Pharmaceutical Tablet Inspection

Emil Sauer Lynge

Kongens Lyngby 2012
IMM-B.Sc.-2012-06

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
www.imm.dtu.dk IMM-B.Sc.-2012-06

Summary (English)

The goal of this thesis is to investigate, how an industrial machine vision solution can be improved using spectral analysis - especially the UV spectrum. It is a case study taken from Glostrup Dosispak, but the findings can be generalized as they are independent of the existing image processing implementation. The results thus only reflects the explanatory power of the tablets spectral response. A method is proposed for classifying that involves a variation of SVM, a voting system and histograms for evaluating observed to expected responses.

Summary (Danish)

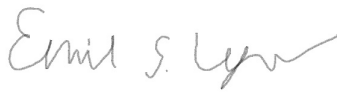
Målet for denne afhandling er at undersøge hvordan en eksisterende, industriel machine vision løsning kan forbedres ved hjælp af spektralanalyse. Der tages udgangspunkt i et case fra Glostrup Dosispak, men fundene kan generaliseres, idet de er afkoblet den eksisterende image processing implementation. Det forsøges altså at afklare hvor meget forklaringsevne tabletters spektrale respons besidder.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring an B.Sc. in Mathematics.

The thesis project, entitled Pharmaceutical Tablet Inspection, was carried out in the period of 3rd of October 2011 to 27th of February 2012 and corresponds to 15 ETCS points. This thesis was supervised by Associate Professor Jens Michael Carstensen at IMM, DTU.

Lyngby, 27-January-2012



Emil Sauer Lynge

Acknowledgements

I would like to thank my supervisor Jens Michael Carstensen for his guidance, and especially his technical assisistance with the VideometerLab. And i would like to point out, that without the highly specialized UV-equipment he has developed, this project would not have been realized.

I would also like to thank Glostrup Dosispak for the time that i worked there, which provided me with inspiration for a project, and their good will during the project. Especially i would like to thank Claus Lund Christensen, my former colleague who was integral in establishing a cooperation with Glostrup Dosispak and supplying the samples needed to build my data set.

Contents

1 Introduction

1.1 Terminology

This report deals a lot with the concept of binary classification and errors hereof. In this context it can be ambiguous which outcome is deemed positive and which is negative. We therefore must introduce a convention which will be followed throughout the report:

When inspecting tablet samples the null hypothesis is that no error has occurred and the event of an error is a *positive* and the event of no error is a negative. It follows that

- A true positive is catching an error
- A true negative is letting an OK sample pass.
- A false positive is rejecting an OK sample
- A false negative is not catching and error

This convention is justified in section 4.1

When binary classifying pixels as belonging to either a group A or B. it will usually be defined as classifying A against B. in the context of this classifier group A will be defined as class 1 and B as -1 or 0. Class A will then be the positive event and B the negative.

- A true positive is classifying A as A
- A true negative is classifying B as B
- A false positive is classifying B as A
- A false negative is is classifying A as B

1.2 Background

For elderly people, drug addicts in rehabilitation, psychiatric patients and other unstable or forgetful patient groups, taking the right medication at the right time is critical yet difficult. A common way to alleviate such difficulties is to organize the medication using a pill box. Either the patient themselves pack these on a weekly basis or – more commonly among weaker patients – a nurse does this. Having this done by nurses poses several problems:

- Manual repetitive work such as this is prone to error
- there is usually no safe guard if such an error occurs
- It is a very time consuming and hence costly way of dispensing medicine.
- Dispensing from blister packs and

Picture 1 Medidos pill box from Kibodan

snapping tablets for half dosages can be quite wearisome for the fingers.

Picture 2 Automatic dose packing machine

A way to get around these issues is automation of the dosage packing. A large machine with 300-400 canisters containing different tablets dispenses medicine directly into a throwaway "pill box". The "pill box" is a long strip of plastic foil with welded pockets equivalent to the dose rooms in the normal pill box. Each pocket has a time and date printed onto it describing when the medicine inside is to be taken.

Having a machine doing this work, the effect of human errors is greatly reduced. It is far less work intensive than manual dosing, and the un-blistering and tablet snapping is centralized to a place where appropriate equipment can be used. Errors still occur however, and the question of a safe guard still remains.

Originally the safe guard consisted of visual inspection by a human operator, thus reintroducing the element of human error and adding to the amount of manual labour required.

1.3 The Case – current vision system

To get rid of the manual inspection Glostrup Dosispak acquired a vision system for tablet inspection. The setup involves a conveyor to move the foil strip under a camera with backlight. This provides a silhouette picture of each pocket which undergoes binary image processing to determine whether or not an error has occurred. When the machine predicts a sample to be positive (erroneous) a human operator must then visually inspect the sample to determine whether this is a true or false positive. This being the pharmaceutical industry the tolerance for false negatives is very low and hence the recall must be near 100%. Naturally this drives the precision down considerably. This is also been the case and the result is a lot of false positives, which is very work intensive and therefore costly.

Picture 3 Manual tablet inspection

It is this cost that I we are interested in reducing by decreasing the amount of false positives i.e. the precision.

Since recall is bound to near 100% the only way to increase precision is improving the test accuracy i.e. discriminative power. This report will be concerned with this problem of adding more discriminative power to the existing model.

2 Improving model performance

Although the purpose of the study case is to improve the existing model it is not necessary to be familiar with the details of this model. We can simply assume

that the existing model makes full use of the information available to it and instead focus on making another model which uses none of this information. Any discriminatory power that we gain from such a model will be completely additive to the existing if it is based on data completely uncorrelated to the existing data.

Here it might be useful to clarify what information the existing and new model is based upon:

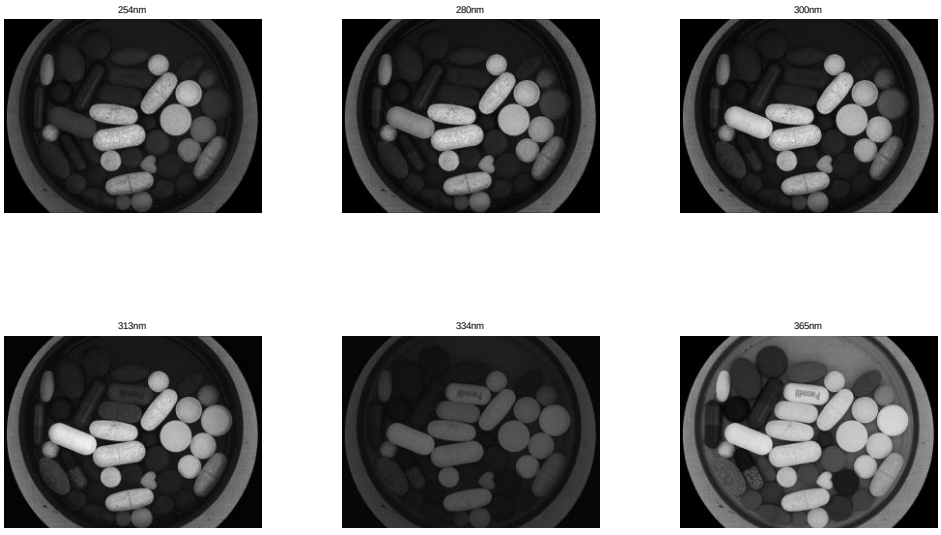
- As the existing model uses a binary silhouette picture of the sample it can be assumed that only spatial information is used such as area, count, and shape of the tablets.
- The new information that will be used is spectral data. Such as overall color, distribution and variation.

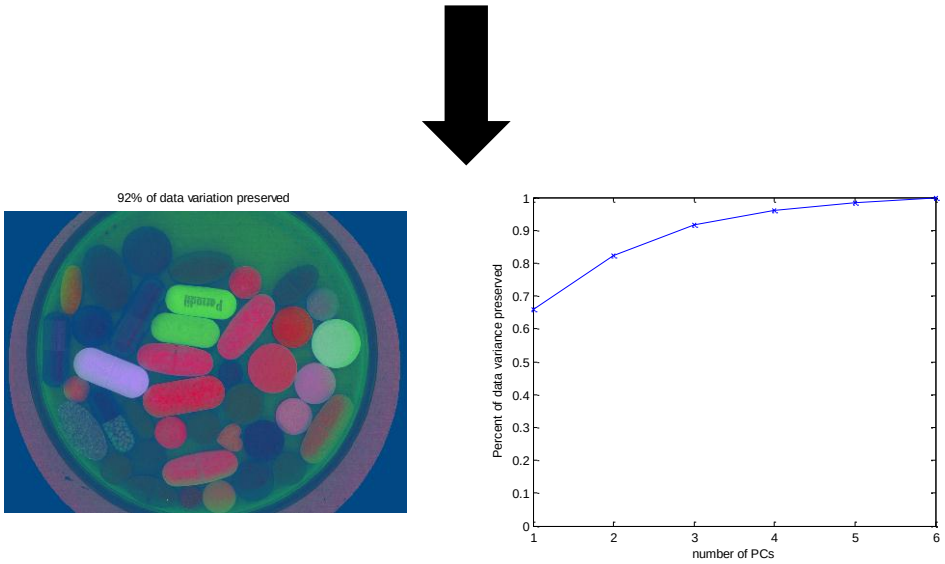
In order to assume that the two models are independent it is important to note that the new model should refrain from using techniques that exploit, that spectral data inherently preserves a lot of the spatial information. Therefore a true test of the model performance must rearrange the pixels randomly and remove parts of the picture.

3 Initial data analysis

In all machine learning tasks it is important to begin with a visualization of the data. Initial collected data consist of images of several tablets in both UV-spectrum and the visible.

The UV data has 6 bands equivalent to 6-dimensional data and the visible has 20. It is therefore inconvenient to show every band as grayscale pictures. Instead PCA-analysis is performed on the picture and represented by taking the Red colorband as PC1, the blue as PC2 and green as PC3.





Initial results here indicate what you would expect. In visible light coloured tablets have large variation and the white tablets have next to nothing. This changes completely in the UV spectrum where the white tablets are hugely variated but the coloured not so much.

This is due to the fact that it's the same mechanism that absorbs UV light and visible light only 'scaled differently'. It has to do with conjugated π-bonds and how delocalized they are. Higher delocalization makes it easier for to excite an electron in the bond and requires less energy. This means that longer molecules with high delocalization absorb light with lower energy and conversely with shorter molecules with less delocalization. As UV-light has a shorter wavelength and therefore higher energy white tablets must generally have low delocalization.

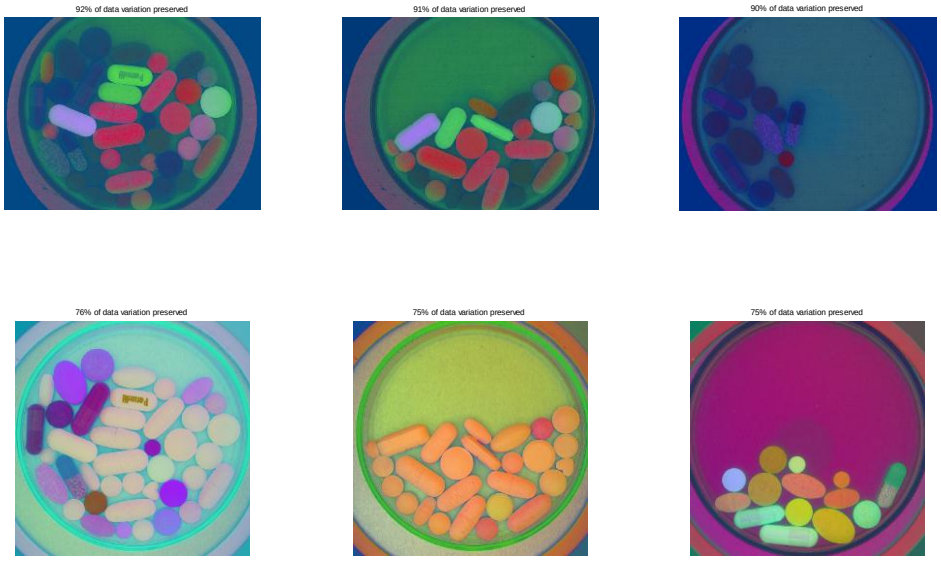


figure 1 first 3 PCs depicted as R G B color bands. Top is UV-bands and bottom is visible light. Left is all tablets, middle is white only and right is coloured only.

Based on these observations and the assumption that white tablets are more frequent than coloured I decided early to pursue a model based on UV light. But since the principle is the same as with visible light results can easily be generalized to this.

3.1 The Data set

In all there the data of UV pictures of 42 different types of tablets (non prescription such as vitamins and chalk, the full list is in the Appendix). Pictures were taken with 6 different UV bands (254nm, 280nm, 300nm, 313nm, 334nm, 365nm). There are pictures of 2 different tablets of each type and pictures were taken from the same angle (face up).

4 Theory

4.1 Sample inspection null hypothesis

The test for errors in the samples can be described as hypothesis testing. In this context choosing the correct null hypothesis is important. We can either define the null hypothesis as:

A - “An error has occurred”

Or

B - “No error occurred”

In hypothesis testing a decision is made by rejection or non-rejection of the null hypothesis. In case A you would have to reject every conceivable error to fully reject the hypothesis. This imply exhaustive prior knowledge about which errors can occur - a combinatorial infeasibility.

In case B the rejection of the null hypothesis implies rejection of normality. E.g. the sample does not conform to normality. Since the normal behavior can be extensively described it is quite easy to measure how similar or dissimilar the sample is to the known normality.

H_0: The sample is normal – no error has occurred

H_1: The sample is not normal – an error has occurred

The rejection and non-rejection is fundamentally a binary classification problem where observations are predicted to be normal (class 0) or not normal (class 1).

4.2 Cross-validation

All classifiers require some kind of parameters that manipulates the behavior of the data. For example

- Layers in artificial neural networks
- k in k-nearest neighbours
- class imbalance in the training data for class imbalance sensitive classifiers (most classifiers are sensitive to this)
- Cost function in Naïve Bayes
- Feature selection
- And many more

Usually the appropriate value for a parameter cannot be derived directly and we must test different values and pick the one that gave the best performance.

Two issues arise:

- What is meant by performance
- How do we test

4.2.1.1 Performance

The performance of a classifier is how well it does on a test set according to some goal function. A typical goal function is the accuracy, but this is not always the case and thoughtless use of this measure can lead to a classifier of no use in the intended application. E.g. a test for a terminal but rare disease would have great accuracy if all patients were automatically deemed healthy, but this would lead to the death of people unfortunate enough to have the disease. Other criteria could be robustness, f-score, unequal cost functions, theoretical error rates using prior knowledge of error types.

4.2.1.2 Independency of test data

A test should be designed such that it can be reliably estimated, how well the classifier will perform on future test sets. To do this the test set must have the have the same preconditions as a future test sets. Therefore the classifier must be built as if the test set didn't exist, which means we can't use the same data for training and testing. So either we have a large training set which accurately describes the construction of the final classifier but a small test set and therefor unreliable performance estimate. Or we have a large test set that accurately estimates the built classifier but a small training set which poorly describes the construction of the final classifier.

4.2.1.3 Hold out cross validation

To work around this paradox we use cross-validation. The simplest (and most cumbersome) version of this is called the "hold out-method". It works by building a classifier using all data

except one observation. This observation is subsequently classified using the built classifier. The process is then repeated until all observations has been held out and classified. A final test score is the average test score of all the classifications.

This way no test score have been previously "seen" by the classifier that classified it and the classifiers are almost identical to a classifier built using the full data set.

4.2.1.4 k-fold cross validation

This however is very cumbersome so a less precise method is usually applied. Instead of holding out a single observation a larger subset of the data is held out. This method I called k-fold cross validation and the size of the subset is such that k subsets contain all the data. Being less precise is in a lot of cases mostly theoretical. When the total dataset becomes large enough the sample size will become so much greater than the variation of the data set that it is unlikely that the training set will become skewed in some way.

4.2.2 Parameter tuning

The process of choosing appropriate parameters is called parameter tuning. The easiest and most robust way is to define some bounds on the parameters such that any reasonable value of the parameter must lie between the bounds. The parameter is then tested for a subset of values inside this interval (usually evenly distributed). If more than one parameter exist it's necessary to consider whether they are independent on another. Independent parameters can be tuned independently, but most likely they won't be independent in which case all combinations parameter values must be tested.

This is reasonable with 2-3 parameters, but unfortunately the number of classifiers that must be built increases exponentially with the number of different parameters.

If the number of dependent parameters rise above 4-5 it is usually infeasible to compute every combination. In this case a greedy numerical optimization would seem more applicable. A natural choice could be using Hooke and Jeeves direct search method which will not be covered in this report.

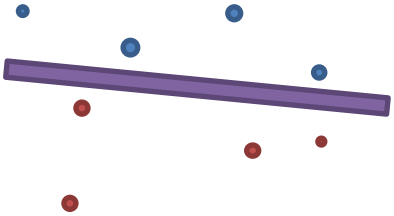
4.3 Support Vector Machine

This classification method is inherently binary, meaning that it can only handle two classes at a time in its basic form. In order to use it for multiclass-classification several binary classifiers must be combined. This means that in order to handle n classes, a minimum of n-1 classifiers must be built.

Still it's a very powerful method as it provides a flexible and, to some extent, non-parametric decision boundary.

- Flexibility. In its basic formulation it is possible to construct any conceiveable decision boundary, although the necessary complex models would be infeasibly slow
- Non-parametric. The method can often be used without any prior knowledge about the underlying mechanism that created the data.

4.3.1 Linear SVM



The simplest form of SVM is a linear classifier and only works on linearly separable datasets. This means that the classes can be perfectly separated by a hyperplane

$$a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n = 0$$

For example one could draw red and blue dots on a sheet of paper. These would be linearly separable if and only if the red and blue dot can be perfectly separated using a ruler.

Given such a separable set in d-dimensional space we try to find a hyperplane

$$\sum_{\boldsymbol{d}} w_d \cdot x_d = b \quad \Leftrightarrow \quad \boldsymbol{w} \cdot \boldsymbol{x} = b$$

Such that there exist 2 parallel hyperplanes on the form

$\boldsymbol{w} \cdot \boldsymbol{x} - 1 = b$ and $\boldsymbol{w} \cdot \boldsymbol{x} + 1 = b$

Subject to

$\boldsymbol{w} \cdot \boldsymbol{x_i} - 1 \geq b \quad for\ x_i$ belonging to the positive class

and

$\boldsymbol{w} \cdot \boldsymbol{x_i} + 1 \geq b \quad for$ belonging to the negative class

This implies that there can be no points between the 2 parallel hyperplanes henceforth called the margin. And that no points of different classes can exist on the same side of the boundary. This should be no surprise as it's simply a way of expressing conditions which are both necessary and sufficient for linear proving separability.

An equivalent and convenient way of expressing these constraint is

$y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} - b) \geq 1 \quad \forall_{x_i}$ where $\begin{cases} y_i = 1 & for\ y_i\ in\ positive\ class \\ y_i = -1 & for\ y_i\ in\ negative\ class \end{cases}$

But assuming the points are not aligned this condition is met for infinitely many $\boldsymbol{w}$ and b.

to provide uniqueness to the solution and to maximize robustness of the classification we introduce the goal function

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|^2$$

Which maximizes the margin.

This problem has convex goal function an linear classifier and is therefore easily solved using quadratic programming for the lagrange multipliers.

Using one of the the 1st order optimality conditions

$$\boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x_i}$$

We can find a **w** and then classify a new point **z** using

$$signum(\boldsymbol{w} \cdot \boldsymbol{z} - \boldsymbol{b})$$

But since not all constraints are active a lot of the λ will be zero. So instead of directly calculating $\boldsymbol{w}$ we simply keep the $\boldsymbol{x_i}$ which have active constraints and classify using

$$signum\left(b + \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x_i} \cdot \boldsymbol{z}\right)$$

Where N is now the number of active constraints. These vectors we kept are called support vectors.

4.3.2 Non-separable data

If the data is non linearly separable, which it often isn't. we can add so called slack variables to the inequality constraints such that points can now exist inside and even on the wrong side of the boundary. To maintain a useful model penalize such points in the goal function by adding the summed slack variables.

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 + C \cdot \sum_i \xi_i$$

The C is a coefficient that describes the tradeoff between a wide margin and more errors. Lowering C widens the margin but allows more points to exist in the margin. This parameter is useful when tuning to avoid under/overfitting.

4.3.3 Nonlinear transformation

Most exciting thing about SVM it's ability to do make non linear boundaries as if they were linear. To inject non linearity to the model we transform the variables into a higher dimensional space, in which a linear classifier corresponds to a nonlinear classifier in the original space. We could for instance transform the point $\{x_1, x_2\}$ into a 2-dimentional space using the nonlinear transformation:

$$\mathbb{R}^1 \to \mathbb{R}^2 \quad \Phi(x_1, x_2) \to \{x_1, x_1^2, x_2, x_2^2\}$$

We denote these new coordinates $\{y_1, y_2, y_{3,} y_4\}$

In the transformed space we see that the linear equation

$$ay_1 + by_2 + cy_3 + d_{y4} + e$$

Is equivalent to the nonlinear equation

$$ax_1 + bx_1^2 + cx_2 + dx_2^2 + e$$

This makes it possible to create infinitely complex and versatile nonlinear classifiers using just simple transformations and linear classifiers. When classifying in transformed space we modify the classification function slightly

$$signum\left(b + \sum_{i=1}^{N} \lambda_i y_i \Phi(\boldsymbol{x_i}) \cdot \Phi(\boldsymbol{x})\right)$$

4.3.4 *Kernel trick*

There is a catch however. This transformation is computationally expensive especially for datasets that has high dimensionality to start with. To solve this we we use what is called the kernel trick[1].

We note that the dot product

$$\begin{pmatrix} \sqrt{2}x_1 \\ x_1^2 \\ \sqrt{2}x_2 \\ x_2^2 \\ 1 \end{pmatrix} \circ \begin{pmatrix} \sqrt{2}y_1 \\ y_1^2 \\ \sqrt{2}y_2 \\ y_2^2 \\ 1 \end{pmatrix}$$

Is equal to

$$x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 + 2x_2 y_2 + 1 = (\boldsymbol{x} \cdot \boldsymbol{y} + 1)^2$$

This is called a kernel and is expressed

$$K(\boldsymbol{u}, \boldsymbol{v}) = \Phi(\boldsymbol{u}) \cdot \Phi(\boldsymbol{v}) = (\boldsymbol{u} \cdot \boldsymbol{v} + 1)^p$$

Where p=2 in the preceding case.

As mentioned above the classification uses

$$signum\left(b + \sum_{i=1}^{N} \lambda_i y_i \Phi(\boldsymbol{x_i}) \cdot \Phi(\boldsymbol{z})\right)$$

Which is now substituted with

$$signum\left(b + \sum_{i=1}^{N} \lambda_i y_i \cdot K(\boldsymbol{x_1}, \boldsymbol{z})\right)$$

The kernel used in this project is a Radial basis function and is expressed

$$K(\boldsymbol{u}, \boldsymbol{v}) = e^{\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{2\sigma}}$$

This kernel represent the dot product in an infinite space and

4.3.5 *Separable Case Approximation*

[1] Page 273 eq. 5.6 [**3**]

With large data sets it can become a problem that solutions has a large number of points inside the margin. As mentioned the vectors we keep for classifying new observations are the one with active constraint, which means that points inside and on the boundary of the margin becomes support vectors and having too many support vectors is very bad for computation time.

We therefore implement an algorithm called separable case approximation [**1**]. Which as the name says approximates the solution had the dataset been separable.

We generate a SVM for the non-separable case and use this to classify all the points used to generate the SVM. We then throw away any points that were misclassified.

We keep all points that satisfy: $y_j\left(b + \sum_{i=1}^{N} \lambda_i y_i \cdot K(\boldsymbol{x_i}, \boldsymbol{z_j})\right) > 0$

These correctly classified points make up a data set that is guaranteed to be separable as any points on the wrong side of the boundary or on it has been thrown away.

With this pruned dataset we generate a new SVM without using slack variables. This ensures that the new SVM has very nearly the same decision boundary as the original but greatly reduces the amount of support vectors used.

The drawback is that the margin of the feasible solution may be so narrow that it will take a very long time before the QP-solver finds the exact combination of Lagrange multipliers that solves the problem perfectly. Furthermore as the boundary has to has to twist and turn through narrow gaps in the separable data the margin boundary will often have to support vectors thus retaining a large set of support vector.

This can be alleviated by pruning a bit more than just throwing away misclassified points. We introduce a parameter d that defines how far away from the boundary a point has to be in order to be included in training set for the separable case.

We keep all points that satisfy: $y_j\left(b + \sum_{i=1}^{N} \lambda_i y_i \cdot K(\boldsymbol{x_i}, \boldsymbol{z_j})\right) \geq d$

The larger d becomes the quicker the separable case SVM can be generated and the fewer support vectors will be need to be included in the solution.

4.3.6 Calculating multiple SVM

As SVM is a binary classifier will be necessary to combine several SVMs in an ensemble in order to classify multiple classes. This can be done quite handily using the correct kernel and doing computations in large matrices

on

An advantage of the SVM model is to classify an array of observations with respect

5 The algorithm

5.1 Goal

The algorithm is based on an ensemble of binary classifiers. Every classifier separates a set A from a set B, where set A is denoted as positive and B is denoted negative.

By choosing an appropriate combination of classifiers we hope to identify pixels unique to a given tablet. However, the demand for uniqueness must be held against the demand for robustness. Finding truly unique pixels is not of much use if they rarely occur even in the tablet we're looking for.

It is therefore more viable to find pixels particular, but not unique to a given tablet such that response from other tablets is minimized and robustness is maintained.

Since the response of the ensemble classifier cannot be assumed to be unique to a certain tablet we must further asses how the response compares to our hypothesis of no error having occurred.

This means taking into account prior knowledge of what is likely to be in the picture, and finding a measure of similarity between observed and expected response.

As established the primary goal is to achieve good classification without the use of spatial information. This means that identifying a contiguous region of pixels is of no importance.

5.2 Strategy outline

Building a combined classifier has been done this way:

1. pictures are divided into group 1 and 2 with 1 of each type of tablet in each group. Group 1 will be used for training the classifier and it's performance evaluated for on group 2
2. An initial SVM constructed for each class. This one separates the tablet against all other tablets and the background
3. We measure the response of the classifiers response on every tablet. and note the data in a 42x42 matrix with classifier along 1^{st} dimension and tablet along 2^{nd} dimension. in this every row is scaled to the (I,i) element (response of classifier i on the i'th tablet)
4. For each initial classifier we build additional classifiers against any classes that with a response close to the response of the class it was intended to find.
5. A weighted voting system is constructed such that a combined classifier is constructed for each tablet. maximum votesum for every combined classifier is 2 and minimum -2.

6. Each combined classifiers response is measured on every classes. This response is unevenly discrete but is made more or less continuous by making a smoothed histogram
7. Combine votesystem, SVMs, histograms and indexed tablet sizes into a method that takes expected classes and observed spectral data as input.

5.3 Constructing the initial SVM

The initial SVM is a means of discerning which pixels in the class are particular and which are common. It is therefore expected and necessary that this classifier yields false negatives in return for very few false positives.

In classification these features of a classification are called recall and precision[2]:

$$r = \frac{TP}{TP + FP}$$

$$p = \frac{TP}{TP + FN}$$

using bayes theorem we see that this is a conditional probability[3] . we denote event of belonging to class 1 (being positive) as B_1,being predicted as class 1 (predicted as positive) as A_1 and being predicted as class -1 A_2 (predicted as negative)

$$P(A_1|B_1) = \frac{A_1 \cap B_1}{A_1 \cap B_1 + A_2 \cap B_1} = \frac{P(B_1|A_1)P(A_1)}{P(B_1|A_1)P(A_1) + P(B_1|A_2)P(A_2)}$$

Using $A_1 \cap B_1 = P(B_1|A_1)P(A_1)$

Which reads

Probability belonging to set A_1 *given that it belongs to* B_1

Hence recall is

Probability of an observation being predicted to be positive, given that it is in fact positive

And precision is

Probability of an observation being positive, given that it has been predicted to be positive.

In this context we can then say that a high precision implies that pixels predicted as positive are highly particular to the class of interest, whilst low recall implies that they are rare in the class.

[2] Data mining p?
[3] probability

Ideally we can get both high recall and precision, but as this is unlikely we must consider what kind of tradeoff we are prepared to make.

Let's for example say that want to maximize the precision subject to the demand that recall is at least 50%. Using simple programming this is an easy tradeoff to implement. unfortunately it implies that we would accept going from a recall of 100% to 50% just to gain one percent-point in precision or less. This would not be rational, so instead we use a continuous function called the F_β measure:

$$F_\beta = \frac{pr}{\beta^2 p + r}(1 + \beta)$$

When beta is 1 it is called the F_1score and is a harmonic mean[4]. This mean that it harmonizes recall and precision, pulling the score towards of the two and thus penalizing any difference between the two.

The harmonic mean however is not suitable here as we don't value recall and precision equally. To express this we can lower beta and thus reward high values of precision but still penalize difference between recall and precision.

But what value beta do we choose? Let's assume that we would accept a 50% recall if we at least get a 95% precision. This means that our F_β should yield 1 when $p = 0.95$ and $r = 0.5$. finding beta is then just solving a 2nd order polynomial which gives $\beta \approx 0.44$

This is not a lower bound on recall but it means that if precision is above 95% and recall is below 50% we will reward an increase in recall at the expense of an equal decrease in precision.

5.3.1 Parameter tuning

The F-score is a way to determine the performance of a classifier given a tradeoff expressed β. This however is redundant without a way to manipulate the performance of the classifier.

In an SSCA-SVM with a RBF-kernel there is 3 parameters that can be manipulated:

- σ is a parameter of the RBF-kernel and determines how the influence of support vectors is spread out. If it is small the influence of support vectors is very local and strong becoming almost a nearest neighbor classifier towards infinity. This can lead to overfitting. On the other hand a very high value will tend to underfit. Tuning this parameter mostly affect performance.
- d is a pruning parameter that removes points close to the decision boundary from the training data. A high value will remove more points smoothening the decision boundary and tend to underfit whilst a lower might even overfit slightly. The potential positive effect on performance is not significant. but an appropriate value can decrease the number of support vectors significantly with little or no damage on performance.

[4] Data mining

- Threshold controls the decision boundary, shifting it perpendicular to the decision boundary in the transformed space. This directly affects the tradeoff between recall and precision, but will most certainly harm the accuracy.

When tuning parameters cross-validation is used.

When generating an SVM it is always adviseable to subsample the data and balancing the data. Especially imbalances in the negative class will be problematic at this stage.

5.4 Evaluation of overlap

Some classes lie very close to one another and this is almost impossible so account for during the initial SVM where this overlap is drowned in the sheer amount of data. We therefore must evaluate which classes a classifier is particularly responsive to.

5.5 Additional classifiers

If a classifier is too responsive to a class compared to the response of the class it was trained to find it an additional SVM should be generated against this class. In order to be able to reuse classifier the f1-score is the most appropriate here.

5.5.1 Voting system

The voting system is a matrix where a positive element j,i contains the probability of a pixel being in class j given that SVM i has predicted it be positive. And negative element j,i contains the probability of a pixel being in class j given that SVM i has predicted it be negative. Every additional SVM contain one negative and one positive weight. When looking for tablet j every SVM that has a non-zero entry for this tablet will be evaluated for the observed data for each pixel there is now a row of -1 and 1's which is then multiplied with the corresponding weights in

the votesystem. This yields a score between –2 and 2 for every tablet as seen in figure 2

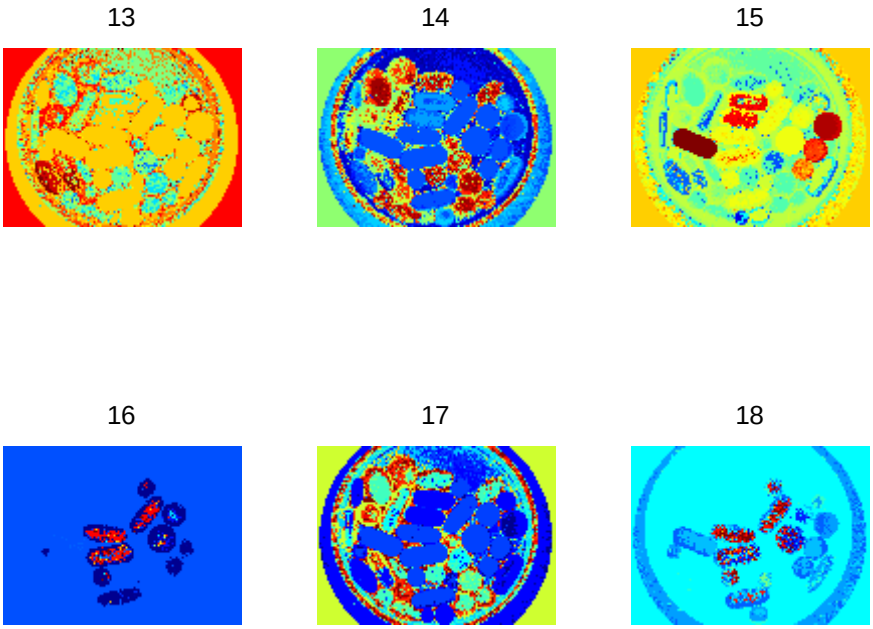


figure 2 combined classifier 13 to 18 used on the u picture containing all tablets

5.6 Histogram

As said the pixels are not necessarily unique to each tablet. but the combination of it's pixels are. The combined classifiers output discrete values where every value is almost unique to a certain combination of positive an negative predictions from the SVMs. A 1d histogram therefore says a lot about the types of pixels in the image and will be more or less the same for tablets of the same type, but especially particular to the tablet the classifier has been built for. This histogram should be made for values between 1 and 2 scaled so it'a area is 1;

This way it will be invariant to changes in tablet area and won't be affected by pixel clearly not particular for the tablet were looking for.

See figure 3.

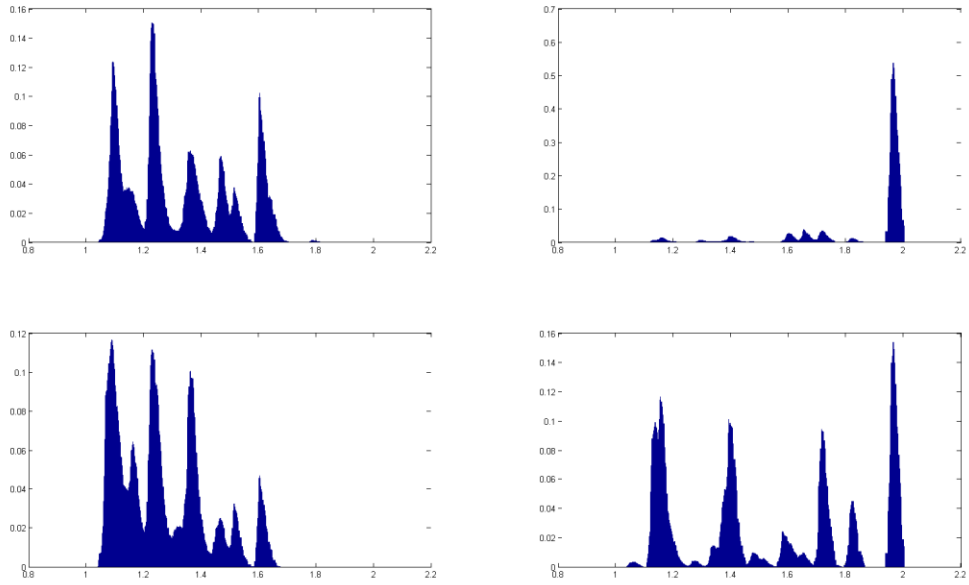


figure 3 histograms over response of classifier i on tablet j. UL i:1 j:1 UR i:42 j:42 LL i:1 J:42 LR i:42 j:1

6 Conclusion

Unfortunately the final results are not documented yet. But my tests with comparing histograms indicate a good performance for detecting error when given prior knowledge of the expected tablets. It is uncertain how these method will respond to upscaling with up 400 different tablets. But it must be stated that a spectral data good deal of discriminatory power that would certainly benefit the industrial application.

[1] Johan A.K. Suykens, and Joos Vandewalle Dries Geebelen, Reducing the Number of Support Vectors of SVM Classifiers, 2010, Submitted to IEEE Trans. on neural networks, march 26 2010, but no included in the issue.

[2] Jorge Nocedal, *Numerical Optimization.*: Springer, 2006.

[3] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, *Introduction to Data Mining.*: Pearson/Addison Wesley, 2006.

208230	B1-vitamin tablet Hvid rund 115	B1-vitamin
208229	B-combin tablet Gul rund 395	B-combin
101378	Cardopax Ret dep 40 mg Hvid 169	Cardopax
147934	Cardopax tablet 5 mg Hvid mrk. 5 55	Cardopax
077670	Cetirizin 10 mg oval hvid mrk C, J, E med delekærv 288	Cetirizin
101825	Creon 10.000 enterokapsel Brun/klar kapsel 201	Creon
210036	C-vitamin tab 500 mg Oval hvid med delekærv 350	C-vitamin
208170	C-vitamin tab 75 mg Rund hvid 270	C-vitamin
211709	D-vitamin 25 mikg tab Rund hvid 324	D-vitamin
210936	D-vitamin 35 mikg Rund hvid 173	D-vitamin
210907	D-vitamin Tabl. 10 mikg rund hvid 9	D-vitamin
133092	Ferro Dur 100 mg Depot Rund lysegul mrk. ADD 39	Ferro
208224	Folinsyre tabl. 0,4 mg Rund gul 185	Folinsyre
076736	Glucosamin tabl. 400 mg Aflang 359	Glucosamin
142323	Haiprex tablet 1 g Hvid mrk. H X 3 M 391	Haiprex
024503	Hjertemagnyl 150 mg Aflang hvid med delekærv 397	Hjertemagnyl
024618	Hjertemagnyl 75 mg hjerteformet hvid 45	Hjertemagnyl
036459	Imodium tablet 2 mg Hvid mrk. L2 152	Imodium
470708	Ipren tablet 200 mg Hvid 163	Ipren
003995	Isodur 60 mg depottablet Aflang lysegul med delekærv mrk. A D 254	Isodur
004017	Isodur depottab 30 mg Aflang lyserød m. delekærv mrk. A-II 161	Isodur
796821	Jern C tablet Grålig 56	Jern
037853	Kaleorid depot 750 mg Aflang hvid 388	Kaleorid
491787	Kinin tablet 100 mg Hvid 367	Kinin
510586	Kodimagnyl i.s. tablet Aflang hvid med delekærv 0	Kodimagnyl
061687	Kurazid 150 mg Rund gul med delekærv 111	Kurazid
206269	Longo Vital Classic Brun tablet 50	Longo
235226	Mablet tablet Aflang hvid 298	Mablet
004676	Magnesia tablet 500 mg Rund hvid 150	Magnesia
519371	Magnyl 100 mg entero Rund hvid 314	Magnyl
212093	Multi-tabs 50+ Oval gul 368	Multi-tabs
211833	Multivitamin tablet Rund gul 53	Multivitamin
249243	Natriumklorid enterot Gullig tablet 165	Natriumklorid
083644	Panodil tabl. 500 mg Hvid mrk. Panodil 246	Panodil
383745	Paraghurt tablet Rund hvid 336	Paraghurt
502336	Perilax entero 5 mg Rød 135	Perilax
166611	Phenergan tablet 25 mg Rund blå mrk. PN 25 219	Phenergan
056250	Pinex 500 mg tablet Hvid mrk. PINEX 500 AL 197	Pinex
207660	Unikalk Forte tablet aflang hvid 293	Unikalk
203387	Unikalk Plus tablet Aflang hvid mrk. Unikalk 358	Unikalk
207898	Unikalk Silver tablet Aflang hvid 351	Unikalk
759647	Vitabutin Tranebær kap Rød stor kapsel 245	Vitabutin
236935	Zinklet tablet Rund hvid 180	Zinklet