Ashoka University

Monsoon 2021

CS-1390/PHY-1390: Introduction to Machine Learning

Expository Project on Real-Time Face Detection and Recognition Using Haar-Cascades and Local Binary Pattern Histograms

Rohit Vasishta, Manish Yadav, Nidup Dorji

14 December 2021

ABSTRACT

Facial Detection and Recognition software are universally present across political and social environments. Development in the reliability of such methods has begun to raise questions over its easy accessibility, indicating a need for thought into privacy-oriented aspects of facial recognition and detection. With every laptop in the world currently equipped with armory for reliable face detection and recognition, we wish to explore one of the most commonly used methods of detecting an object face- the Haar Cascades classifier. This project is an expository exploration of the working of the Haar Cascades classifiers proposed by Paul Viola and Micheal Jones in "Rapid Object Detection Using a Boosted Cascade of Simple Features" in 2001. The Recognition part of the project explores a popular algorithm for Facial Recognition- the Local Binary Patterns Histograms(LBPH). It is an extremely simple and efficient algorithm which involves labelling of the pixels in an image according to a threshold value set relative to the central pixel, returning a binary number as a result. The algorithm is based on Euclidean Distance Minimization between Histograms.

CONTENTS

1 INTRODUCTION

The project involves an end-to-end implementation of facial detection and recognition. The project is split into two parts- real-time face detection, and facial recognition. For face detection, we have used the popular Haar-Cascades classifier, proposed by Viola and Jones in the paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. The algorithm takes in a magnitude of positive images consisting of faces and a lot of negative images not consisting of any face to train on them.

For facial detection, we will be using Local Binary Patterns Histogram, an algorithm that identifies the owner of a face by minimizing the Euclidean Distance between histograms generated by the images. A more detailed explanation will follow in the upcoming sections of this paper.

2 LITERATURE SURVEY

Facial Recognition has been around for over 50 years. Woodrow W Bledsoe, along with Helen Chan and Charles Bisson of Panoramis Research researched facial recognition using computers in 1964(Bledsoe 1966a, 1966b; Bledsoe and Chan 1965). The method was to select from a small set of records, an image that matched the photograph in question the closest.

With time, there have been tremendous developments in the field. There exist algorithms like Eigenfaces, Fisherfaces, **S**cale-**I**nvariant **F**eature **T**ransform and **S**peeded **U**p **R**obust **F**eatures that have different methods of conducting the recognition.

Facial Detection is a specific implementation of object-detection algorithms. There are various methods- Knowledge-based methods, Feature-based methods, Template-Matching, Eigenface-based methods, Nueral Network-based, Support Vector Machine based, Naive-Bayes methods, and so on. In this particular project, we have chosen the face detection implementation of the Haar Cascades Object Classifier. https://www.overleaf.com/project/61b8376898d5a5748d92f5c7

3 DESCRIPTION AND GOALS

The project is conducted in three parts-Face Detection and Data Gathering, Recognition Training, and Face Recognition. The goals of the project include building a recognizer that is both accurate and precise, coupled with a detector that can detect images well in sparse lighting conditions, without the guarantee of a frontal face for detection.

4 DATASET SPECIFICATION

We used Haar-Cascade pre-trained model in OpenCV to extract images and also for face detection. The haar-cascade is trained on a huge number of positive and negative examples which means it is on both face and non face images. According to Viola and Jones paper, model was fed 4960 man 4960 manually labelled images and 9544 non-facial images. During the training, the images are scaled up to 24 * 24 because for a 24*24 windows of image, there are 162336 features. Training these features is very expensive so **adaboosting** is used to reduce the number of features. Ada-boosting takes few of the features out of 16233 and multiples them with the weights. The weights decide how important features is an facial features. These features are termed as weak features and these weak classifiers can be combined to form a strong classifiers.

$F(X) = w_0 f_0(x) + w_1 f_1(x) + w_{2}{}_2(x) + ... + m$
$F(X)$ is a strong classifier.

5 METHODS, EXPERIMENTS, EXPLANATIONS AND SCHEMATICS

Facial Detection: Haar Cascades

The Haar-Cascade is an object detection algorithm that can identify faces in real-time. It uses features known as the Haar features, proposed in the paper "Rapid Object Detection Using a Boosted Cascade of Simple Features" published in 2001.

Haar features make it easy to find edges and lines in an image, and also pinpoint the coordinates where there is a sudden change in the intensities in the pixels. Darker features take the values 1, while the lighter areas are pixels with values 0. The objective is to find the difference between the sum of all the features in the darker regions and the sum of said features in the lighter regions. This difference returns a value known as the Haar Value.

In essence,

Haar value= ($\sum$Dark Features) - ($\sum$Light features)

To detect an edge, the Haar Feature needs to traverse the entire image. This can prove to be computationally very expensive. For a single rectangle image, there are multiple pixel additions and subtractions that the feature will have to compute. Moreover, there are other Haar Features as well, such as the ones responsible for finding a line, or detecting a sudden change in pixel intensity. To battle this, Viola and Jones introduced the concept of the Integral Image. It is a table that holds the sum of all pixel values to the left and the top of a given pixel. The result of said summation will be the brightness value of the specific integral image.

In essence, the Integral Image of the (x,y) pixel of the image matrix will be given by:

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y')$$

The integral image drastically reduces the computational size for any feature, as compared to the previous method of image traversal. As in any image of a face, one would expect to find a certain set of features only in specific portions of the image. It becomes important hence, to narrow down on the features that actually contain the feature of the face, and discard the features that span the rest of the image.

In order to narrow down on the most important set of features for facial detection, the method of AdaBoost is used. Using this boosting technique, the final set of images is computed.In order to further narrow down on the facial features present within this finalized features, Viola and Jones proposed a technique known as Cascading, where the features are applied on the images in stages. Only the images that show facial features to be present continue to be applied, whilst the rest are discarded. This process continues, and with each phase, more features are discarded. The initial phases are ones in which all windows without any facial features will be removed. In the later stages, there is a greater focus on reducing the errors and focusing on more complex facial features.

Facial Recognition with LBPH:

We decided to utilise the algorithm of Local Binary Patterns Histogram as it is popularly known to be one of the oldest and the best algorithms in terms of its performance and how it is able to recognize the face of of an image, whether it is a frontal or a side-based image. The algorithm of LBPH starts by applying a binary classifier on pixels by comparing with the central pixel of a 3x3 matrix of pixels(classification of its 8-nearest neighbors). If a non-central pixel is greater than the central element of the matrix, it is classified as null, else, it attains the value 1. The algorithm then collects a binary value associated with this 3x3 block by concatenating the individual binary values. The decimal representation of this binary value is then obtained.

This algorithm works well in dynamic lighting conditions, as higher values are associated with brighter images, and lower values with darker images. The algorithm then creates several squares, within each of which lies the previous square. It then applies the previous condition to each of the squares, considering the central pixel/matrix as the reference point. We will have a new image which represents better the characteristics of the image at the end of the procedure. We can also experiment with different radii, and not just restrict ourselves tot he 8 nearest neighbors! Next, a histogram is created, which allows one to count the frequency of the appearance of each color in each square. The histogram will contain 256 positions representing the occurrences of each pixel intensity. Each histogram is then concatenated to create new and bigger histograms. The final histogram represent the characteristics of the original image.

The facial recognition is based on minimizing the distance between histograms. The algorithm has already been trained. Each histogram created represent an image from the training set. The image with the histogram closest to the histogram of the image of the subject will be the returned value, and the ID of this histogram will be the output.

In other words, we intend to minimize:

$$D = \sqrt{\sum_{i=1}^{n}(A-B)^2} \text{ where A,B is histograms}$$

A higher confidence represents lower accuracy. The lower the distance, the greater is the accuracy towards our recognition software. Based on a pre-set threshold, we can estimate if the algorithm has correctly recognized the image.

6 IMPLEMENTATION

We can break down the model implementation in the following steps;
1. Data Generation
2. Data Cleaning
3. Training
4. Face Detection

Data Generation and Data Cleaning:

To train our model for face recognition, we downloaded FR (face recognition) dataset from the course-drive folder. The dataset consists of 46 classes each with five features.

Each class is labelled with IDs from 1 to 46 and is indexed in the "name-mapping.csv" file. We moved these classes of images in a separate, "image-dataset," directory where we changed each classes IDs with their relevant names. "image-processing.py" helped in processing these images. We also added the "face-make-dataset.py" file to generate the dataset for a new user whose face we want our model to recognize. It takes a name as user input and creates a folder with the same name in the images directory. The user needs to click on the camera pop box and press c to capture his/her image. To quit the pop window, when there are enough images for detection, need to press "q".

Face Training:

Now, here comes the important part, training the model with classes stored in the "images" directory. "face-training.py" does this job by fetching image files and feeding them to the model. We define the detector for face detection using OpenCVs' inbuild cascade classifier as **cv2.CascadeClassifier(file)**

File here is the path of the required "XML" file which helps in detecting the specific object.

We have also defined recogniser - the crucial part of the model. The recogniser takes an image as the argument, converts it into binary format, which is then converted to a histogram which is then saved as a "yml" file. Each histogram saved in "yml" file represent each image from the training dataset. Every name in the dataset got assigned an id and will be stored as a pickle file.

Face Detection:

Once the face is detected in real-time using cascade detector, we perform the above steps again and create a histogram which represents the image. To find the closest image which matches the input image, we compare the input histogram and try to find the closest histogram from our dataset. LBPH algorithm uses Euclidean distance, and output the ID from the image with the closest histogram.

7 OBSERVATIONS

So the algorithm output is the ID from the image with the closest histogram. The algorithm returns the distance which can be used to calculate the confidence of the recognizer. Confidence is inversely proportional to accuracy, which means higher confidence has lower accuracy as the histograms of trained image and real-time image are far away. Contrasting, lower confidence means higher accuracy and histograms are close to each other

Haar-based classifier is limited in some cases because it doesn't detect face if we manipulate the face a bit (say, cover up the eyes, or tilt the head to a side).

8 CONCLUSIONS AND FUTURE DIRECTIONS

Facial detection and recognition can be easily implemented by systems using suitable algorithms and boosting techniques. With already-high ac curacies of artificial facial recognition systems, their applications, particularly at times when human facial recognition is compromised plays an extremely important role in application. Possible future directions for our project include the usage of different recognition algorithms, to compare them along the lines of accuracy, reliability and precision. It will be interesting to come up with our own set of classifiers for the same.

REFERENCES

1. Dinalankara, Lahiru. 2017/08/04, Face Detection Face Recognition Using Open Computer Vision Classifies. `https://www.researchgate.net/publication/318900718_Face_Detection_Face_Recognition_Using_Open_Computer_Vision_Classifies`

2. Viola, P., amp; Jones, M. (n.d.). Rapid object detection using a boosted cascade of Simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. https://doi.org/10.1109/cvpr.2001.990517