PREDICTIVE ANALYTICS

MSc INFORMATION SYSTEMS AND SERVICES

SPECIALIZATION: BIG DATA AND ANALYTICS

PROJECT MATLAB DEEP LEARNING

PART 3

PANAGIOTAKOPOULOS GEORGIOS (ME2030)

SUPERVISOR

FILIPPAKIS MICHAEL

Deadline: 08/07/2021

Table of contents

1. Support Vector Machine (SVM)

Using SVM will solve the binary problem of classification of persons by gender (Srivastava & Bhambhu, 2010). The model will be trained with a set of 100 face photos. For the training set, a file is provided in which the gender to which it belongs is recorded for each image. In this way the model is trained so that she can then classify a person into the appropriate sex. A total of 20 photos are provided to test the model.

The model is trained with the fitcsvm function, which can be applied to classification problems in one or two classes. In the case of work, it accepts the following two parameters:

trainData: This is a table with photos of people used for training.

Train_attr: This is a text file, in which each line indicates the gender of the corresponding photo in the trainData table (0 for male, 1 for female). Indicatively, its form is:

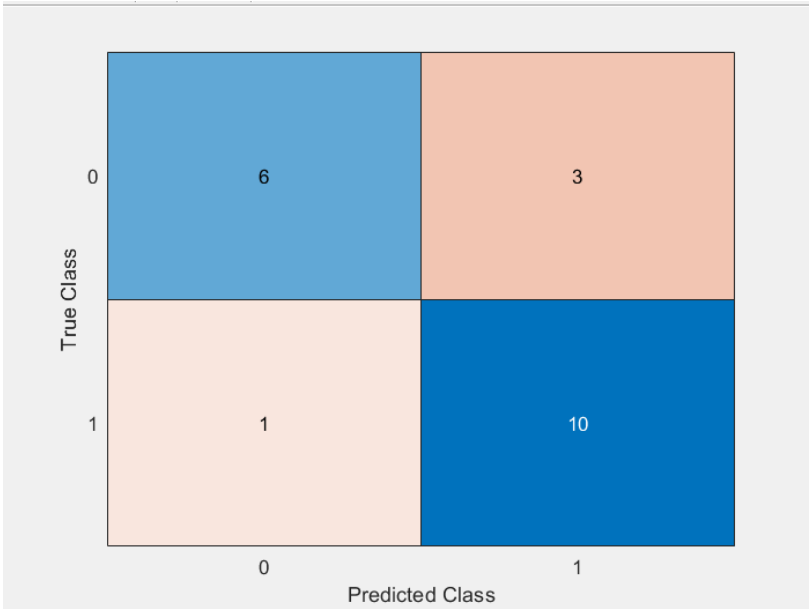
```
0.0000
1.0000
0.0000
1.0000
1.0000
1.0000
0.0000
```

1.1. Execution results of the default model

The model was initially run using the default fitcsvm function, as in the following line:

```
classifier = fitcsvm(trainData,train_attr)
```

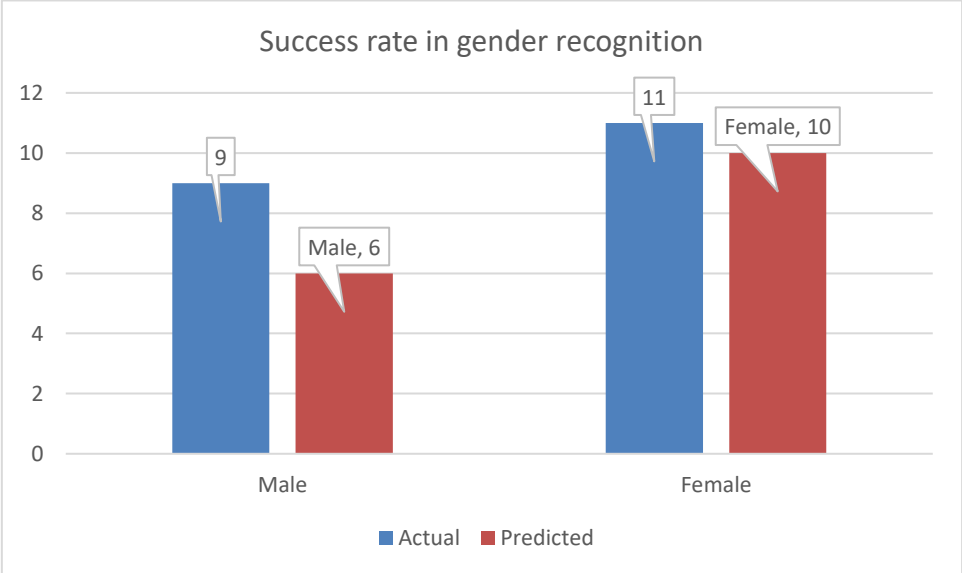
The results the classifier gave, are recorded in the following confusion chart:



The corresponding confusion matrix is as follows:

	Predicted Male	**Predicted Female**	**Totals**
Actual male	6	3	9
Actual female	1	10	11

In conclusion, the model correctly identified 6 males and 10 females. Graphically, the result is shown in the following chart:



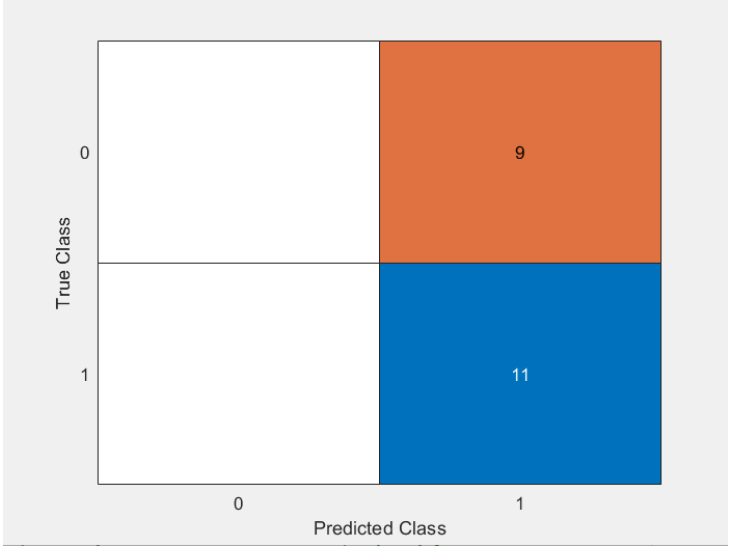
1.2. RBF kernel execution results

The fitcsvm function has the ability to modify its operation through the kernel change mechanism. In this way the forecast data is mapped using additional features, such as those created by the kernel function.

The syntax of the fitcsvm classifier for using the kernel function RBF is as follows:

```
classifier = fitcsvm(trainData,train_attr,'KernelFunction','rbf','BoxConstraint',5)
```

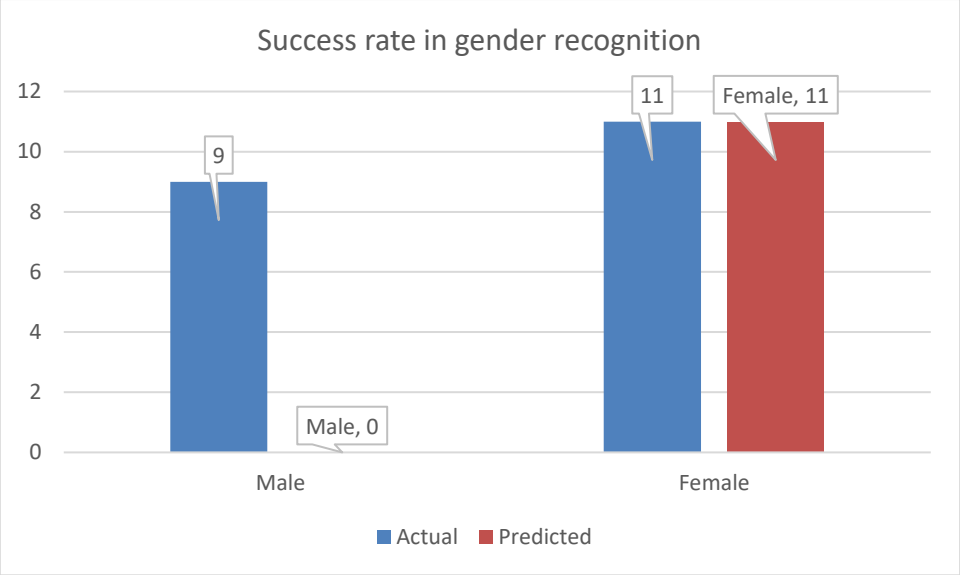
The results of the model are recorded in the next confusion chart:



The corresponding confusion matrix is the following:

	Predicted Male	**Predicted Female**	**Totals**
Actual male	0	9	9
Actual female	0	11	11

The model did not work properly, because it classified all the images of faces in the female gender. This led to the correct identification of all 11 female faces (100% success), but completely failed in the identification of the male faces. This is recorded in the chart below:

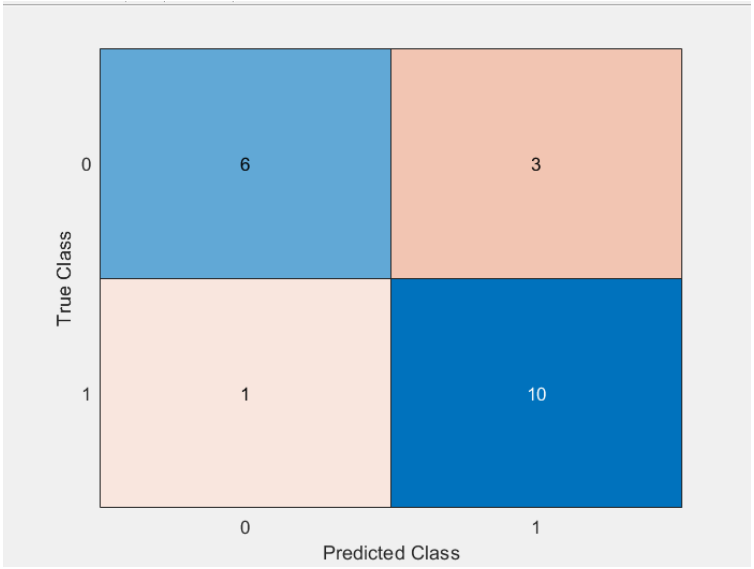


1.3. Execution results with LINEAR kernel

Another available kernel of the fitcsvm function is LINEAR. The syntax of the classifier is as follows:

```
classifier = fitcsvm(trainData,train_attr,'KernelFunction','linear')
```

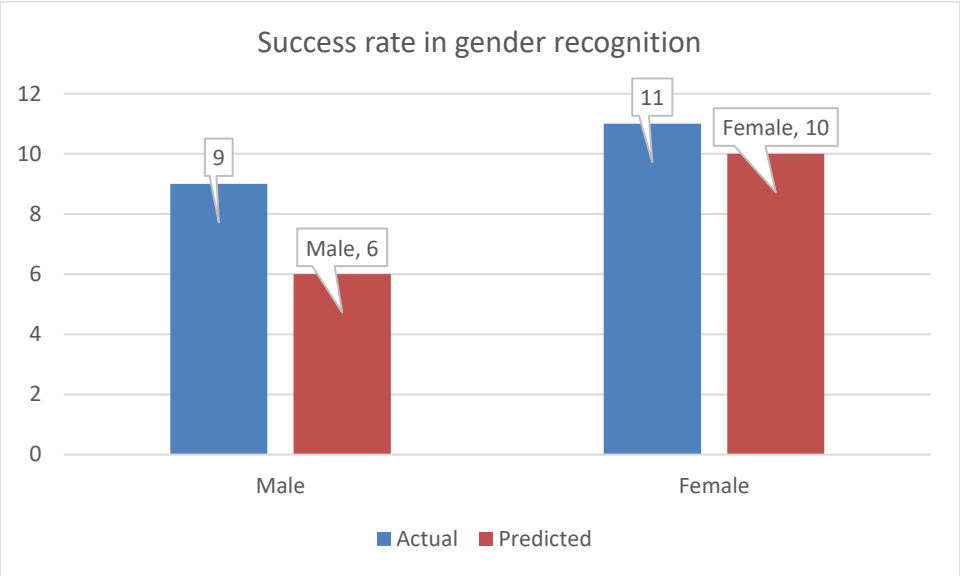
The results the classifier gave, are recorded in the following confusion chart:



The confusion matrix is:

	Predicted Male	Predicted Female	Totals
Actual male	6	3	9
Actual female	1	10	11

In conclusion, the model correctly identified 6 males and 10 females. Graphically, the result is shown in the following chart:



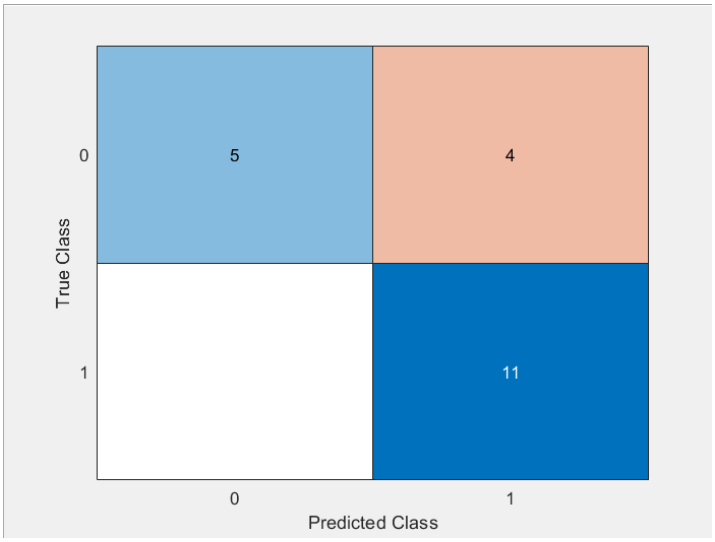
The results of the LINEAR kernel are exactly the same as the default fitcsvm classifier.

1.4. Execution results with POLYNOMIAL kernel

The implementation of the kernel function POLYNOMIAL is done with the following command:

```
classifier = fitcsvm(trainData,train_attr,'KernelFunction','polynomial')
```

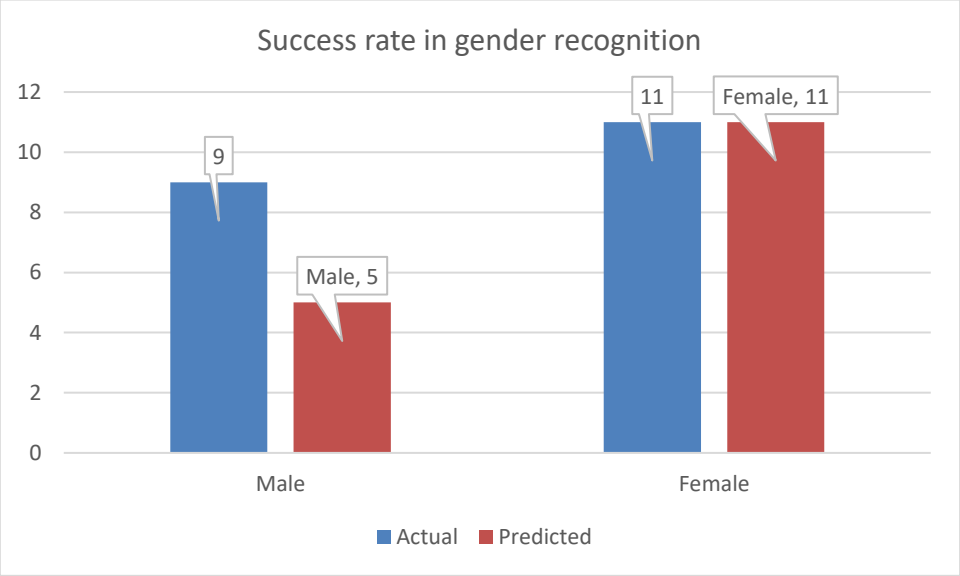
The results are recorded in the next confusion chart:



The confusion matrix is:

	Predicted Male	**Predicted Female**	**Totals**
Actual male	5	4	9
Actual female	0	11	11

This kernel function has proven to be particularly effective in identifying females. In conclusion, the model correctly identified 11 females and 5 males. Graphically, the result is shown in the following graph:

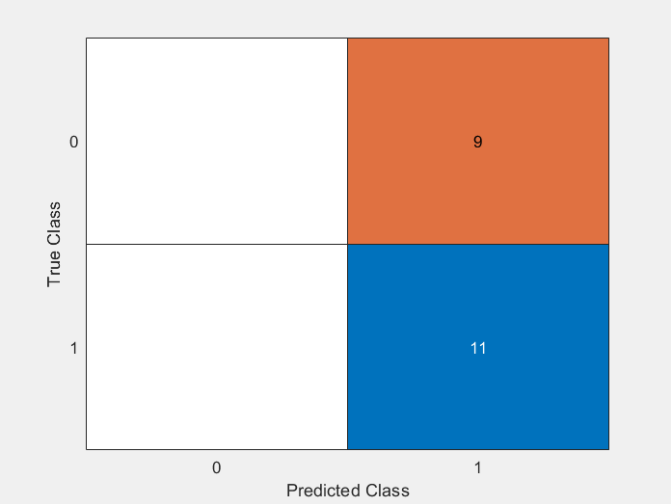


1.5. Execution results with GAUSSIAN kernel

The syntax of the fitcsvm classifier for using the GAUSSIAN kernel function is as follows:

```
classifier = fitcsvm(trainData,train_attr,'KernelFunction','gaussian')
```

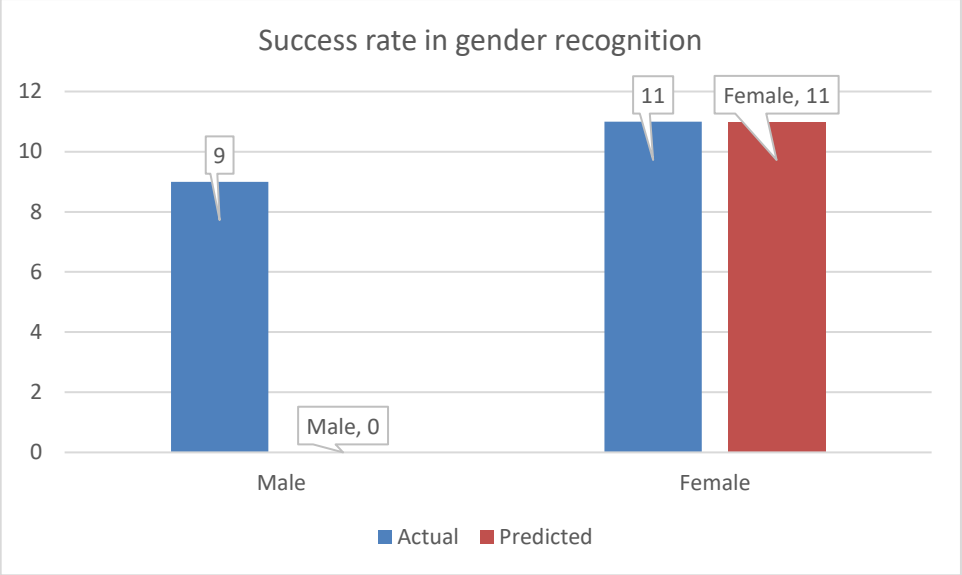
The result of the model is recorded in the next confusion chart:



The corresponding confusion matrix is as follows:

	Predicted Male	**Predicted Female**	**Totals**
Actual male	0	9	9
Actual female	0	11	11

The model did not work properly, because it classified all the images of faces in the female gender. This led to the correct identification of all 11 female faces (100% success), but completely failed in the identification of the male faces. This is recorded in the chart below:


Success rate in gender recognition
12
10
8
6
4
2
0
9
Male, 0
11
Female, 11
Male
Female
Actual
Predicted

2. Convolutional Neural Network (CNN)

Using the neural network, a corresponding model training cycle is performed using the same 100 images of male and female faces. The neural network consists of the following layers:

```
imageInputLayer([256 256 1])

convolution2dLayer(3,8,'Padding','same')
batchNormalizationLayer
reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,16,'Padding','same')
batchNormalizationLayer
reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,32,'Padding','same')
batchNormalizationLayer
reluLayer

fullyConnectedLayer(2)
softmaxLayer
classificationLayer];
```

Two sets of settings (trainingOptions) were used to model the model in order to evaluate the respective neural network performance (Albawi, Abed Mohammed, & Al-Zawi). The neural network runs with the following command:

```
net = trainNetwork(trainData,categorical(train_attr),layers,options);
```

2.1. Execution with the 1st set of settings

The set of settings is as follows:

```
options = trainingOptions('sgdm', ...
  'InitialLearnRate',0.01, ...
  'MaxEpochs',4, ...
  'MiniBatchSize', 1, ...
```

```
'InitialLearnRate', 1e-3, ...
'Verbose', true, ...
'Plots','training-progress', ...
'CheckpointPath', tempdir);
```

The behavior of the neural network is illustrated for each epoch in the following diagram.

The diagram above shows the accuracy of the network, as it evolves in each of the 100 repetitions done during each epoch. The diagram below shows the percentage of losses in the correct recognition of images.

We observe that the accuracy shows large fluctuations, depending on the mixture of images that participates in each batch. In a few cases the value of accuracy reaches the value of 100%, while the percentages in the range 50% - 70% are presented with greater frequency.

In terms of loss rate, this starts at prices in the range of 10% at the beginning of the 1st epoch, and then tends to prices around 1%. A small increase is observed at the end of the 4th epoch, in the area of 2.5%.

The aggregate results per epoch are recorded in the table below:

```
Training on single GPU.
Initializing input data normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:03 |      100.00% |       0.2672 |          0.0010 |
|       1 |          50 |       00:00:05 |        0.00% |      15.5141 |          0.0010 |
|       1 |         100 |       00:00:06 |        0.00% |       0.7919 |          0.0010 |
|       2 |         150 |       00:00:07 |        0.00% |       0.7708 |          0.0010 |
|       2 |         200 |       00:00:08 |        0.00% |       0.8724 |          0.0010 |
|       3 |         250 |       00:00:09 |        0.00% |       0.8412 |          0.0010 |
|       3 |         300 |       00:00:11 |        0.00% |       1.0047 |          0.0010 |
|       4 |         350 |       00:00:12 |        0.00% |       0.8867 |          0.0010 |
|       4 |         400 |       00:00:14 |        0.00% |       1.0766 |          0.0010 |
|=========================================================================================|

accuracy =

    0.5500
```

The total accuracy of the neural network with this set of settings is 55%.

2.2. Execution with the 2nd set of settings

The 2nd set of neural network settings is as follows:

```
options = trainingOptions('sgdm', ...
   'InitialLearnRate',0.01, ...
   'MaxEpochs',4, ...
   'MiniBatchSize', 1, ...
   'InitialLearnRate', 1e-3, ...
   'Verbose', true, ...
   'Plots','training-progress', ...
   'Shuffle','every-epoch', ...
   'ValidationFrequency',30, ...
   'CheckpointPath', tempdir);
```

The behavior of the neural network with the 2nd set of settings is illustrated for each epoch in the following diagram.

We observe that the accuracy shows large fluctuations, depending on the mixture of images that participates in each batch. Compared to the 1st set of settings, the value of accuracy reaches the value of 100% in a significant number of iterations.

As for the percentage of losses, this starts from prices in the area of 13% at the beginning of the 1st epoch and escalates during the 1st epoch to prices around 1.5%. It then tends to prices around 0%. Small inoculation increases in the 2% area occur in a few repetitions of the 2nd and 3rd epoch.

The table below lists the neural network performance per epoch. Overall network accuracy is 75%.

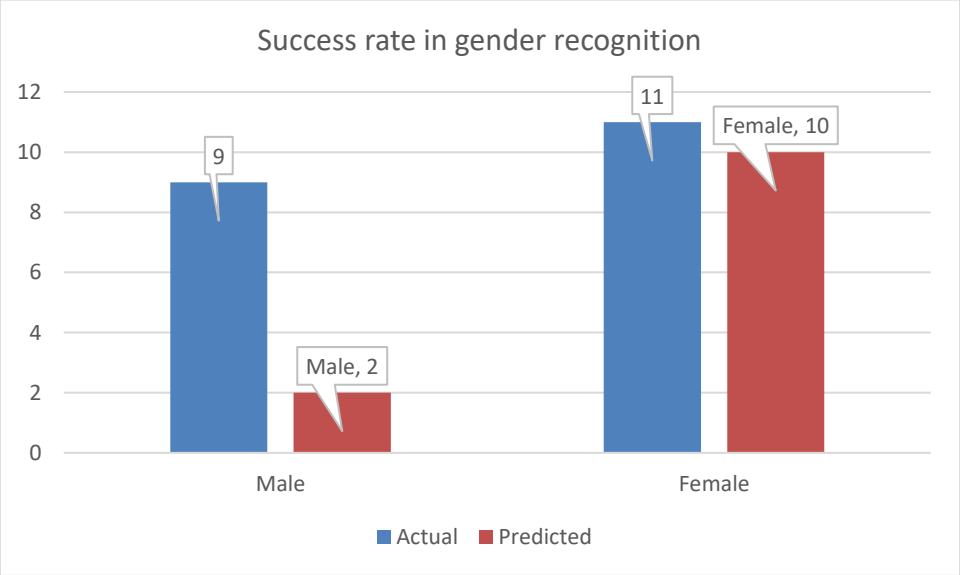
```
Training on single GPU.
Initializing input data normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:02 |        0.00% |       0.9601 |          0.0010 |
|       1 |          50 |       00:00:03 |        0.00% |      15.9424 |          0.0010 |
|       1 |         100 |       00:00:04 |        0.00% |       2.9341 |          0.0010 |
|       2 |         150 |       00:00:05 |      100.00% |       0.3677 |          0.0010 |
|       2 |         200 |       00:00:06 |      100.00% |       0.1150 |          0.0010 |
|       3 |         250 |       00:00:07 |      100.00% |       0.0547 |          0.0010 |
|       3 |         300 |       00:00:08 |      100.00% |       0.0034 |          0.0010 |
|       4 |         350 |       00:00:09 |      100.00% |       0.0808 |          0.0010 |
|       4 |         400 |       00:00:10 |      100.00% |       0.0118 |          0.0010 |
|=========================================================================================|

accuracy =

    0.7500
```

2.3. Model Resilts

Based on the classify method, the results recorded in the following confusion matrix were obtained:

	Actual	Predicted
Male	9	2
Female	11	10

Based on this table, the following diagram shows the results of the gender prediction for the test set.



3. Feature extraction

The problem is locating specific features on people's faces. This technique accepts an image as input (and possibly the definition of one or more Regions Of Interest (ROI)) and graphically marks the location of the ROIs that were searched. The result is returned as in the figure below:

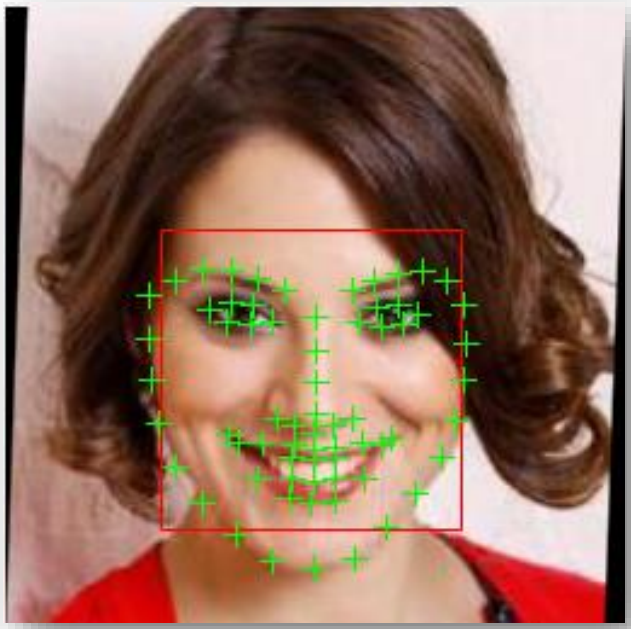


Figure 1. Marking Region Of Interest (ROIs) points on a person's photo, using the ***dlib*** *library.*

The purpose is to identify important structural elements of the face, using methods of predicting shapes. Identifying areas of the face is a two-step process:

1. Locating the face within the overall subject of the photo..
2. Recognition of the main elements of the morphology of the face (such as eyes, mouth, nose, jaw, etc.).

Since in the dataset the photos have only human faces, the first step can be bypassed in this application. The problem therefore lies in locating the frame that encloses the face (bounding rectangle, the red box in Figure 1). Once this framework is identified,

the second step of the method can be applied, which is the identification of the main morphological elements of the face.

All morphological analyzers have the ability to identify the following areas on a person:

- Mouth
- Right eyebrow
- Left eyebrow
- Right eye
- Left eye
- Nose
- Jaw

The dlib library used in this work was based on the work entitled "One Millisecond Face Alignment with an Ensemble of Regression Trees" (Kazemi & Sullivan, 2014).

3.1. How It Works

The detection method is based on training the model using a set of images, which depict faces. A file with the coordinates of the points enclosing the areas of interest (ROI) of each image is also provided.

The model is also updated with the possibility of two points of the face being at a certain distance from each other.

Using this training data, a set of regression trees is created, which are used to calculate the morphological features of the face, using the pixel density on the image.

3.2. Dlib library detector operation

The dlib library has a trained facial morphology analyzer. This analyzer divides the face into 68 small areas, which correspond to morphological structures of the face. The correspondences of these 68 areas with the corresponding structures of the face are typically shown in Figure 2.

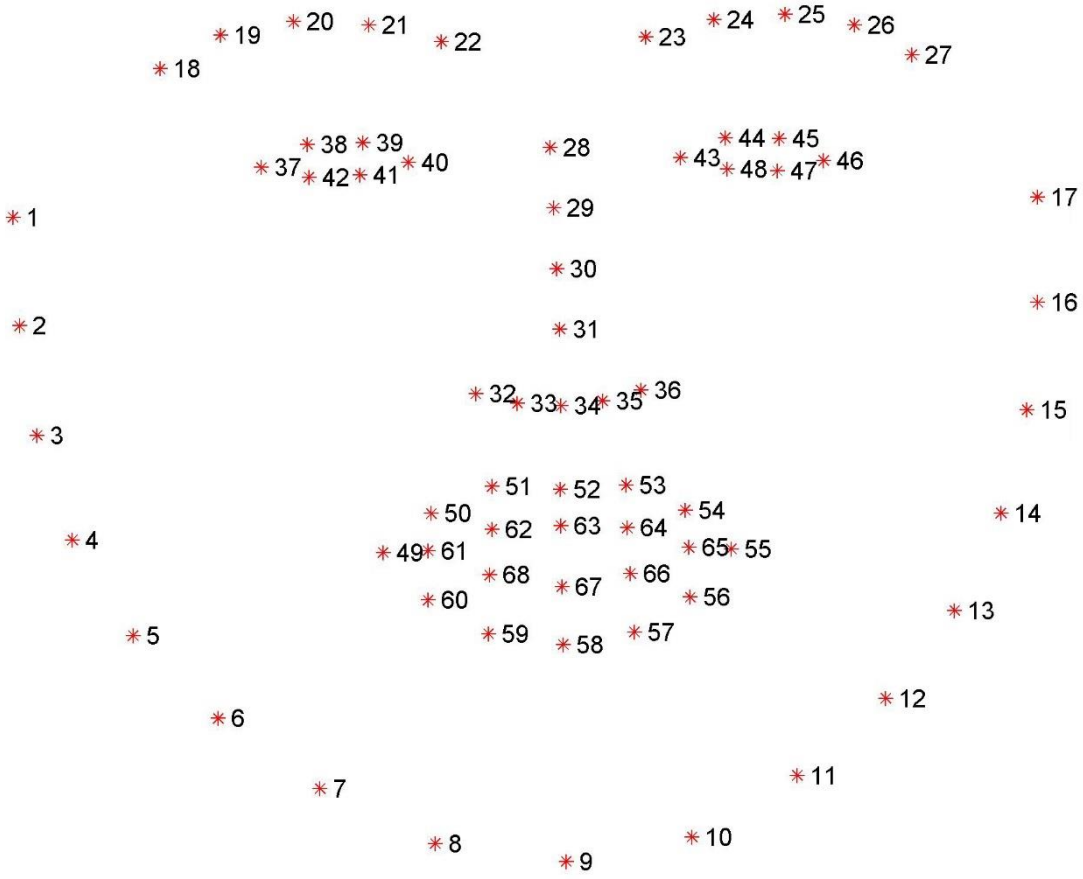


Figure 2. Numbering points used by dlib parser to locate Regions Of Interest on a person's image

Based on this mapping, each Region Of Interest (ROI) corresponds to a set of points. Analytically:

Mouth	49-68
Right eyebrow	23-27
Left eyebrow	18-22
Right eye	43-48
Left eye	37-40
Nose	28-36
Jaw	1-17

These mappings are recorded in the **shape_predictor_68_face_landmarks.dat** binary file, which is passed as a parameter to the "detector" class during the construction of the corresponding object, as shown in the code:

```
% Import predictor
h = detector('new', 'shape_predictor_68_face_landmarks.dat');
```

3.3. Execution of the model

Using Matlab, the detector is executed for each image of the test set, and the program displays in a window the corresponding image of the face, having marked the points of interest with green crosses (Figure 3).

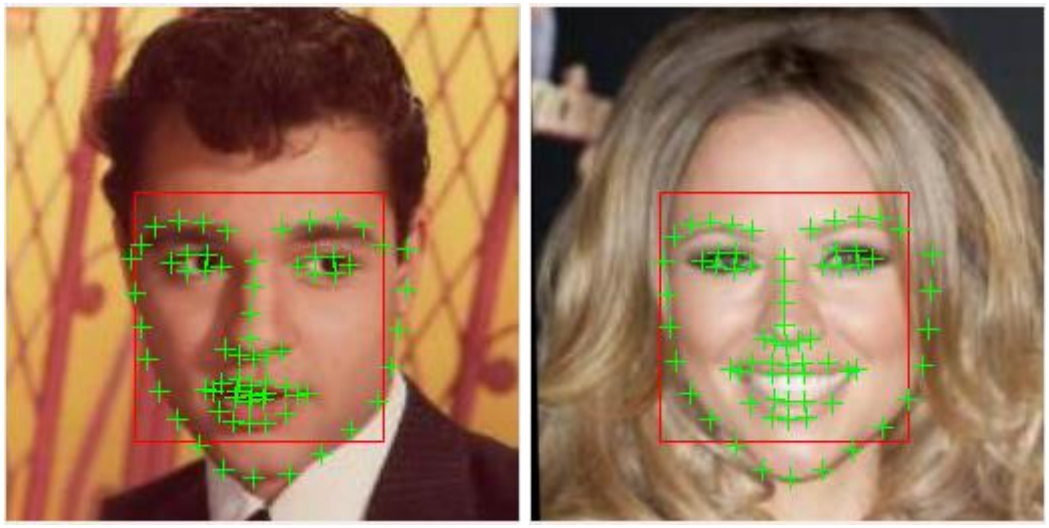


Figure 3. Result of the recognition of ROIs in the image of the face of a man and of a woman

The model accurately identifies the positions of morphological features on the faces of both men and women.

Bibliographical References

Albawi, S., Abed Mohammed, T., & Al-Zawi, S. (χ.χ.). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, (σσ. 1-6).

Kazemi, V., & Sullivan, J. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees. Columbus, Ohio, USA: Computer Vision and Pattern Recognition.

Srivastava, D., & Bhambhu, L. (2010). Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, 1-7.