Image Recognition Web Application

Sharmila kanthaiya Srinivasan
Texas State university
University in San Marcos, Texas
s_k309@txstate.edu

Abstract

Face recognition from images or videos is a popular topic in Computer Vision. This technology is used in surveillance cameras in grocery stores, tagging friends on social media, security measures on smartphones, and in many other places. There are many benefits in using facial recognition like increasing security, preventing crimes, and medical efforts. In this technology, accuracy and speed of identification is the main issue.

This paper aims to create a web application that can evaluate images and videos to perform image detection to provide a complete solution for image recognition with higher speed and accuracy. The application is based on the data and machine learning model. Given an image or video, the application would recognize the person's facial structure and gender.

1. Introduction

Let's start by thinking about how important vision can be. Image is the highest band sense, and it provides a vast set of information about the state of the world and acts on it. Based on this, computer vision has been born.

The goal of computer vision is the ability to extract high-level understanding from digital images and video. We all know that computers and smartphones are good at taking pictures. But can it see or recognize the image? Images are stored as big grids of pixels on the computer. Each pixel is defined by color and kept as a combination of Red, Green, Blue. The variety of these three colors on different intensities (RGB value) will give any shade. The computer can see the image via a machine learning algorithm. The picture looks like an array of integer values based on the intensities across the color spectrum to an algorithm.

The evolution of machine learning, deep learning, and neural networks created a great revolution in this field. In recent years, computer vision has widened the tasks related to detecting and labeling objects.

For the past few years, a lot of effort and improvement into face recognition [9], and it became an efficient way for person identification [6]. There are lots of methods introduced for image detection [7, 5, 2, 3, 1] and recognition [2, 4, 8] which is considered a milestone.

1.1. Paper Organization

The following paper discusses the problem and application components used to build the website in section 2, section 3 discuss the gender recognition module and steps involved in the process. The model result and evaluation has been provided in the form of tables and charts in section 4. Section 5 provides a detailed pipeline of the model that has been built in section 4. Section 6 is a conclusion phase, and 7 is future work.

2. Problem description

Although there are lots of methods to perform image recognition. There is no single website to explain and display various image recognition activities.

In this paper, I developed a web application for image recognition that can evaluate images and videos to perform image recognition. This application is based on the data and machine learning model. Given an image or video, the application would recognize the person's facial structure and gender. The process involves various steps like grayscale conversion, face/object cropping, Eigen image creation, training machine learning model, and prediction analysis

2.1. Application Components

Machine Learning Model built upon multiple sub-steps like image processing, image analysis and preprocessing, and model training.

Image processing is a method to perform some operations on an image, to extract useful information from it using OpenCV and NumPy.

1.1 OpenCV: Open Source Computer Vision Library is an open-source computer vision and machine learning software library.

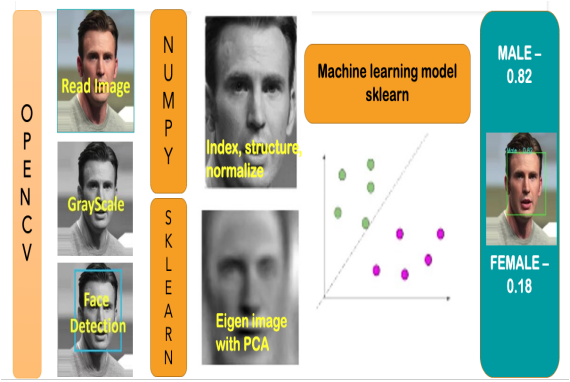

Figure 1. Gender recognition model. This displays the subsets of the model building.

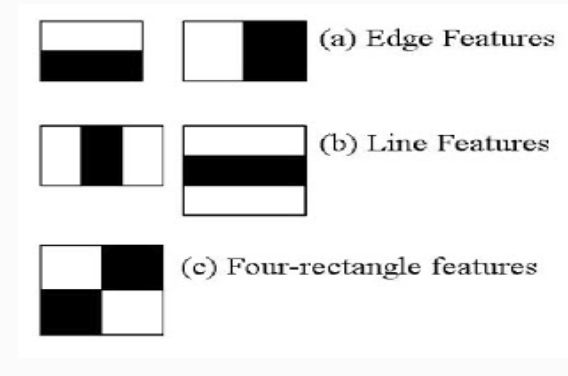

Figure 2. Haar feature-based cascade classifiers.

1.2 NumPy: NumPy stands for Numerical Python. It is an open-source module of Python which provides fast mathematical computation on arrays and matrices.

Image analysis and preprocessing will help apply some transformations to the image to convert raw data into a clean data set using Pandas and Matplotlib.

2.1 Pandas: Pandas is fast, powerful, flexible, and easy to use open- source data analysis and manipulation tool, built on top of the Python programming language.

2.2 Matplotlib: Matplotlib is a visualization library in Python for 2D plots of arrays.

To build the model, I used Scikit Learn. It is a Python module for machine learning models.

The Website is built upon the Flask web framework. Then HTML, CSS, and JavaScript are used for frontend creation.

3. Gender recognition

The first module of the image recognition website is gender prediction. Gender recognition is the process of detecting the face structure and predicting the appropriate gender of the given person in the image. Once the face structure is detected, the aim is to get a bounding box with gender and confidence scores of the person as the output.

To achieve it, I performed several sub-modules like reading image, grayscale conversion, face detection – using the OpenCV library. Then to get the confidence score and gender name, I used the machine learning model. So for that, I cropped and structured the image then normalize the image using NumPy as displayed in figure 1.

The eigenvalues of the images are computed using the principal component analysis of sklearn. This resulted in the model with a gender and confidence score.

3.1. Data Cropping

The dataset used in this paper for the training and testing model is from the data vision website. This dataset contains images in two separate folders called male and female folders. Total data are around 7000 images. Once the images are extracted and cropped, placed in appropriate new folders named male and female crop. The face cropping of an image is achieved using the Haar cascade classifier.

Haar feature-based cascade classifiers are the best and effective object detection method. It is a machine learning-based approach where a cascade function is trained from a lot of positive and negative images. The trained algorithm is used to detect objects in other images.

In this project, I used the cascade classifier for face detection. The algorithm works on a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then it extracts features from the image. For this, Haar features shown in figure 2 are used. This is similar to the convolutional kernel. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.

Each kernel's possible sizes and locations are calculated to get the features. I order to get each feature calculation, integral images are used. However, large your image reduces the calculations for a given pixel to an operation involving just four pixels.

Haar-cascade Detection in OpenCV comes with a trainer as well as a detector. OpenCV already contains many pre-trained classifiers for face, eyes, smile, etc. So once I read the image, converted it to grayscale. Then extracted facial features using Haar cascade detection in OpenCV.

3.2. Data structuring

Initially, the dataset contains unstructured images. Images are of different widths and heights. Training the model with the unstructured data gives an improper result. It is es-

Title	Size
count	6059.000000
mean	154.000660
std	71.320063
min	24.000000
25 percent	81.000000
50 percent	122.000000
75 percent	182.000000
max	410.000000

Table 1. EDA of Dataset.

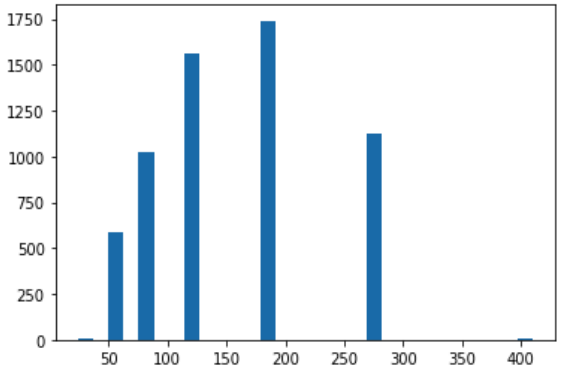

Figure 3. Histogram of bin sized 30 displays the distribution of the image size's of the dataset.

sential to structure the dataset to get a perfect model. So, I structured the images by resizing it.

To get the ideal size for resizing the image, I did exploratory data analysis by using pandas. Table 1 denotes the EDA of the dataset. It shows the size of the image ranges from the minimum width value of 24 to a maximum width of 410. The 25 percent of data are around image width of 81, 50 percent are around 122 and, 75 percent are around the dimension of 182.

Figure 3 gives the histogram representation of the dataset. It shows most of the values are above 50. At width 80, there are 1000 images. The maximum amount of data are more than a width of 100.

Generally, enlarging a small-sized image leads to poor resolution. Since the majority of the images are greater than the dimension of 54, I removed the small portion of data sized below 54. This resulted in 60 percent of the data are female images and, the remaining 40 percent are male data.

The data with a dimension of more than 60 are resized to the width of 100 and a height of 100. This is achieved by shrinking the images of width greater than equal to 100 and enlarging the images of width less than 100.

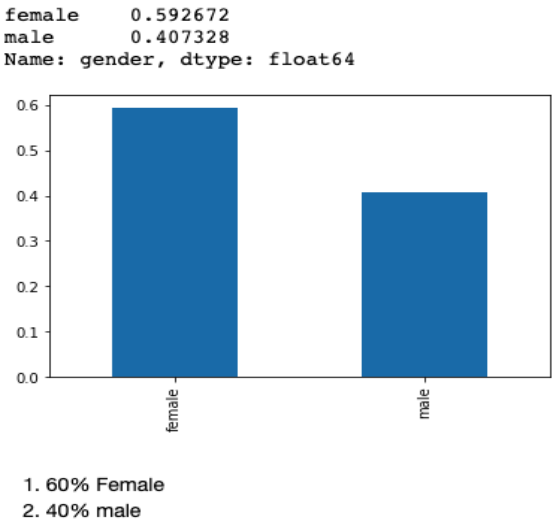

Figure 4. Histogram displays the distribution of male and female images after removing the images with size less than 60.

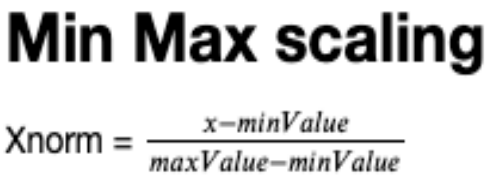

Figure 5. Normalization formula used to perform min max scaling.

3.3. Data processing

Data processing is an important step where structured images are processed to train the model. The first step of processing is flattening the image. This converts the multi-dimension array to one dimension array of the image.

I Checked for missing values from the flattened image data and removed the values with zero. Then the flattened image is normalized further to remove noise but at the same time to bring the image into a range of intensity values that is statistically a normal distribution and physically less stressful to our visual sense. The mean value will depend on the actual intensity distribution in the image, but the aim will be to ultimately state this mean value with a high confidence level. The normalized image mean value = 0 and the variance =1.

For normalizing the data, I split the data into 2 parts namely, independent and dependent features. The formula I used for min-max scaling is displayed in figure 5.

Since the images are 8 bit, min and max value ranges from 0 to 255After normalization, I got the Xnorm shape is (5461, 10000) and Y values as female and male. To train

the model I converted the male string to 0 and the female string to 1

3.4. Principal Component Analysis

Principal Component Analysis is unsupervised machine learning, it is used for dimensionality reduction problem. It is also a non-parametric statistical technique.

The dataset with a large number of features is called High dimensionality. The dataset with high-dimensionality may lead to model overfitting, which reduces the ability to generalize beyond the data in the training set. Reducing the number of components or features costs some accuracy and on the other hand, it makes the large data set simpler, easy to explore, and visualize. Also, it reduces the computational complexity of the model which makes machine learning algorithms run faster.

The first principal component expresses the most amount of variance. Each additional component expresses less variance and more noise, so representing the data with a smaller subset of principal components preserves the signal and discards the noise.

PCA uses eigenvalues to represent variance and eigenvectors represent a direction. All the values are from the covariance matrix. So larger the variance means larger information is available in a particular feature. To do the PCA calculation, I need some components. Initially, I just defined none. The PCA fit transform returned the covariance of (5461, 5461). Then using that, I computed eigenvalues and vector using the elbow method

3.5. Eigen Images

Eigen images are a technique of extracting features using Principal Component Analysis. It helps in reducing the dimensionality of the dataset. I calculated the Eigen ratio and Eigen ratio cumulative on PCA value and created an elbow chart on 200 components to understand the data.

In the elbow chart, graph 1 represents the individually explained variance of every components. The X-axis is the number of components, and the Y-axis is the variance ratio. For example, the first component is at 25 percent of the variance ratio, and the second is at 12 percent of the variance ratio.

Graph 2 of the elbow chart states cumulative analysis. The X-axis is the number of components, and the Y-axis is the cumulative explained variance ratio. If you consider only one component, you will have a 25 percent variance ratio, for two components nearly 40 percent, and so on. If you check the 50 components you can get 80 percent of the information. This result shows there is no need to look at all the component's variance ratios.

In graph 1, the optimal number of components settle down somewhere between 25 to 30 components on the X-axis. In graph 2, at 25 to 30 components you will get 70

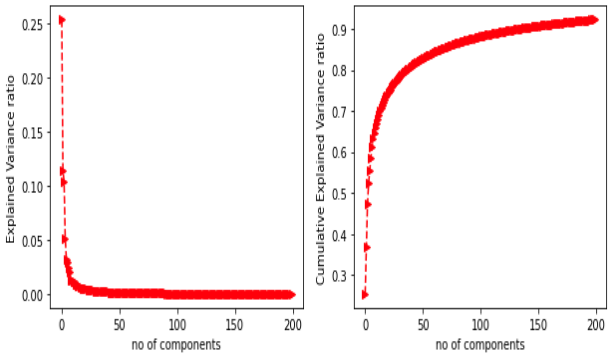

Figure 6. Elbow method results for 200 components over Explained variance ratio and Cumulative explained variance ratio.

percent of the variance on the Y-axis. So I considered a minimum of 80 percent of the variance to get better information. This resulted in 50 components.

On the analysis of the elbow method, if I consider the value between 25 to 30 components, then the explained variance is around 75 percent. So, to get a minimum of 80 percent variance, selecting 50 components works better. So this gave a reduced dimension and converted the images into Eigen images.

In figure 7, the given input images are converted into images with reduced components called Eigen images. This will help in identifying the main facial features like eyebrow, nose, and mouth position instead of calmness. Based on those, gender can be classified. Before applying the PCA, the dataset contains 10 thousand directions. After the PCA process, it was reduced to 50 directions.

4. ML model result

In this project, I trained the Machine learning model using a Support Vector Machine. Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression, and outliers detection. The advantages of support vector machines are Effective in high dimensional spaces. Still effective in cases where the number of dimensions is greater than the number of samples.

Eigen data partitioned into training and testing data using Sklearn split. The training and testing data are in the ratio of 80 percent (4368 images) and 20 percent(1093 images) respectively.

Initially, I trained the model with Support vector machine parameters C = 1.0, kernel = 'rbf', gamma = 0.01, probability = True. This resulted in the training model score of 0.8571428571428571 and testing model score of 0.8060384263494969.

Figure 7. Eigen images for random images from the dataset

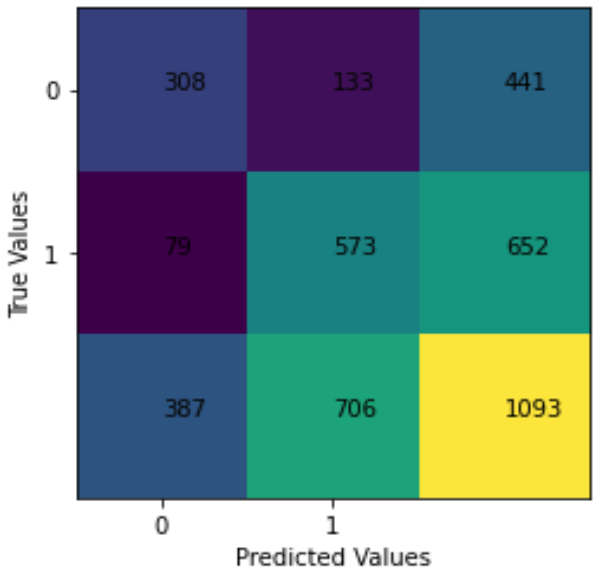

Figure 8. Confusion matrix for the ML model (C = 1.0, kernel = 'rbf',gamma = 0.01, probability = True).

	precision	recall	f1-score	support
male	0.795866	0.698413	0.743961	441.000000
female	0.811615	0.878834	0.843888	652.000000
accuracy	0.806038	0.806038	0.806038	0.806038
macro avg	0.803740	0.788624	0.793925	1093.000000
weighted avg	0.805260	0.806038	0.803570	1093.000000

Figure 9. classification report for the ML model (C = 1.0, kernel = 'rbf',gamma = 0.01, probability = True).

4.1. ML model evaluation

Model Evaluation is an essential part of the ML model building process. It helps to find the best model that represents our data and how well the chosen model will work in the future. I Evaluated the model performance with the Confusion matrix, Classification report, Kappa score , and AUC and ROC.

A confusion matrix is a table used to represent the performance of a classification model on the test data. In figure 8, the confusion matrix displays the total test data of 1093 samples. In that 441 on the top right corner are male samples. Out of 441, 317 are correctly classified as male, and the remaining 124 are misclassified. Similarly, out of 651 female samples, 557 correctly classified, 95 are misclassified. This shows 71 and 85 percent of correct classification for male and female class respectively.

A Classification report is used to measure the quality of the model predictions from a classification algorithm. The below table shows the classification metrics precision, recall, f1-score, and support on a male and female class basis. Precision denotes the accuracy of positive predictions, and it is defined as the ratio of true positives to the sum of true and false positives. In the report, male precision is 79 percent, and female is 81 percent. This states that for each class, the true positive is higher. The second criterion is Recall, it denotes the fraction of positives that were correctly identified. For each class, it is defined as the ratio of true positives to the sum of true positives and false negatives. The report shows the Recall of male is 69 percentage and female is 87 percent. Then the F1 score is a weighted harmonic mean of precision and Recall such that the best score is 1.0, and the worst is 0.0. In the report, f1-score for both classes is on the higher side. Finally, support is the number of actual occurrences of the class in the specified dataset. So based on the analysis of all the parameters of the classification report states that this model is working perfectly.

The third evaluator is the kappa score. It will provide random accuracy. In cases like imbalanced classes. E.g. we have two classes, say A and B, and A shows up on 5 percent of the time. classification report can be misleading sometimes and the measure does not have a very good intuitive explanation. So Kappa score statistic is a very good

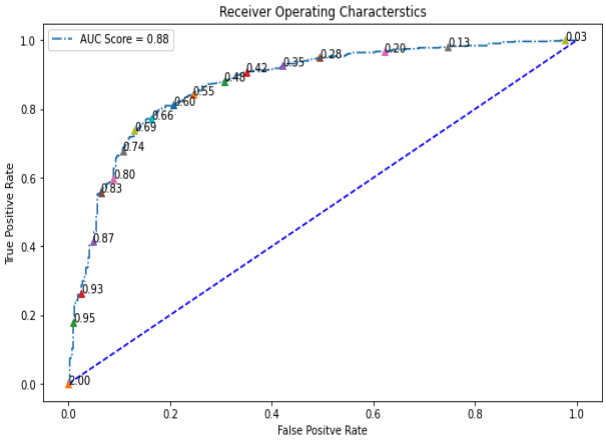

Figure 10. ROC and AUC (Probability) for the ML model (C = 1.0, kernel = 'rbf',gamma = 0.01, probability = True).

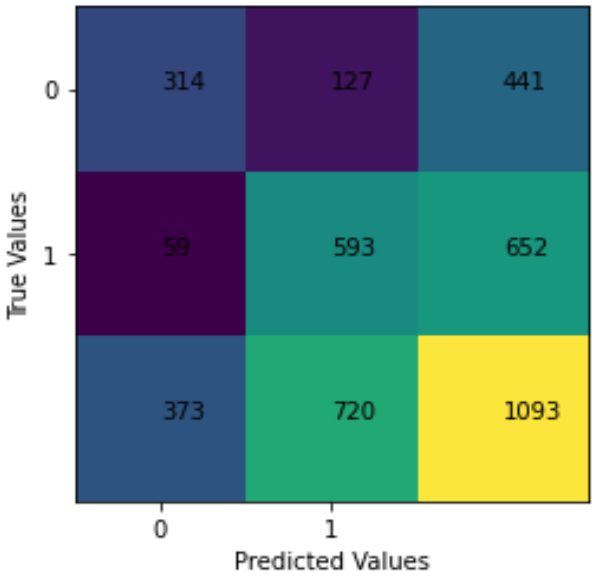

Figure 11. confusion matrix for the hyper tuned model ('C': 1, 'coef0': 0, 'gamma': 0.05, 'kernel': 'rbf').

measure that can handle imbalanced class problems well. The kappa score of the model is 57 percent - it is again near to the perfect model range.

Finally, I used ROC and AUC as the fourth model evaluator. AUC (Area Under The Curve) ROC (Receiver Operating Characteristics) curve is one of the most important evaluation metrics for checking any classification model's performance at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at distinguishing between female and male classes.

In the ROC and AUC, the blue line is the baseline and the above is the arc. At the default threshold point of 0.60, I got a false positive rate is 20 percent and a true positive is 80 percent. If you see the classification report, the recall of female class is 87 percent for 50 percent default point. In the ROC, 50 percent is near the 80 percent TPR, if I decrease it to 25 percent, then it is 95 percent TPR. This shows as the percent of the ROC curve decreases, the TPR increases. Now the AUC is 88 percent, If the AUC is greater than 80 percent, then it is a good model.

So based on the results of the model evaluation , the model is working perfectly well.

4.2. Hyper Parameter Tuning

Tuning the model is the process of increasing a model's performance without overfitting or creating a high variance. This can be achieved by selecting the appropriate "hyperparameters." Hyperparameters are significant because they control the overall behavior of the model. The main goal of performing hyperparameter tuning is to find an ideal combination of hyperparameters that can minimize a predefined loss function to give better results.

I did the grid search on the parameters of the support vector classifier to get the best hyperparameters. This resulted in 'C': 1, 'coef0': 0, 'gamma': 0.05, 'kernel': 'rbf'. Retrained the model with the best hyperparameter by changing the increasing the C (slack variable), changing the gamma coefficient and the kernel functions. This tuning resulted in the 82.9 percent model test score which is 2 percent more than the previous model test score. So hyper tuning helped in increasing the efficiency of the model.

In figure 11, confusion matrix evaluation of the hyper tuned model resulted in 71 percent of correct classification for male class and 90 percent of correct classification for female class.

Classification report of the hyper tuned model in figure 12 shows, 5 percent and 1 percent hike for male and female precision respectively. Recall of male is increased by 2 percent and female by 3 percent. Then the F1 score of the male class is also improved by 3 percent and the female class by 2 percent.

The Kappa score of the model is also risen by 4 percent which helped it to reach greater than 60 percent. If the kappa score is near 60 percent, then it is the best model to validate. In this case, it is 63 percent which shows the model can handle the extreme case and can provide a perfect result.

In figure 13, there is also a good improvement in the ROC and AUC scores. So after hyper tuning, the model became the best model to test and rely on for future work.

	precision	recall	f1-score	support
male	0.841823	0.712018	0.771499	441.000000
female	0.823611	0.909509	0.864431	652.000000
accuracy	0.829826	0.829826	0.829826	0.829826
macro avg	0.832717	0.810764	0.817965	1093.000000
weighted avg	0.830959	0.829826	0.826935	1093.000000

Figure 12. classification report for the hyper tuned model ('C': 1, 'coef0': 0, 'gamma': 0.05, 'kernel': 'rbf').

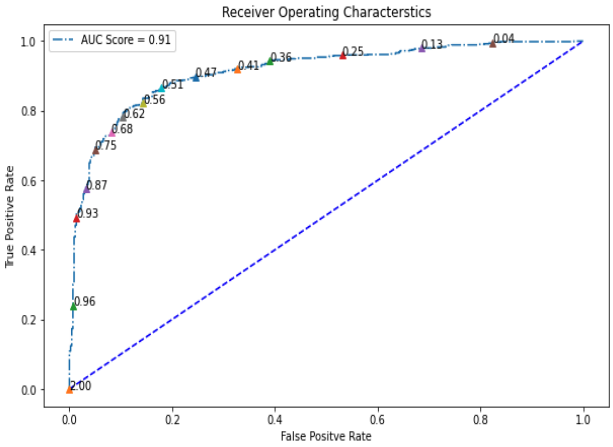

Figure 13. ROC and AUC (Probability) for the hyper tuned model ('C': 1, 'coef0': 0, 'gamma': 0.05, 'kernel': 'rbf').

5. Pipeline

The pipeline of the model is built by reading the given image and converting it into a grayscale image. Then performing the face cropping using cv2 Haar cascade classifier. This is followed by data preprocessing steps like normalizing, resizing, and flattening the given image.

Once the image is processed, then it is converted into an Eigen image to extract important features and finally, the model is built using a support vector classifies. This brings us to the end of the pipeline with the appropriate result. The digramatic representation of pipeline is explained in figure 14.

6. Result

As a result of this project, I can able to build the gender recognition module using OpenCV, NumPy, and sklearn. The process involves various steps like grayscale conversion, face/object cropping, eigen image creation, training

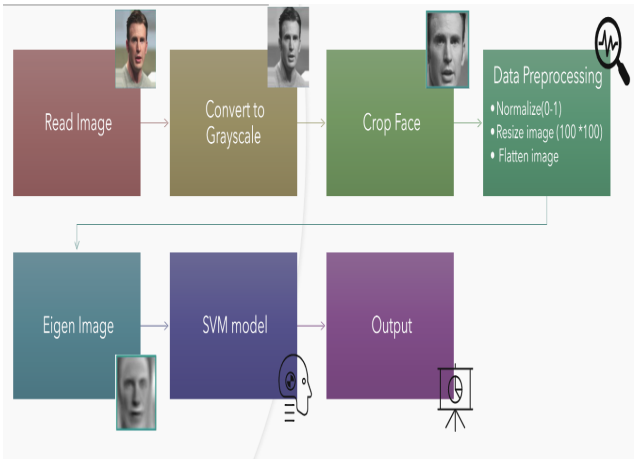

Figure 14. Pipeline for the gender recognition module

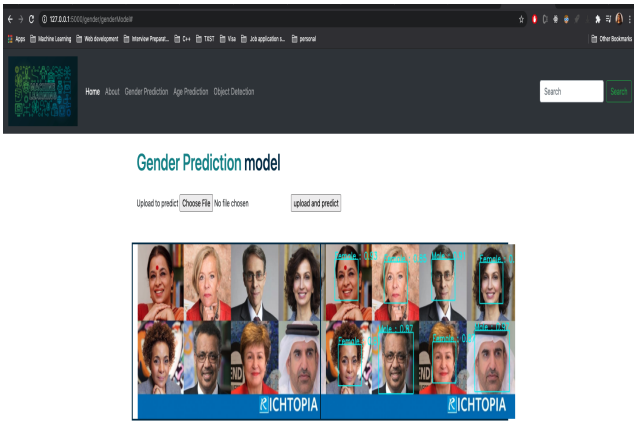

Figure 15. Website screenshot for the gender recognition module

machine learning model, and prediction analysis.

I attached my website screenshot in figure 15 with appropriate detections of the given group image. This model also works for the video and can able to detect the facial structure with a bounding box and appropriate confidence score with the appropriate class label.

My code work: https://github.com/SharmilaSrinivasa/Image-Processing-Application

7. Future Work

I intend to expand this project by creating various image recognition modules like Age Detection , Race Detection , and Object Detection by exploring various machine learning models. I also plan to improve the appearance of this

website and I think adding a detailed explanation for each module will help the users to learn how each module works. I hope to make this website public soon.

References

[1] M. M. Abdelwahab, S. A. Aly, and I. Yousry. Efficient web-based facial recognition system employing 2dhog. *arXiv preprint arXiv:1202.2449*, 2012. 1

[2] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European conference on computer vision*, pages 469–481. Springer, 2004. 1

[3] I. Kukenys and B. McCane. Support vector machines for human face detection. In *Proceedings of the New Zealand computer science research student conference*, pages 226–229. Citeseer, 2008. 1

[4] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Face recognition using lda-based algorithms. *IEEE Transactions on Neural networks*, 14(1):195–200, 2003. 1

[5] T. Mita, T. Kaneko, and O. Hori. Joint haar-like features for face detection. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1619–1626. IEEE, 2005. 1

[6] A. Suman. Automated face recognition. *Diambil dari http://www. npia. police. uk/en/docs/Face_Recognition_Report. pdf*, 2006. 1

[7] K. Talele, S. Kadam, and A. Tikare. Efficient face detection using adaboost. In *IJCA Proc on International Conference in Computational Intelligence*, volume 10, 2012. 1

[8] L. Wiskott, N. Krüger, N. Kuiger, and C. Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):775–779, 1997. 1

[9] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003. 1