Face Grouping using Haar Cascade Classifier

1st Albino Alex Braganza
School of Computer Science
MIT World Peace University
Pune, India

2nd Amit Gour
School of Computer Science
MIT World Peace University
Pune, India

3rd Ravi Kumar
School of Computer Science
MIT World Peace University
Pune, India

4th Dr. C. H. Patil
School of Computer Science
MIT World Peace University
Pune, India

***Abstract*—Image processing has improved greatly with a vast domain of applications under its umbrella. To name few we have hand writing character recognition, face detection and recognition, detecting motorist if they have worn helmets or not and much more.**

The final goal of our research was to develop a software which would detect and recognize faces and group them into separate folder depending on face selection made by the user. The system would then use machine learning to understand which image must be grouped together under one folder, even if new images are clicked.

We broke down the problem into number of steps

1. **Detecting and cropping faces from an image.**
2. **Separating images that do not have faces.**
3. **To detect whether a given input face is present in the given image.**
4. **Detecting multiple faces if they are present in an image.**
5. **Separating all the images from a give set, when an input face is given.**
6. **Separating all the images from a given set, when multiple face selections are given as input**.

I. INTRODUCTION

Having tons of images and having a tough time organising everything. There are software's that organise images based on various factors like time, date and location of place. Well we are working on a system that organises images based faces selected by the user. The user is given an option to select a maximum of 10 faces from the already exist faces, the system will then try to run some algorithms and group faces together depending the selected faces.

II. PROPOSED SYSTEM

The current system is capable of identifying faces and crop the detected face and save it in a separate folder. If no face is detected it filters them out. Next, the system checks whether a given face is present in the input image. Later the same system will detect multiple faces if they are present. The systems final objective is to iterate through all the image in a given set of images and separate all the image depending on the faces selected by the user.

A. Architecture of the proposed system

1. Flow control

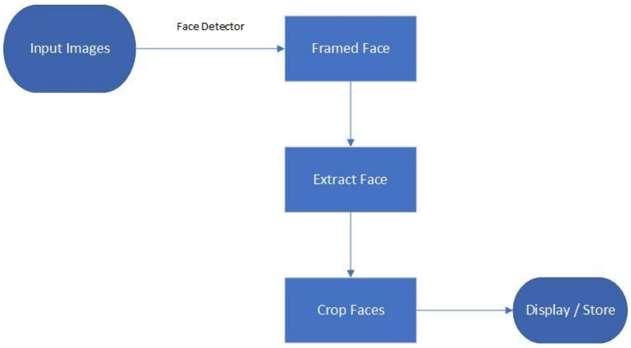


Figure 1 Face Filtering Process

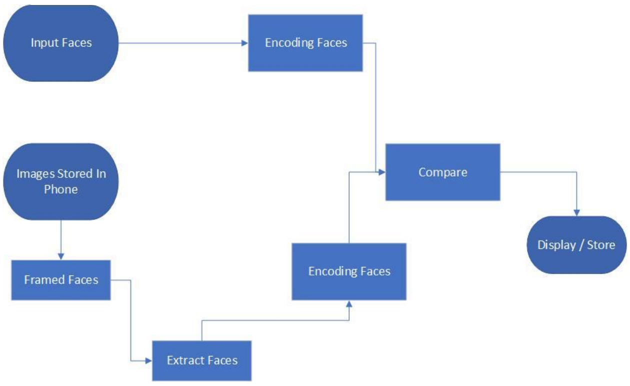


Figure 2- Face Cropping

III. WORKING AND ALGORITHMS

Face Detection is the technique to detect the faces in the various images. For face detection, we have used the Haar feature-based cascade classifiers.

It is a technique base on machine learning where cascade function is trained from a lot of positive images of faces and negative images without faces, and extract features from it. Haar cascade classifier is based on the Haar Wavelet technique to analyse pixels in the image into a square by function. The Haar cascade classifier uses the Adaboost algorithm for learning, which chooses a small number of important features from a large set to give an efficient result of classifiers.

An adjacent rectangular region is considered by Haar features at a specific location in the detection window, it then sums the pixel intensities in each region and estimates the difference between these sums.

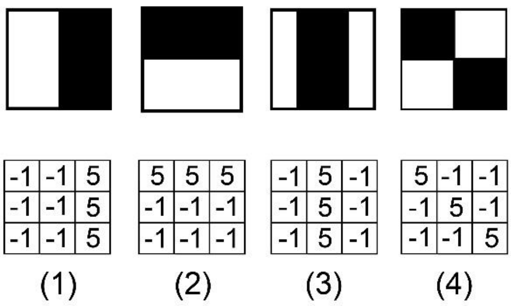


Figure 3 Haar Features

All the features which have calculated, most of them are irrelevant. The Ad boost method is used to remove the irrelevant feature. For detection of the face, a structuring element of the fixed size is moved on the input image, for each subsection of the image Haar feature is calculated.

Whenever an image is taken most of the image has very less area which is contains faces. So one way would be to check if a window does not have a face region. If it doesn't, we discard it, and don't process it again. Instead, we focus only on those regions where faces are present. Doing so, the system will take less time in processing only the face region.

To achieve this, we use the Cascade of Classifiers. Rather than applying all the features on a structuring element, we group the different stages of the classifiers and apply it one-by-one. In the first pass if the window doesn't detect a face, we discard the window and don't process it again during the next passes. We don't consider features that donton it. If in case it passes, apply the second stage of features and continue the process until the structuring element passes all stages is a face region.

IV. RECOGNITION

AI is revolutionizing the world. Face recognition is one such spectrum of it. Face recognition schema requires a face picture and predicts if the face matches with the other pictures. Face-recognition schema is used to compare and forecast possible face matches irrespective of speech, face, hair, and age.

The process of Face matching consists of three basic steps:

1. Face Detection process finds the position of the face in the image.
2. Face encoding process transforms a face into a set of mathematical representations based on the person's facial features.
3. Face Comparison process verifies if two faces are of the same person or not.

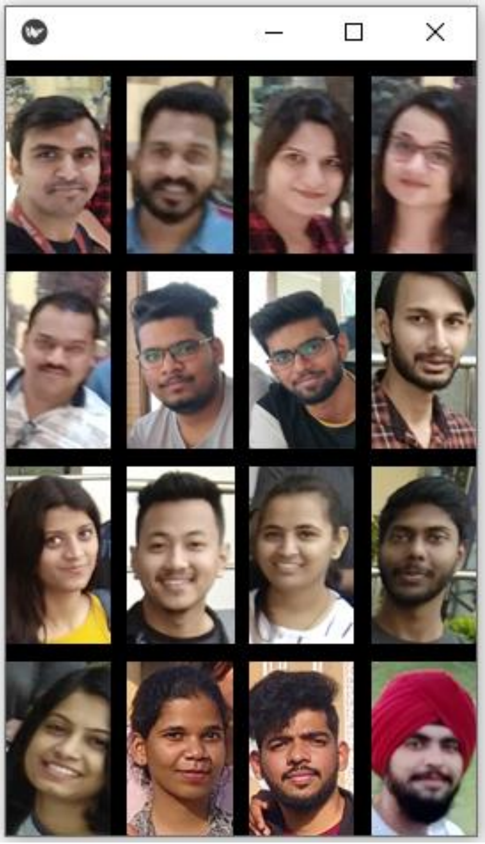
V. IMAGE ENCODING

After the detection of the face in an image, we have to encode the image into a mathematical representation. Mathematical representation is known as embeddings. This mathematical representation is a high dimensional vector. Each face has it embedding These embeddings are important to describe an image in a high dimensional space. In the end, we collect all the images and their corresponding embedding in a list.

VI. IMAGE COMPARISON

In image comparison, we compare the input face will all the faces present in the image. For comparing the faces, we compare the list of the input face embedding with the embedding of all the faces in an image. If the face matches with any of the images it returns the true value if not matched then it returns the false value.

VII. SCREENSHOTS



VIII. STATISTICS

Face Cropping				
Input		Output		
Images	**Actual Faces**	**Output**	**Output Faces**	**Faults**
1	4	3	3	0
1	5	5	5	0
1	3	4	3	1
1	8	7	7	0
1	3	7	3	4
1	6	6	3	3
1	4	4	4	0
1	11	11	11	6

Face Detection			
Input		Output	
Faces	**Total Images**	**Images which has particular face**	**Output Faces**
1	50	43	43
1	50	39	39
2	50	26	21
2	50	31	30
3	50	19	19
3	50	33	30
4	50	10	10
5	50	4	3

CONCLUSION

Our moto was to develop an application better than Google Photos, while Google Photos has some limitation like, it segregate images with only a single face. But we wanted to create an app which can segregate images according to the number of faces selected by user, so that all the images can be segregated in which those number of faces are selected. By using above described technology we were able to create an app like that, it can be very useful for group of friends or family photos segregation. To conclude, we can say that it was very challenging for us but our idea was unique so we tried and got successful.

REFERENCE

[1] Anna Liza A.Ramos, Dania May P.Aguila Anne, Catlyne B.Karununga, Jon-Jon B.Patino, Vincent L.Polintan 2019 “Face recognition with or without makeup using haar cascade classifier algorithm and local bibary pattern histogram algorithm” (IRJCS) Vol 6. Page 193-199.

[2] S. Bobde, et al. 2014. “Face Recognition Technology”. Internationa Journal of Computer Science and Information Technology (IJCSMC) Vol. 3 Issue 10.

[3] V. Morash, et al. 2009. “Face Recognition”. Journal of Vision. Vol. 9 Issue 8.

[4] https://docs.opencv.org/master/d9/df8/tutorial_root.html -OpenCV documentation