariekiks serie 1 - Marie 1</i> Bånd 1 til 9 . . . . .	286
H.6	<i>Mariekiks serie 1 - Marie 1</i> Bånd 10 til 18 . . . . .	287
H.7	<i>Mariekiks serie 1 - Marie 5</i> Bånd 1 til 9 . . . . .	288
H.8	<i>Mariekiks serie 1 - Marie 5</i> Bånd 10 til 18 . . . . .	289
I.1	Standardafvigelse og korrelation for HB43-estimat med hhv. max 10, 20 og 40 variable og varierende $\lambda$ . . . . .	292
I.2	Standardafvigelse og korrelation for HB43-estimat hhv. max 100, 200 og 400 variable og varierende $\lambda$ . . . . .	293
I.3	Standardafvigelse og korrelation for log(HB43)-estimat med hhv. max 10, 20 og 40 variable og varierende $\lambda$ . . . . .	294
I.4	Standardafvigelse og korrelation for log(HB43)-estimat med hhv. max 100, 200 og 400 variable og varierende $\lambda$ . . . . .	295
I.5	Standardafvigelse og korrelation for LP16-estimat med hhv. max 10, 20 og 40 variable og varierende $\lambda$ . . . . .	296
I.6	Standardafvigelse og korrelation for LP16-estimat hhv. max 100, 200 og 400 variable og varierende $\lambda$ . . . . .	297
I.7	Standardafvigelse og korrelation for log(LP16)-estimat med hhv. max 10, 20 og 40 variable og varierende $\lambda$ . . . . .	298
I.8	Standardafvigelse og korrelation for log(LP16)-estimat med hhv. max 100, 200 og 400 variable og varierende $\lambda$ . . . . .	299



# Tabeller

---

12.1	Varians for PCA komponent 7 og 8 . . . . .	55
12.2	Variationskoefficienter (CV) for spegepølse <i>sample 2</i> , med og uden fedtkant . . . . .	58
12.3	Farvekodningsskema for residualbillede . . . . .	63
14.1	Gennemsnit, minimum og maximum for scoreværdier før og efter preprocessing . . . . .	92
15.1	FOGL statistikker før og efter intensitetsniveauudligningen af et billede . . . . .	100
15.2	Farvekodningsskema benyttet til PCA-komponent billeder . . . . .	101
15.3	Gennemsnitsintensiteter fedt/kød fordelinger i et PCA score-billede af <i>sample 1</i> . . . . .	104
15.4	Gennemsnitsintensiteter af normaliserede fedt/kød fordelinger i et PCA score-billede for <i>sample 2</i> . . . . .	105
15.5	MSE og korrelation (corr.) af lineær regression for <i>sample 1-5</i> . . . . .	111
15.6	MSE og korrelation (corr.) af logistisk regression for <i>sample 1-5</i> . . . . .	111

15.7	MSE og korrelation (corr.) af den lineære model for $K=1$ overført på <i>sample 1-5</i> . . . . .	111
15.8	MSE og korrelation (corr.) af den logistiske model for $K=1$ overført på <i>sample 1-5</i> . . . . .	111
15.9	Eigenvektorer for farveskalakonvertering . . . . .	116
15.10	Minima og maksima for farveskala konvertering før skalering . . . . .	116
15.11	Farveværdier for kødet i de fem spegepølsesamples . . . . .	127
15.12	Farveværdier for fedtet i de fem spegepølsesamples . . . . .	127
15.13	Subjektive betegnelser for de 5 spegepølsesamples . . . . .	128
15.14	Variationskoefficienter for hhv. PCA og MAF . . . . .	130
15.15	Procentvist overlap af de to populationer af kød og fedt . . . . .	131
16.1	Barrierelag for Mariekiks model-datasæt for <i>serie 1</i> . . . . .	137
16.2	Barrierelag for Mariekiks model-datasæt for <i>serie 2</i> . . . . .	137
16.3	Barrierelag for vaffel kiks . . . . .	137
16.4	Barrierelag for Mariekiks med pålagt mønt . . . . .	137
16.5	MSE-værdier for modeller af kiks <i>serie 1</i> og <i>2</i> . . . . .	140
16.6	Standard afvigelser for modeller af kiks <i>serie 1</i> og <i>2</i> . . . . .	140
16.7	Gennemsnitsværdier for estimerede og målte tykkelser . . . . .	142
16.8	Standardafvigelser for de to modeller . . . . .	142
16.9	Målte og estimerede værdier for <i>vaffler</i> , med model genereret ud fra <i>Mariekiks serie 2</i> . . . . .	142
17.1	Brøddataenes undergrupper . . . . .	152
17.2	Gennemsnit og standardafvigelse for HB43-målinger af brøddata	153

---

17.3	Gennemsnit og standardafvigelse for LP16-målinger af brøddata	153
17.4	Modelresultater for estimering af vandindhold ud fra features af hele billedet . . . . .	163
17.5	Standardafvigelserne for modellerne fordelt på brødets alder . .	165
17.6	De udvalgte features til estimat af vandindhold (HB43) . . . . .	168
17.7	De udvalgte features til estimat af vandindhold (LP16) . . . . .	169
17.8	Modelresultater for estimering af vandindhold ud fra features af en fraktil af billedet . . . . .	170
A.1	De 18 bånd som VideometerLab er i stand til at måle lysintensitet ved . . . . .	244
A.2	Forskellige data for VideometerLab . . . . .	244



Del VIII

Bilag





BILAG A

# VideometerLab

---

Bånd	Område	Bølgelængde (nm)
1	Violet	395
2	Ultra-blå	430
3	Blå	450
4	Blå(B)	470
5	Grøn(G)	505
6	Grøn	565
7	Orange	590
8	Rød(R)	630
9	Rød	645
10	Rød	660
11	Rød	700
12	NIR	850
13	NIR	870
14	NIR	890
15	NIR	910
16	NIR	940
17	NIR	950
18	NIR	970

**Tabel A.1:** De 18 bånd som VideometerLab er i stand til at måle lysintensitet ved. De bølgelængder der er benyttet til RGB farver er markeret med hhv. R, G og B.

Egenskab	
Ca. pixelstørrelse	0,077 mm/px
Billedstørrelse	1280 × 960 pixels

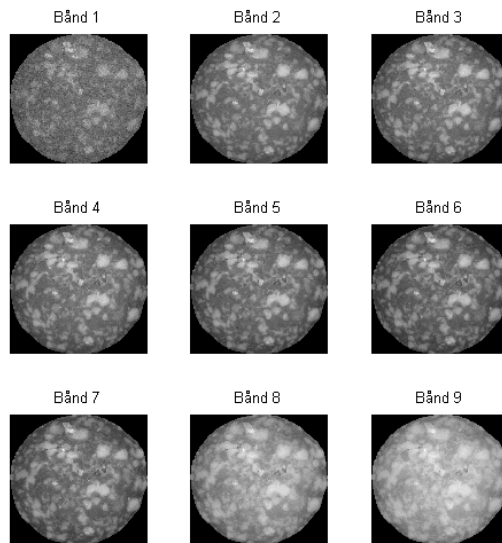
**Tabel A.2:** Forskellige data for VideometerLab.

## BILAG B

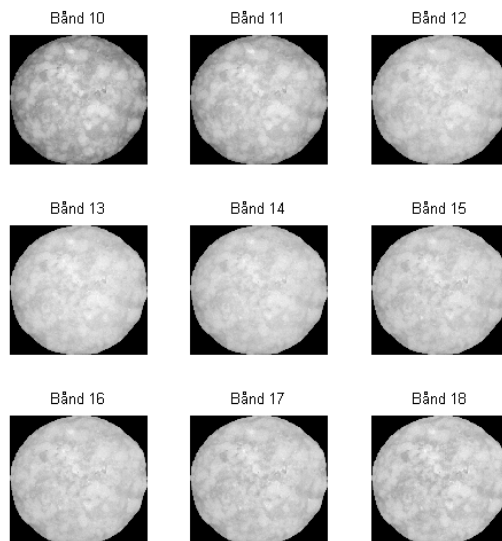
# Spegepølser - rådata

---

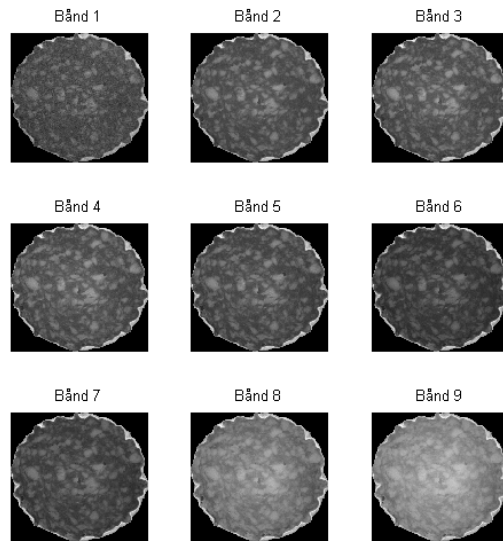
Dette bilag indeholder alle billederne fra *spegepølse* datasættet. Billederne er først blevet renset som beskrevet i kapitel 14.



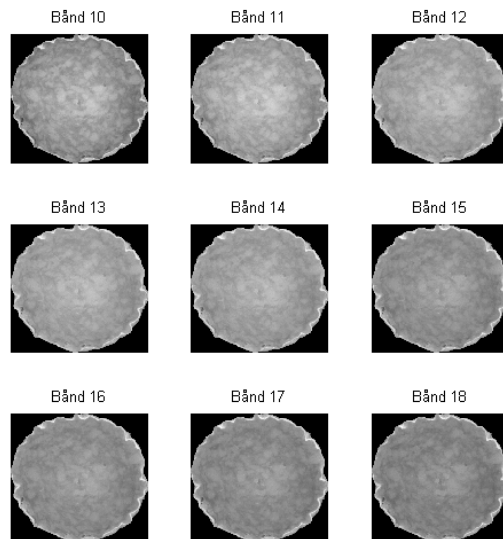
**Figur B.1:** Bånd 1-9 for spegepølse *sample 1*.



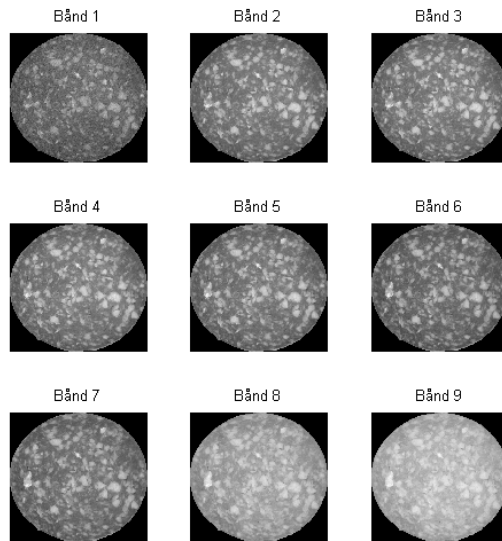
**Figur B.2:** Bånd 10-18 for spegepølse *sample 1*.



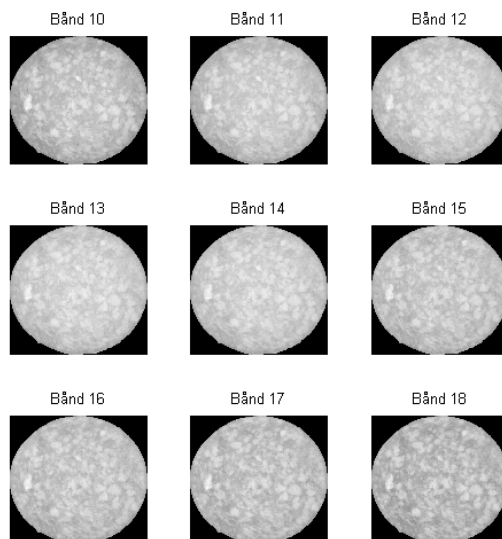
**Figur B.3:** Bånd 1-9 for spegepølse *sample 2*.



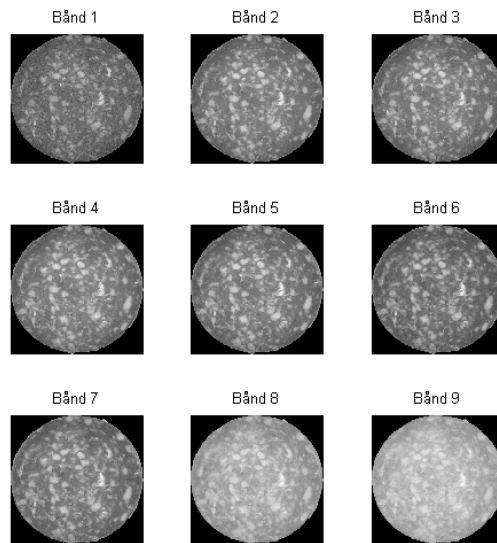
**Figur B.4:** Bånd 10-18 for spegepølse *sample 2*.



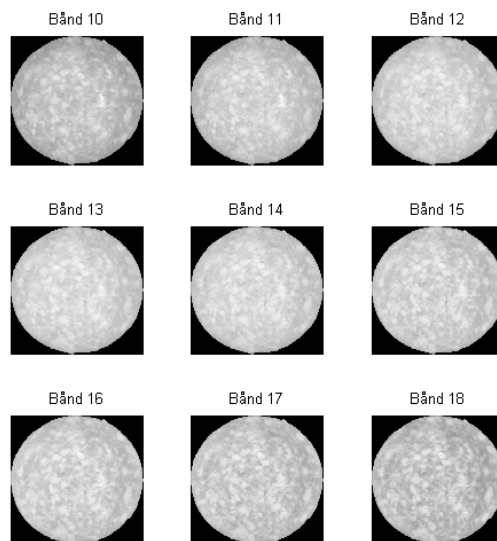
**Figur B.5:** Bånd 1-9 for spegepølse *sample 3*.



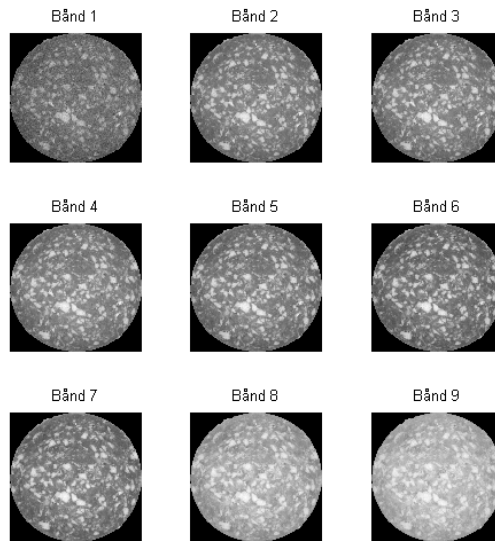
**Figur B.6:** Bånd 10-18 for spegepølse *sample 3*.



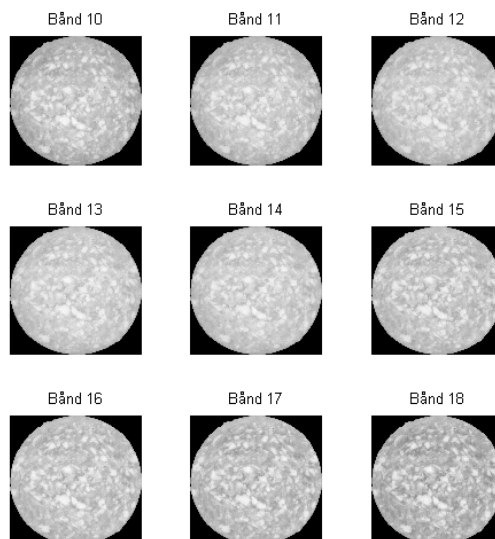
**Figur B.7:** Bånd 1-9 for spegepølse *sample 4*.



**Figur B.8:** Bånd 10-18 for spegepølse *sample 4*.



**Figur B.9:** Bånd 1-9 for spegepølse *sample 5*.

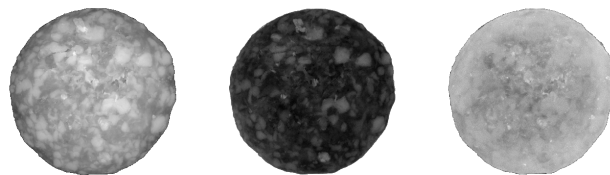


**Figur B.10:** Bånd 10-18 for spegepølse *sample 5*.

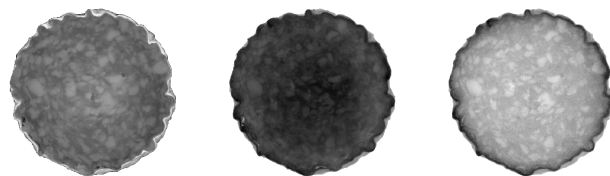


## Spegepølser - PCA

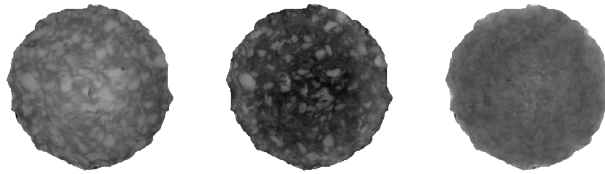
---



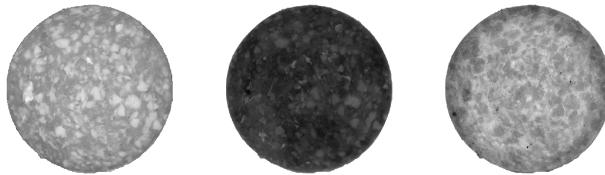
**Figur C.1:** PCA komponent 1-3 for spegepølse *sample 1*.



**Figur C.2:** PCA komponent 1-3 for spegepølse *sample 2* med fedtkant.



**Figur C.3:** PCA komponent 1-3 for spegepølse *sample 2* uden fedtkant.



**Figur C.4:** PCA komponent 1-3 for spegepølse *sample 3*.



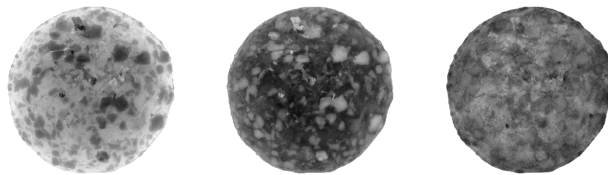
**Figur C.5:** PCA komponent 1-3 for spegepølse *sample 4*.



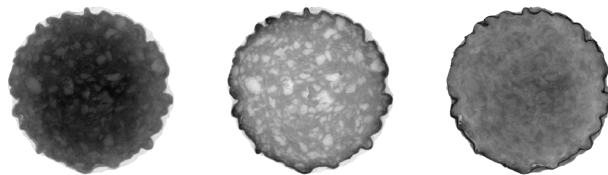
**Figur C.6:** PCA komponent 1-3 for spegepølse *sample 5*.

## Spegepølser - MAF

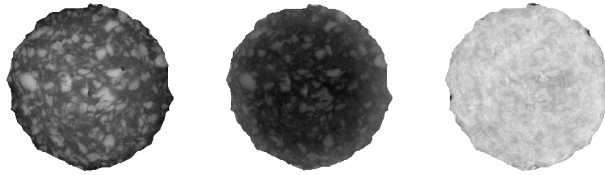
---



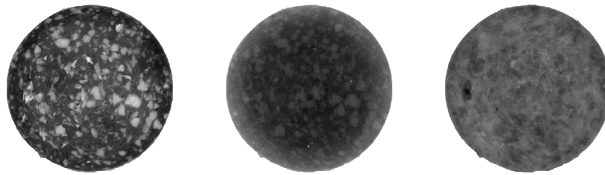
**Figur D.1:** MAF komponent 1-3 for spegepølse *sample 1*.



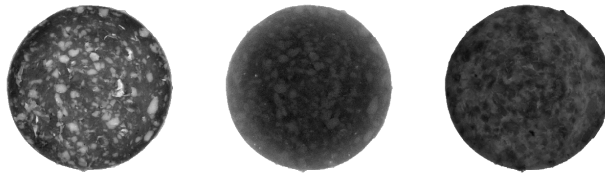
**Figur D.2:** MAF komponent 1-3 for spegepølse *sample 2* med fedtkant.



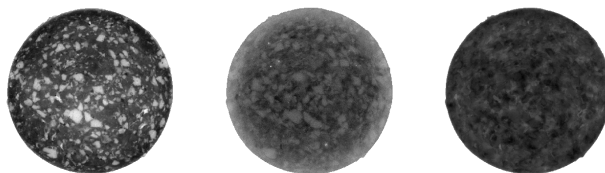
**Figur D.3:** MAF komponent 1-3 for spegepølse *sample 2* uden fedtkant.



**Figur D.4:** MAF komponent 1-3 for spegepølse *sample 3*.



**Figur D.5:** MAF komponent 1-3 for spegepølse *sample 4*.



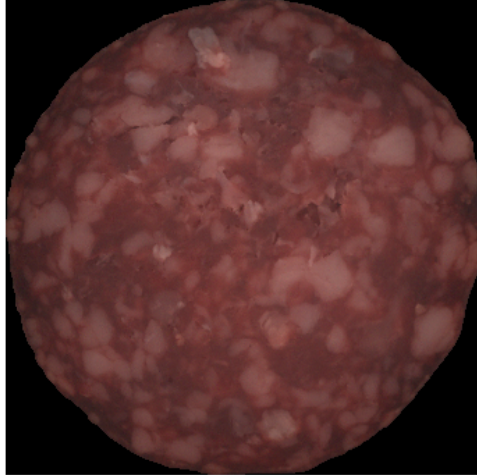
**Figur D.6:** MAF komponent 1-3 for spegepølse *sample 5*.

BILAG E

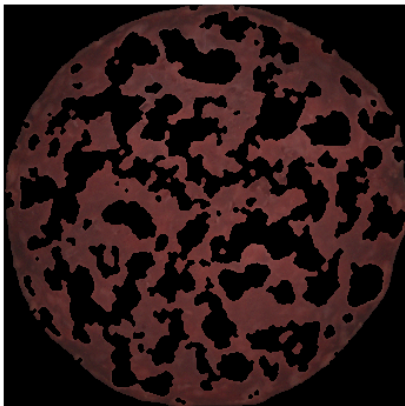
# Spegepølser - Resultater

---

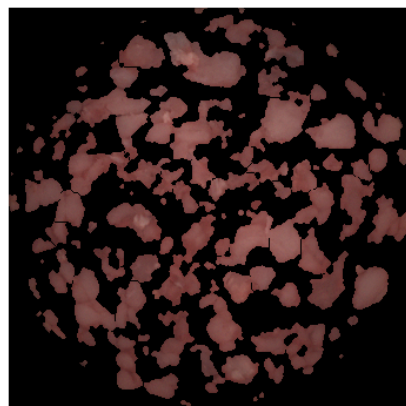
E.1 Sample 1



Figur E.1: Farvebillede af *sample 1*.

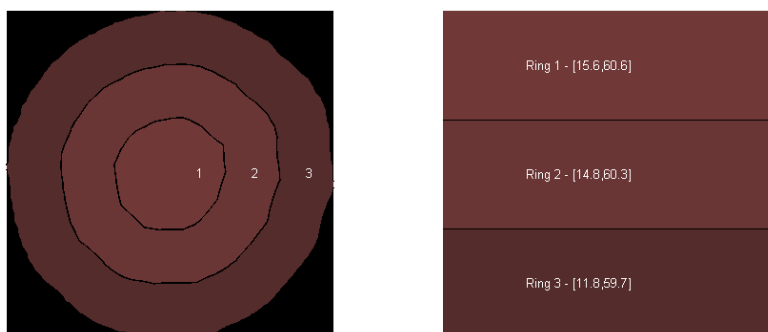


(a) Kødets fundet i *sample 1*

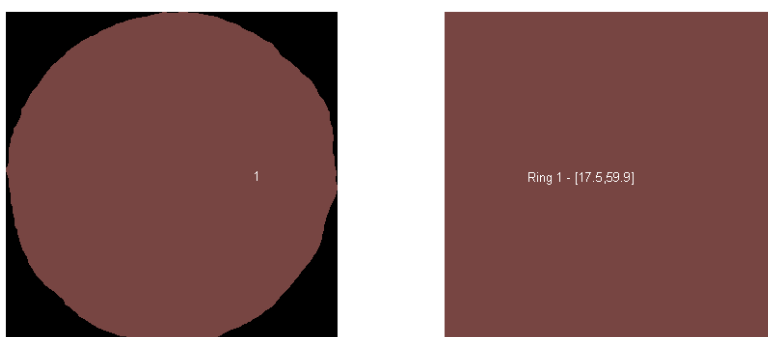


(b) Fedtet fundet i *sample 1*

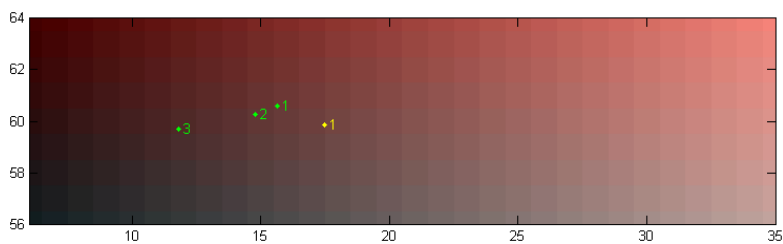
Figur E.2: Kød- og fedtopdelingen af *sample 1*.



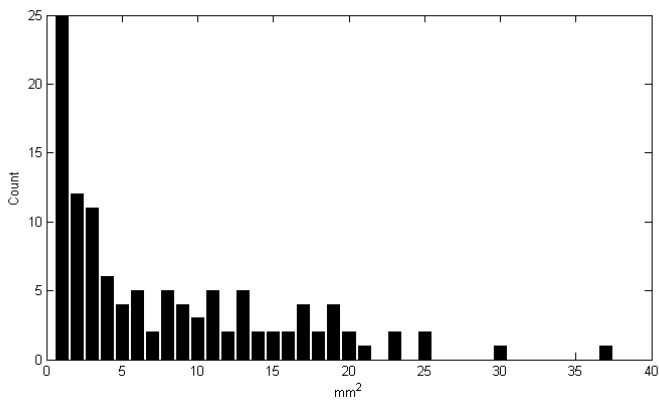
**Figur E.3:** Gennemsnitsintensitet for kødet fundet i *sample 1*.



**Figur E.4:** Gennemsnitsintensitet for fedtet fundet i *sample 1*.



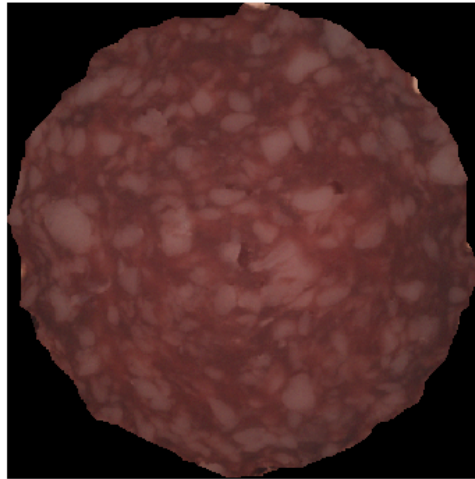
**Figur E.5:** Farvekort med gennemsnitsintensiteter for *sample 1*. Kødringenes farver er angivet med grønt og fedtringenes farver med gult.



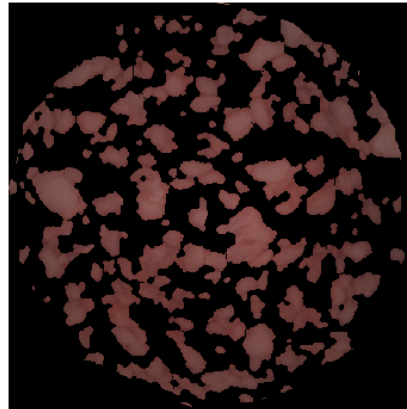
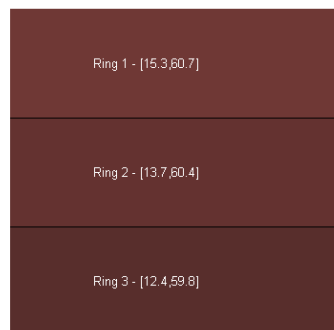
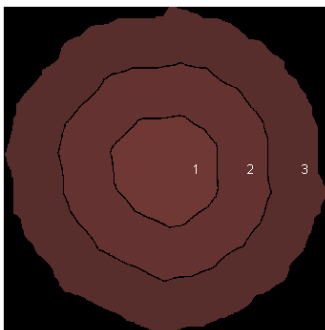
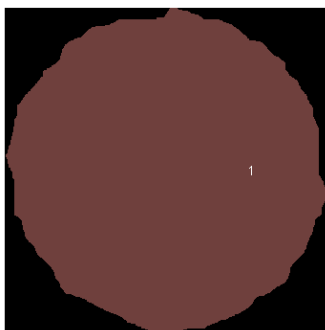
**Figur E.6:** Størrelsesfordelingen af fedtområderne i *sample 1*.

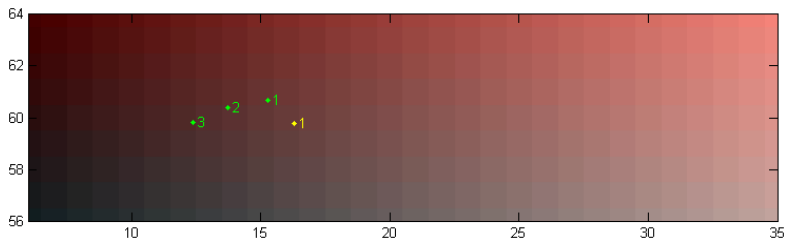


## E.2 Sample 2

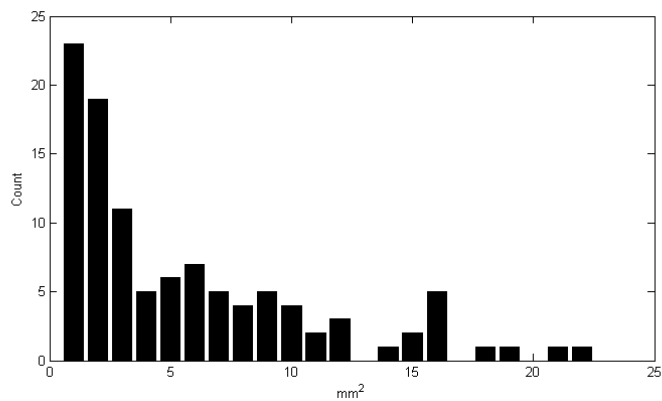


**Figur E.7:** Farvebillede af *sample 2*.

(a) Kødets fundet i *sample 2*(b) Fedtet fundet i *sample 2***Figur E.8:** Kød- og fedtopdelingen af *sample 2*.**Figur E.9:** Gennemsnitsintensitet for kødet fundet i *sample 2*.**Figur E.10:** Gennemsnitsintensitet for fedtet fundet i *sample 2*.

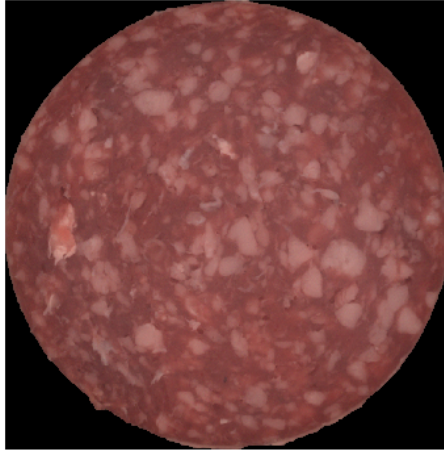


**Figur E.11:** Farvekort med gennemsnitsintensiteter for *sample 2*. Kødringenes farver er angivet med grønt og fedtringenes farver med gult.

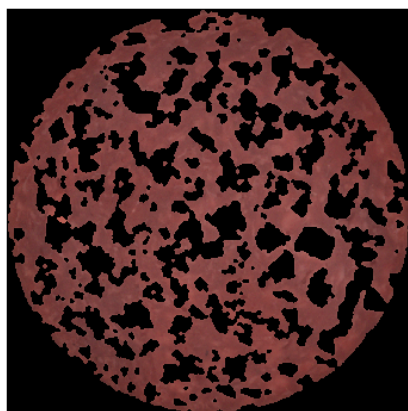
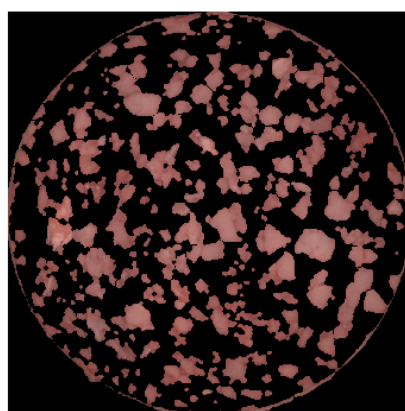
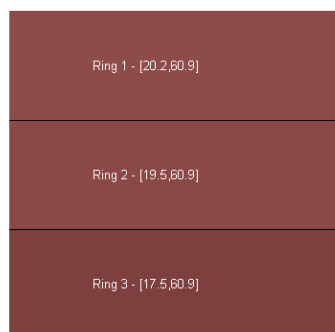
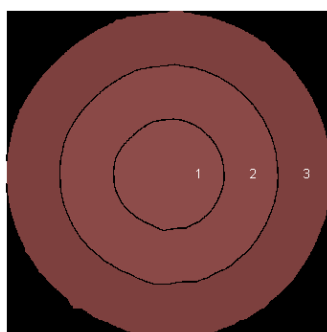
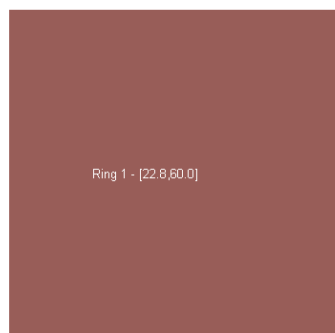
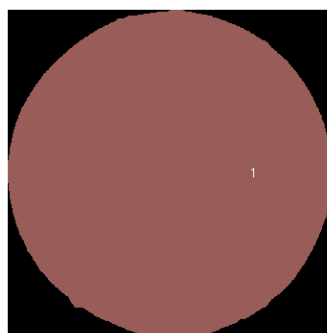


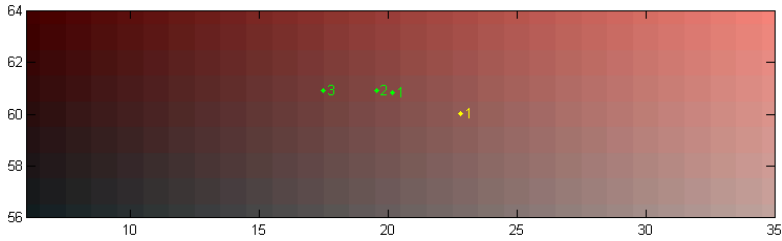
**Figur E.12:** Størrelsesfordelingen af fedtområderne i *sample 2*.

### E.3 Sample 3

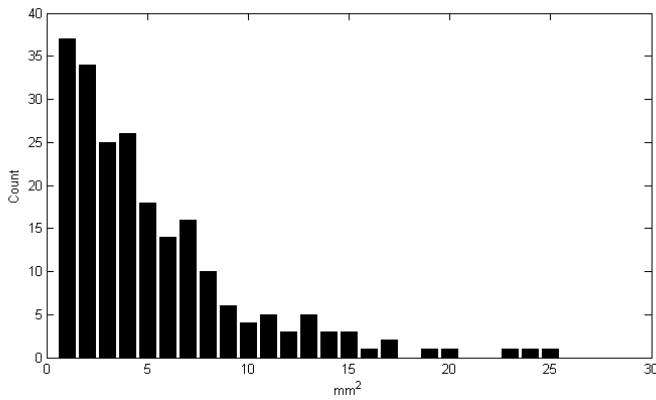


**Figur E.13:** Farvebillede af *sample 3*.

(a) Kødet fundet i *sample 3*(b) Fedtet fundet i *sample 3***Figur E.14:** Kød- og fedtopdelingen af *sample 3*.**Figur E.15:** Gennemsnitsintensitet for kødet fundet i *sample 3*.**Figur E.16:** Gennemsnitsintensitet for fedtet fundet i *sample 3*.

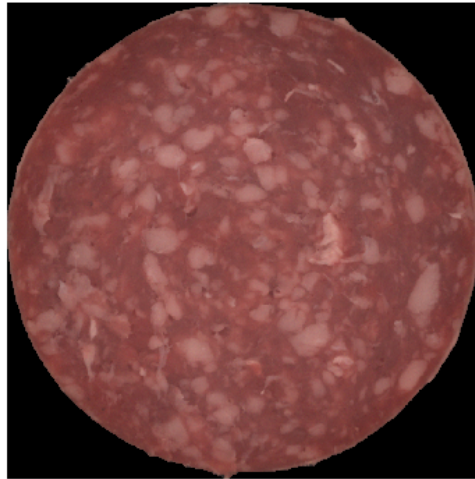


**Figur E.17:** Farvekort med gennemsnitsintensiteter for *sample 3*. Kødringenes farver er angivet med grønt og fedtringenes farver med gult.

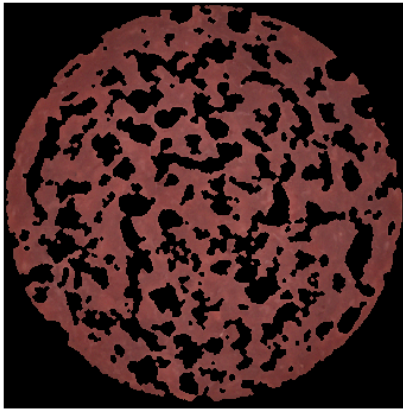
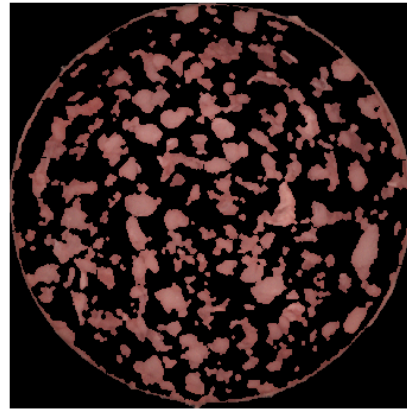
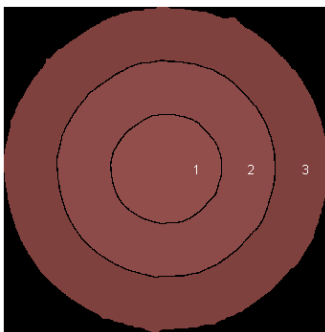
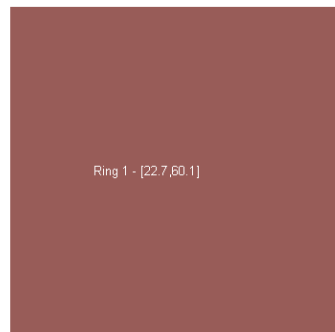
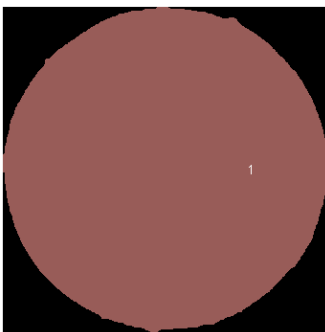


**Figur E.18:** Størrelsesfordelingen af fedtområderne i *sample 3*.

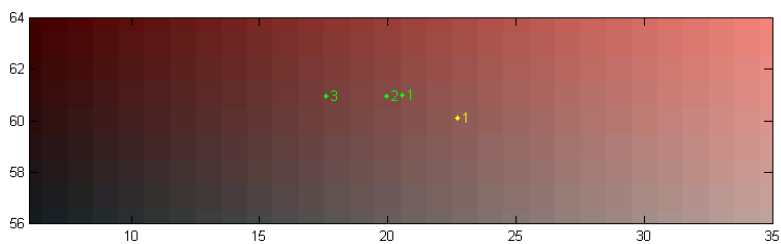
E.4 Sample 4



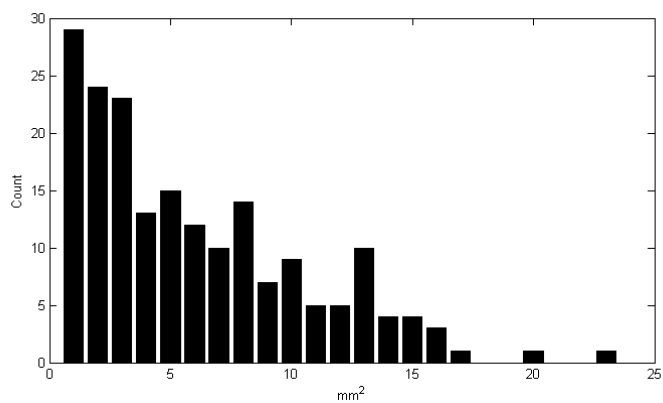
**Figur E.19:** Farvebillede af *sample 4*.

(a) Kødet fundet i *sample 4*(b) Fedtet fundet i *sample 4***Figur E.20:** Kød- og fedtopdelingen af *sample 4*.**Figur E.21:** Gennemsnitsintensitet for kødet fundet i *sample 4*.**Figur E.22:** Gennemsnitsintensitet for fedtet fundet i *sample 4*.



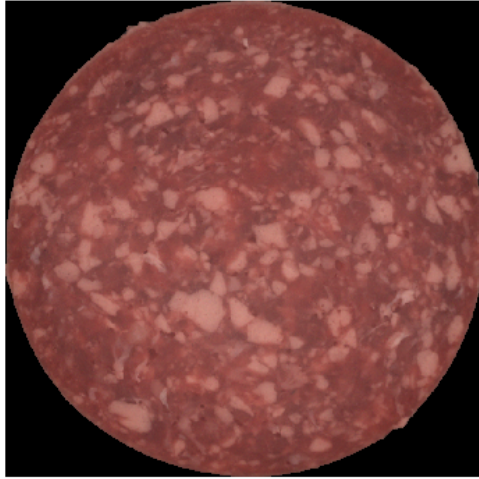


**Figur E.23:** Farvekort med gennemsnitsintensiteter for *sample 4*. Kødringenes farver er angivet med grønt og fedtringenes farver med gult.

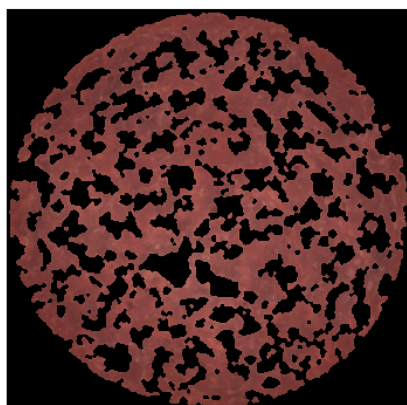
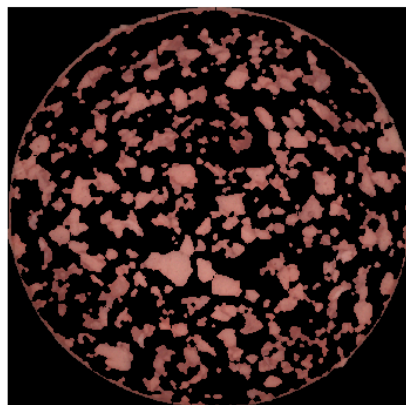
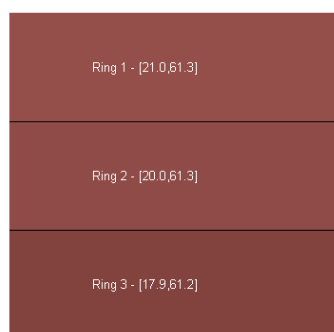
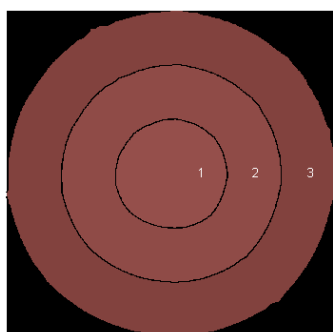
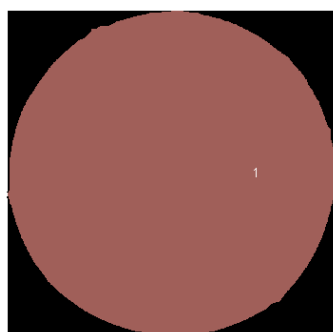


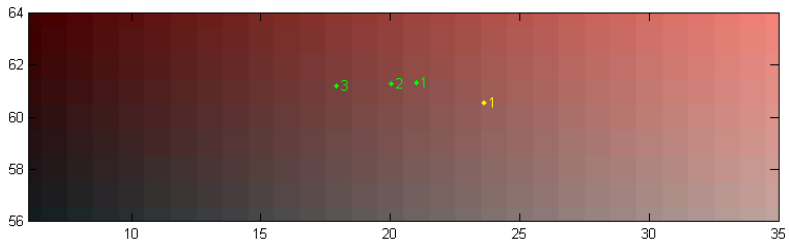
**Figur E.24:** Størrelsesfordelingen af fedtområderne i *sample 4*.

## E.5 Sample 5

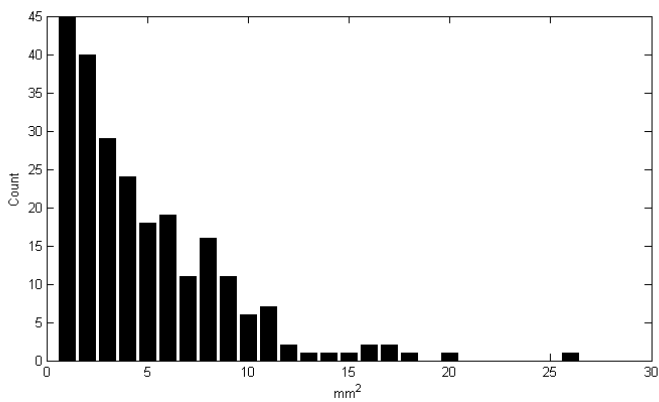


**Figur E.25:** Farvebillede af *sample 5*.

(a) Kødet fundet i *sample 5*(b) Fedtet fundet i *sample 5***Figur E.26:** Kød- og fedtopdelingen af *sample 5*.**Figur E.27:** Gennemsnitsintensitet for kødet fundet i *sample 5*.**Figur E.28:** Gennemsnitsintensitet for fedtet fundet i *sample 5*.



**Figur E.29:** Farvekort med gennemsnitsintensiteter for *sample 5*. Kødringenes farver er angivet med grønt og fedtringenes farver med gult.



**Figur E.30:** Størrelsesfordelingen af fedtområderne i *sample 5*.

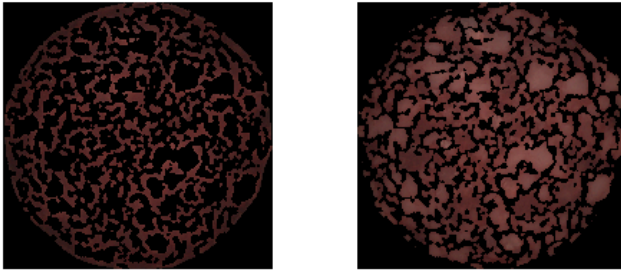
# Spegepølser - Test af adaptive thresholding

---

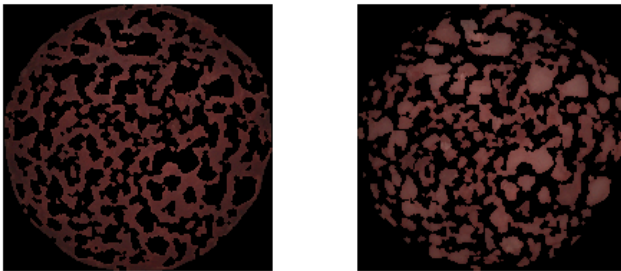
## F.1 Variation af vinduesstørrelsen

Vinduesstørrelsen bestemmer hvor stort et område det lokale gennemsnit skal udregnes for. Det skal være stort nok til at indeholde både fedt og kød, sådan at gennemsnittet ligger sig imellem de to populationer, men den må heller ikke være så stor at der ikke kan kompenseres for lokale variationer. På figur F.1-F.8 er vinduesstørrelsen varieret fra 25 til 200 pixels på hver led, og den fundne opdeling af kød og fedt er vist for spegepølse *sample 1*. Tolerance-konstanten er sat til 0. De øvrige samples viser tilsvarende resultater men er ikke medtaget i dette bilag.

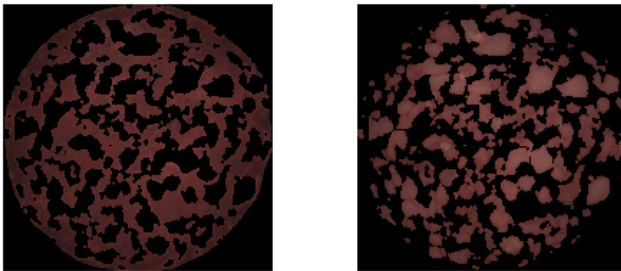
Det er en subjektiv vurdering, hvornår opdelingen er bedst og på billederne af spegepølser kan vi se at det er en glidende overgang imellem kød og fedt, så det er ikke entydigt hvor der skal deles. Det kan være svært at se forskel på billederne når de vises hver for sig, og ændringen sker meget gradvist efterhånden som vinduesstørrelsen øges. Indenfor 50-150 i vinduesstørrelse varierer de udregnede kød- og fedtfarver højst med 0,1 for de forskellige samples, og for langt de fleste målinger er de identiske. For det numeriske resultat er vinduesstørrelsen altså af minimal betydning, når den ligger indenfor 50-150.



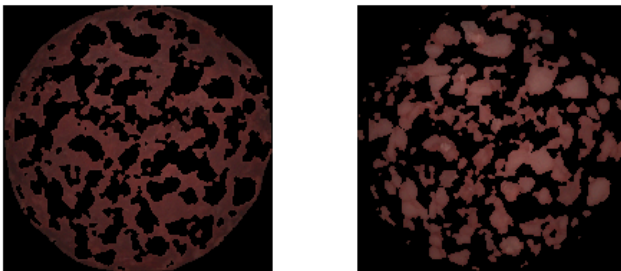
**Figur F.1:** *Sample 1*, kød/fedt adskillelse, adaptiv vinduesstørrelse 25.



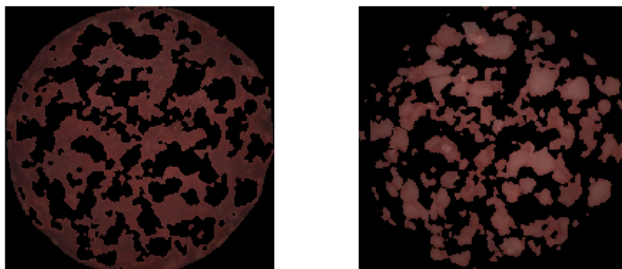
**Figur F.2:** *Sample 1*, kød/fedt adskillelse, adaptiv vinduesstørrelse 50.



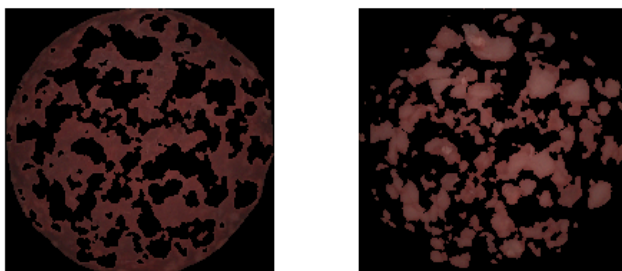
**Figur F.3:** *Sample 1*, kød/fedt adskillelse, adaptiv vinduesstørrelse 75.



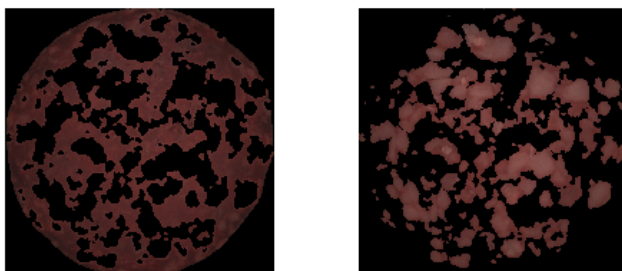
**Figur F.4:** *Sample 1*, kød/fedt adskillelse, adaptiv vinduesstørrelse 100.



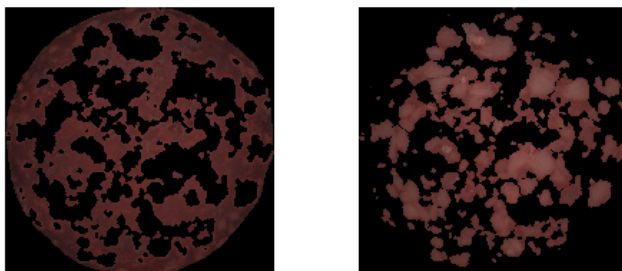
**Figur F.5:** *Sample 1*, kød/fedt adskillelse, adaptiv vinduesstørrelse 125.



**Figur F.6:** *Sample 1*, kød/fedt adskillelse, adaptiv vinduesstørrelse 150.



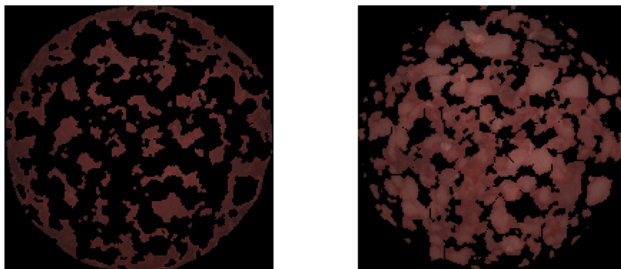
**Figur F.7:** *Sample 1*, kød/fedt adskillelse, adaptiv vinduesstørrelse 175.



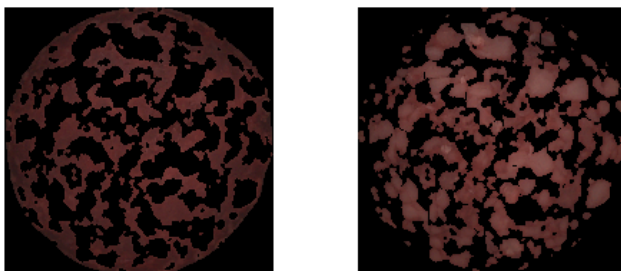
**Figur F.8:** *Sample 1*, kød/fedt adskillelse, adaptiv vinduesstørrelse 200.

## F.2 Variation af tolerance-konstanten

Tolerance-konstanten i den adaptive thresholding bestemmer hvor meget større værdien i den enkelte pixel skal være i forhold til det lokale gennemsnit før den klassificeres som fedt, hvis fedtet har højere intensitet end kødet. Fastsættelsen kan derfor have stor betydning for delingen mellem kød og fedt. På figur F.11-F.14 varieres konstanten mellem  $-0,04$  og  $0,06$  med et interval på  $0,02$  og en vinduesstørrelsen på  $100$ . Pixelværdier er skaleret mellem  $0$  og  $1$ , så sættes konstanten til  $0,04$  skal en pixel have  $4\%$ -point højere intensitet end det lokale gennemsnit. Det er en glidende overgang, dog er det tydeligt at lysere områder der ser ud som fedt kommer med i kød-billedet efterhånden som konstanten øges. For de negative værdier kommer der for meget kød med på fedt-billedet, så det bedste resultat ser ud til at være lige omkring en værdi på  $0$ .

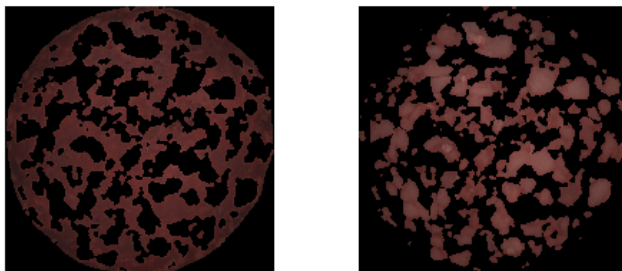


**Figur F.9:** *Sample 1*, kød/fedt adskillelse, adaptiv konstant  $-0,04$ .



**Figur F.10:** *Sample 1*, kød/fedt adskillelse, adaptiv konstant  $-0,02$ .

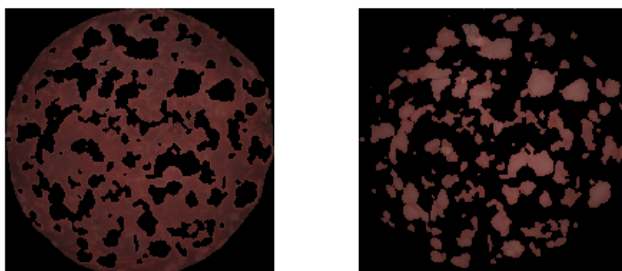




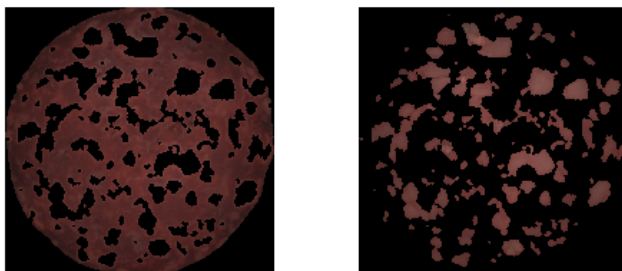
**Figur F.11:** *Sample 1*, kød/fedt adskillelse, adaptiv konstant 0,00.



**Figur F.12:** *Sample 1*, kød/fedt adskillelse, adaptiv konstant 0,02.



**Figur F.13:** *Sample 1*, kød/fedt adskillelse, adaptiv konstant 0,04.



**Figur F.14:** *Sample 1*, kød/fedt adskillelse, adaptiv konstant 0,06.



BILAG G

# Barriersystem rapport - Casper Lund

---

Title:	Coating trials with FLM 01-03-06
Objective:	FLM and two students are working on image analysis of items coated with Grindsted Barrier System. Objective is to coat different items with barrier system in varying thickness for use in their further testing. FLM and two students will participate in trials.

Procedure:

Coating material:  
GRINDSTED Barrier System 2000  
Item 1205210  
Lot 4010235077

Example:

Coating layer is 400 my in one spray  
This equals 1.84 g coating on a standard marie biscuit.  
How to calculate thickness:

$r \cdot r \cdot \pi = \text{area} * 0,04 \text{ cm} (400 \text{ my}) * 0,9 \text{ (density of Barrier System)} * 2 \text{ (sides)}$

Set-up of coating equipment:

Conveyer belt where a tunnel has been build on and a two fluid nozzle is fixed on top.

Heating is applied with a heating gun on the item and conveyer before it is being sprayed.

Heating gun is 30 cm over conveyer, step 1

Definition of spray nozzle from Spraying Systems: Housing for inlet of air and liquid, washer with eight holes for air flow, insert 35100 liquid in centre and air in sides, insert/nozzle 120432 where air and fluid is mixed and atomized through straight cut to give flat cone spraying.

Liquid is in closed container where pressure is controlled by Norgreen regulator 11-818-100 according to how much output of barrier is desired =  $\sim +0.2$  bar Nozzle is fixed 8 cm above the conveyer.

Air flow through nozzle 1.75 l/sec (Hedland flowmeter, 5 bar in). Air is heated through Wegener Duratherm.

Heating is set to 2.

speed of conveyer was adjustable

Coating was done with a barrier temperature of  $65^{\circ}\text{C}$ , but atomizing air in the nozzle is  $\sim 85^{\circ}$  with setting 2

Coating trial (all biscuits from Netto):

Marie biscuits (sprayed on the smooth side), vanilla vafers, All in One "muesli" biscuits with figs and sesame seeds (sprayed on the upper side, Griesen Zartbitter chocolate biscuits (sprayed on the chocolate side):

Amount per spray measured on Marie biscuits =  $0.6 \text{ g} / 2 \text{ sides} = 130 \text{ my}$  per side per spray.

5 of each type were taken and sprayed 1-2-3-4-5 times to give different thickness.

System was then re-adjusted and amount per spray measured on Marie biscuits =  $0.7 \text{ g} / 2 \text{ sides} = 152 \text{ my}$  per side per spray.

5 rice biscuits were taken and sprayed 1-2-3-4-5 times to give different thickness. As rice biscuits were bigger than the spray pattern of the nozzle, a marking was made on the side of each biscuit, and used as a centerline compared

to the nozzle, to ensure that it was placed in the same position for each spray.

System was then re-adjusted and amount per spray measured on Marie biscuits = 1.4 g / 2 sides = 304 my per side per spray.

Marie biscuit were then sprayed on the smooth side

#1 = 2 \* spray

#2 = 10 Pfenning coin placed in the centre of the biscuit, then 2 \* spray

#3 = 1 \* spray, then a 10 Pfenning coin placed in the centre of the biscuit, then 1 \* spray

#4 = 2 \* spray, then a 10 Pfenning coin placed in the centre of the biscuit, then 1 \* spray

Results:

Pictures of system:

(See attached file: Setup spray af Barrier system 2000 10-07-03.jpg)(See attached file: Barrier 2000 semi close up.JPG)(See attached file: Barrier 2000 close up.JPG)

Picture of all biscuits (See attached file: IMG\_0439.jpg)

Coated material was left on trays, put into plastic bags and stored to the next morning until picked up by FLM and students for further work.

Conclusion:

Coating was done by Casper Lund and students Jesper and Lars.

Remarks:

At first it was tried to get as little as 50 my barrier per spray but this would not give a homogenous coating.

Minimum coating layer per spray was then found by adjusting the nozzle down to giving the lowest possible amount while still maintaining a full and homogenous coating. Conveyer speed was then adjusted to maximum.

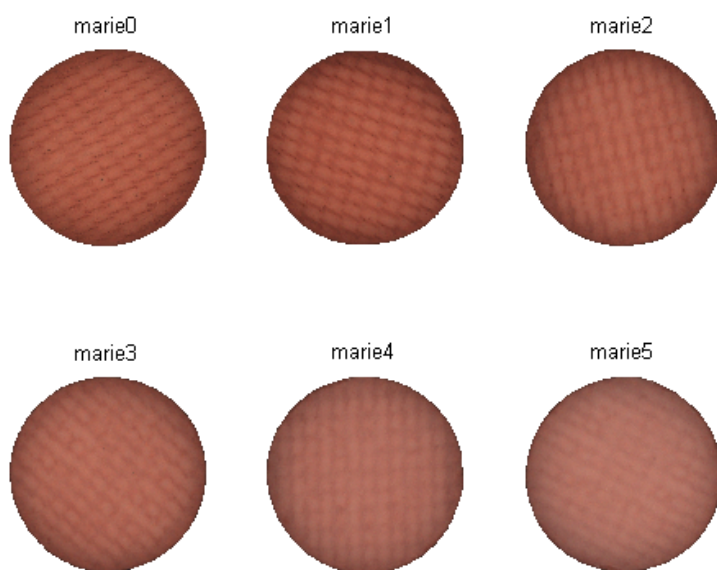


## BILAG H

# Kiks - rådata

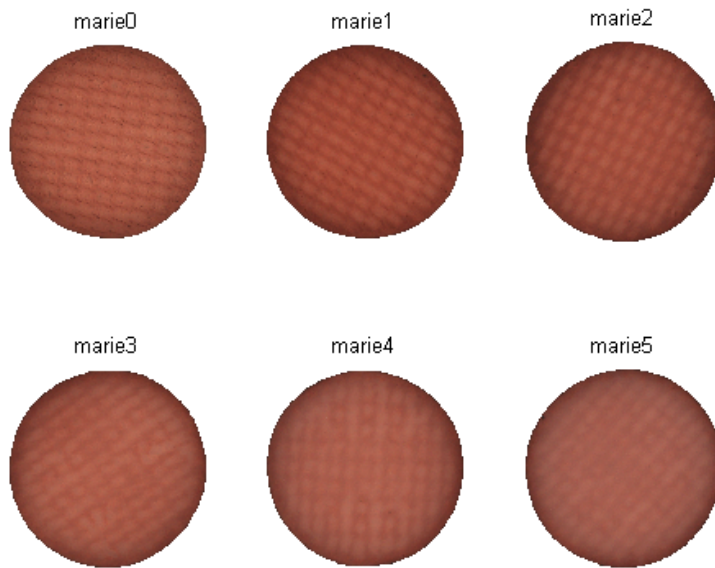
---

Dette bilag indeholder alle billederne fra *kiks med barrierelag*, *Vaffel* og *Mari-ekiks med fejl* datasættene. Billederne er først blevet rensset som beskrevet i kapitel 14.

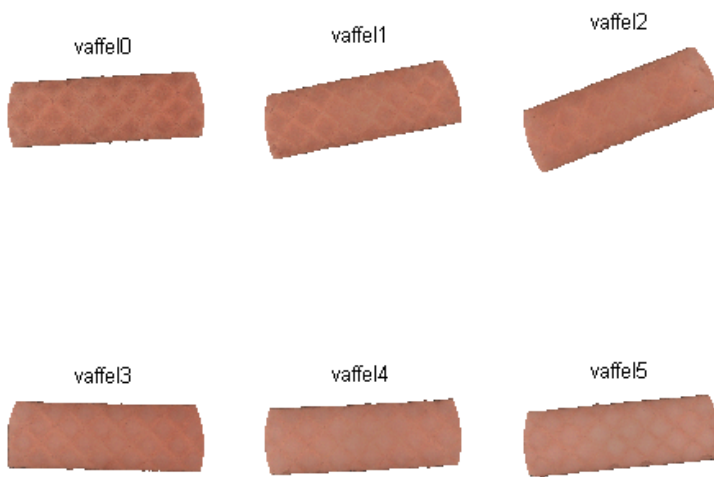


**Figur H.1:** *Marieikiks serie 1.*

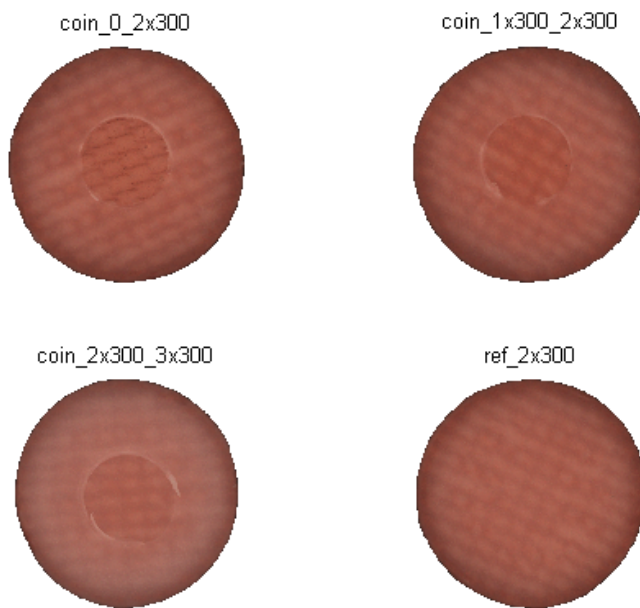




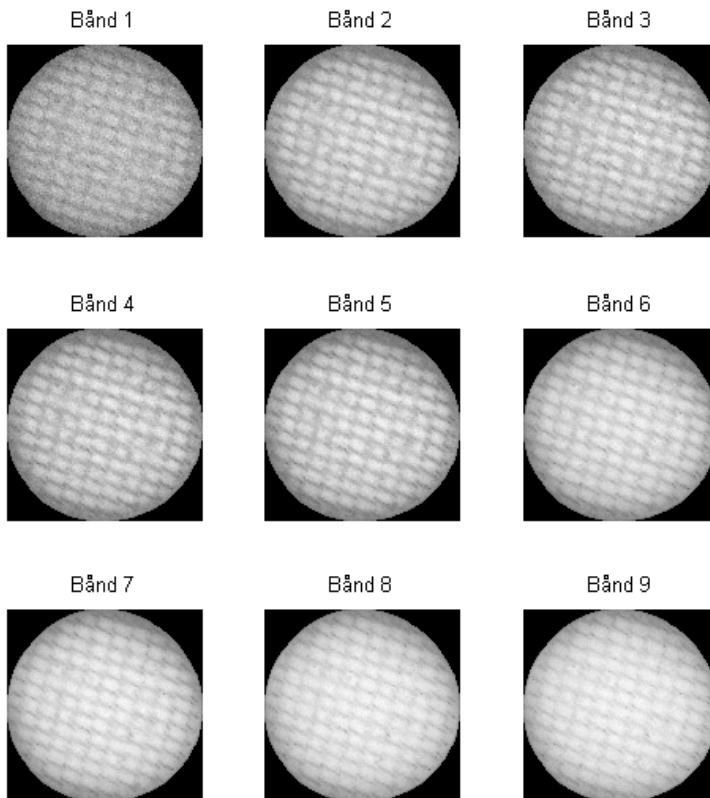
**Figur H.2:** *Mariekiiks serie 2.*



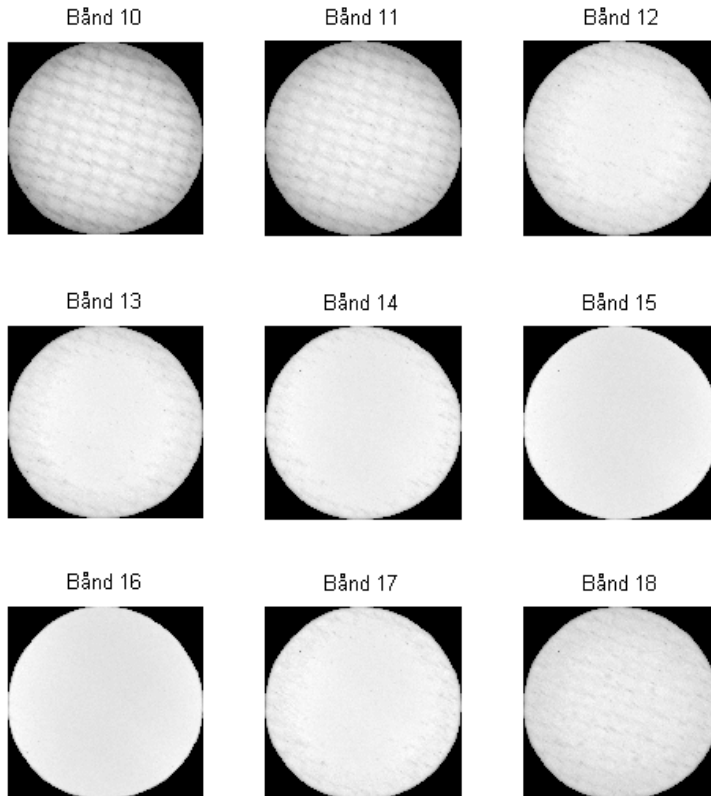
**Figur H.3:** *Vaffel kiks.*



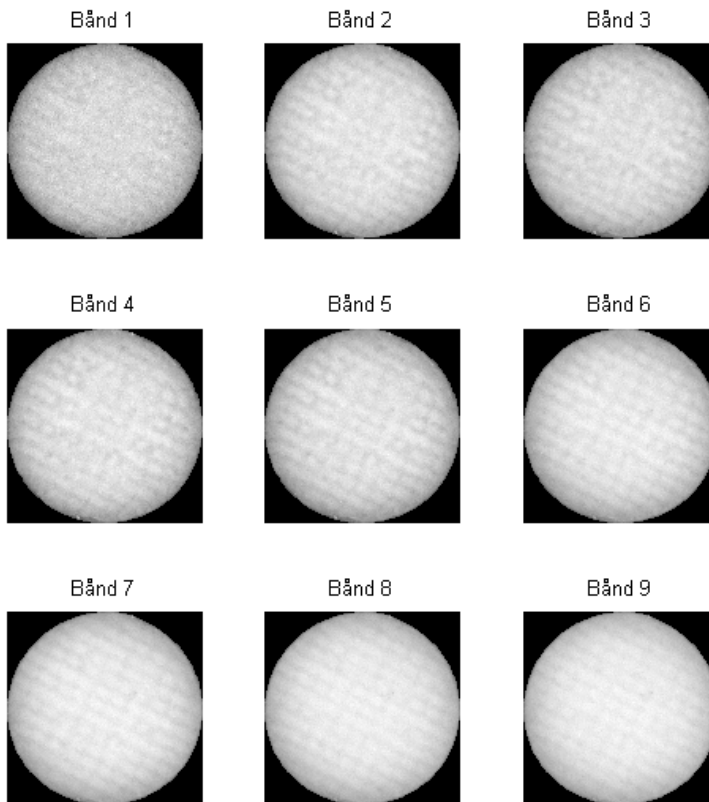
**Figur H.4:** *Mariekiks med fejl.*



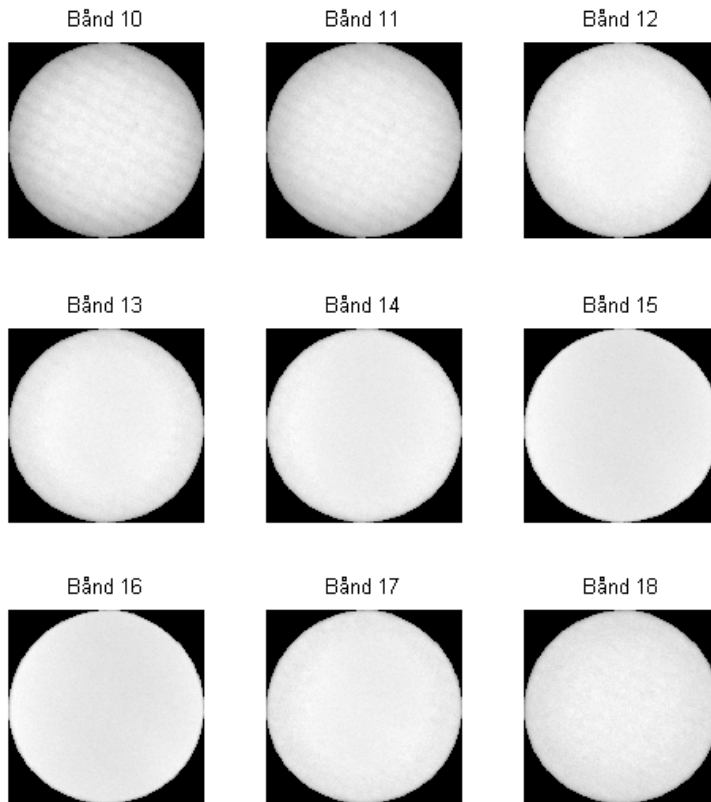
**Figur H.5:** *Mariekiks serie 1 - Marie 1* Bånd 1 til 9.



**Figur H.6:** *Mariekiks serie 1 - Marie 1* Bånd 10 til 18.



**Figur H.7:** *Mariekiks serie 1 - Marie 5* Bånd 1 til 9.



**Figur H.8:** *Mariekiks serie 1 - Marie 5* Bånd 10 til 18.

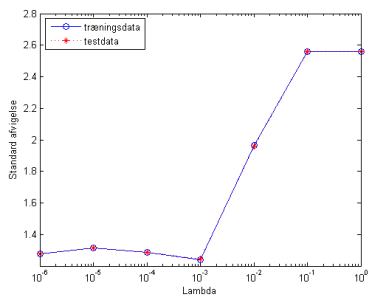




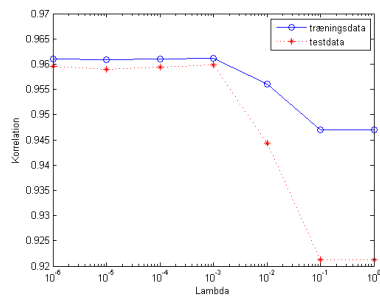
BILAG I

# LARS-EN analyseresultater af brøddata

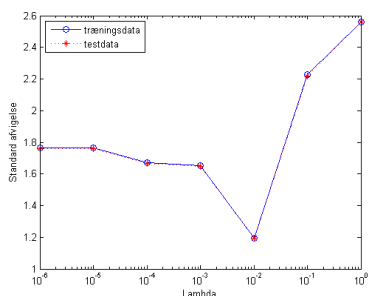
---



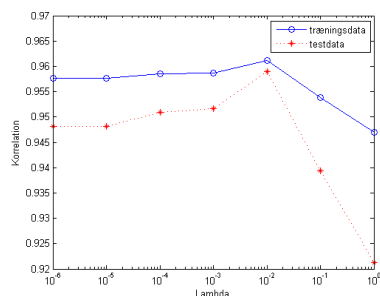
(a) Standardafvigelse for max 10 variable og varierende  $\lambda$



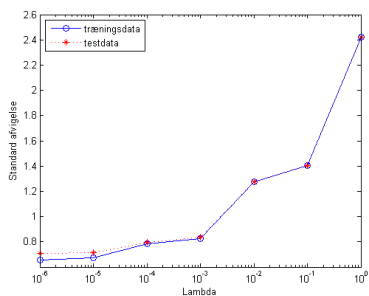
(b) Korrelation for max 10 variable og varierende  $\lambda$



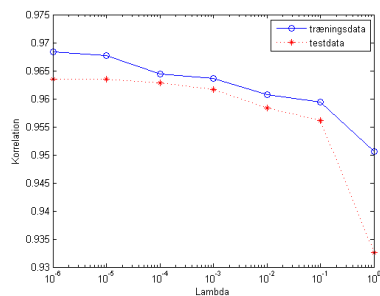
(c) Standardafvigelse for max 20 variable og varierende  $\lambda$



(d) Korrelation for max 20 variable og varierende  $\lambda$

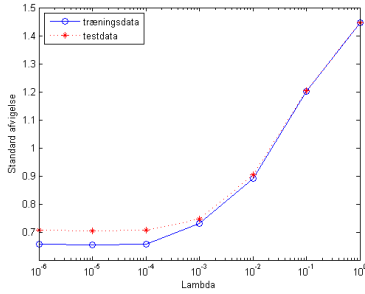


(e) Standardafvigelse for max 40 variable og varierende  $\lambda$

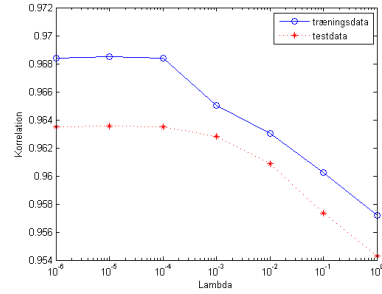


(f) Korrelation for max 40 variable og varierende  $\lambda$

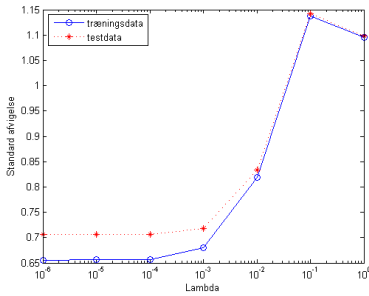
**Figur I.1:** Standardafvigelse og korrelation for HB43-estimat med hhv. max 10, 20 og 40 variable og varierende  $\lambda$ .



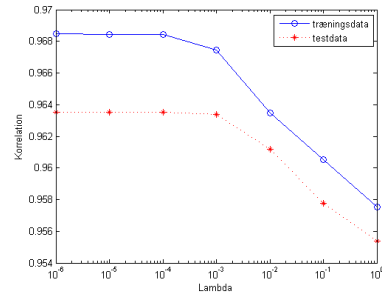
(a) Standardafvigelse for max 100 variable og varierende  $\lambda$



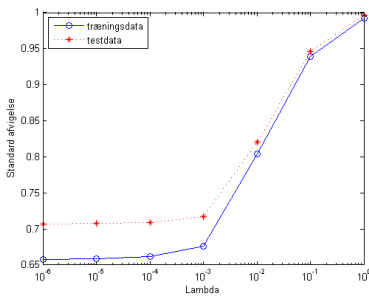
(b) Korrelation for max 100 variable og varierende  $\lambda$



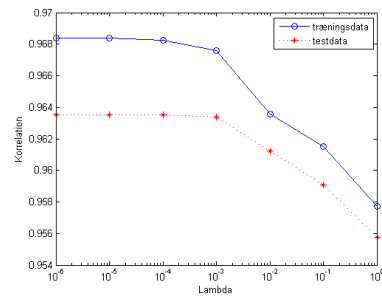
(c) Standardafvigelse for max 200 variable og varierende  $\lambda$



(d) Korrelation for max 200 variable og varierende  $\lambda$

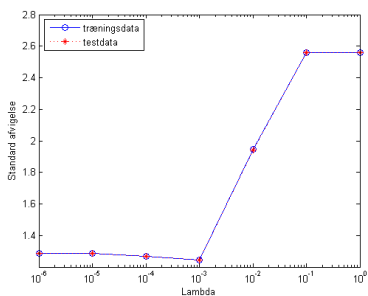


(e) Standardafvigelse for max 400 variable og varierende  $\lambda$

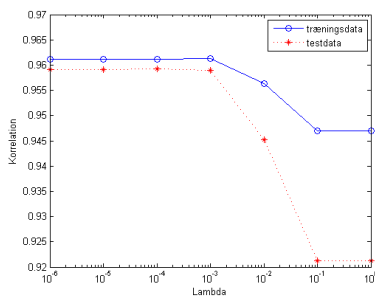


(f) Korrelation for max 400 variable og varierende  $\lambda$

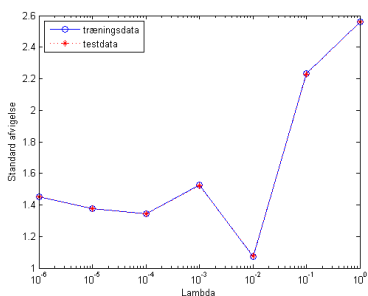
**Figur I.2:** Standardafvigelse og korrelation for HB43-estimat hhv. max 100, 200 og 400 variable og varierende  $\lambda$ .



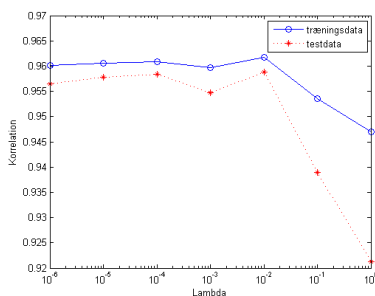
(a) Standardafvigelse for max 10 variable og varierende  $\lambda$



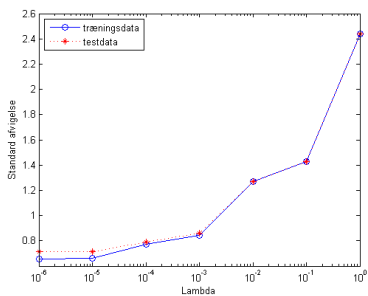
(b) Korrelation for max 10 variable og varierende  $\lambda$



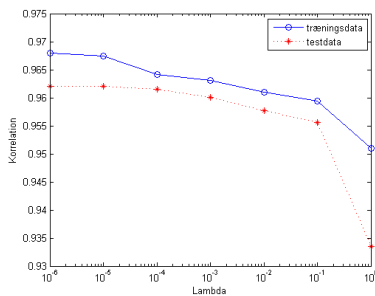
(c) Standardafvigelse for max 20 variable og varierende  $\lambda$



(d) Korrelation for max 20 variable og varierende  $\lambda$

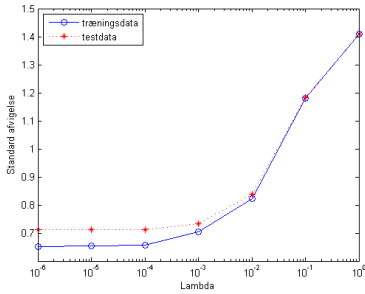


(e) Standardafvigelse for max 40 variable og varierende  $\lambda$

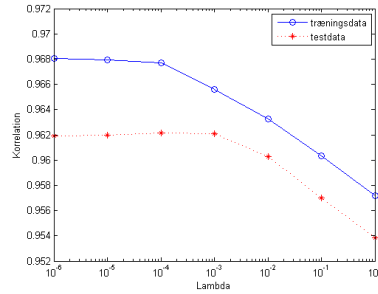


(f) Korrelation for max 40 variable og varierende  $\lambda$

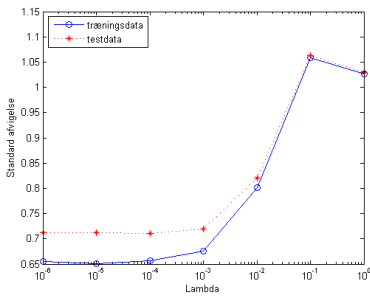
**Figur I.3:** Standardafvigelse og korrelation for  $\log(\text{HB43})$ -estimat med hhv. max 10, 20 og 40 variable og varierende  $\lambda$ .



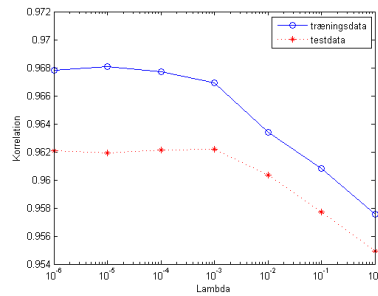
(a) Standardafvigelse for max 100 variable og varierende  $\lambda$



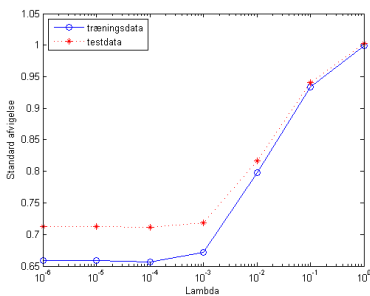
(b) Korrelation for max 100 variable og varierende  $\lambda$



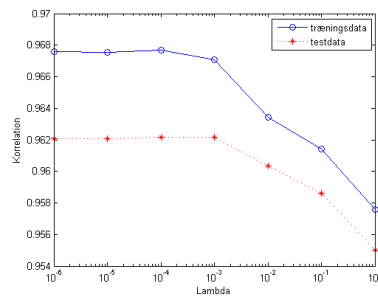
(c) Standardafvigelse for max 200 variable og varierende  $\lambda$



(d) Korrelation for max 200 variable og varierende  $\lambda$

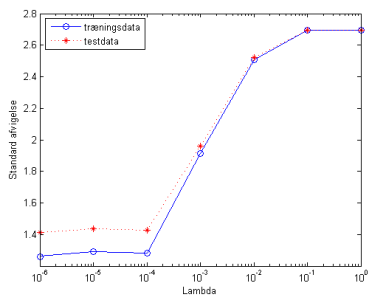


(e) Standardafvigelse for max 400 variable og varierende  $\lambda$

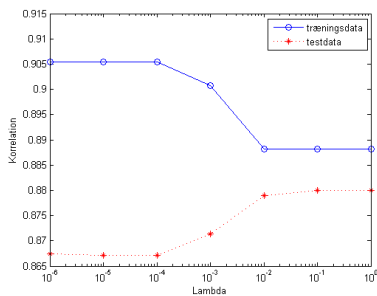


(f) Korrelation for max 400 variable og varierende  $\lambda$

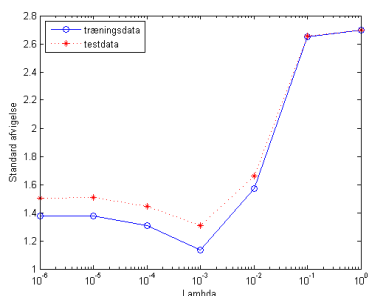
**Figur I.4:** Standardafvigelse og korrelation for  $\log(\text{HB43})$ -estimat med hhv. max 100, 200 og 400 variable og varierende  $\lambda$ .



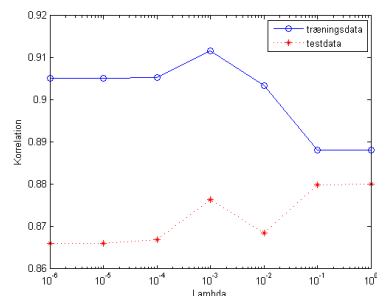
(a) Standardafvigelse for max 10 variable og varierende  $\lambda$



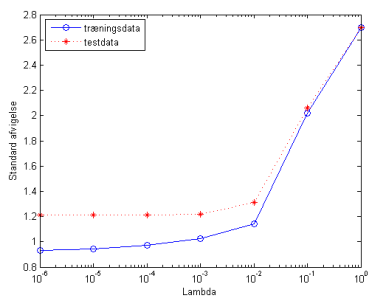
(b) Korrelation for max 10 variable og varierende  $\lambda$



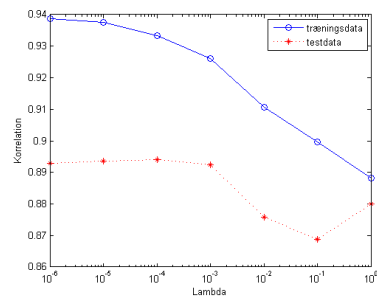
(c) Standardafvigelse for max 20 variable og varierende  $\lambda$



(d) Korrelation for max 20 variable og varierende  $\lambda$

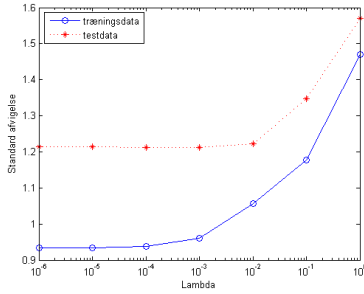
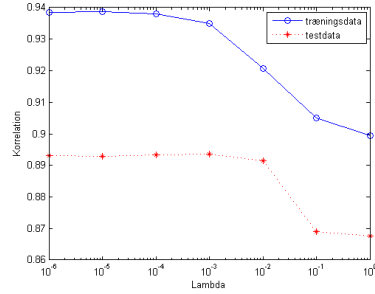
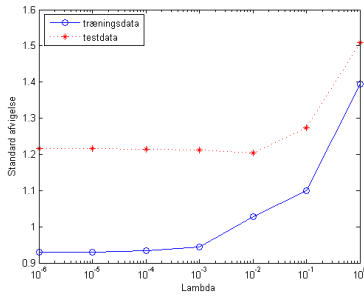
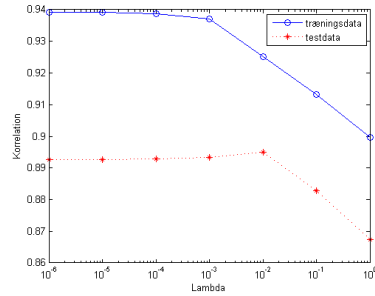
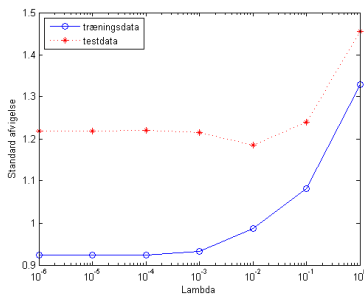
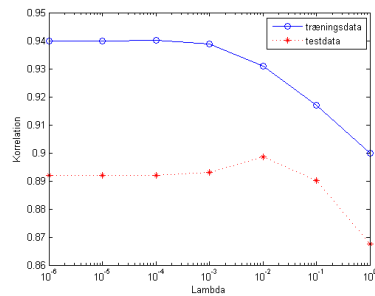


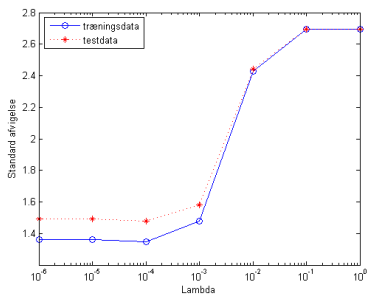
(e) Standardafvigelse for max 40 variable og varierende  $\lambda$



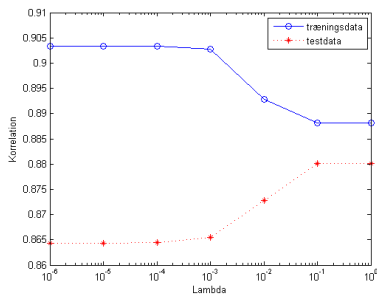
(f) Korrelation for max 40 variable og varierende  $\lambda$

**Figur I.5:** Standardafvigelse og korrelation for LP16-estimat med hhv. max 10, 20 og 40 variable og varierende  $\lambda$ .

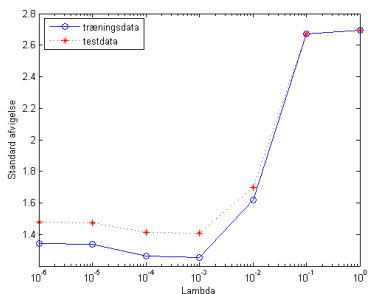
(a) Standardafvigelse for max 100 variable og varierende  $\lambda$ (b) Korrelation for max 100 variable og varierende  $\lambda$ (c) Standardafvigelse for max 200 variable og varierende  $\lambda$ (d) Korrelation for max 200 variable og varierende  $\lambda$ (e) Standardafvigelse for max 400 variable og varierende  $\lambda$ (f) Korrelation for max 400 variable og varierende  $\lambda$ **Figur I.6:** Standardafvigelse og korrelation for LP16-estimat hhv. max 100, 200 og 400 variable og varierende  $\lambda$ .



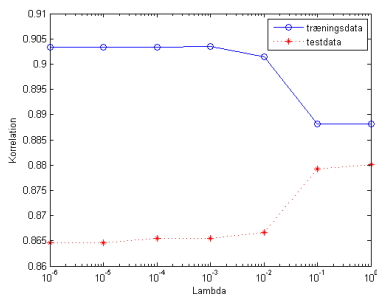
(a) Standardafvigelse for max 10 variable og varierende  $\lambda$



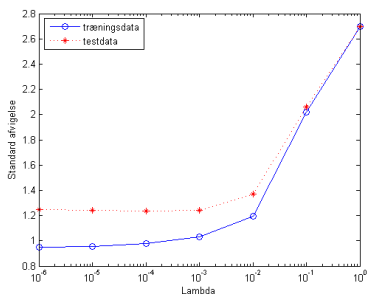
(b) Korrelation for max 10 variable og varierende  $\lambda$



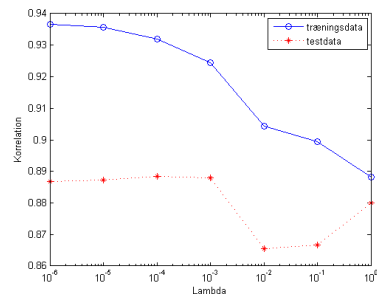
(c) Standardafvigelse for max 20 variable og varierende  $\lambda$



(d) Korrelation for max 20 variable og varierende  $\lambda$



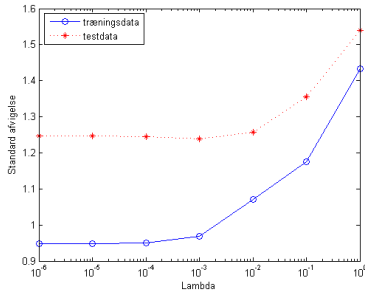
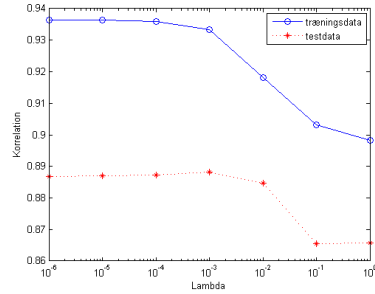
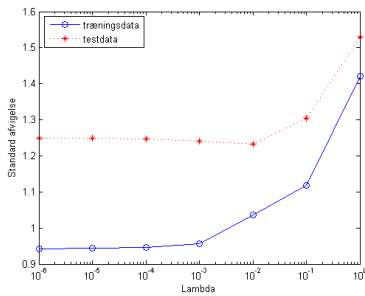
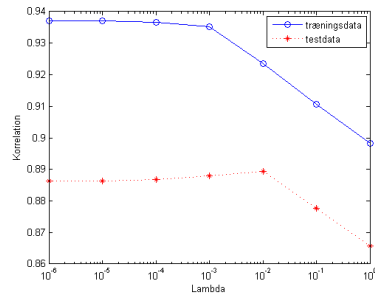
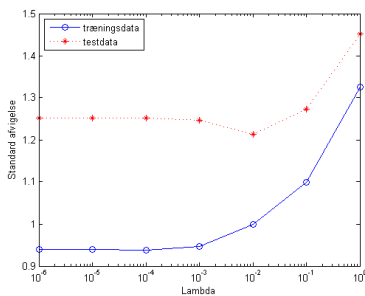
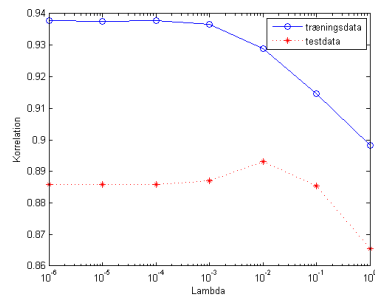
(e) Standardafvigelse for max 40 variable og varierende  $\lambda$



(f) Korrelation for max 40 variable og varierende  $\lambda$

**Figur I.7:** Standardafvigelse og korrelation for  $\log(\text{LP16})$ -estimat med hhv. max 10, 20 og 40 variable og varierende  $\lambda$ .



(a) Standardafvigelse for max 100 variable og varierende  $\lambda$ (b) Korrelation for max 100 variable og varierende  $\lambda$ (c) Standardafvigelse for max 200 variable og varierende  $\lambda$ (d) Korrelation for max 200 variable og varierende  $\lambda$ (e) Standardafvigelse for max 400 variable og varierende  $\lambda$ (f) Korrelation for max 400 variable og varierende  $\lambda$ 

**Figur I.8:** Standardafvigelse og korrelation for  $\log(\text{LP16})$ -estimat med hhv. max 100, 200 og 400 variable og varierende  $\lambda$ .



BILAG J

# Kildekode

---

Al kildekoden som Matlab filer ligger på CD'en herunder:





# Indeks

---

- adaptiv thresholding, 108, 124, 184, 214
- amylopektin, 21
  
- barrieresystem, 9, 20
  - BS2000, 15, 27
  - densitet, 21
  - hovedparametre, 29
  
- Danisco A/S, 12, 43, 114, 129, 135
  
- eigenværdier og -vektorer, 51–53
- EVD, 51, 53
  
- fødevareindustrien, 12, 13
- features, 156
- Forward Selection, 71
- Forward Stagewise regression, 72
- FWHM, 86
  
- grayscale reconstruction, 121
- GretagMacbeth ColorChecker, 218
  
- h-dome transformation, 121
  
- intensitetsniveau udligning, 97, 183
  
- KAS toolbox, 10, 41
- krydsvalidering, 67, 156, 193, 216
  
- LARS, 72, 201
- LARS-EN, 68, 73, 156, 163, 193, 201, 215
  
- Lasso regression, 71
- Lasso-betingelsen, 71
- LF-NMR, 21
- lineær regression, 69, 110, 139, 188, 214
- logistisk regression, 76, 77, 110, 139, 214
- LOO krydsvalidering, 10, 68, 110, 139, 188
  
- MAF, 10, 55, 106, 124, 183, 214
- Matlab, 25, 26, 41, 53, 109, 177, 202, 207
  - hukommelse, 26
- Minolta-farvemåler, 19
- MSI toolbox, 41
  
- OLS, 70
  
- PCA, 10, 51–54, 103, 115, 183, 214
  
- Ridge regression, 70
  
- score-billede, 54, 58
- score-plot, 58, 102
- spegepølser
  - bakteriekultur, 14
  - pH-værdi, 14
  - sensorikdata, 19, 35, 114, 127, 214
- SVD, 51, 54
  
- vandaktivitet, 16
- vandindhold i brød, 21

DSC, 21  
HB43, 21  
LP16, 21  
retrogradering, 17, 21  
Videometer A/S, 12  
VideometerLab, 23, 26, 43, 115, 128,  
129, 144, 161, 181, 218  
bølgelængder, 244  
Ulbricht kugle, 24  
  
watershed segmentering, 119