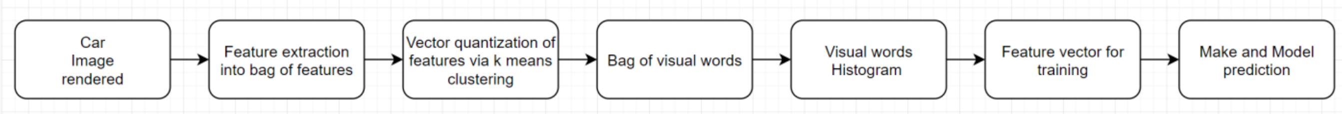
1a)



1a). **Car Image Rendered**

The system may accept either photos or videos as input, where frames from videos may be retrieved at frequent intervals depending on the hardware and software being used. The vehicle detection module checks the presence of a vehicle in the current picture and also localises the vehicle inside the picture. If there is a car detected within the bounds of the picture, it is further processed, while if not the photo is released from memory.

The model can render a range of care car images from multiple viewing angles which can cause further difficulties when trying to classify car models and makes, fortunately all the car photos in the database required to be rendered are from the same frontal view of the vehicles.

1a). **Feature Extraction into bag of features**

The next stage being the detection and extraction of features from within the car image. The feature detector algorithm implemented offers the computer better suited and interpretable features that can be used in computer vision.

The local features and their descriptors are determined, these features may be specific structures in the image such as points, edges or objects. A generic neighbourhood procedure or feature detection performed to the picture might potentially result in features.

Examples of corner detectors being FAST, Harris, and Shi & Tomasi, and blob detectors being SURF, KAZE, and MSER.

A Bag of features is then employed using the image features as the visual words that characterise the car photo, rather than real words such as in document retrieval. The bag of features is a simple encoding system that uses a small number of visual word histograms to describe a big number of pictures. An inverted index data structure allows for compact storage and effective searching.

1a) **Quantization for feature space with K-Means**

by using the k-means clustering algorithm on the feature descriptors extracted from trainingSets. The algorithm iteratively groups the descriptors into *k* mutually exclusive clusters which are dense and of similar patterns. Each cluster centres are the selected dominant feature that make up the dictionary, and are referred to as codeworks or visual words, where the number of clusters determines the overall dictionary size.

1a) **Bag of Visual Words**

The bag of visual words provided from the bag of features is used to implement images into the image set from which the histogram is visual words is created. To train the classifier, the positive and negative values from the histogram of visual words is used.

For each car photo, the approximate closest neighbour approach is used to create a feature histogram. The function then increases histogram bins depending on the descriptor's vicinity to a certain cluster centre. The length of the histogram determining the amount of visual words created.

1a) **Feature vector for training**

The classifier is a binary support vector machine(SVM) which acts as a multiclass classifier, trained using training set and the feature vectors from the bag of words.

The support vector machine algorithm's goal is to find a hyperplane in an N-dimensional space (N = the number of characteristics) that categorises data points clearly.

1a) **Make and Model Prediction**

The classifier’s performance can be measured in terms of correctly identified, incorrectly identified and missed observations. The testing dataset contains the new unseen observations which are used to determine classifier’s performance in terms of a successful recognition rate.

1b)

1c) Experiment & Results

Vocab Size	Strongest feature %	Mean Accuracy%
5000	100%	0.77
5000	80%	0.77
5000	60%	0.72
5000	40%	0.68

Vocab Size	Strongest feature %	Mean Accuracy%
2000	100%	0.76
2000	80%	0.76
2000	60%	0.7

Method	Strongest feature %	Mean Accuracy%
Vocab size = 5000, Strongest features = 1, Point Selection = Grid	100%	0.8002

The performance of the proposed method was tested and evaluated using one car datasets consisting of 27 categories, Audi A4, BMW 3 Series, BMW 5 Series, Citroen AX, Citroen Saxo, Fiat Brava, Fiat Punto, Fiat Punto New, Ford Fiesta, Ford Fiesta New, Ford Focus, Ford KA, Ford Mondeo New, Honda Civic New, Peugeot 306, Peugeot 306 New, Renault Megane, Renault Megane Coupe, Renault19, Rover25, Toyota Corolla, Toyota Yaris, Vauxhall Astra, Vauxhall Astra New, Vauxhall Vectra, VW Golf, and the VW.

This dataset contains sample frontal view images of these vehicles in different sizes and varying backgrounds while having their license plate redacted. The aim of this classification is to classify the vehicle make and model using standard machine learning methods and not neural networks. The summary for the dataset is shown in Table 1.

The image sizes within the dataset varies from 640 x 480 pixels down to 140 x 70 pixels. The images varied from being in colour to being in greyscale. All the coloured images were taken in the day time either in a well lit out side parking space or in a slightly darker parking space. The smaller images tended to be just cropped versions of the larger images. Some of the images had other cars next to them. As previously mentioned the license plate was removed from all images, Lastly a special note in the Renault19 category some of the images had circular shapes partially blocking the vehicle, which will be shown below.

Figure 3.

Then the data is divided into two partitions which are training and testing set. There is no specific rule to determine the data division of training dataset and testing dataset where In most cases, the researchers used different combinations of data division and it varies according to the problems. Common practice though has found a 70/30 or 80/20 split is best. In this study, the datasets are split into training-test partitions namely, 80-20% initially, following the Pareto principle a theory suggesting that 20% of the inputs/causes directly impact 80% of the outputs/effects . The training set contains 70% of data from each tumour which are benign and malignant tumours while another 30% of the data used for testing set. This training and validation set being randomly assigned to reduce bias within the results.

returns a bag of features, where local image features in this case SURF features are extracted from all images in all image categories and then the visual vocabulary of words is created by reducing the number of features of features through quantization of feature space using K-means clustering.

Key parameters that were chosen included the vocab size which represented the Number of visual words to include in the bagOfFeatures where the The VocabularySize value corresponds to *K* in the K-means clustering algorithm

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2$$

the visual vocabulary is created from SURF features extracted from the images training data. Different vocab sizes were tested to see how the results would change and if the difference in computational time was worth doing or not. The default vocab size being 500 this was then tested with a vocab size of 1000, 2000, 5000 and the maximum size of 8343.

The fraction of the strongest features was specified as 1 meaning it will take all the strongest features. The test was trailed with the default value of 0.8, 0.2 and 1 where it was found setting the parameter to 1 produced the best results.

There are two stages for the feature extraction the first, you select a method for picking the point locations, (SURF 'Detector' or 'Grid'), with the PointSelection property. The second stage extracts the features using the surf extractor regardless of whether the Surf detector or grid was chosen. Tests were carried out with both methods in which they both produced similar result one major difference though being the time it took to compute the grid method being significantly longer than when using surf.

* The confusion matrix for this test set is:

PREDICTED

KNOWN	audi_a4	bmw3	bmw5_new	citroen_ax	citroen_saxo	fiat_brava	fiat_punto	fiat_punto_new	ford_fiesta	ford_fiesta_new	ford_focus	ford_ka	ford_mondeo_new	honda_civic_new
audi_a4	0.50	0.00	0.08	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00
bmw3	0.00	0.40	0.15	0.05	0.00	0.00	0.00	0.05	0.05	0.00	0.00	0.00	0.00	0.00
bmw5_new	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
citroen_ax	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
citroen_saxo	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
fiat_brava	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
fiat_punto	0.00	0.00	0.00	0.00	0.00	0.00	0.79	0.00	0.00	0.00	0.00	0.05	0.00	0.00
fiat_punto_new	0.00	0.00	0.00	0.00	0.04	0.04	0.07	0.67	0.00	0.00	0.00	0.00	0.00	0.00
ford_fiesta	0.03	0.00	0.03	0.07	0.00	0.00	0.00	0.00	0.63	0.00	0.00	0.10	0.00	0.00
ford_fiesta_new	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.78	0.00	0.06	0.00	0.00
ford_focus	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00
ford_ka	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.07	0.00	0.73	0.00	0.07
ford_mondeo_new	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.92	0.00
honda_civic_new	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.69
peugeot306	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
peugeot306_new	0.00	0.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
renault19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
renault_megane	0.00	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
renault_megane_coupe	0.00	0.00	0.07	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00
rover25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00
toyota_corolla	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
toyota_yaris	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00
vauxhall_astra	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.00	0.04	0.00	0.00	0.00	0.00
vauxhall_astra_new	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00
vauxhall_vectra	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07
vw_golf3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.00
vw_polo	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.00	0.00

KNOWN	peugeot306	peugeot306_new	renault19	renault_megane	renault_megane_coupe	rover25	toyota_corolla	toyota_yaris	vauxhall_astra	vauxhall_astra_new	vauxhall_vectra	vw_golf3	vw_polo
audi_a4	0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00
bmw3	0.15	0.00	0.00	0.00	0.05	0.05	0.00	0.00	0.00	0.00	0.05	0.00	0.00
bmw5_new	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
citroen_ax	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
citroen_saxo	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
fiat_brava	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
fiat_punto	0.05	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00
fiat_punto_new	0.00	0.04	0.04	0.00	0.00	0.07	0.04	0.00	0.00	0.00	0.00	0.00	0.00
ford_fiesta	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.03
ford_fiesta_new	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.06	0.00
ford_focus	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.09	0.00	0.00	0.00	0.00	0.04
ford_ka	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00
ford_mondeo_new	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00
honda_civic_new	0.00	0.08	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
peugeot306	0.83	0.00	0.00	0.00	0.00	0.00	0.17	0.00	0.00	0.00	0.00	0.00	0.00
peugeot306_new	0.00	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.00
renault19	0.00	0.00	0.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00
renault_megane	0.00	0.00	0.00	0.82	0.00	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00
renault_megane_coupe	0.00	0.00	0.00	0.07	0.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
rover25	0.00	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00
toyota_corolla	0.00	0.09	0.00	0.00	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.00	0.00
toyota_yaris	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.82	0.00	0.00	0.00	0.00	0.00
vauxhall_astra	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.79	0.04	0.00	0.00	0.00
vauxhall_astra_new	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.88	0.00	0.00	0.00
vauxhall_vectra	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.80	0.00	0.00
vw_golf3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.93	0.00
vw_polo	0.00	0.00	0.03	0.00	0.00	0.03	0.00	0.00	0.00	0.03	0.00	0.03	0.77

* Average Accuracy is 0.80.

Confusion matrix for 'Grid', 5000 'Vocab size', 1 'Strongest features'

Grid', 5000 'Vocab size', 1 'Strongest features'			
Precision	Recall	f1 scores	
0.9375	0.5000	0.6522	
0.4074	0.2857	0.3359	
0.3989	0.4737	0.4331	
0.4174	0.5000	0.4550	
0.4743	0.5000	0.4868	
0.4909	0.5000	0.4954	
0.4659	0.4412	0.4532	
0.4250	0.4000	0.4121	
0.4730	0.3958	0.4310	
0.4223	0.4375	0.4298	
0.4428	0.4524	0.4475	
0.4199	0.4231	0.4215	
0.4813	0.4783	0.4798	
0.4561	0.4091	0.4313	
0.3779	0.4545	0.4127	
0.3983	0.4211	0.4093	
0.4676	0.4706	0.4691	
0.4480	0.4500	0.4490	
0.4831	0.4167	0.4474	
0.4492	0.4706	0.4597	
0.4193	0.4762	0.4459	
0.4490	0.4516	0.4503	
0.4700	0.4419	0.4555	
0.4795	0.4667	0.4730	
0.4615	0.4444	0.4528	
0.4223	0.4828	0.4505	Mean F1
0.4761	0.4340	0.4541	0.4516

From using 'Grid', 5000 'Vocab size', 1 'Strongest features' the mean accuracy for the test set was found to be 0.8002 which implies the classifier peforms quite well, but isn't quite up to standard in real world applications. The results shows Bmw 3 series was the vehicle it struggled to classify to the most at 40% closely followed by the Audi A4 at 50%.

In general, it is believed that having a larger dataset by increasing the number of images in each category would have been greatly beneficial to the outcome results where the dataset was rather small for such a task with so many different features to be extracted and categories to classify. Additionally, the number of categories could also be reduced.

Further tests were carried out changing the size of the dataset to confirm whether the previous statements were true, first the dataset was reduced to only 5 categories. It was found that the accuracy significantly increased to 0.8692 and categories that the classifier previously struggled to categorise when there were more categories (Audi A4 & Bmw 3 Series) it no longer struggled to classify.

```
* The confusion matrix for this test set is:

                                       PREDICTED
KNOWN           | audi_a4   bmw3   bmw5_new   citroen_ax   citroen_saxo
-----------------------------------------------------------------------
audi_a4         | 0.92      0.00   0.00       0.00         0.08
bmw3            | 0.10      0.90   0.00       0.00         0.00
bmw5_new        | 0.10      0.20   0.60       0.10         0.00
citroen_ax      | 0.08      0.00   0.00       0.92         0.00
citroen_saxo    | 0.00      0.00   0.00       0.00         1.00

* Average Accuracy is 0.87.
confMatrix = 5×5
      0.9231         0         0         0    0.0769
      0.1000    0.9000         0         0         0
      0.1000    0.2000    0.6000    0.1000         0
      0.0769         0         0    0.9231         0
           0         0         0         0    1.0000

ans = 0.8692
```

Next, tests were carried out where the dataset size was reduced, this was carried out by removing images from the dataset so that there were only five images of each make and model. The test accuracy dropped drastically to 0.52.

```
* Average Accuracy is 0.52.
```

Accuracy of classifier using vocab size of 5000

Conclusion & Further Work

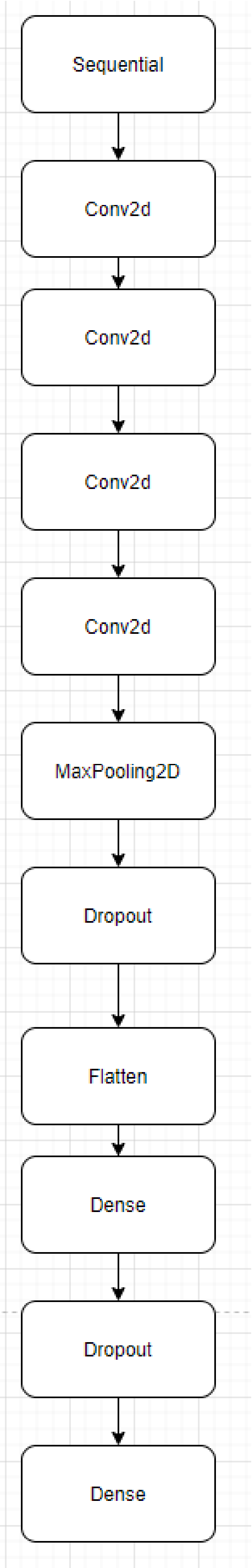
Surf feature extraction with a bag of features are quite optimal for successfully classifying make and models of vehicles for a VMMR system. Advantages of the bag of features system being its reduced storage requirements, and fast searching especially in comparison to other methods like sift due to its inverted file system for comparing vectors. Disadvantages of bag of words method being it can easily overfit.

Further work could be conducted into improving the accuracy of the network including, transfer learning, data augmentation where flipping, or translating images could vastly increase the dataset size allowing for more accurate results.

As well as this, further research that could have been carried out is using the histogram to prove Zipf's law states that given a large sample of words used, the frequency of any word is inversely proportional to its rank in the frequency table. In which word number n has a frequency proportional to 1/n.

Additionally, other methods such KAZE, MSER and Sift could have been tested. Though Research into the Sift method, it was found that it very computationally slow.

1d)


Sequential
Conv2d
Conv2d
Conv2d
Conv2d
MaxPooling2D
Dropout
Flatten
Dense
Dropout
Dense

Convolutional layers

Convolution Filters are initialized by **Gaussian distribution**, randomly. These filters are defined such that each filter learns about certain patterns, and these filters learn more patterns as the network gets deeper.

Pooling

The use of pooling is to reduce the dimension of the input image after getting convolved. There are two types, max pooling, and average pooling.

Dropout

Dropout of Individual nodes from the net with probability *1-p* or kept with probability *p* so that a reduced network is left;

A fully connected layer occupies most of the parameters, so neurons develop co-dependency amongst each other during training which leads to over-fitting of training data.

1e) data augmentation

Appendix

```
carDatabasePath = fullfile('Car Database');%% builds a full file specification from the specified folder and file names
```

```
carImageDatabase = imageDatastore(carDatabasePath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames'); %% Load the image collection using an
imageDatastore
```

```
tbl = countEachLabel(carImageDatabase);%% inspect the  car database to see
number of images per category as well as category labels
%% 27 x 2 table created
```

```
figure
montage(carImageDatabase.Files(1:64:end)) %% display some of the car images
for visual aid of whats in the dataset.
[trainingSet, validationSet] = splitEachLabel(carImageDatabase, 0.8,
'randomize'); %% randomly assigns the car database dataset into a trainingSet
and ValidationSet. with the standard ratio of 80/20 (Pareto Principle) from
each label to the new datastores. Randomising the split to reduce bias.
```

```
%%  create visual bag of features from the training data.


bag = bagOfFeatures(trainingSet, 'VocabularySize', 4500, 'StrongestFeatures',
1, 'PointSelection', 'Detector','Upright',true); %% returns a bag of features
object By default, the visual vocabulary is created from SURF features
extracted from the images training data
%% Number of visual words set as 5000, Fraction of 1 for strongest features to
use from each label, Surf Detector point selection method for picking point
locations rather than the grid default, Upright orientation of SURF feature
vector, needed due to CustomExtractor function not being called.
```

```
img = readimage(carImageDatabase, 1);
featureVector = encode(bag, img);
%% The Histogram is a reduced representation of the image is used to train a
classifier as well as the classifiaction of the image.


figure %% Plot the histogram of visual word occurrences
bar(featureVector)
title('Visual word occurrences')
xlabel('Visual word index')
ylabel('Frequency of occurrence')


vmmr_classifier = trainImageCategoryClassifier(trainingSet, bag);%% Train the
image classifier by feeding the encoded training images from each category
```

```
confMatrix = evaluate(vmmr_classifier, trainingSet);%% Evaluate the trained categoryclassifier by testing against the training set and using a confustion matrix


confMatrix = evaluate(vmmr_classifier, validationSet)%% Evaluate the trained categoryclassifier by testing against the validation set and using a confustion matrix
```

```


mean(diag(confMatrix)) %% Calculate the average accuracy.
```

```
trueP = diag(confMatrix)%%Diag produces the true positive
```

```
falseP = []; %% Create array for false positives
falseN = [];%% Create array for false negatives
for i = 1:length(trueP)%% for loop to append results
    tP = confMatrix(i);
    fP = sum(confMatrix(:,i),1) - tP;%% false positive is an outcome where the model incorrectly predicts the positive class
    fN = sum(confMatrix(i,:),2) - tP;%% false negative is an outcome where the model incorrectly predicts the negative class.

    falseP = [falseP;fP];
    falseN = [falseN;fN];
end %% note true negative isnt needed
```

```
precision = trueP ./ (trueP + falseP)%%Calculate the precision Scores
recall = trueP ./ (trueP + falseN)%%Calculate the recall Scores
f1Scores =  (2*(precision.*recall))./(precision+recall)%%Calculate the f1 Scores
meanF1 =  mean(f1Scores)%%Calculate the mean f1 Scores
```

Code references

https://uk.mathworks.com/help/vision/ug/image-category-classification-using-bag-of-features.html#ImageCategoryClassificationExample-4

https://uk.mathworks.com/help/vision/ref/bagoffeatures.html

https://uk.mathworks.com/help/vision/ug/image-retrieval-using-customized-bag-of-features.html

https://uk.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html