Person Identification System Using Facial Recognition with Interpretability Emphasis

Jawad Hussain, Abdullah Siraj Khan, Abdul Hadi
Department of Computer Science, IBA University Karachi

***Abstract*—This report discusses our initial understanding and the setup phase of our project focused on developing a person identification system using facial recognition. The focus is not only on achieving high accuracy but also on ensuring model interpretability through computational thinking principles. A dataset shared by the TA was explored, and image preprocessing has been initiated using OpenCV and Matplotlib to prepare for model training.**

I. Introduction

Face recognition has become an essential element in modern identification systems, utilized across security, authentication, and surveillance domains. However, the lack of interpretability, often regarded as the "black box" nature of machine learning models, raises concerns regarding trust, bias, and accountability [1]. This project aims to develop a lightweight yet interpretable machine learning-based face identification system by applying computational thinking principles such as decomposition, abstraction, and pattern recognition.

II. Initial Problem Understanding

The dataset comprises facial image frames of 11 individuals, with approximately 400 frames per person. Upon inspection, the images show variations in eye contact, face orientation, presence of glasses, and lighting conditions. These variations simulate real-world challenges in feature extraction and consistent classification. The system aims to identify individuals by analyzing key facial features such as eyes, eyebrows, lips, birthmarks, jawline, and nose.

Future evaluations may involve more complex images, including blurred or low-light scenarios. The initial focus is on developing a robust preprocessing pipeline to standardize input data for both training and interpretability analysis.

III. Literature Review

Numerous facial recognition methods have been proposed. Traditional approaches, such as Eigenfaces and Fisherfaces, utilized Principal Component Analysis (PCA)-based dimensionality reduction [2]. More recently, Convolutional Neural Networks (CNNs) have achieved state-of-the-art performance in facial recognition [3]. However, deep learning models often lack transparency.

Interpretability techniques like saliency maps [4], Local Interpretable Model-Agnostic Explanations (LIME) [5], and Grad-CAM [6] provide visualization of model decisions. These methods align with computational thinking principles and will be considered for integration into this project.

IV. Dataset and Preprocessing Insights

The dataset consists of image frames grouped into folders, each representing one class. Images are resized to 256x256 pixels and converted to grayscale to reduce dimensionality and computational complexity. This step reflects decomposition by breaking down the problem into manageable components such as feature detection, input standardization, and classification.

Core libraries such as NumPy, Pandas, Matplotlib, and OpenCV have been installed and verified. Preliminary visualization indicates that pose normalization or pose-invariant features may enhance accuracy. Project repositories have been created on GitHub for collaborative development.

Fig. 1. Sample image frames from one subject in the dataset.

V. Work Distribution

- Jawad Hussain: Responsible for preprocessing, including dataset setup, resizing images, and grayscale conversion using OpenCV and Matplotlib.
- Abdul Hadi Javed: Leads feature extraction using HOG and CNN filters and will develop the GUI.
- Abdullah Siraj Khan: Manages classification, model training, accuracy testing, and the addition of interpretability tools.

VI. Conclusion

The team has initiated the development of an interpretable facial recognition system by exploring the dataset, establishing preprocessing workflows, and reviewing relevant literature. Next steps involve implementing feature extraction methods, experimenting with classifiers, and integrating interpretability tools. Computational thinking principles will guide future development and evaluation.

VII. Exploratory Data Analysis

A. Introduction

This Exploratory Data Analysis (EDA) explores and highlights our Face Recognition Dataset, comprising images of 12 individuals. The objective is to understand data's weaknesses, distribution, and insights from the data.

B. Dataset Summary

- Total Persons: 12
- Total Images: 4025
- Image Format: RGB

C. Class Distribution

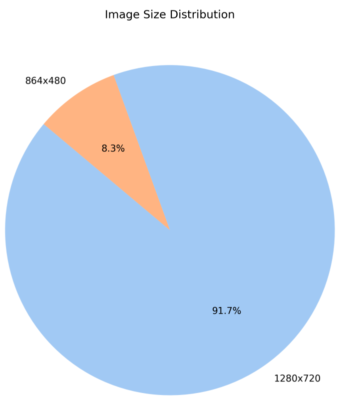

Fig. 2. Number of images per class.

Observation: The dataset shows a slight class imbalance. Class 7 has the highest number of images, while Classes 06 and 11 have fewer images. This imbalance can potentially affect our model's performance.

D. Sample Images

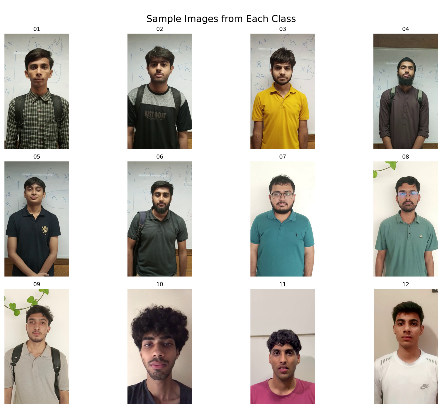

Fig. 3. Sample images from each class.

Observation: Visual inspection reveals variations in lighting, facial orientation, and background consistency across the dataset.

E. Average Pixel Intensity per Class

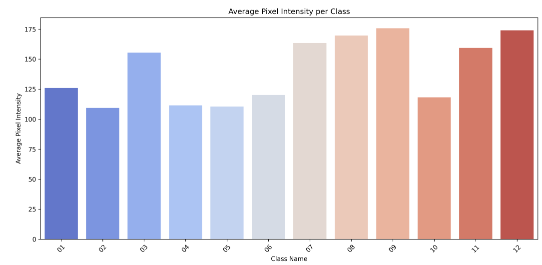

Fig. 4. Average pixel intensity per class.

Observation: Some classes exhibit slightly brighter or darker images on average, which could affect feature extraction consistency.

F. Pixel Intensity Distribution

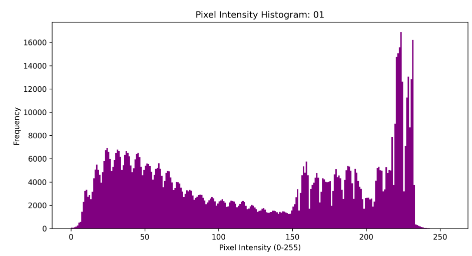

Fig. 5. Pixel intensity histogram for Class 01.

Observation: Peaks are visible in both the dark (0–100) and bright (180–250) pixel ranges, indicating high contrast images.

G. Image Size Distribution

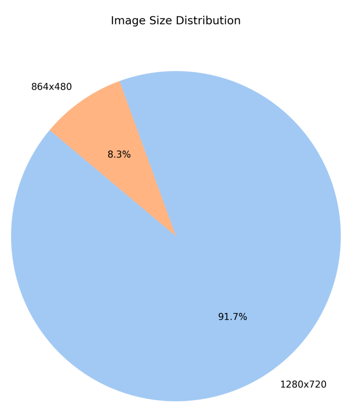

Fig. 6. Image size distribution.

Observation: Most images share the same size; only one class has different image dimensions.

H. Conclusion and Next Steps

- The dataset exhibits mild class imbalance.
- Face variability supports robustness for real-world applications.
- Brightness normalization could enhance model performance.

Recommendations include applying face alignment, using image augmentation, and considering class balancing techniques.

[conference]IEEEtran graphicx

VIII. Model Training

Model Training
In the initial phase of training of our model, we have used the Decision Tree Classifier algorithm to classify the extracted facial features for person identification. The Decision Tree was chosen because it is simple and interpretable. It is efficient in handling small datasets like ours. This model aligns with our aim of creating an interpretable face recognition system.

We have implemented the model using the Scikit-learn (sklearn) library in Python. The process involved splitting the dataset into training and testing sets. We have splited the dataset into 80:20 ratio. Then we extract features using the Histogram of Oriented Gradients (HOG) method. These extracted features served as the input to the Decision Tree model.

The model was successfully trained on the available dataset, achieving an initial accuracy of 95%.

Some challenges that we faced during this phase are:

- A limited dataset size, which affects the model's ability to generalize well to new images.
- Different shirts' and backgrounds' pattern which effect the model's performance
- Class imbalance, as some individuals have more images than others, our model sometimes predict the individual with more images more often
- Variations in lighting, facial angles, and expressions across images.

For future development, we plan to explore more advanced classification models such as Support Vector Machines (SVM), cropping pictures to only extract facial features and training model sufficient times.

IX. Feature Extraction

Feature Extraction:
We have used the Histogram of Oriented Gradients (HOG) technique for feature extraction. This algorithm is best for the projects which involves facial recognition and feature extracting. It processes the facial feature's shape and edges.

The key steps involved in extracting features using HOG are:

- Grayscale Conversion: All images are converted into grayscale image.
- Gradient Computation: The algorithm than computes the gradient by calculating the angle and the magnitudes of the pixels' cell.
- Abstraction: The image is divided into small regions (called cells), and for each cell, a histogram of gradient directions is computed.
- Feature Vector Generation: In last the histograms are then merged to make a final feature vector. Which are then used for the training of the model.

The decision to use HOG was based on its ability to:

- Capture the edges to capture the shape of the image
- Reduce the impact of changes in brightness and background
- Produce compact feature vectors suitable for efficient classification with interpretable models like Decision Trees.

We also have used the HOG visualization feature to enhance the interpretability of our model.
We have given more importance to some facial features such as eye shapes, dark hair and shape of eyebrow.

References

[1] D. Castelvecchi, "Can we open the black box of AI?" *Nature*, vol. 538, no. 7623, pp. 20–23, 2016.

[2] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[3] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in *Proc. CVPR*, 2015.

[4] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *arXiv preprint arXiv:1312.6034*, 2013.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier," in *Proc. KDD*, 2016.

[6] R. R. Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," in *Proc. ICCV*, 2017.