Universiteit Leiden
The Netherlands

Opleiding Bioinformatica

Age Classification of Zebrafish Larvae

using Machine Learning from HOG Features

Hermes A. J. Spaink

Supervisors:
Prof. Dr. Ir. Fons J. Verbeek
Dr. Kristian F. D. Rietveld
Leon S. Helwerda

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)

Abstract

Zebrafish (*Danio rerio*) are important and widely used model organisms in biological and biomedical research. Imaging is key for studying of the embryonic and larval stages of zebrafish. The dorsal, ventral, and both lateral views of zebrafish acquired through vertebrate automated screening technology (VAST) imaging are used for age prediction of larvae of 3, 4 and 5 days post fertilisation. Histogram of oriented gradients (HOG) is used as a feature extraction method on these images and a machine learning model is trained to perform a classification on this data. We show that a linear support vector machine achieves an accuracy of 97% in classifying the age. A distinct image processing, however, including filtering, image smoothing and feature selection is essential to achieve an adequate classification.

Contents

Chapter 1

Introduction

Zebrafish (*Danio rerio*), a freshwater fish native to the Himalayan region, is a widely used vertebrate model organism in biological and biomedical research. [1] Being one of the most important model organisms in developmental biology, it is increasingly used in toxicology, pharmacology and behavioural studies. [2] Moreover, there is a large resemblance of human clinical disorders in zebrafish mutant and pathogen induced phenotypes. [3] Being such an important research organisms, many studies have focused on the imaging of zebrafish, especially during its larval stages. An aspect of zebrafish is the almost complete transparency during its embryonic stage which is a great advantage during imaging. Imaging is crucial as it provides information on the (defect) phenotype of the specimen studied. A wide range of screening technologies exist specialised in studying zebrafish. [4] Among these is the vertebrate automated screening technology (VAST) capable of high-throughput *in vivo* imaging of small specimens such as zebrafish embryos. [5, 6]

During a VAST cycle the system images zebrafish larvae by loading them from either a multiwell plate or a reservoir, and after detection, passing them through a capillary tube and, positioning and rotating them. [5] This rotation capability allows for imaging from many different angles. Digital reconstructions into a three dimensional (3D) volume is then possible [7] which is crucial to get a better understanding of the structural architecture of biological samples and get a better insight into a disease or defect phenotype. [8] A VAST microscope is also present at the Institute of Biology Leiden (IBL). With this apparatus a turntable sequence of 84 two-dimensional (2D) axial-view images are generated. [9] An annotation pipeline reconstructs these images into 3D shapes and, more importantly, annotates them with various properties such as volume and surface area. [10]

As organisms get older their phenotype changes. For example they grow larger or develop new organs. A study by Guo et al. showed the distribution of volume and surface area of 3, 4 and 5 days post fertilisation (dpf) wild type (wt) zebrafish embryos. Figure 1.1 indicates the average size increases with maturation, but it also shows that a lot of overlap is present in these features. [10] Using the descriptors *volume* and *surface area* of the 3D reconstruction of the imaged zebrafish larvae is therefore not a good indicator for age. Another way of predicting maturation of the larval stage is using an image based identification strategy. This may be more distinct and hence provide more insights in what phenotypic aspects of the embryo are defining for age. Various development effecting mutations or pathogens can have an effect on the growth of an organism. A good maturation identification can be very helpful in understanding how certain diseases, toxins or mutations may have an effect on age through effecting these phenotypes.

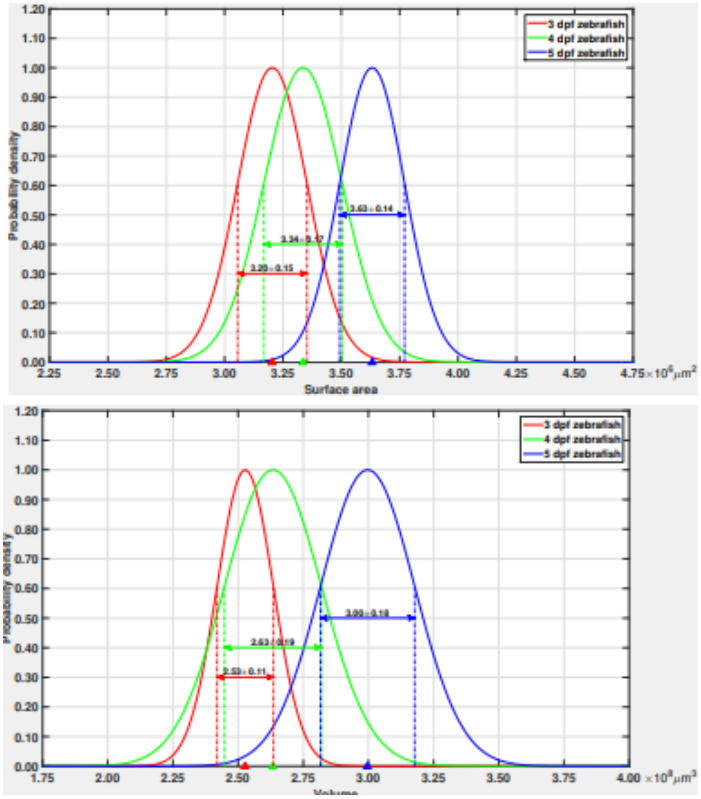


Figure 1.1: Distribution of surface area (top) and volume (bottom) of 3, 4 and 5 dpf old zebrafish larvae. Y axis: normalised probability density. The coloured triangles on the x axis indicate the standard deviation from the mean for each of the distributions. [10]

1.1 Tools for age prediction

A means for making predictions based on data is machine learning. Machine learning is the process of training a computer or machine on a data set in order to make estimations for future samples of the same data type. [11] A type of machine learning is the process of classification. During classification a trained machine learning algorithm (classifier) tries to predict a certain aspect (the class) of the sample. Many different types of classifiers exist, all with different properties, advantages and disadvantages. Machine learning is a powerful tool used in image classification and is used widely in many research areas such as artificial intelligence, biology and security. In this project a classifier will be trained on image data of zebrafish larvae of 3, 4 and 5 dpf in order to provide an estimation for the maturation of new samples. During training it is especially interesting to note what happens to wrongly classified specimens as this may provide information on phenotypes which are characteristic for maturation.

A classifier cannot be trained on the raw input image data. A collection of red, green and blue (RGB) values for all the pixels does not give any information about what is actually in the image. Therefore a so called feature extraction has to be performed in order to characterise the image. There are multiple approaches to this of

which the two major ones are feature engineering and using convolutional layers. The latter one creates its own form of feature extraction and is used mainly in combination with deep neural networks. This approach, however, requires large amounts of training data. In this project a custom feature engineering will be applied to the zebrafish images based on the Histogram of Oriented Gradients (HOG) method first described by McConnell et al. in 1982. [12] HOG provides a measure for gradients in images and is a good tool for edge detection. Therefore, HOG features are commonly used for object detection as it provides a good descriptor for shapes.

1.2 Problem statement and research questions

In the previous section a brief introduction has been given to the research done in this project. A descriptor for age, or maturation, in zebrafish embryos is necessary in order to provide better insight into what phenotypic features are defining for this. This led as motivation to our problem statement (**PS**):

> **PS:** *To what extent can we estimate the age of zebrafish larvae based on features extracted from imaging data using machine learning?*

To address this **PS** we are inclined to train a classifier on images of wild type zebrafish larvae. A feature extraction method will have to be designed which gives a good indication of phenotypic aspects relevant to age. This will have to be applied to zebrafish image data which will have to be preprocessed and normalised. A fitting classifier will have to be selected and trained, and parameters of this classifier will have to be optimised. Furthermore speed is of relevance in order to integrate the age prediction into the VAST zebrafish annotation pipeline. To reach our goal we formulate and investigate four research questions (**RQs**).

As mentioned above and in the previous section a feature extraction method is necessary in order to accurately train a classifier. Volume and surface area of zebrafish larvae provide an ambiguous descriptor for age. Histogram of Oriented Gradients (HOG) is a feature extraction method that might fit our goal for shape detection. Previous research has shown it has potential for bioimage analysis and identification of subtle details as well as the possibilities for age classification of zebrafish embryos. This will be investigated further and our first research question is as follows:

> **RQ 1:** *To what extent are HOG features a good indicator for maturation in zebrafish and how does this compare to using volume and surface area?*

The image data collected from VAST microscopy consists of 84 RGB images of 1024 by 250 pixels stored in a single bitmap file per sample. This data has to be processed to be suitable for feature extraction. There are various approaches to this. We will investigate these in order to define an optimal preprocessing pipeline which is fast and accurate. Our second research question is thus formulated as follows:

> **RQ 2:** *How can we perform and optimise the preprocessing of the zebrafish image data?*

Various algorithms for classification exist. Image data is high dimensional, dense data. Classifiers such as a

support vector machine (SVM) or an artificial neural network (ANN) might be well suited for this type of data and commonly used in image classification. Both types of classifiers have advantages and disadvantages. SVMs are, for example, much faster to train and less prone to overfitting (depending on the kernel). ANNs on the other hand are much more flexible in the way they interpret data. Because these two classifiers have shown their potential in previous research they will be compared. The requirements for a good classification is a high F1-score and a short classification time. This leads to our third research question:

> **RQ 3:** *How does classification using an artificial neural network compare to using a support vector machine?*

In the previous section an outline of the current annotation pipeline has been given. The aim is to integrate an age annotation into this pipeline and for that reason speed is essential. The pipeline runs on a computer cluster capable of running many processes in parallel. We will also pay some attention to parallelising computational expensive parts if necessary. We therefore define our fourth research question as such:

> **RQ 4:** *To what extent can we optimise parameters of the classifier and how can we optimise the run time of classification?*

1.3 Thesis Overview

In this section the structure of this thesis is outlined. Chapter 1 provides a brief introduction to the research described in this thesis. It presents an overview of the problem statement and the research questions.

Chapter 2 describes background theory on methods used in this thesis. It contains brief descriptions of techniques and how they can be applied. Furthermore it provides a short survey of other studies focusing on related topics. Here we discuss methods which will possibly be influential for our approach as a basis for why certain decisions and considerations have been made. Moreover we provide a background of research which led as motivation to this work.

Chapter 3 contains the materials and methods section. Here we elaborate on the software and data, our specific implementation, and what steps were taken to achieve results.

In Chapter 4 an overview of the results will be given alongside a discussion of these results.

In Chapter 5 we present our conclusions from the research and provide an overview of our findings. The answers to the research questions and the problem statement are summarised and a glance at possible future research is given.

Chapter 2

Materials & Methods

In this chapter a description will be given on techniques and methods applied in our research. An overview of how these are put in practice is given in Chapter 3. Here we will provide background knowledge required for a full understanding. Furthermore, at the end of the chapter a brief survey of related works is given.

2.1 Zebrafish

Zebrafish (*Danio rerio*) is an important model organism in biological and biomedical research being commonly used in toxicology (the study of the effect of toxins), pharmacology (the study of the effect of medicines and other pharmaceutics), behavioural studies, developmental biology and genetics. [1,2] There are several reasons for its popularity. The first is the key point of the large resemblance of response to diseases by humans. In the past mainly mice (*Mus musculus*) and fruitflies (*Drosophila melanogaster*) were used for understanding human disease. Not only do they resemble human mutant or defect phenotypes but their key superiority's over other animals are their quick reproduction rate and small size which allows for simple screening. This also goes for zebrafish but there are some other benefits as well such as *ex vivo* fertilisation and embryogenesis, optical transparency of embryos and larvae, fast embryonic development, and relative cheap housing housing costs. [13] Besides this, the genetics of zebrafish has been studied extensively causing thorough knowledge of its genetic markup. From mutagenic screens many mutants have generated with defects that are analogous to human genetic diseases at the molecular and cellular level. Next to genetic disorders, zebrafish also prove to be good models for several acquired diseases among which are tuberculosis, diabetes and cancer.

During its embryonic (0-72 hours) and larval stages (3-29 days) zebrafish are almost completely optically transparent. This allows for easy light microscopy imaging. Pathogens can, for example, be labelled with fluorescence and possibly that way traced through the organism. Along with their small size, about 3.5 mm at 3 days past fertilisation (dpf), they are easily studied. [14] During the embryonic stage rapid development takes place of cell division to formation of fins and organs. During the larval stage a lot of development occurs as well. However, it is more difficult to define developmental stages at this point in time as environmental

factors such as temperature, population density, and water quality have much more effect on growth. The phenotype of zebrafish larvae is thus subject to a certain amount of individual variation. [15, 16]

2.2 Imaging

Bio-imaging is defined as the process of acquiring images of biological samples. There are numerous methods for this often through microscopy. Popular imaging tools in biology include electron microscopy, confocal microscopy and light microscopy, all of which have their own specific variations and appliances to fit the required needs as best as possible. More recent developments have been in the field of three-dimensional (3D) imaging. This is the process of scanning the sample from multiple angles or at multiple layers and reconstructing the so obtained 2D planes into a 3D volume. Confocal microscopy can do this at very high magnifications (up to 63x) with a high resolution but is time-intensive, moreover because of its high magnification it can only be used to study small (areas of) samples (at 63x a scan area of 246.03 μm^2). [17] Other techniques for 3D imaging include magnetic resonance imaging (MRI), optical projection tomography (OPT), and vertebrate automated screening technology (VAST) bio-imaging. Last named bio-imaging system will be used in this study as an image data acquisition method through axial scanning of zebrafish.

2.3 Python

All data processing, the classification and implementation for this research was done in Python 3. Python is a widely used high-level programming language designed by G. van Rossum in the late 1980's. [18] The first version of Python was released in 1991 with the design philosophy of code readability. Python 3 was launched in 2008 and implemented several major revisions to the language. Its versatility and portability along with the advantage of speed and good possibilities for data science made it an excellent choice to use as implementation for this project.

To extend the base functionality of Python, additional packages (libraries) were used. The library NumPy allows for more complex mathematical operations, data storage and data handling. The library scikit learn is a widely used data science toolkit containing implementations of many classifiers and analysis methods.

2.4 Data science

As the name suggests data science is the application of techniques from computer science and statistical science to data sets. This data can have many forms and shapes and can be very unstructured. There are two main paradigms within the field of data science: question-driven, and data-driven. The first, question-driven, is common in the social sciences and starts with a research question after which an experiment is set up and the required data is collected. This data is then analysed and the hypothesis to the research question is then either

accepted or rejected. The second paradigm, data-driven, starts with the data. The data is explored and from this a research question is formulated. In order to answer the question the data is structured and annotated so a machine learning approach can be applied which is then evaluated on the data.[1]

The research presented in this thesis is without doubt a data-driven type data science study. The data consists of zebrafish image data collected through VAST microscopy from which the goal is to derive the age of the larvae. As mentioned above, it is general practice when working with data to process it through various steps before being able to make some predictions based on the data. In the following paragraphs these steps will be elaborated specifically for the techniques applied in this research.

2.5 Image processing

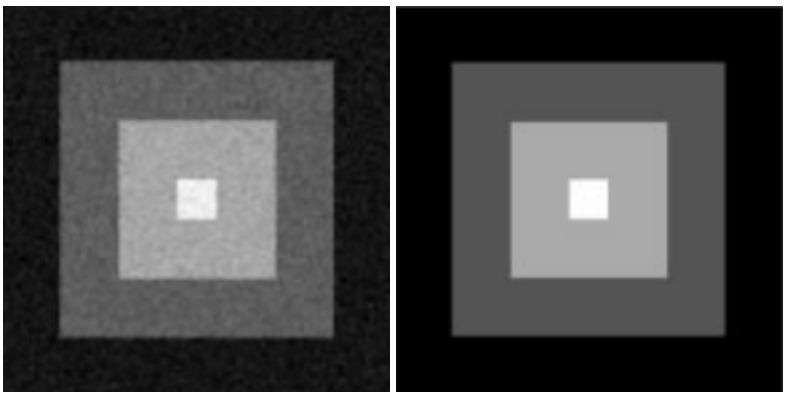
In order to make accurate and relevant predictions based on data, this data usually first has to undergo a distinct stage of processing. This includes structuring and annotating the data, cleaning the data of useless or empty entries, and extracting representative values or structures from the data through which one can answer the research question. For image based data a good image processing stage is therefore critical in order to make proper predictions, which requires thorough knowledge concerning the contents of the data. Image data is a quite specific type of data when it comes to data processing. As images usually contain a lot of information, thousands to millions of pixels, the relevant parts usually undergo a cropping and scaling stage to remove unimportant parts of the image or reduce the size. Moreover feature extraction from images is key for detection of what is actually in the image.

Preprocessing of images is the task of extracting parts which describe its contents. This usually involves cropping, scaling, filtering, normalisation, segmentation, and object identification in order to output a set of significant regions and objects. [19] From the preprocessed image features are extracted which are used in machine learning, where a computer learns patterns from the data and can then make predictions on new data of the same type. Extraction of normalised identifiers from a image is critical in image recognition, identification and classification. A mere collection of RGB intensity values says little about the actual contents of the image. Moreover it is essential to take the context of each pixel into account. A good method for feature extraction is critical because the particular features directly influence the efficacy of the classification task. In image classification features are stored in a feature vector which contains all ”relevant” information describing the image, or the object in the image. This vector should be considered a mathematical abstraction of the image, but also a function of one or more measurements. [19]

2.5.1 Image blurring

The HOG feature extraction which is applied in the research presented in this thesis is focused on edge detection. This edge finding is based on inter-pixel gradients (intensity variations) which is explained in

[1] Lecture series *Data science 2018* - Dr. S. Verberne



(a) (b)

Figure 2.1: Bilateral filter result after ten iterations. (a) Noisy input test image and (b) result after filtering. *Image source: M. Elad* [20]

more detail in the next section. Gradient detection, however, can be negatively impacted by noise present in images and especially photos. A blur or smoothing filter can therefore be applied before gradient computation. Various types of blurs exist such as a mean blur, Gaussian blur, or bilateral filtering. A mean blur weighs all surrounding pixels in the given radius r equally. A Gaussian blur weighs surrounding pixel intensities according to a Gaussian, or normal, distribution. For that reason it requires the user to specify a standard deviation σ for the pixels to be considered. Compared to a mean blur, a Gaussian blur has the advantage of better edge preservation. A third type of blur is a bilateral filter (Figure 2.1). This blur has even better edge preservation as it takes colour value and spacial information into account. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. [20, 21]

2.5.2 Histogram of Oriented Gradients

The Histogram of Oriented Gradients (HOG) is a feature descriptor able to accurately identify objects in an image. After the preprocessing stage a HOG feature extraction is applied to the zebrafish images. Previous studies have shown the potential for HOG as an identifier for people or object detection, and its advantage of great dimensionality reduction. Images are a collection of pixels, a standard image can contain millions of pixels which are a lot of features if directly used for training and classification. For example the cropped and scaled zebrafish images have a size of 704 by 112 pixels, which is a total of 78848 pixels. Every pixel has a RGB value, which is a combination of the colour intensities in the RGB (red, green, blue) colour space. Nowadays it is the standard that every colour can have a value in the range of 0 to 255, which corresponds to 8 bits (1 byte) of information, as $2^8 = 256$. For the total image this would thus result in $78848 * 3 * 8 = 1892352$ bits of information. HOG reduces the size significantly and is therefore a compact way of representing the image. The HOG feature vector of such an image has a file size about 12 times smaller per image. However, because the files undergo compression this is not an explicit measure for the theoretical size reduction of an image to its HOG vector.

Dalal and Triggs [22] propose a method for the calculation of the HOG vector from an image. Preprocessing of

the input image results in a region of interest on which the feature extraction is then performed (Figure 2.2a). First colours have to be normalised which can be done by computing the grey scale image, RGB or LAB (CIELAB) colour spaces. A grey scale image is computed by taking a weighted average of the three RGB colour channels. This preprocessed sub-image is then divided into a number of equally sized groups of adjacent pixels called cells (Figure 2.2b). For every pixel in the cells their directed gradient is then computed (Figure 2.2c) according to the following formulas:

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

Where θ is the direction of gradient with magnitude g. This is computed using a 1-D mask with g_x kernel: $[-1, 0, 1]$, and g_y kernel: $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$. This means the gradient at a pixel will be the sum of the surrounding pixel intensities multiplied by the values in the mask.

After gradients for all the pixels in a cell have been computed every pixel adds its gradient magnitude to an edge orientation histogram based on the orientation of its gradient element (Figure 2.2d). The magnitudes are summed into a number of bins corresponding to the gradient directions. These orientation bins are evenly spaced over $0° - 180°$. The magnitudes are divided bilinearly over the neighbouring bins to reduce aliasing for the orientations resulting in a histogram. Figure 2.2d illustrates how the gradient magnitudes are placed in the histogram determined by the gradient direction at their corresponding position.

Especially in non standardised images, pixel gradient strengths vary significantly due to variations in illumination and foreground-background contrasts. E.g. in outdoor camera pictures natural lighting varies at every moment. To compensate for this all cells in the HOG descriptor are normalised using a block normalisation scheme. One block contains one or multiple cells causing the normalisation to take gradient variation of various cells into account, thereby accounting for illumination differences between pixels in different cells. The size of the input image has to be a multiple of the block size which is also the reason for the image scaling step as mentioned in Section 3.2. However, a large scaling factor would have a considerable impact on the data by possibly causing distortions. For that reason, ideally, a block and cell size is chosen which fits the cropped image with the least amount of scaling necessary. This is determined by choosing a block size which is close divisor of the image width and height.

The histograms are normalised using the *L2-Hys* normalisation scheme based on the *L2-norm*

$$\mathbf{v} \rightarrow \mathbf{v} / \sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}$$

where $\mathbf{v}$ is an unnormalised descriptor vector (the combined gradient histogram for all cells in the block), $\|\mathbf{v}\|_2$ its Euclidean norm as given by $\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}_1^2 + \mathbf{v}_2^2 + ... + \mathbf{v}_n^2}$ for a vector $\mathbf{v}$ of size n, and ϵ is a very small constant. The *L2-Hys* follows the *L2-norm* by limiting the maximum values of $\mathbf{v}$ to 0.2, and then renormalising if required. [22]

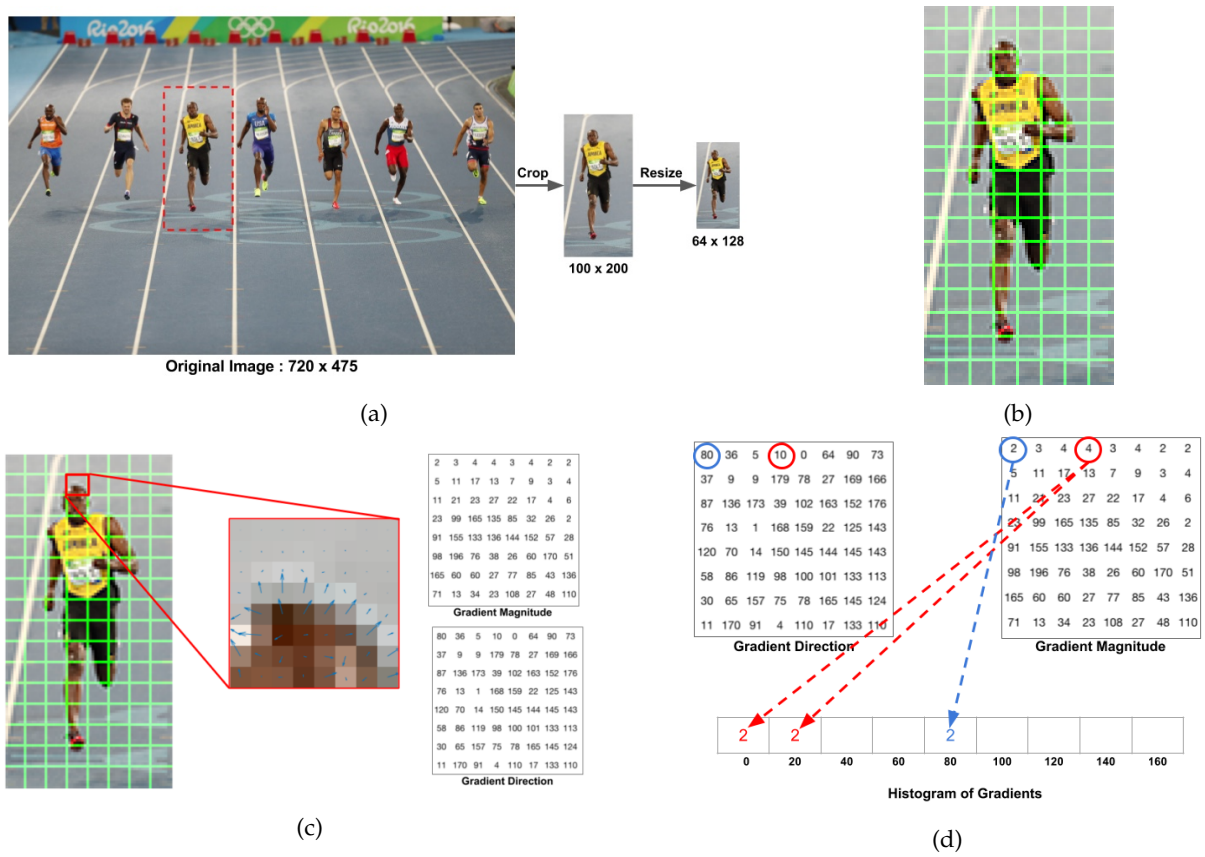


Figure 2.2: Processing steps in HOG feature extraction. (a) Preprocessing of the image. This can include identifying a region of interest, cropping and scaling, and possibly applying a cropping mask. (b) Division of the preprocessed image into cells of equal size. (c) Calculation of gradients per cell. Gradient magnitude and direction is calculated for every pixel. (d) Determination of Histogram of Oriented Gradients per cell. Afterwards normalised using block normalisation. *Image source:* `www.learnopencv.com/histogram-of-oriented-gradients`

2.5.3 Dimensionality reduction

Although feature extraction as discussed in the previous section already reduces the size of the image description, the feature space (the features representing a sample) is still quite large. HOG can reduce tens of thousands of pixels to a couple thousand features but also includes redundant features better left out for classification. This can reduce the complexity of the data, improve performance by reducing the risk on overfitting (the issue where the model is shaped too much to the training data as in Figure 2.3), and gives better interpretable information. The problem then arises, how to represent the high-dimensional input in a low-dimensional feature space? [23, 24]

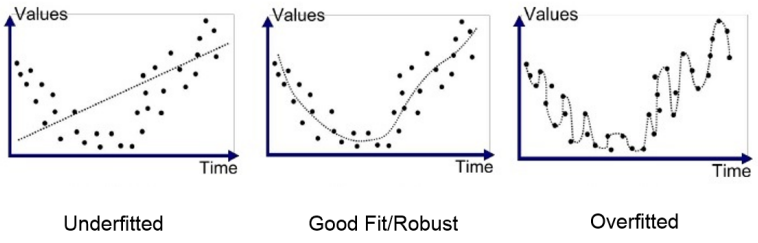


Figure 2.3: Illustration of three models represented by the dashed lines fitted to the same training data represented by the black dots. The underfitted model does not give a good prediction because it does not represent the data. The overfitted model does not give a good prediction as it is not a good generalisation of the data, unseen data will probably not be estimated correctly. *Image source:* `medium.com`

Various methods for dimensionality reduction exist. Traditional ones such as principle component analysis (PCA) only select features which are uncorrelated with each other and drop the connected features. Another method is univariate feature selection which works by selecting the best features based on univariate statistical tests. Every feature undergoes individual statistical analysis to detect variation in the feature across the complete dataset.

2.6 Classification

Classification is a type of machine learning where an algorithm is trained on existing data in order to predict a class or type for a new entry of the same data type. Machine learning, a sub-field of artificial intelligence, is the process in which a given standard algorithm 'learns' to predict outcomes based on training data without being explicitly programmed or designed for this. The term machine learning was coined by A. Samuel in 1959 who also notes the philosophical aspects to whether computers can learn. [11] The last few years machine learning has become quite a *hot topic* because of the large amount of easy accessible data available which is often referred to as "big data," but mainly because of the new advances in computing power in CPUs, GPUs and memory arrays. This significantly improved the speed of data handling adding to the ease of processing big data.

Large companies such as Google and Facebook collect huge amounts of personal data which is used for creation of personalised advertisements. This is a goldmine for data mining and machine learning plays a major role in this. But that is definitely not its only utilisation. In science machine learning is not just studied but also applied for making new discoveries or optimising certain processes. Because of the non-specificity of machine learning its applications are numerous.

Three distinctions can be made in machine learning concerning the type of learning. (1) Supervised learning, (2) unsupervised learning, and (3) semi-supervised learning. [25, 26]

1. The goal of supervised learning is to train a model to learn a function $y = f(x)$. In other words once the model is trained on a subset of data X, it should be able to predict a class or value y corresponding to any given $x \in X$. This requires, however, all the training data to be labelled with some type of class or value for every entry, which is used as template during the learning stage.

2. Unsupervised differs from the previous type in the sense that there is no defined correct answer. New data is not used to predict a class or value for new data types, but rather to explore existing data such as clustering it into groups based on similarities in the data. It is very useful for finding patterns or "hidden" information in the data.

3. Semi-supervised learning falls somewhere in between the two previous named types of machine learning in the sense that there is a data label present, but not always the complete answer is given to the machine learning algorithm. Or labelled data is used in combination with unlabelled data, which has proved to give considerable improvement in learning accuracy.

Supervised learning can be subdivided into two groups as well: classification and regression. As mentioned above, classification is the task of predicting a class or type given a new sample. Regression on the other hand predicts a value. The main difference between these two is that a regression estimator can predict any value for a given input, whereas a classifier is limited to a number of classes defined during training. In this study we will focus only on classification as we will predict a distinct age category.

Many algorithms for classification exist, along with a diverse range of platforms for implementation (e.g. Scikit learn, Weka). The reason for the variety in algorithms is the variability in data such as shapes and sizes (the number of features), the density over the features (e.g. all features have a value, or some are zero), and the content can be very different (e.g. image data compared to social media data). This and many more factors including the amount of training data influence the choice for selecting a proper classifier for the experiments. In the sections below the two classifiers used in this project are discussed along with with a brief background.

2.6.1 Support vector machines

Support vector machines (SVM) are models used for supervised learning. After training on a dataset, the algorithm can make predictions for new data points. This can be through classification or regression. During training SVMs represent the data entries as points in space and divide the examples of the different categories by a clear gap that is as wide as possible. The defined boundary is a hyperplane in the feature space of the training data. Figure 2.4 shows an abstracted example of a SVM, the hyperplane H_3 divides the data in the most optimal way, meaning it has the largest minimal distance between all data points.

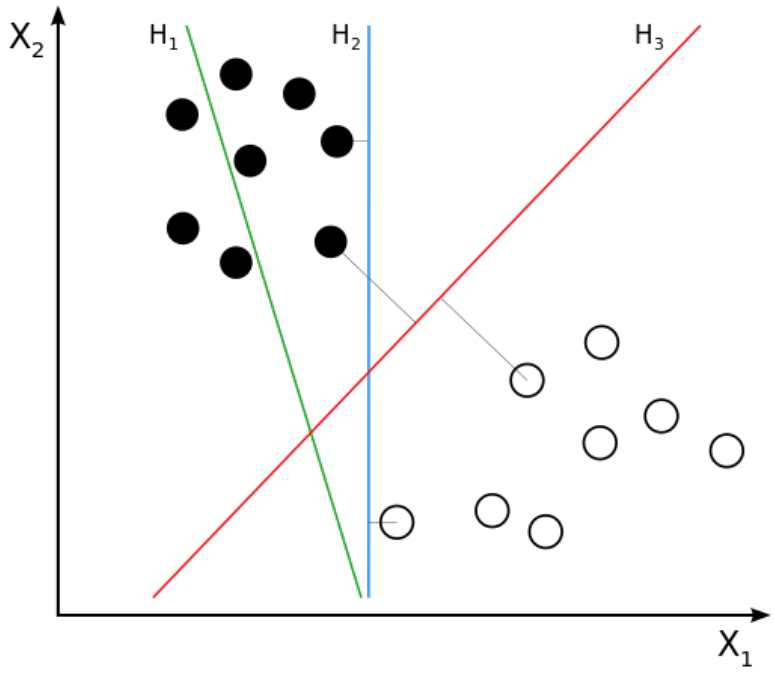


Figure 2.4: A linear support vector machine finds the best hyperplane in a spacial representation of the training data. H_1 does not divide the data, H_2 divides the data not optimal, H_3 has the largest minimal distance between all points and is thus the best fitting divider. *Image source:* `https://en.wikipedia.org/wiki/Support_vector_machine`

The SVM shown in Figure 2.4 is a linear SVM, it finds linear (flat) hyperplanes as division between the data points. Using a different kernel it is possible to also make non-linear separations. Figure 2.5 shows how a different kernel allows for non-linear separation of the data points.

The main advantage of SVMs are their ability to handle small data of about 20 samples, but to also work with large data. Furthermore their ability to handle high dimensional data even when the number of dimensions is higher than the number of samples, their versatility due to varying kernel functions, and their memory efficiency. Disadvantages, however, include that if the number of samples is much lower than the number of features they can be prone to overfitting (Figure 2.3), this can be regulated by careful selection of a kernel. Besides this, SVMs do not provide probability estimates, these are calculated using expensive five-fold cross-validation scores.[2,3]

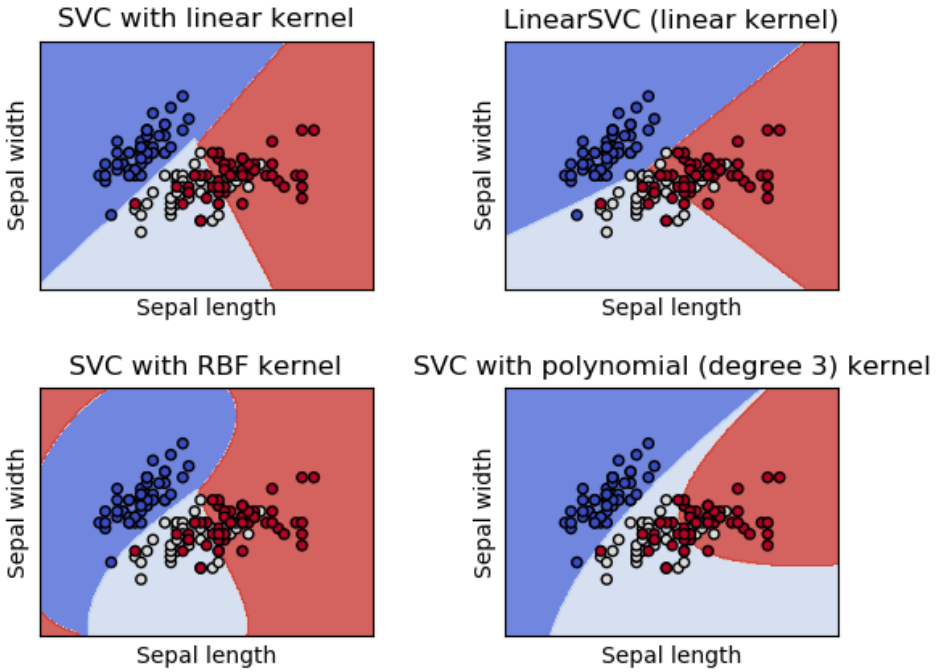


Figure 2.5: Overview of data point seperation using different kernels in a support vector classifier (SVC). `LinearSVC` is another implementation of the `SVC with linear kernel`. A different kernel results in a different separation plane for the data points. *Image source:* `http://scikit-learn.org/stable/modules/svm`

2.6.2 Artificial neural networks

Artificial neural networks (ANN) are based on the principles of neuronal networks present in animal brains. An animal acquires knowledge on doing tasks by performing them and evaluating the result. This led as an inspiration to the perceptron, the building block of ANNs. This concept was first proposed by Rosenblatt in 1957 [27]. ANN is considered a supervised learning algorithm which maps an input to an output through

[2]Lecture series *Data science 2018* - Dr. S. Verberne
[3]Scikit learn *Support vector machines*, `http://scikit-learn.org/stable/modules/svm` [last accessed 01-07-2018]

various mathematical operations. Such that $\mathbb{R}^m \rightarrow \mathbb{R}^o$, where m is the number of dimensions for the input and o the number of dimensions for the output. Given a set of features $X = \{x_1, x_2, ..., x_m\}$ and a target y it can approximate a non-linear function for classification or regression. ANN is composed of so called perceptrons which can be layered. A perceptron is the most basic form of an ANN. It contains a number of inputs which are scaled by individual weights, the sum of which passes through an activation function (such as the sigmoid-function or the rectified linear unit) and then results in a certain output. The input layer consists of perceptrons $\{x_i | x_1, x_2, ..., x_m\}$ representing the input features. The output layer transforms the values from the last hidden layer into output values. Hence there are several parameters for an ANN including the number of hidden layers, the size of the hidden layers, and the activation function.[4]

ANNs have the advantage of being capable to learn non-linear models and also the ability to keep learning whilst being used. Moreover, they are very versatile and can handle high-dimensional data. Especially their modern update, deep learning (an ANN with many hidden layers and nodes), can be applied to many problems without the need for expert knowledge on the data because the feature engineering step is skipped. The downside of this, however, is the danger of overfitting (Figure 2.3) on the training data. Over-training is common and especially risky with a small training dataset. Another disadvantage is ANNs are not always optimal, as hidden layers have a non-convex loss function where there exists more than one local minimum. Therefore random weight initialisation can lead to inconsistent results.

2.7 Result interpretation

After training, a classier can make predictions about unseen data, however, how accurate are these predictions? Training a machine learning algorithm is done on a dataset which is a sample of all possible data, it is therefore important that the sample is representative for the complete dataset. In our case this means the training data should contain a general overview of all possible phenotype in wild type zebrafish larvae. If this is not the case, the trained algorithm will not make optimal predictions on any new data. For that reason it is critical to test the trained classifier with a separate dataset which is not used for training.

2.7.1 Cross-validation

Cross-validation is a method for training and testing a machine learning model to get an estimate of how well it will perform in practice. The training data is split up into k parts and the algorithm is trained k times. Every time one of the data parts is used for testing the model which is trained on the remaining data. This results in accurate metrics for the quality of the classifier as it is trained and tested over the whole dataset. It should be noted, however, that increasing k will cause the training and evaluation process to take significantly more time. For large datasets it is therefore common to choose a $k = 5$ for instance. For smaller datasets n-fold cross-validation (leave-one-out) is a good metric for performance quality. Here every individual item in the

[4]Scikit learn *Neural network models*, `http://scikit-learn.org/stable/modules/neural_networks_supervised` [last accessed on 02-07-018]

dataset is taken as a split for testing, resulting in more training data but less generalised testing. In general it is essential for every split to have a training and testing containing a representative distribution of all classes. [28] The dataset used in this research consists of three classes (3, 4 and 5 dpf) with 12 samples for 3 dpf and 24 samples for the 4 and 5 dpf class. A 5-fold cross-validation results in a decent distribution over the class.

2.7.2 Accuracy and F1-score

The cross-validation results in predictions for the training data. In the case of classifications, the predicted classes can be compared with the actual classes. This delivers a confusion matrix containing the number of true positives, false positives, true negatives, and false negatives, as shown in Table 2.1. The wrong cases (false positives and false negatives) differ in their effect depending on the factor that is being predicted.

Table 2.1: Confusion matrix containing predicted and actual values after training and testing of a dataset. TP, FP, TN, and FN correspond to true positives, false positives, true negatives, and false negatives.

		Predicted	
		True	False
Actual	True	TP	FN
	False	FP	TN

There are various metrics providing the quality of the classifier. The most straightforward is the accuracy, the percentage of correct decisions, which is given by $\frac{\text{Correctly classified cases}}{\text{Total number of items}} = \frac{TP+TN}{TP+FP+TN+FN}$. However, in most cases of binary, multi-class or multi-label classification, accuracy is not suitable because the classes are often unbalanced. This means, high accuracy in one class may mean low accuracy in another. Good alternative metrics are the precision, recall, and their combined F1-score. [29]

The precision of a classifier corresponds to the proportion of the assigned labels which are correctly classified. The precision p for the *True* class is given by the following formula: $p = \frac{TP}{TP+FP}$. The recall of a classifier corresponds to the proportion of correct labels that are assigned. The recall r for the *True* class is given by the following formula: $r = \frac{TP}{TP+FN}$. These metrics can be extended to include multiple classes besides true and false. An average of precision and recall is given by the F1-score. This is the harmonic average where $0 \leq F1 \leq 1$. The F1-score is given by the following formula: $F1 = \frac{2*p*r}{p+r}$.

2.8 Related Work

In this Section related studies will be discussed which have different uptakes to a similar problem, or a similar uptake for a different problem or application. Our research can be divided into two parts: (1) creating an accurate and reliable feature extraction method, and (2) training a classifier to get a good prediction rate for new zebrafish data. Here we give a brief survey of other studies related to either (1) or (2).

(1) Histogram of Oriented Gradients feature extraction from biological samples

Histogram of oriented gradients (HOG) is an image feature extraction method related to texture recognition. It is commonly used for human detection in images. Dalal and Triggs [22] show that HOG descriptors are significantly superior to other edge and gradient detection based methods for human detection. They display that a fine-tuning of the parameter settings for the feature extraction is essential to get accurate results. They report on a well performing preprocessing stage and parameter values for human detection. This provides a solid basis for other detection purposes.

Dahmane and Meunier [30] propose a system for emotion recognition in human faces. Automated facial expression analysis is important in human computer interaction (HCI), video analysis, inteligent interfaces and clinical research in order to figure out the main element of facial human communication and for tracking user experience. A main issue in this is, however, variability due to environmental changes, appearance variability under different head orientations, and non-rigidity of the face. A "baseline" method using Uniform Local Binary Patterns with 8 neighbours and radius 1 to extract appearance features, principal component analysis to reduce dimensionality, and classification with support vector machines (SVM) with radial basis function kernels achieved a performance of 56%. From experiments it was found that using a HOG feature extraction with a variable cell size dependent on the image size gave a significantly better result; the preprocessed image was divided into 8 rows and 6 columns with an adaptive grid-size depending on the distance between the two eyes. Classifying using the HOG feature set had a performance of 70%. This performance was explained as being specifically due to the HOG global characteristics that are based on orientation binning of the local edges and the corresponding gradient magnitude.

That HOG feature extraction also has a purpose in biomedical research is shown by Acharya et al. [31] who propose an automated screening tool for dry and wet age-related macular degeneration (ARMD). Dry ARMD is the formation of small pale yellowish deposits under the retina, causing atrophy at the macula. Wet ARMD is caused due to the abnormal growth of blood vessels under the retina, which can lead to scarring at the macula or atrophic changes causing severe visual impairment. ARMD can be fully cured, but early detection can reduce the visual loss. Manual diagnosis is difficult and subjective. They show that HOG feature extraction paired with a particle swarm feature selection method, a technique derived from natural computing, achieves an accuracy of 85.12% for categorising images of dry and wet ARMD using a SVM with radial basis function kernel.

(2) Classification of zebrafish

A study by Jeanray et al. [32] focused on classification of phenotype (defects) of zebrafish embryos. A dataset containing images of embryos with their phenotype labelled by three biologists was first classified into 3 classes: "Dead", "Chorion" and "Other". The "Other" class was then reclassified into the various other phenotypes. Classification and feature extraction was done using dense random subwindows extraction, their description by raw pixel values and, finally the use of ensembles of extremely randomised trees. The ensembles were

either directly used for classification or as a feature descriptor and then classified using a SVM. Classification of “Dead” and “Chorion” was quite accurate with an error rate of respectively 1.12% and 2.6%. The two choice decision models for other phenotypes had varying results where some had an accuracy of 85-88% but others such as “Hemostasis” had much lower agreement rates (51.31%).

Alshut et al. [33] propose a method to automatically distinguish between “normal” and “coagulated” zebrafish embryos. As zebrafish are an important and commonly used model for toxicity studies this tool reduces the need for manual classification of embryos. Their system can distinguish between many orientations of the embryos. This causes a high level of variety in the images, hence a very specific feature extraction is required. They used a combination of 7 feature engineering methods. Of which the first two were found the most powerful: HOG, and mean intensity value within the chorion centre. Using a Bayes classifier with these top two features presented an error of 3.6%.

A deep learning-based classification of deformed zebrafish phenotypes proposed by Ishaq et al. [34] shows the potential of deep learning for accurate high-throughput classification. Using image data of 3dpf larvae in 96-well plates with about 5 larvae per well, of which 79 were intoxicated larvae and 60 untreated larvae. The data was augmented through flipping and rotation into 8 times as much data. Using 5 fold cross validation with 2000 iterations on the AlexNet deep convolutional network they achieved an accuracy of 92.8% for a two class problem classification into treated or untreated.

2.8.1 Relation to our research

The above sections present various techniques and tools used for classification and feature selection. In our research we will, however, limit the use to standard methods. SVMs have proved their potential in a previous bachelor thesis by van Heijnen and Neerbos [35]. ANNs have similar properties to SVMs but may require more training data and will therefore also be investigated. In the next chapter the specific implementation will be discussed.

Chapter 3

Implementation

In this chapter the setup of the experiments will be discussed. The data acquisition will be explained and we will elaborate on the processing and use hereof. Moreover a description of the final workflow (Figure 3.1) will be presented. Here we discuss our implementation of the before mentioned methods. In Chapter 4 we will specify and discuss the results.

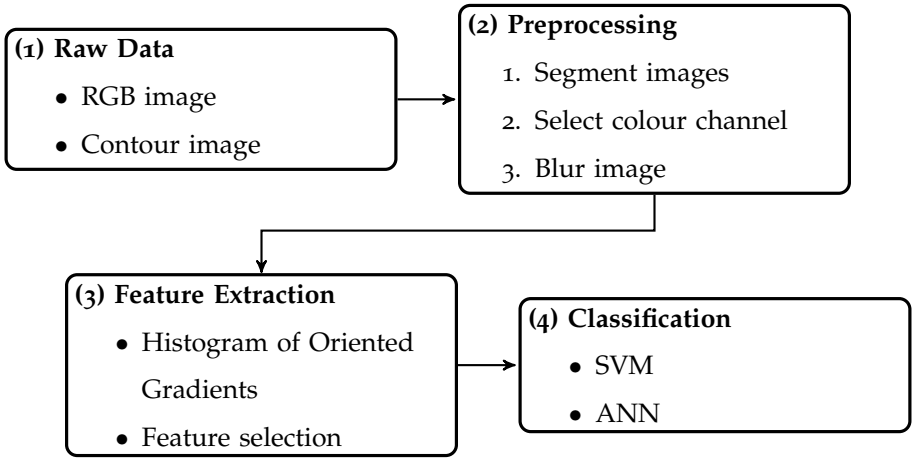


Figure 3.1: Overview of the workflow as implemented in this research. The number in the top left corner of each box identifies the corresponding section in this chapter.

3.1 Image data

3.1.1 Animals

Wild type zebrafish (*Danio rerio*) larvae of 3, 4 and 5 dpf were used for imaging. Zebrafish were grown for 3, 4 or 5 days and fixed in GA 4% fixative, then imaged according to the VAST bioimager based multi-modal high-throughput axial-view imaging (MM-HTAI) architecture proposed by Guo et al. [10] which will be briefly summarised below. Although the age of the zebrafish larvae is known there will be some variances in their

actual phenotype due to environmental factors. There is a chance some specific samples can be outliers, as is also shown by the volume and surface area distribution in Figure 1.1.

3.1.2 Data acquisition

In order to study the phenotype of an organism, that is the shape and morphology caused by the genetics, environment and possible diseases, imaging is key.

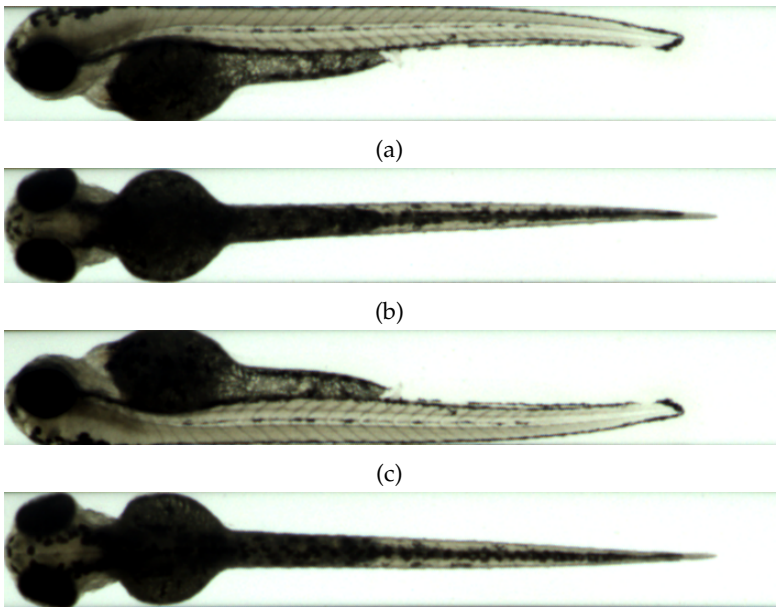
Imaging was performed using an adaptation to the VAST bioimaging system [5], the MM-HTAI architecture. This system loads zebrafish larvae from a reservoir into a capillary tube with a refraction index equal to water. The capillary tube can be rotated by two stepper-motors allowing imaging from different angles. The specimen is detected by a first camera (Allied Vision Systems, Pro Silica GE 1050) which allows for accurate positioning of the specimen in the field of view of the microscope. The microscope can then capture high resolution images of (a part of) the whole specimen. In the MM-HTAI method, the original VAST system is adapted to also use the positioning camera for imaging. This has a wider field of view and can therefore contain the whole specimen but are of lower resolution due to its lens characteristics. [10] This allows for fast, high-throughput imaging.

In our research the low resolution images were used. Each dataset consists of 84 bright field images of different equally distributed views per specimen along with their contour image (Figure 3.3a). Both image sets are in stored in `.tif` format and have a resolution of 1024 by 250 per image. In total 60 samples were available. 12 of 3dpf, 24 of 4dpf and 24 of 5dpf.

3.2 Preprocessing

From the preliminary results of a research project by W. Verhoef which used the same dataset a preprocessing script was acquired through internal communication. This script reads in the raw image data and performs various preprocessing steps on it (Figure 3.1.1). In the original preprocessing only the fish head was used. Here we use the whole fish image for classification. Using the complete body of the larvae results in an image of the sample containing much more information on the phenotype. Of the 84 images, four were selected for further processing: the dorsal, ventral and both lateral views (Figure 3.2). The Python library OpenCV2 is used for loading in the images and their masks. The selection is done by finding the contour image with the largest surface area. This is considered the most upright image and hence the left lateral view with image number i. Since the views are distributed equally, the other three are selected by taking the $i + 21 mod(84)^{\text{th}}$ images (Figure 3.3a and 3.3b).

The images are cropped and flipped if necessary. If the head is oriented to the right, the images are flipped horizontally. The necessity for flipping is determined as follows. From the image mask the moment is calculated from which the centroid point is extracted. From the mask a contour is computed, this is a curve joining all



(a) (b) (c) (d)

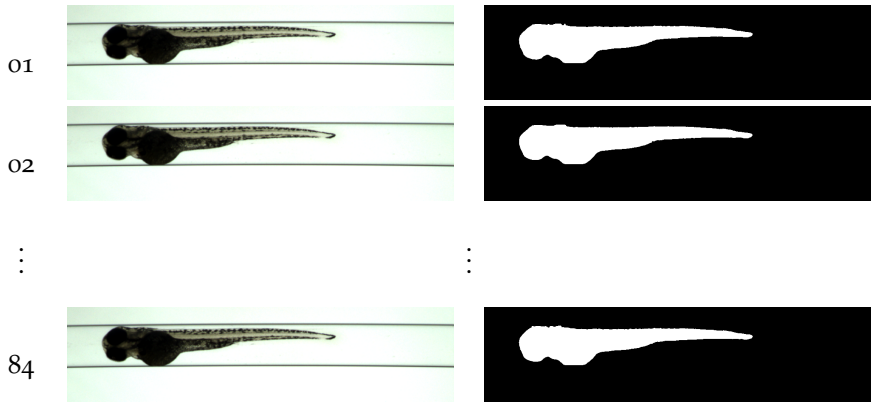
Figure 3.2: Example of four views automatically selected from series of 84 angles of zebrafish embryo of 3 dpf. Images acquired using VAST based MM-HTAI system. [10] (a), (b), (c), (d) respectively correspond to left lateral, ventral, right lateral, dorsal views. Images are cropped with equal size for specimens of all age groups. Size manually set to get a well fitting result for fish of all lengths in general.

the continuous points along a boundary having the same colour or intensity according to the algorithm by Suzuki and Abe [36]. A straight bounding rectangle is drawn along the contour of which the centre point is determined. If the centroid point of the moment is larger than the centre point of the bounding rectangle (i.e. the centroid point is oriented more to the left), the head is considered to be oriented to the right and thus the images are flipped horizontally.

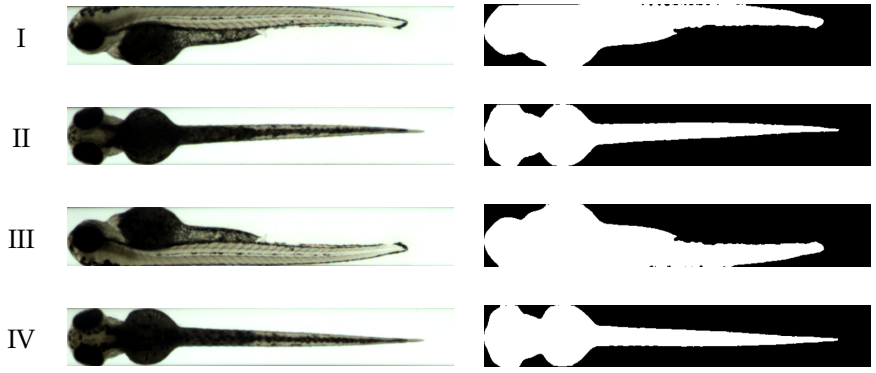
Next, the images are cropped to only contain the actual fish. A cropping size of 704 by 104 pixels is chosen. This size fully contains each fish without having any redundant information in the image, such as the edge of the capillary tube, which could possibly lead to a classification bias. Images are then slightly vertically scaled to 112 pixels in order to fit the block size for the HOG feature extraction. Figure 3.3 shows an overview of the processing pipeline of the data.

3.3 Feature extraction

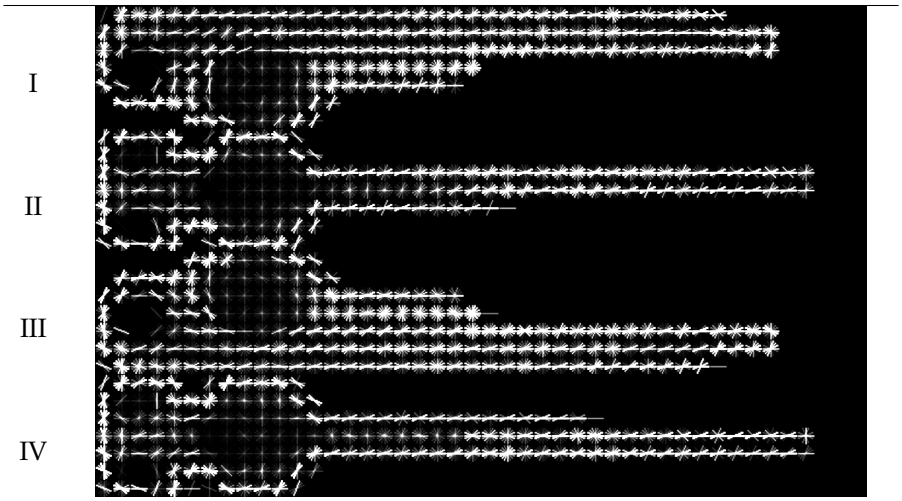
A good feature extraction is critical for an accurate representation of the image contents, in this case the age defining phenotype of the zebrafish larvae. Histogram of Oriented Gradients (HOG) as described in Chapter 2 is a well known feature extraction method and its suitability is investigated in this research. HOG is an image feature extraction method well suited for object detection. Its functioning is determined by various factors such as image processing, cell size, block size for normalisation and normalisation schemes. Dalal and Triggs [22] report a HOG extraction method well suited for human detection which is used as a baseline for our work.



(a) Raw data



(b) Processed data



(c) HOG features

Figure 3.3: Overview of image processing pipeline. (a) Raw data is acquired from MM-HTAI imaging [10] and a set of 84 images is stored per specimen. The provided dataset consists of a bright field image and a black and white cropping mask. (b) Four out of 84 images are selected based on orientation of sample: the left lateral, ventral, right lateral and dorsal views (Figure 3.2). (c) HOG feature extraction is performed after multiplying each view with its corresponding cropping mask.

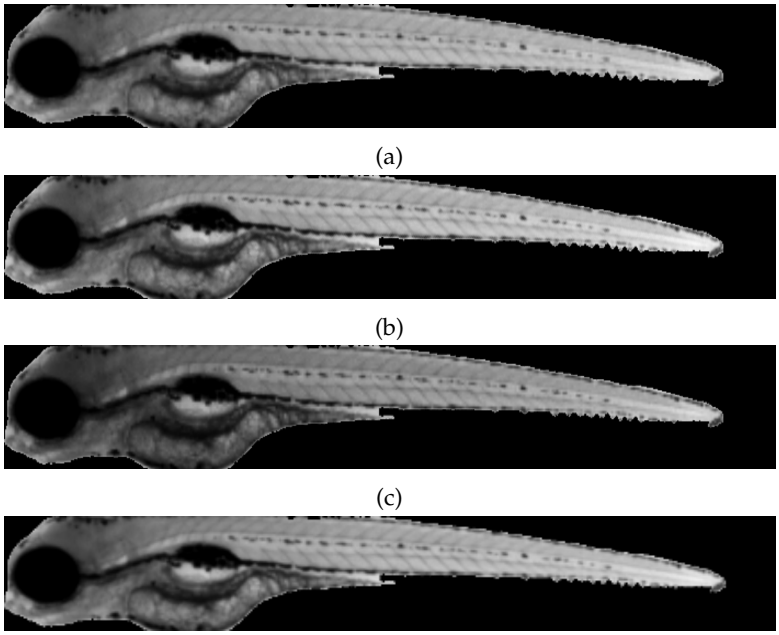
3.3.1 Colour conversion

A gradient can be computed over only one channel. This means that in order to calculate the HOG the three colour channels (RGB) will have to either be (1) used by themselves or (2) converted to grayscale.

(1) All colour channels are selected individually for HOG feature extraction by limiting to the specific array element from the segmented image.

(2) The segmented image is converted to grayscale using the OpenCV2 method `cvtColor()` with colour space conversion code set to `COLOR_BGR2GRAY`. This converts the colour image to grayscale using the following weighting scheme: $Y = 0.299 * R + 0.587 * G + 0.114 * B$ where Y is the output pixel intensity.

The separate colour channels have different intensities as can be seen in Figure 3.4.



(a)

(b)

(c)

(d)

Figure 3.4: Colour channel intensities of the same sample (left lateral view). (a), (b), (c), (d), respectively correspond to red, green, blue, grayscale intensity. As a original colour image such as in Figure 3.2 is mostly green, the green channel shows the most contrast.

3.3.2 Image blurring

As mentioned in the section 2.5.1, in order to suppress local gradients a smoothing may result in a better descriptor. Dalal and Triggs [22] report the effect of applying a Gaussian smoothing has merely a negative effect on human detection. Various smoothing methods were tested: (1) mean blur (averaging), (2) Gaussian blur, (3) bilateral filtering. The Python library OpenCV2 contains methods corresponding to these smoothing filters.

(1) A mean blur is implemented using the OpenCV2 method `blur()` with various kernel sizes, and default anchor point and pixel extrapolation method (which corresponds to a reflection at the border without

reflecting the pixel itself).

(2) Gaussian blurring is implemented using the OpenCV2 method `GaussianBlur()` with various kernel sizes and values for σ (automatically computed from kernel size) where $\sigma_x = \sigma_y$, and the default pixel extrapolation method.

(3) Bilateral filtering is implemented using the OpenCV2 method `bilateralFilter()` with various pixel diameters and values for σ_{colour} and σ_{space}, and the default pixel extrapolation method.

After applying a mean blur on Figure 3.4b the effect is as in Figure 3.5.

Figure 3.5: Mean blur *kernel = 7* applied to segmented green colour channel, result is a much more vague image.

3.3.3 Histogram of Oriented Gradients

The Python library OpenCV2 contains a method for HOG feature extraction with options for setting parameters. The `HOGDescriptor` class is initialised as follows. Window size is set to equal the dimensions of the input image (704 by 112 pixels), no gamma correction is applied, no Gaussian smoothing is applied here, the *L2-Hys* scheme is used for normalisation with a threshold of 0.2, various values are set for block size and block stride for the normalisation, number of bins per cell, and cell size. Other parameters were kept at their default setting. The `compute()` method returns the HOG feature vector for the segmented image. Per sample the feature vector of every view is flattened using the Python library NumPy to produce one vector per sample.

3.3.4 Feature selection

Dimensionality reduction is applied to reduce the feature space and only train on features with high variability. Depending on the cell size and number of bins the amount of features is quite high which can result in overfitting and long training times. Univariate feature selection is applied using the method `SelectPercentile()` from the Python library Scikit learn [37]. This allows for selecting the top x percent of features with a high variance according to a scoring function. Three scoring functions are compared: a Chi-squared test, an ANOVA F-value computation, and estimation of mutual information. Parameter settings, if any, are kept at their default value(s). Various values for x were tested, but for other experiments the top 20% is used along with an ANOVA test for variance.

3.4 Classification

The dataset of feature vectors is used for training a classifier using cross-validation. Five-fold cross-validation is used for training and testing as this delivers a quick results and causes a representative distribution of all classes over the splits as explained in section 2.7.1. For this the Scikit learn [37] method `StratifiedKFold()` is used which shuffles the data causing a random division of class types for training and testing across each fold. N-fold cross-validation using the leave-one-out method is used for individual detection of samples which are wrongly classified. For this the equally named method from Scikit learn is used.

The data is classified using (1) a support vector machine (SVM) and (2) an artificial neural network (ANN). As mentioned earlier, SVM showed its potential for classification of (bio-)images represented by HOG. Moreover, because of the small size of our dataset (60 samples with 3 classes) using a SVM is a good choice. Since there are similarities between SVMs and ANNs in their classification, a comparison is made between these two classification algorithms.

(1) A SVM is implemented using the Scikit learn method `SVC()`, which is the support vector classifier method of that library. Various kernel settings are possible but a linear kernel is used in all experiments. All other prameters are kept at default.

(2) An ANN is implemented using the Scikit learn method `MLPClassifier()`, the multi-layer perceptron method. Three hidden layers of 100 nodes per layer are used for experiments as this configuration showed good results in early experiments. The result of other layer/node combinations are compared. The 'lbfgs' solver is used, all other parameters are kept at their default setting.

3.4.1 Significance

The significance of the difference in classification was computed using a McNemar's test for variance. This statistical test gives a measure for the significance of a difference in results between two classifications on the same data. It is implemented using the equally named method from the Python library StatsModels. The correctly classified instances of the two classifiers are compared using a binomial distribution which delivers a *P-value* for significance. The H_0 of there being no difference in classification result is rejected if $p < 0.05$.

Chapter 4

Results

4.1 Performance study

In this section we will present details of the effects of various parameter settings on the classification performance. As the goal of this study is to create an optimal classification strategy, various parts in the image processing, feature extraction, and classification are optimised. An overview of these experiments is given below in order where they appear in the processing and classification pipeline, it should be noted that for every parameter in the performance study other settings were kept at standardised levels based on the final optimal settings and earlier observations. Those not given in Chapter 3 are given below. In the following paragraphs we often refer to the *performance* of the classification. In this thesis the F1-score as received through five-fold cross-validation is given as a performance measure. It should, however, be noted that run time of the preprocessing and classification is also of importance. This will be discussed briefly in section 4.1.5.

4.1.1 Colour conversions

As stated in Chapter 3 HOG calculates gradients from a single channel of the image. Therefore there are two methods for representation of the original image: using one colour channel, or a grayscale version of the image. We evaluated the effect of these image representations. Every individual colour channel of the RGB image and a grayscale pixel representation were compared. The resulting F1-score for classification with a SVM and ANN are given in Table 4.1. There is a slight performance increase by using just the green channel in comparison to the others in the F1-score of the ANN classifier.

Table 4.1: Overview of effects on classification performance by colour channel selection. Results given in F1-scores of classification with a SVM and an ANN. Grayscale is computed by a weighed average of all three colour channels. Image blurring with a bilateral filter $d = 5$, $\sigma_{colour} = 50$, $\sigma_{space} = 50$. HOG cell size = 16, block size = 16 and number of bins = 8.

	Red channel	**Green channel**	Blue channel	Grayscale
SVM	0.92	**0.92**	0.92	0.92
ANN	0.83	**0.87**	0.80	0.83

4.1.2 Filtering

The effects of applying a smoothing filter are noteworthy although inter-differences are subtle. The most straightforward mean-blur gives the best result. A Gaussian blur and bilateral filter with various settings were also compared, all on the green colour channel. A complete overview of the three blur types with their optimal setting is given in Table 4.2. Effect of parameter variances per smoothing filter are given in Table 4.3, 4.4, and 4.5. HOG feature extraction with a cell size = 16, block size = 16, block stride = 16, and number of bins = 8.

Table 4.2: Overview of effects on classification performance by image smoothing. Results given in F1-scores of classification with a SVM and an ANN.

	No blur	**Mean blur**	Gaussian blur	Bilateral filter
SVM	0.88	**0.93**	0.92	0.92
ANN	0.83	**0.88**	0.85	0.88

Table 4.3: MEAN BLUR. Overview of effects on classification performance by kernel size setting ($d_x = d_y$) using a mean blur. Results given in F1-scores of classification with a SVM and an ANN.

	3	5	**7**
SVM	0.92	0.92	**0.93**
ANN	0.85	0.85	**0.88**

Table 4.4: GAUSSIAN BLUR. Overview of effects on classification performance by kernel size setting ($d_x = d_y$) using a Gaussian blur. σ_{colour} and σ_{space} derived automatically from kernel size. Results given in F1-scores of classification with a SVM and an ANN.

	3	5	**7**
SVM	0.90	0.92	**0.92**
ANN	0.85	0.83	**0.85**

Table 4.5: BILATERAL FILTER. Overview of effects on classification performance by diameter size, σ_{space}, and σ_{colour} settings. Results given in F1-scores of classification with a SVM (top in cell) and an ANN (bottom in cell).

σ_{space}		**25**			50			75		
σ_{colour}		**25**	50	75	25	50	75	25	50	75
Diameter	3	0.90 0.83	0.90 0.83	0.90 0.85	0.90 0.85	0.90 0.83	0.90 0.87	0.90 0.85	0.90 0.85	0.90 0.87
	5	**0.92** **0.88**	0.92 0.85	0.92 0.85	0.92 0.87	0.92 0.87	0.92 0.83	0.92 0.87	0.92 0.87	0.92 0.83
	7	0.92 0.85	0.92 0.85	0.92 0.85	0.92 0.85	0.92 0.83	0.92 0.87	0.92 0.83	0.92 0.85	0.92 0.87

4.1.3 HOG block size and number of bins

The cropped and filtered input images have a size of 704 by 112 pixels. To reduce this size but still retain a relevant representation of the image a number of well fitting cell sizes were chosen based on this. A cell size of 16 pixels with a block size of 32 pixels appeared to give the best results. When using a mean blur, 8 bins seem to be optimal. Values for block size for normalisation and number of histogram bins were compared, all on the green colour channel. The results of this are given in Table 4.6.

Table 4.6: Overview of effects on classification performance of cell size, block size, and number of bins for HOG feature extraction. Results given in F1-scores of classification with a SVM (top in cell) and an ANN (bottom in cell). Image blurring with a mean blur *kernel size* = 7.

cell size, block size		8, 8	16, 16	8, 16	**16, 32**
number of bins	**8**	0.95 0.90	0.93 0.88	0.92 0.87	**0.97** **0.88**
	9	0.95 0.90	0.90 0.87	0.92 0.88	0.95 0.87

However, if the image is smoothed using a bilateral filter the results are somewhat different (Table 4.7). Although cell and block size have the same optimal values, having an histogram with 9 bins shows better performance.

Table 4.7: Overview of effects on classification performance of cell size, block size, and number of bins for HOG feature extraction. Results given in F1-scores of classification with a SVM (top in cell) and an ANN (bottom in cell). Image blurring with a bilateral filter $d = 5$, $\sigma_{colour} = 50$, $\sigma_{space} = 50$.

cell size, block size		8, 8	16, 16	8, 16	**16, 32**
number of bins	8	0.90 0.88	0.92 0.88	0.92 0.87	0.90 0.88
	9	0.92 0.88	0.92 0.88	0.92 0.87	**0.93** **0.88**

4.1.4 Dimensionality reduction

Using a HOG cell size of 16 pixels and 8 bins, the total amount of features per sample is 33024 which is quite a high amount. Dimensionality reduction through feature selection not only improves the speed of training and testing, but also improves the performance of the classification. Using an ANOVA F-value as univariate score for selecting the top-x percentage significantly improves classification performance (Figure 4.1). At $x = 20$ the F1-score is maximal for classification with a SVM as well as an ANN using green colour channel and smoothing using a mean blur with kernel size = 7. Other univariate scoring functions performed worse: Figure 4.1b and 4.1c.

4.1.5 Run time

Continuing on the previous section, it was found classification time is shorter when using a smaller feature space. Figure 4.2 illustrates this has a somewhat linear relation. Without feature selection the run time for five-fold cross-validation of the linear SVM was 0.84s. For the ANN this was 63s. After dimensionality reduction to 20% of the features, the cross-validation time for the linear SVM was 0.12s and 11s for the ANN.

The run time for preprocessing of the complete dataset (60 samples) consisting of finding the right views, orienting, cropping and scaling them, is about 80 seconds. A time performance increase could be achieved by parallelising this over all samples in the dataset. HOG feature extraction including blurring is relatively quick and takes only about 1.2 seconds for the complete dataset.

4.1.6 Classifiers

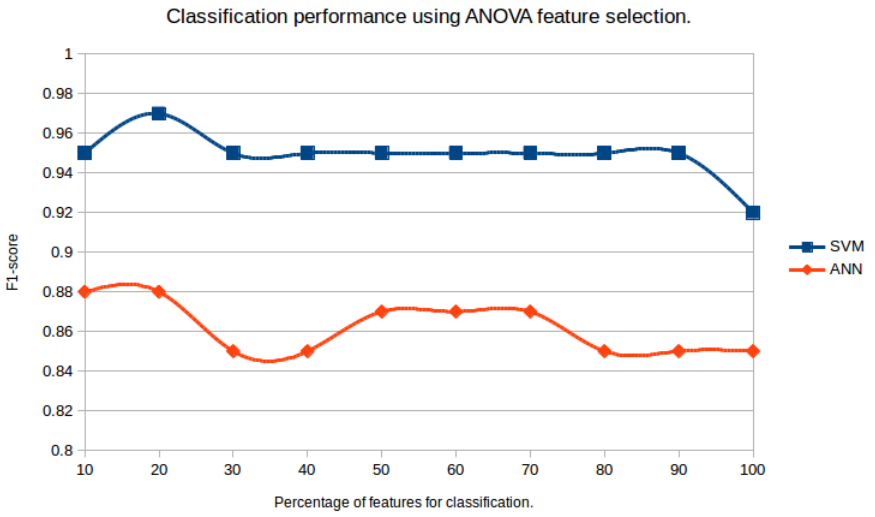
Classification using a SVM with linear kernel gives the best results. Several other kernels were also tested but had significantly lower performance (Table 4.8). The ANN results are comparable and statistically as significant. A McNemar's test gave a P-value of $p = 0.0625$ which is too high to reject H_0. Varying ANN node and layer combinations had only a slight effect on the classification performance (Figure 4.3).

Table 4.8: Overview of classification performance using different kernels in a SVM. Results given in F1-scores of classification.

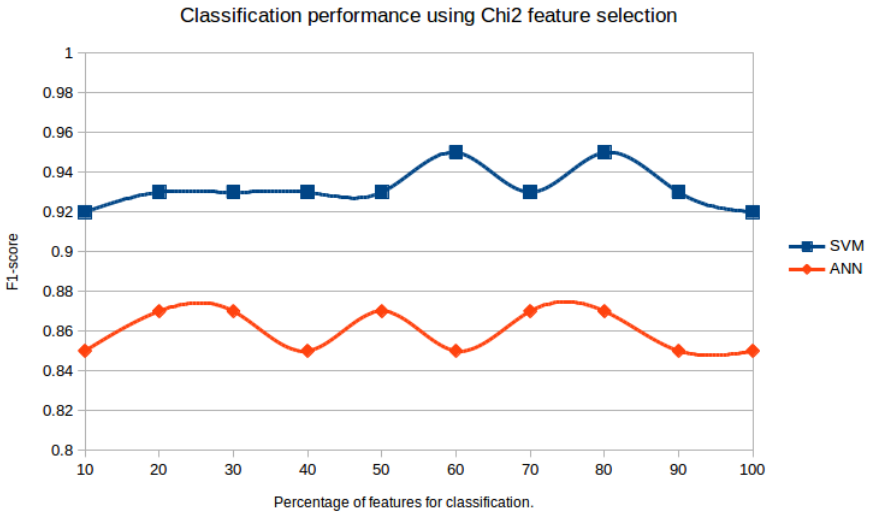
SVM-kernel	**Linear**	Polynomial	Radial basis function	Sigmoid
F1-score	**0.97**	0.60	0.65	0.66

4.1.7 N-fold cross-validation

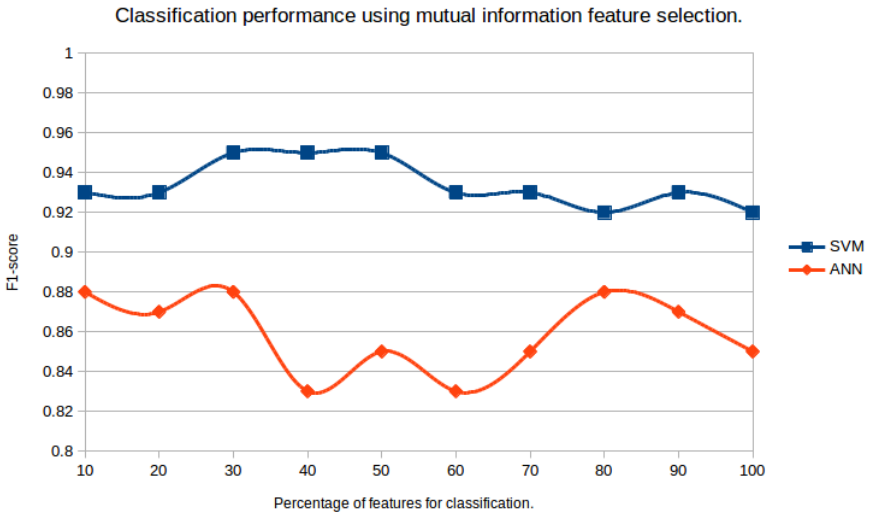
The above listed results are the product of classifier validation using 5-fold cross-validation. This is a faster way for validation than using n-fold cross-validation (or leave-one-out). Because the dataset in this project is relatively small (60 samples) n-fold cross-validation may be a more accurate measure for the quality of



(a)



(b)



(c)

Figure 4.1: Overview of classification performance using a SVM and an ANN and feature selection using different univariate scoring methods. (a) Univariate scoring based on ANOVA F-value, (b) scoring based on Chi^2 statistics, (c) scoring based on estimated mutual information for a discrete target variable.

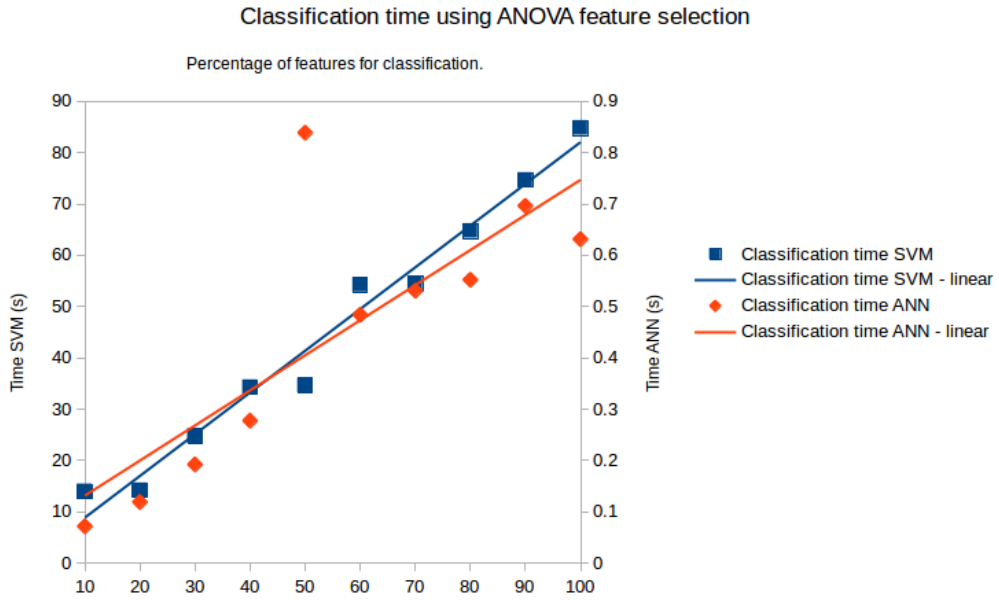


Figure 4.2: Overview of training and testing using 5-fold cross-validation time for classification with a SVM and an ANN. Feature selection using univariate scoring based on ANOVA F-value and different percentages of feature set used for classification.

the classification. Using n-fold cross-validation produces slightly different results, the performance of the classification goes down from 0.97 to 0.92 for the SVM and from 0.88 to 0.87 for the ANN.

4.2 Discussion

Overall, there are several notable findings in this work but they require some side notes. Here we will discuss the above listed results in the same order. Through using the green colour channel an increased performance is found. A possible explanation for this is the superiority in contrast. Figure 3.4b illustrates this, more subtle shapes are visible here than in those of the other colour channels which could possibly explain its superiority as a phenotype descriptor. The reason for this may be the configuration of the camera sensor. As the pixel matrix in a photo sensor has twice as many green pixels than red or blue pixels, there is an improved sensitivity in the green channel.

Our results show applying any type of blur before feature extraction gives a significant increase in classification performance compared to no blurring. Although Dalal and Triggs [22] report image smoothing before HOG feature extraction has a negative effect on human detection, our results are contradictory to this. Image smoothing reduces small local gradients but retains the general contrast structure in the image. Our higher resolution images of 1024 by 112 pixels may therefore benefit from smoothing due to noise reduction, where the images in [22] of 64 by 128 pixels would only suffer detail loss. This would also explain the improved performance through a cell size of 16 pixels. As this results in a similar cell to image size proportion as in [22], since $112/16 = 7$ and $64/8 = 8$.

Thirdly dimensionality reduction boosts classification performance quite significantly from 92% to 97% in

Nodes per layer	Layers						
	1 layer	2 layers	3 layers	4 layers	**5 layers**	6 layers	7 layers
10	85	87	85	85	87	78	80
20	88	85	82	88	**92**	85	87
30	85	85	85	83	87	77	88
40	88	83	87	85	83	85	87
50	88	85	83	88	87	83	88
60	88	87	85	88	83	83	80
70	88	88	87	87	78	88	87
80	88	87	87	87	82	88	87
90	87	87	87	85	85	85	83
100	88	83	88	85	85	82	87
110	88	85	82	87	83	83	87
120	88	85	87	87	82	85	87
130	87	88	83	87	82	82	83
140	88	87	82	83	83	90	82
150	88	87	87	85	85	83	87
160	88	87	83	82	85	83	87
170	88	87	85	85	87	87	85
180	88	87	85	87	87	82	80
190	88	85	83	85	82	87	87
200	88	87	85	85	87	87	85

Legend	
High	
Low	

Figure 4.3: Heatmap corresponding to the percentage accuracy of classification using an ANN with different node and (hidden) layer combinations.

SVM classification (Figure 4.1a). Moreover feature selection also increases cross-validation speed from about 63 to 11 seconds in ANN classification (Figure 4.2).

The classifiers used to produce these classification results perform quite well. Variations such as a different kernel in SVM, or different hidden layer and node combinations in ANN have a negative or only a slightly positive performance impact. It is possible there is a moderate bias to these results, as all other settings in preprocessing and feature extraction have been optimised for these classifiers.

4.2.1 Incorrect classified samples

It is possible to individually identify the incorrectly classified samples. These appear to commonly be the same samples in all experiments done. When these are looked up in the volume and surface area probability distributions by Guo [10], they demonstrate to lie in the 2σ-tail in those distributions (Figure 1.1). This shows that these specific samples probably are slightly outside of their labelled age category. From this we can conclude HOG really describes the maturation phenotype as it is correlated in some sense with how volume and surface area represent age.

4.2.2 Noise

A small side experiment was done to probe the effect of using a larger dataset for classification. Although no new data was available, a dataset can be enlarged by duplicating data and adding noise. When Gaussian noise

was added to copies of the original data and the combined data was used for classification it resulted in a F1-score of 1.0 for both the SVM and the ANN. This reliability of this outcome is highly subject to discussion, though. As all images underwent a mean blur, this may completely remove the effect of the added noise thereby essentially causing the cross-validation to use nearly identical data in the test and training sets.

Chapter 5

Conclusions

In this thesis we have investigated the potential of using histogram of oriented gradients (HOG) as a feature descriptor for the phenotype of zebrafish larvae defining for age. VAST imaging delivers a revolution sequence of 84 images per zebrafish larva. Applying HOG feature extraction on the dorsal, ventral, and both lateral views after fine-tuned preprocessing demonstrates to be an accurate measure for age prediction using a linear support vector machine (SVM) or artificial neural network (ANN) classifier.

We can conclude that HOG features are a good indicator for maturation in zebrafish. However, it is difficult to make a quantitative comparison to using volume and surface area as there is no classification done on these features in [10]. Commonly incorrect classified samples are also outliers in the surface area and volume probability distributions by Guo [10]. An important condition for the quality of HOG as a feature descriptor is to what extent it can represent the age of the larvae. As cross-validation shows a 97% accuracy we can deduce this condition has been met.

The preprocessing of the zebrafish image data has been optimised to increase the classification performance. Choosing a precise cropping size, using the right colour channel, and applying an image blur are essential factors in the image preprocessing. We show that after segmentation, using the green colour channel for feature extraction has some impact on the classification performance. Applying a mean blur with a kernel size of seven pixels gives a 0.05 increase in F1-score. Additionally the settings for the HOG feature extraction have been optimised.

A SVM with linear kernel is better fit for classification than an ANN. Although the higher performance is statistically not significant, five-fold cross-validation of the SVM is about 100 times faster than of the ANN. The combined classification accuracy and training times make the SVM a superior choice over the ANN. It is possible though that more training data would result in a better classification through using an ANN. In general about 20 samples per class (as is about the case of the dataset used in this study) is plenty for a SVM but relatively small for an ANN. Future research could investigate the effect of a using larger training dataset. It should be noted that although the ANN performs slower in the cross-validation the speed would not be an issue. If integrated in the final annotation pipeline it would have a shorter run time than other processes such

as the 3D reconstruction. Besides this it would be possible to speed up by parallelising parts in the processing pipeline including the preprocessing of the four views.

In summary we can address our problem statement as follows. Using 5-fold cross-validation we show that a linear SVM can make predictions for the age of zebrafish larvae with a F1-score of 0.97. The trained estimator can be implemented in the annotation pipeline to label zebrafish larvae of 3, 4 or 5dpf with their age which will enable high throughput scanning of larvae of various ages combined.

5.1 Future work

In its current form the classification performs well and achieves a maximal F1-score of 0.97. However, as mentioned before, the dataset used for training is quite small. More data is required in order to get a more generalised model. This should include zebrafish larvae from various batches to represent variation. This would probably also allow for better training of the ANN which requires larger amounts of training data in general.

Another use of a larger dataset would be to investigate deep learning. As deep learning composes its own feature extraction method through training of convolutional layers. Although this requires much training data, a deep learning approach would not be limited to the features presented. It would be interesting to study how a deep learning method would differ from the more traditional data science approach presented in this thesis.

For an increased classification performance, a combination between the HOG features and the descriptors volume and surface area of the reconstructed 3D model might provide an even more solid representation of age. Using these three descriptors as a model for age may be more reliable than only using HOG features.

In this study there was a limited evaluation of techniques. There was a focus on two classifiers and only three types of dimensionality reduction were compared. As there exist many other algorithms for classification, others could be investigated. Besides this, other feature extractions and different dimensionality reductions could be experimented with.

Additionally classification of larvae with a mutant or defect phenotype through the trained classifier could show an interesting outcome. This method would allow for studying how a disease or mutation effects the maturation of a larva. A high throughput screen of diseased or intoxicated zebrafish may provide new insights in how the pathogen or chemical influences growth by comparing the predicted age to the actual maturity. A regression approach instead of classification might be more insightful in this case as ageing is a constant factor and not just defined in days. A regression approach may therefore provide a more detailed overview of the age as deduced through HOG feature extraction.

Acknowledgements

Wilco Verhoef and Yuanhao Guo provided an essential basis for the research done in this thesis through their previous work. Many thanks go to Fons Verbeek for supervising this project and providing helpful advice at many points along with Kristian Rietveld for support with cluster computing and installing Python libraries on the LLSC.

Bibliography

[1] D. R. Love, F. B. Pichler, A. Dodd, B. R. Copp, and D. R. Greenwood, "Technology for high-throughput screens: the present and future using zebrafish," *Current Opinion in Biotechnology*, vol. 15, pp. 564–571, 12 2004.

[2] H. Teraoka, W. Dong, and T. Hiraga, "Zebrafish as a novel experimental model for developmental toxicology," *Congenital Anomalies*, vol. 43, pp. 123–132, 6 2003.

[3] K. Dooley and L. I. Zon, "Zebrafish: a model system for the study of human disease," *Current Opinion in Genetics & Development*, vol. 10, no. 3, pp. 252–256, 2000.

[4] C. A. Lessman, "The developing zebrafish (Danio rerio): A vertebrate model for high-throughput screening of chemical libraries," *Birth Defects Research Part C: Embryo Today: Reviews*, vol. 93, pp. 268–280, 9 2011.

[5] C. Pardo-Martin, T.-Y. Chang, B. K. Koo, C. L. Gilleland, S. C. Wasserman, and M. F. Yanik, "High-throughput in vivo vertebrate screening," *Nature Methods*, vol. 7, p. 634, 7 2010.

[6] T.-Y. Chang, C. Pardo-Martin, A. Allalou, C. Wahlby, and M. F. Yanik, "Fully automated cellular-resolution vertebrate screening platform with parallel animal processing," *Lab on a Chip*, vol. 12, no. 4, pp. 711–716, 2012.

[7] Y. Guo, Y. Zhang, and F. J. Verbeek, "A Two-Phase 3-D Reconstruction Approach for Light Microscopy Axial-View Imaging," *IEEE Journal on Selected Topics in Signal Processing*, vol. 11, pp. 1034–1046, 10 2017.

[8] W. Denk and H. Horstmann, "Serial Block-Face Scanning Electron Microscopy to Reconstruct Three-Dimensional Tissue Nanostructure," *PLoS Biology*, vol. 2, p. e329, 10 2004.

[9] A. W. Fitzgibbon, G. Cross, and A. Zisserman, "Automatic 3D Model Construction for Turn-Table Sequences," pp. 155–170, Springer, Berlin, Heidelberg, 1998.

[10] Y. Guo, *Shape Analysis for Phenotype Characterisation from High-Throughput Imaging*. PhD thesis, LIACS - Universiteit Leiden, 2017.

[11] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, pp. 210–229, 7 1959.

[12] R. K. McConnell, "Method of and apparatus for pattern recognition," July 22, 1982. U.S. Patent 4 567 610 A.

[13] G. J. Lieschke and P. D. Currie, "Animal models of human disease: zebrafish swim into view," *Nature Reviews Genetics*, vol. 8, pp. 353–367, 5 2007.

[14] C. B. Kimmel, W. W. Ballard, S. R. Kimmel, B. Ullmann, and T. F. Schilling, "Stages of Embryonic Development of the Zebrafish," *DEVELOPMENTAL DYNAMICS*, vol. 203, pp. 253–310, 1995.

[15] C. Singleman and N. G. Holtzman, "Growth and maturation in the zebrafish, Danio rerio: a staging tool for teaching and research.," *Zebrafish*, vol. 11, pp. 396–406, 8 2014.

[16] D. M. Parichy, M. R. Elizondo, M. G. Mills, T. N. Gordon, and R. E. Engeszer, "Normal table of postembryonic zebrafish development: staging by externally visible anatomy of the living fish.," *Developmental dynamics : an official publication of the American Association of Anatomists*, vol. 238, pp. 2975–3015, 12 2009.

[17] MyScope, "Confocal Microscope Training module," *Australian Microscopy & Microanalysis Research Facility*.

[18] G. van Rossum, "Python Programming Language," *USENIX Annual Technical Conference*, 2007.

[19] R. S. Choras, "Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems," *INTERNATIONAL JOURNAL OF BIOLOGY AND BIOMEDICAL ENGINEERING*, 2007.

[20] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, pp. 1141–1151, 10 2002.

[21] F. Banterle, M. Corsini, P. Cignoni, and R. Scopigno, "A Low-Memory, Straightforward and Fast Bilateral Filter Through Subsampling in Spatial Domain," *Computer Graphics Forum*, vol. 31, pp. 19–32, 2 2012.

[22] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, 2005.

[23] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding.," *Science (New York, N.Y.)*, vol. 290, pp. 2323–6, 12 2000.

[24] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction.," *Science (New York, N.Y.)*, vol. 290, pp. 2319–23, 12 2000.

[25] I. H. I. H. Witten, E. Frank, M. A. M. A. Hall, and C. J. Pal, *Data mining : practical machine learning tools and techniques*.

[26] N. M. Nasrabadi, "Pattern Recognition and Machine Learning," *Journal of Electronic Imaging*, vol. 16, p. 049901, 1 2007.

[27] F. Rosenblatt, "The Perceptron, a perceiving and recognizing automaton," Tech. Rep. 85-460-1, Cornell Aeronautical Laboratory, 1957.

[28] R. Kohavi and R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," pp. 1137–1143, 1995.

[29] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," pp. 345–359, Springer, Berlin, Heidelberg, 2005.

[30] M. Dahmane and J. Meunier, "Emotion recognition using dynamic grid-based HoG features," in *Face and Gesture 2011*, pp. 884–888, IEEE, 3 2011.

[31] U. R. Acharya, Y. Hagiwara, J. E. Koh, J. H. Tan, S. V. Bhandary, A. K. Rao, and U. Raghavendra, "Automated screening tool for dry and wet age-related macular degeneration (ARMD) using pyramid of histogram of oriented gradients (PHOG) and nonlinear features," *Journal of Computational Science*, vol. 20, pp. 41–51, 5 2017.

[32] N. Jeanray, R. Marée, B. Pruvot, O. Stern, P. Geurts, L. Wehenkel, and M. Muller, "Phenotype classification of zebrafish embryos by supervised learning," *PLoS ONE*, vol. 10, p. e0116989, 1 2015.

[33] R. Alshut, J. Legradi, L. Yang, U. Strähle, R. Mikut, and M. Reischl, "Robust Identification of Coagulated Zebrafish Eggs using Image Processing and Classification Techniques," *Workshop Computational Intelligence*, vol. 19, p. 921, 2009.

[34] O. Ishaq, S. K. Sadanandan, and C. Wählby, "Deep Fish," *SLAS DISCOVERY: Advancing Life Sciences R&D*, vol. 22, pp. 102–107, 1 2017.

[35] C. Heijnen and K. van Neerbos, "Organizing and classifying historical butterfly collections from Indonesia," Bachelor Thesis, LIACS - Universiteit Leiden, 2017.

[36] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 32–46, 4 1985.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.