**CS-5710 Machine Learning Project Proposal**

MACHINE LEARNING TO IMPROVE THE EASE IN ATTENDANCE TAKING USING FACE RECOGNITION USING KNN

Submitted by T. JOHN EMMANUEL (700742724)  
Under the esteemed Guidance of Professor Muhammad Zubair Khan.

**Introduction:**

The recognition of the face is one of the most important and secure identifications of a human input formats for secure authentication. The idea of face recognition has been around for a long time, but the development of automated face recognition systems began in the 1960s and 1970s. One of the earliest pioneers in this field was Woodrow Bledsoe, who worked on facial recognition at the Panoramic Research Laboratory at Stanford University in the 1960s. Bledsoe's work involved manually marking key facial features on photographs and using a computer to measure the distances between these features for identification purposes. The initial ideas for face recognition were started around 1960s and 70s with the development of pattern recognition and visual abilities of computer architecture. The earliest full model was developed by Woodrow Bledsoe in the 1960s and it was called as “Candide” system. So, in this project I have decided to learn how Machine Learning can improve this recognition of human facial structures and Identify individuals accurately as it can be useful to multiple fields in the technologically advanced world.

**Objective:**

The main objective of this project is to develop a robust face recognition system that can accurately identify individuals in real-time using the stored face details. The algorithm uses the KNN to match the extracted features of the input face and match it with the one that has the most similar facial features in the database. The K-nearest neighbors (KNN) method is a face recognition technique that aims to create a system that can properly recognize and categorize people based on their facial attributes. Face recognition can be performed using KNN, a well-liked machine learning method for classification problems, by comparing the distances between feature vectors of known faces and an unknown face to identify the most similar faces. The important goal is to achieve high accuracy and in an efficient programming structure allowing for a domain to have a quick and reliable recognition of individuals.

**Motivation:**

Face recognition has numerous practical applications, including security systems, access control, surveillance, and personalized user experiences. The motivation behind this project is, by using this facial recognition one can gain easy access to their specified roles using their facial data rather than an input text which is not a reliable way to access a highly confidential data or role. The main motivation doing this using KNN is its simplicity, efficacy, and adaptability for classification tasks are the driving forces behind its use in face recognition. The main justifications would be KNN is a Straightforward and Natural Method, it’s Effective in High-Dimensional Spaces, Non-parametric Learning and its Resistance to Change.

**Evaluation:**

Metrics including accuracy, precision, recall, and F1 score can be used to assess the face recognition system's performance. The accuracy shows the proportion of tested faces that could be properly identified. Recall represents the percentage of correctly recognized faces out of all actual faces, whereas precision measures the proportion of correctly recognized faces out of all positive identifications. The F1 score offers a balanced evaluation metric by combining precision and recall into a single value.

**Significance:**

The K-nearest neighbors (KNN) technique for face recognition is very important in many disciplines. The importance of face recognition using KNN can include. Security and monitoring, Authentication via biometrics, A Customizable User Experience, Access Management Systems, Human-Computer Interaction, Forensic Examining**.** Systems for face recognition can automate access control procedures, and enable accurate identification of people, all of which can increase security. Face recognition can also improve individualized user experiences in programs like social media, online shopping, and entertainment. This project advances face recognition technology, opening the door for future systems that are more effective and precise.

**Execution:**

The project implementation involves several steps. At first, we import all the required libraries such as OpenCV, NumPy, Pickle and KN-Neighbor classifier from scikit library. The face data is then used to train a KNN classifier, where each face is represented by a feature vector and it accumulates the vectorized values for the comparison. During the recognition phase, the system captures video frames, detects faces using a Haar cascade classifier, extracts feature from the detected faces, and matches them against the features in the trained KNN model. The recognized individuals are displayed on the video capture box, along with bounding boxes around their faces and their names displayed. This identification can be used for several purposes and one of which is gathering large attendance in any domain in mere seconds which could save potential time and effort, thus leaving more time for important activities.

**References/Bibliography:**

1. Bledsoe, W. (1996). “Face Recognition and Identification”. Technical Report, Panoramic Research, Inc.

2. Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection." IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 711-720.

3. Face Recognition using OpenCV and Python: https://www.datacamp.com/community/tutorials/face-recognition-python-opencv

4. Hastie, T., Tibshirani, R., & Friedman, J. (2009). "The Elements of Statistical Learning: Data Mining, Inference, and Prediction." Springer.

5. Bishop, C. M. (2006). "Pattern Recognition and Machine Learning." Springer.

6. Haar Cascade Classifiers for Object Detection: <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html>

7. Pattern Recognition and Machine Learning by Christopher M. Bishop

8. K-Nearest Neighbors Algorithm Explained: <https://towardsdatascience.com/knn-algorithm-and-implementation-from-scratch-b9f9b739c02e>

9. Hands-On Machine Learning with Scikit-Learn and TensorFlow by Aurélien Géron

10. scikit-learn: Machine Learning in Python: https://scikit-learn.org/