

Fish Component Recognition

Yan Li



Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2012

Abstract

This project aims firstly to segment fish components from fish images captured by underwater camera in Taiwan. A tail segmentation algorithm was developed which used curvatures of boundary points to find a set of candidate positions of tail joint. The algorithm firstly located one tail tip and iteratively searched through the candidate locations to find the other tail tip and then the joints between tail and fish body. A backup tail segmentation was also implemented which tracked the sudden changes in the row length of each column and took the second sudden change as the tail segmentation column.

For each segmented tail, it was rescaled, rotated and transferred into the common coordinate system. Principal components analysis was used to reduce the dimensionality from 900 to 10. These 10 principal component features along with other 9 image moment features were combined as the feature set. In order to improve the classification accuracy and use less features, sequential forward selection was implemented to select the best set of features. After all these procedures, 12 features were selected and used to represent tails. An overall tail classes classification accuracy of 73.7% was achieved by using a Naive Bayes classifier.

Acknowledgements

I would like to sincerely thank my supervisor Dr. Robert Fish for all his continuous helps, insightful suggestions and valuable feedbacks throughout the whole project. Also I would like to thank Dr. Xuan Huang and Dr. Bas Boom for their kind helps and supports. Finally and more importantly I would like to thank my family for their selfless love and constant support during all years of study.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Yan Li)

Table of Contents

1	Introduction	1
1.1	Motivation	1
1.2	System Overview	2
1.3	Research Objectives	3
1.4	Thesis Overview	4
2	Background	5
2.1	Literature Review	5
2.1.1	MSc Project of Rory McGrath[9]	5
2.1.2	R. Larsen et al[8]	6
2.1.3	C. Spampinato et al[16]	6
2.1.4	T. Cootes et al[4]	7
2.1.5	S. Sclaroff et al[15]	8
2.2	Data	8
2.3	Summary	11
3	Fish Component Segmentation	12
3.1	Component Segmentation	12
3.1.1	Fish Boundary Extraction	12
3.1.2	Boundary Tracking	13
3.1.3	Curvature Calculation	14
3.1.4	Tail Segmentation	17
3.2	Summary	22
4	Classification	23
4.1	Naive Bayes	23
4.2	Tail Classification	25
4.2.1	Tail Classes	25

4.2.2	Features	28
4.2.3	Feature Selection	33
4.3	Summary	42
5	Evaluation	43
5.1	Segmentation Performance	43
5.1.1	Performance Measure	44
5.1.2	Ground-truth Tails	45
5.1.3	Segmentation Results	46
5.1.4	Analysis of Tail Segmentation Performance	48
5.2	Classification Performance	50
5.2.1	Tail Classes Classification	50
5.2.2	Fish Species Classification	53
5.3	Future Work	55
5.4	Conclusion	56
	Bibliography	57

List of Figures

1.1	Fish4Knowledge Logo	2
2.1	Examples of fish images	8
2.2	Examples of binary images	9
2.3	Comparison of Ideal and Unhelpful Fish Images	9
3.1	Example of component segmentation	12
3.2	Boundary Tracking Algorithm	13
3.3	Raw Contour(Black) and Smoothed Contour(Red)	14
3.4	Raw Boundary VS Smoothed Boundary	14
3.5	A Sample Fish Image	15
3.6	Curvature Plot of a Fish Sample	15
3.7	Extreme Curvature Points of a Fish Sample	16
3.8	Extreme Curvature Points in the Fish Contour	17
3.9	Positive Extrema VS Negative Extrema	17
3.10	Fish Contours and Extreme Points for Different Samples	18
3.11	Joints Search Example	19
3.12	An Example of Backup Tail Segmentation	20
3.13	Example of Tail Segmentation	21
3.14	Example of Backup Tail Segmentation	21
4.1	Fish Species Classification	25
4.2	Component Type Classification	25
4.3	Examples of Fish Tail Type 1	26
4.4	Examples of Fish Tail Type 2	27
4.5	Examples of Fish Tail Type 3	27
4.6	The Common Coordinate System of Tail	28
4.7	Examples of Reshaped Tails	29

4.8	The Plot of Cumulated Variation of Eigenvectors	30
4.9	The Plot of Cumulated Variation of Top 80 Eigenvectors	31
4.10	Classification Performance	35
4.11	Histogram of Number of Selected Features (Training Sets)	35
4.12	Histogram of Number of Selected Features (Validation Sets)	36
4.13	Attribute and the Number of Appearance in the Feature Sets	37
5.1	Tail Representation Using 9 Points	45
5.2	Example of Ground-truth Tail Annotated from RGB Images	45
5.3	Example of Binary Ground-truth	46
5.4	Examples of Tail Segmentation	46
5.5	Percentage of Satisfaction vs Threshold	47
5.6	Percentage of Satisfaction vs Threshold	48
5.7	Outlier Fish Examples	49
5.8	Bad Fish Extraction Examples	50
5.9	Fish Instances in 2-D Feature Space	52
5.10	Ideal Type 1 Tails vs Bad Type 1 Tails	53

List of Tables

4.1	Percentage of Variation Using Different Number of Eigenvectors . . .	31
4.2	Features Used to Represent Tail	32
4.3	Attribute Index and the Number of Appearance in the Feature Sets . .	37
4.4	Attribute Index and the Number of Appearance in the Feature Sets . .	38
4.5	Features Selected by Sequential Forward Selection Algorithm	38
4.6	Table of Ranked Attributes by Information Gain	40
4.7	Classification Accuracy Using Top N Features	40
5.1	Percentage of Satisfaction (Binary Ground-truth)	47
5.2	Percentage of Satisfaction(RGB Ground-truth)	47
5.3	Data Sets for Tail Classes Classification	51
5.4	Confusion Matrix of Tail Classes Classification	51
5.5	Classification Accuracy for Each Tail Class	51
5.6	Confusion Matrix of Fish Species Classification	54
5.7	Classification Accuracy for Each Tail Class	54

Chapter 1

Introduction

1.1 Motivation

The study of marine ecosystems is becoming more and more important for understanding the underwater environments. Although human beings have been studying and benefiting from the rivers, lakes, oceans for thousands of years. Unlike other parts of the world, the underwater world is still more than a mystery to us. Certainly there are lots of stories about mysterious lives living somewhere under the water, For example the famous 'monster' Nessie in Loch Ness. Every now and then, the news of unknown species discovered in the water could be heard. Nobody knows how many species there are under the water, and how big the population is. Thus the study of marine ecosystem could help people in monitoring the underwater world, estimating the population and species, following their movements.

Studies nowadays are always based on videos and images captured by underwater cameras. However the amount of information is a very headache problem. The information is too much for human experts to process, people will easily get overwhelmed by viewing that amount of videos and images. And it is not a also not a sensible which wastes way too much time and resources. The development of computer programs that can reduce the workload of people become essential. It's better to have computers work on those tedious parts, such as segmenting fish components, tracking fish movement, and have human experts work on more high-level works.

This thesis develops the algorithms of segmenting fish components from detected fish images, investigates the possible methods of clustering fish components, such as the tail and develop classifiers that can discriminate different fish tail shapes.

1.2 System Overview

The Fish4Knowledge project aims to build a platform that could coordinate computer vision, semantic web, database management and query and workflow techniques, in order to facilitate non-programming scientists working on their studies. The project investigates methods for capture, storage, analysis and query of multiple video streams of undersea environment video, so that useful information could be efficient extracted and presented to the scientists. Below is the logo of the Fish4Knowledge project:

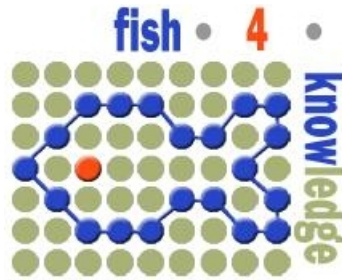


Figure 1.1: Fish4Knowledge Logo

At the moment, there are 10 embedded video cameras set undersea in Taiwan observing and recording the underwater environment. Video data captured by these cameras are used as source data in the project. However for each camera, 20 gigabytes video data are generated per hour and with 12+ hours of useful daylight 100 terabytes of data are generated each year. The amount of data is too massive to be processed efficiently. The project investigates methods for extracting useful information from videos, storing information in an efficient and meaningful way, so as to build a flexible architecture for data acquisition and analysis.

The project aims to research on the following objectives:

- Detecting targets in noisy environments.
- Characterising interactions between the targets.
- Recognising fish species by integrating multiple perspectively distorted views.
- Exploiting ontologies to interpret user queries.
- Exploiting ontologies to convert queries into workflow sequences.
- Storing and accessing massive amounts of data in a timely manner.
- Integration of the research in a publically usable web tool.
- Creation of a fish database suitable for behavioural and environmental studies.

- Training of staff in cross-disciplinary methods

1.3 Research Objectives

The project aims to explore and achieve three main objectives, each of which represents a different prospective of the problem. And step by step, they build on each other. Below are the objectives of this project:

- locating and extract fish component
- representing fish component efficiently
- classifying fish component using selected properties

Firstly, this project plans to segment fish components from extracted fish images. Fish is divided into four components, namely head, tail, upper body with dorsal fin and lower body with ventral fin. In this project, all research are concentrated on tail part.

Secondly, the project aims to find an efficient way to represent shapes of components other than using all pixels. Common features for each component are planned to be found, so that later a fish could be represented as a combination of different component features.

Lastly, two types of classification problems are to dealt with. The first problem is the fish component classes classification which classify fish component into different component classes. The second problem is to classify species of fish using component features.

1.4 Thesis Overview

The thesis is developed into five chapters, each of which focuses on one different aspect of the project. The thesis states the various algorithms and techniques used in research, evaluates the performance and provides possible future work.

Chapter 2 introduces the relevant researches and techniques that have been done previously and briefly discusses the limitations in their researches and their differences between this project. The data used in this project is also discussed and presented in this chapter.

Chapter 3 presents the image processing techniques and algorithms used in the fish component segmentation stage. The algorithms are mainly focused on tail segmentation.

Chapter 4 illustrates the features used to represent fish components and discusses the methods for feature selection. The classifier is also discussed in this chapter.

Chapter 5 focus on the evaluation all previous algorithms. It starts with evaluating the performance of component segmentation. Then the classification results of fish component are also examined. Future work and conclusion is also discussed in this chapter.

Chapter 2

Background

This chapter introduces the relevant researches and techniques that have been done previously. The limitations in their researches and differences between this project are also briefly discussed. Previous work that related to the Fish4Knowledge is firstly introduced. Then followed by other work about fish classification and component segmentation. Data used in this project is also discussed and presented in this chapter.

2.1 Literature Review

2.1.1 MSc Project of Rory McGrath[9]

In this MSc project research, numbers of properties were explored to represent fish images and feature selection procedures were applied in order to achieve a good classification results. The data used in this project was a set of fish tracking video, which consisted of the movements of 4 fish species. Other than the tracking video, the coordinates of detected fish and binary masks of detected fish were also provided. Ground truth information of fish species was manually labelled.

After adjusting all fish into the common orientation, different feature properties were generated. Colour features were extracted using histograms which represented the general colour information of fish. Texture features were co-occurrence matrices of each two different colour channels. Shape features were extracted using Fourier boundary descriptors and complex moments. Altogether, a total of 2242 features were extracted for each fish image.

A sequential forward feature selection method was then used to select a small set of features in order to achieve a better classification result and a robust fish representation.

K-Nearest Neighbour was used for both feature selection and fish species classification. At last an overall accuracy of 70.6% was achieved.

This research produced relatively good results, although this could highly relate to the difference in colour between different species and also there were only 4 species. In general the research explored a set of features that could be used to represent fish and algorithms.

2.1.2 R. Larsen et al[8]

The research in this project was conducted on the fish species classification. Both shape features and texture features were used for the classification, which were derived from an active appearance model[3]. Principle Component Analysis[1] was then applied to these two features sets which were then weighted by their variances. Predictions were made by comparing principal component scores with a pairwise Fisher discriminate analysis.

The data set used in this paper consisted of 20 cod, 58 haddock, and 30 whiting. Altogether there were only 108 fish. The landmarks used in active appearance model were manually annotated, which included fish eye, spine and general boundary. By training on these annotated fish images, active appearance model would be able to locate landmarks automatically for new fish. The overall classification accuracy achieved in this project was 76%.

One of the limitations of this project was that fish images were taken on dead fish by a standard colour CCD camera under a standardized white light illumination. The environment was fully under control, which could not be generalized to images captured by real-time marine monitors.

Another limitation was that the landmarks of fish in the training set were manually annotated. So the variations of the model was highly constrained by the training images, it was unable to generalize well to unknown fish. Besides, the data used in this project was quite small, so the result was not very reliable.

2.1.3 C. Spampinato et al[16]

In this paper research was carried on fish classification and fish movement tracking. For the classification task, both texture features and shape features were used. Texture features included statistical moments of the gray-level histogram, spatial Gabor filtering and properties of the co-occurrence matrix. Shape Features were extracted

by using the Curvature Scale Space transform and histogram of Fourier descriptor of boundaries. Moreover, An affine transformation was applied so that fish in 3D could be represented by multiple views. PCA was later used to reduce the dimensionality of feature sets.

The dataset used in this paper contained 320 images of 10 species. For each species, 14 fish images were acquired from underwater cameras in Taiwan and 18 images obtained by affine transformation[10]. Besides, another test set contained 100 images of 10 species were also used for evaluation. An overall classification accuracy of 92% was later achieved.

The research explored extracting both texture features and shape features to represent fish image, and achieve a very good classification result. Using affine images to describe arbitrary views in video sequence was brilliant idea. However, using more affine images than original images seems inappropriate. It is better to have more original images to capture variations in the data set instead of using too many affine images.

2.1.4 T. Cootes et al[4]

In this paper, the idea of the famous Active Shape Model was first introduced. The models started with labelling the training set images with a specified number of points. Then points of training samples were rescaled, rotated and transferred into a common coordinate system so that a point distribution model was created. A multivariate Gaussian distribution was used to model intensity changes along the marked points. Principal component analysis was then applied to the point distribution model in order to generate a more efficient representation.

For a new unlabelled image, the Active Shape Model iteratively searched through candidate locations by the tuning the parameters of principal components. Local searches were proposed by the Gaussian distribution. Different resolutions were also considered by smoothing and sub-sampling.

Several examples of Active Shape Model were also presented, which included bottle shape modelling, hand shape modelling and heart chamber modelling. The performances of modelling and automatic localising were very impressive. However, all images were in the similar conditions where model shapes and intensity differences between background and foreground were almost identical. In comparison, there are thousands of 2-D fish images captured while swimming, the variations of fish shapes are much difficult to model. Besides, different species usually have much distinct con-

tours.

2.1.5 S. Sclaroff et al[15]

In this paper, research in object deformable shape detection were carried out. The object segmentation was achieved by model-based region grouping. At first a template model was build which could be deformed into different shapes by tuning shape parameters. Then for a new colour image, the standard region-merging algorithm was applied which produced a raw segmentation result. An edge map was computed for the image which were later used constrains for region merging. Different region grouping hypotheses were tested by fitting the template model to the region. Iteratively various alignment and deformation parameters were tuned in order to match the region. At last a set of best region grouping hypotheses were used along with the raw segmentation result to generate the final segmentation result.

A set of segmentation examples were also presented in the paper, which all achieved favourable results. However the problem of this system is the computational complexity. The hypotheses testing takes too much computational effort in order to find the optimal region grouping hypothesis.

2.2 Data

The image data given for this project are based on the outcome of previous work on the Fish4Knowledge project. The image data are basically fish extracted from video sequences by [12] and manually labelled fish species. There are 6874 RGB fish images of 15 species in the data set. Below are some examples of RGB images:

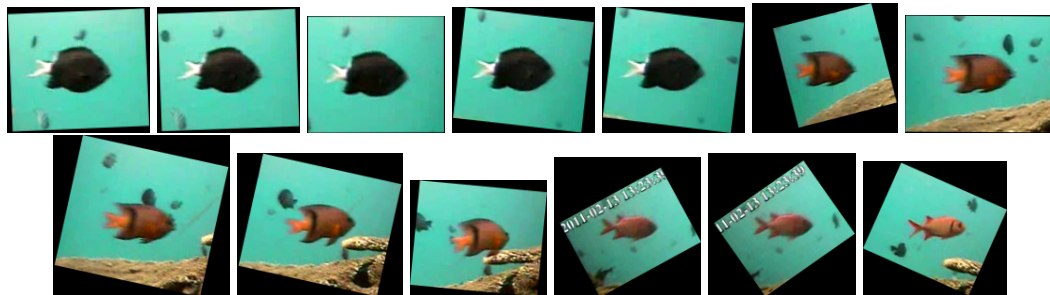


Figure 2.1: Examples of fish images

The fish images have been pre-processed so that the positions of fish are standardized. Fish are always at the centre of images and are placed horizontally left to right.

Other than the RGB images, the segmented binary fish images are also provided. And below are the corresponding binary fish images:

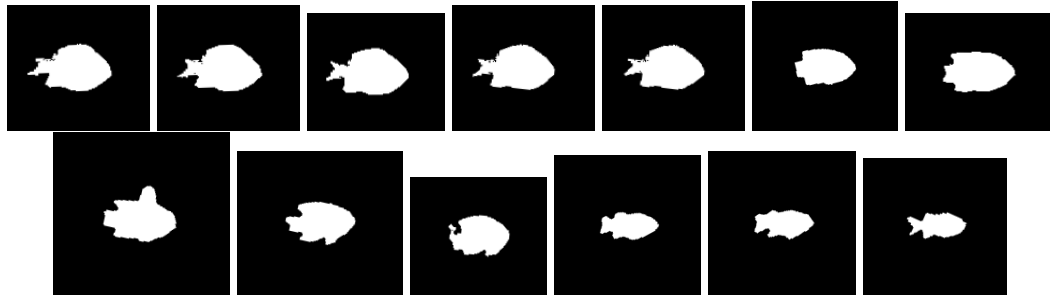


Figure 2.2: Examples of binary images

Because of so many uncertainties of these fish images, such as the variation of lighting condition, body movements while swimming, background (open sea or overlap with other fish), the binary images segmentation results are generally good but not perfect. As could be seen from above images, the fish bodies are dilated, sometimes contain parts of other fish and more importantly the details of fish tails are lost during the segmentation procedure.

Moreover, fish in this project are supposed to be 2-D images and are best obtained when fish are parallel to the camera. However, due to the abundance of fish images, this is not always the case. Below are the comparison of ideal fish images and the unhelpful fish images:

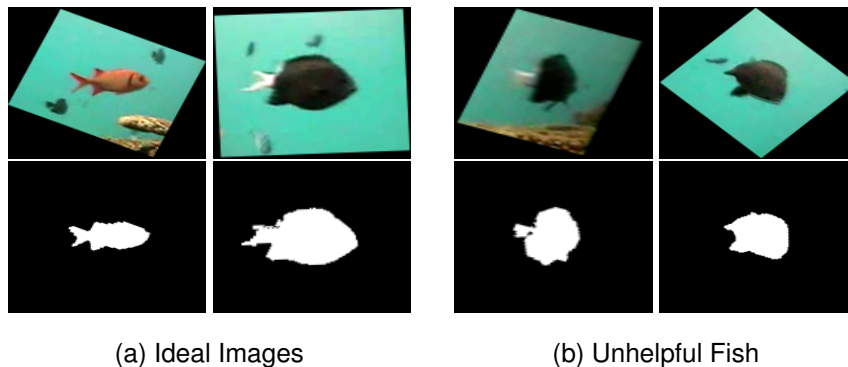


Figure 2.3: Comparison of Ideal and Unhelpful Fish Images

At the moment, a set of useful fish images are manually selected from the dataset. Ideally a machine learning classifier should be trained and applied to select images when tracking the movement of fish. All the following segmentation and classification procedures are processed on this selected dataset, which contains 4945 fish images of

15 species. Among all 15 species, one species takes about half of the whole population. The top several species almost dominate the whole populations, which shows the dataset is quite unbalanced. In order to achieve a reliable classification performance, additional attention is paid when selecting the training and test set.

2.3 Summary

In this chapter, previous researches and approaches about fish classification, deformable object segmentation and representation were discussed and summarized. Although there is no previous research done in the fish component segmentation so far, these previous researches have been very inspiring.

Data used in this project was also illustrated along with the limitations. The main characteristic of fish images in this project is the low resolution compared with other researches, which makes the project very special. Because small number changes in pixel values would not make much difference in high-resolution images, but that could affect a lot in low-resolution images.

Chapter 3

Fish Component Segmentation

This chapter deals with the image processing techniques and algorithms used in the fish component segmentation stage. The algorithms are mainly focused on tail segmentation.

3.1 Component Segmentation

As discussed before, ideally after extracting the body contour, the fish should be segmented into 4 parts, namely head, tail, upper body with dorsal fin and lower body with ventral fin. Below is an example of the ideal fish component segmentation.

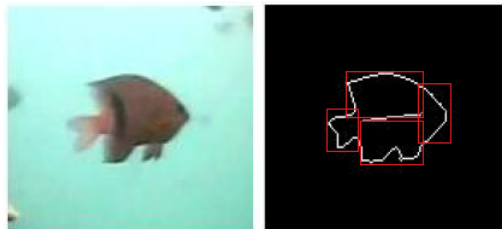


Figure 3.1: Example of component segmentation

3.1.1 Fish Boundary Extraction

The first step of component segmentation is the boundary extraction, which extracts and smooths the contour of the fish and make it ready to compute curvatures[11] along the fish contour.

3.1.2 Boundary Tracking

After extracting the raw boundary of fish, a boundary tracking algorithm is then applied in order to achieve a smoothed and clear boundary.

The main idea of the boundary tracking algorithm is quite simple and clear. The tracking algorithm firstly computes the middle column of the raw fish contour and then starts the tracking with the upper boundary point in that column. Below is an example to illustrate the boundary tracking.

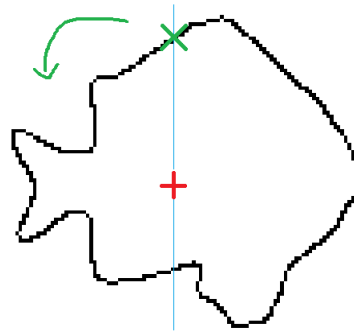


Figure 3.2: Boundary Tracking Algorithm

The red point in the image is the centroid of the raw boundary points, the blue line shows the column of the centroid and the green cross above the centroid is the starting point of the tracking algorithm. The algorithm continues tracking the boundary point anti-clockwise.

The next visited point is the nearest point within a specified gap size and the point between the gap is also filled. If there is no point within the gap size in the anti-clockwise direction, then it continues tracking from the start point in the clockwise direction. If there is no next visited point in both directions, then the algorithm stops the current curve tracking and starts another tracking from a random point that has not been visited. The tracking algorithm stops when all raw boundary points are visited, so a list of tracked curves is found.

If the endpoint of a curve is nearly connected to another curve within a specified distance, the gap is filled and curves are connected. Also a threshold is used to remove curves that are very short. In the end, the tracked boundary is smoother than the raw boundary and has no gaps. Below is the comparison between these two contours, where the black curve is the raw boundary and the red curve is the smoothed boundary.

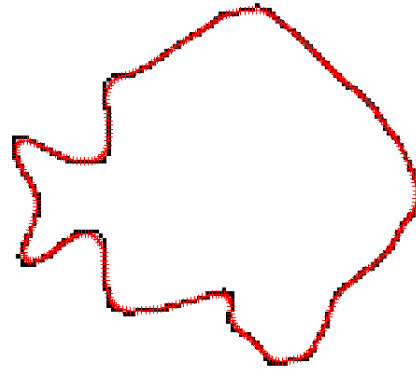
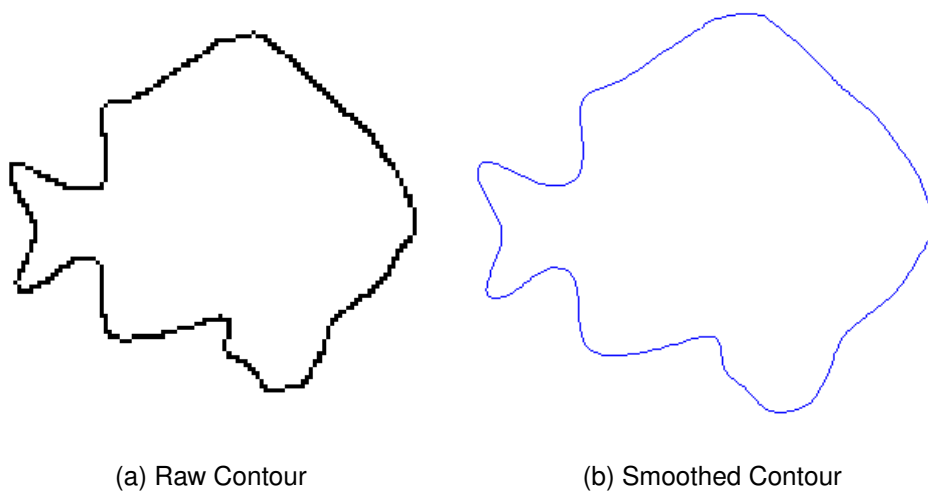


Figure 3.3: Raw Contour(Black) and Smoothed Contour(Red)



(a) Raw Contour

(b) Smoothed Contour

Figure 3.4: Raw Boundary VS Smoothed Boundary

3.1.3 Curvature Calculation

After retrieving the smoothed fish contour, the next step is to compute the curvature of each point along the boundary. The reason why computing curvatures of boundary pixels is to find the corners of the fish contour. By doing that, important locations, such as tail tips, the joints points of tail and body are supposed to be located.

The equation for computing the curvature of point $u = \{x, y\}$ is:

$$k(u) = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{\frac{3}{2}}} \quad (3.1)$$

where x' and y' represent the first derivative of x and y , x'' and y'' denote the second derivative of x and y .

However, curvatures computed boundary points are always very sensitive to local corners and the values can be very extreme. So the logarithm function is used to normalize curvatures:

$$k_{normalized} = \begin{cases} \log(k) & \text{if } k \geq 1 \\ -\log(2-k) & \text{if } k < 1 \end{cases} \quad (3.2)$$

For a fish image as below:

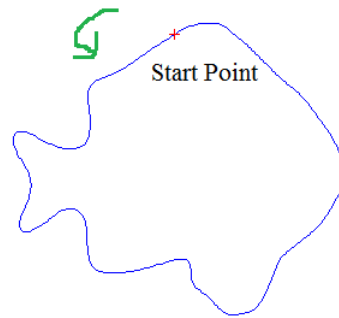


Figure 3.5: A Sample Fish Image

The curvatures of boundary points can be plotted as:

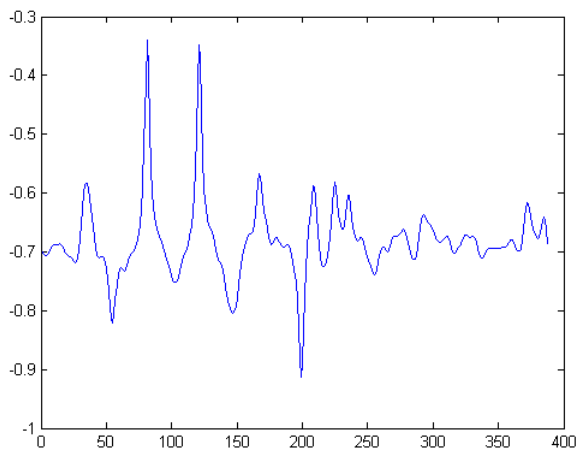


Figure 3.6: Curvature Plot of a Fish Sample

As shown in the above image, the curvatures of contour points change quickly along the fish boundary. And those extreme points in the peaks are assumed to be the important corner points which are candidate locations for tail tips and the joints of tail and body. Then the next task is to locate these extreme points.

The search algorithm is quite simple which is a linear search method. It starts with a positive flag indicating extreme point with positive curvature is supposed to be found. Then it loops through the curve points if the curvature keeps increasing. When the curvature of a point drops and the change of curvature exceeds the threshold value, that point is regarded as a positive extreme point and the flag changes into negative. It continues looping through the curve points, but starts looking for an increase in curvature. When the curvature starts increasing, a negative extreme point is found and the flag changes back to positive. The algorithm keeps repeating this procedure until all points in the boundary have been visited.

After running the algorithm described above, all extreme points could be located as follows:

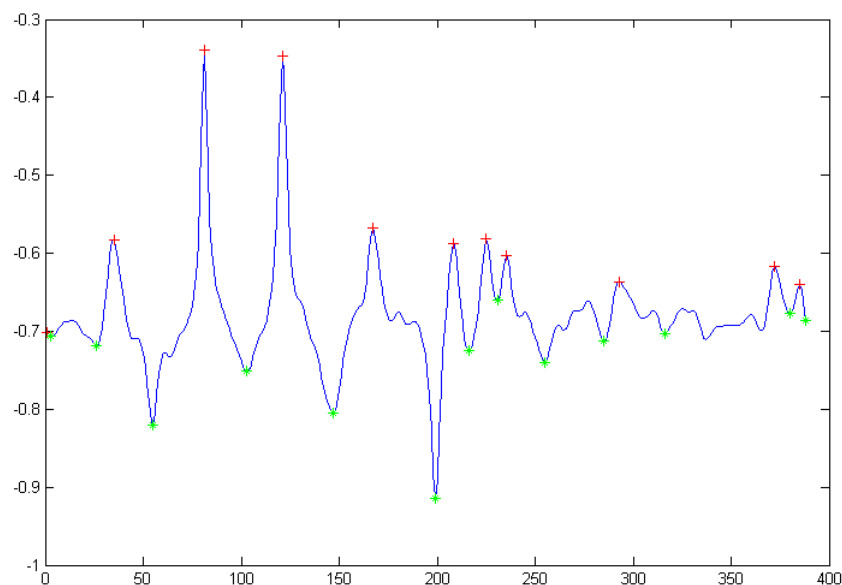


Figure 3.7: Extreme Curvature Points of a Fish Sample

The red cross(+) of above image represents the positive extreme points, and the green star(*) shows the negative extreme points. As could be seen, some local extrema with small changes in curvature are ignored. If we plot these extreme points back to the fish contour, then:

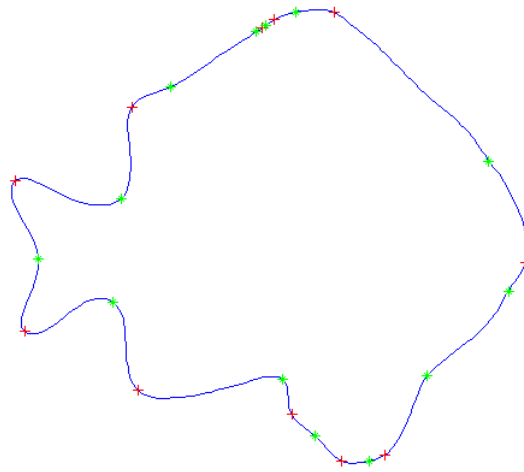


Figure 3.8: Extreme Curvature Points in the Fish Contour

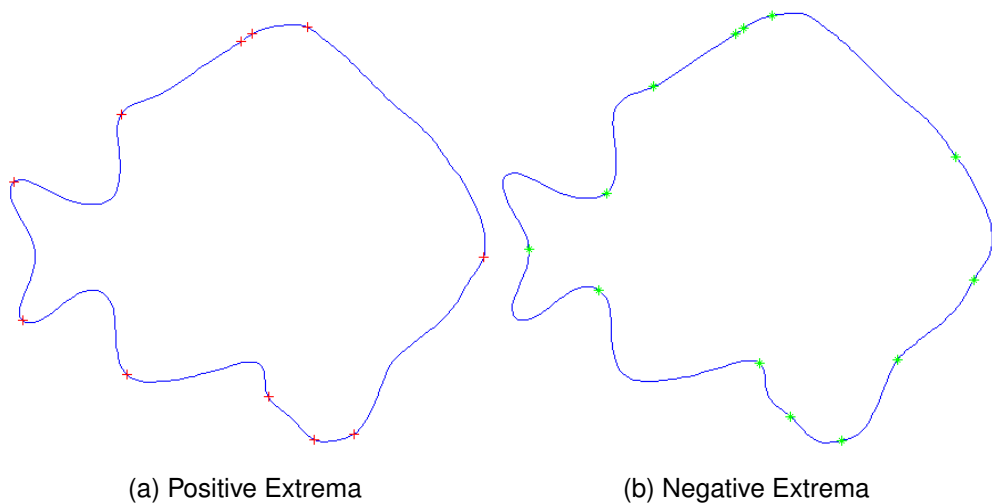


Figure 3.9: Positive Extrema VS Negative Extrema

3.1.4 Tail Segmentation

The tail segmentation is based on the outcome of previous steps. Previously the contour of fish is smoothed and extracted from the binary fish image. Then the curvature of each point on the fish boundary is calculated. The extreme corner points have also been located when looping through the curvatures. Then these points with the largest curvatures are marked as extreme points, in other words they are candidate points of the joint points of tail and body.

Below are some examples of fish curves extracted by the tracking algorithm along with extreme curvature points. The fish contour is in blue and the extreme points are

marked as green stars(*) and red crosses(+). Green stars(*) represent extreme points on convex curves with negative curvatures, while red crosses(+) represent extreme points on concave curves with positive curvatures.

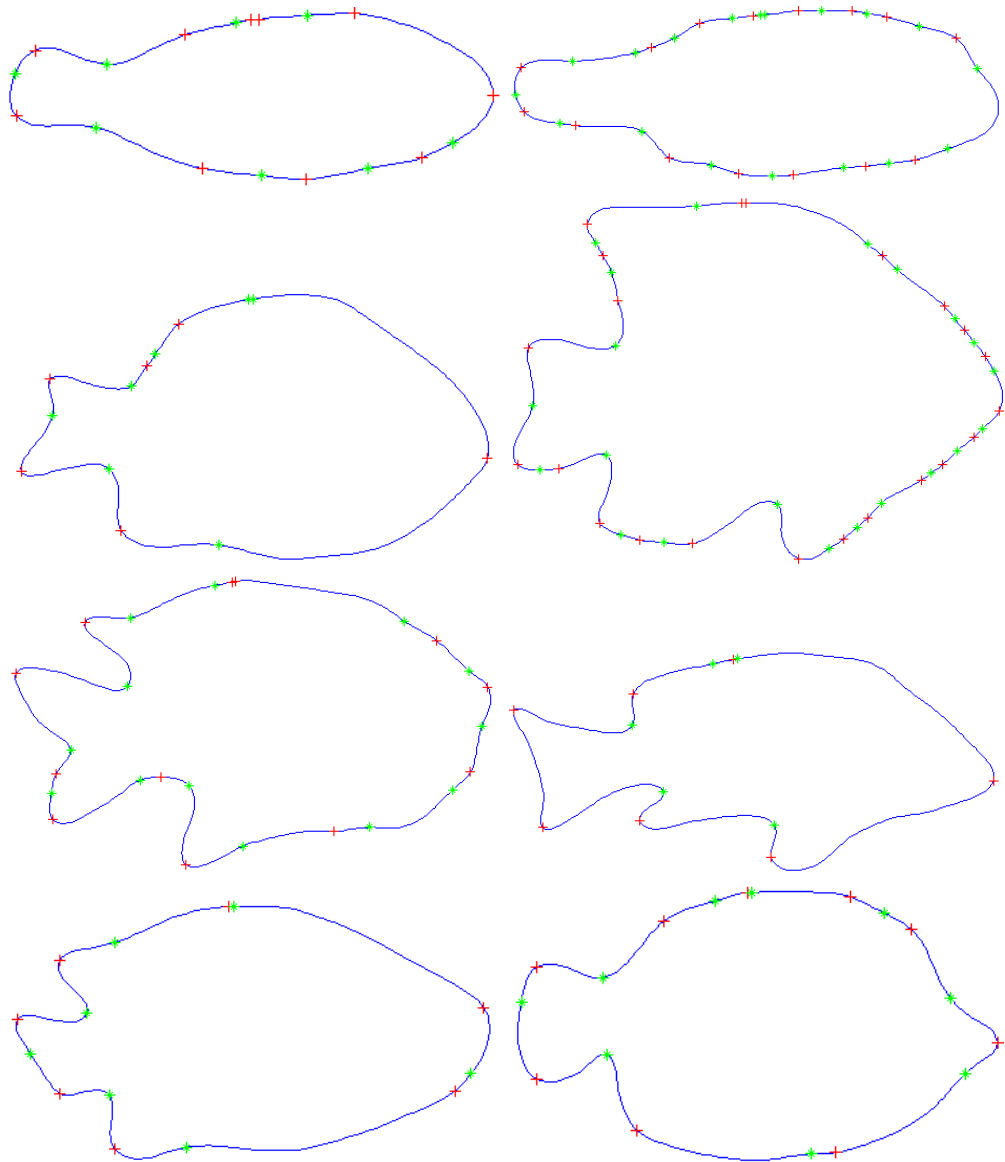


Figure 3.10: Fish Contours and Extreme Points for Different Samples

Observing the above images, joint points are always the green stars(*) and tail tips are always the red crosses(+). Then the locations of these extrema, especially the negative extreme points, can be regarded as the candidate location of joints.

After locating these candidate points, finding the joint points is closer. But still a lot of work needs to be done. Ideally joint points are supposed to be the two extreme points on convex curves which are green stars(*) shown in the above images. However,

due to the variation of species and difference in each fish, directly locating joint points by looping through the negative extreme points is not possible. Because the rankings of these joint points by absolute curvature value vary by fish species, directly using this information is not very reliable. So a different approach has to be adopted.

From the previous chapter, fish are assumed to be horizontally located from tail to head. So the tail is always on the left under this assumption, then locating the tips of tail fins is easier and more reliable. The tail segmentation algorithm starts from locating tip points and then searches through the joint points based on these located tail tips. At first it takes the leftmost positive extreme point as the one of the two tips. The algorithm continues searching through the positive extreme points from left to right, which aims to locate the other tip point.

By observation, it can be concluded that tail tips usually have the top positive curvatures. So many of the candidate locations could be threshed out by ranking the curvatures. And the closest qualified positive extreme point which also satisfies the minimum and maximum distance criteria is regarded as the other tail tip. And by comparing the row values of these two points, the upper tip and lower tip could be distinguished.

The next step then is to locate the joints, and this time only the top m negative (green *) candidate points are examined. The assumption made in this step is that the joints are supposed to be the negative extrema which are the closest to the tail tips. The algorithm starts searching each joint from either tip separately. From the upper tip, it searches in a clockwise direction and picks the closest one. From the lower tip, it searches in an anti-clockwise direction. The two that are closest to the tip points are selected as the joint points.

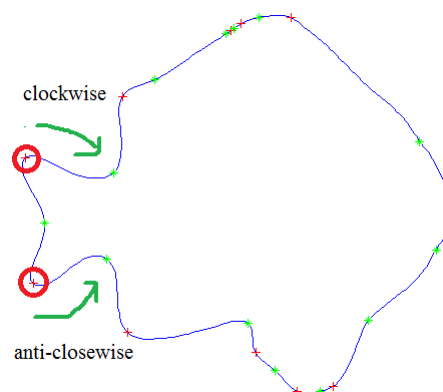


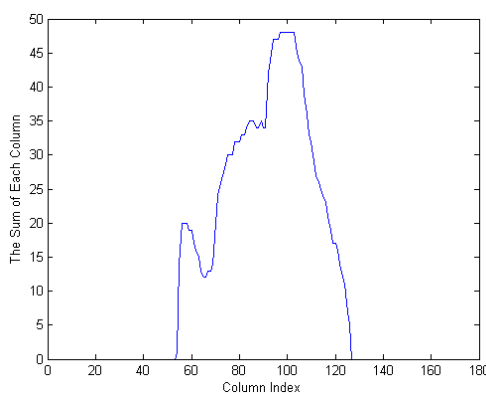
Figure 3.11: Joints Search Example

A constraint of distance between these two joint points is added, so that the joints should not be too far away from each other. If joints do not satisfy the constraint and the algorithm fails to locate the joint points, then a backup segmentation procedure is used.

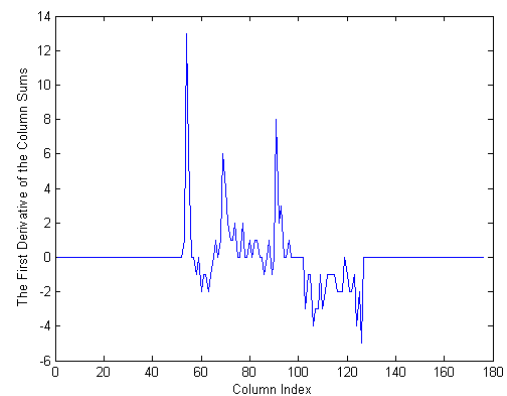
This backup algorithm is straightforward and less accurate, but it works alright to be a backup. It assumes that fish are located horizontally, and joint points are at the same column. The algorithm sums up the binary fish image by column and then calculates the first derivative of column sums. Below is the relation of column number and column sum for a sample fish:



(a) Sample Fish Image



(b) Column Sum Plot



(c) First Derivative of Column Sums

Figure 3.12: An Example of Backup Tail Segmentation

Ideally joint points should have a strong response to the change of column sum, because the width of tail is always smaller than that of body. So by taking the deviation and ignoring the small changes, the joint points could be located. The extreme point locating procedure works very similar to the one used in previous tail segmentation algorithm which also eliminates the impact of small changes. After peak points are located, the algorithm treats the second peak value of column sums (from left to right) as the segmentation column. Then tail is segmented vertically by that column.

After the previous stage, either two joint points are located or a vertical line is retrieved. In either way, the segmentation line could be represented as:

$$a_1r + a_2c + a_3 = 0 \quad (3.3)$$

or

$$c = a_4 \quad (3.4)$$

So for every pixel in the binary fish image, if the pixel lies in or on the left of the line, then it is preserved in the segmented tail image. Otherwise it is removed. This procedure might end up with several unconnected part. In order to fix this problem, one more step is to fill in small gap and eliminate small regions.

Below are examples of tail segmentation using two different algorithms:



Figure 3.13: Example of Tail Segmentation

The green curve in the right image is the smoothed fish curve after curve generating. The red are joint points located by the tail segmentation algorithm. And the blue line is border line for segmentation. The image on the left is the segmentation result.

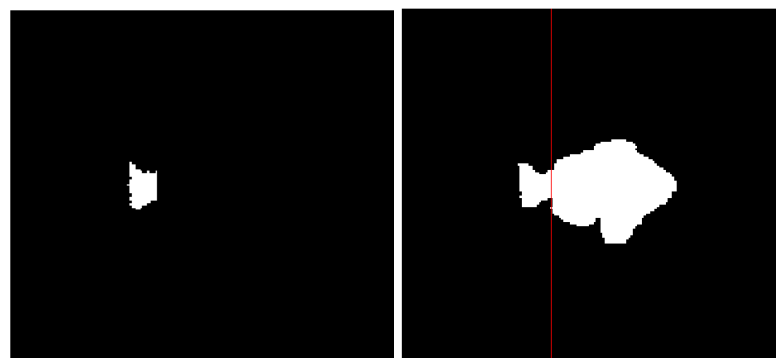


Figure 3.14: Example of Backup Tail Segmentation

The image on the left is the segmentation result. In the right image, the red vertical line represents the column used to segment the tail.

3.2 Summary

In this chapter the algorithms and techniques used in fish component segmentation were explained. At first the boundary tracking algorithm used to extract a smoothed fish boundary was introduced. Then the method to compute the curvature of each boundary point of smoothed fish contour was discussed. After that two tail segmentation algorithms were illustrated, one of which was the principal method and the other as backup algorithm.

After all these procedures, fish tails were segmented which would then used as the dataset for the classification problems. The result and evaluation of tail segmentation are reported and discussed in Chapter 5.1.

Chapter 4

Classification

This chapter illustrates the features used to represent fish components, discusses the methods for feature selection and explains the classification procedure.

Two types of classification are considered in this project. The first classification problem is to classify features into different shape classes(see figures 4.1-4.3), which is based on the component categorizing results. It focuses on the classification of each fish component. The inputs are the features of each component, such as PCA projection and other properties. The outputs are the shape categories of each component.

The second classification problem is to distinguish different species of fish based on component features. This time instead of classifying the shapes of each component, the species of fish are trained and classified. For the fish species classification problem, the input set is the combinations of PCA projections and other features of fish component and the output is the species of fish.

The dataset for these classification problems consists of 15 fish species and each species has 20 fish images. This dataset is used for both the principle component analysis and feature selection.

4.1 Naive Bayes

A Naive Bayes classifier[17][14] is used to learn and distinguish between fish species and component classes. The Naive Bayes classifier is a simple probabilistic model based on Bayes' theorem which assumes that attributes are conditionally independent of each other. Bayes' theorem:

$$P(C|A) = \frac{P(A|C)P(C)}{P(A)} = \frac{P(A|C)P(C)}{\sum_c P(A|C)P(C)} \quad (4.1)$$

If $A = \{a_1, a_2, a_3, \dots, a_n\}$ denotes an instance of a property, b_i denotes each value of a_i and C denotes the class label, then the Naive Bayes classifier could be represented as follows:

$$P(C = c_j|A) = \frac{P(a_1, a_2, \dots, a_n|C = c_j)P(C = c_j)}{P(A)} = \frac{\prod_i P(a_i = b_i|C = c_j)P(C = c_j)}{P(A)} \quad (4.2)$$

where $P(C = c_j|A)$ represents the probability of instance A belonging to class c_j , $P(C = c_j)$ is the prior probability of class c_j and $P(a_i = b_i|C = c_j)$ is the probability of attribute a_i belonging to c_j in the training set with value b_i . When taking the logarithm, the probabilistic model of Naive Bayes can be written as:

$$\log P(C = c_j|A) = \log\left(\frac{\prod_i P(a_i = b_i|C = c_j)P(C = c_j)}{P(A)}\right) \quad (4.3)$$

$$= \sum_i (\log P(a_i = b_i|C = c_j)) + \log P(C = c_j) - \log P(A) \quad (4.4)$$

The last term $\log P(A)$ is identical for the same instance and could be eliminated when comparing the posterior.

In general, the Gaussian distribution is used here to model the probability distribution of continuous data. Using the maximum likelihood solution, parameters of the Gaussian distribution could be estimated from the training data. And in this project, the Gaussian distribution is another assumption for Naive Bayes. The univariate Gaussian distribution is:

$$N(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x - \mu)^2\right\} \quad (4.5)$$

where μ is mean of the distribution and σ is the standard deviation, which could be both estimated from the training set.

The disadvantage of Naive Bayes is the conditional independence assumption which assumes that every attribute is independent from each other when conditioning on the class label. The conditional independence assumption is very weak and not true in most cases, but surprisingly the Naive Bayes classifier usually achieves a relatively good result and is quite popular in many research fields.

One key advantage of Naive Bayes classifier is that it does not need to estimate the full covariance matrix and only standard deviations of each class if using the Gaussian distribution. In another sense, the Naive Bayes classifier requires a small amount of training data in order to estimate the parameters. Unlike estimating covariance matrix which takes at least N instances to estimate an N feature covariance matrix, a Naive Bayes classifier needs as few as 2 samples to estimate the parameters.

Another advantage of the Naive Bayes is its training speed which is very promising compared with other popular machine learning classifiers. In this project there are 15 species in the dataset, and there will be hundreds of species in the real underwater environment. While using a binary classifier, such as a Support Vector Machine and Artificial Neural Network, would take much more time than the Naive Bayes classifier. As this project is more about researching on features used to represent fish, rather than the classification of fish species, a Naive Bayes classifier should be capable of achieving the goal.

4.2 Tail Classification

Tail classification is based on the segmentation results. For each fish image, the tail is located and extracted from the image. And the feature properties are calculated and selected to represent each tail. Two types of classifications are applied based on these selected features after that. One is classification of the fish species, which is a 15-class classification problem. The other is the classification of the tail shapes, which categorizes the similar tail shapes as the same class.

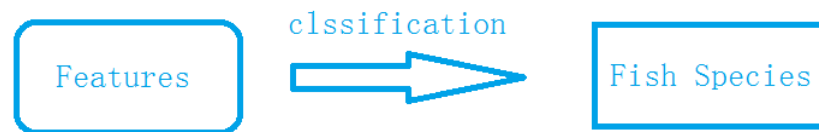


Figure 4.1: Fish Species Classification

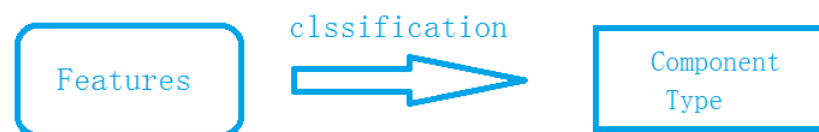


Figure 4.2: Component Type Classification

4.2.1 Tail Classes

After observing the fish images in the dataset, fish tails are categorised into 3 classes based on their shapes. The categorization is not derived from a biological point of view, but it makes sense. Still it is better to have some professional biologists work on

the clustering of tail shapes, which would be more systematic and handy when dealing with hundreds more species.

These tail categories are later used as the class labels in the tail shape classification problem. Below are some examples of the RGB fish images and their extracted tail images.

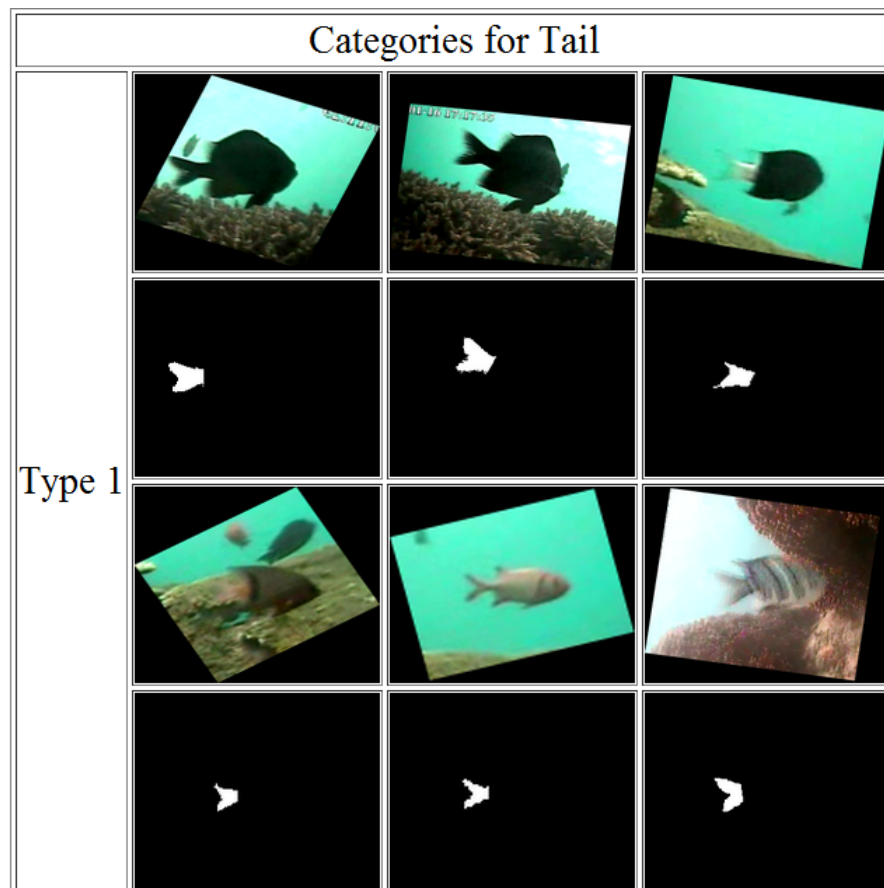


Figure 4.3: Examples of Fish Tail Type 1

Above are some examples of type 1 tail which consists 5 out of total 15 species. This type of tail has a forked shape and can be symmetrically split from the middle point. There are two obvious tips at the end of the tail. Regardless the detailed differences between these species, their tails all have the 'v' shape.

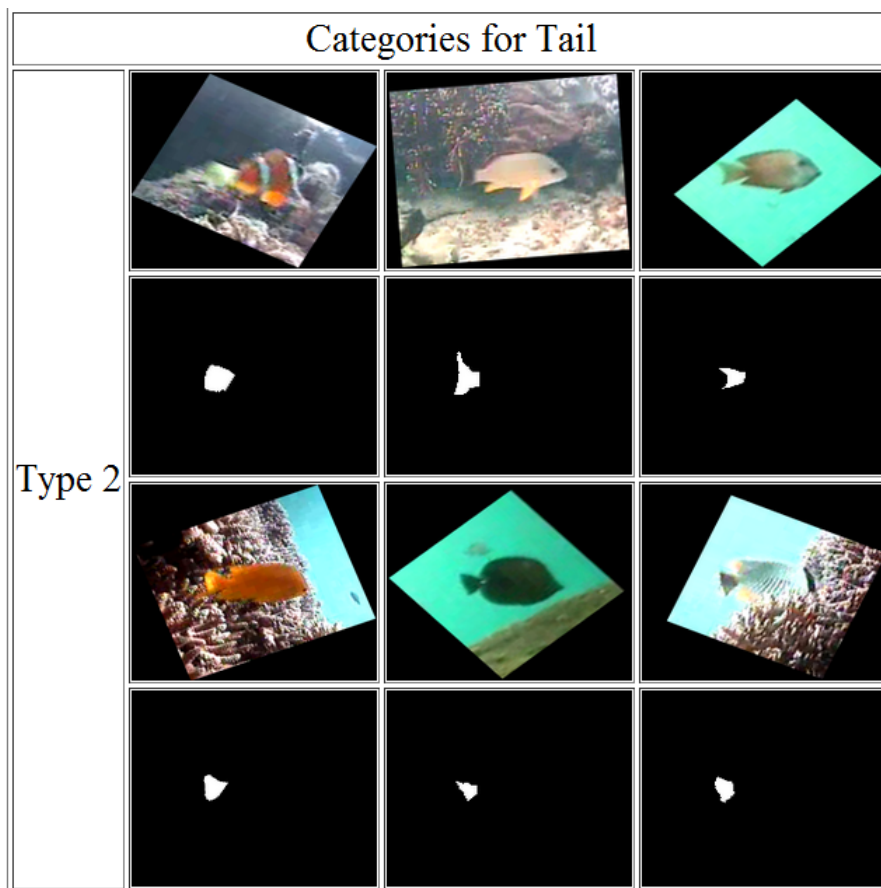


Figure 4.4: Examples of Fish Tail Type 2

Above are some examples of type 2 tail. This type of tail has no obvious tips at the end and has a triangular shape. In the dataset type 2 consists 9 out of total 15 species. Compared with type 1 shape and type 3 shape, type 2 shape is neither forked shape nor looks like stick with no caudal fin. That is the most logical concern to categorize fish species into type 2.

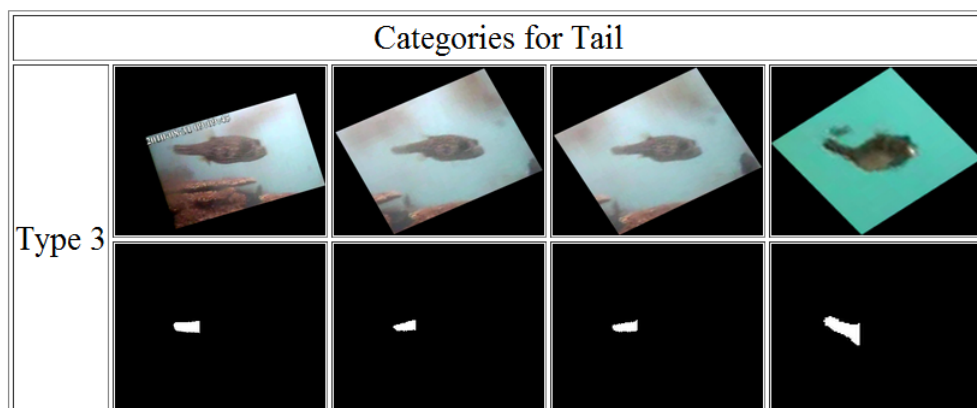


Figure 4.5: Examples of Fish Tail Type 3

Above are some examples of type 3 tail. As could be seen, this type of tail looks like a stick and there is no obvious caudal fin at the end of the tail. And this type of tail is relatively rare in the current dataset, only 1 species belongs to this category.

4.2.2 Features

4.2.2.1 Tail Image Transformation

After having the fish tail segmented from the fish image, features used to represent each tail are generated. But before generating any properties, another essential step is applied which is aligning each segmented tail image into the same coordinate system. Below is the example of the common tail coordinate system:

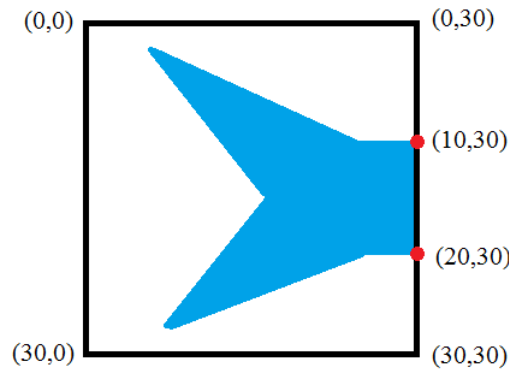


Figure 4.6: The Common Coordinate System of Tail

As shown in the above example, all tail images are rescaled, rotated and translated into the 30×30 matrix, so that the differences caused by scale, rotation can be adjusted. For each point X in the original image, the new position Y in the common tail coordinate system can be calculated as:

$$Y = s * (R * X + T) \quad (4.6)$$

where s is the scale factor, $T = (r_t, c_t)'$ is the translation vector and R is the rotation matrix which can be written as:

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

In order to standardize the reshaping procedure, the two joint points calculated in the segmentation step are used. These two joint points are always placed at (10,30)

and $(20, 30)$ in the new coordinate system. If $A = (r_1, c_1)$ and $B = (r_2, c_2)$ represents the position these two points in the original image, the scale factor s can be calculated easily as:

$$s = \frac{10}{|AB|}$$

where $|AB|$ is the length of vector \vec{AB} , and if we denote the unit vector of \vec{AB} as (u, v) , then the rotation matrix could be calculated as the solution of the following equation:

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \times \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

And the solution is :

$$\begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} = \begin{bmatrix} \frac{u}{\sqrt{v^2+u^2}} \\ \frac{-v}{\sqrt{v^2+u^2}} \end{bmatrix} = \begin{bmatrix} u \\ -v \end{bmatrix} \quad (4.7)$$

Then translation vector could be calculated as:

$$T = \frac{1}{s} \times \begin{bmatrix} 10 \\ 30 \end{bmatrix} - R \times A = \frac{1}{s} \times \begin{bmatrix} 10 \\ 30 \end{bmatrix} - \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \times \begin{bmatrix} r_1 \\ c_1 \end{bmatrix} \quad (4.8)$$

Below are some examples of the tail images after transformation:

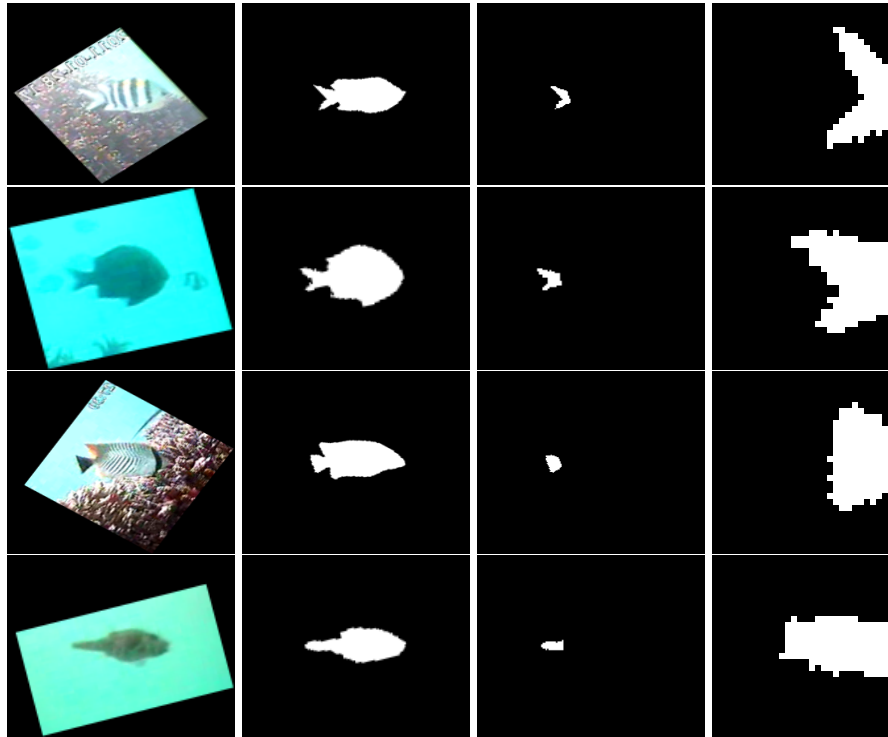


Figure 4.7: Examples of Reshaped Tails

Column one is the RGB colour image; Column two is the binary fish image; Column three is the fish tail image after segmentation and before transformation; Column four is the tail image after transforming into the new coordinate system.

4.2.2.2 Principal Component Analysis

After transforming all tail images into the same coordinate system, all transformed tail images are 30×30 matrices. Each tail image now contains 900 attributes, which are too many for the classification problems. Principal component analysis is applied in order to reduce the dimensionality. And the reduced feature vector will be then used as part of the feature properties.

Principal Component Analysis[4][1] is a popular tool for feature selection and feature reduction, which projects features onto a different space. When using PCA, the mean image patch will firstly be calculated:

$$\bar{X} = \frac{1}{N} \sum_{n=1}^N X_n \quad (4.9)$$

Then subtract the mean value and calculate the covariance:

$$S = \frac{1}{N-1} \sum_{n=1}^N (X_n - \bar{X})(X_n - \bar{X})^T \quad (4.10)$$

Then the eigenvectors of the covariance, $E = [e_1, e_2, \dots, e_N]$:

$$S e_i = \lambda e_i \quad (4.11)$$

New attributes are represented as:

$$Y = E^T (X - \bar{X}) \quad (4.12)$$

As there are 900 attributes in the transformed tail image, there are then 900 principal components. In order to reduce the dimensionality, the top M features should be selected instead of using all features. Below is the plot of percentage of variation covered in the principal components against the number of eigenvectors used:

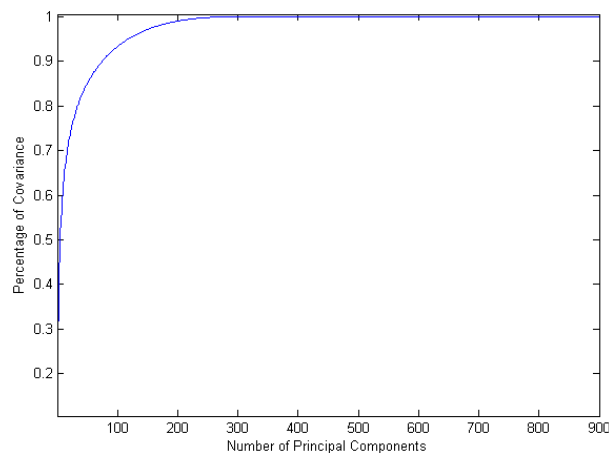


Figure 4.8: The Plot of Cumulated Variation of Eigenvectors

Below is the short version which covers the top 80 eigenvectors:

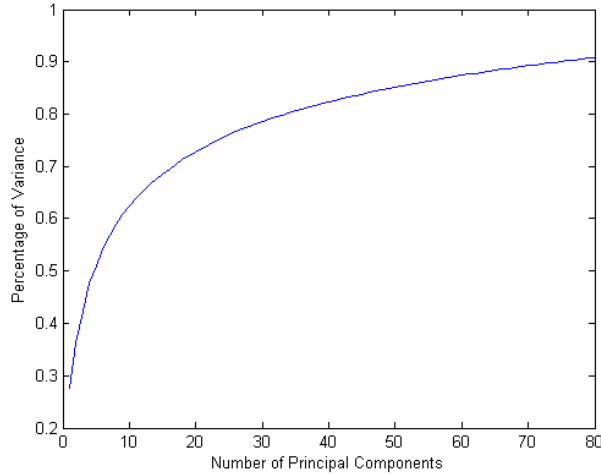


Figure 4.9: The Plot of Cumulated Variation of Top 80 Eigenvectors

Percentage of Variation	65%	75%	80%	85%	90%	92%	95%	98%
Number of Eigenvectors	12	24	35	49	74	88	118	168

Table 4.1: Percentage of Variation Using Different Number of Eigenvectors

The table above shows the explicit numbers of principal components which cover the corresponding percentage of variations. And in this project, top 10 eigenvectors are used to reduce the dimensionality. So that after PCA, there are 10 new attributes left which is much smaller than the original 900 attributes.

4.2.2.3 Image Moments[2]

Other than the 10 attributes selected after principal component analysis, image moments are also used as feature properties to represent the tail. These moments are applied to the tail image after transformation. And for an image I , below is the equation to calculate the raw moment:

$$M_{ij} = \sum_x \sum_j x^i y^j I(x, y) \quad (4.13)$$

So that Area could be represented as:

$$Area = M_{00} = \sum_x \sum_j I(x, y) \quad (4.14)$$

And the centroid (\bar{x}, \bar{y}) can be calculated as:

$$(\bar{x}, \bar{y}) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right) \quad (4.15)$$

Then central moments which are also regarded as the translational invariant moments can be calculated as:

$$\mu_{pq} = \sum_x \sum_j (x - \bar{x})^p (y - \bar{y})^q I(x, y) \quad (4.16)$$

Divided by the area, the scale invariant moments can be computed:

$$s_{ij} = \frac{\mu_{ij}}{\mu_{00}^{(1+\frac{i+j}{2})}} \quad (4.17)$$

Altogether there are 19 properties to represent a fish tail:

Properties Number	Description
1-10	New features after principal component analysis
11	Area of the tail: $\text{Area} = M_{00} = \sum_x \sum_j I(x, y)$
12	Compactness of the tail: $\text{Compactness} = \frac{\text{perimeter}^2}{4\pi\text{area}}$
13	Scale invariant moment s11
14	Scale invariant moment s20
15	Scale invariant moment s02
16	Scale invariant moment s21
17	Scale invariant moment s12
18	Scale invariant moment s30
19	Scale invariant moment s03

Table 4.2: Features Used to Represent Tail

4.2.3 Feature Selection

After computing the PCA projections and calculating the image moments, all together there are 19 features. As described above, these features consist of 10 PCA projections using the top 10 eigenvectors, 7 image moments, 1 area and 1 compactness. In order to have a relatively good classification performance with less features, feature selection algorithms are tested and used. In fact, two types of feature selection algorithms are used. One is the sequential forward feature selection and the other is the information gain. Either of them is used separately to select a subset of features and the classification performances of these selected features are also compared.

The data set used for both sequential forward selection and information gain consist of 15 fish species with 20 images for each species. The reason why using fish species dataset other than component classes dataset for feature selection is because that the fish species classification problem is more difficult to tackle. As only one set of features is supposed to represent a fish component, the one derived from the fish species dataset should be more generalizable than the other.

4.2.3.1 Sequential Forward Feature Selection

Sequential Forward Feature Selection[13][5] is an iterative selection method that starts with an empty subset and adds one feature into the subset each time. One by one, the subset of features is selected which aims to achieve the best classification performance. At each iteration, every feature that has not been included into the subset is trained and tested along with the selected subset. The feature that achieves the best classification accuracy is then added to the subset and excluded from the remaining features.

The sequential forward selection is an exhaustive selection algorithm which tests all combinations of every unselected feature with the selected subset. It is more like a conceptual idea than a solid algorithm which could derive different versions depending on the implementation details. As the sequential forward selection highly depends on training data, different training sets would usually produce different selections. So in this project, a number of trials of the algorithm are evaluated and the termination criterion for the sequential forward selection algorithms is when the expected number of features have been selected.

The reason why using a fixed number of features as the termination criterion instead of choosing a flexible number of features when the performance decreases, is because that the total number of candidate features is relatively small (only 19 fea-

tures) and exhaustively try different number of features will no take too much time. By exhaustively selecting all 19 features, a ranked list of features with their classification accuracies can be obtained, then it is easier to evaluated these results over a flexible number of features.

Below is the pseudo code of the sequential forward selection algorithm:

```
main
  generate N train (2/3 data) sets A and test (1/3 data) sets C
  for i = 1 : N
    train_classifier(A)
    evaluate_classifier(C)
  end
  report max, mean and std dev of classifier

train_classifier(A)
  generate M train (1/2 of A) sets T and validation (1/2 of A)
    sets V
  for i = 1 : M
    do forward_selection estimating parameters using T
  end
  pick best classifier of M
```

Algorithm 1: Algorithm for Forward Selection

In the implementation, 100 sets of training data and testing data are firstly generated. For each set of training data, a forward selection procedure is applied, and a set of selected features is generated.

However, as each time the training set is different, the result of a single selection result is a little bit different. After 100 runs of the forward selection procedures, 100 set of selected features are generated.

Below is a figure of the classification results for validation sets. The X axis shows the number of selected features, and Y axis shows the classification accuracy. The red crosses connected by blue line are the classification accuracy averaging over 100 validation sets, The black crosses above the blue line are the maximum classification accuracies, while blue crosses below the blue line are the minimum classification accuracies. The circles between the maxima and minima are the positions of standard deviation.

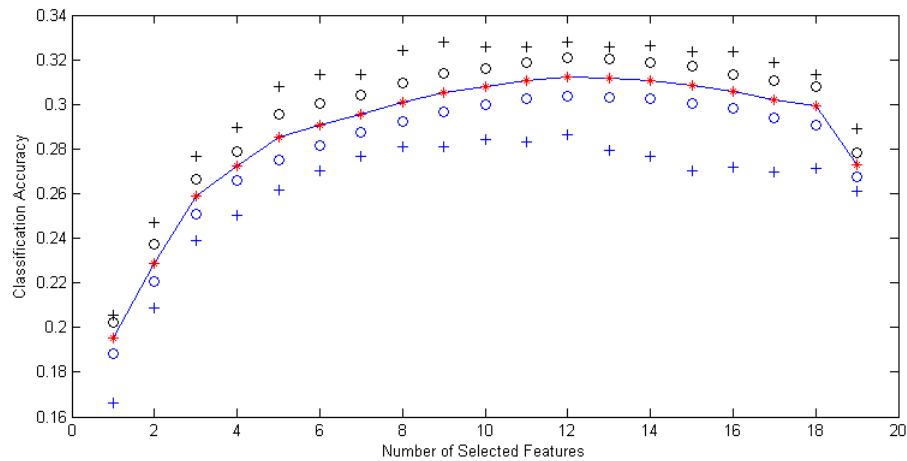


Figure 4.10: Classification Performance

The image below is a histogram of the number of features that achieve the best classification results for the training set. The X axis shows the number of selected features, and Y axis shows the number of sets that also use the corresponding number of features.

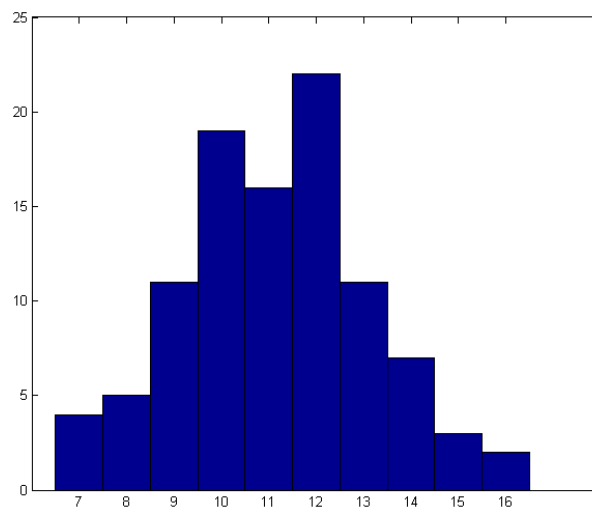


Figure 4.11: Histogram of Number of Selected Features (Training Sets)

Another histogram is also generated which shows the number of features that achieve the best results in the validation set:

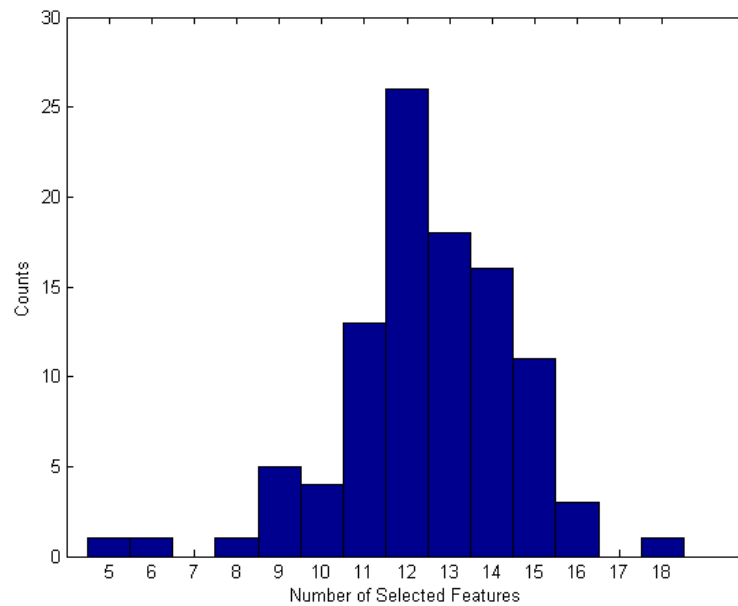


Figure 4.12: Histogram of Number of Selected Features (Validation Sets)

Both the two histograms shows that by using 12 features the classifier usually achieves its best performance. It is very likely that there are about 12 most valuable features. In order to find out these top 12 features, the first 13 features (in case not to miss one valuable attribute) of each selection result are examined. The procedure is very simple which just counts the number of appearances for each attribute and selects the top 12 features.

Below is a histogram that shows the number of appearance of a specified attribute in 100 selected feature sets. The X axis shows the index of the corresponding attribute which could be referred in table 4.2 and the Y axis show the number of appearance:

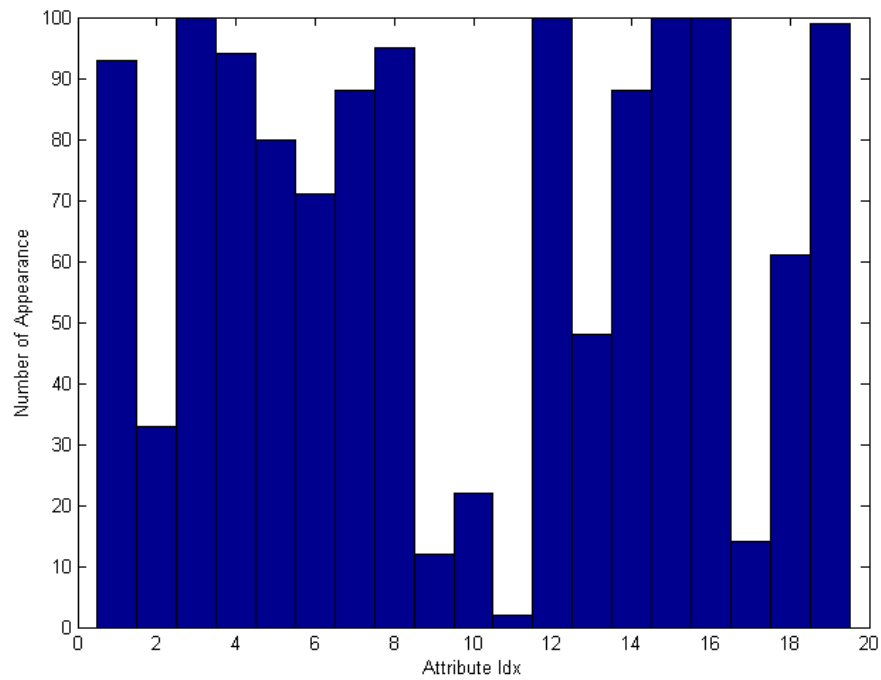


Figure 4.13: Attribute and the Number of Appearance in the Feature Sets

The above histogram can also be represented as the following table:

Attribute Index	1	2	3	4	5	6	7	8	9	10
Number of Appearance	93	33	100	94	80	71	88	95	12	22
Attribute Index	11	12	13	14	15	16	17	18	19	
Number of Appearance	2	100	48	88	100	100	14	61	99	

Table 4.3: Attribute Index and the Number of Appearance in the Feature Sets

Attribute 15, 16, 3 and 12 appear in every selection set and attribute 19, 8, 4 and 1 also have relative large number of appearances in the selection sets.

A list of attributes sorted by the number of appearance could be represented as:

Number of Appearance	100	100	100	100	99	95	94	93	88	88
Attribute Index	15	16	3	12	19	8	4	1	7	14
Number of Appearance	80	71	61	48	33	22	14	12	2	
Attribute Index	5	6	18	13	2	10	17	9	21	

Table 4.4: Attribute Index and the Number of Appearance in the Feature Sets

So the 12 features selected by sequential forward selection are:

Attribute Index	Attribute	Description
15	s02	scale invariant image moment s02
16	s21	scale invariant image moment s21
3	e3	projection using the 3rd eigenvector
12	compactness	compactness of the tail
19	s03	scale invariant image moment s03
8	e8	projection using the 8th eigenvector
4	e4	projection using the 4th eigenvector
1	e1	projection using the 1st eigenvector
7	e7	projection using the 7th eigenvector
5	e5	projection using the 5th eigenvector
6	e6	projection using the 6th eigenvector
14	s20	scale invariant image moment s20

Table 4.5: Features Selected by Sequential Forward Selection Algorithm

4.2.3.2 Information Gain

Information Gain[7] is another very popular method that can be used to select important attributes from a set of candidate features. The Information Gain is actually another term for Kullback–Leibler divergence, which is widely used in information theory and machine learning. A typical example of Information Gain is the decision tree where it is used to pick an importation node when building the tree. In information theory, the entropy H of a discrete random variable X with possible values x_1, \dots, x_n and probability mass function $p(X)$ is represented as:

$$H(X) = \sum_{i=1}^n p(x_i) \log \frac{1}{p(x_i)} = - \sum_{i=1}^n p(x_i) \log p(x_i) \quad (4.18)$$

If C denotes all classes and A represents an attribute then Information Gain could be expressed as:

$$\text{InformationGain}(C,A) = H(C) - H(C|A) \quad (4.19)$$

The expected value of $\text{InformationGain}(C,A)$ is the mutual information $I(C;A)$ which is the information gained from learning on the state of attribute A .

The advantage of Information Gain over Forward Selection is that the computation cost is much lower which does not require the classifier to train and test using different combinations of features. The only thing it needs is to deal with all kinds of probabilities and conditional probabilities in the dataset.

However, Information Gain pays too much attention to attributes that could best separate class labels. Sometimes that could cause problems and select misleading attributes. If there are a large number of distinct values for an attribute, for example an attribute for people names which are very distinct from each other, the Information Gain of that attribute will be very high. But very likely that attribute is not very helpful for classification.

The Information Gain is actually not implemented in this project, instead Weka is used to fulfil the Information Gain function and evaluates the selected features with 5-fold cross-validation.

After running the Information Gain on the whole dataset and ranking attributes by their gain values, below is the table of ranked attributes (the result is achieved by the Weka implementation):

Rank	Attribute	Description
1	e4	projection using the 4th eigenvector
2	s02	scale invariant image moment s02
3	s20	scale invariant image moment s20
4	compactness	compactness of the tail
5	s21	scale invariant image moment s21
6	e5	projection using the 5th eigenvector
7	e7	projection using the 7th eigenvector
8	e6	projection using the 6th eigenvector
9	e2	projection using the 2nd eigenvector
10	e3	projection using the 3rd eigenvector
11	e1	projection using the 1st eigenvector
12	s12	scale invariant image moment s12
13	s11	scale invariant image moment s11
14	s03	scale invariant image moment s03
15	s30	scale invariant image moment s30
16	e9	projection using the 9th eigenvector
17	e8	projection using the 8th eigenvector
18	area	area of the tail
19	e10	projection using the 10th eigenvector

Table 4.6: Table of Ranked Attributes by Information Gain

N Top Features	Classification Accuracy	N Top Features	Classification Accuracy
1	15.7%	2	26.1%
3	23.7%	4	28.1%
5	29.8%	6	32.1%
7	35.1%	8	35.8%
9	32.1%	10	34.8%
11	34.1%	12	34.1%
13	32.8%	14	33.8%
15	32.8%	16	32.8%
17	32.8%	18	33.8%

Table 4.7: Classification Accuracy Using Top N Features

Above is a table of classification accuracies using different number of top features ranked by Information Gain, and the classification accuracies are achieved by 5-fold cross-validation. The classification accuracy of using the top 8 features achieves the best performance, which is 35.8%. These features include e_4 , s_{02} , s_{20} , compactness, s_{21} , e_5 , e_7 and e_6 .

When comparing with the 12 features selected by sequential forward selection, a very interesting finding is that these 8 features are all included by the forward selection algorithm although their order of importance might be different. The forward selection algorithm adds another 4 features into the dataset, and achieves a classification result of 37.5% (the result is also achieved by 5-fold cross-validation using Weka), which is a little better than the Information Gain. As a result, the 12 features selected by sequential forward selection is then used for the two classification tasks. The classification results are discussed in Chapter 5.2.

4.3 Summary

In this chapter, two types of classification problems for this project were introduced. The advantages and disadvantages of Naive Bayse were also compared and discussed.

When precessing fish tails, all tail images were rescaled, rotated and transferred into a new coordinate system of size 30×30 . Principle component analysis and image moments were then used to generate a set of features. The algorithms and equations for these computations were also illustrated and discussed.

In order to have a robust represent of fish tail, a sequential forward feature selection algorithm was developed and used to improve the classification accuracy and use less features. In comparison, Information Gain was also used to evaluate and select a set of candidate features.

Chapter 5

Evaluation

This chapter evaluates all results of previous algorithms. It starts with presenting the results of those algorithms and then evaluates the performance. Performance of component segmentation is first evaluated. After that feature selection results and the corresponding classification results are reported and evaluated. This includes the classification of fish component categories and the classification of fish species using component features.

5.1 Segmentation Performance

All segmentation results are evaluated against two sets of the ground truth data, one of which is produced from annotating component in the RGB fish images and the other is annotated in binary fish images after fish extraction. As the component segmentation algorithms are applied to the binary images, the ground truth results generated from the binary images are used as the actual measurement of performance. In contrast, the ground truth results generated from RGB images are regarded as the ultimate measurement of performance, because the provided binary masks are not perfectly generated and sometimes important details are not captured. The reason why having two ground truth data is to measure the general segmentation performances using bias data and at the same time compares the results with the perfect segmentation.

However, ground-truth data for segmentation result is not provided and has to be manually annotated point by point. And there are thousands of fish images, it takes too much time and effort to annotate all fish images. So a compromising approach is to randomly select a reasonable amount of images from each species and then manually annotated each fish component. In this project, 20 fish from each species are randomly

selected and altogether there are 300 fish images. The annotating work is done through an active shape model tool written in C++ by Dr Tim Cootes¹.

5.1.1 Performance Measure

In order to evaluate the segmentation result, a measurement of good segmentation performance must be defined first. If the segmentation outcome is denoted as S , and the ground truth as G , then the segmentation accuracy is measured as[6]:

$$P(S, G) = \frac{|S \cap G|}{|S \cup G|} = \frac{|S \cap G|}{|S| + |G| - |S \cap G|} \quad (5.1)$$

where $|\cdot|$ represents the operation of computing the area. $|S \cap G|$ measures how well the part is segmented by calculating how much area of ground-truth structure is detected. $|S \cup G|$ actually performs as a normalizing factor which constrains the segmentation accuracy to be in $[0, 1]$. So that this measurement is invariant to the size of the area to be segmented.

For a list of segmentation results, the correctness of segmentation can be justified by setting up a threshold. If the segmentation accuracy is above the threshold, then this sample is counted as a correctly segmented result. Otherwise, it is incorrectly segmented. Thus for the list of segmentation accuracies A and threshold t , the percentage of satisfaction p could be computed as:

$$p(A, t) = \frac{\sum_{a \in A} f(a, t)}{|A|} \quad (5.2)$$

where a represents the accuracy of a segmentation result, $|A|$ is the total number of instances and f is:

$$f(a, t) = \begin{cases} 1 & \text{if } a > t \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

The percentage of satisfaction gives an idea about how much portion of segmentation result in the evaluation set could be regarded as correctly segmented using different thresholds.

¹http://www.isbe.man.ac.uk/bim/software/am_tools_doc/index.html

5.1.2 Ground-truth Tails

As discussed previously, two sets of ground truth data are used to evaluate the component segmentation from different perspectives. For the tail segmentation, both the ground-truth data generated from RGB fish images and that generated from binary fish images are also used. In order to keep terms short, the ground-truth data generated from RGB fish images is denoted as RGB ground-truth and binary ground-truth for the other.

In order to achieve a relatively accurate ground-truth segmentation result and spend less effort on this time-consuming annotating work, 9 points are used to represent the annotated tail shape using RGB images for all species. Below is an example of the model using 9 points:

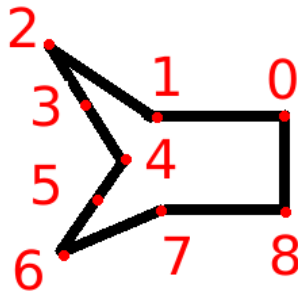


Figure 5.1: Tail Representation Using 9 Points

In this model, point 0 and point 8 represents the joint points between tail and body and point 2 and 6 are the tips of the tail fin. Below is an example of the ground truth value projected onto RGB fish image:



(a) RGB Fish

(b) Zoomed Annotated Tail

(c) Zoomed Filled Tail

Figure 5.2: Example of Ground-truth Tail Annotated from RGB Images

For the ground truth segmentations generated from binary fish images, the proce-

ture is much easier where only two joint points are needed as the segmenting points. A partition line could then be defined by these two points where any pixel to the left of that is preserved. This procedure is identical to the one used in segmentation stage, but this time two joints are given.

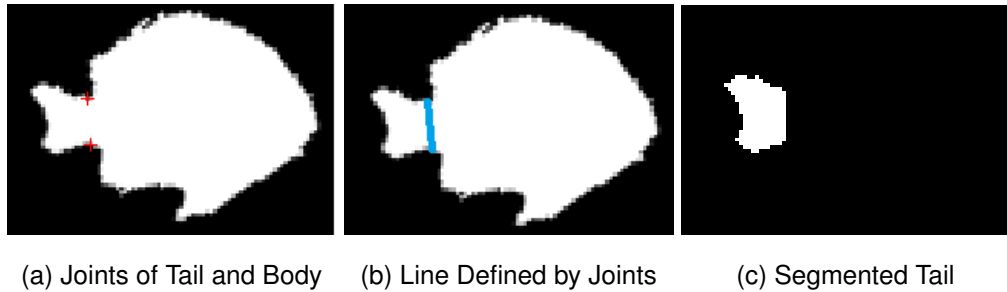


Figure 5.3: Example of Binary Ground-truth

5.1.3 Segmentation Results

As has discussed above, 20 fish images are randomly picked from each species and altogether there are 300 fish images. When running the tail segmentation algorithm, a set of tail images could be retrieved. The images below are some examples of the random selected fish image and their tail segmentation results.

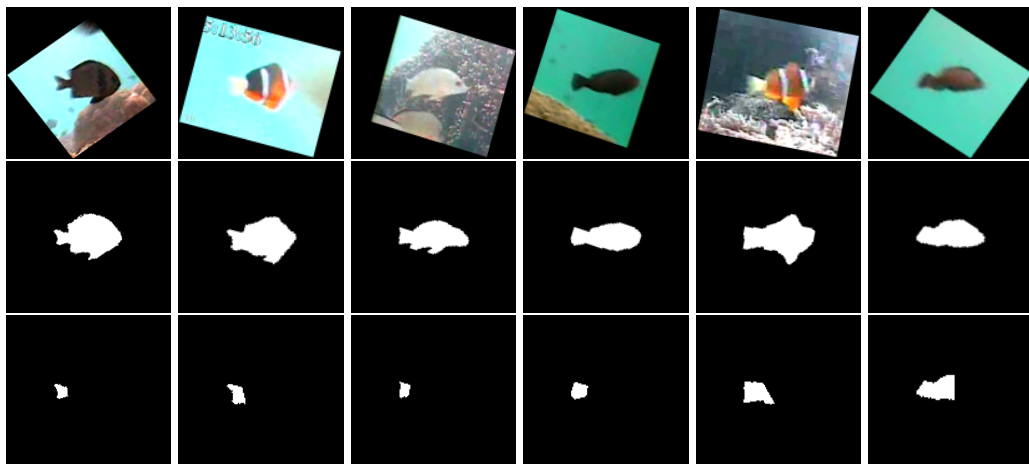


Figure 5.4: Examples of Tail Segmentation

Images in the first row are RGB fish images, the second row is about binary image after extraction and the last are the tail segmentation images. The first 4 columns are samples that have relatively better performance and the last 2 columns are samples with poor segmentation.

When evaluating the tail segmentation result using both ground-truth tails, the following percentages of satisfaction under different thresholds can then be computed:

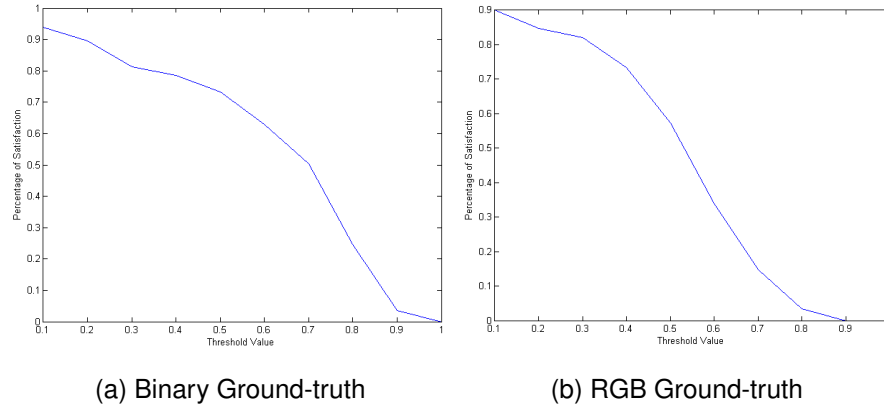


Figure 5.5: Percentage of Satisfaction vs Threshold

Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
PoS	94%	89%	84%	78%	73%	63%	50%	24%	3%

Table 5.1: Percentage of Satisfaction (Binary Ground-truth)

Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
PoS	90%	85%	82%	73%	58%	34%	14%	3%	0%

Table 5.2: Percentage of Satisfaction(RGB Ground-truth)

The evaluation results retrieved by comparing with binary ground-truth tail can be regarded as the actual performance of tail segmentation algorithm. As the tail segmentation algorithm is actually performed on these binary fish images, the quality of fish extraction directly effects the performance of the segmentation algorithm.

In contrast, the evaluation results using RGB ground-truth tail are seen as the ultimate measurement of the segmentation algorithm, where limitations resulted from the fish extraction are not considered. Therefore, by having this kind of comparison, the gaps between perfect segmentations can also be evaluated.

5.1.4 Analysis of Tail Segmentation Performance

When viewing from the plots of variate percentages of satisfaction, the tail segmentation result seems not very promising. While there are several reasons why producing such a performance. Before analysing the factors behind the scene, a deeper analysis should be conducted about the performance. Below is a combinations of evaluations using two different ground-truth tails.

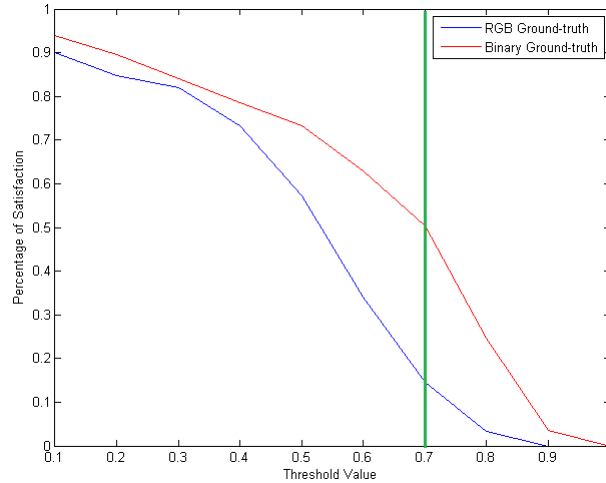


Figure 5.6: Percentage of Satisfaction vs Threshold

The blue line represents the percentages of satisfaction computed using binary ground-truth tails, while the red line is compared with the RGB ground-truth tails. X axis is the threshold value used to decide the satisfaction of segmentation, and Y axis shows the corresponding percentage of satisfaction with different threshold.

As has discussed previously, the percentage of satisfaction is equal to the intersection area normalized by the union of the two areas. Because of the small resolution of fish images (usually less than 200 pixels for a tail), a small amount of differences in pixels could produce quite a big difference in segmentation satisfaction. For example for a tail image of 200 pixels, if 40 pixels is incorrectly segmented, then the segmentation accuracy drops quickly to 66% from 100%. Although 40-pixel difference is not very distinguishable for human eyes. When analysing the percentage of satisfaction, the value 0.7 seems to be a good threshold to measure a good segmentation result. In this way, small pixel difference could be adjusted.

When using 0.7 as the threshold, 50% of tail images in the evaluation set can be regarded as correctly segmented with binary ground-truth and 15% correct with RGB ground-truth tails. The evaluation result derived by binary ground-truth is acceptable

and much better than that derived by RGB ground-truth. This implies that the segmentation algorithm works OK on the given binary fish images, but there is some distance compared to the ultimate segmentation.

One point worth to be mentioned is that even when the threshold is set to 0.1, there still a small portion of tails that are regarded unsuccessfully segmented. This actually means no matter what the threshold is, the algorithm fails to localize the rough position of some tails. By looking into details of this problem, two main reasons can be concluded.

The first reason is that the tail segmentation algorithm is not very generic, which takes several assumptions about the position and the pose of the fish. For example, the algorithm assumes that all fish are perfectly aligned left to right, tail to head. And the contours of fish are clear and detailed. However, comparing with the abundant fish species and individual image, these assumptions seem to be too constrained and not generic enough to apply well in most conditions. Here are some examples in the evaluation set that do not satisfy these assumptions:

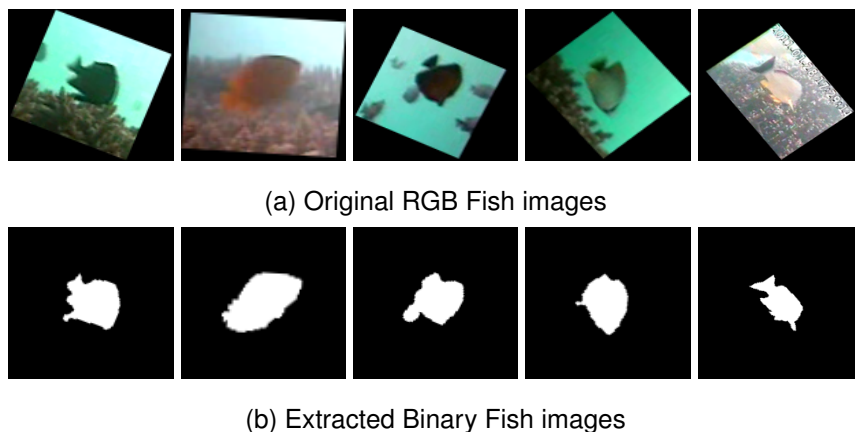


Figure 5.7: Outlier Fish Examples

The above images are not horizontally orientated as the previous assumption, some of which are almost vertical.

Other than the orientation issue, the second reason is that the extracted binary fish images are not very accurate. The shape of extracted fish is always dilated and some details in the tail are lost during the extraction. Below are some examples of bad fish extraction:

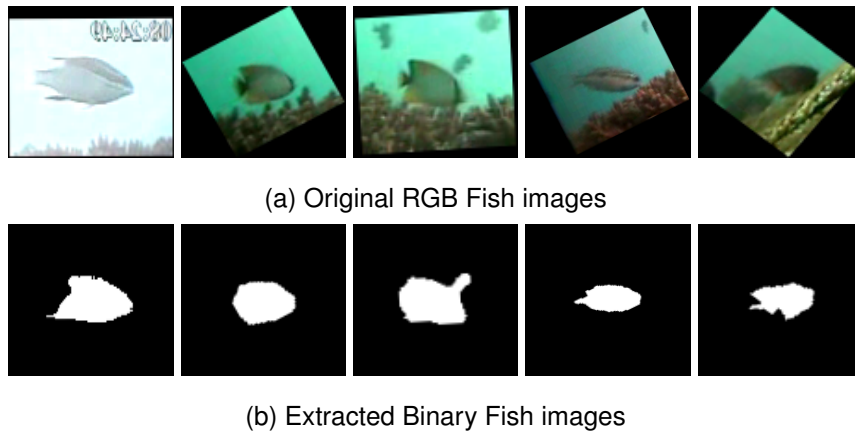


Figure 5.8: Bad Fish Extraction Examples

So when comparing the segmentation result with the ground-truth data, the segmentation result are always bigger in size and less detailed in the tail tips. If these two main factors are considered in the evaluation and their impacts are ignored, the segmentation performance could be improved at least 6%.

5.2 Classification Performance

In order to evaluate the classification performance, a separate dataset is used as the test set for each classification tasks. In both dataset, 12 features are used to represent each tail instance which are selected from the previous sequential forward selection algorithm.

5.2.1 Tail Classes Classification

As introduced in Chapter 4.2, fish tails are categorized into 3 classes based on their shapes. The tail classification task is to distinguish between these different tail classes.

5.2.1.1 Dataset

The fish images in the dataset are unbalanced which is dominated by several fish species. When categorized into different tail classes, the tail classes are also balanced. Type 3 tail is only observed in one fish species among all 15 species. In order not to have a biased classifier which pays little attention to type 3, training set and test set are picked as follows:

	Training Set	Test Set
Type 1	40	100
Type 2	40	180
Type 3	20	20

Table 5.3: Data Sets for Tail Classes Classification

5.2.1.2 Result

After training on the training set of 100 tail images, the Naive Bayes classifier achieves an average classification accuracy of 73.7% on the test set of 300 tail images. Below is the confusion matrix of the classification performance, where the rows indicate the classification results and the columns indicate the actual classes.

	Type 1	Type 2	Type 3
Type 1	42	58	0
Type 2	18	161	1
Type 3	0	2	18

Table 5.4: Confusion Matrix of Tail Classes Classification

When evaluating each tail class separately, the classification performance for each tail class could be concluded:

Tail Class	Classification Accuracy
Type 1	42%
Type 2	89.4%
Type 3	90%

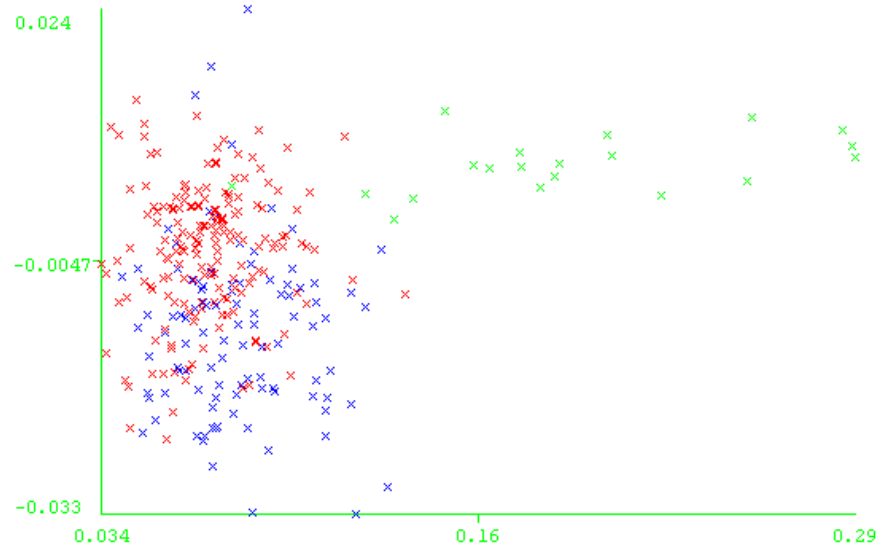
Table 5.5: Classification Accuracy for Each Tail Class

5.2.1.3 Evaluation

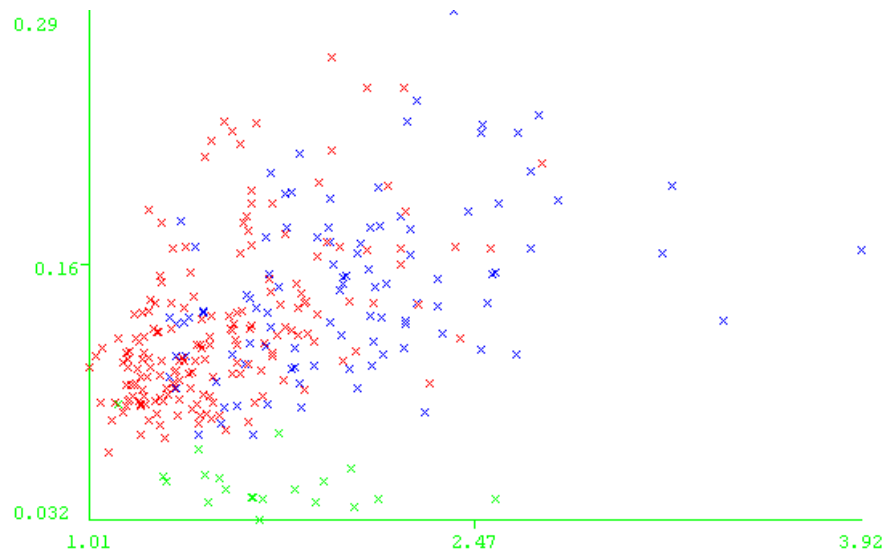
The classification results for tail type 2 and type 3 were very good which were both about 90%. In comparison, the classification performance for type 1 was quite poor which was just about 42%. More than half the tails of type 1 were misclassified as type 2. 18 of 180 tails from type 2 were also misclassified as type 1 and 1 tail was classified as type 3. Neither any instance of type 3 was classifier as type 1, nor the other way

around which indicated that type 1 and type 3 were quite distinguishable. In contrast, 2 tails from type 3 were misclassified as type 3 which took 10% of total 20 instance.

Below are some figures of fish instances in 2-D feature space, the blue dots represent type 1 tails, red indicates type 2 tails and green for type 3:



(a) s02 (X Axis) vs s21 (Y Axis)



(b) compactness (X Axis) vs s20 (Y Axis)

Figure 5.9: Fish Instances in 2-D Feature Space

The above figures shows that type 3 instances (green) are usually linear separable from the other two types. In contrast, type 1 and type 2 tails are quite correlated to each other. The locations of type 1 and type 2 instances look like points drawn from two Gaussian distributions with mean values close to each other, which are quite difficult

to separable from each other.

When looking into details of type 1 fish tails, a very interesting point can be raised:

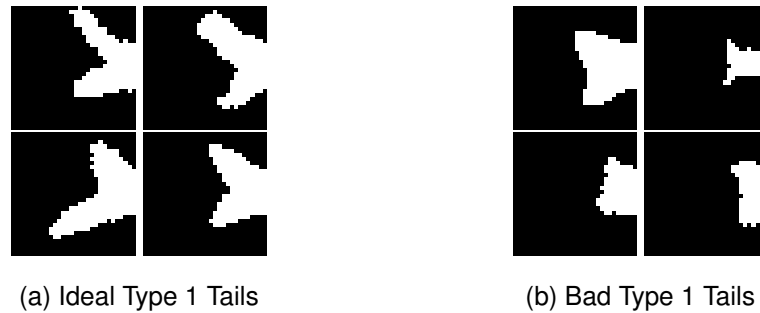


Figure 5.10: Ideal Type 1 Tails vs Bad Type 1 Tails

Again this raise the problem of segmentation and fish extraction. During the previous procedures, details for some of type 1 tails are lost leaving the tails imperfectly segmented. Although this segmentation problem happens to other types, the structure of type 1 tails makes it more easily effected than others. Type 1 tail usually has a forked shape with two tail tips. However, the tail tips can be easily distorted into type 2 shapes. For example, a smooth algorithm, such as Gaussian smooth and dilation, could lose details in tips so that a type 1 tail may look like type 2 tail.

In conclusion, the inseparable problem of type 1 and type 2 tails is caused by the segmentation problem where details of fish tails are lost during the extraction procedure. Many of type 1 tails end up having the shapes of type 2 tails. If the performance of fish extraction was improved, the classification result would have a much better result.

5.2.2 Fish Species Classification

For the fish species classification problem, the task is to explore whether the fish species could be distinguished by their tail shapes. The goal is not to classify fish species using the features of the whole fish, but only the features of tails.

The dataset used for evaluating the performance of fish species classification consist 300 fish instances, each of which contains 12 features selected from the sequential forward selection algorithm. For each fish species, there are 20 instances. The previous dataset used for feature selection is used as the training set, and the separate dataset is used as the test set.

5.2.2.1 Result

The Naive Bayes classifier achieves an average classification accuracy of 30% on the test set. Below is the confusion matrix of the classification performance and the classification result of each species:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Species 1	3	6	1	0	0	1	0	2	0	0	4	0	2	0	1
2	4	8	1	0	0	0	0	0	2	0	3	0	2	0	0
3	2	1	8	5	0	0	0	1	1	0	0	0	2	0	0
4	1	0	5	2	0	3	3	0	1	0	5	0	0	0	0
5	0	0	2	2	7	1	1	2	1	0	1	2	0	0	1
6	0	1	3	3	2	7	1	2	0	0	0	0	0	0	1
7	3	1	1	3	4	3	1	0	2	0	1	1	0	0	0
8	2	2	0	0	0	3	0	5	1	2	3	1	0	0	1
9	3	2	1	1	1	0	0	0	2	2	3	0	1	0	4
10	2	3	1	1	2	0	1	0	2	6	1	0	0	0	1
11	0	3	0	0	0	0	1	0	1	0	13	0	1	0	1
12	2	0	0	3	5	3	1	0	0	2	0	4	0	0	0
13	4	5	0	0	0	0	1	0	1	0	2	0	5	2	0
14	0	0	1	0	0	0	1	0	0	0	1	0	0	17	0
15	0	2	2	3	1	2	1	2	0	0	5	0	0	0	2

Table 5.6: Confusion Matrix of Fish Species Classification

Fish Species	Classification Accuracy	Fish Species	Classification Accuracy
Species 1	15%	Species 2	40%
Species 3	40%	Species 4	10%
Species 5	35%	Species 6	35%
Species 7	5%	Species 8	25%
Species 9	10%	Species 10	30%
Species 11	65%	Species 12	20%
Species 13	25%	Species 14	85%
Species 15	10%		

Table 5.7: Classification Accuracy for Each Tail Class

5.2.2.2 Evaluation

The classification result of fish from species 14 had the best performance of 85% classification accuracy. This was actually not very surprising, because species 14 was the one that categorized into tail type 3 which was quite different from other tails based on the previous observations. Moreover, species 11 also achieved a very good result of 65% accuracy, which indeed was a little bit surprising. Species 1, 4, 7, 9, 15 achieved a very poor performance which is about 10% and the performances of rest species were quite moderate.

The classification accuracy of 30% is not acceptable, because tails of different species have many features in common. It will be strange if the classifier achieves a very good result, for example 70%. The result will indicate that fish are distinguishable by only using tails, which is wrong in the current dataset.

The good classification result of species 11 also indicates that a different tail class may exist in addition to the 3 tail types which are categorized previously. All the results together give a positive response to the exploration of fish species classification using tail features. Some fish species do have unique tail features compared with others.

5.3 Future Work

In general, results obtained by the segmentation algorithm and classification tasks were reasonable. But as discussed previously, most of the negative effects came from the binary masks. The binary masks of extracted fish were not very pleasant. For some species, there were too many details lost in extraction which was the main reason why so many type 1 tails were misclassified as type 2.

A better fish extraction algorithm should be developed which could reserve more information around tail tips and fins. At the moment, binary fish were smoothed and dilated which raised the difficulty of segmenting fish components. If more details can be reserved, both the performance of segmentation and classification will be improved.

Moreover, deformable shape modelling methods, such as Active Shape Model, should be instigated and explored more. So that a more generalizable model could be developed to represent unknown fish.

The dimensionality difference between 3-D real fish and 2-D fish images may be taken into consideration. Depending on the relative position of swimming fish and cameras, fish images are quite different. For example, fish tails are always visible in

fish images which would end up bad tail segmentation results.

5.4 Conclusion

The goal of this project is to explore methods to segment fish components, investigate the possible methods of clustering fish components and develop classifiers that can discriminate different component categories. In this thesis, a tail segmentation algorithm was developed, which used curvatures of boundary points to find the candidate tail joint positions. The algorithm firstly located one tail tip and iteratively searched through the candidate locations to find the other tail tip and then the joints between tail and fish body. A backup tail segmentation was also implemented which tracked the sudden changes in the row length of each column and took the sudden change as the tail segmentation column.

For each segmented tail, it was rescaled, rotated and transferred into the common coordinate system. Principal components analysis was used to reduce the dimensionality from 900 to 10. These 10 principal component features along with other 9 image moment features were combined as the feature set. In order to improve the classification accuracy and use less features, sequential forward selection was implemented to select the best set of features. After all these procedures, 12 features were selected and used to represent tails. An overall tail classes classification accuracy of 73.7% was achieved by using a Naive Bayes classifier.

Bibliography

- [1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*, chapter 12.1. Springer, 2006.
- [2] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *Robotics, IEEE Transactions on*, 20(4):713 – 723, aug. 2004.
- [3] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681 –685, jun 2001.
- [4] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, and Others. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.
- [5] Shelley Derksen and H. J. Keselman. Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, 45(2):265–282, 1992.
- [6] Feng Ge, Song Wang, and Tiecheng Liu. Image-segmentation evaluation from the perspective of salient object extraction. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1146 – 1153, june 2006.
- [7] Igor Kononenko. Estimating attributes: Analysis and extensions of relief. In Francesco Bergadano and Luc De Raedt, editors, *Machine Learning: ECML-94*, volume 784 of *Lecture Notes in Computer Science*, pages 171–182. Springer Berlin / Heidelberg, 1994.
- [8] Rasmus Larsen, Hildur Olafsdottir, and Bjarne Kjaer Ersboll. Shape and texture based classification of fish species. In *Proceedings of the 16th Scandinavian*

- Conference on Image Analysis, SCIA '09*, pages 745–749, Berlin, Heidelberg, 2009. Springer-Verlag.
- [9] Rory McGrath. Learning to recognise fish. Master's thesis, University of Edinburgh, 2011.
- [10] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I, ECCV '02*, pages 128–142, London, UK, 2002. Springer-Verlag.
- [11] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1376 –1381, dec 1998.
- [12] Gayathri Nadarajan, Yun heh Chen-burger, and Robert B. Fisher. A knowledge-based planner for processing unconstrained underwater videos. *IJCAI09 Workshop on Learning Structural Knowledge From Observations (STRUCK09)*, 2009.
- [13] P. Pudil, F.J. Ferri, J. Novovicova, and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. In *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision Image Processing., Proceedings of the 12th IAPR International. Conference on*, volume 2, pages 279 –283 vol.2, oct 1994.
- [14] I Rish. An empirical study of the naive bayes classifier. *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 3(22):4146, 2001.
- [15] Stan Sclaroff and Lifeng Liu. Deformable shape detection and description via model-based region grouping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.
- [16] Concetto Spampinato, Daniela Giordano, Roberto Di Salvo, Yun-Heh Jessica Chen-Burger, Robert Bob Fisher, and Gayathri Nadarajan. Automatic fish classification for underwater species behavior understanding. In *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams, ARTEMIS '10*, pages 45–50, New York, USA, 2010. ACM.
- [17] Harry Zhang. The optimality of naive bayes. In *FLAIRS Conference*. AAAI Press, 2004.