FISH SEGMENTATION AND CLASSIFICATION FOR LARGE SCALE DATASET FROM TURKEY

NUR AMIRAH SHAFIQAH BINTI SALLEH

B.ENG (HONS.) ELECTRICAL ENGINEERING (ELECTRONICS)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : Nur Amirah Shafiqah Binti Salleh

Date of Birth : 17 October 1997

Title : Fish Segmentation and Classification for Large Scale Datatset from Turkey

Academic Session : 2021/2022

I declare that this thesis is classified as:

☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*

☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*

☑ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

amirahshafiqah

(Student's Signature)

971017125136

New IC/Passport Number
Date: 24 June 2022

(Supervisor's Signature)

Dr. Mahfuzah Binti Mustafa

Name of Supervisor
Date: 17 June 2022

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period

Author's Name	Nur Amirah Shafiqah Binti salleh
Thesis Title	Fish Segmentation and Classification for Large Scale Datatset from Turkey
Reasons	(i)
	(ii)
	(iii)

of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Thank you.

Yours faithfully,

____
(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Electrical Engineering (Electronics).

____

(Supervisor's Signature)

Full Name : Dr. Mahfuzah Mustafa

Position : Senior lecturer

Date : 25 June 2022

____

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

amirahshafiqah

____

(Student's Signature)

Full Name : Nur Amirah Shafiqah Binti Salleh

ID Number : EA18024

Date : 24 June 2022

FISH SEGMENTATION AND CLASSIFICATION FOR LARGE SCALE DATASET FROM TURKEY

NUR AMIRAH SHAFIQAH BINTI SALLEH

Thesis submitted in fulfillment of the requirements
for the award of the
B.Eng (Hons.) Electrical Engineering (Electronics)

College of Engineering
UNIVERSITI MALAYSIA PAHANG

JUNE 2022

ACKNOWLEDGEMENTS

First and foremost, Alhamdulilah, I completed my Engineering Project with flying colors. I want to thank God for leading and blessing me through all of challenges I had while conducting the research.

I am very thankful that Universiti Malaysia Pahang (UMP) has given their student the opportunity to take and complete the Engineering project within a year and it will be our experience that can be used in the future.

Next, I would like to express gratitude towards my supervisor Dr. Mahfuzah Binti Mustafa, for guiding me to complete my research from Engineering Project 1 until Engineering Project 2. During the Engineering Project period, ensuring that I was able to comprehend all of the theoretical and simulation knowledge to succeed and achieve the objectives of my research. All the knowledge will undoubtedly be of great value in the future.

Lastly, I wanted to thanks all of my family and friend for their spiritual support throughout completion of this research. Finally, I want to express my gratitude and indebtedness to every individual who has helped me during my research but whose name does not present in this acknowledgement.

ABSTRAK

Pengelasan membantu manusia belajar tentang pelbagai jenis ikan, ciri, persamaan dan perbezaannya. Dalam projek ini, imej daripada lapan jenis ikan dikumpul dari kaunter ikan pasar raya; setiap jenis ikan mempunyai 1000 imej. Kajian ini bertujuan untuk mengekstrak tekstur, warna, dan bentuk ikan serta menggunakan pengelas K-Nearest Neighbors (KNN) dan Mesin Vektor Sokongan (SVM) untuk mengkategorikan lapan jenis ikan di Izmir, Turki. Keputusan daripada eksperimen menunjukkan ketepatan KNN ialah 100% dan SVM ialah 100%.

ABSTRACT

Classification helps humans learn about different kinds of fish, their features, similarities, and differences. In this project, images from eight fish types are collected from a supermarket's fish counter; every kind of fish has 1000 images. This study aims to extract fish's texture, color, and shape and utilize K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) classifiers to categorize the eight different types of fish in Izmir, Turkey. The results from the experiment show the accuracy of KNN is 100% and SVM is 100%.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
KNN	K-Nearest Neighbors
SegNet	Semantic Segmentation Model
RGB	Red Green Blue
HSV	Hue Saturation Value
BoF	Bag of Features
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
CNNsF	Convolutional Neural Networks Features
R-CNN	Region Based Convolutional
ROI	Region og Interest
RPN	Reverse Polish Notation
IoU	Intersection Over Union
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
EHTC	Enhanced Hybrid Task Cascade
HTC	Hybrid Task Cascade
SURF	Speeded Up Robust Feature
MI	Moment Invariants
CSS	Curvature Scale Space
CV	Circular Vector
CP	Curvature Function
FISA	Fish Image Segmentation Algorithm
SIFT	Scale Invariant Feature Transform
FC	Fish Classification
GF	Gabor Filter
DT	Decision Tree
MDC	Minimum Distance Classifier
TADA	Turn Angle Distribution Analysis
BP	Back Propagation
GAILS-BPC	Hybrid Meta-Heuristic
DL	Deep Learning
CLIs	Command-Line Interface
GUIs	Graphical User Interface

CHAPTER 1

INTRODUCTION

1.1 Research Background

Izmir is a metropolitan city in the western extremity of Anatolia, capital of the province of the same name. It is the third most populous city in Turkey, after Istanbul and Ankara. This project presents about a large dataset containing nine distinct seafood widely consumed in the Aegean Region of Turkey is formed. The aim of this project is to classify 8 classes of fish as follows: black sea sprat, gilt-head bream, horse mackerel, red mullet, red sea bream, sea bass, striped red mullet and trout that shown in figure 1.1. Images of eight different seafood types are collected from the fish counter of a supermarket and each type of fish has 1000 images. All fish are fresh and they are positioned in various displacement and angles but lighting conditions do not change significantly. In order to make the dataset useable in studies with real-life situations, a blue and noisy background is preferred instead of a clean white background. Feature extraction is the process of computing and collecting fish properties from texture (Gray-Level-Co-Occurrence Matrix), color and shape. The basic procedure begins with feature extraction, followed by fish classification by using K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) classifiers. It should be able to execute feature extractions and two classifications for eight different species as an expected result.

To increase the efficiency of data storage and processing, feature extraction resolves the challenge of obtaining the most compact and informative set of features (Guyon et al., 2006). Feature extraction is the process of reducing a raw picture to make decision-making easier, such as pattern identification, categorization, or recognition (Elnemr et al., 2015). For classification and regression tasks, defining feature vectors remains the most frequent and convenient method of data representation (Guyon et al., 2006). Finding and extracting accurate and discriminative features is a critical step in completing image recognition and computer vision tasks. A feature is a function of fundamental measurement variables or attributes that identifies a measurable quality of an item and may be used for classification and/or pattern recognition (Guyon et al., 2006).

Support-vector machines are supervised learning models that analyse data for classification and have related learning methods. Because of its mathematical base in statistical learning theory, SVM provides a systematic solution to machine learning challenges. SVM builds a solution using just a portion of the training data. It's used for classification, regression, novelty detection, and feature reduction, among other things (Evgeniou & Pontil, 2001).

When there is little or no prior knowledge about the distribution of the data, K-Nearest Neighbors (KNN) classification is one of the most fundamental and straightforward classification methods. It should be one of the initial options for a classification research. KNN classification was created from the requirement to do discriminant analysis when trustworthy parametric values of probability densities are unknown or impossible to determine. An item is categorised by a majority vote of its neighbours, with the object allocated to the most common class among its k nearest neighbours where k is a positive integer, typically small. If k = 1, the item is simply assigned to that single nearest neighbors class. The function is only approximated locally in KNN classification, and all computation is delayed until the function is evaluated (Peterson, 2009). As a result, the KNN classification process comprises two stages: the first is determining the nearest neighbours, and the second is determining the class based on those neighbours (Cunningham & Delany, 2021).

GLCM presents grayscale frequencies of different pixel intensities appearing in the image. GLCM was used to extract contrast and energy parameters in this study. Although the contrast feature uses the contrast between a pixel and its neighbors, the energy feature calculates the volume of an equal rectangular element. Various color effects are meant to be observed in the classification process using contrast features, which can capture color variations off different fish species. After that, for color extraction use color histogram where each color space RGB is equal to 256 pixels. Then, can convert the pixel data into histogram plot. For shape extraction start by calculating the size of the area, length and width where it calculates for all the classes of fish (Ulucan et al., 2020).

Figure 1.1 From the collected dataset, here the 8 classes of fish images

Source: (Ulucan et al., 2020)

1.2 Problem Statement

Based on the previous research, feature extractions such as texture, GLCM, moments, Beg of Feature (BoF) algorithm, shape, color, and others were used. Some techniques do not accurately extract the data and sometimes fail to detect images with extremely similar texture and color. Then, some features are still incompatible with some images and require the addition of the corresponding functionality.

Next, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Deep Learning, Convolutional Neural Networks (CNNs) and other classification method were used. After that, because the values on the data are different, certain classifications cannot provide high accuracy and require appropriate procedures. Some of the classifiers do not produces the best and most accurate results.

1.3 Research Objective

The objectives of this research are:

1. To extract the texture, color, and shape of fish from Izmir, Turkey.

2. To classify 8 classes of fish from Izmir Turkey market by using K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) classifiers.

1.4 Research Scope

Based on the objectives stated, the following scopes have been established. The large scale fish dataset was collected from a supermarket in Izmir, Turkey where the data set was prepared publicly by Oguzhan, Diclehan &Mehmet based on fish widely used in Izmir, Turkey. This fish data collection has 8 different fish classes, which each have 1000 images. I only took 40 images for each type of fish in my final year project, thus the total amount of images I processed was 320. The simulation and GUI design was done using MATLAB R2021a program. A simulation based on features: texture by using GLCM to measure the contrast and energy values, color by using color histogram that measure Red, Green and Blue values and shape where measure the area, width and height values. Then, perform classification for both SVM and KNN for eight classes of fish.

1.5 Thesis Outline

This study is divided into five (5) chapter. Each chapter will explain future detail based on the research.

In Chapter one (1), will presenting about the introduction which including the research background, problem statement, research objective, research scope of the study and the last one is the explanation of the chapter outline.

In Chapter two (2), literature review will be done to review about previous study related to the topic of this study.

In Chapter three (3), the methodology is explained start with the flow of the process of the simulation then the future detail of the method used during undergo this study.

In Chapter four (4), the result obtained using the method in chapter three will be compared to the result of previous study.

Last but not least, the conclusion and discussion will be the summary of this research later.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This literature review is based on a journal that has been published for at least 5 years. The journal covers pre-processing, segmentation, feature extraction, and classification. The information provided is quite useful in creating this study. This research also looks at features: texture (GLCM), which calculates contrast and energy value, color, by using color histogram where measure the Red Green and Blue pixels and converted to histogram plot, shape, which calculates area width and height. After collecting the features, use SVM and KNN to classify them. This objectives of the research are focuses on extract feature extraction and as well as classification for eight different types of fish.

Feature extraction is a technique for extracting the visual content of images so that they may be indexed and retrieved. Primitive or low-level image attributes can be either generic or domain-specific, such as colors, texture, and shape extraction. Feature extraction is the process of reducing the number of resources needed to correctly describe a huge set of data. One of the primary issues with completing complicated data analysis is the large number of variables involved. When dealing with a high number of variables, it will either need a lot of memory and processing capacity, or it will need a classification method that overfitting the training sample and does not really generalise well to new samples. Feature extraction is a broad phrase that refers to strategies for creating combinations of variables to get past these issues while still accurately representing the data (Mohanaiah et al., 2013).

The process of allocating information classes to spectral classes is just what it means when it talks about classification. Groups of pixels that are consistent with one another in terms of the brightness values that they exhibit throughout all of the data's spectral channels make up what are known as spectral classes. Information classes are several types of categories of interest that an analyst looks for in an image by using his prior knowledge and expertise of the region. It is essential to keep in mind that there may be numerous spectral classes included inside one information class. This may be the case based on the characteristics of the features that the image depicts or the reasons for classifying the data. In other words, a variety of spectral classes could be categorised together as part of the same information class. In a nutshell, image classification may be described as the process of all of the pixels in an image being assigned to certain categories or subjects (Anand, 2018).

2.2 Segmentation

(Ulucan et al., 2020) presented the ground-truth segmentation masks that shown in figure 2.1 where all seafood images are manually removed by a human operator in the initial round of studies. The masks are then smoothed using morphological operators to provide finer shaped binary labels. An erosion operation with the shape of a diamond and a distance of 8 is used throughout this process. To obtain final ground-truth masks, a sphere shaped dilation operator with a distance of 25 is applied to all masks. After manually acquiring ground-truth masks, the fish will be automatically isolated from its context using a semantic segmentation method called SegNet shown in table 2.1, which is a ten-layer neural network. The data is split into two parts: a 70% training set and a 30% test set. The filter size and number of filters for the two classes (in this example, fish and backdrop) were found to be 3x3 and 64, respectively. The mini-batch size is set to 8 and the maximum epoch number is set at 10. The backdrop appears to get a much higher number of pixels than the fish. As a result, inverse frequency weighting is used to determine the weights of the classes, and the network's final layer is updated. SegNet ended up with average training and test accuracy rates of 98:01% and 88:69%, respectively. On the enhanced dataset, several examples of ground-truth masks.

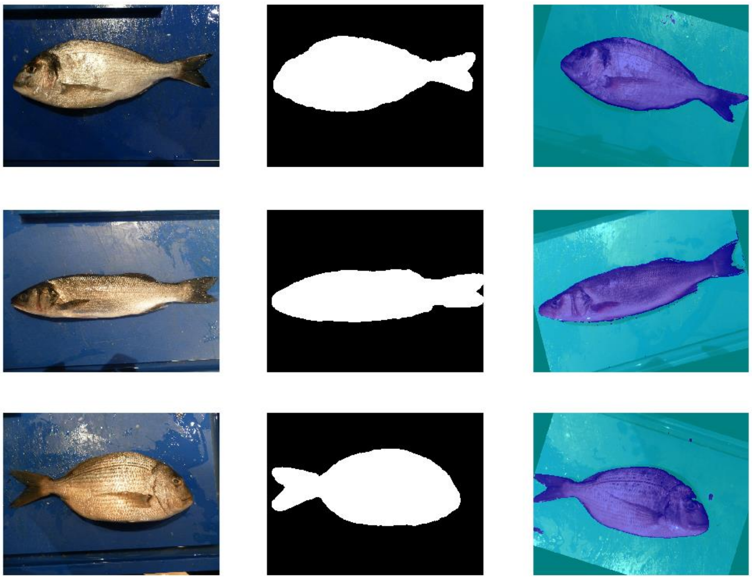


Figure 2.1 From left to right, fish picture samples, ground-truth segmentation masks, and SegNet segmentation results.

Source: (Ulucan et al., 2020)

Table 2.1 SegNet segmentation results in percentages

Test	*Train Accuracy*	*Test Accuracy*
Gilt Head Bream	*97.12*	*96.82*
Red Sea Bream	*98.21*	*91.84*
Sea Bass	*95.69*	*82.98*
Red Mullet	*99.02*	*90.32*
Horse Mackerel	*98.90*	*86.97*
Black Sea Sprat	*97.52*	*89.66*
Stripped Red Mullet	*99.24*	*89.60*
Trout	*97.32*	*80.45*
Shrimp	*99.06*	*89.59*
Average	*98.01*	*88.69*

Source: (Ulucan et al., 2020)

(Yu et al., 2020) presented the Mask R-CNN algorithm has the following process shown in figure2.2. To begin, the images are sent into a trained ResNext101, which produces a feature map. ResNext101 is a convolutional neural network that may enhance accuracy by reducing the number of hyper-parameters while keeping the parameters simple. Second, for each point in the feature map, a fixed ROI is created, which often sets a number of various fixed ratios to support numerous form recognitions. Then, in a binary classification and Bounding-Box regression, several ROIs are supplied into RPN. To some extent, binary classification is utilised to separate the background from the target item. In approaching the true contour of the object, the Bounding-Box regression serves a regulatory function. These two steps delete some unmatched ROIs. Finally, ROI align operation on the remaining ROIs is performed.

Mask R-CNN is used to segment and quantify the morphological aspects of fish. Figure below shows depicts the overall flow chart. The fish body image is first obtained by the image acquisition equipment, then pre-processed. The contrast adjustment technique is employed at the pre-processing stage. In the studies, we used 100 photos to do contrast augmentation and 100 images to perform contrast reduction (Yu et al., 2020).

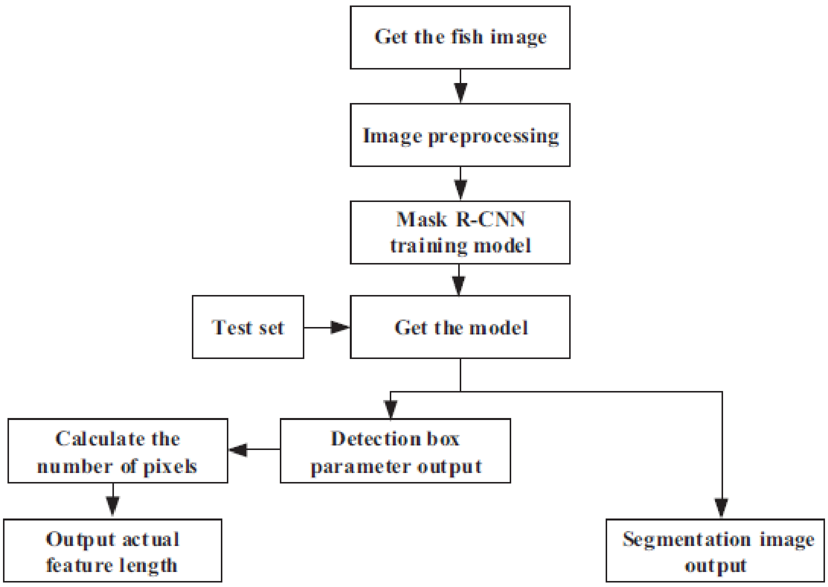


Figure 2.2 The block diagram of the experimental setup
Source: (Yu et al., 2020)

The pre-processing effect is shown in figure 2.3. The data set is extended as a result of these actions, and the impact of light and shadow on detection is decreased, improving the trained model's resilience. The data set is then annotated in order to create a training set, which is then fed into the Mask RCNN network to train the learned model. The test set is then sent into the trained model, which produces a segmented image and the target frame's relevant properties. Finally, by computing the number of pixels in the detection frame and translating them to the real fish feature parameters, the final result is achieved (Yu et al., 2020).

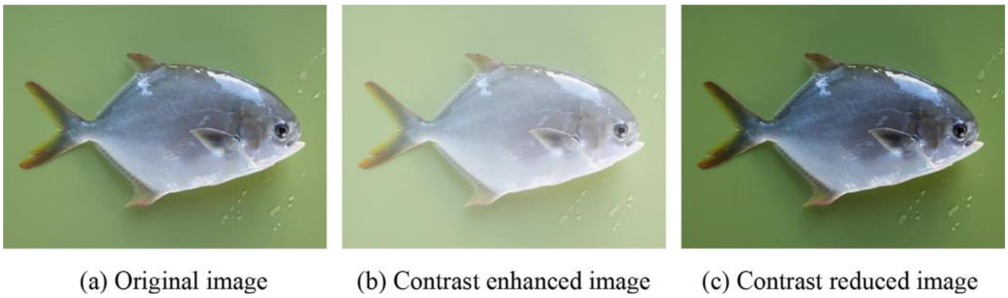


Figure 2.3 Original image and contrast changed images
Source: (Yu et al., 2020)

(Garcia et al., 2020) presented a subsea system and a topside system make up the Deep Vision system. A stereo camera, strobe lighting, battery, and an enclosing studio frame are all included in the subsea system for best image quality and consistency. The processing pipeline depicted is used to analyse all of the captured images. First, nonlinearities and non-uniform illumination effects are corrected in the images. After that, we utilise a Mask R-CNN architecture to segment and locate each individual fish in the image. The acquired segmentation is then refined in the next phase by estimating the boundary of each fish using the local gradient. Finally, using stereo information, the fish's length is calculated.

Figure 2.4 show, Deep Vision is a subsea imaging system. The device, which includes a stereovision camera set and an indirect illumination source, is housed inside a trawl net (a). The arrows in the centre image (b) clearly define the studio section, which corresponds to the region where the catch flows and is depicted in greater detail in the bottom schematic (c). Fish pass through a trapezoidal plexiglass portion that keeps them at least 20 cm away from the cameras and lights while being inside the cameras' range of view (Garcia et al., 2020).

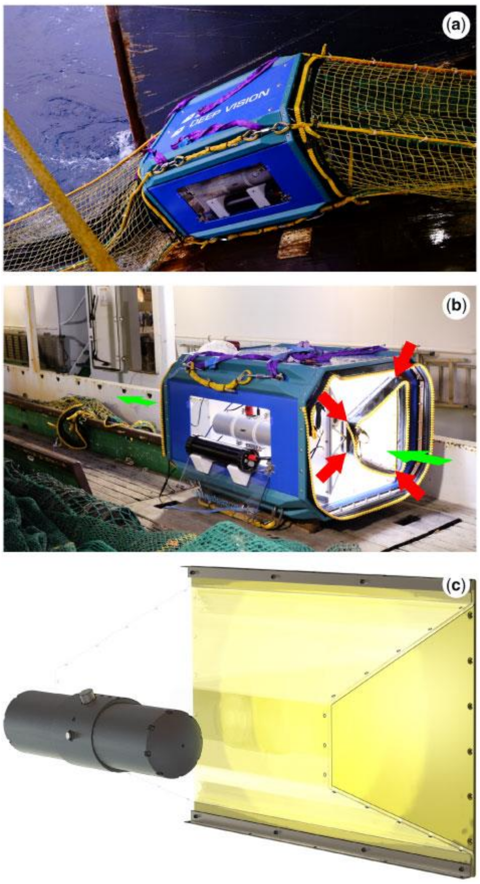


Figure 2.4 Deep Vision subsea system
Source: (Garcia et al., 2020)

Figure 2.5 shows, pipeline for measuring fish automatically. The image is first pre-processed, after which a CNN is used to locate each individual fish. For the fish, the CNN also offers a segmentation mask. The length of the specimen is next determined using stereo cues, after which these masks are improved using local contrast information to outline the boundary of each fish (Garcia et al., 2020) .

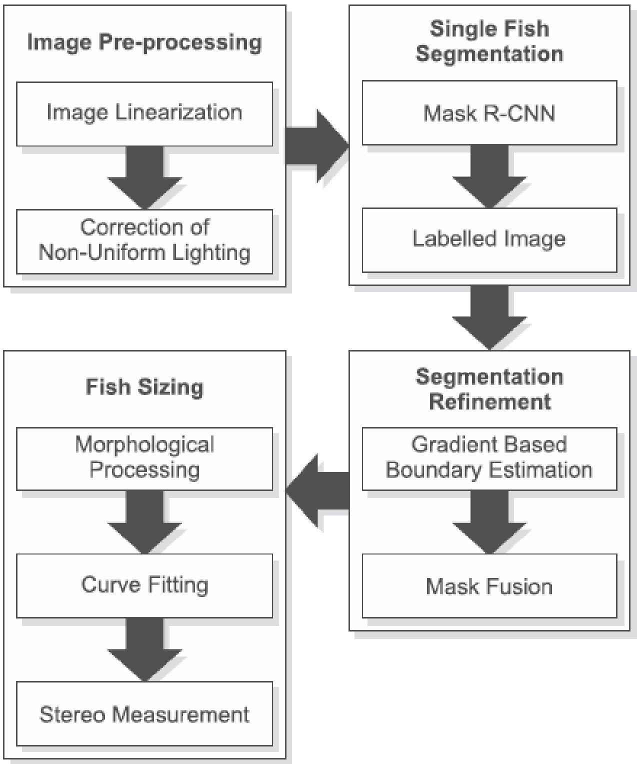


Figure 2.5 Automatic fish measurement pipeline
Source: (Garcia et al., 2020)

Segmentation performance is a detection accuracy measurement is required to evaluate the performance of the masks generated by our processing pipeline. Because intersection over union (IoU) and pixel accuracy are the de facto assessment metrics employed in object detection, they are utilised to quantify the segmentation outcomes. IoU, commonly known as the Jaccard index, is a statistic for determining the accuracy of object segmentation on a given dataset. The bounding box predicted by the CNN detector and the ground-truth (i.e. manually marked) bounding box are frequently used to calculate IoU. In our situation, IoU is computed using these two regions since our detector creates a pixel region (mask) containing the pixels that belong to a certain fish, and the ground-truth is also a hand-labelled pixel region (Garcia et al., 2020). The final score is calculated by dividing the amount of overlap between the expected and ground-truth regions by the area of union between both the predicted and ground-truth region and it is calculated by Eqn (2.1).

$$\text{IoU} = \frac{\text{ground-truth} \cap \text{prediction}}{\text{ground-truth} \cup \text{prediction}}$$

2.1

Segmentation refinement is a pixel accuracy, on the other hand, refers to the proportion of pixels in an image that were properly categorised. It is usually shown for each class, with the average of all courses supplied. Because we only have the "fish" class, both values are the same in our situation. The concepts of TP, TN, FP, and FN must be introduced for this measure. True positive (TP) pixels are accurately predicted to belong to the specified class, whereas true negative (TN) pixels are correctly detected as not belonging to the specified class (Garcia et al., 2020). False positives (FP) and false negatives (FN) are terms used to describe these situations. After that, the accuracy metric is calculated by Eqn (2.2).

$$\text{Accuracy} = \frac{\Sigma\, TP + TN}{\Sigma\, TP + TN + FP + FN}$$

2.2

Figure 2.6 shows, segmentation of fish where it can be separated into distinct specimens in simple examples like (a) by subtracting the background (b). However, in order to segment the three fish instances in (c), require a cognitive grasp of the visual (d).

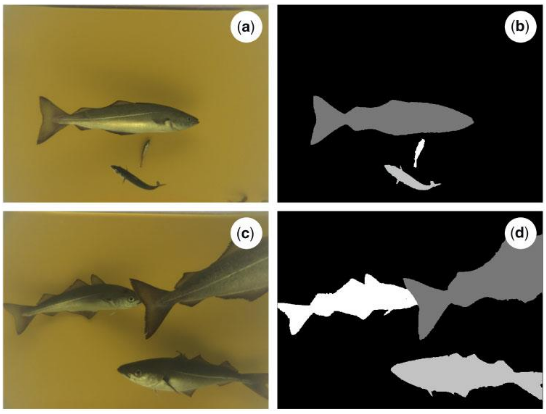


Figure 2.6 Fish segmentation

Source: (Garcia et al., 2020)

(Zhang et al., 2021) presented the fish datasets, such as Fish4Knowledge, use screenshots from films rather than raw images, therefore the images are tiny and blurry, making them unsuitable for deep learning models to analyse. The supervised super-resolution approach cannot be utilised to train the data since there is no high-resolution ground truth. As a result, in the enhanced hybrid task cascade (EHTC), we use the unsupervised deep image prior. The whole network in our pre-processing system has an encoder-decoder structure, as indicated in the diagram below. The model can learn feature information of various sizes thanks to multi-level down-sampling and up-sampling processes. The network extracts the fundamental components that make up the image in the encoder first, such as points, lines, and edge contours. Higher-level and abstract information, such as the structures and layouts of complete items, may be recovered as the network grows deeper. We restore the learnt characteristics to an image with the requisite size in the decoder stage by stage. The skip operation cascades the high-level features and low-level features obtained by the up-sampling process. As a consequence, the feature maps contain not just high-level features but also a large number of low-level features, allowing for the fusion of features at various scales and improving the model's accuracy. Our pre-processing technique differs from previous CNNs in that it does not separate the training and testing procedures, as Algorithm 1 describes.

Most deep learning models for segmentation, for example, are now built on the Mask R-CNN, which adds a mask branch to outline objects in images. They also employ the RoIAlign feature size operation to decrease the inaccuracy caused by resizing region proposals. The RoIAlign uses interpolation to enlarge the feature size, unlike prior feature size procedures (e.g., RoIPool). The Mask R-CNN was then expanded to create Mask R-CNN, which uses the IoU to indicate the mask's quality. To acquire a perspective of the global image, used the cascade architecture to the Mask R-CNN and added one global semantic segmentation branch. In this research, we employ the hybrid task cascade (HTC) to conduct fish instance segmentation and apply the mask scoring (Zhang et al., 2021).

2.3 Feature Extraction

(Ulucan et al., 2020) presented that all seafood varieties were categorised using the retrieved characteristics from segmented images after automated segmentation. The following are four distinct feature extraction approaches that are used to supply information to the classifier shown in figure2.7. In an image, the GLCM shows how frequently different grayscale intensities of pixels occur. GLCM is used to extract contrast and energy characteristics in this investigation. The energy feature computes the sum of squared components, whereas the contrast feature utilises the contrast between each pixel and its neighbour. The contrast feature can capture colour changes in various fish species, therefore the effect of diverse hues is intended to be seen throughout the categorization process. On the other hand, the energy characteristic was picked due to its shown effectiveness in food quality evaluation assignments. Both of these features are supplied to SVMs independently and as a single concatenated feature.

Next, to observe the distributions in the data and their effects on the classification success rates, the second order moment (variance) and the third order moment (skewness) are extracted. The concatenation of the second and third order moments is tested to see if their combination improves test accuracy rates in the classification process, similar to GLCM features (Ulucan et al., 2020).

After that, the Bag of Words method is used in the BoF algorithm for image processing. To develop a visual vocabulary, this feature extraction methodology used the SURF method to extract discrete pixel-based information. To achieve the final version of the visual vocabulary, all characteristics are extracted and the weak ones are deleted using K-means clustering. BoF is used in this study since it takes into account a large variety of characteristics (Ulucan et al., 2020).

Finally, CNNs are image processing systems that are inspired by the human visual system and the biological structure of the brain. An input layer, a convolutional layer, a pooling layer, and a fully connected layer are the four major layers of a CNNs. The number of layers and their sequence are determined by the difficulty of the challenge at hand.

A bigger number of layers is frequently required for more difficult tasks. CNNsF is used to extract numerous characteristics such as edges, blobs, and microscopic detail in this work (Ulucan et al., 2020). AlexNet is used as a pre-trained network since training an end-to-end network involves a lot of computing effort. The epoch number is set to 10 and the size of the micro batch is set to 8.

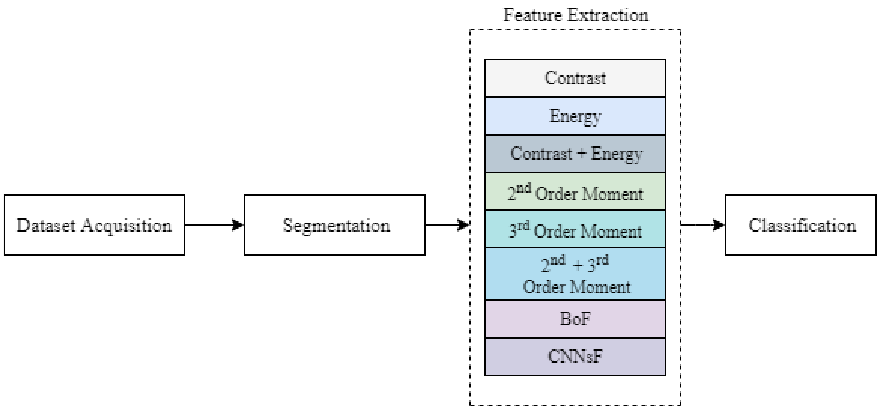


Figure 2.7 The block diagram of the experimental setup
Source: (Ulucan et al., 2020)

(Alsmadi & Almarashdeh, 2020) presented the approaches based on shape features, for feature extraction, Moment-Invariants (MI) are paired with geometrical considerations, which are texture descriptors. Apart from being two-dimensional and placed inside the camera's view area, the approach was indifferent to fish size. CSS (Curvature Scale Space) is another shape descriptor that may be used to extract features. In general, the target object foreground section is turned into a Curvature Scale Space map in CSS. The CSS map is then converted to a Circular Vector (CV) map, which allows for rotation and invariant matching. To extract the form data, Curvature Function (CF) analysis was utilised to determine significant landmark spots on the fish contour and extract the shape features. The FishID method is a real-time contour matching approach that is especially designed for applications involving fish species categorization. Using size and form criteria, the approach properly classifies the species with higher than 90% accuracy. One of the most common approaches for extracting size and form data is to use landmarks/anchor points. The distance and angle measurements from the identified Landmarks/anchor points are used to derive fish size and form attributes.

The following are the local and global feature-based methods: Because various fish species have distinct textures, colour signatures, and form features in their body sections, local features can produce better results for fish picture segmentation than global features. The Fish Image Segmentation Algorithm (FISA) is used to divide a fish into three body segments (head, abdomen, and tail). The Scale Invariant Feature Transform (SIFT), which are descriptors used for identifying and recognizing, tracking, and extracting areas of interest of objects, was partially inspired by the Speeded Up Robust Features (SURF). Local feature descriptors such as pixel gradients and key point localisation are extracted using the SIFT and SURF methods (Alsmadi & Almarashdeh, 2020).

The following are the colour feature-based methods: Another descriptor that is utilised to extract characteristics for FC is color Space. The red, green, and blue components of the RGB colour space are utilised to extract feature sets for both colour systems (RGB and HSV), whereas the hue, saturation, and value components of the HSV colour space are used independently for FC. When the background and foreground colour distributions are not well separated, the Grabcut technique can be used to extract the foreground. The Grabcut method is used to recognise and segment fish from real photos in RGB colour space under a variety of conditions (Alsmadi & Almarashdeh, 2020).

The following are the texture feature-based extraction techniques: The gray-level co-occurrence matrix (GLCM) is a texture descriptor that considers spatial relationships of pixels and characterises the texture of a fish image by calculating the frequency of occurrence of specific values and specific spatial relationships of pixel pairs in an image, creating a GLCM, and then extracting statistical measures from the GLCM. GLCM is used to extract colour signatures from the ventral portion of the fish object utilising RGB colour space and colour histogram approach. After that, three statistical characteristics were calculated: standard deviation, homogeneity, and energy. Another study employed GLCM to extract colour texture from a colourful fish image that was divide into 4x4 blocks (Alsmadi & Almarashdeh, 2020).

Then, using GLCM, six statistical characteristics were determined for each block: average, dissimilarity, standard deviation, homogeneity, contrast, and energy. Gabor Filter (GF) is a texture descriptor for feature extraction and edge detection that uses orientation representations and frequency. Using the GF output image, four characteristics (contrast, standard deviation, mean, and homogeneity) were determined for FC. A sinusoidal plane with a certain direction and frequency modulated by a Gaussian envelope is known as GF (Alsmadi & Almarashdeh, 2020).

Combination-based feature extraction algorithms analyse a sequence of texture, colour, and form changes in a fish image and combine them as a fundamental step in discriminating between fish species and improving classification accuracy. The texture and form were weighted using a combined principal component analysis based on the generalised variances of the two types of variation, and the weights were then utilised to distinguish between the three fish species. For FC, a huge collection of combined collected form measurements (19 features), size measurements (4 features), texture measurements (16 features), and colors signature (8 features) from several fish species were retrieved (47 features). FC's performance was greatly enhanced as a result of these integrated characteristics(Alsmadi & Almarashdeh, 2020). To classify the fish images into dangerous and non-dangerous families, a general FC was performed using a combination of significant extracted anchor points (using distance and angle measurements), texture, and statistical measurements, with the dangerous fish families being divided into predatory and poison fish families, and the non-dangerous fish families being divided into garden and food fish families. To extract feature maps from noisy fish images, the deep Convolutional Neural Network (CNN) approach is applied.

2.4 Classification

(Ulucan et al., 2020) presented the eight different SVMs are trained for the experiments. Figure 2.8 and Figure 2.9 detail the training and testing outcomes. The lowest average training and test accuracy rates are obtained by using BoF (84:36% and 81:55%, respectively). Actually, when the images have extremely similar colors and textures, the BoF algorithm selects the strongest characteristics among others, and it occasionally fails to choose the appropriate features for the task. On the other hand, when the GLCM contrast feature is used in the SVMs classifier, the best training and test results are obtained. In the training procedure, a mean accuracy of 98:74 % is reached, which is over 1% greater than the result obtained using CNN-based features.

The gap between the contrast feature and CNNsF for average test accuracy climbed to more than 4%. Given that CNNs are primarily intended for effective vision-based applications, it can be concluded that using the GLCM contrast feature for this dataset is a smart decision. As seen in this study, the concatenation of various attributes can have varying impacts on accuracy for different data. For example, while the combination of contrast and energy characteristics improves red sea bream training and test accuracy, it reduces the system's performance for black sea sprat (Ulucan et al., 2020). When the energy feature is used alone, it gives a test accuracy of 97:15 % for trout; but, when paired with the contrast feature, it produces a success rate of 97:28 %. Another issue worth mentioning is that utilising the contrast and energy characteristics separately leads to greater accuracy on average than combining them.

The second order moment leads to improved accuracy rates on average, according to the classification findings derived by statistical characteristics. It delivers greater accuracy rates than the GLCM contrast feature in training for horse mackerel and trout. Another finding is that combining second and third order moments yields worse average accuracy than utilising these characteristics separately. Finally, using CNNsF in the SVMs algorithm results in much greater accuracy rates than using BoF, and it closely matches the training success rates of the GLCM contrast and energy features. Low and high level features generated using CNNsF also produce the highest training and test accuracy rates in shrimp (Ulucan et al., 2020).

	Contrast	Energy	Contrast+Energy	2^{nd} Order Moment	3^{rd} Order Moment	2^{nd}+ 3^{rd} Order Moment	BoF	CNNsF
Gilt Head Bream	99.56	98.44	98.09	88.93	91.85	90.52	86.06	96.31
Red Sea Bream	98.85	99.33	99.39	98.22	89.00	89.89	66.61	97.86
Sea Bass	97.30	98.07	96.74	95.59	95.22	90.09	78.75	97.31
Red Mullet	99.56	99.19	98.81	89.78	96.67	91.52	92.14	97.62
Horse Mackerel	98.81	99.22	97.93	98.96	95.96	92.09	83.04	98.50
Black Sea Sprat	99.00	97.81	96.85	97.89	95.37	91.48	95.18	98.45
Striped Red Mullet	98.48	98.89	98.17	91.67	88.93	88.89	67.11	95.71
Trout	99.48	97.85	97.98	99.74	94.04	93.50	97.39	98.75
Shrimp	97.63	98.22	97.37	93.59	93.70	92.02	92.93	98.74
Average	98.74	98.56	97.96	94.93	93.42	91.11	84.36	97.81

Figure 2.8 Feature-based classification results for training of SVMs in percentages

Source: (Ulucan et al., 2020)

	Contrast	Energy	Contrast+Energy	2^{nd} Order Moment	3^{rd} Order Moment	2^{nd}+ 3^{rd} Order Moment	BoF	CNNsF
Gilt Head Bream	98.00	97.26	96.41	87.93	90.11	89.04	81.67	93.07
Red Sea Bream	97.41	98.67	98.81	95.04	86.15	88.48	64.58	96.17
Sea Bass	95.56	96.37	95.24	91.93	91.30	89.19	75.00	92.08
Red Mullet	98.89	98.44	98.06	86.85	92.93	90.46	87.25	89.90
Horse Mackerel	97.41	98.15	96.74	96.89	92.85	90.81	80.33	92.74
Black Sea Sprat	97.85	96.41	95.83	89.37	92.37	90.70	93.33	96.07
Striped Red Mullet	97.33	97.85	97.11	87.67	86.93	86.72	65.00	86.66
Trout	99.19	97.15	97.28	98.15	91.78	91.07	95.83	95.00
Shrimp	97.15	96.96	95.57	89.00	90.52	89.67	90.95	97.56
Average	97.64	97.47	96.78	91.43	90.55	89.57	81.55	93.25

Figure 2.9 Feature-based classification results for testing of SVMs in percentages

Source: (Ulucan et al., 2020)

(Alsmadi & Almarashdeh, 2020) presented with great results, the Support Vector Machine (SVMs) is used for regression and classification of high-dimensional data sets. Based on the amount of features retrieved from the fish image dataset, one of the classification algorithms utilised for FC was Support Vector Machine (SVM). SVM is also used to improve fish species classification by removing the constraints of existing approaches such as K-mean Clustering, K-Nearest Neighbor (KNN), and Neural Network. Multiclass SVM is also utilised for fish species classification. For fish species classification, SVM and decision trees (DT) were utilised; the fish species were categorised depending on their class or order. For accurate underwater live fish recognition, a linear SVM classifier is utilised. Other work used three types of classifiers which are SVM, KNN, and Decision Tree for FC.

The Minimum Distance Classifier (MDC) is an FC distance-based classifier that employs distance estimates between database feature vectors and the test fish feature vector. TADA (Turn Angle Distribution Analysis) is a matching approach that compares the current fish image's contour to species-specific contours in the FishID database. For the FC challenge, the retrieved fish characteristics were input into a well-designed 3-layer neural network classifier that was trained using the Back Propagation (BP) technique. The fish images are classified into families and species using a hybrid meta-heuristic method called GAILS-BPC, which combines a genetic algorithm with iterated local search and back-propagation (Alsmadi & Almarashdeh, 2020).

Deep CNN is based on an untrained VGG-16 network with five blocks of 13 convolution layers and three fully connected layers. CNN is used for FC with noisy fish images. A pooling layer and a fully connected layer are the two convolutional layers that make up CNN (Alsmadi & Almarashdeh, 2020). It is used to identify fish species based on their images. In another study, a CNN was utilised in conjunction with fish species classification. It was trained on features extracted from fish images by the CNN in supervised deep learning (DL), and it was based on conventional classifiers like SVM and KNN. Another study employed two CNN architectures for FC: scaled-down VGG-16 and conventional VGG-16.

2.5 Summary of Literature Review

A large-scale dataset for fish segmentation and classification taken from a publication where the focus was on the detection and classification of fish diseases. It is possible to construct a large and effective dataset that includes nine various types of seafood that are often eaten in the Aegean Region of Turkey. During the process of taking pictures, fresh fish are utilised, and each one is rearranged into a unique formation before being photographed from a number of different perspectives. Convolutional neural networks, GLCM, moments, and BoF algorithm were the four feature extraction techniques that were utilised (CNNs). Select SVM as the appropriate classifier. The requirement for a dataset that is accessible to the public has been fulfilled, and its utilisation in further research will be beneficial.

Following this, a survey on the performance of fish classification techniques is compared, with the emphasis being placed on the availability of pre-processing and feature extraction methods, the number of extracted features and classification accuracy, as well as the number of fish families and species recognised. Utilized the colour feature, the texture feature, the form feature, the local and global feature, and the combination-based feature extraction methods.

After that comes the third journal, which is titled "Segmentation and Measurement Scheme for Fish Morphological Features Based on Mask R-CNN. In this article, we offer a method for segmenting fish images and assessing fish morphological feature indicators that is based on the R-CNN masking algorithm. In intelligent mariculture, the morphological characteristics of fish, such as the length of the body, the width of the body, the length of the caudal peduncle, the width of the caudal peduncle, the pupil diameter, and the eye diameter, are indicators that play a very essential role.

Last but not least, of our topics is the automated segmentation of fish through the use of deep learning, with applications to the determination of fish size. The processing of stereo images that were obtained by the deep vision imaging device, which was immediately installed in the trawl, is the foundation of the concept. In order to compensate for any nonlinearities in the camera response, the images go through a pre-processing step. After that, a Mask R-CNN architecture is utilised so that each individual fish may be localised and segmented inside the images. Capability to successfully cope with images that are crowded and feature fish that overlap one other.

The final journal is called fish image instance segmentation, and it contains the following sections: A n Improved Hybrid Task Cascade Methodology. The innovative CNN model known as the Hybrid Task Cascade (HTC) uses cascade architecture to obtain improved performance in the instance segmentation task. This is accomplished through the application of the cascade technique. To get around these restrictions, it should make use of an Enhanced Hybrid Task Cascade (EHTC) model. EHTC assists in resizing images and optimising features so that they are able to be comprehended more readily by the instance segmentation network that comes later. The purpose of MaskIoU is to create mask confidence ratings in order to provide the mask that results in improved instance segmentation accuracy.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will provide a comprehensive overview of the approaches used to successfully execute and manage this project. It is purpose to guarantee that the project's development is proceeding smoothly and that desired goals are attained. It also guarantees that the project adheres to the goal oriented guideline. The simulation was conducted using MATLAB in this project.

The objective of this study is to divide fish into eight distinct categories. Each kind of fish includes a total of one thousand images, and there are images of eight different kinds of seafood that were taken from the fish counter of a supermarket. The process of calculating and compiling fish features based on their texture, colour, texture, shape and overall form is referred to as feature extraction. The first step of the fundamental technique is the extraction of fish features, which is then followed by the classification of fish using the Support Vector Machine (SVM) and the K-Nearest Neighbors (KNN) classifiers. As a result, it is expected that it will be capable of performing three feature extractions and two classifications for a total of eight distinct species of fish.

3.2 Overview of methodology

The data set was prepared publicly by Oguzhan, Diclehan & Mehmet based on fish widely used in Izmir, Turkey. This fish data collection has 8 different fish classes, which each have 1000 images. For each classes, 40 images used and the total images are 320. After data collection, next step is pre-processing.

Following the collection of the data, go on to the pre-processing step. The image should first be cropped and rotated. This is due to the fact that each image has its own date and time tag, which prevents it from being affected in the primary image. After that, proceed to do segmentation based on color. This segmentation acts in detecting the colour space of the Lab and makes the final result as a black background and the image of the fish in color. This segmentation is separated up into three different options: the first option uses color-based segmentation determined by k-means clustering; the second and third segmentations both use Image Segmenter apps that detect using colour space. So, the user has the option of selecting Seg1, Seg2, or Seg3 as the optimal outcome for segmentation at this point in the process. After finished with the process of segmentation, continue to extract feature extraction.

There are three different kinds of extraction features, and the first one is used in the process of extracting the texture. This method employs the GLCM, which calculates the value of contrast parameters and energy. The second order of GLCM includes both contrast and energy. Graycoprops are used to quantify the value of statistical stats when the value of the parameter is being measured, and graycomatrix is used to measure the offset for four different angles: 0 degrees, 40 degrees, 95 degrees, and 135 degrees.

Histograms of colour are flexible constructs that may be constructed from images stored in a wide variety of colour spaces, including RGB and any other colour space that has any dimension. Simply count the number of pixels in each of the three RGB channels for each of the 256 different scales. The histogram of an image begins with the discretization of the image's colours into a certain number of bins, followed by the counting of the number of image pixels that fall into each bin (Wikipedia, 2010).

For instance, a Red-Blue chromaticity histogram may be created by first normalising colour pixel values by dividing RGB values by R+G+B, and then quantizing the normalised R and B coordinates into N bins each. This creates a histogram that displays the relationship between red and blue colours divided into four bins (N=4) might yield a histogram. Following that, the very last feature is the shape, which is where the value of the area, width, and height for each variety of fish is measured (Wikipedia, 2010).

Last but not least, is to continue classifying all 773 of the data obtained through feature extraction. The initial classification is accomplished by the application of the KNN classifier then followed by SVM classification where use the application of the learner classifier that is accessible through MATLAB applications. First the data should be entered into the workspace, and then the data should be imported from the workspace. Select all of the classifications, the results obtained will include the % value for each type of KNN and SVM classification. Therefore, the results showed that the Fine-KNN classifier achieves a perfect score 100%, and fundamentally stated, this classifier is simpler than the others. Because of this, this certain type classifier is used. Because they are quantitative predictors, the distances that are used are Euclidean. Next, the result showed that the Linear Support Vector Machine (SVM) achieved a perfect score 100% and the fundamental findings shows that linear is an appropriate classifier since all of the data associated with this project are equivalent. Therefore, the kernel is linear as well, resulting in a linear classifier being obtained. Use of a kernel for determining the border of a large number of data sets.

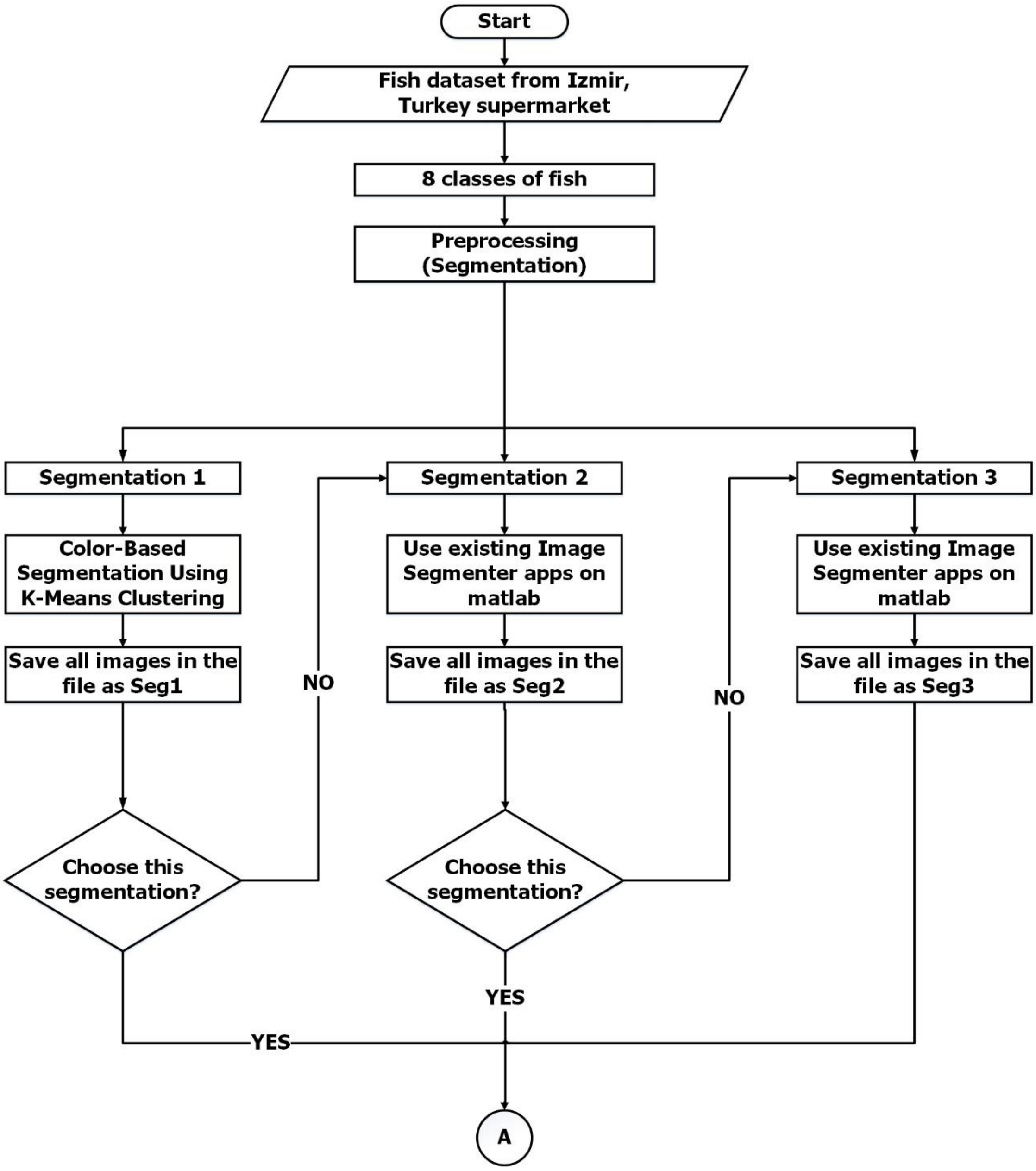


Figure 3.1 Flowchart of overall process

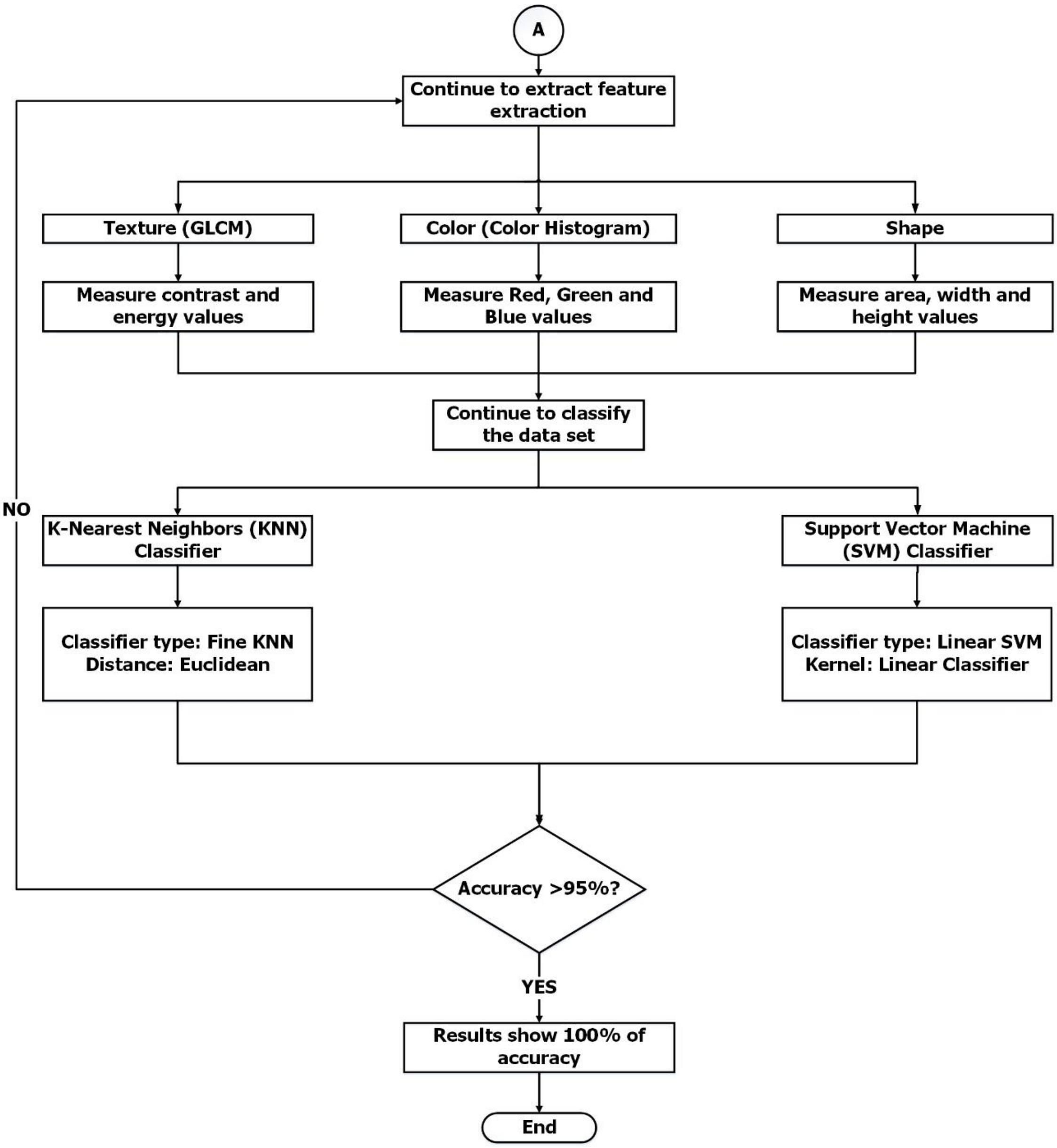


Figure 3.2 Continue of flowchart for overall process

3.3 Segmentation

In digital image processing and analysis, image segmentation is a technique that is frequently used to partition an image into various portions or areas. This partitioning is typically done on the basis of the properties of the pixels included within the image. The process of segmenting an image may include separating the foreground from the background, or it may require grouping together sections of pixels based on similarities in colour or shape.

The process of clustering is a method for dividing into groupings of objects. When K-means clustering is performed, each item is considered to have a position in space. It locates partitions in such a way that items within each cluster are situated as closely as feasible to one another and as far as possible from objects located in other clusters. Within a given colour space, it is able to segment image pixels according to their value into clusters with the use of the image segmentation k-means function. In order to demonstrate how employing a variety of colour spaces may lead to more accurate segmentation results, this example conducts k-means clustering on the same picture in both the RGB and L*a*b* colour spaces.

The first thing that needs to be done for color-based segmentation using k-means clustering is to read the image. After that, the colours need to be classified in the RBG colour space using k-means clustering, and then the image needs to be divided into three regions using k-means clustering in the RGB colour space. The image segmentation k-means function will return the label that corresponds to the cluster for each and every pixel that is included in the input image. The image showed the label overlay on top of the original image. The sections are white, light blue-purple, and light pink are improperly grouped together in the images. The red, green, and blue channels of the RGB colour space combine information about brightness and colour, the brighter version of two different colours is more difficult to divide than the darker version of the same two colours. This is due to the fact that the RGB colour space combines information about brightness and colour in each channel.

The picture should then be converted from its current colour space of RGB to one of L*a*b*, which distinguishes between the image's brightness and colour. Because of this, it is much simpler to section off the region according to colour, regardless of intensity. Additionally, the colour space is more compatible with the way human eyes see the distinct white, blue-purple, and pink areas present in the image. After that, make a picture that separates the H&E image according to colour, then complete the morphology, and lastly, save all of the photos in the save file under the name Seg1.

For segmentation 2 and 3 by using apps to interactively threshold images, image segmenter app. When segmenting an image, it used the interactive Image Segmenter app to check out several different approaches before settling on the one that gives the result. The first thing that need to do is open the image segmenter app. The following step is to segment the image where thresholding will be used to produce a mask. Sometimes the objects in the image that want to segment have pixel intensity values that are quite similar to one another, and these values can be clearly differentiated from the values in other parts of the image, such as the background. First need to create a mask then refine the segmentation. The app for image segmentation contains a number of tools that may be utilised for the purpose of satisfying holes and performing other operations. Click the fill hole button to make the initialization mask clearer. Create a mask image that have achieved the desired level of segmentation, which is the final step. Select export images after clicking the export button. In the end, save the image in the file using the Seg2 and Seg3.

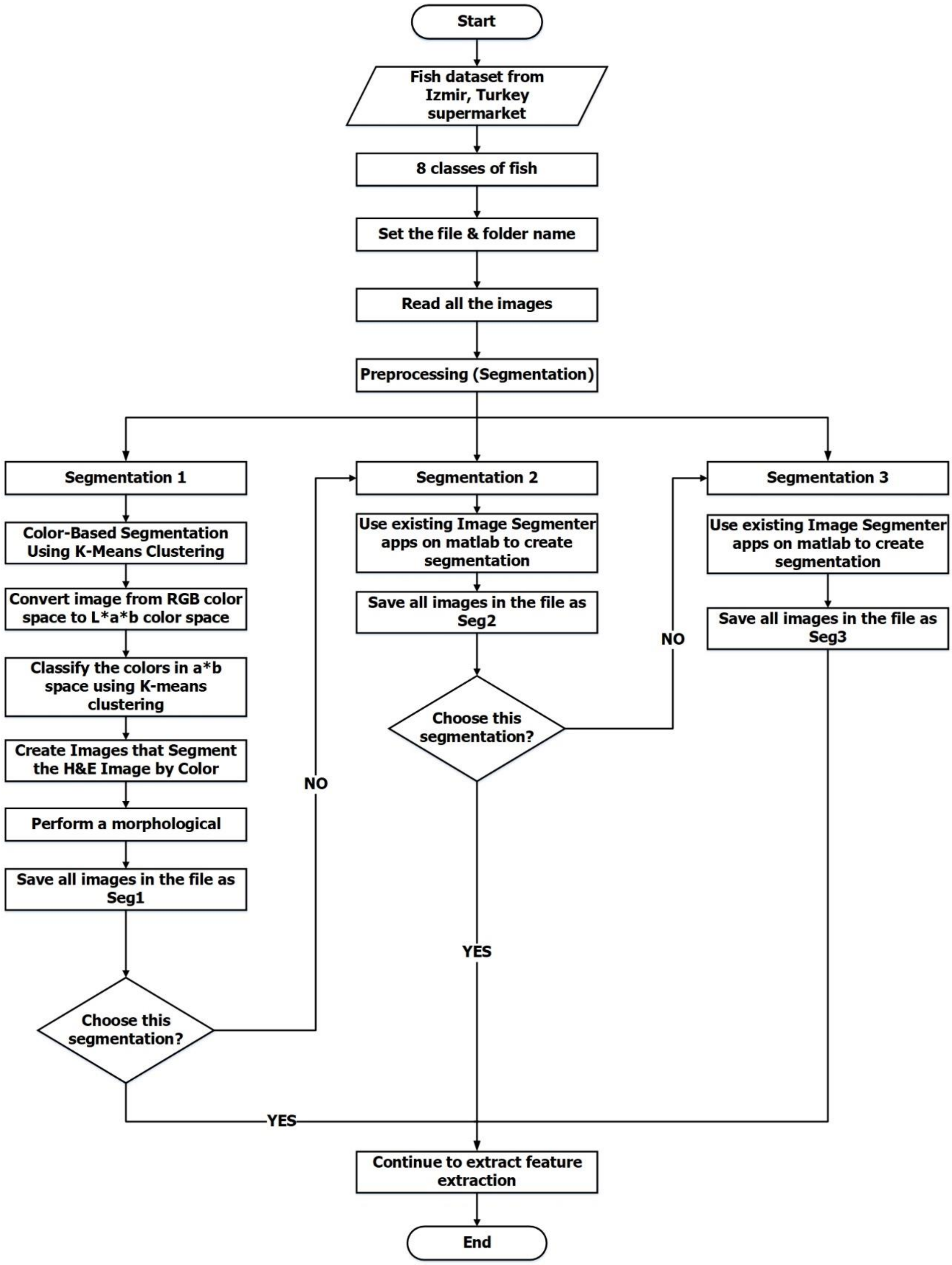


Figure 3.3 Flowchart of segmentation process

3.4 Feature Extraction

3.4.1 Texture

Image analysis approaches include the gray-level-co-occurrence Matrix (GLCM) and related texture feature estimates. The GLCM is a tabulation of how often distinct combinations of grey levels co-occur in an image or image part given an image made of pixels each with an intensity a specific grey level. The contents of the GLCM are used in texture feature computations to provide a measure of the variance in intensity for an example image texture at the pixel of interest (Uppuluri, 2008).

For a given pair of distance and angle, GLCM is computed. The GLCM matrix is created by calculating the relative frequencies of each reference pixel and its surrounding pixel at a given distance and angle. To obtain a normalised matrix, divide the resulting matrix by the sum of all frequencies. For D=1 and =0°, Figure 3.4 shows how GLCM is generated from the graycomatrix of 4-by-5 image (Singh et al., 2017).

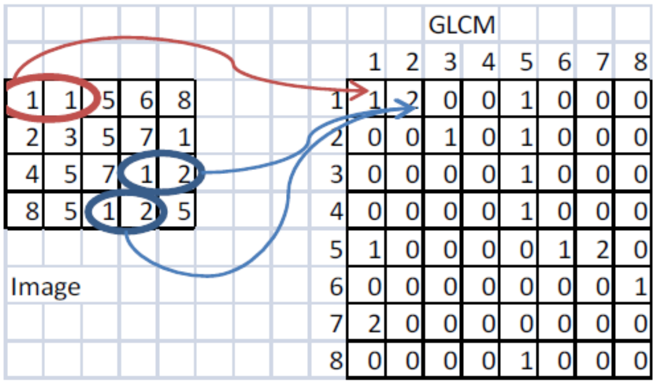


Figure 3.4 Calculation of the GLCM from the greycomatrix of a 4-by-5 picture I
Source: (Singh et al., 2007)

For varied offsets, several GLCMs can be computed. Pixel connections of different direction and distance are defined by these offsets. The spatial connections of pixels that are determined offsets for various angles, i.e., 0°, 45°, 90°, and 135° and distance D, where D is any fixed integer between 1 and the size of the image, are shown in Figure 3.5. Other angles, such as 180°, 225°, 270°, and 315°, can be used, but the results will be the same as 0°, 45°, 90°, and 135°. As a result, only four angles are taken into account (Singh et al., 2017).

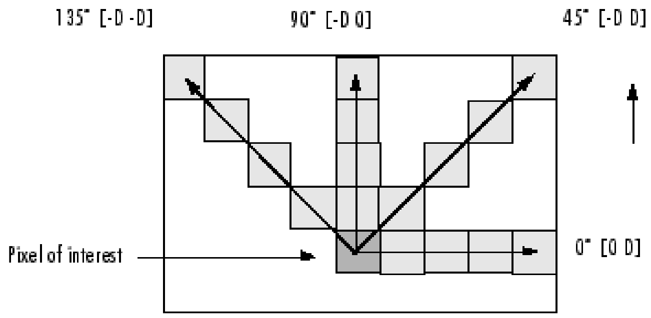


Figure 3.5 Multiple GLCM calculations with various offsets
Source: (Singh et al., 2007)

In a GLCM matrix, contrast represents the grey level variation. It essentially describes the linear relationship between the grey levels of two adjacent pixels. Its value ranges from 0 to Range = [0 (size(GLCM,1)-1)^2]. Energy is a metric measuring an image's textural homogeneity, or the number of pixel pair repeats. It also helps in the detection of texture issues. Its value is in the range [0 1] (Example & Example, n.d.).

Table 3.1 GLCM feature

Feature	*Formula*
Contrast Contrast is 0 for a constant image. The property contrast is also known as variance and inertia.	$\sum_{i,j} \lvert i-j\rvert^2 \, p(i,j)$ $Contrast\ (d,\theta) = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} \lvert i-j\rvert^2 p_d^{\theta}(i,j)$
Energy Energy is 1 for a constant image. The property Energy is also known as uniformity, uniformity of energy, and angular second moment. Source: (Singh et al., 2007), (Example)	$\sum_{i,j} p(i,j)^2$ $Energy\ (d,\theta) = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} [p_d^{\theta}(i,j)]^2$

3.4.2 Color

The construction of a colour histogram is a rather straightforward process. Based on the specification that was just presented, all that has to be done is to count the number of pixels for each of the 256 scales in each of the three RGB channels, and then plot those counts on three separate bar graphs.

A colour histogram is constructed on the basis of a particular colour space, such as RGB or HSV. When calculate the pixels of an image that are of different colours, it may first split the colour space into a given number of little intervals if the colour space is large enough. If the colour space is small, however, cannot divide it. A "bin" is the name given to each of the intervals. The quantization of colour is the procedure in question. After that, to obtain the colour histogram of the image by counting the number of pixels that are contained inside each of the bins. The intensities 0-63 are represented in the 0 bin. Bin 1 is 64-127. Bin 2 contains numbers 128-191, and Bin 3 has numbers 192-255.

```
%% Color Histogram

   r=RGB(:,:,1);
   g=RGB(:,:,2);
   b=RGB(:,:,3);

   hr=imhist(r);
   hg=imhist(g);
   hb=imhist(b);

   histvert=[hr; hg; hb];
   imghist=histvert';
```

Figure 3.6 Coding for color feature

3.4.3 Shape

Within the confines of this shape feature are the dimensions of area, width, and height. Where there will only be a single reading generated for each parameter. Begin with the image that has been transformed, then fill in the blanks and identify the object. The regionprops should then have the bwlabel, area, and bounding box values set. The next step is to locate the space that has area values that are more than 9000, and then to input that space's width and height values into the bounding box.

```
%% Area

%Convert image to black and white
d=im2bw(Img_gray);

%Fill the areas/holes
e=imfill(d,'holes');

%label object
f=bwlabel (e);

%vislabels(f),title('Each object labelled');
%get are boundingBox for each object
g=regionprops(f, 'Area','BoundingBox');

%display area
g(1);
area_values = [g.Area];
idx=find(area_values>9000);
h=ismember(f,idx);
bBox = g(idx).BoundingBox;
fishW = bBox(3);
fishH = bBox(4);
```

Figure 3.7 Coding for shape feature

3.5 Graphical User Interface (GUI) Design

As opposed to text-based user interfaces, written command labels, or text navigation, the graphical user interface, often known as the GUI, is a type of user interface that enables users to interact with electronic devices by means of graphical icons and auditory indication, such as primary notation. The apparent difficulty in learning command-line interfaces (CLIs), which need commands to be entered on a computer keyboard, was the impetus behind the development of graphical user interfaces (GUIs). first of all, after the full coding run save the data as (.mat) and after that put the training data, training response, testing data and testing response into the train and testing fish dataset. Next press browse image to choose a fish image. Then, for segmentation it is separated into three choices so the user must select the optimum segmentation either Seg1, Seg2 and Seg3. After that, proceed to extract the extraction feature where when the extract feature button is pressed all the features will come out where the values for contrast, energy, area, width, height and plotting of histogram for Red, Green and Blue. Following that, proceed to classify the data from the feature using KNN and SVM Classification where it can predict the name of the fish and will provide a prediction image. Finally, to receive the result for validation accuracy and testing accuracy click on choose classifier, select KNN or SVM and after that press the run button and the result of accuracy will come out. If user want to attempt for different type of fish just press the reset all button and redo all the procedures above.

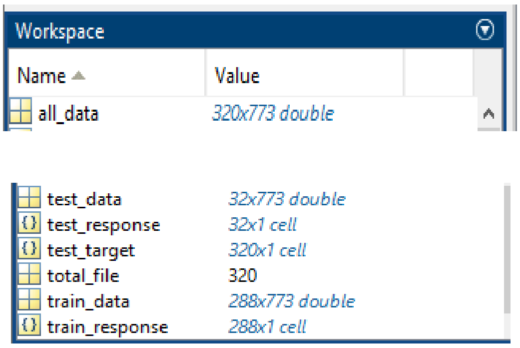
CHAPTER 4

Result

4.1 Introduction

The primary goal of this chapter is to extract the features of texture, colour, and shape, and then proceed with the classification process using KNN and SVM classifier. MATLAB is the platform that is used to run all the simulations. This will show where the data that was obtained from MATLAB will be displayed into graph form and then examined when it has been completed. Beginning with the segmentation process and continuing all the way up to the GUI design, this chapter will go through all the procedures and outcomes. The final number of records in this data set is 773, with one record each coming from the comparison, energy, contrast, area, width, and height categories, and 256 records coming from the histogram colour categories of red, green, and blue.

The ratio of training to testing to validation that is used to determine the accuracy of results for process classification is 80% training, 10% testing, and 10% validation. The data that are used for testing and training represent the data that are used for the entirety of the extraction feature, and the data that are used for training and testing responses represent the class name for each fish. Figure 4.1 shows the data for the training are 288 rows by 773 columns, and the training response is 288 rows by 1 column. The data for the testing are 32 rows by 773 columns, and the testing response is 32 rows by 1 column. The figure also shows all the data stored in the matlab file as (.mat). Therefore, the data set was prepared publicly by Oguzhan, Diclehan & Mehmet based on fish widely used in Izmir, Turkey.



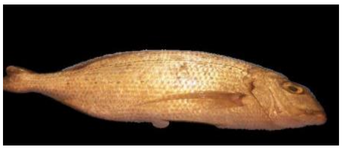

```
save 'train_data.mat' 'train_data';
save 'test_data.mat' 'test_data';
save 'train_response.mat' 'train_response';
save 'test_response.mat' 'test_response';
```

Figure 4.1 The compilation of data for each classes of fish

4.2 Segmentation

In the process of segmentation, there are three possible outcomes, which are designated by the numbers Seg1, Seg2, and Seg3. There is an image of a fish that, when utilising only one segmentation, is unable to generate a perfect segmentation. This is due to the size of the picture of the fish as well as the luminance that is present in the image of the fish. Then, the process of segmentation includes three different choices. Therefore, it is up to the user to determine which of the three possible segmentations will result in the highest quality and most ideal image. In order to segment the images for this research, color-based segmentation with K-means clustering and image segmenter applications that employ colour space Lab were utilised as segmentation methodologies. In the table that follows, there are 8 various types of fish images presented both before and after segmentation. Table 4.1 shows, the differences between all classes of fish images where without segmentation and after segmentation.

Table 4.1 Segmentation for 8 classes of fish

Without Segmentation	**After Segmentation**
Black Sea Sprat 	Black Sea Sprat 
Gilt-Head Bream 	Gilt-Head Bream 
Horse Mackerel 	Horse Mackerel 
Red Mullet 	Red Mullet 
Red Sea Bream 	Red Sea Bream 
Sea Bass 	Sea Bass 
Stripped Red Mullet 	Stripped Red Mullet 
Trout 	Trout 

4.3 Feature Extraction

4.3.1 Texture

It takes the GLCM representation of the texture and derives two parameters from that representation: contrast and energy. The findings indicate that the contrast and energy values are highest at angles of 0 degrees, 40 degrees, 95 degrees, and 135 degrees. After that, it determines the total amount of energy by adding together the squares of each element. The statistics obtained from each species of fish are distinct from one another. This is due to the fact that every image of every fish has a distinct lighting scheme, in addition to a varied angle, and a different size. Therefore, increased contrast and energy levels lead to better overall levels of accuracy. Table 4.2 show the data for contrast (C) and energy (E) for 320 images. All the data will combine with other feature and then extracted the data into KNN and SVM classifier.

Table 4.2 Data of Contrast and Energy

C	**E**	**C**	**E**	**C**	**E**	**C**	**E**	**C**	**E**
0.0939	0.6579	0.1548	0.3985	0.0720	0.6988	0.1399	0.4788	0.0845	0.6900
0.1088	0.6150	0.1201	0.4319	0.1108	0.5372	0.1314	0.4854	0.0743	0.7236
0.1080	0.6214	0.1077	0.4734	0.1094	0.5433	0.1631	0.3587	0.0651	0.7512
0.1150	0.5972	0.1167	0.4456	0.0951	0.5905	0.2071	0.3034	0.0614	0.7692
0.1234	0.5753	0.1275	0.4091	0.1005	0.6028	0.1735	0.3261	0.0916	0.6695
0.1002	0.6422	0.1540	0.4080	0.0692	0.7433	0.1487	0.3778	0.0786	0.7053
0.1113	0.6097	0.1527	0.4144	0.0687	0.7480	0.1506	0.4046	0.0742	0.7173
0.1023	0.6066	0.1889	0.3055	0.0751	0.7264	0.1893	0.3455	0.0864	0.6841
0.1237	0.5331	0.1638	0.3697	0.0633	0.7322	0.2096	0.4944	0.0647	0.7725
0.1270	0.5936	0.1417	0.4494	0.0874	0.6810	0.2039	0.4933	0.0608	0.7489
0.0890	0.6663	0.1363	0.4661	0.1204	0.5855	0.2659	0.3671	0.0568	0.7583
0.0961	0.6297	0.1393	0.3688	0.1304	0.5543	0.1973	0.4312	0.1017	0.5847
0.1050	0.6283	0.1408	0.3585	0.0750	0.7215	0.2030	0.5141	0.0880	0.6368

0.0968	0.6505	0.1139	0.5266	0.0705	0.7085	0.2122	0.4104	0.1020	0.6801
0.0902	0.6826	0.1107	0.5396	0.1117	0.5693	0.3109	0.4259	0.0995	0.6434
0.1018	0.6324	0.1087	0.4717	0.0977	0.5812	0.2166	0.4683	0.0622	0.7420
0.1068	0.6245	0.1790	0.4468	0.1050	0.6029	0.2318	0.4837	0.0674	0.7288
0.0868	0.6804	0.1406	0.4062	0.1060	0.5676	0.1418	0.5713	0.0799	0.6819
0.0867	0.6907	0.1400	0.4108	0.0772	0.6879	0.1770	0.5852	0.1105	0.6062
0.0799	0.7119	0.1295	0.4270	0.0901	0.6866	0.1862	0.4626	0.1248	0.4121
0.0949	0.6646	0.1332	0.4246	0.1061	0.5942	0.2386	0.4727	0.0654	0.7303
0.0869	0.6893	0.2014	0.4932	0.1158	0.5714	0.2017	0.5104	0.0618	0.7863
0.1117	0.6058	0.1106	0.4622	0.1172	0.6034	0.2068	0.5050	0.1560	0.5651
0.1091	0.6116	0.1428	0.4320	0.1247	0.5964	0.1777	0.4846	0.0999	0.6810
0.1017	0.6101	0.1079	0.5236	0.0660	0.7320	0.2244	0.3503	0.1529	0.2610
0.0940	0.6340	0.1161	0.4246	0.0706	0.7173	0.2143	0.4076	0.2550	0.2249
0.1139	0.5610	0.0986	0.4711	0.0830	0.6709	0.2053	0.5325	0.2643	0.1956
0.0966	0.5850	0.1214	0.4380	0.0835	0.6715	0.2654	0.4744	0.1788	0.3185
0.1050	0.5619	0.0979	0.4934	0.0745	0.6942	0.2262	0.4726	0.2387	0.1998
0.1039	0.5949	0.1286	0.4603	0.0787	0.6813	0.2279	0.4868	0.2488	0.2279
0.1015	0.6087	0.1341	0.4421	0.1286	0.5291	0.2349	0.4491	0.2443	0.2329
0.1001	0.6110	0.1180	0.4883	0.0792	0.6792	0.1832	0.4740	0.2477	0.1824
0.0966	0.5850	0.1151	0.4977	0.1376	0.4382	0.2120	0.4334	0.2211	0.1775
0.1093	0.6106	0.1109	0.5352	0.1597	0.4277	0.1926	0.4498	0.2525	0.2315
0.1164	0.5892	0.1040	0.5722	0.1825	0.3618	0.1727	0.5082	0.2674	0.2039
0.0987	0.6216	0.1337	0.4776	0.1859	0.3294	0.1767	0.5535	0.2316	0.1834
0.1043	0.6037	0.1209	0.5098	0.1358	0.4800	0.1941	0.4948	0.3210	0.1768
0.1196	0.5478	0.0970	0.5393	0.1700	0.4130	0.2220	0.5245	0.2273	0.2672
0.0853	0.6306	0.0972	0.5214	0.1691	0.4134	0.1550	0.5568	0.2164	0.2431

0.0528	0.7976	0.1181	0.4479	0.1340	0.4958	0.1496	0.5922	0.2439	0.1629
0.1153	0.5216	0.1258	0.4397	0.1367	0.4904	0.1840	0.5741	0.2382	0.1757
0.1151	0.5226	0.1706	0.5321	0.1209	0.4576	0.1765	0.5077	0.2602	0.1940
0.1111	0.4657	0.0930	0.5644	0.1677	0.3246	0.1789	0.4946	0.2636	0.2128
0.1597	0.3862	0.1399	0.4012	0.1207	0.5232	0.1788	0.6020	0.2102	0.2113
0.1713	0.3286	0.1458	0.5035	0.1265	0.5021	0.1699	0.5379	0.2250	0.1832
0.1678	0.2541	0.1115	0.4993	0.1178	0.5353	0.2096	0.4944	0.2223	0.1811
0.1238	0.3715	0.1181	0.4709	0.1588	0.4178	0.1575	0.5736	0.2302	0.1831
0.1396	0.4164	0.1616	0.4982	0.1860	0.3704	0.1780	0.5270	0.2365	0.1854
0.2122	0.4141	0.1774	0.4914	0.1421	0.4029	0.1560	0.5651	0.2198	0.1899
0.1566	0.3756	0.1532	0.5034	0.1455	0.3915	0.0982	0.7004	0.2414	0.2491
0.1308	0.4448	0.1678	0.4626	0.1792	0.3622	0.0858	0.6878	0.1929	0.2319
0.1388	0.4215	0.1563	0.4509	0.1279	0.4407	0.0929	0.6533	0.2381	0.2459
0.1574	0.3565	0.1102	0.5308	0.1824	0.3561	0.0901	0.6664	0.2186	0.3032
0.1350	0.4253	0.1030	0.5524	0.1819	0.3559	0.0662	0.7513	0.2462	0.1677
0.1301	0.4538	0.1391	0.3879	0.1187	0.4666	0.0654	0.7578	0.2327	0.1750
0.1379	0.4288	0.1296	0.4210	0.1235	0.4479	0.0593	0.7736	0.1450	0.3401
0.1073	0.4510	0.0951	0.6168	0.1804	0.3717	0.0606	0.7692	0.1742	0.3284
0.1186	0.4020	0.1489	0.5163	0.1602	0.3673	0.0831	0.6742	0.1665	0.3484
0.0950	0.5717	0.1252	0.5630	0.1308	0.5081	0.0633	0.7482	0.1686	0.3922
0.1052	0.5360	0.1116	0.6154	0.1294	0.4918	0.0943	0.6508	0.1543	0.3840
0.1012	0.4845	0.1129	0.6060	0.1655	0.3486	0.1437	0.5799	0.1610	0.4055
0.0997	0.4907	0.1032	0.5611	0.1473	0.4040	0.0682	0.7453	0.1235	0.5674
0.1695	0.3298	0.0966	0.5846	0.1208	0.5275	0.0838	0.6827	0.1232	0.5674
0.1460	0.3929	0.0748	0.6903	0.1363	0.4820	0.0824	0.6872	0.0760	0.6594

4.3.2 Color

The construction of a colour histogram is a rather simple task. Based on the specification that was just presented, all that has to be done is to count the number of pixels for each of the 256 scales in each of the three RGB channels, and then plot those counts on three separate bar graphs. In general, a colour histogram is constructed using the RGB colour space as its core. When computing the pixels of a distinct colour in an image, if the colour space is high, it is possible to first partition the colour space into a given amount of small intervals. This is only possible if the colour space has a large number of small intervals. A "bin" is the name given to each of the intervals. The quantization of colour is the procedure in issue. After that, to obtain the color histogram of the image by counting the number of pixels that are contained inside each of the bins. Based on the results below, this is a reading for black sea sprat fish(Wikipedia, 2010). Figure 4.1, 4.2 and 4.3 show the histogram of red, green and blue where it shows a more detailed result for each color found in the image of the fish, whether the color has a brighter or darker color.

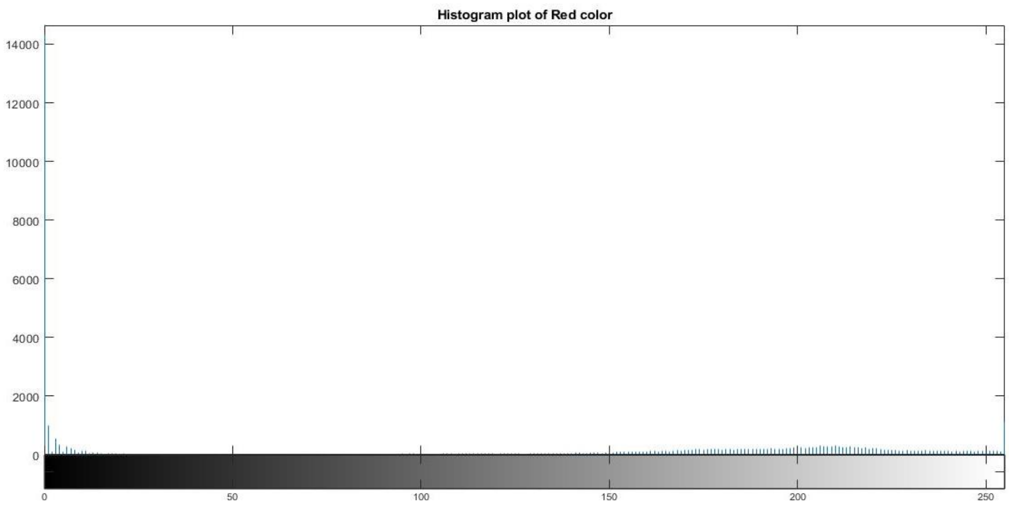


Figure 4.2 Histogram plot of Red Color

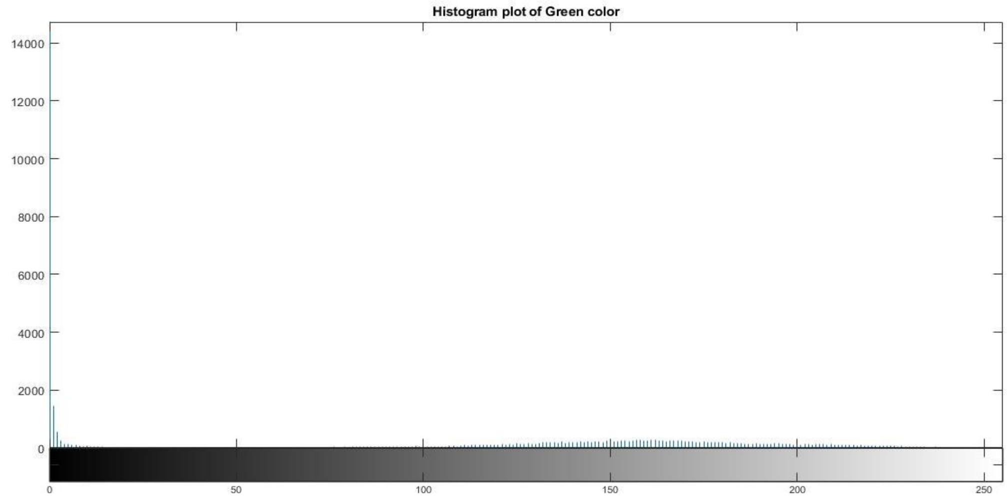


Figure 4.3 Histogram plot of Green Color

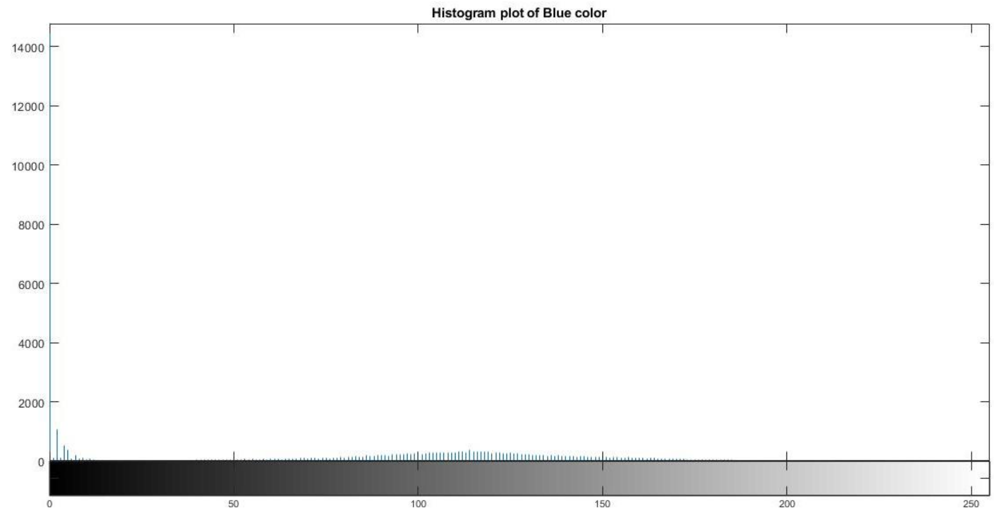


Figure 4.4 Histogram plot of Blue Color

4.3.3 Shape

The image of the fish is used to estimate the area, as well as the width and height of the fish. Where each piece of generated data likewise has a wide range of value differences. It generates unique data and this can facilitate the simplification of categorization processes. The results of 320 images for area (A), width (W) and height (H) are presented in Table 4.2. All of the data will be combined with other features, and then KNN and SVM classifiers will be used to extract the data from those combinations.

Table 4.3 Data of Area, Width and Height Extractions

A	W	H	A	W	H	A	W	H	A	W	H	A	W	H
19176	372	79	40170	335	301	9936	336	79	38403	339	181	15139	277	124
19187	359	127	40226	396	218	10406	329	108	38395	355	168	14775	277	124
22078	393	140	40236	401	202	9915	330	77	37246	335	182	14792	289	101
19216	351	149	40245	408	165	10389	319	128	38273	353	192	14796	288	101
21967	394	142	40230	413	162	10758	179	286	38086	311	224	14844	272	134
19204	297	230	40172	409	163	9740	381	87	37342	363	155	15161	273	133
19191	326	191	40176	410	164	10468	381	87	38388	367	168	14834	283	112
19709	360	136	39797	358	255	9672	375	100	38254	353	190	14887	264	146
19707	373	93	40213	361	276	10295	328	102	22750	372	111	14331	295	79
19751	326	198	40271	415	171	9740	382	77	23025	376	112	15359	294	92
19121	284	242	40181	415	171	9556	312	201	22794	364	108	15203	294	91
18537	301	196	40170	376	210	9467	315	202	21251	338	219	15689	289	107
19210	341	171	40217	229	369	9836	381	77	23304	345	182	15570	290	101
19193	322	197	40197	414	167	10453	341	79	23311	361	185	14364	280	144
19196	319	202	40181	415	167	10677	254	224	22465	219	349	15518	294	107
19179	280	247	40237	411	183	10527	321	108	23136	310	270	15308	295	86
19202	190	331	24639	337	137	12762	312	186	23088	269	316	15314	295	86
19228	368	102	24507	313	198	11194	336	78	23328	342	174	15350	295	85
19180	372	88	24522	313	197	12158	339	88	23343	345	183	15738	297	92
19197	372	89	26988	376	143	10738	386	76	23327	296	277	49512	410	229
19152	352	150	26937	343	236	10266	333	92	22576	238	327	15375	279	123
19232	352	150	26853	352	194	10322	348	82	23715	375	107	14288	289	98
19178	372	80	26896	349	211	10487	338	192	23071	375	108	14161	297	78
19178	372	80	26991	348	215	12280	321	263	22724	372	158	14297	273	130
21981	372	169	26964	343	239	12236	344	89	22246	337	152	31726	360	193
22063	356	200	26832	246	336	12625	344	94	23181	360	149	44236	437	221
21962	313	258	27059	322	260	12311	330	108	22881	365	108	42500	419	175
22212	193	371	27042	325	254	12898	338	107	22398	368	108	32534	369	208
21858	354	181	26975	324	267	12631	338	94	22740	364	112	43959	404	243
21778	172	362	26885	349	214	12289	342	89	23282	310	268	44529	419	229
22042	174	379	26840	349	211	12541	292	220	22197	293	237	44580	442	229
21794	363	176	26971	375	143	12461	342	95	22132	316	256	44175	348	296
22212	193	371	26753	374	144	33738	387	155	22001	272	293	44092	427	201
21968	346	217	26929	381	160	33890	357	210	22099	333	225	42115	406	175
21963	346	217	27058	381	159	33685	384	151	22144	351	194	42428	420	175
22039	356	203	26918	339	231	37334	353	188	22833	378	141	42464	363	243
21855	351	211	27054	232	342	37443	339	179	20987	379	140	43109	289	331
21932	403	103	26994	346	235	33849	389	143	20312	233	329	43033	168	426
22185	405	92	27074	355	221	33796	388	143	20274	320	249	42412	167	416
21969	94	397	24556	236	283	33955	357	210	21418	380	117	43838	221	395
40011	415	167	24563	313	197	33990	357	208	19991	379	98	44535	446	200

40011	415	167	24649	313	194	34544	261	312	20342	391	123	42502	416	185
40220	412	180	24486	304	207	34120	363	184	20001	355	106	42569	417	184
40318	415	167	24491	303	209	34522	356	212	20108	381	95	44382	450	204
39524	440	160	24447	294	193	34164	356	213	21170	399	115	43646	405	215
38980	399	185	24500	231	286	34439	364	198	22750	372	111	41108	360	174
38964	234	388	24475	282	234	33430	152	367	21521	381	112	42717	309	321
39604	388	245	24492	313	192	33581	381	152	20449	380	102	43534	262	354
39249	385	250	24546	306	195	33781	351	205	14161	297	78	41717	336	252
39412	447	167	24528	322	176	33993	341	183	14293	295	80	44376	424	231
39440	440	161	24668	338	136	34378	386	152	14910	295	93	42827	405	202
39478	442	161	24529	342	137	34451	389	143	15341	281	151	43079	425	172
39349	369	255	24536	338	138	34053	384	155	15225	273	133	43079	425	172
39573	369	268	24517	338	138	33983	385	154	14840	297	87	43815	391	260
39517	430	161	24135	300	173	34217	358	210	14839	295	87	41879	252	376
39424	431	160	25226	320	208	34494	362	200	14753	295	90	31727	194	363
39599	279	367	10233	335	79	38213	352	182	14795	295	90	32410	369	208
39486	368	255	9855	349	163	37707	345	178	14901	281	117	32473	312	262
39585	447	160	9672	374	77	38396	338	190	14870	287	103	39999	151	414
39493	448	160	9605	383	76	38374	359	167	15122	293	112	31762	152	385
39540	442	164	9725	381	87	38462	321	228	14161	294	96	31203	155	375
39427	444	164	10335	337	82	36820	296	219	14763	297	85	31811	156	384
39424	404	194	10493	335	83	37557	339	189	14864	292	108	32208	156	377
39453	402	194	10448	335	79	37485	332	189	14876	293	105	40250	152	422

4.3.4 Average and Standard Deviation

Table 4.4 shows, all the data of feature from texture and shape. Data were generated to analyse the difference, as well as grouping and plotting line graphs based on average and standard deviation in the findings. Figure 4.5 shows, all the generated data are plotting in a line graph. This assists in the determination of the mean value, which is the average of the data for each species of fish. In addition, the potential may be observed based on the result of the standard deviation applied to the data collected from the fish images.

Table 4.4 Average and Standard Deviation for 8 classes of fish

Name of fish	**8 classes of fish from Izmir, Turkey**		
	Feature Extraction	**Average**	**Standard Deviation**
Black Sea Sprat	Contrast	0.1014	0.0137
	Energy	0.6243	0.0487
	Area	20475	1404
	Width	323	73
	Length	193	90
Gilt-Head Bream	Contrast	0.1356	0.0262
	Energy	0.4271	0.0683
	Area	39800	407
	Width	396	51
	Length	206	63
Horse Mackerel	Contrast	0.1300	0.0257
	Energy	0.4764	0.0475
	Area	25751	1224
	Width	324	38
	Length	206	52
Red Mullet	Contrast	0.0958	0.0216
	Energy	0.6341	0.0693
	Area	10810	1107
	Width	337	37
	Length	118	57
Red Sea Bream	Contrast	0.1521	0.0243
	Energy	0.4196	0.0693
	Area	35781	1107
	Width	349	37
	Length	191	57
Sea Bass	Contrast	0.2018	0.0335
	Energy	0.4938	0.0585
	Area	22167	1120
	Width	342	45
	Length	178	78
Striped Red Mullet	Contrast	0.2018	0.0335
	Energy	0.4938	0.0585
	Area	22167	1120
	Width	342	45
	Length	178	78
Trout	Contrast	0.0853	0.0252
	Energy	0.6891	0.0748
	Area	15783	5487
	Width	291	21
	Length	108	28

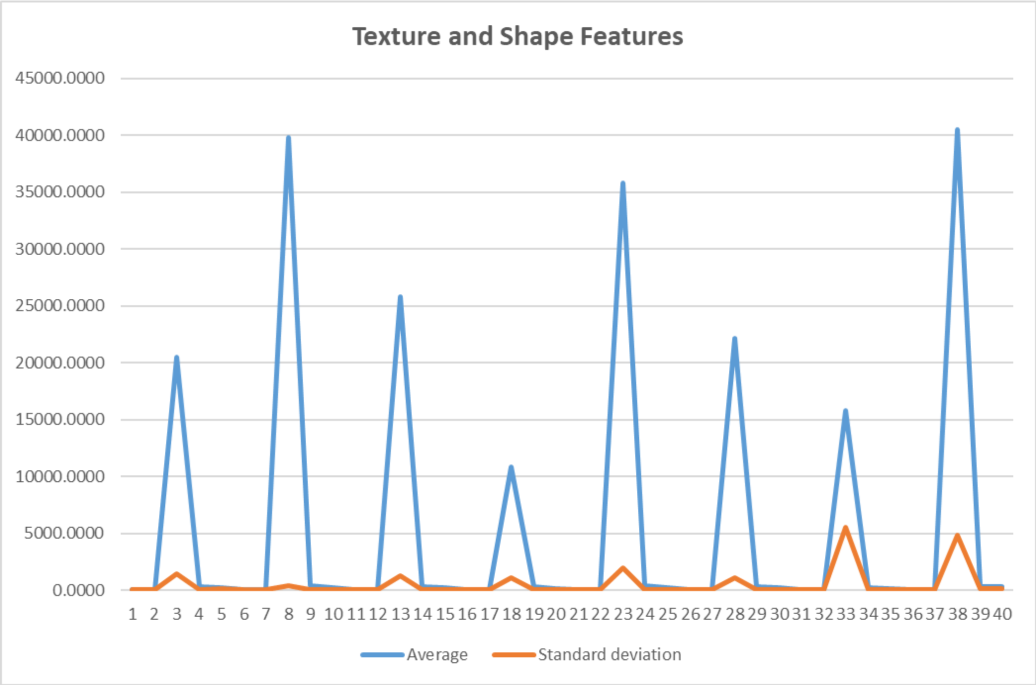


Figure 4.5 Line graph for texture and shape features

4.4 K-Nearest Neighbors (KNN)

The first thing that has to be done for KNN classification is to construct a function. This is the outcome of running MATLAB applications, namely the learner classification. Within this categorization, there are a few different kinds that have an accuracy rate of one hundred percent. In this project, the KNN classification that will be used is Fine KNN since the accuracy that it produces is one hundred percent, and because it is connected to basic theory, which makes this particular classifier simpler than the others. Table 4.5 show, accuracy for KNN with varying K from 1 to 30. From the figure 4.6, K value from 1 to 20 obtained accuracy 100%. The best result is K=1, with Euclidean distance. The Euclidean distance is utilised wherever possible since it is best suited for numerical predictors. The data have been grouped as random training data, testing data, and validation data for the 80:10:10 ratio.

Table 4.5 Number of K and Accuracy

KNN Classification	
No. of K	**Accuracy**
1	100%
2	100%
3	100%
4	100%
5	100%
6	100%
7	100%
8	100%
9	100%
10	100%
11	100%
11	100%
12	100%
13	100%
14	100%
15	100%
16	100%
17	100%
18	100%
19	100%
20	96.88%
21	96.88%
22	96.88%
23	96.88%
24	96.88%
25	96.88%

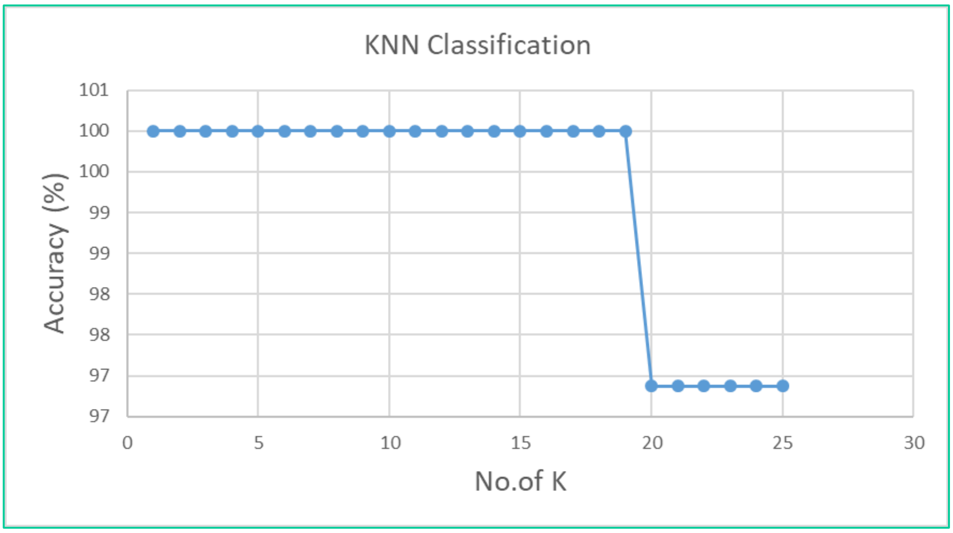
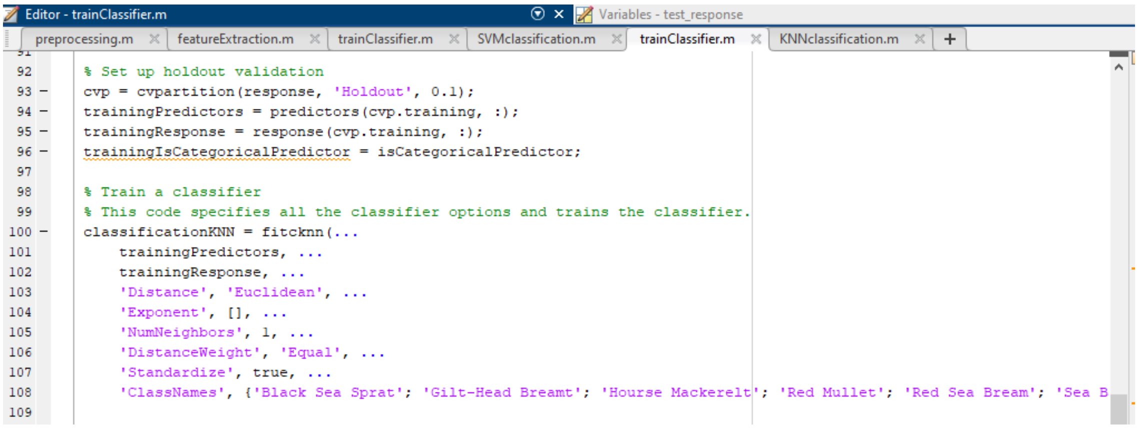


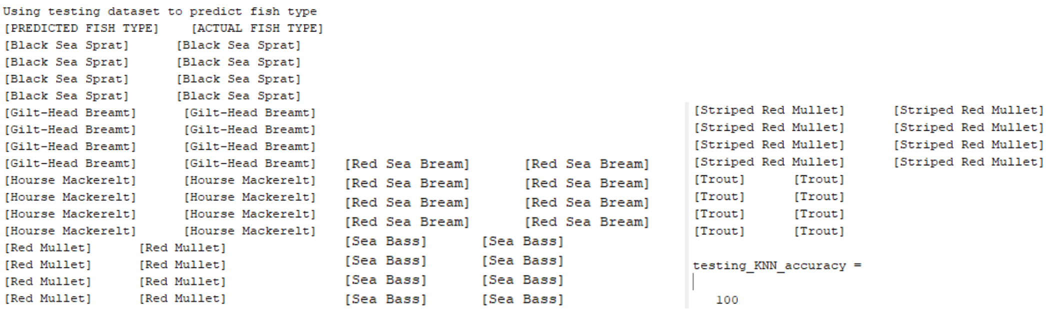
Figure 4.6 Plotting the KNN Classification line graph for number of K vs accuracy

After developing the function train classifier that shown in figure 4.7, input it into the KNN coding, and here the output reveals that there will be a fish name for both the actual name and the predicted name, in addition to the accuracy value that shown in figure 4.8.



```
% Set up holdout validation
cvp = cvpartition(response, 'Holdout', 0.1);
trainingPredictors = predictors(cvp.training, :);
trainingResponse = response(cvp.training, :);
trainingIsCategoricalPredictor = isCategoricalPredictor;

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
classificationKNN = fitcknn(...
    trainingPredictors, ...
    trainingResponse, ...
    'Distance', 'Euclidean', ...
    'Exponent', [], ...
    'NumNeighbors', 1, ...
    'DistanceWeight', 'Equal', ...
    'Standardize', true, ...
    'ClassNames', {'Black Sea Sprat'; 'Gilt-Head Breamt'; 'Hourse Mackerelt'; 'Red Mullet'; 'Red Sea Bream'; 'Sea B
```

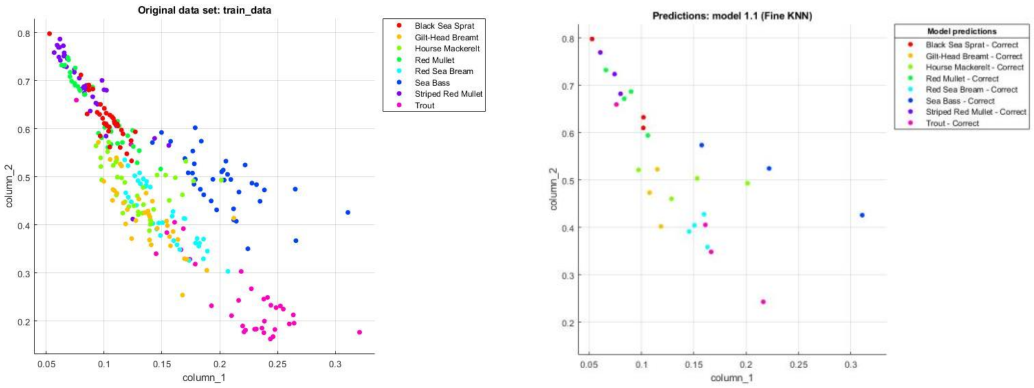
Figure 4.7 Function for KNN Classifier



```
Using testing dataset to predict fish type
[PREDICTED FISH TYPE]    [ACTUAL FISH TYPE]
[Black Sea Sprat]      [Black Sea Sprat]
[Black Sea Sprat]      [Black Sea Sprat]
[Black Sea Sprat]      [Black Sea Sprat]
[Black Sea Sprat]      [Black Sea Sprat]
[Gilt-Head Breamt]     [Gilt-Head Breamt]
[Gilt-Head Breamt]     [Gilt-Head Breamt]
[Gilt-Head Breamt]     [Gilt-Head Breamt]
[Gilt-Head Breamt]     [Gilt-Head Breamt]
[Hourse Mackerelt]     [Hourse Mackerelt]
[Hourse Mackerelt]     [Hourse Mackerelt]
[Hourse Mackerelt]     [Hourse Mackerelt]
[Hourse Mackerelt]     [Hourse Mackerelt]
[Red Mullet]      [Red Mullet]
[Red Mullet]      [Red Mullet]
[Red Mullet]      [Red Mullet]
[Red Mullet]      [Red Mullet]

[Red Sea Bream]      [Red Sea Bream]
[Red Sea Bream]      [Red Sea Bream]
[Red Sea Bream]      [Red Sea Bream]
[Red Sea Bream]      [Red Sea Bream]
[Sea Bass]      [Sea Bass]
[Sea Bass]      [Sea Bass]
[Sea Bass]      [Sea Bass]
[Sea Bass]      [Sea Bass]

[Striped Red Mullet]      [Striped Red Mullet]
[Striped Red Mullet]      [Striped Red Mullet]
[Striped Red Mullet]      [Striped Red Mullet]
[Striped Red Mullet]      [Striped Red Mullet]
[Trout]      [Trout]
[Trout]      [Trout]
[Trout]      [Trout]
[Trout]      [Trout]

testing_KNN_accuracy =

   100
```

Figure 4.8 Results that compare the predicted fish species to the actual fish type and their accuracy values

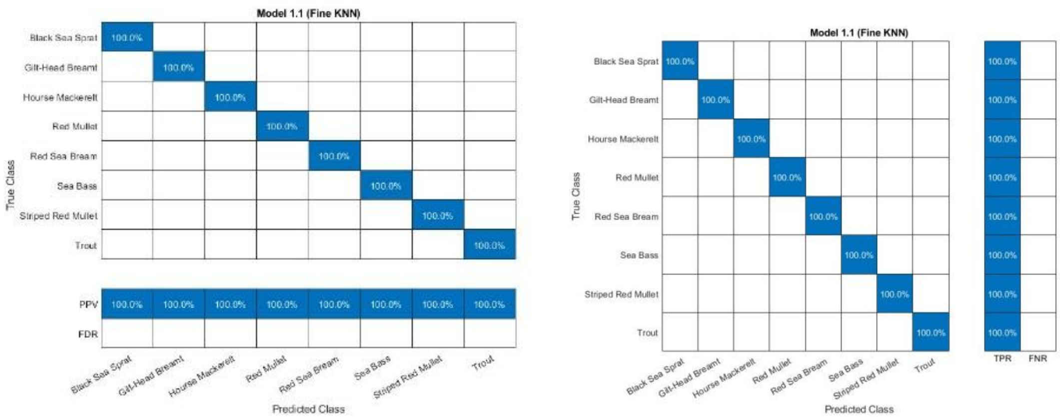
After that, there is something called a scatter diagram, which is essentially a representation of the data as an x-y graph that shown in figure 4.9 (a). It represents value for two separate numeric variables. Additionally, from figure 4.9 (b) prediction data is created in order to illustrate if the outcome was correct or incorrect. The data that were predicted using this categorization came out exactly as expected.



(a) (b)

Figure 4.9 Scatter diagram (a), predictions model of contrast and energy (b)

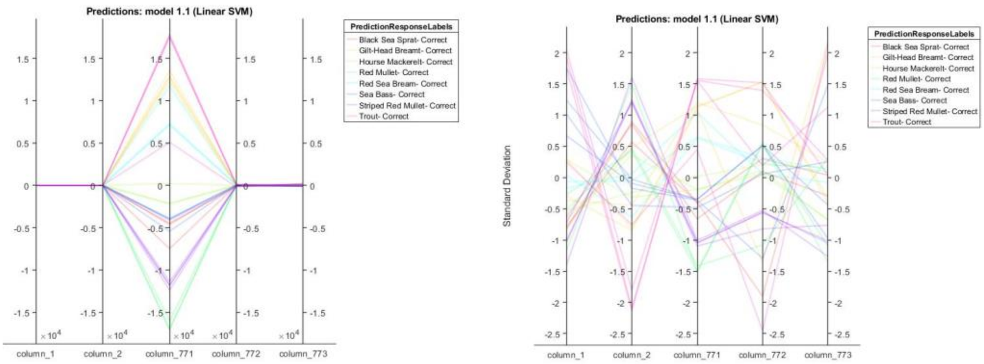
In addition, the confusion matrix, which includes validation confusion matrix validation (positive predictive value PPV AND False Discovery Rates FDR) that shown in figure 4.10 (a) and (True Positive Rates TPR AND False Negative Rates FNR) shown in figure 4.10 (b). The confusion matrix for the KNN shows that all the true class and predicted class get 100%.



(a) (b)

Figure 4.10 Validation confusion matrix (positive predictive value PPV AND False Discovery Rates FDR) (a), (True Positive Rates TPR AND False Negative Rates FNR) (b)

The measurements that are used for machine learning categorization. This demonstrates that all categories of fish receive perfect result. Figure 4.11 (a) and (b) show the classification includes not only a graph for the average but one for the standard deviation of the data as well. With the help of this graph, one is able to calculate not only the potential of the data but also the mean value of the average data for each species of fish.



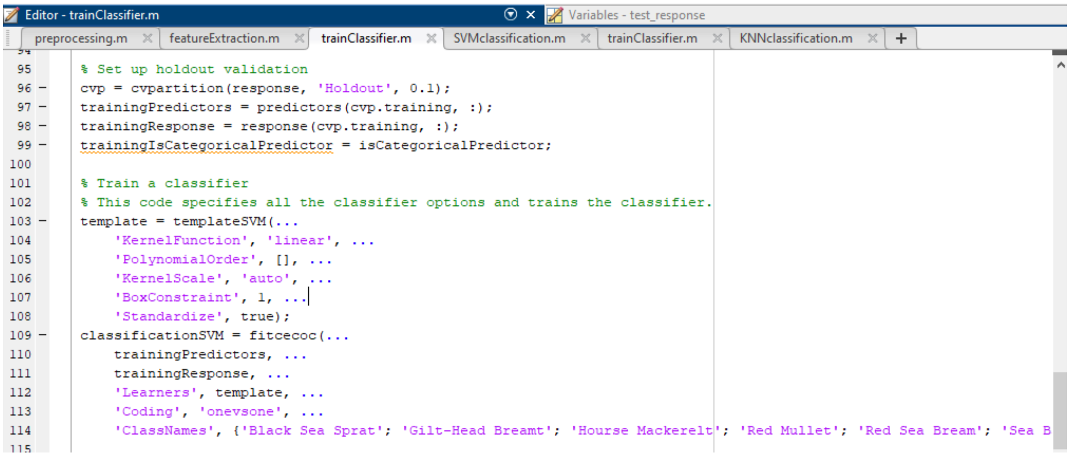
(a) (b)

Figure 4.11 Average (a), standard deviation (b) for texture and shape features

4.5 Support vector machine (SVM)

In order to get started with SVM classification, the first thing that needs to be done is the construction of a function. The learner classification is the result of executing programmes in MATLAB. Specifically, this is the output. There are a few distinct types that fall under this classification that are capable of achieving an accuracy 100%. The Linear Support SVM will be utilised as the classification strategy for this project. This is due to the fact that the accuracy that it generates is one hundred percent, and because it is connected to fundamental theory, it creates a straightforward linear separation between classes. Linear Kernel is used when the data is linearly separable then it can be separated using a single line. It is one of the most common kernels to be used. It is mostly used when there are a large number of features in a particular data set. Advantage of using Linear Kernel is training a SVM with a Linear Kernel is faster than with any other Kernel.

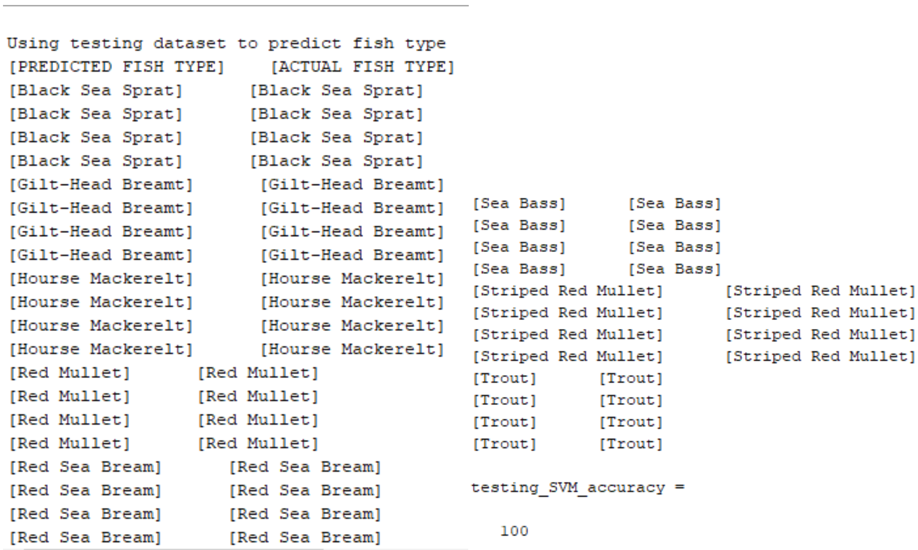
After the function train classifier that shown in figure 4.12, has been developed, it should be entered into the SVM coding.



```
% Set up holdout validation
cvp = cvpartition(response, 'Holdout', 0.1);
trainingPredictors = predictors(cvp.training, :);
trainingResponse = response(cvp.training, :);
trainingIsCategoricalPredictor = isCategoricalPredictor;

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
template = templateSVM(...
    'KernelFunction', 'linear', ...
    'PolynomialOrder', [], ...
    'KernelScale', 'auto', ...
    'BoxConstraint', 1, ...
    'Standardize', true);
classificationSVM = fitcecoc(...
    trainingPredictors, ...
    trainingResponse, ...
    'Learners', template, ...
    'Coding', 'onevsone', ...
    'ClassNames', {'Black Sea Sprat'; 'Gilt-Head Breamt'; 'Hourse Mackerelt'; 'Red Mullet'; 'Red Sea Bream'; 'Sea B
```

Figure 4.12 Function for SVM Classifier

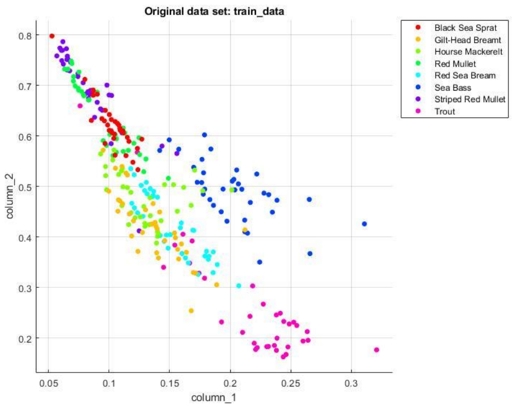
From figure 4.13, the output show that there will be a fish name for both the real name and the predicted name and also the accuracy value. As a result, it has an ideal accuracy of 100%.

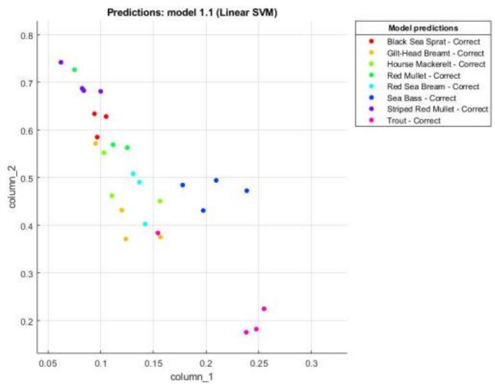


```
Using testing dataset to predict fish type
[PREDICTED FISH TYPE]    [ACTUAL FISH TYPE]
[Black Sea Sprat]      [Black Sea Sprat]
[Black Sea Sprat]      [Black Sea Sprat]
[Black Sea Sprat]      [Black Sea Sprat]
[Black Sea Sprat]      [Black Sea Sprat]
[Gilt-Head Breamt]      [Gilt-Head Breamt]
[Gilt-Head Breamt]      [Gilt-Head Breamt]
[Gilt-Head Breamt]      [Gilt-Head Breamt]
[Gilt-Head Breamt]      [Gilt-Head Breamt]
[Hourse Mackerelt]      [Hourse Mackerelt]
[Hourse Mackerelt]      [Hourse Mackerelt]
[Hourse Mackerelt]      [Hourse Mackerelt]
[Hourse Mackerelt]      [Hourse Mackerelt]
[Red Mullet]      [Red Mullet]
[Red Mullet]      [Red Mullet]
[Red Mullet]      [Red Mullet]
[Red Mullet]      [Red Mullet]
[Red Sea Bream]      [Red Sea Bream]
[Red Sea Bream]      [Red Sea Bream]
[Red Sea Bream]      [Red Sea Bream]
[Red Sea Bream]      [Red Sea Bream]
[Sea Bass]      [Sea Bass]
[Sea Bass]      [Sea Bass]
[Sea Bass]      [Sea Bass]
[Sea Bass]      [Sea Bass]
[Striped Red Mullet]      [Striped Red Mullet]
[Striped Red Mullet]      [Striped Red Mullet]
[Striped Red Mullet]      [Striped Red Mullet]
[Striped Red Mullet]      [Striped Red Mullet]
[Trout]      [Trout]
[Trout]      [Trout]
[Trout]      [Trout]
[Trout]      [Trout]

testing_SVM_accuracy =

   100
```

Figure 4.13 Results that compare the predicted fish species to the actual fish type and their accuracy values

The next step is something called a scatter diagram, which is just a representation of the data as an x-y graph shown in figure 4.14 (a). This step follows the previous one from the KNN classifier. It represents value for two separate numeric variables. In addition, prediction data is produced so that it may be shown if the outcome was correct or incorrect that shown in figure 4.14 (b). The data were able to be predicted using this categorisation, and the predictions were spot on with the actual data.

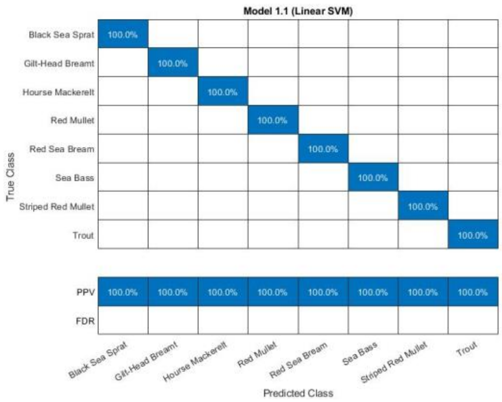




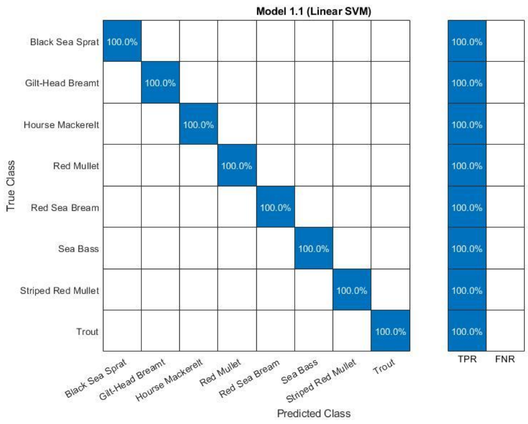
(a) (b)

Figure 4.14 Scatter diagram (a), predictions model(b) of contrast and energy

In addition to it, there is also the confusion matrix, which contains validation confusion matrix validation (positive predictive value PPV AND False Discovery Rates FDR) that shown in figure 4.15 (a) and (True Positive Rates TPR AND False Negative Rates FNR) metrics shown in figure 4.15 (b) that are utilised for classification in machine learning. The confusion matrix for the SVM shows that all the true class and predicted class get 100%. It is able to make predictions by applying machine learning, which is based on the data. When the data match the expectations exactly, this indicates that the data are valid.



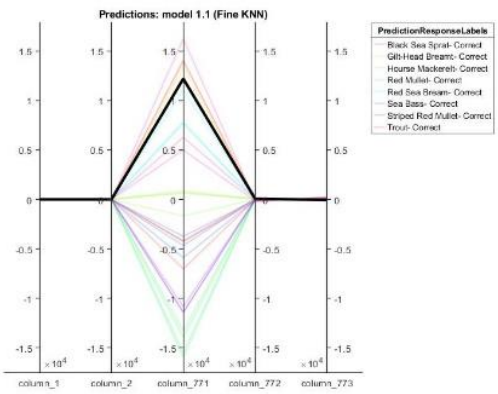
(a)



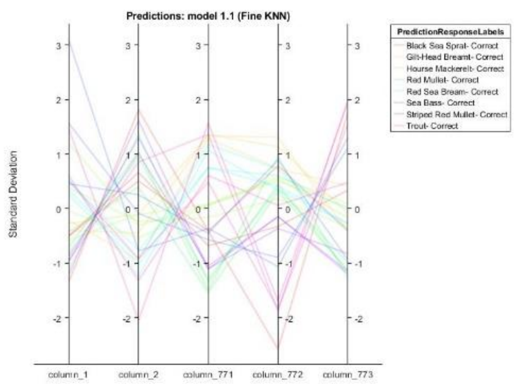
(b)

Figure 4.15 Validation confusion matrix (positive predictive value PPV AND False Discovery Rates FDR) (a), (True Positive Rates TPR AND False Negative Rates FNR) (b)

Figure 4.16 (a) and (b) show line graph for the average and the standard deviation included in this categorization. Using this graph, one can determine the mean value of the average data for each species of fish as well as the potential of the data.



(a)



(b)

Figure 4.16 Average (a), standard deviation (b) for texture and shape features

This is a graphical user interface design for the entire project. Figure 4.17 show interface before run where it begins with inputting all of the data, then moves on to selecting an image, and finally selects segmentation. After that, proceed with the feature extraction process, which includes analysing the texture, colour and shape of the image. Lastly, proceed with the classification process using the KNN and SVM classifiers. Following that attempt, when the accuracy result was obtained. Figure 4.18 show, the result where the interface after run where all the features are extracted and the classification get the 100% accuracy.

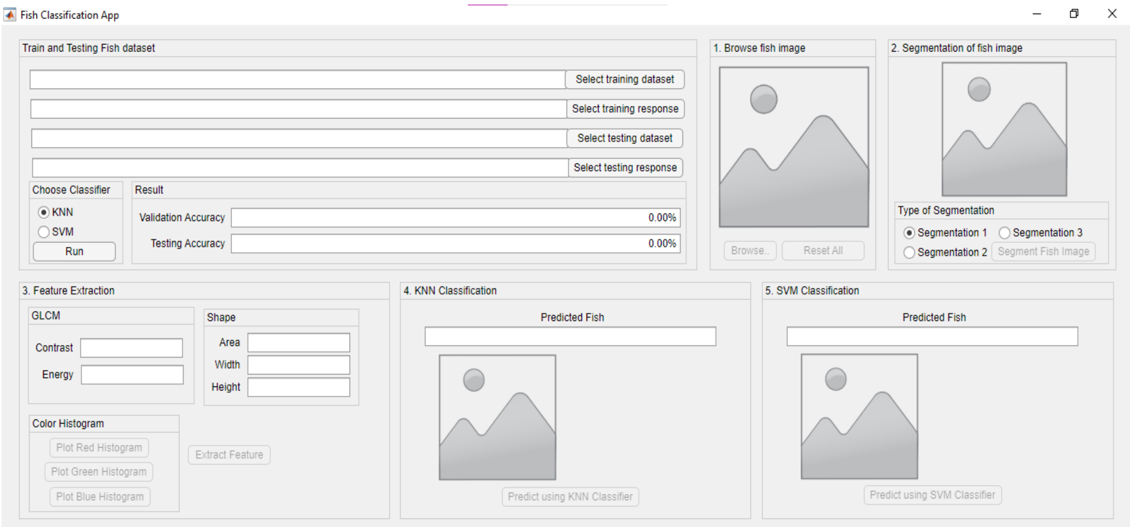


Figure 4.17 Graphical user interface (GUI) design before run

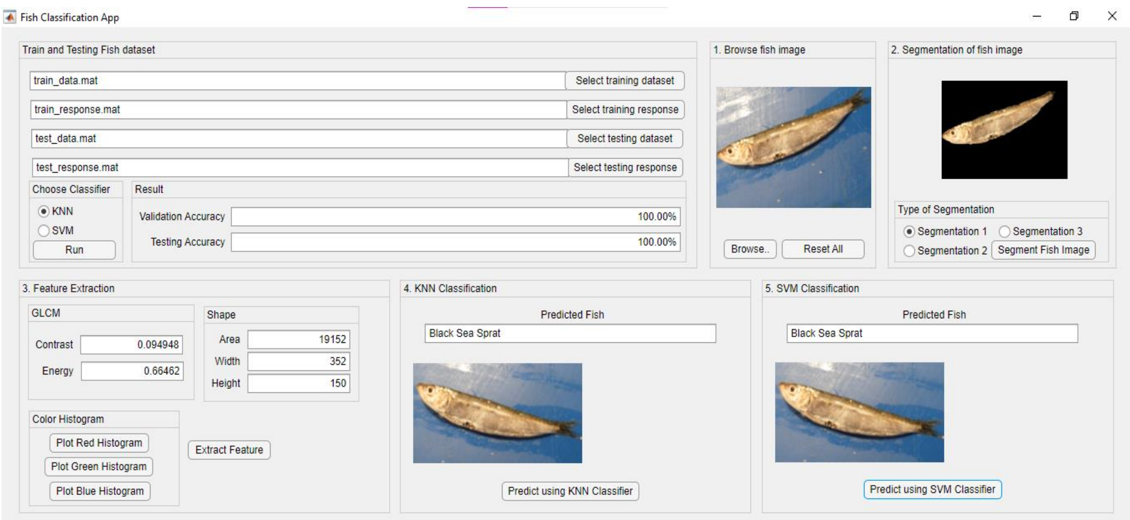


Figure 4.18 Graphical user interface (GUI) design after run

4.6 Summary

By using machine learning, the accuracy of both the KNN and SVM classification methods was determined to 100%. This is due to the fact that the image on each fish is quite clear in terms of lighting, and the background that was utilised is also highly appropriate for the image. That why when the process of extraction and classification of features is done, the data can be predicted accurately.

CHAPTER 5

CONCLUSION

5.1 Conclusion

In conclusion, this research uses four different extraction characteristics for each species of fish that it of fish that it analyses. One of these features is GLCM's texture analysis, which measures the value of contrast and energy. The second type of feature is the colour feature, which is generated by employing the colour histogram in order to acquire the values of red, green, and blue respectively. The shape is the last feature, and it is the one that can measure the values for area, width, and height. Following the discovery of the extraction feature, the data set will be classified utilising KNN and SVM classifiers. For the purpose of producing a percentage value based on the classification, which selects the percentage for training and validation to be 90% and testing to be 10%. Both classifiers have an accuracy rate of one hundred percent.

From the first objective, this experiment succeeded in extracting a features from texture using GLCM to measure the contrast and energy values. Then, extract the feature from color using the color histogram to measure the Red, Green, and Blue values. Extract the last feature from the shape where measure the area, width, and height values. Next, the experiment then successfully classifies all the 8 classes of fish for KNN and SVM with k=1 and accuracy=100%.

5.2 Recommendation

The planning for future researchers is to use the latest algorithm which is deep learning. Prior to this, all image processing projects employed machine learning to extract features from images and then classifiers to determine accuracy, sensitivity, and other parameters. However, modern deep learning algorithms are able to create new features on their own without further human input from a small selection of characteristics included in the training dataset. Therefore, deep learning can handle challenging jobs that frequently involve substantial feature engineering. Finally, due to the 100% accuracy value, researchers also may improve by adding new fish species for future research.

REFERENCES

Alsmadi, M. K., & Almarashdeh, I. (2020). A survey on fish classification techniques. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2020.07.005

Anand, A. (2018). Unit 13 Image Classification. *Tutorial on Processing and Classification of Remotely Sensed Images*, *May*, 41–58.

Cunningham, P., & Delany, S. J. (2021). K-Nearest Neighbour Classifiers-A Tutorial. *ACM Computing Surveys*, *54*(6). https://doi.org/10.1145/3459665

Elnemr, H. A., Zayed, N. M., & Fakhreldein, M. A. (2015). *Feature Extraction Techniques*. 264–294. https://doi.org/10.4018/978-1-4666-8654-0.ch013

Evgeniou, T., & Pontil, M. (2001). Support vector machines: Theory and applications. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2049 LNAI*(January 2001), 249–257. https://doi.org/10.1007/3-540-44673-7_12

Example, T. T., & Example, T. T. (n.d.). *Try This Example Copy Command* □. 2–5.

Garcia, R., Prados, R., Quintana, J., Tempelaar, A., Gracias, N., Rosen, S., Vågstøl, H., & Løvall, K. (2020). Automatic segmentation of fish using deep learning with application to fish size measurement. *ICES Journal of Marine Science*, *77*(4), 1354–1366. https://doi.org/10.1093/icesjms/fsz186

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. (2006). *Feature Extraction Foundations*. 1–8.

Mohanaiah, P., Sathyanarayana, P., & Gurukumar, L. (2013). Image Texture Feature Extraction Using GLCM Approach. *International Journal of Scientific & Research Publication*, *3*(5), 1–5.

Peterson, L. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2), 1883. https://doi.org/10.4249/scholarpedia.1883

Singh, S., Srivastava, D., & Agarwal, S. (2017). GLCM and its application in pattern recognition. *5th International Symposium on Computational and Business Intelligence, ISCBI 2017*, 20–25. https://doi.org/10.1109/ISCBI.2017.8053537

Ulucan, O., Karakaya, D., & Turkan, M. (2020). A Large-Scale Dataset for Fish Segmentation and Classification. *Proceedings - 2020 Innovations in Intelligent Systems and Applications Conference, ASYU 2020*. https://doi.org/10.1109/ASYU50717.2020.9259867

Uppuluri, A. (2008). *GLCM texture features*. 1–4. https://de.mathworks.com/matlabcentral/fileexchange/22187-glcm-texture-features

Wikipedia, F. (2010). Color Histogram. *Image Rochester NY,* 1–4. http://rsb.info.nih.gov/ij/plugins/color-histogram.html

Yu, C., Fan, X., Hu, Z., Xia, X., Zhao, Y., Li, R., & Bai, Y. (2020). Segmentation and measurement scheme for fish morphological features based on Mask R-CNN. *Information Processing in Agriculture*, *7*(4), 523–534. https://doi.org/10.1016/j.inpa.2020.01.002

Zhang, T. T., Chow, C. Y., & Zhang, J. D. (2021). Fish Image Instance Segmentation: An Enhanced Hybrid Task Cascade Approach. *Proceedings - 2021 IEEE 15th International Conference on Semantic Computing, ICSC 2021*, 306–313. https://doi.org/10.1109/ICSC50631.2021.00058

APPENDIX

Summary of literature review for journal 1

Journal	Author, Year	Remark
A Large-Scale Dataset for Fish Segmentation and Classification	O. Ulucan, D. Karakaya and M. Turkan, "A Large-Scale Dataset for Fish Segmentation and Classification," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 2020, pp. 1-5, doi: 10.1109/ASYU50717.2020.9259867.	• Fish disease identification and classification • A practical and large dataset containing nine distinct seafood widely consumed in the Aegean Region of Turkey is formed. • Fresh fish are used in the picture acquisition process, and they are positioned in a variety of positions and angles. • Used 4 feature extractions which are GLCM, moments, BoF algorithm and Convolutional neural networks (CNNs). Svm as a classifier. • The need for a publicly available dataset has been fulfilled, and it will be useful in future research.

Summary of literature review for journal 2 and 3

Journal	Author, Year	Remark
A Survey on Fish Classification Techniques	Alsmadi, Mutasem & Almarashdeh, Ibrahim. (2020). A Survey on fish classification Techniques. Journal of King Saud University - Computer and Information Sciences. 10.1016/j.jksuci.2020.07.005.	• FC techniques performance is compared relying on the availability of preprocessing and feature extraction methods, the number of extracted features and classification accuracy, the number of fish families /species recognized. • Used the shape feature, the local and global feature, the color feature, the texture feature and the combination-based feature extraction methods.
Segmentation and measurement scheme for fish morphological features based on Mask R-CNN	Yu, C., Fan, X., Hu, Z., Xia, X., Zhao, Y., Li, R., & Bai, Y. (2020). Segmentation and measurement scheme for fish morphological features based on Mask R-CNN. Information Processing in Agriculture, 7(4), 523-534.	• This paper proposes a scheme for segmenting fish image and measuring fish morphological features indicators based on Mask R-CNN. • The morphological features of fish, such as the body length, the body width, the caudal peduncle length, the caudal peduncle width, the pupil diameter, and the eye diameter are very important indicators in smart mariculture.

Summary of literature review for journal 4 and 5

Journal	Author, Year	Remark
Automatic segmentation of fish using deep learning with application to fish size measurement.	García, R., Prados, R., Quintana, J., Tempelaar, A., Gracias, N.R., Rosen, S., Vågstøl, H., & Løvall, K. (2019). Automatic segmentation of fish using deep learning with application to fish size measurement. ICES Journal of Marine Science.	• The proposal is based on the processing of stereo images acquired by the Deep Vision imaging system, directly placed in the trawl. • The images are pre-processed to correct for nonlinearities of the camera response. • Then, a Mask R-CNN architecture is used to localize and segment each individual fish in the images. • Ability to successfully deal with cluttered images containing overlapping fish.
Fish Image Instance Segmentation: An Enhanced Hybrid Task Cascade Approach	T. -T. Zhang, C. -Y. Chow and J. -D. Zhang, "Fish Image Instance Segmentation: An Enhanced Hybrid Task Cascade Approach," 2021 IEEE 15th International Conference on Semantic Computing (ICSC), 2021, pp. 306-313, doi: 10.1109/ICSC50631.2021.00058.	• The Hybrid Task Cascade (HTC) is a novel CNN model which applies cascade architecture to achieve boosted performance in the instance segmentation task. • Used an Enhanced Hybrid Task Cascade (EHTC) model to overcome these limitations. • EHTC help to resize images and optimize features that can be more easily understood by the later instance segmentation network. • MaskIoU, to generate mask confidence scores providing the mask that improves the instance segmentation accuracy.

```
% image segmentation

clear all; clc; close all;

folder_name= 'Mix 2';
file_name= dir(fullfile(folder_name, '*.png'));
total_file= length(file_name);
test_data= size(total_file,2);S
i=1;

%% Segmentation 1
for k =1:total_file
    RGB= imread(fullfile(folder_name, file_name(k).name));
    %figure,imshow(Img)

    %kmeans-clustering coding
    numColors = 3; %RGB
    L = imsegkmeans(RGB,numColors); % segments image RGB into numcolors clusters by↙
performing k-means clustering and returns the segmented labeled output in L.
    B = labeloverlay(RGB,L); %Overlay label matrix regions on 2-D image
    %menggabungkan imej input, RGB, dengan warna yang berbeza untuk setiap label bukan↙
sifar dalam matriks label L. Fungsi tindanan label tidak menggabungkan latar belakang


    lab_he = rgb2lab(RGB); %Convert RGB to LAB color space

    %Classify the Colors in 'a*b*' Space Using K-Means Clustering
    ab = lab_he(:,:,2:3); %2:3 mewakili a dan b
    ab = im2single(ab);
    pixel_labels = imsegkmeans(ab,numColors,'NumAttempts',3); % repeat the clustering 3↙
times to avoid local minima
    B2 = labeloverlay(RGB,pixel_labels);

    %Create Images that Segment the H&E Image by Color
    mask2 = pixel_labels == 2;

    %Perform a morphological
    SE = strel('disk',10); % Dilate mask with disk
    mask2 = imclose(mask2,SE);
    mask2 = imfill(mask2,'holes');
    cluster2 = RGB.*uint8(mask2);

%    subplot(8,6,i);
%     imshow(cluster2);
%     title(file_name(k).name);
%     i=i+1;
%simpan semua data dalam folder save as Seg1
    [filepath,name,ext] = fileparts(file_name(k).name);
    filename = strcat('Seg1/',name,'.jpg');
    imwrite(cluster2,filename,'jpg');
end
```

```
%% Segmentation 2
for k =1:total_file
    RGB= imread(fullfile(folder_name, file_name(k).name));

    [BW,maskedImage] = segmentImage2(RGB);

    [filepath,name,ext] = fileparts(file_name(k).name);
    filename = strcat('Seg2/',name,'.jpg');
    imwrite(maskedImage,filename,'jpg');
end

%% Segmentation 3
for k =1:total_file
    RGB= imread(fullfile(folder_name, file_name(k).name));

    [BW,maskedImage] = segmentImage3(RGB);

    [filepath,name,ext] = fileparts(file_name(k).name);
    filename = strcat('Seg3/',name,'.jpg');
    imwrite(maskedImage,filename,'jpg');
end
```

```
clear all; clc; close all;

folder_name= 'Segmented Image';
file_name= dir(fullfile(folder_name, '*.jpg'));
total_file= length(file_name);

for k =1:total_file
    RGB= imread(fullfile(folder_name, file_name(k).name));
    %figure,imshow(Img)

    %RGB TO GRAY
    Img_gray=rgb2gray (RGB);

    pixel_dist =1;
    GLCM= graycomatrix(Img_gray, 'Offset',[ 0 pixel_dist; -pixel_dist pixel_dist; -↙
pixel_dist 0; -pixel_dist -pixel_dist]);
    stats = graycoprops(GLCM,'Contrast Energy');
    Contrast = mean(stats.Contrast);
    Energy = mean(stats.Energy);

%% Color Histogram

   r=RGB(:,:,1);
   g=RGB(:,:,2);
   b=RGB(:,:,3);

   hr=imhist(r);
   hg=imhist(g);
   hb=imhist(b);

   histvert=[hr; hg; hb];
   imghist=histvert';


%% Area

%Convert image to black and white
d=im2bw(Img_gray);

%Fill the areas/holes
e=imfill(d,'holes');

%label object
f=bwlabel (e);

%vislabels(f),title('Each object labelled');
%get are boundingBox for each object
g=regionprops(f, 'Area','BoundingBox');

%display area
g(1);
```

```
area_values = [g.Area];
idx=find(area_values>9000);
h=ismember(f,idx);
bBox = g(idx).BoundingBox;
fishW = bBox(3);
fishH = bBox(4);


data = [Contrast Energy imghist area_values(idx) fishW fishH];
all_data(k,:) = data;


end

test_data = all_data(1:10:320,:); % 10% of all data %ambik data 1 daripada 10
train_data = [];

for i=2:10:320
    train_data =[train_data; all_data(i:i+8,:)]; %ambik data yang kedua daripada 10
end

%% Classification

test_target= cell(total_file,1);

for k=1:40
    test_target{k}= 'Black Sea Sprat';
end

for k=41:80
    test_target{k}= 'Gilt-Head Breamt';
end

for k=81:120
    test_target{k}= 'Hourse Mackerelt';
end

for k=121:160
    test_target{k}= 'Red Mullet';
end

for k=161:200
    test_target{k}= 'Red Sea Bream';
end

for k=201:240
    test_target{k}= 'Sea Bass';
end

for k=241:280
    test_target{k}= 'Striped Red Mullet';
```

```
end

for k=281:320
    test_target{k}= 'Trout';
end

i=1;
test_response = cell(32,1);
for j = 1:10:320
    test_response(i) = test_target(j);
    i=i+1;
end

d=1:320;
d2=[];
for i=2:10:320
    d2=[d2 d(i:i+8)];
end

train_response = cell(288,1);
for i = 1:length(d2)
 train_response(i) = test_target(d2(i));
end

save 'train_data.mat' 'train_data';
save 'test_data.mat' 'test_data';
save 'train_response.mat' 'train_response';
save 'test_response.mat' 'test_response';
```

```
function [trainedClassifier, validationAccuracy] = trainClassifier(trainingData,↙
responseData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData,
% responseData)
% Returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: A matrix with the same number of columns and data type
%       as the matrix imported into the app.
%
%      responseData: A vector with the same data type as the vector
%       imported into the app. The length of responseData and the number of
%       rows of trainingData must be equal.
%
%  Output:
%      trainedClassifier: A struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: A function to make predictions on new
%       data.
%
%      validationAccuracy: A double containing the accuracy in percent. In
%       the app, the History list displays this overall accuracy score for
%       each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input arguments trainingData and responseData.
%
% For example, to retrain a classifier trained with the original data set T
% and response Y, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T, Y)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a matrix containing only the predictor columns used for
% training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 28-May-2022 10:23:11


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
```

```
% Convert input to table
inputTable = array2table(trainingData, 'VariableNames', {'column_1', 'column_2',↙
'column_3', 'column_4', 'column_5', 'column_6', 'column_7', 'column_8', 'column_9',↙
'column_10', 'column_11', 'column_12', 'column_13', 'column_14', 'column_15',↙
'column_16', 'column_17', 'column_18', 'column_19', 'column_20', 'column_21',↙
'column_22', 'column_23', 'column_24', 'column_25', 'column_26', 'column_27',↙
'column_28', 'column_29', 'column_30', 'column_31', 'column_32', 'column_33',↙
'column_34', 'column_35', 'column_36', 'column_37', 'column_38', 'column_39',↙
'column_40', 'column_41', 'column_42', 'column_43', 'column_44', 'column_45',↙
'column_46', 'column_47', 'column_48', 'column_49', 'column_50', 'column_51',↙
'column_52', 'column_53', 'column_54', 'column_55', 'column_56', 'column_57',↙
'column_58', 'column_59', 'column_60', 'column_61', 'column_62', 'column_63',↙
'column_64', 'column_65', 'column_66', 'column_67', 'column_68', 'column_69',↙
'column_70', 'column_71', 'column_72', 'column_73', 'column_74', 'column_75',↙
'column_76', 'column_77', 'column_78', 'column_79', 'column_80', 'column_81',↙
'column_82', 'column_83', 'column_84', 'column_85', 'column_86', 'column_87',↙
'column_88', 'column_89', 'column_90', 'column_91', 'column_92', 'column_93',↙
'column_94', 'column_95', 'column_96', 'column_97', 'column_98', 'column_99',↙
'column_100', 'column_101', 'column_102', 'column_103', 'column_104', 'column_105',↙
'column_106', 'column_107', 'column_108', 'column_109', 'column_110', 'column_111',↙
'column_112', 'column_113', 'column_114', 'column_115', 'column_116', 'column_117',↙
'column_118', 'column_119', 'column_120', 'column_121', 'column_122', 'column_123',↙
'column_124', 'column_125', 'column_126', 'column_127', 'column_128', 'column_129',↙
'column_130', 'column_131', 'column_132', 'column_133', 'column_134', 'column_135',↙
'column_136', 'column_137', 'column_138', 'column_139', 'column_140', 'column_141',↙
'column_142', 'column_143', 'column_144', 'column_145', 'column_146', 'column_147',↙
'column_148', 'column_149', 'column_150', 'column_151', 'column_152', 'column_153',↙
'column_154', 'column_155', 'column_156', 'column_157', 'column_158', 'column_159',↙
'column_160', 'column_161', 'column_162', 'column_163', 'column_164', 'column_165',↙
'column_166', 'column_167', 'column_168', 'column_169', 'column_170', 'column_171',↙
'column_172', 'column_173', 'column_174', 'column_175', 'column_176', 'column_177',↙
'column_178', 'column_179', 'column_180', 'column_181', 'column_182', 'column_183',↙
'column_184', 'column_185', 'column_186', 'column_187', 'column_188', 'column_189',↙
'column_190', 'column_191', 'column_192', 'column_193', 'column_194', 'column_195',↙
'column_196', 'column_197', 'column_198', 'column_199', 'column_200', 'column_201',↙
'column_202', 'column_203', 'column_204', 'column_205', 'column_206', 'column_207',↙
'column_208', 'column_209', 'column_210', 'column_211', 'column_212', 'column_213',↙
'column_214', 'column_215', 'column_216', 'column_217', 'column_218', 'column_219',↙
'column_220', 'column_221', 'column_222', 'column_223', 'column_224', 'column_225',↙
'column_226', 'column_227', 'column_228', 'column_229', 'column_230', 'column_231',↙
'column_232', 'column_233', 'column_234', 'column_235', 'column_236', 'column_237',↙
'column_238', 'column_239', 'column_240', 'column_241', 'column_242', 'column_243',↙
'column_244', 'column_245', 'column_246', 'column_247', 'column_248', 'column_249',↙
'column_250', 'column_251', 'column_252', 'column_253', 'column_254', 'column_255',↙
'column_256', 'column_257', 'column_258', 'column_259', 'column_260', 'column_261',↙
'column_262', 'column_263', 'column_264', 'column_265', 'column_266', 'column_267',↙
'column_268', 'column_269', 'column_270', 'column_271', 'column_272', 'column_273',↙
'column_274', 'column_275', 'column_276', 'column_277', 'column_278', 'column_279',↙
'column_280', 'column_281', 'column_282', 'column_283', 'column_284', 'column_285',↙
'column_286', 'column_287', 'column_288', 'column_289', 'column_290', 'column_291',↙
'column_292', 'column_293', 'column_294', 'column_295', 'column_296', 'column_297',↙
```

```
'column_298', 'column_299', 'column_300', 'column_301', 'column_302', 'column_303',↙
'column_304', 'column_305', 'column_306', 'column_307', 'column_308', 'column_309',↙
'column_310', 'column_311', 'column_312', 'column_313', 'column_314', 'column_315',↙
'column_316', 'column_317', 'column_318', 'column_319', 'column_320', 'column_321',↙
'column_322', 'column_323', 'column_324', 'column_325', 'column_326', 'column_327',↙
'column_328', 'column_329', 'column_330', 'column_331', 'column_332', 'column_333',↙
'column_334', 'column_335', 'column_336', 'column_337', 'column_338', 'column_339',↙
'column_340', 'column_341', 'column_342', 'column_343', 'column_344', 'column_345',↙
'column_346', 'column_347', 'column_348', 'column_349', 'column_350', 'column_351',↙
'column_352', 'column_353', 'column_354', 'column_355', 'column_356', 'column_357',↙
'column_358', 'column_359', 'column_360', 'column_361', 'column_362', 'column_363',↙
'column_364', 'column_365', 'column_366', 'column_367', 'column_368', 'column_369',↙
'column_370', 'column_371', 'column_372', 'column_373', 'column_374', 'column_375',↙
'column_376', 'column_377', 'column_378', 'column_379', 'column_380', 'column_381',↙
'column_382', 'column_383', 'column_384', 'column_385', 'column_386', 'column_387',↙
'column_388', 'column_389', 'column_390', 'column_391', 'column_392', 'column_393',↙
'column_394', 'column_395', 'column_396', 'column_397', 'column_398', 'column_399',↙
'column_400', 'column_401', 'column_402', 'column_403', 'column_404', 'column_405',↙
'column_406', 'column_407', 'column_408', 'column_409', 'column_410', 'column_411',↙
'column_412', 'column_413', 'column_414', 'column_415', 'column_416', 'column_417',↙
'column_418', 'column_419', 'column_420', 'column_421', 'column_422', 'column_423',↙
'column_424', 'column_425', 'column_426', 'column_427', 'column_428', 'column_429',↙
'column_430', 'column_431', 'column_432', 'column_433', 'column_434', 'column_435',↙
'column_436', 'column_437', 'column_438', 'column_439', 'column_440', 'column_441',↙
'column_442', 'column_443', 'column_444', 'column_445', 'column_446', 'column_447',↙
'column_448', 'column_449', 'column_450', 'column_451', 'column_452', 'column_453',↙
'column_454', 'column_455', 'column_456', 'column_457', 'column_458', 'column_459',↙
'column_460', 'column_461', 'column_462', 'column_463', 'column_464', 'column_465',↙
'column_466', 'column_467', 'column_468', 'column_469', 'column_470', 'column_471',↙
'column_472', 'column_473', 'column_474', 'column_475', 'column_476', 'column_477',↙
'column_478', 'column_479', 'column_480', 'column_481', 'column_482', 'column_483',↙
'column_484', 'column_485', 'column_486', 'column_487', 'column_488', 'column_489',↙
'column_490', 'column_491', 'column_492', 'column_493', 'column_494', 'column_495',↙
'column_496', 'column_497', 'column_498', 'column_499', 'column_500', 'column_501',↙
'column_502', 'column_503', 'column_504', 'column_505', 'column_506', 'column_507',↙
'column_508', 'column_509', 'column_510', 'column_511', 'column_512', 'column_513',↙
'column_514', 'column_515', 'column_516', 'column_517', 'column_518', 'column_519',↙
'column_520', 'column_521', 'column_522', 'column_523', 'column_524', 'column_525',↙
'column_526', 'column_527', 'column_528', 'column_529', 'column_530', 'column_531',↙
'column_532', 'column_533', 'column_534', 'column_535', 'column_536', 'column_537',↙
'column_538', 'column_539', 'column_540', 'column_541', 'column_542', 'column_543',↙
'column_544', 'column_545', 'column_546', 'column_547', 'column_548', 'column_549',↙
'column_550', 'column_551', 'column_552', 'column_553', 'column_554', 'column_555',↙
'column_556', 'column_557', 'column_558', 'column_559', 'column_560', 'column_561',↙
'column_562', 'column_563', 'column_564', 'column_565', 'column_566', 'column_567',↙
'column_568', 'column_569', 'column_570', 'column_571', 'column_572', 'column_573',↙
'column_574', 'column_575', 'column_576', 'column_577', 'column_578', 'column_579',↙
'column_580', 'column_581', 'column_582', 'column_583', 'column_584', 'column_585',↙
'column_586', 'column_587', 'column_588', 'column_589', 'column_590', 'column_591',↙
'column_592', 'column_593', 'column_594', 'column_595', 'column_596', 'column_597',↙
'column_598', 'column_599', 'column_600', 'column_601', 'column_602', 'column_603',↙
```

```
'column_604', 'column_605', 'column_606', 'column_607', 'column_608', 'column_609',↙
'column_610', 'column_611', 'column_612', 'column_613', 'column_614', 'column_615',↙
'column_616', 'column_617', 'column_618', 'column_619', 'column_620', 'column_621',↙
'column_622', 'column_623', 'column_624', 'column_625', 'column_626', 'column_627',↙
'column_628', 'column_629', 'column_630', 'column_631', 'column_632', 'column_633',↙
'column_634', 'column_635', 'column_636', 'column_637', 'column_638', 'column_639',↙
'column_640', 'column_641', 'column_642', 'column_643', 'column_644', 'column_645',↙
'column_646', 'column_647', 'column_648', 'column_649', 'column_650', 'column_651',↙
'column_652', 'column_653', 'column_654', 'column_655', 'column_656', 'column_657',↙
'column_658', 'column_659', 'column_660', 'column_661', 'column_662', 'column_663',↙
'column_664', 'column_665', 'column_666', 'column_667', 'column_668', 'column_669',↙
'column_670', 'column_671', 'column_672', 'column_673', 'column_674', 'column_675',↙
'column_676', 'column_677', 'column_678', 'column_679', 'column_680', 'column_681',↙
'column_682', 'column_683', 'column_684', 'column_685', 'column_686', 'column_687',↙
'column_688', 'column_689', 'column_690', 'column_691', 'column_692', 'column_693',↙
'column_694', 'column_695', 'column_696', 'column_697', 'column_698', 'column_699',↙
'column_700', 'column_701', 'column_702', 'column_703', 'column_704', 'column_705',↙
'column_706', 'column_707', 'column_708', 'column_709', 'column_710', 'column_711',↙
'column_712', 'column_713', 'column_714', 'column_715', 'column_716', 'column_717',↙
'column_718', 'column_719', 'column_720', 'column_721', 'column_722', 'column_723',↙
'column_724', 'column_725', 'column_726', 'column_727', 'column_728', 'column_729',↙
'column_730', 'column_731', 'column_732', 'column_733', 'column_734', 'column_735',↙
'column_736', 'column_737', 'column_738', 'column_739', 'column_740', 'column_741',↙
'column_742', 'column_743', 'column_744', 'column_745', 'column_746', 'column_747',↙
'column_748', 'column_749', 'column_750', 'column_751', 'column_752', 'column_753',↙
'column_754', 'column_755', 'column_756', 'column_757', 'column_758', 'column_759',↙
'column_760', 'column_761', 'column_762', 'column_763', 'column_764', 'column_765',↙
'column_766', 'column_767', 'column_768', 'column_769', 'column_770', 'column_771',↙
'column_772', 'column_773'});

predictorNames = {'column_1', 'column_2', 'column_3', 'column_4', 'column_5', 'column_6',↙
'column_7', 'column_8', 'column_9', 'column_10', 'column_11', 'column_12', 'column_13',↙
'column_14', 'column_15', 'column_16', 'column_17', 'column_18', 'column_19',↙
'column_20', 'column_21', 'column_22', 'column_23', 'column_24', 'column_25',↙
'column_26', 'column_27', 'column_28', 'column_29', 'column_30', 'column_31',↙
'column_32', 'column_33', 'column_34', 'column_35', 'column_36', 'column_37',↙
'column_38', 'column_39', 'column_40', 'column_41', 'column_42', 'column_43',↙
'column_44', 'column_45', 'column_46', 'column_47', 'column_48', 'column_49',↙
'column_50', 'column_51', 'column_52', 'column_53', 'column_54', 'column_55',↙
'column_56', 'column_57', 'column_58', 'column_59', 'column_60', 'column_61',↙
'column_62', 'column_63', 'column_64', 'column_65', 'column_66', 'column_67',↙
'column_68', 'column_69', 'column_70', 'column_71', 'column_72', 'column_73',↙
'column_74', 'column_75', 'column_76', 'column_77', 'column_78', 'column_79',↙
'column_80', 'column_81', 'column_82', 'column_83', 'column_84', 'column_85',↙
'column_86', 'column_87', 'column_88', 'column_89', 'column_90', 'column_91',↙
'column_92', 'column_93', 'column_94', 'column_95', 'column_96', 'column_97',↙
'column_98', 'column_99', 'column_100', 'column_101', 'column_102', 'column_103',↙
'column_104', 'column_105', 'column_106', 'column_107', 'column_108', 'column_109',↙
'column_110', 'column_111', 'column_112', 'column_113', 'column_114', 'column_115',↙
'column_116', 'column_117', 'column_118', 'column_119', 'column_120', 'column_121',↙
'column_122', 'column_123', 'column_124', 'column_125', 'column_126', 'column_127',↙
```

```
'column_128', 'column_129', 'column_130', 'column_131', 'column_132', 'column_133',↙
'column_134', 'column_135', 'column_136', 'column_137', 'column_138', 'column_139',↙
'column_140', 'column_141', 'column_142', 'column_143', 'column_144', 'column_145',↙
'column_146', 'column_147', 'column_148', 'column_149', 'column_150', 'column_151',↙
'column_152', 'column_153', 'column_154', 'column_155', 'column_156', 'column_157',↙
'column_158', 'column_159', 'column_160', 'column_161', 'column_162', 'column_163',↙
'column_164', 'column_165', 'column_166', 'column_167', 'column_168', 'column_169',↙
'column_170', 'column_171', 'column_172', 'column_173', 'column_174', 'column_175',↙
'column_176', 'column_177', 'column_178', 'column_179', 'column_180', 'column_181',↙
'column_182', 'column_183', 'column_184', 'column_185', 'column_186', 'column_187',↙
'column_188', 'column_189', 'column_190', 'column_191', 'column_192', 'column_193',↙
'column_194', 'column_195', 'column_196', 'column_197', 'column_198', 'column_199',↙
'column_200', 'column_201', 'column_202', 'column_203', 'column_204', 'column_205',↙
'column_206', 'column_207', 'column_208', 'column_209', 'column_210', 'column_211',↙
'column_212', 'column_213', 'column_214', 'column_215', 'column_216', 'column_217',↙
'column_218', 'column_219', 'column_220', 'column_221', 'column_222', 'column_223',↙
'column_224', 'column_225', 'column_226', 'column_227', 'column_228', 'column_229',↙
'column_230', 'column_231', 'column_232', 'column_233', 'column_234', 'column_235',↙
'column_236', 'column_237', 'column_238', 'column_239', 'column_240', 'column_241',↙
'column_242', 'column_243', 'column_244', 'column_245', 'column_246', 'column_247',↙
'column_248', 'column_249', 'column_250', 'column_251', 'column_252', 'column_253',↙
'column_254', 'column_255', 'column_256', 'column_257', 'column_258', 'column_259',↙
'column_260', 'column_261', 'column_262', 'column_263', 'column_264', 'column_265',↙
'column_266', 'column_267', 'column_268', 'column_269', 'column_270', 'column_271',↙
'column_272', 'column_273', 'column_274', 'column_275', 'column_276', 'column_277',↙
'column_278', 'column_279', 'column_280', 'column_281', 'column_282', 'column_283',↙
'column_284', 'column_285', 'column_286', 'column_287', 'column_288', 'column_289',↙
'column_290', 'column_291', 'column_292', 'column_293', 'column_294', 'column_295',↙
'column_296', 'column_297', 'column_298', 'column_299', 'column_300', 'column_301',↙
'column_302', 'column_303', 'column_304', 'column_305', 'column_306', 'column_307',↙
'column_308', 'column_309', 'column_310', 'column_311', 'column_312', 'column_313',↙
'column_314', 'column_315', 'column_316', 'column_317', 'column_318', 'column_319',↙
'column_320', 'column_321', 'column_322', 'column_323', 'column_324', 'column_325',↙
'column_326', 'column_327', 'column_328', 'column_329', 'column_330', 'column_331',↙
'column_332', 'column_333', 'column_334', 'column_335', 'column_336', 'column_337',↙
'column_338', 'column_339', 'column_340', 'column_341', 'column_342', 'column_343',↙
'column_344', 'column_345', 'column_346', 'column_347', 'column_348', 'column_349',↙
'column_350', 'column_351', 'column_352', 'column_353', 'column_354', 'column_355',↙
'column_356', 'column_357', 'column_358', 'column_359', 'column_360', 'column_361',↙
'column_362', 'column_363', 'column_364', 'column_365', 'column_366', 'column_367',↙
'column_368', 'column_369', 'column_370', 'column_371', 'column_372', 'column_373',↙
'column_374', 'column_375', 'column_376', 'column_377', 'column_378', 'column_379',↙
'column_380', 'column_381', 'column_382', 'column_383', 'column_384', 'column_385',↙
'column_386', 'column_387', 'column_388', 'column_389', 'column_390', 'column_391',↙
'column_392', 'column_393', 'column_394', 'column_395', 'column_396', 'column_397',↙
'column_398', 'column_399', 'column_400', 'column_401', 'column_402', 'column_403',↙
'column_404', 'column_405', 'column_406', 'column_407', 'column_408', 'column_409',↙
'column_410', 'column_411', 'column_412', 'column_413', 'column_414', 'column_415',↙
'column_416', 'column_417', 'column_418', 'column_419', 'column_420', 'column_421',↙
'column_422', 'column_423', 'column_424', 'column_425', 'column_426', 'column_427',↙
'column_428', 'column_429', 'column_430', 'column_431', 'column_432', 'column_433',↙
```

```
'column_434', 'column_435', 'column_436', 'column_437', 'column_438', 'column_439',↙
'column_440', 'column_441', 'column_442', 'column_443', 'column_444', 'column_445',↙
'column_446', 'column_447', 'column_448', 'column_449', 'column_450', 'column_451',↙
'column_452', 'column_453', 'column_454', 'column_455', 'column_456', 'column_457',↙
'column_458', 'column_459', 'column_460', 'column_461', 'column_462', 'column_463',↙
'column_464', 'column_465', 'column_466', 'column_467', 'column_468', 'column_469',↙
'column_470', 'column_471', 'column_472', 'column_473', 'column_474', 'column_475',↙
'column_476', 'column_477', 'column_478', 'column_479', 'column_480', 'column_481',↙
'column_482', 'column_483', 'column_484', 'column_485', 'column_486', 'column_487',↙
'column_488', 'column_489', 'column_490', 'column_491', 'column_492', 'column_493',↙
'column_494', 'column_495', 'column_496', 'column_497', 'column_498', 'column_499',↙
'column_500', 'column_501', 'column_502', 'column_503', 'column_504', 'column_505',↙
'column_506', 'column_507', 'column_508', 'column_509', 'column_510', 'column_511',↙
'column_512', 'column_513', 'column_514', 'column_515', 'column_516', 'column_517',↙
'column_518', 'column_519', 'column_520', 'column_521', 'column_522', 'column_523',↙
'column_524', 'column_525', 'column_526', 'column_527', 'column_528', 'column_529',↙
'column_530', 'column_531', 'column_532', 'column_533', 'column_534', 'column_535',↙
'column_536', 'column_537', 'column_538', 'column_539', 'column_540', 'column_541',↙
'column_542', 'column_543', 'column_544', 'column_545', 'column_546', 'column_547',↙
'column_548', 'column_549', 'column_550', 'column_551', 'column_552', 'column_553',↙
'column_554', 'column_555', 'column_556', 'column_557', 'column_558', 'column_559',↙
'column_560', 'column_561', 'column_562', 'column_563', 'column_564', 'column_565',↙
'column_566', 'column_567', 'column_568', 'column_569', 'column_570', 'column_571',↙
'column_572', 'column_573', 'column_574', 'column_575', 'column_576', 'column_577',↙
'column_578', 'column_579', 'column_580', 'column_581', 'column_582', 'column_583',↙
'column_584', 'column_585', 'column_586', 'column_587', 'column_588', 'column_589',↙
'column_590', 'column_591', 'column_592', 'column_593', 'column_594', 'column_595',↙
'column_596', 'column_597', 'column_598', 'column_599', 'column_600', 'column_601',↙
'column_602', 'column_603', 'column_604', 'column_605', 'column_606', 'column_607',↙
'column_608', 'column_609', 'column_610', 'column_611', 'column_612', 'column_613',↙
'column_614', 'column_615', 'column_616', 'column_617', 'column_618', 'column_619',↙
'column_620', 'column_621', 'column_622', 'column_623', 'column_624', 'column_625',↙
'column_626', 'column_627', 'column_628', 'column_629', 'column_630', 'column_631',↙
'column_632', 'column_633', 'column_634', 'column_635', 'column_636', 'column_637',↙
'column_638', 'column_639', 'column_640', 'column_641', 'column_642', 'column_643',↙
'column_644', 'column_645', 'column_646', 'column_647', 'column_648', 'column_649',↙
'column_650', 'column_651', 'column_652', 'column_653', 'column_654', 'column_655',↙
'column_656', 'column_657', 'column_658', 'column_659', 'column_660', 'column_661',↙
'column_662', 'column_663', 'column_664', 'column_665', 'column_666', 'column_667',↙
'column_668', 'column_669', 'column_670', 'column_671', 'column_672', 'column_673',↙
'column_674', 'column_675', 'column_676', 'column_677', 'column_678', 'column_679',↙
'column_680', 'column_681', 'column_682', 'column_683', 'column_684', 'column_685',↙
'column_686', 'column_687', 'column_688', 'column_689', 'column_690', 'column_691',↙
'column_692', 'column_693', 'column_694', 'column_695', 'column_696', 'column_697',↙
'column_698', 'column_699', 'column_700', 'column_701', 'column_702', 'column_703',↙
'column_704', 'column_705', 'column_706', 'column_707', 'column_708', 'column_709',↙
'column_710', 'column_711', 'column_712', 'column_713', 'column_714', 'column_715',↙
'column_716', 'column_717', 'column_718', 'column_719', 'column_720', 'column_721',↙
'column_722', 'column_723', 'column_724', 'column_725', 'column_726', 'column_727',↙
'column_728', 'column_729', 'column_730', 'column_731', 'column_732', 'column_733',↙
'column_734', 'column_735', 'column_736', 'column_737', 'column_738', 'column_739',↙
```

```
'column_740', 'column_741', 'column_742', 'column_743', 'column_744', 'column_745',↙
'column_746', 'column_747', 'column_748', 'column_749', 'column_750', 'column_751',↙
'column_752', 'column_753', 'column_754', 'column_755', 'column_756', 'column_757',↙
'column_758', 'column_759', 'column_760', 'column_761', 'column_762', 'column_763',↙
'column_764', 'column_765', 'column_766', 'column_767', 'column_768', 'column_769',↙
'column_770', 'column_771', 'column_772', 'column_773'};
predictors = inputTable(:, predictorNames);
response = responseData;
isCategoricalPredictor = [false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
```

```
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
classificationKNN = fitcknn(...
    predictors, ...
    response, ...
    'Distance', 'Euclidean', ...
    'Exponent', [], ...
    'NumNeighbors', 1, ...
    'DistanceWeight', 'Equal', ...
    'Standardize', true, ...
    'ClassNames', {'Black Sea Sprat'; 'Gilt-Head Breamt'; 'Hourse Mackerelt'; 'Red↙
Mullet'; 'Red Sea Bream'; 'Sea Bass'; 'Striped Red Mullet'; 'Trout'});

% Create the result struct with predict function
predictorExtractionFcn = @(x) array2table(x, 'VariableNames', predictorNames);
knnPredictFcn = @(x) predict(classificationKNN, x);
trainedClassifier.predictFcn = @(x) knnPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.ClassificationKNN = classificationKNN;
trainedClassifier.About = 'This struct is a trained model exported from Classification↙
Learner R2021a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new predictor column↙
matrix, X, use: \n  yfit = c.predictFcn(X) \nreplacing ''c'' with the name of the↙
variable that is this struct, e.g. ''trainedModel''. \n \nX must contain exactly 773↙
columns because this model was trained using 773 predictors. \nX must contain only↙
predictor columns in exactly the same order and format as your training \ndata. Do not↙
include the response column or any columns you did not import into the app. \n \nFor more↙
```

```
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'', ''stats.map''),↙
''appclassification_exportmodeltoworkspace'')">How to predict using an exported↙
model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
% Convert input to table
inputTable = array2table(trainingData, 'VariableNames', {'column_1', 'column_2',↙
'column_3', 'column_4', 'column_5', 'column_6', 'column_7', 'column_8', 'column_9',↙
'column_10', 'column_11', 'column_12', 'column_13', 'column_14', 'column_15',↙
'column_16', 'column_17', 'column_18', 'column_19', 'column_20', 'column_21',↙
'column_22', 'column_23', 'column_24', 'column_25', 'column_26', 'column_27',↙
'column_28', 'column_29', 'column_30', 'column_31', 'column_32', 'column_33',↙
'column_34', 'column_35', 'column_36', 'column_37', 'column_38', 'column_39',↙
'column_40', 'column_41', 'column_42', 'column_43', 'column_44', 'column_45',↙
'column_46', 'column_47', 'column_48', 'column_49', 'column_50', 'column_51',↙
'column_52', 'column_53', 'column_54', 'column_55', 'column_56', 'column_57',↙
'column_58', 'column_59', 'column_60', 'column_61', 'column_62', 'column_63',↙
'column_64', 'column_65', 'column_66', 'column_67', 'column_68', 'column_69',↙
'column_70', 'column_71', 'column_72', 'column_73', 'column_74', 'column_75',↙
'column_76', 'column_77', 'column_78', 'column_79', 'column_80', 'column_81',↙
'column_82', 'column_83', 'column_84', 'column_85', 'column_86', 'column_87',↙
'column_88', 'column_89', 'column_90', 'column_91', 'column_92', 'column_93',↙
'column_94', 'column_95', 'column_96', 'column_97', 'column_98', 'column_99',↙
'column_100', 'column_101', 'column_102', 'column_103', 'column_104', 'column_105',↙
'column_106', 'column_107', 'column_108', 'column_109', 'column_110', 'column_111',↙
'column_112', 'column_113', 'column_114', 'column_115', 'column_116', 'column_117',↙
'column_118', 'column_119', 'column_120', 'column_121', 'column_122', 'column_123',↙
'column_124', 'column_125', 'column_126', 'column_127', 'column_128', 'column_129',↙
'column_130', 'column_131', 'column_132', 'column_133', 'column_134', 'column_135',↙
'column_136', 'column_137', 'column_138', 'column_139', 'column_140', 'column_141',↙
'column_142', 'column_143', 'column_144', 'column_145', 'column_146', 'column_147',↙
'column_148', 'column_149', 'column_150', 'column_151', 'column_152', 'column_153',↙
'column_154', 'column_155', 'column_156', 'column_157', 'column_158', 'column_159',↙
'column_160', 'column_161', 'column_162', 'column_163', 'column_164', 'column_165',↙
'column_166', 'column_167', 'column_168', 'column_169', 'column_170', 'column_171',↙
'column_172', 'column_173', 'column_174', 'column_175', 'column_176', 'column_177',↙
'column_178', 'column_179', 'column_180', 'column_181', 'column_182', 'column_183',↙
'column_184', 'column_185', 'column_186', 'column_187', 'column_188', 'column_189',↙
'column_190', 'column_191', 'column_192', 'column_193', 'column_194', 'column_195',↙
'column_196', 'column_197', 'column_198', 'column_199', 'column_200', 'column_201',↙
'column_202', 'column_203', 'column_204', 'column_205', 'column_206', 'column_207',↙
'column_208', 'column_209', 'column_210', 'column_211', 'column_212', 'column_213',↙
'column_214', 'column_215', 'column_216', 'column_217', 'column_218', 'column_219',↙
'column_220', 'column_221', 'column_222', 'column_223', 'column_224', 'column_225',↙
'column_226', 'column_227', 'column_228', 'column_229', 'column_230', 'column_231',↙
'column_232', 'column_233', 'column_234', 'column_235', 'column_236', 'column_237',↙
'column_238', 'column_239', 'column_240', 'column_241', 'column_242', 'column_243',↙
'column_244', 'column_245', 'column_246', 'column_247', 'column_248', 'column_249',↙
'column_250', 'column_251', 'column_252', 'column_253', 'column_254', 'column_255',↙
```

```
'column_256', 'column_257', 'column_258', 'column_259', 'column_260', 'column_261',↙
'column_262', 'column_263', 'column_264', 'column_265', 'column_266', 'column_267',↙
'column_268', 'column_269', 'column_270', 'column_271', 'column_272', 'column_273',↙
'column_274', 'column_275', 'column_276', 'column_277', 'column_278', 'column_279',↙
'column_280', 'column_281', 'column_282', 'column_283', 'column_284', 'column_285',↙
'column_286', 'column_287', 'column_288', 'column_289', 'column_290', 'column_291',↙
'column_292', 'column_293', 'column_294', 'column_295', 'column_296', 'column_297',↙
'column_298', 'column_299', 'column_300', 'column_301', 'column_302', 'column_303',↙
'column_304', 'column_305', 'column_306', 'column_307', 'column_308', 'column_309',↙
'column_310', 'column_311', 'column_312', 'column_313', 'column_314', 'column_315',↙
'column_316', 'column_317', 'column_318', 'column_319', 'column_320', 'column_321',↙
'column_322', 'column_323', 'column_324', 'column_325', 'column_326', 'column_327',↙
'column_328', 'column_329', 'column_330', 'column_331', 'column_332', 'column_333',↙
'column_334', 'column_335', 'column_336', 'column_337', 'column_338', 'column_339',↙
'column_340', 'column_341', 'column_342', 'column_343', 'column_344', 'column_345',↙
'column_346', 'column_347', 'column_348', 'column_349', 'column_350', 'column_351',↙
'column_352', 'column_353', 'column_354', 'column_355', 'column_356', 'column_357',↙
'column_358', 'column_359', 'column_360', 'column_361', 'column_362', 'column_363',↙
'column_364', 'column_365', 'column_366', 'column_367', 'column_368', 'column_369',↙
'column_370', 'column_371', 'column_372', 'column_373', 'column_374', 'column_375',↙
'column_376', 'column_377', 'column_378', 'column_379', 'column_380', 'column_381',↙
'column_382', 'column_383', 'column_384', 'column_385', 'column_386', 'column_387',↙
'column_388', 'column_389', 'column_390', 'column_391', 'column_392', 'column_393',↙
'column_394', 'column_395', 'column_396', 'column_397', 'column_398', 'column_399',↙
'column_400', 'column_401', 'column_402', 'column_403', 'column_404', 'column_405',↙
'column_406', 'column_407', 'column_408', 'column_409', 'column_410', 'column_411',↙
'column_412', 'column_413', 'column_414', 'column_415', 'column_416', 'column_417',↙
'column_418', 'column_419', 'column_420', 'column_421', 'column_422', 'column_423',↙
'column_424', 'column_425', 'column_426', 'column_427', 'column_428', 'column_429',↙
'column_430', 'column_431', 'column_432', 'column_433', 'column_434', 'column_435',↙
'column_436', 'column_437', 'column_438', 'column_439', 'column_440', 'column_441',↙
'column_442', 'column_443', 'column_444', 'column_445', 'column_446', 'column_447',↙
'column_448', 'column_449', 'column_450', 'column_451', 'column_452', 'column_453',↙
'column_454', 'column_455', 'column_456', 'column_457', 'column_458', 'column_459',↙
'column_460', 'column_461', 'column_462', 'column_463', 'column_464', 'column_465',↙
'column_466', 'column_467', 'column_468', 'column_469', 'column_470', 'column_471',↙
'column_472', 'column_473', 'column_474', 'column_475', 'column_476', 'column_477',↙
'column_478', 'column_479', 'column_480', 'column_481', 'column_482', 'column_483',↙
'column_484', 'column_485', 'column_486', 'column_487', 'column_488', 'column_489',↙
'column_490', 'column_491', 'column_492', 'column_493', 'column_494', 'column_495',↙
'column_496', 'column_497', 'column_498', 'column_499', 'column_500', 'column_501',↙
'column_502', 'column_503', 'column_504', 'column_505', 'column_506', 'column_507',↙
'column_508', 'column_509', 'column_510', 'column_511', 'column_512', 'column_513',↙
'column_514', 'column_515', 'column_516', 'column_517', 'column_518', 'column_519',↙
'column_520', 'column_521', 'column_522', 'column_523', 'column_524', 'column_525',↙
'column_526', 'column_527', 'column_528', 'column_529', 'column_530', 'column_531',↙
'column_532', 'column_533', 'column_534', 'column_535', 'column_536', 'column_537',↙
'column_538', 'column_539', 'column_540', 'column_541', 'column_542', 'column_543',↙
'column_544', 'column_545', 'column_546', 'column_547', 'column_548', 'column_549',↙
'column_550', 'column_551', 'column_552', 'column_553', 'column_554', 'column_555',↙
'column_556', 'column_557', 'column_558', 'column_559', 'column_560', 'column_561',↙
```

```
'column_562', 'column_563', 'column_564', 'column_565', 'column_566', 'column_567',↙
'column_568', 'column_569', 'column_570', 'column_571', 'column_572', 'column_573',↙
'column_574', 'column_575', 'column_576', 'column_577', 'column_578', 'column_579',↙
'column_580', 'column_581', 'column_582', 'column_583', 'column_584', 'column_585',↙
'column_586', 'column_587', 'column_588', 'column_589', 'column_590', 'column_591',↙
'column_592', 'column_593', 'column_594', 'column_595', 'column_596', 'column_597',↙
'column_598', 'column_599', 'column_600', 'column_601', 'column_602', 'column_603',↙
'column_604', 'column_605', 'column_606', 'column_607', 'column_608', 'column_609',↙
'column_610', 'column_611', 'column_612', 'column_613', 'column_614', 'column_615',↙
'column_616', 'column_617', 'column_618', 'column_619', 'column_620', 'column_621',↙
'column_622', 'column_623', 'column_624', 'column_625', 'column_626', 'column_627',↙
'column_628', 'column_629', 'column_630', 'column_631', 'column_632', 'column_633',↙
'column_634', 'column_635', 'column_636', 'column_637', 'column_638', 'column_639',↙
'column_640', 'column_641', 'column_642', 'column_643', 'column_644', 'column_645',↙
'column_646', 'column_647', 'column_648', 'column_649', 'column_650', 'column_651',↙
'column_652', 'column_653', 'column_654', 'column_655', 'column_656', 'column_657',↙
'column_658', 'column_659', 'column_660', 'column_661', 'column_662', 'column_663',↙
'column_664', 'column_665', 'column_666', 'column_667', 'column_668', 'column_669',↙
'column_670', 'column_671', 'column_672', 'column_673', 'column_674', 'column_675',↙
'column_676', 'column_677', 'column_678', 'column_679', 'column_680', 'column_681',↙
'column_682', 'column_683', 'column_684', 'column_685', 'column_686', 'column_687',↙
'column_688', 'column_689', 'column_690', 'column_691', 'column_692', 'column_693',↙
'column_694', 'column_695', 'column_696', 'column_697', 'column_698', 'column_699',↙
'column_700', 'column_701', 'column_702', 'column_703', 'column_704', 'column_705',↙
'column_706', 'column_707', 'column_708', 'column_709', 'column_710', 'column_711',↙
'column_712', 'column_713', 'column_714', 'column_715', 'column_716', 'column_717',↙
'column_718', 'column_719', 'column_720', 'column_721', 'column_722', 'column_723',↙
'column_724', 'column_725', 'column_726', 'column_727', 'column_728', 'column_729',↙
'column_730', 'column_731', 'column_732', 'column_733', 'column_734', 'column_735',↙
'column_736', 'column_737', 'column_738', 'column_739', 'column_740', 'column_741',↙
'column_742', 'column_743', 'column_744', 'column_745', 'column_746', 'column_747',↙
'column_748', 'column_749', 'column_750', 'column_751', 'column_752', 'column_753',↙
'column_754', 'column_755', 'column_756', 'column_757', 'column_758', 'column_759',↙
'column_760', 'column_761', 'column_762', 'column_763', 'column_764', 'column_765',↙
'column_766', 'column_767', 'column_768', 'column_769', 'column_770', 'column_771',↙
'column_772', 'column_773'});

predictorNames = {'column_1', 'column_2', 'column_3', 'column_4', 'column_5', 'column_6',↙
'column_7', 'column_8', 'column_9', 'column_10', 'column_11', 'column_12', 'column_13',↙
'column_14', 'column_15', 'column_16', 'column_17', 'column_18', 'column_19',↙
'column_20', 'column_21', 'column_22', 'column_23', 'column_24', 'column_25',↙
'column_26', 'column_27', 'column_28', 'column_29', 'column_30', 'column_31',↙
'column_32', 'column_33', 'column_34', 'column_35', 'column_36', 'column_37',↙
'column_38', 'column_39', 'column_40', 'column_41', 'column_42', 'column_43',↙
'column_44', 'column_45', 'column_46', 'column_47', 'column_48', 'column_49',↙
'column_50', 'column_51', 'column_52', 'column_53', 'column_54', 'column_55',↙
'column_56', 'column_57', 'column_58', 'column_59', 'column_60', 'column_61',↙
'column_62', 'column_63', 'column_64', 'column_65', 'column_66', 'column_67',↙
'column_68', 'column_69', 'column_70', 'column_71', 'column_72', 'column_73',↙
'column_74', 'column_75', 'column_76', 'column_77', 'column_78', 'column_79',↙
'column_80', 'column_81', 'column_82', 'column_83', 'column_84', 'column_85',↙
```

```
'column_86', 'column_87', 'column_88', 'column_89', 'column_90', 'column_91',↙
'column_92', 'column_93', 'column_94', 'column_95', 'column_96', 'column_97',↙
'column_98', 'column_99', 'column_100', 'column_101', 'column_102', 'column_103',↙
'column_104', 'column_105', 'column_106', 'column_107', 'column_108', 'column_109',↙
'column_110', 'column_111', 'column_112', 'column_113', 'column_114', 'column_115',↙
'column_116', 'column_117', 'column_118', 'column_119', 'column_120', 'column_121',↙
'column_122', 'column_123', 'column_124', 'column_125', 'column_126', 'column_127',↙
'column_128', 'column_129', 'column_130', 'column_131', 'column_132', 'column_133',↙
'column_134', 'column_135', 'column_136', 'column_137', 'column_138', 'column_139',↙
'column_140', 'column_141', 'column_142', 'column_143', 'column_144', 'column_145',↙
'column_146', 'column_147', 'column_148', 'column_149', 'column_150', 'column_151',↙
'column_152', 'column_153', 'column_154', 'column_155', 'column_156', 'column_157',↙
'column_158', 'column_159', 'column_160', 'column_161', 'column_162', 'column_163',↙
'column_164', 'column_165', 'column_166', 'column_167', 'column_168', 'column_169',↙
'column_170', 'column_171', 'column_172', 'column_173', 'column_174', 'column_175',↙
'column_176', 'column_177', 'column_178', 'column_179', 'column_180', 'column_181',↙
'column_182', 'column_183', 'column_184', 'column_185', 'column_186', 'column_187',↙
'column_188', 'column_189', 'column_190', 'column_191', 'column_192', 'column_193',↙
'column_194', 'column_195', 'column_196', 'column_197', 'column_198', 'column_199',↙
'column_200', 'column_201', 'column_202', 'column_203', 'column_204', 'column_205',↙
'column_206', 'column_207', 'column_208', 'column_209', 'column_210', 'column_211',↙
'column_212', 'column_213', 'column_214', 'column_215', 'column_216', 'column_217',↙
'column_218', 'column_219', 'column_220', 'column_221', 'column_222', 'column_223',↙
'column_224', 'column_225', 'column_226', 'column_227', 'column_228', 'column_229',↙
'column_230', 'column_231', 'column_232', 'column_233', 'column_234', 'column_235',↙
'column_236', 'column_237', 'column_238', 'column_239', 'column_240', 'column_241',↙
'column_242', 'column_243', 'column_244', 'column_245', 'column_246', 'column_247',↙
'column_248', 'column_249', 'column_250', 'column_251', 'column_252', 'column_253',↙
'column_254', 'column_255', 'column_256', 'column_257', 'column_258', 'column_259',↙
'column_260', 'column_261', 'column_262', 'column_263', 'column_264', 'column_265',↙
'column_266', 'column_267', 'column_268', 'column_269', 'column_270', 'column_271',↙
'column_272', 'column_273', 'column_274', 'column_275', 'column_276', 'column_277',↙
'column_278', 'column_279', 'column_280', 'column_281', 'column_282', 'column_283',↙
'column_284', 'column_285', 'column_286', 'column_287', 'column_288', 'column_289',↙
'column_290', 'column_291', 'column_292', 'column_293', 'column_294', 'column_295',↙
'column_296', 'column_297', 'column_298', 'column_299', 'column_300', 'column_301',↙
'column_302', 'column_303', 'column_304', 'column_305', 'column_306', 'column_307',↙
'column_308', 'column_309', 'column_310', 'column_311', 'column_312', 'column_313',↙
'column_314', 'column_315', 'column_316', 'column_317', 'column_318', 'column_319',↙
'column_320', 'column_321', 'column_322', 'column_323', 'column_324', 'column_325',↙
'column_326', 'column_327', 'column_328', 'column_329', 'column_330', 'column_331',↙
'column_332', 'column_333', 'column_334', 'column_335', 'column_336', 'column_337',↙
'column_338', 'column_339', 'column_340', 'column_341', 'column_342', 'column_343',↙
'column_344', 'column_345', 'column_346', 'column_347', 'column_348', 'column_349',↙
'column_350', 'column_351', 'column_352', 'column_353', 'column_354', 'column_355',↙
'column_356', 'column_357', 'column_358', 'column_359', 'column_360', 'column_361',↙
'column_362', 'column_363', 'column_364', 'column_365', 'column_366', 'column_367',↙
'column_368', 'column_369', 'column_370', 'column_371', 'column_372', 'column_373',↙
'column_374', 'column_375', 'column_376', 'column_377', 'column_378', 'column_379',↙
'column_380', 'column_381', 'column_382', 'column_383', 'column_384', 'column_385',↙
'column_386', 'column_387', 'column_388', 'column_389', 'column_390', 'column_391',↙
```

```
'column_392', 'column_393', 'column_394', 'column_395', 'column_396', 'column_397',↙
'column_398', 'column_399', 'column_400', 'column_401', 'column_402', 'column_403',↙
'column_404', 'column_405', 'column_406', 'column_407', 'column_408', 'column_409',↙
'column_410', 'column_411', 'column_412', 'column_413', 'column_414', 'column_415',↙
'column_416', 'column_417', 'column_418', 'column_419', 'column_420', 'column_421',↙
'column_422', 'column_423', 'column_424', 'column_425', 'column_426', 'column_427',↙
'column_428', 'column_429', 'column_430', 'column_431', 'column_432', 'column_433',↙
'column_434', 'column_435', 'column_436', 'column_437', 'column_438', 'column_439',↙
'column_440', 'column_441', 'column_442', 'column_443', 'column_444', 'column_445',↙
'column_446', 'column_447', 'column_448', 'column_449', 'column_450', 'column_451',↙
'column_452', 'column_453', 'column_454', 'column_455', 'column_456', 'column_457',↙
'column_458', 'column_459', 'column_460', 'column_461', 'column_462', 'column_463',↙
'column_464', 'column_465', 'column_466', 'column_467', 'column_468', 'column_469',↙
'column_470', 'column_471', 'column_472', 'column_473', 'column_474', 'column_475',↙
'column_476', 'column_477', 'column_478', 'column_479', 'column_480', 'column_481',↙
'column_482', 'column_483', 'column_484', 'column_485', 'column_486', 'column_487',↙
'column_488', 'column_489', 'column_490', 'column_491', 'column_492', 'column_493',↙
'column_494', 'column_495', 'column_496', 'column_497', 'column_498', 'column_499',↙
'column_500', 'column_501', 'column_502', 'column_503', 'column_504', 'column_505',↙
'column_506', 'column_507', 'column_508', 'column_509', 'column_510', 'column_511',↙
'column_512', 'column_513', 'column_514', 'column_515', 'column_516', 'column_517',↙
'column_518', 'column_519', 'column_520', 'column_521', 'column_522', 'column_523',↙
'column_524', 'column_525', 'column_526', 'column_527', 'column_528', 'column_529',↙
'column_530', 'column_531', 'column_532', 'column_533', 'column_534', 'column_535',↙
'column_536', 'column_537', 'column_538', 'column_539', 'column_540', 'column_541',↙
'column_542', 'column_543', 'column_544', 'column_545', 'column_546', 'column_547',↙
'column_548', 'column_549', 'column_550', 'column_551', 'column_552', 'column_553',↙
'column_554', 'column_555', 'column_556', 'column_557', 'column_558', 'column_559',↙
'column_560', 'column_561', 'column_562', 'column_563', 'column_564', 'column_565',↙
'column_566', 'column_567', 'column_568', 'column_569', 'column_570', 'column_571',↙
'column_572', 'column_573', 'column_574', 'column_575', 'column_576', 'column_577',↙
'column_578', 'column_579', 'column_580', 'column_581', 'column_582', 'column_583',↙
'column_584', 'column_585', 'column_586', 'column_587', 'column_588', 'column_589',↙
'column_590', 'column_591', 'column_592', 'column_593', 'column_594', 'column_595',↙
'column_596', 'column_597', 'column_598', 'column_599', 'column_600', 'column_601',↙
'column_602', 'column_603', 'column_604', 'column_605', 'column_606', 'column_607',↙
'column_608', 'column_609', 'column_610', 'column_611', 'column_612', 'column_613',↙
'column_614', 'column_615', 'column_616', 'column_617', 'column_618', 'column_619',↙
'column_620', 'column_621', 'column_622', 'column_623', 'column_624', 'column_625',↙
'column_626', 'column_627', 'column_628', 'column_629', 'column_630', 'column_631',↙
'column_632', 'column_633', 'column_634', 'column_635', 'column_636', 'column_637',↙
'column_638', 'column_639', 'column_640', 'column_641', 'column_642', 'column_643',↙
'column_644', 'column_645', 'column_646', 'column_647', 'column_648', 'column_649',↙
'column_650', 'column_651', 'column_652', 'column_653', 'column_654', 'column_655',↙
'column_656', 'column_657', 'column_658', 'column_659', 'column_660', 'column_661',↙
'column_662', 'column_663', 'column_664', 'column_665', 'column_666', 'column_667',↙
'column_668', 'column_669', 'column_670', 'column_671', 'column_672', 'column_673',↙
'column_674', 'column_675', 'column_676', 'column_677', 'column_678', 'column_679',↙
'column_680', 'column_681', 'column_682', 'column_683', 'column_684', 'column_685',↙
'column_686', 'column_687', 'column_688', 'column_689', 'column_690', 'column_691',↙
'column_692', 'column_693', 'column_694', 'column_695', 'column_696', 'column_697',↙
```

```
'column_698', 'column_699', 'column_700', 'column_701', 'column_702', 'column_703',↙
'column_704', 'column_705', 'column_706', 'column_707', 'column_708', 'column_709',↙
'column_710', 'column_711', 'column_712', 'column_713', 'column_714', 'column_715',↙
'column_716', 'column_717', 'column_718', 'column_719', 'column_720', 'column_721',↙
'column_722', 'column_723', 'column_724', 'column_725', 'column_726', 'column_727',↙
'column_728', 'column_729', 'column_730', 'column_731', 'column_732', 'column_733',↙
'column_734', 'column_735', 'column_736', 'column_737', 'column_738', 'column_739',↙
'column_740', 'column_741', 'column_742', 'column_743', 'column_744', 'column_745',↙
'column_746', 'column_747', 'column_748', 'column_749', 'column_750', 'column_751',↙
'column_752', 'column_753', 'column_754', 'column_755', 'column_756', 'column_757',↙
'column_758', 'column_759', 'column_760', 'column_761', 'column_762', 'column_763',↙
'column_764', 'column_765', 'column_766', 'column_767', 'column_768', 'column_769',↙
'column_770', 'column_771', 'column_772', 'column_773'};
predictors = inputTable(:, predictorNames);
response = responseData;
isCategoricalPredictor = [false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
```

```
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false];

% Set up holdout validation
cvp = cvpartition(response, 'Holdout', 0.1);
trainingPredictors = predictors(cvp.training, :);
trainingResponse = response(cvp.training, :);
trainingIsCategoricalPredictor = isCategoricalPredictor;

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
classificationKNN = fitcknn(...
    trainingPredictors, ...
    trainingResponse, ...
    'Distance', 'Euclidean', ...
    'Exponent', [], ...
    'NumNeighbors', 1, ...
    'DistanceWeight', 'Equal', ...
    'Standardize', true, ...
    'ClassNames', {'Black Sea Sprat'; 'Gilt-Head Breamt'; 'Hourse Mackerelt'; 'Red↙
Mullet'; 'Red Sea Bream'; 'Sea Bass'; 'Striped Red Mullet'; 'Trout'});

% Create the result struct with predict function
knnPredictFcn = @(x) predict(classificationKNN, x);
```

```
validationPredictFcn = @(x) knnPredictFcn(x);

% Add additional fields to the result struct


% Compute validation predictions
validationPredictors = predictors(cvp.test, :);
validationResponse = response(cvp.test, :);
[validationPredictions, validationScores] = validationPredictFcn(validationPredictors);

% Compute validation accuracy
correctPredictions = strcmp( strtrim(validationPredictions), strtrim↙
(validationResponse));
isMissing = cellfun(@(x) all(isspace(x)), validationResponse, 'UniformOutput', true);
correctPredictions = correctPredictions(~isMissing);
validationAccuracy = sum(correctPredictions)/length(correctPredictions);
```

```
clear all
clc

load('train_data.mat'); % 80% training 10% validation
load('train_response.mat');

load('test_data.mat'); % 10% from fish dataset
load('test_response.mat');

N = length(test_response);


[trainKNNClassifier, KNNvalidationAccuracy] = trainClassifier(train_data, train_response)↙
%retrain

disp('Using testing dataset to predict fish type');
yfit = trainKNNClassifier.predictFcn(test_data);  %test

disp('[PREDICTED FISH TYPE]    [ACTUAL FISH TYPE]');
true_total=0;
for k=1:N
    output = strcat('[',yfit{k},']       [',test_response{k},']');
    disp(output);
    if isequal(yfit{k}, test_response{k})
        true_total=true_total+1;
    end
end

testing_KNN_accuracy=true_total/N*100
```

```
function [trainedClassifier, validationAccuracy] = trainClassifier(trainingData,↙
responseData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData,
% responseData)
% Returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: A matrix with the same number of columns and data type
%       as the matrix imported into the app.
%
%      responseData: A vector with the same data type as the vector
%       imported into the app. The length of responseData and the number of
%       rows of trainingData must be equal.
%
%  Output:
%      trainedClassifier: A struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: A function to make predictions on new
%       data.
%
%      validationAccuracy: A double containing the accuracy in percent. In
%       the app, the History list displays this overall accuracy score for
%       each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input arguments trainingData and responseData.
%
% For example, to retrain a classifier trained with the original data set T
% and response Y, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T, Y)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a matrix containing only the predictor columns used for
% training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 28-May-2022 10:20:28


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
```

```
% Convert input to table
inputTable = array2table(trainingData, 'VariableNames', {'column_1', 'column_2',↙
'column_3', 'column_4', 'column_5', 'column_6', 'column_7', 'column_8', 'column_9',↙
'column_10', 'column_11', 'column_12', 'column_13', 'column_14', 'column_15',↙
'column_16', 'column_17', 'column_18', 'column_19', 'column_20', 'column_21',↙
'column_22', 'column_23', 'column_24', 'column_25', 'column_26', 'column_27',↙
'column_28', 'column_29', 'column_30', 'column_31', 'column_32', 'column_33',↙
'column_34', 'column_35', 'column_36', 'column_37', 'column_38', 'column_39',↙
'column_40', 'column_41', 'column_42', 'column_43', 'column_44', 'column_45',↙
'column_46', 'column_47', 'column_48', 'column_49', 'column_50', 'column_51',↙
'column_52', 'column_53', 'column_54', 'column_55', 'column_56', 'column_57',↙
'column_58', 'column_59', 'column_60', 'column_61', 'column_62', 'column_63',↙
'column_64', 'column_65', 'column_66', 'column_67', 'column_68', 'column_69',↙
'column_70', 'column_71', 'column_72', 'column_73', 'column_74', 'column_75',↙
'column_76', 'column_77', 'column_78', 'column_79', 'column_80', 'column_81',↙
'column_82', 'column_83', 'column_84', 'column_85', 'column_86', 'column_87',↙
'column_88', 'column_89', 'column_90', 'column_91', 'column_92', 'column_93',↙
'column_94', 'column_95', 'column_96', 'column_97', 'column_98', 'column_99',↙
'column_100', 'column_101', 'column_102', 'column_103', 'column_104', 'column_105',↙
'column_106', 'column_107', 'column_108', 'column_109', 'column_110', 'column_111',↙
'column_112', 'column_113', 'column_114', 'column_115', 'column_116', 'column_117',↙
'column_118', 'column_119', 'column_120', 'column_121', 'column_122', 'column_123',↙
'column_124', 'column_125', 'column_126', 'column_127', 'column_128', 'column_129',↙
'column_130', 'column_131', 'column_132', 'column_133', 'column_134', 'column_135',↙
'column_136', 'column_137', 'column_138', 'column_139', 'column_140', 'column_141',↙
'column_142', 'column_143', 'column_144', 'column_145', 'column_146', 'column_147',↙
'column_148', 'column_149', 'column_150', 'column_151', 'column_152', 'column_153',↙
'column_154', 'column_155', 'column_156', 'column_157', 'column_158', 'column_159',↙
'column_160', 'column_161', 'column_162', 'column_163', 'column_164', 'column_165',↙
'column_166', 'column_167', 'column_168', 'column_169', 'column_170', 'column_171',↙
'column_172', 'column_173', 'column_174', 'column_175', 'column_176', 'column_177',↙
'column_178', 'column_179', 'column_180', 'column_181', 'column_182', 'column_183',↙
'column_184', 'column_185', 'column_186', 'column_187', 'column_188', 'column_189',↙
'column_190', 'column_191', 'column_192', 'column_193', 'column_194', 'column_195',↙
'column_196', 'column_197', 'column_198', 'column_199', 'column_200', 'column_201',↙
'column_202', 'column_203', 'column_204', 'column_205', 'column_206', 'column_207',↙
'column_208', 'column_209', 'column_210', 'column_211', 'column_212', 'column_213',↙
'column_214', 'column_215', 'column_216', 'column_217', 'column_218', 'column_219',↙
'column_220', 'column_221', 'column_222', 'column_223', 'column_224', 'column_225',↙
'column_226', 'column_227', 'column_228', 'column_229', 'column_230', 'column_231',↙
'column_232', 'column_233', 'column_234', 'column_235', 'column_236', 'column_237',↙
'column_238', 'column_239', 'column_240', 'column_241', 'column_242', 'column_243',↙
'column_244', 'column_245', 'column_246', 'column_247', 'column_248', 'column_249',↙
'column_250', 'column_251', 'column_252', 'column_253', 'column_254', 'column_255',↙
'column_256', 'column_257', 'column_258', 'column_259', 'column_260', 'column_261',↙
'column_262', 'column_263', 'column_264', 'column_265', 'column_266', 'column_267',↙
'column_268', 'column_269', 'column_270', 'column_271', 'column_272', 'column_273',↙
'column_274', 'column_275', 'column_276', 'column_277', 'column_278', 'column_279',↙
'column_280', 'column_281', 'column_282', 'column_283', 'column_284', 'column_285',↙
'column_286', 'column_287', 'column_288', 'column_289', 'column_290', 'column_291',↙
'column_292', 'column_293', 'column_294', 'column_295', 'column_296', 'column_297',↙
```

```
'column_298', 'column_299', 'column_300', 'column_301', 'column_302', 'column_303',↙
'column_304', 'column_305', 'column_306', 'column_307', 'column_308', 'column_309',↙
'column_310', 'column_311', 'column_312', 'column_313', 'column_314', 'column_315',↙
'column_316', 'column_317', 'column_318', 'column_319', 'column_320', 'column_321',↙
'column_322', 'column_323', 'column_324', 'column_325', 'column_326', 'column_327',↙
'column_328', 'column_329', 'column_330', 'column_331', 'column_332', 'column_333',↙
'column_334', 'column_335', 'column_336', 'column_337', 'column_338', 'column_339',↙
'column_340', 'column_341', 'column_342', 'column_343', 'column_344', 'column_345',↙
'column_346', 'column_347', 'column_348', 'column_349', 'column_350', 'column_351',↙
'column_352', 'column_353', 'column_354', 'column_355', 'column_356', 'column_357',↙
'column_358', 'column_359', 'column_360', 'column_361', 'column_362', 'column_363',↙
'column_364', 'column_365', 'column_366', 'column_367', 'column_368', 'column_369',↙
'column_370', 'column_371', 'column_372', 'column_373', 'column_374', 'column_375',↙
'column_376', 'column_377', 'column_378', 'column_379', 'column_380', 'column_381',↙
'column_382', 'column_383', 'column_384', 'column_385', 'column_386', 'column_387',↙
'column_388', 'column_389', 'column_390', 'column_391', 'column_392', 'column_393',↙
'column_394', 'column_395', 'column_396', 'column_397', 'column_398', 'column_399',↙
'column_400', 'column_401', 'column_402', 'column_403', 'column_404', 'column_405',↙
'column_406', 'column_407', 'column_408', 'column_409', 'column_410', 'column_411',↙
'column_412', 'column_413', 'column_414', 'column_415', 'column_416', 'column_417',↙
'column_418', 'column_419', 'column_420', 'column_421', 'column_422', 'column_423',↙
'column_424', 'column_425', 'column_426', 'column_427', 'column_428', 'column_429',↙
'column_430', 'column_431', 'column_432', 'column_433', 'column_434', 'column_435',↙
'column_436', 'column_437', 'column_438', 'column_439', 'column_440', 'column_441',↙
'column_442', 'column_443', 'column_444', 'column_445', 'column_446', 'column_447',↙
'column_448', 'column_449', 'column_450', 'column_451', 'column_452', 'column_453',↙
'column_454', 'column_455', 'column_456', 'column_457', 'column_458', 'column_459',↙
'column_460', 'column_461', 'column_462', 'column_463', 'column_464', 'column_465',↙
'column_466', 'column_467', 'column_468', 'column_469', 'column_470', 'column_471',↙
'column_472', 'column_473', 'column_474', 'column_475', 'column_476', 'column_477',↙
'column_478', 'column_479', 'column_480', 'column_481', 'column_482', 'column_483',↙
'column_484', 'column_485', 'column_486', 'column_487', 'column_488', 'column_489',↙
'column_490', 'column_491', 'column_492', 'column_493', 'column_494', 'column_495',↙
'column_496', 'column_497', 'column_498', 'column_499', 'column_500', 'column_501',↙
'column_502', 'column_503', 'column_504', 'column_505', 'column_506', 'column_507',↙
'column_508', 'column_509', 'column_510', 'column_511', 'column_512', 'column_513',↙
'column_514', 'column_515', 'column_516', 'column_517', 'column_518', 'column_519',↙
'column_520', 'column_521', 'column_522', 'column_523', 'column_524', 'column_525',↙
'column_526', 'column_527', 'column_528', 'column_529', 'column_530', 'column_531',↙
'column_532', 'column_533', 'column_534', 'column_535', 'column_536', 'column_537',↙
'column_538', 'column_539', 'column_540', 'column_541', 'column_542', 'column_543',↙
'column_544', 'column_545', 'column_546', 'column_547', 'column_548', 'column_549',↙
'column_550', 'column_551', 'column_552', 'column_553', 'column_554', 'column_555',↙
'column_556', 'column_557', 'column_558', 'column_559', 'column_560', 'column_561',↙
'column_562', 'column_563', 'column_564', 'column_565', 'column_566', 'column_567',↙
'column_568', 'column_569', 'column_570', 'column_571', 'column_572', 'column_573',↙
'column_574', 'column_575', 'column_576', 'column_577', 'column_578', 'column_579',↙
'column_580', 'column_581', 'column_582', 'column_583', 'column_584', 'column_585',↙
'column_586', 'column_587', 'column_588', 'column_589', 'column_590', 'column_591',↙
'column_592', 'column_593', 'column_594', 'column_595', 'column_596', 'column_597',↙
'column_598', 'column_599', 'column_600', 'column_601', 'column_602', 'column_603',↙
```

```
'column_604', 'column_605', 'column_606', 'column_607', 'column_608', 'column_609',↙
'column_610', 'column_611', 'column_612', 'column_613', 'column_614', 'column_615',↙
'column_616', 'column_617', 'column_618', 'column_619', 'column_620', 'column_621',↙
'column_622', 'column_623', 'column_624', 'column_625', 'column_626', 'column_627',↙
'column_628', 'column_629', 'column_630', 'column_631', 'column_632', 'column_633',↙
'column_634', 'column_635', 'column_636', 'column_637', 'column_638', 'column_639',↙
'column_640', 'column_641', 'column_642', 'column_643', 'column_644', 'column_645',↙
'column_646', 'column_647', 'column_648', 'column_649', 'column_650', 'column_651',↙
'column_652', 'column_653', 'column_654', 'column_655', 'column_656', 'column_657',↙
'column_658', 'column_659', 'column_660', 'column_661', 'column_662', 'column_663',↙
'column_664', 'column_665', 'column_666', 'column_667', 'column_668', 'column_669',↙
'column_670', 'column_671', 'column_672', 'column_673', 'column_674', 'column_675',↙
'column_676', 'column_677', 'column_678', 'column_679', 'column_680', 'column_681',↙
'column_682', 'column_683', 'column_684', 'column_685', 'column_686', 'column_687',↙
'column_688', 'column_689', 'column_690', 'column_691', 'column_692', 'column_693',↙
'column_694', 'column_695', 'column_696', 'column_697', 'column_698', 'column_699',↙
'column_700', 'column_701', 'column_702', 'column_703', 'column_704', 'column_705',↙
'column_706', 'column_707', 'column_708', 'column_709', 'column_710', 'column_711',↙
'column_712', 'column_713', 'column_714', 'column_715', 'column_716', 'column_717',↙
'column_718', 'column_719', 'column_720', 'column_721', 'column_722', 'column_723',↙
'column_724', 'column_725', 'column_726', 'column_727', 'column_728', 'column_729',↙
'column_730', 'column_731', 'column_732', 'column_733', 'column_734', 'column_735',↙
'column_736', 'column_737', 'column_738', 'column_739', 'column_740', 'column_741',↙
'column_742', 'column_743', 'column_744', 'column_745', 'column_746', 'column_747',↙
'column_748', 'column_749', 'column_750', 'column_751', 'column_752', 'column_753',↙
'column_754', 'column_755', 'column_756', 'column_757', 'column_758', 'column_759',↙
'column_760', 'column_761', 'column_762', 'column_763', 'column_764', 'column_765',↙
'column_766', 'column_767', 'column_768', 'column_769', 'column_770', 'column_771',↙
'column_772', 'column_773'});

predictorNames = {'column_1', 'column_2', 'column_3', 'column_4', 'column_5', 'column_6',↙
'column_7', 'column_8', 'column_9', 'column_10', 'column_11', 'column_12', 'column_13',↙
'column_14', 'column_15', 'column_16', 'column_17', 'column_18', 'column_19',↙
'column_20', 'column_21', 'column_22', 'column_23', 'column_24', 'column_25',↙
'column_26', 'column_27', 'column_28', 'column_29', 'column_30', 'column_31',↙
'column_32', 'column_33', 'column_34', 'column_35', 'column_36', 'column_37',↙
'column_38', 'column_39', 'column_40', 'column_41', 'column_42', 'column_43',↙
'column_44', 'column_45', 'column_46', 'column_47', 'column_48', 'column_49',↙
'column_50', 'column_51', 'column_52', 'column_53', 'column_54', 'column_55',↙
'column_56', 'column_57', 'column_58', 'column_59', 'column_60', 'column_61',↙
'column_62', 'column_63', 'column_64', 'column_65', 'column_66', 'column_67',↙
'column_68', 'column_69', 'column_70', 'column_71', 'column_72', 'column_73',↙
'column_74', 'column_75', 'column_76', 'column_77', 'column_78', 'column_79',↙
'column_80', 'column_81', 'column_82', 'column_83', 'column_84', 'column_85',↙
'column_86', 'column_87', 'column_88', 'column_89', 'column_90', 'column_91',↙
'column_92', 'column_93', 'column_94', 'column_95', 'column_96', 'column_97',↙
'column_98', 'column_99', 'column_100', 'column_101', 'column_102', 'column_103',↙
'column_104', 'column_105', 'column_106', 'column_107', 'column_108', 'column_109',↙
'column_110', 'column_111', 'column_112', 'column_113', 'column_114', 'column_115',↙
'column_116', 'column_117', 'column_118', 'column_119', 'column_120', 'column_121',↙
'column_122', 'column_123', 'column_124', 'column_125', 'column_126', 'column_127',↙
```

```
'column_128', 'column_129', 'column_130', 'column_131', 'column_132', 'column_133',↙
'column_134', 'column_135', 'column_136', 'column_137', 'column_138', 'column_139',↙
'column_140', 'column_141', 'column_142', 'column_143', 'column_144', 'column_145',↙
'column_146', 'column_147', 'column_148', 'column_149', 'column_150', 'column_151',↙
'column_152', 'column_153', 'column_154', 'column_155', 'column_156', 'column_157',↙
'column_158', 'column_159', 'column_160', 'column_161', 'column_162', 'column_163',↙
'column_164', 'column_165', 'column_166', 'column_167', 'column_168', 'column_169',↙
'column_170', 'column_171', 'column_172', 'column_173', 'column_174', 'column_175',↙
'column_176', 'column_177', 'column_178', 'column_179', 'column_180', 'column_181',↙
'column_182', 'column_183', 'column_184', 'column_185', 'column_186', 'column_187',↙
'column_188', 'column_189', 'column_190', 'column_191', 'column_192', 'column_193',↙
'column_194', 'column_195', 'column_196', 'column_197', 'column_198', 'column_199',↙
'column_200', 'column_201', 'column_202', 'column_203', 'column_204', 'column_205',↙
'column_206', 'column_207', 'column_208', 'column_209', 'column_210', 'column_211',↙
'column_212', 'column_213', 'column_214', 'column_215', 'column_216', 'column_217',↙
'column_218', 'column_219', 'column_220', 'column_221', 'column_222', 'column_223',↙
'column_224', 'column_225', 'column_226', 'column_227', 'column_228', 'column_229',↙
'column_230', 'column_231', 'column_232', 'column_233', 'column_234', 'column_235',↙
'column_236', 'column_237', 'column_238', 'column_239', 'column_240', 'column_241',↙
'column_242', 'column_243', 'column_244', 'column_245', 'column_246', 'column_247',↙
'column_248', 'column_249', 'column_250', 'column_251', 'column_252', 'column_253',↙
'column_254', 'column_255', 'column_256', 'column_257', 'column_258', 'column_259',↙
'column_260', 'column_261', 'column_262', 'column_263', 'column_264', 'column_265',↙
'column_266', 'column_267', 'column_268', 'column_269', 'column_270', 'column_271',↙
'column_272', 'column_273', 'column_274', 'column_275', 'column_276', 'column_277',↙
'column_278', 'column_279', 'column_280', 'column_281', 'column_282', 'column_283',↙
'column_284', 'column_285', 'column_286', 'column_287', 'column_288', 'column_289',↙
'column_290', 'column_291', 'column_292', 'column_293', 'column_294', 'column_295',↙
'column_296', 'column_297', 'column_298', 'column_299', 'column_300', 'column_301',↙
'column_302', 'column_303', 'column_304', 'column_305', 'column_306', 'column_307',↙
'column_308', 'column_309', 'column_310', 'column_311', 'column_312', 'column_313',↙
'column_314', 'column_315', 'column_316', 'column_317', 'column_318', 'column_319',↙
'column_320', 'column_321', 'column_322', 'column_323', 'column_324', 'column_325',↙
'column_326', 'column_327', 'column_328', 'column_329', 'column_330', 'column_331',↙
'column_332', 'column_333', 'column_334', 'column_335', 'column_336', 'column_337',↙
'column_338', 'column_339', 'column_340', 'column_341', 'column_342', 'column_343',↙
'column_344', 'column_345', 'column_346', 'column_347', 'column_348', 'column_349',↙
'column_350', 'column_351', 'column_352', 'column_353', 'column_354', 'column_355',↙
'column_356', 'column_357', 'column_358', 'column_359', 'column_360', 'column_361',↙
'column_362', 'column_363', 'column_364', 'column_365', 'column_366', 'column_367',↙
'column_368', 'column_369', 'column_370', 'column_371', 'column_372', 'column_373',↙
'column_374', 'column_375', 'column_376', 'column_377', 'column_378', 'column_379',↙
'column_380', 'column_381', 'column_382', 'column_383', 'column_384', 'column_385',↙
'column_386', 'column_387', 'column_388', 'column_389', 'column_390', 'column_391',↙
'column_392', 'column_393', 'column_394', 'column_395', 'column_396', 'column_397',↙
'column_398', 'column_399', 'column_400', 'column_401', 'column_402', 'column_403',↙
'column_404', 'column_405', 'column_406', 'column_407', 'column_408', 'column_409',↙
'column_410', 'column_411', 'column_412', 'column_413', 'column_414', 'column_415',↙
'column_416', 'column_417', 'column_418', 'column_419', 'column_420', 'column_421',↙
'column_422', 'column_423', 'column_424', 'column_425', 'column_426', 'column_427',↙
'column_428', 'column_429', 'column_430', 'column_431', 'column_432', 'column_433',↙
```

```
'column_434', 'column_435', 'column_436', 'column_437', 'column_438', 'column_439',↙
'column_440', 'column_441', 'column_442', 'column_443', 'column_444', 'column_445',↙
'column_446', 'column_447', 'column_448', 'column_449', 'column_450', 'column_451',↙
'column_452', 'column_453', 'column_454', 'column_455', 'column_456', 'column_457',↙
'column_458', 'column_459', 'column_460', 'column_461', 'column_462', 'column_463',↙
'column_464', 'column_465', 'column_466', 'column_467', 'column_468', 'column_469',↙
'column_470', 'column_471', 'column_472', 'column_473', 'column_474', 'column_475',↙
'column_476', 'column_477', 'column_478', 'column_479', 'column_480', 'column_481',↙
'column_482', 'column_483', 'column_484', 'column_485', 'column_486', 'column_487',↙
'column_488', 'column_489', 'column_490', 'column_491', 'column_492', 'column_493',↙
'column_494', 'column_495', 'column_496', 'column_497', 'column_498', 'column_499',↙
'column_500', 'column_501', 'column_502', 'column_503', 'column_504', 'column_505',↙
'column_506', 'column_507', 'column_508', 'column_509', 'column_510', 'column_511',↙
'column_512', 'column_513', 'column_514', 'column_515', 'column_516', 'column_517',↙
'column_518', 'column_519', 'column_520', 'column_521', 'column_522', 'column_523',↙
'column_524', 'column_525', 'column_526', 'column_527', 'column_528', 'column_529',↙
'column_530', 'column_531', 'column_532', 'column_533', 'column_534', 'column_535',↙
'column_536', 'column_537', 'column_538', 'column_539', 'column_540', 'column_541',↙
'column_542', 'column_543', 'column_544', 'column_545', 'column_546', 'column_547',↙
'column_548', 'column_549', 'column_550', 'column_551', 'column_552', 'column_553',↙
'column_554', 'column_555', 'column_556', 'column_557', 'column_558', 'column_559',↙
'column_560', 'column_561', 'column_562', 'column_563', 'column_564', 'column_565',↙
'column_566', 'column_567', 'column_568', 'column_569', 'column_570', 'column_571',↙
'column_572', 'column_573', 'column_574', 'column_575', 'column_576', 'column_577',↙
'column_578', 'column_579', 'column_580', 'column_581', 'column_582', 'column_583',↙
'column_584', 'column_585', 'column_586', 'column_587', 'column_588', 'column_589',↙
'column_590', 'column_591', 'column_592', 'column_593', 'column_594', 'column_595',↙
'column_596', 'column_597', 'column_598', 'column_599', 'column_600', 'column_601',↙
'column_602', 'column_603', 'column_604', 'column_605', 'column_606', 'column_607',↙
'column_608', 'column_609', 'column_610', 'column_611', 'column_612', 'column_613',↙
'column_614', 'column_615', 'column_616', 'column_617', 'column_618', 'column_619',↙
'column_620', 'column_621', 'column_622', 'column_623', 'column_624', 'column_625',↙
'column_626', 'column_627', 'column_628', 'column_629', 'column_630', 'column_631',↙
'column_632', 'column_633', 'column_634', 'column_635', 'column_636', 'column_637',↙
'column_638', 'column_639', 'column_640', 'column_641', 'column_642', 'column_643',↙
'column_644', 'column_645', 'column_646', 'column_647', 'column_648', 'column_649',↙
'column_650', 'column_651', 'column_652', 'column_653', 'column_654', 'column_655',↙
'column_656', 'column_657', 'column_658', 'column_659', 'column_660', 'column_661',↙
'column_662', 'column_663', 'column_664', 'column_665', 'column_666', 'column_667',↙
'column_668', 'column_669', 'column_670', 'column_671', 'column_672', 'column_673',↙
'column_674', 'column_675', 'column_676', 'column_677', 'column_678', 'column_679',↙
'column_680', 'column_681', 'column_682', 'column_683', 'column_684', 'column_685',↙
'column_686', 'column_687', 'column_688', 'column_689', 'column_690', 'column_691',↙
'column_692', 'column_693', 'column_694', 'column_695', 'column_696', 'column_697',↙
'column_698', 'column_699', 'column_700', 'column_701', 'column_702', 'column_703',↙
'column_704', 'column_705', 'column_706', 'column_707', 'column_708', 'column_709',↙
'column_710', 'column_711', 'column_712', 'column_713', 'column_714', 'column_715',↙
'column_716', 'column_717', 'column_718', 'column_719', 'column_720', 'column_721',↙
'column_722', 'column_723', 'column_724', 'column_725', 'column_726', 'column_727',↙
'column_728', 'column_729', 'column_730', 'column_731', 'column_732', 'column_733',↙
'column_734', 'column_735', 'column_736', 'column_737', 'column_738', 'column_739',↙
```

```
'column_740', 'column_741', 'column_742', 'column_743', 'column_744', 'column_745',↙
'column_746', 'column_747', 'column_748', 'column_749', 'column_750', 'column_751',↙
'column_752', 'column_753', 'column_754', 'column_755', 'column_756', 'column_757',↙
'column_758', 'column_759', 'column_760', 'column_761', 'column_762', 'column_763',↙
'column_764', 'column_765', 'column_766', 'column_767', 'column_768', 'column_769',↙
'column_770', 'column_771', 'column_772', 'column_773'};
predictors = inputTable(:, predictorNames);
response = responseData;
isCategoricalPredictor = [false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
```

```
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
template = templateSVM(...
    'KernelFunction', 'linear', ...
    'PolynomialOrder', [], ...
    'KernelScale', 'auto', ...
    'BoxConstraint', 1, ...
    'Standardize', true);
classificationSVM = fitcecoc(...
    predictors, ...
    response, ...
    'Learners', template, ...
    'Coding', 'onevsone', ...
    'ClassNames', {'Black Sea Sprat'; 'Gilt-Head Breamt'; 'Hourse Mackerelt'; 'Red↙
Mullet'; 'Red Sea Bream'; 'Sea Bass'; 'Striped Red Mullet'; 'Trout'});

% Create the result struct with predict function
predictorExtractionFcn = @(x) array2table(x, 'VariableNames', predictorNames);
svmPredictFcn = @(x) predict(classificationSVM, x);
trainedClassifier.predictFcn = @(x) svmPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.ClassificationSVM = classificationSVM;
trainedClassifier.About = 'This struct is a trained model exported from Classification↙
Learner R2021a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new predictor column↙
matrix, X, use: \n  yfit = c.predictFcn(X) \nreplacing ''c'' with the name of the↙
variable that is this struct, e.g. ''trainedModel''. \n \nX must contain exactly 773↙
```

```
columns because this model was trained using 773 predictors. \nX must contain only
predictor columns in exactly the same order and format as your training \ndata. Do not
include the response column or any columns you did not import into the app. \n \nFor more
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'', ''stats.map''),
''appclassification_exportmodeltoworkspace'')">How to predict using an exported
model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
% Convert input to table
inputTable = array2table(trainingData, 'VariableNames', {'column_1', 'column_2',
'column_3', 'column_4', 'column_5', 'column_6', 'column_7', 'column_8', 'column_9',
'column_10', 'column_11', 'column_12', 'column_13', 'column_14', 'column_15',
'column_16', 'column_17', 'column_18', 'column_19', 'column_20', 'column_21',
'column_22', 'column_23', 'column_24', 'column_25', 'column_26', 'column_27',
'column_28', 'column_29', 'column_30', 'column_31', 'column_32', 'column_33',
'column_34', 'column_35', 'column_36', 'column_37', 'column_38', 'column_39',
'column_40', 'column_41', 'column_42', 'column_43', 'column_44', 'column_45',
'column_46', 'column_47', 'column_48', 'column_49', 'column_50', 'column_51',
'column_52', 'column_53', 'column_54', 'column_55', 'column_56', 'column_57',
'column_58', 'column_59', 'column_60', 'column_61', 'column_62', 'column_63',
'column_64', 'column_65', 'column_66', 'column_67', 'column_68', 'column_69',
'column_70', 'column_71', 'column_72', 'column_73', 'column_74', 'column_75',
'column_76', 'column_77', 'column_78', 'column_79', 'column_80', 'column_81',
'column_82', 'column_83', 'column_84', 'column_85', 'column_86', 'column_87',
'column_88', 'column_89', 'column_90', 'column_91', 'column_92', 'column_93',
'column_94', 'column_95', 'column_96', 'column_97', 'column_98', 'column_99',
'column_100', 'column_101', 'column_102', 'column_103', 'column_104', 'column_105',
'column_106', 'column_107', 'column_108', 'column_109', 'column_110', 'column_111',
'column_112', 'column_113', 'column_114', 'column_115', 'column_116', 'column_117',
'column_118', 'column_119', 'column_120', 'column_121', 'column_122', 'column_123',
'column_124', 'column_125', 'column_126', 'column_127', 'column_128', 'column_129',
'column_130', 'column_131', 'column_132', 'column_133', 'column_134', 'column_135',
'column_136', 'column_137', 'column_138', 'column_139', 'column_140', 'column_141',
'column_142', 'column_143', 'column_144', 'column_145', 'column_146', 'column_147',
'column_148', 'column_149', 'column_150', 'column_151', 'column_152', 'column_153',
'column_154', 'column_155', 'column_156', 'column_157', 'column_158', 'column_159',
'column_160', 'column_161', 'column_162', 'column_163', 'column_164', 'column_165',
'column_166', 'column_167', 'column_168', 'column_169', 'column_170', 'column_171',
'column_172', 'column_173', 'column_174', 'column_175', 'column_176', 'column_177',
'column_178', 'column_179', 'column_180', 'column_181', 'column_182', 'column_183',
'column_184', 'column_185', 'column_186', 'column_187', 'column_188', 'column_189',
'column_190', 'column_191', 'column_192', 'column_193', 'column_194', 'column_195',
'column_196', 'column_197', 'column_198', 'column_199', 'column_200', 'column_201',
'column_202', 'column_203', 'column_204', 'column_205', 'column_206', 'column_207',
'column_208', 'column_209', 'column_210', 'column_211', 'column_212', 'column_213',
'column_214', 'column_215', 'column_216', 'column_217', 'column_218', 'column_219',
'column_220', 'column_221', 'column_222', 'column_223', 'column_224', 'column_225',
'column_226', 'column_227', 'column_228', 'column_229', 'column_230', 'column_231',
'column_232', 'column_233', 'column_234', 'column_235', 'column_236', 'column_237',
```

```
'column_238', 'column_239', 'column_240', 'column_241', 'column_242', 'column_243',↙
'column_244', 'column_245', 'column_246', 'column_247', 'column_248', 'column_249',↙
'column_250', 'column_251', 'column_252', 'column_253', 'column_254', 'column_255',↙
'column_256', 'column_257', 'column_258', 'column_259', 'column_260', 'column_261',↙
'column_262', 'column_263', 'column_264', 'column_265', 'column_266', 'column_267',↙
'column_268', 'column_269', 'column_270', 'column_271', 'column_272', 'column_273',↙
'column_274', 'column_275', 'column_276', 'column_277', 'column_278', 'column_279',↙
'column_280', 'column_281', 'column_282', 'column_283', 'column_284', 'column_285',↙
'column_286', 'column_287', 'column_288', 'column_289', 'column_290', 'column_291',↙
'column_292', 'column_293', 'column_294', 'column_295', 'column_296', 'column_297',↙
'column_298', 'column_299', 'column_300', 'column_301', 'column_302', 'column_303',↙
'column_304', 'column_305', 'column_306', 'column_307', 'column_308', 'column_309',↙
'column_310', 'column_311', 'column_312', 'column_313', 'column_314', 'column_315',↙
'column_316', 'column_317', 'column_318', 'column_319', 'column_320', 'column_321',↙
'column_322', 'column_323', 'column_324', 'column_325', 'column_326', 'column_327',↙
'column_328', 'column_329', 'column_330', 'column_331', 'column_332', 'column_333',↙
'column_334', 'column_335', 'column_336', 'column_337', 'column_338', 'column_339',↙
'column_340', 'column_341', 'column_342', 'column_343', 'column_344', 'column_345',↙
'column_346', 'column_347', 'column_348', 'column_349', 'column_350', 'column_351',↙
'column_352', 'column_353', 'column_354', 'column_355', 'column_356', 'column_357',↙
'column_358', 'column_359', 'column_360', 'column_361', 'column_362', 'column_363',↙
'column_364', 'column_365', 'column_366', 'column_367', 'column_368', 'column_369',↙
'column_370', 'column_371', 'column_372', 'column_373', 'column_374', 'column_375',↙
'column_376', 'column_377', 'column_378', 'column_379', 'column_380', 'column_381',↙
'column_382', 'column_383', 'column_384', 'column_385', 'column_386', 'column_387',↙
'column_388', 'column_389', 'column_390', 'column_391', 'column_392', 'column_393',↙
'column_394', 'column_395', 'column_396', 'column_397', 'column_398', 'column_399',↙
'column_400', 'column_401', 'column_402', 'column_403', 'column_404', 'column_405',↙
'column_406', 'column_407', 'column_408', 'column_409', 'column_410', 'column_411',↙
'column_412', 'column_413', 'column_414', 'column_415', 'column_416', 'column_417',↙
'column_418', 'column_419', 'column_420', 'column_421', 'column_422', 'column_423',↙
'column_424', 'column_425', 'column_426', 'column_427', 'column_428', 'column_429',↙
'column_430', 'column_431', 'column_432', 'column_433', 'column_434', 'column_435',↙
'column_436', 'column_437', 'column_438', 'column_439', 'column_440', 'column_441',↙
'column_442', 'column_443', 'column_444', 'column_445', 'column_446', 'column_447',↙
'column_448', 'column_449', 'column_450', 'column_451', 'column_452', 'column_453',↙
'column_454', 'column_455', 'column_456', 'column_457', 'column_458', 'column_459',↙
'column_460', 'column_461', 'column_462', 'column_463', 'column_464', 'column_465',↙
'column_466', 'column_467', 'column_468', 'column_469', 'column_470', 'column_471',↙
'column_472', 'column_473', 'column_474', 'column_475', 'column_476', 'column_477',↙
'column_478', 'column_479', 'column_480', 'column_481', 'column_482', 'column_483',↙
'column_484', 'column_485', 'column_486', 'column_487', 'column_488', 'column_489',↙
'column_490', 'column_491', 'column_492', 'column_493', 'column_494', 'column_495',↙
'column_496', 'column_497', 'column_498', 'column_499', 'column_500', 'column_501',↙
'column_502', 'column_503', 'column_504', 'column_505', 'column_506', 'column_507',↙
'column_508', 'column_509', 'column_510', 'column_511', 'column_512', 'column_513',↙
'column_514', 'column_515', 'column_516', 'column_517', 'column_518', 'column_519',↙
'column_520', 'column_521', 'column_522', 'column_523', 'column_524', 'column_525',↙
'column_526', 'column_527', 'column_528', 'column_529', 'column_530', 'column_531',↙
'column_532', 'column_533', 'column_534', 'column_535', 'column_536', 'column_537',↙
'column_538', 'column_539', 'column_540', 'column_541', 'column_542', 'column_543',↙
```

```
'column_544', 'column_545', 'column_546', 'column_547', 'column_548', 'column_549',↙
'column_550', 'column_551', 'column_552', 'column_553', 'column_554', 'column_555',↙
'column_556', 'column_557', 'column_558', 'column_559', 'column_560', 'column_561',↙
'column_562', 'column_563', 'column_564', 'column_565', 'column_566', 'column_567',↙
'column_568', 'column_569', 'column_570', 'column_571', 'column_572', 'column_573',↙
'column_574', 'column_575', 'column_576', 'column_577', 'column_578', 'column_579',↙
'column_580', 'column_581', 'column_582', 'column_583', 'column_584', 'column_585',↙
'column_586', 'column_587', 'column_588', 'column_589', 'column_590', 'column_591',↙
'column_592', 'column_593', 'column_594', 'column_595', 'column_596', 'column_597',↙
'column_598', 'column_599', 'column_600', 'column_601', 'column_602', 'column_603',↙
'column_604', 'column_605', 'column_606', 'column_607', 'column_608', 'column_609',↙
'column_610', 'column_611', 'column_612', 'column_613', 'column_614', 'column_615',↙
'column_616', 'column_617', 'column_618', 'column_619', 'column_620', 'column_621',↙
'column_622', 'column_623', 'column_624', 'column_625', 'column_626', 'column_627',↙
'column_628', 'column_629', 'column_630', 'column_631', 'column_632', 'column_633',↙
'column_634', 'column_635', 'column_636', 'column_637', 'column_638', 'column_639',↙
'column_640', 'column_641', 'column_642', 'column_643', 'column_644', 'column_645',↙
'column_646', 'column_647', 'column_648', 'column_649', 'column_650', 'column_651',↙
'column_652', 'column_653', 'column_654', 'column_655', 'column_656', 'column_657',↙
'column_658', 'column_659', 'column_660', 'column_661', 'column_662', 'column_663',↙
'column_664', 'column_665', 'column_666', 'column_667', 'column_668', 'column_669',↙
'column_670', 'column_671', 'column_672', 'column_673', 'column_674', 'column_675',↙
'column_676', 'column_677', 'column_678', 'column_679', 'column_680', 'column_681',↙
'column_682', 'column_683', 'column_684', 'column_685', 'column_686', 'column_687',↙
'column_688', 'column_689', 'column_690', 'column_691', 'column_692', 'column_693',↙
'column_694', 'column_695', 'column_696', 'column_697', 'column_698', 'column_699',↙
'column_700', 'column_701', 'column_702', 'column_703', 'column_704', 'column_705',↙
'column_706', 'column_707', 'column_708', 'column_709', 'column_710', 'column_711',↙
'column_712', 'column_713', 'column_714', 'column_715', 'column_716', 'column_717',↙
'column_718', 'column_719', 'column_720', 'column_721', 'column_722', 'column_723',↙
'column_724', 'column_725', 'column_726', 'column_727', 'column_728', 'column_729',↙
'column_730', 'column_731', 'column_732', 'column_733', 'column_734', 'column_735',↙
'column_736', 'column_737', 'column_738', 'column_739', 'column_740', 'column_741',↙
'column_742', 'column_743', 'column_744', 'column_745', 'column_746', 'column_747',↙
'column_748', 'column_749', 'column_750', 'column_751', 'column_752', 'column_753',↙
'column_754', 'column_755', 'column_756', 'column_757', 'column_758', 'column_759',↙
'column_760', 'column_761', 'column_762', 'column_763', 'column_764', 'column_765',↙
'column_766', 'column_767', 'column_768', 'column_769', 'column_770', 'column_771',↙
'column_772', 'column_773'});

predictorNames = {'column_1', 'column_2', 'column_3', 'column_4', 'column_5', 'column_6',↙
'column_7', 'column_8', 'column_9', 'column_10', 'column_11', 'column_12', 'column_13',↙
'column_14', 'column_15', 'column_16', 'column_17', 'column_18', 'column_19',↙
'column_20', 'column_21', 'column_22', 'column_23', 'column_24', 'column_25',↙
'column_26', 'column_27', 'column_28', 'column_29', 'column_30', 'column_31',↙
'column_32', 'column_33', 'column_34', 'column_35', 'column_36', 'column_37',↙
'column_38', 'column_39', 'column_40', 'column_41', 'column_42', 'column_43',↙
'column_44', 'column_45', 'column_46', 'column_47', 'column_48', 'column_49',↙
'column_50', 'column_51', 'column_52', 'column_53', 'column_54', 'column_55',↙
'column_56', 'column_57', 'column_58', 'column_59', 'column_60', 'column_61',↙
'column_62', 'column_63', 'column_64', 'column_65', 'column_66', 'column_67',↙
```

```
'column_68', 'column_69', 'column_70', 'column_71', 'column_72', 'column_73',↙
'column_74', 'column_75', 'column_76', 'column_77', 'column_78', 'column_79',↙
'column_80', 'column_81', 'column_82', 'column_83', 'column_84', 'column_85',↙
'column_86', 'column_87', 'column_88', 'column_89', 'column_90', 'column_91',↙
'column_92', 'column_93', 'column_94', 'column_95', 'column_96', 'column_97',↙
'column_98', 'column_99', 'column_100', 'column_101', 'column_102', 'column_103',↙
'column_104', 'column_105', 'column_106', 'column_107', 'column_108', 'column_109',↙
'column_110', 'column_111', 'column_112', 'column_113', 'column_114', 'column_115',↙
'column_116', 'column_117', 'column_118', 'column_119', 'column_120', 'column_121',↙
'column_122', 'column_123', 'column_124', 'column_125', 'column_126', 'column_127',↙
'column_128', 'column_129', 'column_130', 'column_131', 'column_132', 'column_133',↙
'column_134', 'column_135', 'column_136', 'column_137', 'column_138', 'column_139',↙
'column_140', 'column_141', 'column_142', 'column_143', 'column_144', 'column_145',↙
'column_146', 'column_147', 'column_148', 'column_149', 'column_150', 'column_151',↙
'column_152', 'column_153', 'column_154', 'column_155', 'column_156', 'column_157',↙
'column_158', 'column_159', 'column_160', 'column_161', 'column_162', 'column_163',↙
'column_164', 'column_165', 'column_166', 'column_167', 'column_168', 'column_169',↙
'column_170', 'column_171', 'column_172', 'column_173', 'column_174', 'column_175',↙
'column_176', 'column_177', 'column_178', 'column_179', 'column_180', 'column_181',↙
'column_182', 'column_183', 'column_184', 'column_185', 'column_186', 'column_187',↙
'column_188', 'column_189', 'column_190', 'column_191', 'column_192', 'column_193',↙
'column_194', 'column_195', 'column_196', 'column_197', 'column_198', 'column_199',↙
'column_200', 'column_201', 'column_202', 'column_203', 'column_204', 'column_205',↙
'column_206', 'column_207', 'column_208', 'column_209', 'column_210', 'column_211',↙
'column_212', 'column_213', 'column_214', 'column_215', 'column_216', 'column_217',↙
'column_218', 'column_219', 'column_220', 'column_221', 'column_222', 'column_223',↙
'column_224', 'column_225', 'column_226', 'column_227', 'column_228', 'column_229',↙
'column_230', 'column_231', 'column_232', 'column_233', 'column_234', 'column_235',↙
'column_236', 'column_237', 'column_238', 'column_239', 'column_240', 'column_241',↙
'column_242', 'column_243', 'column_244', 'column_245', 'column_246', 'column_247',↙
'column_248', 'column_249', 'column_250', 'column_251', 'column_252', 'column_253',↙
'column_254', 'column_255', 'column_256', 'column_257', 'column_258', 'column_259',↙
'column_260', 'column_261', 'column_262', 'column_263', 'column_264', 'column_265',↙
'column_266', 'column_267', 'column_268', 'column_269', 'column_270', 'column_271',↙
'column_272', 'column_273', 'column_274', 'column_275', 'column_276', 'column_277',↙
'column_278', 'column_279', 'column_280', 'column_281', 'column_282', 'column_283',↙
'column_284', 'column_285', 'column_286', 'column_287', 'column_288', 'column_289',↙
'column_290', 'column_291', 'column_292', 'column_293', 'column_294', 'column_295',↙
'column_296', 'column_297', 'column_298', 'column_299', 'column_300', 'column_301',↙
'column_302', 'column_303', 'column_304', 'column_305', 'column_306', 'column_307',↙
'column_308', 'column_309', 'column_310', 'column_311', 'column_312', 'column_313',↙
'column_314', 'column_315', 'column_316', 'column_317', 'column_318', 'column_319',↙
'column_320', 'column_321', 'column_322', 'column_323', 'column_324', 'column_325',↙
'column_326', 'column_327', 'column_328', 'column_329', 'column_330', 'column_331',↙
'column_332', 'column_333', 'column_334', 'column_335', 'column_336', 'column_337',↙
'column_338', 'column_339', 'column_340', 'column_341', 'column_342', 'column_343',↙
'column_344', 'column_345', 'column_346', 'column_347', 'column_348', 'column_349',↙
'column_350', 'column_351', 'column_352', 'column_353', 'column_354', 'column_355',↙
'column_356', 'column_357', 'column_358', 'column_359', 'column_360', 'column_361',↙
'column_362', 'column_363', 'column_364', 'column_365', 'column_366', 'column_367',↙
'column_368', 'column_369', 'column_370', 'column_371', 'column_372', 'column_373',↙
```

```
'column_374', 'column_375', 'column_376', 'column_377', 'column_378', 'column_379',↙
'column_380', 'column_381', 'column_382', 'column_383', 'column_384', 'column_385',↙
'column_386', 'column_387', 'column_388', 'column_389', 'column_390', 'column_391',↙
'column_392', 'column_393', 'column_394', 'column_395', 'column_396', 'column_397',↙
'column_398', 'column_399', 'column_400', 'column_401', 'column_402', 'column_403',↙
'column_404', 'column_405', 'column_406', 'column_407', 'column_408', 'column_409',↙
'column_410', 'column_411', 'column_412', 'column_413', 'column_414', 'column_415',↙
'column_416', 'column_417', 'column_418', 'column_419', 'column_420', 'column_421',↙
'column_422', 'column_423', 'column_424', 'column_425', 'column_426', 'column_427',↙
'column_428', 'column_429', 'column_430', 'column_431', 'column_432', 'column_433',↙
'column_434', 'column_435', 'column_436', 'column_437', 'column_438', 'column_439',↙
'column_440', 'column_441', 'column_442', 'column_443', 'column_444', 'column_445',↙
'column_446', 'column_447', 'column_448', 'column_449', 'column_450', 'column_451',↙
'column_452', 'column_453', 'column_454', 'column_455', 'column_456', 'column_457',↙
'column_458', 'column_459', 'column_460', 'column_461', 'column_462', 'column_463',↙
'column_464', 'column_465', 'column_466', 'column_467', 'column_468', 'column_469',↙
'column_470', 'column_471', 'column_472', 'column_473', 'column_474', 'column_475',↙
'column_476', 'column_477', 'column_478', 'column_479', 'column_480', 'column_481',↙
'column_482', 'column_483', 'column_484', 'column_485', 'column_486', 'column_487',↙
'column_488', 'column_489', 'column_490', 'column_491', 'column_492', 'column_493',↙
'column_494', 'column_495', 'column_496', 'column_497', 'column_498', 'column_499',↙
'column_500', 'column_501', 'column_502', 'column_503', 'column_504', 'column_505',↙
'column_506', 'column_507', 'column_508', 'column_509', 'column_510', 'column_511',↙
'column_512', 'column_513', 'column_514', 'column_515', 'column_516', 'column_517',↙
'column_518', 'column_519', 'column_520', 'column_521', 'column_522', 'column_523',↙
'column_524', 'column_525', 'column_526', 'column_527', 'column_528', 'column_529',↙
'column_530', 'column_531', 'column_532', 'column_533', 'column_534', 'column_535',↙
'column_536', 'column_537', 'column_538', 'column_539', 'column_540', 'column_541',↙
'column_542', 'column_543', 'column_544', 'column_545', 'column_546', 'column_547',↙
'column_548', 'column_549', 'column_550', 'column_551', 'column_552', 'column_553',↙
'column_554', 'column_555', 'column_556', 'column_557', 'column_558', 'column_559',↙
'column_560', 'column_561', 'column_562', 'column_563', 'column_564', 'column_565',↙
'column_566', 'column_567', 'column_568', 'column_569', 'column_570', 'column_571',↙
'column_572', 'column_573', 'column_574', 'column_575', 'column_576', 'column_577',↙
'column_578', 'column_579', 'column_580', 'column_581', 'column_582', 'column_583',↙
'column_584', 'column_585', 'column_586', 'column_587', 'column_588', 'column_589',↙
'column_590', 'column_591', 'column_592', 'column_593', 'column_594', 'column_595',↙
'column_596', 'column_597', 'column_598', 'column_599', 'column_600', 'column_601',↙
'column_602', 'column_603', 'column_604', 'column_605', 'column_606', 'column_607',↙
'column_608', 'column_609', 'column_610', 'column_611', 'column_612', 'column_613',↙
'column_614', 'column_615', 'column_616', 'column_617', 'column_618', 'column_619',↙
'column_620', 'column_621', 'column_622', 'column_623', 'column_624', 'column_625',↙
'column_626', 'column_627', 'column_628', 'column_629', 'column_630', 'column_631',↙
'column_632', 'column_633', 'column_634', 'column_635', 'column_636', 'column_637',↙
'column_638', 'column_639', 'column_640', 'column_641', 'column_642', 'column_643',↙
'column_644', 'column_645', 'column_646', 'column_647', 'column_648', 'column_649',↙
'column_650', 'column_651', 'column_652', 'column_653', 'column_654', 'column_655',↙
'column_656', 'column_657', 'column_658', 'column_659', 'column_660', 'column_661',↙
'column_662', 'column_663', 'column_664', 'column_665', 'column_666', 'column_667',↙
'column_668', 'column_669', 'column_670', 'column_671', 'column_672', 'column_673',↙
'column_674', 'column_675', 'column_676', 'column_677', 'column_678', 'column_679',↙
```

```
'column_680', 'column_681', 'column_682', 'column_683', 'column_684', 'column_685',↙
'column_686', 'column_687', 'column_688', 'column_689', 'column_690', 'column_691',↙
'column_692', 'column_693', 'column_694', 'column_695', 'column_696', 'column_697',↙
'column_698', 'column_699', 'column_700', 'column_701', 'column_702', 'column_703',↙
'column_704', 'column_705', 'column_706', 'column_707', 'column_708', 'column_709',↙
'column_710', 'column_711', 'column_712', 'column_713', 'column_714', 'column_715',↙
'column_716', 'column_717', 'column_718', 'column_719', 'column_720', 'column_721',↙
'column_722', 'column_723', 'column_724', 'column_725', 'column_726', 'column_727',↙
'column_728', 'column_729', 'column_730', 'column_731', 'column_732', 'column_733',↙
'column_734', 'column_735', 'column_736', 'column_737', 'column_738', 'column_739',↙
'column_740', 'column_741', 'column_742', 'column_743', 'column_744', 'column_745',↙
'column_746', 'column_747', 'column_748', 'column_749', 'column_750', 'column_751',↙
'column_752', 'column_753', 'column_754', 'column_755', 'column_756', 'column_757',↙
'column_758', 'column_759', 'column_760', 'column_761', 'column_762', 'column_763',↙
'column_764', 'column_765', 'column_766', 'column_767', 'column_768', 'column_769',↙
'column_770', 'column_771', 'column_772', 'column_773'};
predictors = inputTable(:, predictorNames);
response = responseData;
isCategoricalPredictor = [false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
false, false, false, false, false, false, false, false, false, false, false, false,↙
```

```
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false];

% Set up holdout validation
cvp = cvpartition(response, 'Holdout', 0.1);
trainingPredictors = predictors(cvp.training, :);
trainingResponse = response(cvp.training, :);
trainingIsCategoricalPredictor = isCategoricalPredictor;

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
template = templateSVM(...
    'KernelFunction', 'linear', ...
    'PolynomialOrder', [], ...
    'KernelScale', 'auto', ...
    'BoxConstraint', 1, ...
    'Standardize', true);
classificationSVM = fitcecoc(...
    trainingPredictors, ...
    trainingResponse, ...
    'Learners', template, ...
```

```
    'Coding', 'onevsone', ...
    'ClassNames', {'Black Sea Sprat'; 'Gilt-Head Breamt'; 'Hourse Mackerelt'; 'Red↙
Mullet'; 'Red Sea Bream'; 'Sea Bass'; 'Striped Red Mullet'; 'Trout'});

% Create the result struct with predict function
svmPredictFcn = @(x) predict(classificationSVM, x);
validationPredictFcn = @(x) svmPredictFcn(x);

% Add additional fields to the result struct


% Compute validation predictions
validationPredictors = predictors(cvp.test, :);
validationResponse = response(cvp.test, :);
[validationPredictions, validationScores] = validationPredictFcn(validationPredictors);

% Compute validation accuracy
correctPredictions = strcmp( strtrim(validationPredictions), strtrim↙
(validationResponse));
isMissing = cellfun(@(x) all(isspace(x)), validationResponse, 'UniformOutput', true);
correctPredictions = correctPredictions(~isMissing);
validationAccuracy = sum(correctPredictions)/length(correctPredictions);
```

```
clear all
clc

load('train_data.mat'); % 80% training 10% validation
load('train_response.mat');

load('test_data.mat'); % 10% from fish dataset
load('test_response.mat');

N = length(test_response);

[trainSVMClassifier, SVMvalidationAccuracy] = trainClassifier(train_data, train_response)↙
%retrain

disp('Using testing dataset to predict fish type');
yfit = trainSVMClassifier.predictFcn(test_data); %test

disp('[PREDICTED FISH TYPE]    [ACTUAL FISH TYPE]');
true_total=0;
for k=1:N
    output = strcat('[',yfit{k},']      [',test_response{k},']');
    disp(output);
    if isequal(yfit{k}, test_response{k})
        true_total=true_total+1;
    end
end

testing_SVM_accuracy=true_total/N*100
```

Coding for GUI Design

```
classdef main < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
FishClassificationAppUIFigure matlab.ui.Figure
TrainandTestingFishdatasetPanel matlab.ui.container.Panel
ResultPanel matlab.ui.container.Panel
TestingAccuracyEditField matlab.ui.control.NumericEditField
TestingAccuracyEditFieldLabel matlab.ui.control.Label
ValidationAccuracyEditField matlab.ui.control.NumericEditField
ValidationAccuracyEditFieldLabel matlab.ui.control.Label
ChooseClassifierButtonGroup matlab.ui.container.ButtonGroup
RunButton matlab.ui.control.Button
SVMButton matlab.ui.control.RadioButton
KNNButton matlab.ui.control.RadioButton
TestingResponseEditField matlab.ui.control.EditField
SelecttestingresponseButton matlab.ui.control.Button
SelecttestingdatasetButton matlab.ui.control.Button
TestingDataEditField matlab.ui.control.EditField
TrainingResponseEditField matlab.ui.control.EditField
SelecttrainingresponseButton matlab.ui.control.Button
SelecttrainingdatasetButton matlab.ui.control.Button
TrainingDataEditField matlab.ui.control.EditField
SVMClassificationPanel matlab.ui.container.Panel
Image_4 matlab.ui.control.Image
PredictusingSVMClassifierButton matlab.ui.control.Button
PredictedFishEditField_2 matlab.ui.control.EditField
PredictedFishEditField_2Label matlab.ui.control.Label
KNNClassificationPanel matlab.ui.container.Panel
Image_3 matlab.ui.control.Image
PredictusingKNNClassifierButton matlab.ui.control.Button
PredictedFishEditField matlab.ui.control.EditField
PredictedFishEditFieldLabel matlab.ui.control.Label
FeatureExtractionPanel matlab.ui.container.Panel
ExtractFeatureButton matlab.ui.control.Button
ShapePanel matlab.ui.container.Panel
HeightEditField matlab.ui.control.EditField
HeightEditFieldLabel matlab.ui.control.Label
WidthEditField matlab.ui.control.EditField
WidthEditFieldLabel matlab.ui.control.Label
AreaEditField matlab.ui.control.EditField
AreaEditFieldLabel matlab.ui.control.Label
ColorHistogramPanel matlab.ui.container.Panel
PlotBlueHistogramButton matlab.ui.control.Button
PlotGreenHistogramButton matlab.ui.control.Button
PlotRedHistogramButton matlab.ui.control.Button
GLCMPanel matlab.ui.container.Panel
```

```
EnergyEditField matlab.ui.control.EditField
EnergyEditFieldLabel matlab.ui.control.Label
ContrastEditField matlab.ui.control.EditField
ContrastEditFieldLabel matlab.ui.control.Label
SegmentationoffishimagePanel matlab.ui.container.Panel
Image_2 matlab.ui.control.Image
TypeofSegmentationButtonGroup matlab.ui.container.ButtonGroup
SegmentFishImageButton matlab.ui.control.Button
Segmentation3Button matlab.ui.control.RadioButton
Segmentation2Button matlab.ui.control.RadioButton
Segmentation1Button matlab.ui.control.RadioButton
BrowsefishimagePanel matlab.ui.container.Panel
ResetAllButton matlab.ui.control.Button
BrowseButton matlab.ui.control.Button
Image matlab.ui.control.Image
end


properties (Access = private)
traindatafile % train data file
trainrespfile % train response file
testdatafile % test data file
testrespfile % test resp file
hr % red histogram
hg % green histogram
hb % blue histogram
svmclassifier % SVM Classifier
knnclassifier % KNN Classifier
feat % features vector
end


% Callbacks that handle component events
methods (Access = private)


% Button pushed function: SelecttrainingdatasetButton
function SelecttrainingdatasetButtonPushed(app, event)
[file,path] = uigetfile('*.mat');
app.TrainingDataEditField.Value = file;
app.traindatafile = fullfile(path,file);
end


% Button pushed function: SelecttrainingresponseButton
function SelecttrainingresponseButtonPushed(app, event)
[file,path] = uigetfile('*.mat');
app.TrainingResponseEditField.Value = file;
app.trainrespfile = fullfile(path,file);
end


% Button pushed function: SelecttestingdatasetButton
function SelecttestingdatasetButtonPushed(app, event)
```

```
[file,path] = uigetfile('*.mat');
app.TestingDataEditField.Value = file;
app.testdatafile = fullfile(path,file);
end


% Button pushed function: SelecttestingresponseButton
function SelecttestingresponseButtonPushed(app, event)
[file,path] = uigetfile('*.mat');
app.TestingResponseEditField.Value = file;
app.testrespfile = fullfile(path,file);
end


% Button pushed function: RunButton
function RunButtonPushed(app, event)
app.ValidationAccuracyEditField.Value = 0;
app.TestingAccuracyEditField.Value = 0;
if app.SVMButton.Value == 1
[validationAcc, testingAcc, SVMClassifier] =
SVMclassification(app.traindatafile,app.trainrespfile,app.testdatafile,app.tes
trespfile);
app.svmclassifier = SVMClassifier;
elseif app.KNNButton.Value == 1
[validationAcc, testingAcc, KNNClassifier] =
KNNclassification(app.traindatafile,app.trainrespfile,app.testdatafile,app.tes
trespfile);
app.knnclassifier = KNNClassifier;
end
app.ValidationAccuracyEditField.Value = validationAcc;
app.TestingAccuracyEditField.Value = testingAcc;
app.BrowseButton.Enable = 1;
app.ResetAllButton.Enable = 1;
app.SegmentFishImageButton.Enable = 1;
app.ExtractFeatureButton.Enable = 1;
app.PredictusingKNNClassifierButton.Enable = 1;
app.PredictusingSVMClassifierButton.Enable = 1;
end


% Button pushed function: BrowseButton
function BrowseButtonPushed(app, event)
[file,path] = uigetfile('*.jpg;*.png');
app.Image.ImageSource = fullfile(path,file);


end


% Button pushed function: SegmentFishImageButton
function SegmentFishImageButtonPushed(app, event)
if isempty(app.Image.ImageSource)
msgbox('Please select fish image.');
else
```

```
app.Image_2.ImageSource = '';
imgfile = app.Image.ImageSource;
if app.Segmentation1Button.Value == 1
seg = 1;
elseif app.Segmentation2Button.Value == 1
seg = 2;
elseif app.Segmentation3Button.Value==1
seg = 3;
end
[segImage] = preprocessing(imgfile,seg);
app.Image_2.ImageSource = segImage;
end
end


% Button pushed function: ExtractFeatureButton
function ExtractFeatureButtonPushed(app, event)
if isempty(app.Image.ImageSource)
msgbox('Please select fish image.');
else
segfile = app.Image_2.ImageSource;
[Contrast, Energy, imghist, area, fishW, fishH] = featureExtraction(segfile);
app.ContrastEditField.Value = string(Contrast);
app.EnergyEditField.Value = string(Energy);
app.AreaEditField.Value = string(area);
app.WidthEditField.Value = string(fishW);
app.HeightEditField.Value = string(fishH);
app.PlotRedHistogramButton.Enable = 1;
app.PlotGreenHistogramButton.Enable = 1;
app.PlotBlueHistogramButton.Enable = 1;
app.feat = [Contrast, Energy, imghist, area, fishW, fishH];
end
end


% Button pushed function: PlotRedHistogramButton
function PlotRedHistogramButtonPushed(app, event)
if isempty(app.Image.ImageSource)
msgbox('Please select fish image.');
else
segfile = app.Image_2.ImageSource;
rgb = imread(segfile);
r=rgb(:,:,1);
figure,
imhist(r);
title('Histogram plot of Red color');
end
end


% Button pushed function: PlotGreenHistogramButton
function PlotGreenHistogramButtonPushed(app, event)
if isempty(app.Image.ImageSource)
msgbox('Please select fish image.');
```

```
else
segfile = app.Image_2.ImageSource;
rgb = imread(segfile);
g=rgb(:,:,2);
figure,
imhist(g);
title('Histogram plot of Green color');
end
end


% Button pushed function: PlotBlueHistogramButton
function PlotBlueHistogramButtonPushed(app, event)
if isempty(app.Image.ImageSource)
msgbox('Please select fish image.');
else
segfile = app.Image_2.ImageSource;
rgb = imread(segfile);
b=rgb(:,:,3);
figure,
imhist(b);
title('Histogram plot of Blue color')
end
end


% Button pushed function: PredictusingKNNClassifierButton
function PredictusingKNNClassifierButtonPushed(app, event)
if app.PlotRedHistogramButton.Enable == 1
yfit = app.knnclassifier.predictFcn(app.feat);
app.PredictedFishEditField.Value = yfit{1};
app.Image_3.ImageSource = strcat('actualfish/',yfit{1},'.png');
end
end


% Button pushed function: ResetAllButton
function ResetAllButtonPushed(app, event)
app.Image.ImageSource = '';
app.Image_2.ImageSource ='';
app.ContrastEditField.Value = '';
app.EnergyEditField.Value = '';
app.AreaEditField.Value = '';
app.WidthEditField.Value = '';
app.HeightEditField.Value = '';
app.PlotRedHistogramButton.Enable = 0;
app.PlotGreenHistogramButton.Enable = 0;
app.PlotBlueHistogramButton.Enable = 0;
app.PredictedFishEditField.Value = '';
app.PredictedFishEditField_2.Value = '';
app.Image_3.ImageSource = '';
app.Image_4.ImageSource = '';
end
```

```
% Button pushed function: PredictusingSVMClassifierButton
function PredictusingSVMClassifierButtonPushed(app, event)
if app.PlotRedHistogramButton.Enable == 1
yfit = app.svmclassifier.predictFcn(app.feat);
app.PredictedFishEditField_2.Value = yfit{1};
app.Image_4.ImageSource = strcat('actualfish/',yfit{1},'.png');
end
end
end


% Component initialization
methods (Access = private)


% Create UIFigure and components
function createComponents(app)


% Create FishClassificationAppUIFigure and hide until all components are
created
app.FishClassificationAppUIFigure = uifigure('Visible', 'off');
app.FishClassificationAppUIFigure.Position = [100 100 882 564];
app.FishClassificationAppUIFigure.Name = 'Fish Classification App';


% Create BrowsefishimagePanel
app.BrowsefishimagePanel = uipanel(app.FishClassificationAppUIFigure);
app.BrowsefishimagePanel.Title = '1. Browse fish image';
app.BrowsefishimagePanel.Position = [372 290 201 259];


% Create Image
app.Image = uiimage(app.BrowsefishimagePanel);
app.Image.Position = [8 47 187 182];


% Create BrowseButton
app.BrowseButton = uibutton(app.BrowsefishimagePanel, 'push');
app.BrowseButton.ButtonPushedFcn = createCallbackFcn(app, @BrowseButtonPushed,
true);
app.BrowseButton.Enable = 'off';
app.BrowseButton.Position = [17 11 64 22];
app.BrowseButton.Text = 'Browse..';


% Create ResetAllButton
app.ResetAllButton = uibutton(app.BrowsefishimagePanel, 'push');
app.ResetAllButton.ButtonPushedFcn = createCallbackFcn(app,
@ResetAllButtonPushed, true);
app.ResetAllButton.Enable = 'off';
app.ResetAllButton.Position = [87 11 100 22];
app.ResetAllButton.Text = 'Reset All';
```

```
% Create SegmentationoffishimagePanel
app.SegmentationoffishimagePanel = uipanel(app.FishClassificationAppUIFigure);
app.SegmentationoffishimagePanel.Title = '2. Segmentation of fish image';
app.SegmentationoffishimagePanel.Position = [587 290 276 259];


% Create TypeofSegmentationButtonGroup
app.TypeofSegmentationButtonGroup =
uibuttongroup(app.SegmentationoffishimagePanel);
app.TypeofSegmentationButtonGroup.Title = 'Type of Segmentation';
app.TypeofSegmentationButtonGroup.Position = [8 7 259 70];


% Create Segmentation1Button
app.Segmentation1Button = uiradiobutton(app.TypeofSegmentationButtonGroup);
app.Segmentation1Button.Text = 'Segmentation 1';
app.Segmentation1Button.Position = [11 24 106 22];
app.Segmentation1Button.Value = true;


% Create Segmentation2Button
app.Segmentation2Button = uiradiobutton(app.TypeofSegmentationButtonGroup);
app.Segmentation2Button.Text = 'Segmentation 2';
app.Segmentation2Button.Position = [11 2 106 22];


% Create Segmentation3Button
app.Segmentation3Button = uiradiobutton(app.TypeofSegmentationButtonGroup);
app.Segmentation3Button.Text = 'Segmentation 3';
app.Segmentation3Button.Position = [126 24 106 22];


% Create SegmentFishImageButton
app.SegmentFishImageButton = uibutton(app.TypeofSegmentationButtonGroup,
'push');
app.SegmentFishImageButton.ButtonPushedFcn = createCallbackFcn(app,
@SegmentFishImageButtonPushed, true);
app.SegmentFishImageButton.Enable = 'off';
app.SegmentFishImageButton.Position = [117 3 126 22];
app.SegmentFishImageButton.Text = 'Segment Fish Image';


% Create Image_2
app.Image_2 = uiimage(app.SegmentationoffishimagePanel);
app.Image_2.Position = [65 76 152 164];


% Create FeatureExtractionPanel
app.FeatureExtractionPanel = uipanel(app.FishClassificationAppUIFigure);
app.FeatureExtractionPanel.Title = '3. Feature Extraction';
app.FeatureExtractionPanel.Position = [22 11 322 266];


% Create GLCMPanel
```

```
app.GLCMPanel = uipanel(app.FeatureExtractionPanel);
app.GLCMPanel.Title = 'GLCM';
app.GLCMPanel.Position = [11 129 138 109];


% Create ContrastEditFieldLabel
app.ContrastEditFieldLabel = uilabel(app.GLCMPanel);
app.ContrastEditFieldLabel.HorizontalAlignment = 'right';
app.ContrastEditFieldLabel.Position = [7 52 47 22];
app.ContrastEditFieldLabel.Text = 'Contrast';


% Create ContrastEditField
app.ContrastEditField = uieditfield(app.GLCMPanel, 'text');
app.ContrastEditField.HorizontalAlignment = 'right';
app.ContrastEditField.Position = [63 52 61 22];


% Create EnergyEditFieldLabel
app.EnergyEditFieldLabel = uilabel(app.GLCMPanel);
app.EnergyEditFieldLabel.HorizontalAlignment = 'right';
app.EnergyEditFieldLabel.Position = [8 22 47 22];
app.EnergyEditFieldLabel.Text = 'Energy';


% Create EnergyEditField
app.EnergyEditField = uieditfield(app.GLCMPanel, 'text');
app.EnergyEditField.HorizontalAlignment = 'right';
app.EnergyEditField.Position = [64 22 61 22];


% Create ColorHistogramPanel
app.ColorHistogramPanel = uipanel(app.FeatureExtractionPanel);
app.ColorHistogramPanel.Title = 'Color Histogram';
app.ColorHistogramPanel.Position = [12 8 182 109];


% Create PlotRedHistogramButton
app.PlotRedHistogramButton = uibutton(app.ColorHistogramPanel, 'push');
app.PlotRedHistogramButton.ButtonPushedFcn = createCallbackFcn(app,
@PlotRedHistogramButtonPushed, true);
app.PlotRedHistogramButton.Enable = 'off';
app.PlotRedHistogramButton.Position = [25 61 120 22];
app.PlotRedHistogramButton.Text = 'Plot Red Histogram';


% Create PlotGreenHistogramButton
app.PlotGreenHistogramButton = uibutton(app.ColorHistogramPanel, 'push');
app.PlotGreenHistogramButton.ButtonPushedFcn = createCallbackFcn(app,
@PlotGreenHistogramButtonPushed, true);
app.PlotGreenHistogramButton.Enable = 'off';
app.PlotGreenHistogramButton.Position = [19 34 131 22];
app.PlotGreenHistogramButton.Text = 'Plot Green Histogram';
```

```
% Create PlotBlueHistogramButton
app.PlotBlueHistogramButton = uibutton(app.ColorHistogramPanel, 'push');
app.PlotBlueHistogramButton.ButtonPushedFcn = createCallbackFcn(app,
@PlotBlueHistogramButtonPushed, true);
app.PlotBlueHistogramButton.Enable = 'off';
app.PlotBlueHistogramButton.Position = [25 6 122 22];
app.PlotBlueHistogramButton.Text = 'Plot Blue Histogram';


% Create ShapePanel
app.ShapePanel = uipanel(app.FeatureExtractionPanel);
app.ShapePanel.Title = 'Shape';
app.ShapePanel.Position = [160 127 125 109];


% Create AreaEditFieldLabel
app.AreaEditFieldLabel = uilabel(app.ShapePanel);
app.AreaEditFieldLabel.HorizontalAlignment = 'right';
app.AreaEditFieldLabel.Position = [13 60 31 22];
app.AreaEditFieldLabel.Text = 'Area';


% Create AreaEditField
app.AreaEditField = uieditfield(app.ShapePanel, 'text');
app.AreaEditField.HorizontalAlignment = 'right';
app.AreaEditField.Position = [53 60 61 22];


% Create WidthEditFieldLabel
app.WidthEditFieldLabel = uilabel(app.ShapePanel);
app.WidthEditFieldLabel.HorizontalAlignment = 'right';
app.WidthEditFieldLabel.Position = [8 35 36 22];
app.WidthEditFieldLabel.Text = 'Width';


% Create WidthEditField
app.WidthEditField = uieditfield(app.ShapePanel, 'text');
app.WidthEditField.HorizontalAlignment = 'right';
app.WidthEditField.Position = [53 35 61 22];


% Create HeightEditFieldLabel
app.HeightEditFieldLabel = uilabel(app.ShapePanel);
app.HeightEditFieldLabel.HorizontalAlignment = 'right';
app.HeightEditFieldLabel.Position = [4 10 40 22];
app.HeightEditFieldLabel.Text = 'Height';


% Create HeightEditField
app.HeightEditField = uieditfield(app.ShapePanel, 'text');
app.HeightEditField.HorizontalAlignment = 'right';
app.HeightEditField.Position = [53 10 61 22];
```

```
% Create ExtractFeatureButton
app.ExtractFeatureButton = uibutton(app.FeatureExtractionPanel, 'push');
app.ExtractFeatureButton.ButtonPushedFcn = createCallbackFcn(app,
@ExtractFeatureButtonPushed, true);
app.ExtractFeatureButton.Enable = 'off';
app.ExtractFeatureButton.Position = [204 61 100 22];
app.ExtractFeatureButton.Text = 'Extract Feature';


% Create KNNClassificationPanel
app.KNNClassificationPanel = uipanel(app.FishClassificationAppUIFigure);
app.KNNClassificationPanel.Title = '4. KNN Classification';
app.KNNClassificationPanel.Position = [356 11 245 266];


% Create PredictedFishEditFieldLabel
app.PredictedFishEditFieldLabel = uilabel(app.KNNClassificationPanel);
app.PredictedFishEditFieldLabel.HorizontalAlignment = 'right';
app.PredictedFishEditFieldLabel.Position = [75 215 82 22];
app.PredictedFishEditFieldLabel.Text = 'Predicted Fish';


% Create PredictedFishEditField
app.PredictedFishEditField = uieditfield(app.KNNClassificationPanel, 'text');
app.PredictedFishEditField.Position = [30 194 173 22];


% Create PredictusingKNNClassifierButton
app.PredictusingKNNClassifierButton = uibutton(app.KNNClassificationPanel,
'push');
app.PredictusingKNNClassifierButton.ButtonPushedFcn = createCallbackFcn(app,
@PredictusingKNNClassifierButtonPushed, true);
app.PredictusingKNNClassifierButton.Enable = 'off';
app.PredictusingKNNClassifierButton.Position = [33 14 166 22];
app.PredictusingKNNClassifierButton.Text = 'Predict using KNN Classifier';


% Create Image_3
app.Image_3 = uiimage(app.KNNClassificationPanel);
app.Image_3.Position = [16 43 204 142];


% Create SVMClassificationPanel
app.SVMClassificationPanel = uipanel(app.FishClassificationAppUIFigure);
app.SVMClassificationPanel.Title = '5. SVM Classification';
app.SVMClassificationPanel.Position = [614 11 246 266];


% Create PredictedFishEditField_2Label
app.PredictedFishEditField_2Label = uilabel(app.SVMClassificationPanel);
app.PredictedFishEditField_2Label.HorizontalAlignment = 'right';
app.PredictedFishEditField_2Label.Position = [75 215 82 22];
app.PredictedFishEditField_2Label.Text = 'Predicted Fish';
```

```
% Create PredictedFishEditField_2
app.PredictedFishEditField_2 = uieditfield(app.SVMClassificationPanel,
'text');
app.PredictedFishEditField_2.Position = [30 194 173 22];

% Create PredictusingSVMClassifierButton
app.PredictusingSVMClassifierButton = uibutton(app.SVMClassificationPanel,
'push');
app.PredictusingSVMClassifierButton.ButtonPushedFcn = createCallbackFcn(app,
@PredictusingSVMClassifierButtonPushed, true);
app.PredictusingSVMClassifierButton.Enable = 'off';
app.PredictusingSVMClassifierButton.Position = [33 16 167 22];
app.PredictusingSVMClassifierButton.Text = 'Predict using SVM Classifier';

% Create Image_4
app.Image_4 = uiimage(app.SVMClassificationPanel);
app.Image_4.Position = [16 44 204 142];

% Create TrainandTestingFishdatasetPanel
app.TrainandTestingFishdatasetPanel =
uipanel(app.FishClassificationAppUIFigure);
app.TrainandTestingFishdatasetPanel.Title = 'Train and Testing Fish dataset';
app.TrainandTestingFishdatasetPanel.Position = [22 290 335 259];

% Create TrainingDataEditField
app.TrainingDataEditField = uieditfield(app.TrainandTestingFishdatasetPanel,
'text');
app.TrainingDataEditField.Position = [13 203 164 22];

% Create SelecttrainingdatasetButton
app.SelecttrainingdatasetButton =
uibutton(app.TrainandTestingFishdatasetPanel, 'push');
app.SelecttrainingdatasetButton.ButtonPushedFcn = createCallbackFcn(app,
@SelecttrainingdatasetButtonPushed, true);
app.SelecttrainingdatasetButton.Position = [175 203 145 22];
app.SelecttrainingdatasetButton.Text = 'Select training dataset';

% Create SelecttrainingresponseButton
app.SelecttrainingresponseButton =
uibutton(app.TrainandTestingFishdatasetPanel, 'push');
app.SelecttrainingresponseButton.ButtonPushedFcn = createCallbackFcn(app,
@SelecttrainingresponseButtonPushed, true);
app.SelecttrainingresponseButton.Position = [176 170 144 22];
app.SelecttrainingresponseButton.Text = 'Select training response';

% Create TrainingResponseEditField
```

```
app.TrainingResponseEditField =
uieditfield(app.TrainandTestingFishdatasetPanel, 'text');
app.TrainingResponseEditField.Position = [14 170 164 22];


% Create TestingDataEditField
app.TestingDataEditField = uieditfield(app.TrainandTestingFishdatasetPanel,
'text');
app.TestingDataEditField.Position = [15 137 164 22];


% Create SelecttestingdatasetButton
app.SelecttestingdatasetButton = uibutton(app.TrainandTestingFishdatasetPanel,
'push');
app.SelecttestingdatasetButton.ButtonPushedFcn = createCallbackFcn(app,
@SelecttestingdatasetButtonPushed, true);
app.SelecttestingdatasetButton.Position = [177 137 141 22];
app.SelecttestingdatasetButton.Text = 'Select testing dataset';


% Create SelecttestingresponseButton
app.SelecttestingresponseButton =
uibutton(app.TrainandTestingFishdatasetPanel, 'push');
app.SelecttestingresponseButton.ButtonPushedFcn = createCallbackFcn(app,
@SelecttestingresponseButtonPushed, true);
app.SelecttestingresponseButton.Position = [178 104 140 22];
app.SelecttestingresponseButton.Text = 'Select testing response';


% Create TestingResponseEditField
app.TestingResponseEditField =
uieditfield(app.TrainandTestingFishdatasetPanel, 'text');
app.TestingResponseEditField.Position = [16 104 164 22];


% Create ChooseClassifierButtonGroup
app.ChooseClassifierButtonGroup =
uibuttongroup(app.TrainandTestingFishdatasetPanel);
app.ChooseClassifierButtonGroup.Title = 'Choose Classifier';
app.ChooseClassifierButtonGroup.Position = [12 7 114 93];


% Create KNNButton
app.KNNButton = uiradiobutton(app.ChooseClassifierButtonGroup);
app.KNNButton.Text = 'KNN';
app.KNNButton.Position = [11 47 58 22];
app.KNNButton.Value = true;


% Create SVMButton
app.SVMButton = uiradiobutton(app.ChooseClassifierButtonGroup);
app.SVMButton.Text = 'SVM';
app.SVMButton.Position = [11 25 65 22];
```

```
% Create RunButton
app.RunButton = uibutton(app.ChooseClassifierButtonGroup, 'push');
app.RunButton.ButtonPushedFcn = createCallbackFcn(app, @RunButtonPushed,
true);
app.RunButton.Position = [5 3 100 22];
app.RunButton.Text = 'Run';


% Create ResultPanel
app.ResultPanel = uipanel(app.TrainandTestingFishdatasetPanel);
app.ResultPanel.Title = 'Result';
app.ResultPanel.Position = [136 7 186 93];


            % Create ValidationAccuracyEditFieldLabel
            app.ValidationAccuracyEditFieldLabel = uilabel(app.ResultPanel);
            app.ValidationAccuracyEditFieldLabel.HorizontalAlignment =
'right';
            app.ValidationAccuracyEditFieldLabel.Position = [4 41 109 22];
            app.ValidationAccuracyEditFieldLabel.Text = 'Validation Accuracy';


            % Create ValidationAccuracyEditField
            app.ValidationAccuracyEditField = uieditfield(app.ResultPanel,
'numeric');
            app.ValidationAccuracyEditField.ValueDisplayFormat = '%5.2f%%';
            app.ValidationAccuracyEditField.Position = [120 41 59 22];


            % Create TestingAccuracyEditFieldLabel
            app.TestingAccuracyEditFieldLabel = uilabel(app.ResultPanel);
            app.TestingAccuracyEditFieldLabel.HorizontalAlignment = 'right';
            app.TestingAccuracyEditFieldLabel.Position = [17 12 96 22];
            app.TestingAccuracyEditFieldLabel.Text = 'Testing Accuracy';


            % Create TestingAccuracyEditField
            app.TestingAccuracyEditField = uieditfield(app.ResultPanel,
'numeric');
            app.TestingAccuracyEditField.ValueDisplayFormat = '%5.2f%%';
            app.TestingAccuracyEditField.Position = [120 12 59 22];


            % Show the figure after all components are created
            app.FishClassificationAppUIFigure.Visible = 'on';
        end
```