Mini Project Report On

Attendance System Using Facial Recognition

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by: Tanzeel Khan University Roll Number: 2017677

Under the Mentorship of

Mr. Ashwini Kumar

Assistant Professor

Department of Computer Science and Engineering

Graphic Era (Deemed to be University)

Dehradun, Uttarakhand

January 2023

Candidate's Declaration

I hereby certify that the work which is being presented in the project report entitled **"Attendance System Using Facial Recognition"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the Graphic Era (Deemed to be) University, Dehradun shall be carried out under the mentorship of **Dr. Ashwini Kumar, Assistant Professor,** Department of Computer Science and Engineering, Graphic Era (Deemed to be) University, Dehradun.

Name: Tanzeel Khan University Roll Number: 2017677

Table of Contents

Chapter 1

Introduction

1.1 Introduction

The need to recognize faces and the expressions portrayed by them has been gaining wind, recently. In the past decade, we have seen numerous corporations focusing on this problem. **Kairos, Nviso, Affectiva, and Imotions** have led some leading operations regarding the cause. Recent moves made by Facebook and Apple convey their interest in building a facial expression recognition system, which can efficiently and accurately predict human needs based on expressions.

In a *facial recognition* system, we process an image of a person by either taking it directly as an input or by continuously capturing pictures through a video feed and identify the person. Based on the confidence intervals, we identify the person shown in the image if they are available in our database. In a facial recognition system, we make use of pixels of an image after cropping out the redundant areas (background and rest of the body). The input comprises of an array of numbers which describe the gray-scale color value of each pixel used. A gray-scale pixel value ranges between 0 to 255 and it represents the brightness of a single pixel. Here, 0 is taken to be black while 255 is taken to be white.

Regardless of numerous developments in the field, there are some technical issues that are yet to be addressed. Distinction between expressions, changes in expression perception caused by lighting, different poses and angles of our head while facing a camera are some of them that need to be incorporated in the pipeline. These issues arise while working in a real-time environment where expressions are captured in an unconstrained way and their intensities vary from low to moderate.

In a world where everything is mobile and interactive, projects usually focus on recognizing people from pictures of video feed in an uncontrolled way. But, if we focus too much on these technical challenges, we might lose sight of a bigger question here: 'Can we reliably identify a person based on some pictures?'. There are several other fundamental questions just like this, that need to be concisely answered by scientists [1].

In addition to those fundamental questions, we also need to consider that different cultures bring a huge amount of diversity. In other words, the facial structure of people belonging to different cultures will add a lot of variances in the data that needs to be processed to make informed

identifications. So, we need to have a methodology that performs well in cross-dataset scenarios as opposed to the current prevalent methodologies which prefer intra-dataset performances. We aim to seek that if the patterns or characteristics learned from one group of individuals can be used to make informed predictions for another group of individuals. These groups may differ in age, gender, ethnicity, culture.

The area of facial recognition has been attracting tons of scientists, mathematicians, and researchers for more than 4 decades. The methodologies involved in extracting relevant information from an image vary from simple pixel extraction to identifying the facial landmarks and targeting specific areas on the face that prove to be the most informative. In the former, an array of pixel values from the image is prepared to be used as the input. The length of the array may differ in size depending upon the dimensions of the image being processed. On the other hand, in the method involving facial landmarks detection, we simply extract the positions of all 68 facial landmarks and based on the relative distances and shapes formed, accurate classifications can be made.

With regards to the classifier structures, the methodologies being proposed in literature usually are some modified or hybrid version of the above two methods, already explained. With the introduction of *Convolutional Neural Networks* (CNN), the tasks that involve making predictions or decisions by taking an image or video as input, have been made relatively simpler and much faster. Several models and algorithms based on CNNs have been developed which further simplify the process of decision making with high accuracy.

Chapter 2

Literature Survey

2.1 Related Work

A vast amount of work has been proposed in the field of facial recognition in the last two decades. There are numerous ways of approaching the solution. Some of these ways may include detection of Action Units, EMFACS frameworks, categorical models while other ways may attempt to recognize the basic facial features directly. Even though we have a vast number of ways to reach the solution, there exists an agreed upon consensus in theory on a pipeline for automated emotion recognition [2], [3]. This pipeline can be broken down into the following steps: face localization, face registration, feature extraction and identification. Face localization and registration have been perfected by scientists as they are the first steps towards any face related problem. Our focus is upon feature extraction and identification.

2.2 Handcrafted Features

These features are representations that are created manually based upon the domain knowledge. In other words, these are the features which an expert in a certain field might use to distinguish between different classes. For identification of humans, they are generally divided into two groups: shape and appearance features.

Shape features can be extracted by projecting the facial information onto a Cartesian plane in the form of point coordinates, distances, and angles. This can be done by keeping track of the important facial landmarks and using the distance between them for identification [4], [5], [6]. Even though they are perfected, the performance of shape features is highly sensitive to errors in the registration phase [2]. On the other hand, appearance features focus on using texture information by considering the intensity of the pixel values with respect to its neighbors [7], [8], [9].

2.3 Data Driven Features

These features are generated by algorithms in an automated manner which is then followed by an optimization process to best fit the task we want to pursue. One example of these data driven features is Convolutional Neural Networks in which convolution is the fundamental task that helps in learning correlation among neighboring pixels. Due to the exceptional results shown by CNNs in numerous Computer Vision applications [10] some researchers evaluated their performance in this task [11], [12].

Since the training of a CNN model from scratch is very tedious and cumbersome, we prefer to follow a transfer learning technique where we use the knowledge of a pre-trained model for feature extraction by fine-tuning it to our needs. The greater the similarity between the source task and the target task, the better the results [13]. Even though models based on CNN approach are the most widely used feature extraction across several Computer Vision problems, in theory we have other data-driven methods too. [9] provides a Mean Supervised Deep Boltzmann Machine (msDBM) as a feature extractor for the seven basic facial recognition problems. It works by maximizing inter-class and minimizing intra-class variations by taking the mean of features of a particular class. These learned features can be used to train a Random Forest classifier. [14] employs a Multi-scale Dense LBP for extracting descriptors in different resolutions. These descriptors are then joined together to form a single vector which is then used for feature learning in an unsupervised manner, then a Binarized Neural Network (BNN) is used to train on the learned features.

Even though several general-purpose metric-learning approaches [15], [16], [17] can be employed in the domain of emotion recognition, a few experiments were directly designed and evaluated for this task. [18] puts forward a two-branch CNN architecture that aims at jointly minimizing a cross-entropy loss function in the identification process. Other approaches [19], [20], combine metric learning and handcrafted features while optimizing a distance statistic along with features that are generally used in theory (example: LBP, HOG, Gabor Filters)

Chapter 3

Methodology

3.1 Procedure for Facial Recognition

As mentioned in this report before, I will be using Transfer Learning mechanism to perform **Facial Recognition**. To do this, I have used *face_recognition* library in Python. The functions that I used are described below:

1. face_encodings: Given an image, this function will return a 128-dimension unique encoding for each face in the image.
2. face_locations: This function returns a two-dimensional array of bounding boxes of human faces in an image using the CNN face detector.
3. compare_faces: This function compares a list of face encodings against a target encoding to find a match.
4. face_distance: Given a list of face encodings, this function will compare them to the known faces and give the Euclidean distance between them for each comparison.

Apart from face_recognition library, I have used the following other libraries to implement my project:

1. Open CV: This library is used to read and work on images stored locally.
2. Glob: This library is used for pattern matching in the directory filenames.
3. NumPy: This file is used for implementing arrays which are faster regarding calculations.

3.2 Complete Implementation

Here, I will be discussing the complete flow of my project. Since this project is made to be used by an Educator, all the explanation is made from their point of view.

A. **Sign-Up:** Here, you will be required to sign-up via an email. All the other functionalities will be available to you only after this process.
B. **Create the database:** After signing-up, you will get access to a GUI application, where you can log-in with your credentials and add your students.
C. **Taking Attendance:** After successfully adding all the students in the database, you can now easily take their attendance. Daily attendance will be stored locally on your device and a thorough record of the attendance will be maintained on a remote database.
D. **Monitoring Attendance:** You will be able to see the attendance percentage of every student. Students with an attendance ratio below 0.75 can be sent a warning email with an already provided attachment.
E. **Add Leave Days:** For any student, in case of emergency, you will be able to add a certain number of leave days to their request. For students on leave, the number of working days will not increase for the specified number of leave days.
F. **Broadcast Email:** With this functionality, you can send an email to every student in your concern with an attachment.

G. **Download Complete Data:** From this function, you will be able to download the complete attendance record of each student in your concern.

NOTE: This project supports multiple users and will work seamlessly for everyone.

I have used TKinter to make the GUI App and Flask to make the website. Both apps are synchronized, which means changes done on one app will be reflected on the other app.

Chapter 4

Results and Discussion

4.1 Results

The model that I created resulted in an accuracy of **96.4%**. Following are the different evaluation graphs that I made:

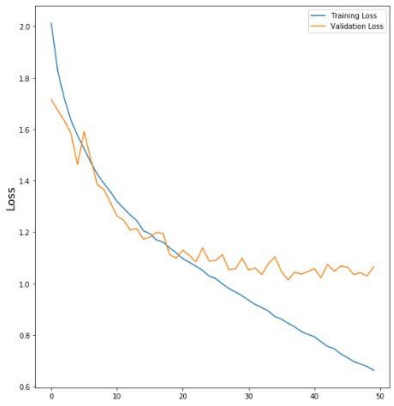

Figure 4.1 Validation and Training Loss

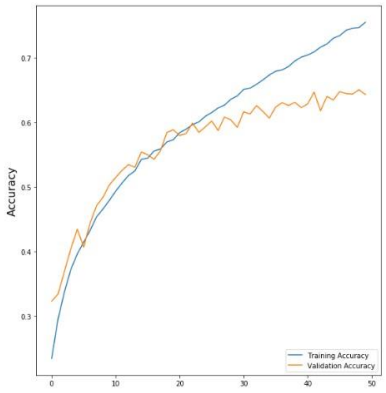

Figure 4.2 Validation and Training Accuracy

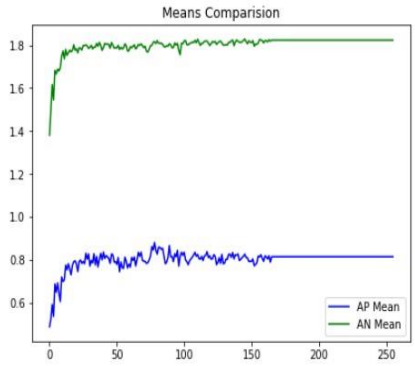

Figure 4.3 Means Comparison

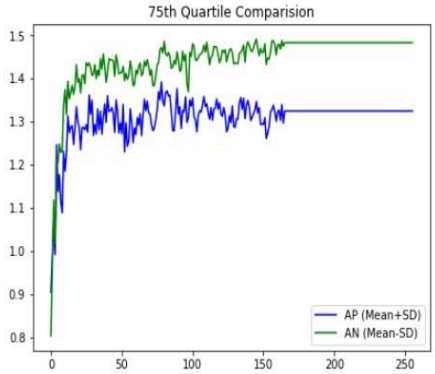

Figure 4.4 75th Quartile Comparison

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The project of implementing an **Attendance System Using Facial Recognition** was implemented using Python as the main programming language. At the end, if a user decides to use this project, they will be able to easily manage their attendance data and filter out the defaulters without any issues. This removes the extra burden of maintaining an attendance register for the students and filling in the entries daily.

5.2 Future Work

There are some features that I have left out because of the limitation of my programming capabilities and unavailability of time. To make this project better, one can add the feature to extract all the images stored in a folder and create the student database automatically. Furthermore, we can give limited access to the students where they will be able to monitor their daily attendance and raise an issue if it occurs.

References

[1] D. Heaven, “Why faces don’t always tell the truth about feelings,” Nature, volume 578, No. 7796, pp. 502–505, 2020.

[2] C. A. Corneanu, M. O. Sim´on, J. F. Cohn, and S. E. Guerrero, “Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications,” IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 8, pp. 1548–1568, 2016.

[3] E. Sariyanidi, H. Gunes, and A. Cavallaro, “Automatic analysis of facial affect: A survey of registration, representation, and recognition,” IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 6, pp. 1113–1133, 2014.

[4] Y. Zhang, L. Zhang, and M. A. Hossain, “Adaptive 3d facial action intensity estimation and emotion recognition,” Expert systems with applications, vol. 42, no. 3, pp. 1446–1464, 2015.

[5] H. A. Ahmed, T. A. Rashid, and A. T. Sadiq, “Face behavior recognition through support vector machines,” International Journal of Advanced Computer Science and Applications, vol. 7, no. 1, 2016.

[6] Z. Zheng, X. Li, J. Barnes, C.-H. Park, and M. Jeon, “Facial expression recognition for children: Can existing methods tuned for adults be adopted for children?” in International Conference on Human-Computer Interaction. Springer, 2019, pp. 201–211.

[7] F. A. M. da Silva and H. Pedrini, “Effects of cultural characteristics on building an emotion classifier through facial expression analysis,” Journal of Electronic Imaging, vol. 24, no. 2, p. 023015, 2015.

[8] A. Savran, B. Sankur, and M. T. Bilge, “Facial action unit detection: 3d versus 2d modality,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops. IEEE, 2010, pp. 71–78.

[9] S. Nagpal, M. Singh, M. Vatsa, R. Singh, and A. Noore, “Expression classification in children using mean supervised deep boltzmann machine,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.

[10] S. Li and W. Deng, “Deep facial expression recognition: A survey,” IEEE transactions on affective computing, 2020.

[11] M. Shin, M. Kim, and D.-S. Kwon, “Baseline cnn structure analysis for facial expression recognition,” in 2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE, 2016, pp. 724–729.

[12] H. Li, J. Sun, Z. Xu, and L. Chen, “Multimodal 2d+ 3d facial expression recognition with deep fusion convolutional neural network,” IEEE Transactions on Multimedia, vol. 19, no. 12, pp. 2816–2831, 2017.

[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” Advances in neural information processing systems, vol. 27, 2014.

[14] W. Sun, H. Zhao, and Z. Jin, “An efficient unconstrained facial expression recognition algorithm based on stack binarized auto-encoders and binarized neural networks,” Neurocomputing, vol. 267, pp. 385–395, 2017.

[15] S. Kim, D. Kim, M. Cho, and S. Kwak, “Proxy anchor loss for deep metric learning,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3238–3247.

[16] Y. Guo, Z. Cheng, J. Jing, Y. Lin, L. Nie, and M. Wang, “Enhancing factorization machines with generalized metric learning,” IEEE Transactions on Knowledge and Data Engineering, 2020.

[17] H. Hu, K. Wang, C. Lv, J. Wu, and Z. Yang, “Semi-supervised metric learning-based anchor graph hashing for large-scale image retrieval,” IEEE Transactions on Image Processing, vol. 28, no. 2, pp. 739–754, 2018.

[18] X. Liu, B. Vijaya Kumar, J. You, and P. Jia, “Adaptive deep metric learning for identity-aware facial expression recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 20–29.

[19] S. Wan and J. Aggarwal, “Spontaneous facial expression recognition: A robust metric learning approach,” Pattern Recognition, vol. 47, no. 5, pp. 1859–1868, 2014.

[20] H. Sadeghi and A.-A. Raie, “Histogram distance metric learning for facial expression recognition,” Journal of visual communication and image representation, vol. 62, pp. 152–165, 2019.