Artificial Intelligence

Assignment 2

BS CS 6 Section B

Sukkur iba university

Contents

[Group Information 2](#_Toc76760225)

[Introduction to dataset 3](#_Toc76760226)

[Methodology 4](#_Toc76760227)

[Preprocessing 4](#_Toc76760228)

[Construction of Model 5](#_Toc76760229)

[Model compilation and training 5](#_Toc76760230)

[Evolution 6](#_Toc76760231)

[Testing 6](#_Toc76760232)

[Test 1 6](#_Toc76760233)

[Test 2 6](#_Toc76760234)

[Test 3 7](#_Toc76760235)

[Test 4 7](#_Toc76760236)

[Test 5 8](#_Toc76760237)

[Test 6 8](#_Toc76760238)

# Group Information

|  |  |
| --- | --- |
| Member Name | Contribution |
| Nageeta Kumari (023-18-0011) | 1. Construct the CNN Model with deployment and loading 2. Observation on different parameters |
| Syed Rizwan Ahmad Shah(053-18-0019) | 1-Dividing into training and testing sets   1. 2- Generating batches of images for training, testing and validation |
| Muskan Batra (023-18-0023) | 1. 1- Organizing Training and Testing Data frames 2. 2- Observing the images, according to size, and position |

# Introduction to dataset

There are many models that solve many problems of real world, also there are many models that solve same problems but differ in accuracy. Solving problems is important but accuracy in solving problems also matters. We have build model that recognize gender from image of human eyes. Our model will take image of eyes as input then process it and give output as male or female. For building such model we must have data set of both type of image of eyes (taken from kaggle). For creating model for image Convolutional neural network (CNN) best fits, because for image processing CNN have high accuracy. Also, CNN have different parameters that are affects the accuracy of model. Like epochs, neurons per layer and number of hidden layers etc.

We took Model from Kaggle and our purpose is to increase the accuracy of previous model by understanding it.

As our model is used to process images so we have to use different computer vision libraries like: OpenCV, Pillow and seaborn-image, etc. and before train CNN model we have to perform pre-processing on our dataset of images for achieving high accuracy. We have two type categories in dataset first images of male eye and images of female eye this means we have only two classes one is male and second is female. Total images are: 115252, male images are: 5202 and female images are: 6323.

# Methodology

We are dealing with two categories of data first set of images of male eye and second set of images of female eye.

First of all, we had loaded both types of images separately in the form list, then had mapped labels with paths of images respectively.

Made series of labels and paths of both types of images.

Concatenated the labels both series and saved in single dataframe.

Data-frame is organized but it may require preprocessing so, observe data frame for pre-processing. Graphs helped to figure out preprocessing techniques need in this dataframe. Plot bar-plot for finding number of images in each category.

*Note: Data frame have only two columns “Label” and “File”.*

*Label defines type of image i.e. male eye or female eye and File contain path of image.*

## Preprocessing

**Split data in train and test set:**

As data-frame is organized, split data into training set of images and testing set of images with the ***ratio*** *of**80:20*respectively. This means we had used 80% of images in training and 20% of images in testing.

**Label Encoder:**

As column named ‘Label’ have categorical data, it is train and test model on categorical data so, we need to convert it into numerical data type. Using Label Encoder categorical data of column Label is converted into numerical data.

**Generating Batches of Image and Normalization:**

We had used” **ImageDataGenerator**” class that generates batches of images for increasing training images. For testing images we had used Normalization.

**Division of Images using flow\_from\_dataframe function:**

Now we had prepared datasets for training, testing and validation using ‘’**flow from dataframe’’.**

**Creating callbacks for model:**

In last we built **callbacks** for our model in order to monitoring the model and when there is no improvement is observed in model for more than n times then training will be ended, also when training set is interrupted then callback restores weights that are best from available data trained.

## Construction of Model

We have used CNN model to classify gender using human eyes. In the first convolutional layer we have used 32 filters with the kernel size of 3 by 3 and used **relu** as the activation function.

After that there is a pooling layer where we add a filter of size 2 by 2. We have 2 hidden layers in our final model, in first hidden layer we have 32 filters of size 3 by 3 while in second we have 64 filters with size 3 by 3 with activation function **relu** and pooling layer with filter size of 2 by 2.

After that we used flattening technique to transform the matrix to a vector to enter the Artificial Neural Network layer then input it to input layer of dense neural network. In input layer of dense neural network we have added 128 neurons with activation function **relu**, while in output layer we have used sigmoid as activation function.

## Model compilation and training

To compile our CNN model we have used adam as optimizer, and binary\_crossentropy as a loss function. To train our model we have done different observations. First it is trained on 50 epochs with one hidden layer and .75 split with 92% validation accuracy and 89% testing accuracy. Later we have added another layer and validation accuracy increases to 93%and testing accuracy increased to 90%, we add another layer but accuracy remains same. Later we make .80 test train split and also increase the number of epcos to 100 and random state to 130, then our validation accuracy increases to 94% and testing to 93%. Another time we make change in epocs to 200, then we get validation accuracy to 93% and testing to 95%.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test\_train\_split | Epochs | Validation Accuracy (%) | Testing Accuracy | Number of hidden layer |
| .75 | 50 | 92 | 89 | 1 |
| .75 | 50 | 93 | 90 | 2 |
| .75 | 50 | 94 | 90 | 3 |
| .80 | 100 | 94 | 93 | 2 |
| .80 | 200 | 93 | 95 | 2 |

# Evolution

After loading the deployed system, we have tested it on some random images from our dataset as well as from outside sources.

## Testing

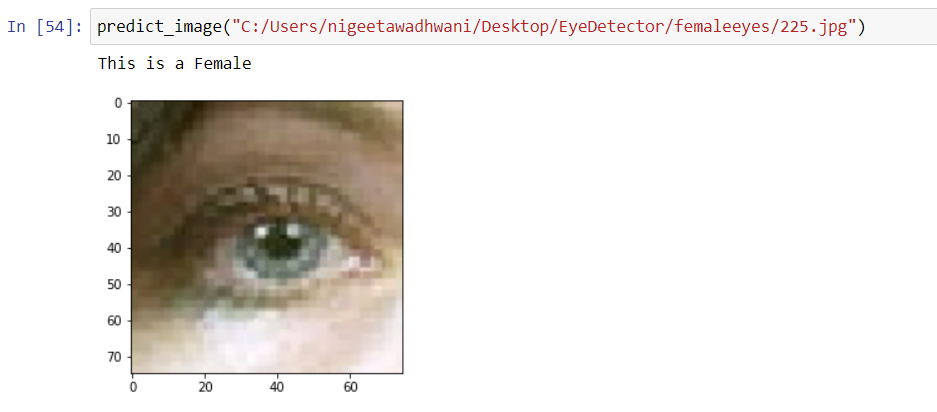
### Test 1

Here we have taken an input image from dataset of male eye, which is classified as male eyes.



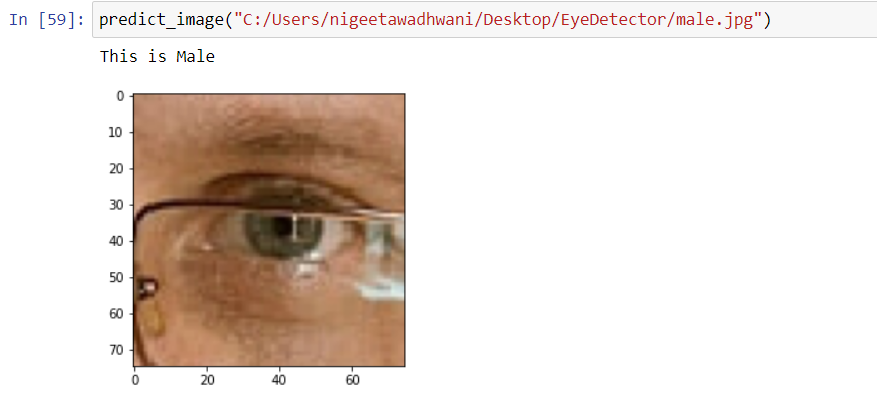
### Test 2

Here we have taken another sample from our dataset of female eye, which is classified as female eyes.



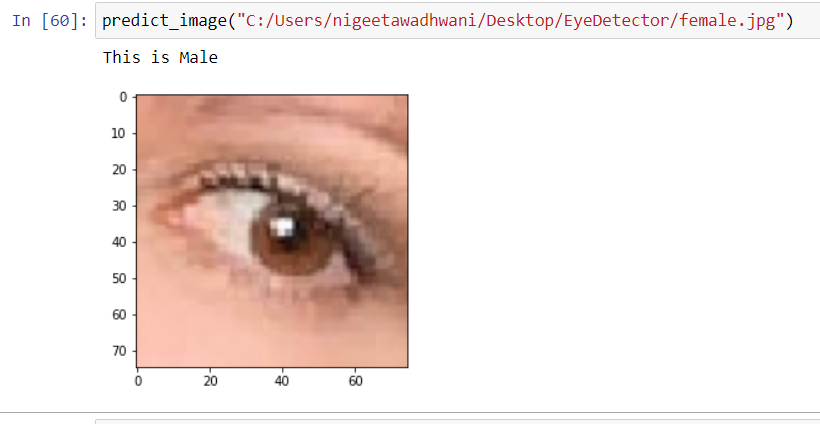
### Test 3

Here we have taken an image of male eye from another source, and it is classified as male.



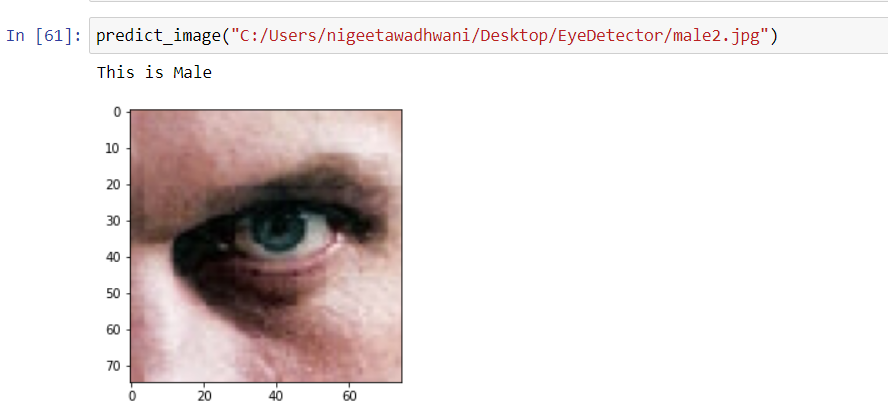
### Test 4

Here we have taken another image of female from outside source, which is not classified accurately.



### Test 5

Another male image from outside source, classified correctly.



### Test 6

Female eye from another source classified correctly

