PRML-Project Mid-Report
Face Identification (7)

Sandeep Soni (B23CM1035) Japneet Singh (B23CS1022)
Atanu Kayal (B23EE1005) Varchasva Raj Saxena (B23CM1062)
Vishal (B23CM1048)

April 14, 2025

Abstract

This project focuses on face identification using feature extraction techniques such as HoG, LBP, and CNN, combined with traditional machine learning classifiers like Random Forest, Logistic Regression, and KNN to achieve robust recognition.

Keywords: Face Identification, Feature Extraction, HoG, LBP, CNN, Machine Learning

Contents

1 Introduction

The primary goal of this project is to develop an efficient face identification system that accurately classifies a given face image into one of K distinct classes. This problem holds substantial importance in various real-world domains such as security, surveillance, and human-computer interaction.

One key application of this system is an automated attendance system, which we are actively developing. In this system, individuals' attendance will be marked in real-time using face recognition, thereby eliminating manual entry, reducing errors, and enhancing overall security and efficiency.

Our approach involves capturing facial images at entry points, preprocessing them (face detection, cropping, and resizing), extracting features using methods such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HoG), and deep features from Convolutional Neural Networks (CNN), and then classifying them against a pre-registered facial database.

The system leverages traditional machine learning models like Support Vector Machines (SVM), Random Forests, etc., with extensive experimentation to compare accuracy and efficiency.

We initially began working with the Labeled Faces in the Wild (LFW) dataset, but its limited number of images per class made it unsuitable for training robust classification models. Therefore, we transitioned to a curated subset of the VGGFace2 dataset, which offers higher image count per individual and greater variability in pose and lighting. Specifically, we selected 50 classes with 300-350 images each, ensuring both scalability and computational feasibility.

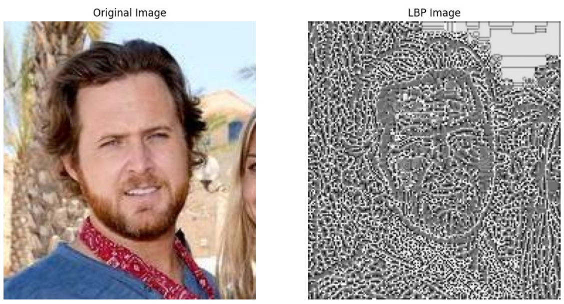
1.1 Citing Papers

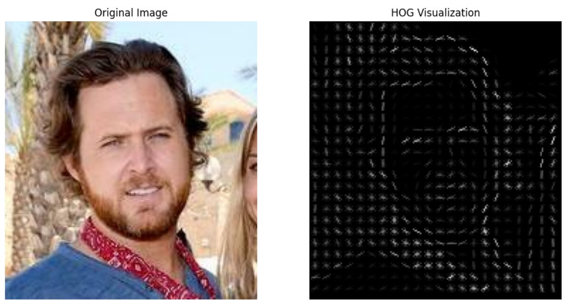
Ref Paper-1 https://www.ijml.org/vol10/987-IJMLC-24.pdf

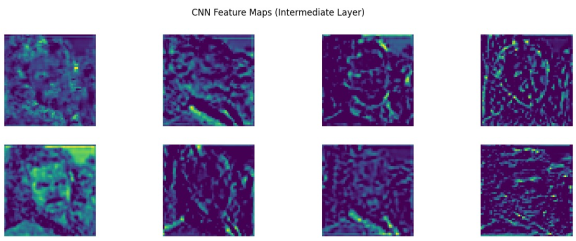
Ref Paper-2- https://ieeexplore.ieee.org/abstract/document/1619082

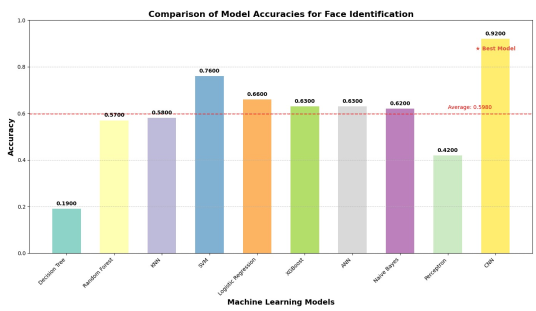
Ref Paper-3 https://link.springer.com/article/10.1186/s13640-018-0324-4

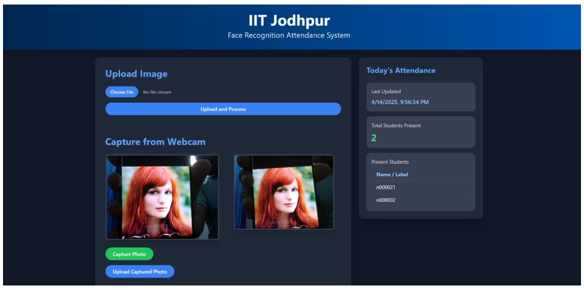
1.2 Figures

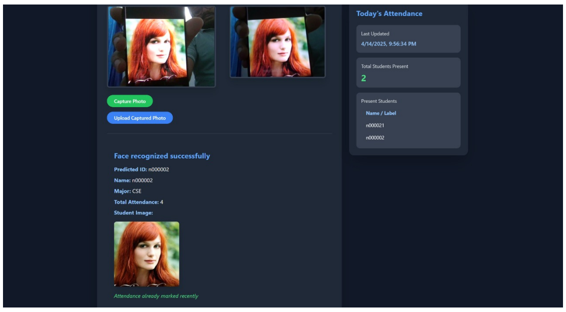


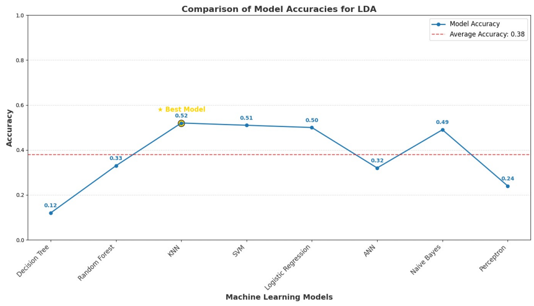


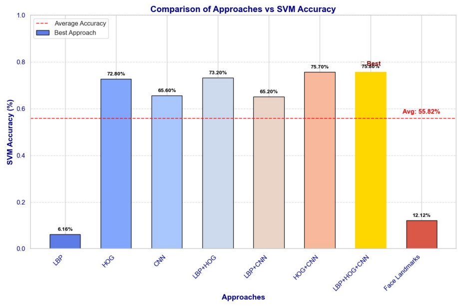












2 Approaches Tried

To build an effective face identification system, we explored multiple approaches involving both traditional feature extraction methods and modern deep learning techniques. The aim was to compare the performance of various representations and models in accurately classifying face images:

- **Using HoG Features:** Histogram of Oriented Gradients (HoG) was used to extract edge-oriented features from facial images.
- **Using LBP Features:** Local Binary Patterns (LBP) are texture-based descriptors that work well under varying lighting conditions. However, their inability to capture global facial structure limited their effectiveness in classification tasks.
- **Using CNN Features:** Deep features were extracted from pretrained Convolutional Neural Networks (such as ResNet18), which generated 512-dimensional embeddings.
- **Using HoG + CNN Features:** This combination merges the structural edge information captured by HoG with high-level abstract features extracted by a CNN. Together, they provide a rich representation of the facial image, balancing texture and semantic details.
- **Using LBP + CNN Features:** Here, the fine-grained texture features from LBP were combined with deep CNN features. While this fusion improved over using LBP alone, the accuracy was moderate, indicating that texture and deep semantic features.
- **Using HoG + LBP Features:** Combining two handcrafted feature extractors—HoG for edge orientation and LBP for texture—proved effective, yielding high accuracy. This pairing captures both local textures and facial contours well, even without deep learning components.
- **Combining LBP + HoG + CNN Features:** To leverage the strengths of all techniques, we concatenated feature vectors from LBP, HoG, and CNN.
- **PCA / LDA for Dimensionality Reduction:** After combining features (LBP, HoG, CNN), we used PCA/LDA to reduce dimensionality before feeding into ML models. This improved training speed and reduced overfitting.
- **Ensemble of ML Models:** Combining predictions from multiple ML models to improve robustness and accuracy through majority voting, stacking, and boosting algorithms.

- **Using CNN on Images Directly:** We trained convolutional neural networks directly on raw facial images, using several pre-trained models including ResNet18, ResNet50, VGG16, VGG19, AlexNet, etc. This end-to-end learning approach allowed the models to automatically extract task-specific features, improving classification accuracy. Among these, ResNet50 provided the best trade-off between performance and computational cost.
- **Using Face Landmarks:** Facial landmarks such as eyes, nose, and mouth were extracted using `dlib`. While fast and lightweight, landmark-based features lacked the depth needed for robust classification across diverse faces.

3 Experiments and Results

3.1 Preprocessing

Before feature extraction, the dataset undergoes several preprocessing steps to ensure uniformity and boost recognition accuracy. We used a curated subset of the VGGFace2 dataset, comprising 50 identities with 300–350 images per class. Each image undergoes the following pipeline:

- **Face Detection & Cropping:** OpenCV's Haar Cascade classifier is used to detect and crop the face region, effectively eliminating background noise and focusing solely on facial features.
- **Resizing:** All cropped face images are resized to a uniform dimension of 128×128 pixels, ensuring compatibility with various feature extraction algorithms.
- **Label Encoding:** To enable classification, the categorical identity labels (person IDs) were converted into numerical form using Label Encoding. This step ensures compatibility with machine learning models, which require numerical input for training and prediction.

3.2 Feature Extraction

The core of the system lies in converting facial images into meaningful numerical representations:

- **LBP (Local Binary Patterns):** Captures fine-grained local texture patterns by encoding pixel intensity differences into histograms. It is fast and robust under varying lighting conditions.
- **HoG (Histogram of Oriented Gradients):** Computes edge orientation histograms within image cells, effectively capturing the structural and contour information of faces.
- **CNN Features (ResNet):** High-level abstract features are extracted using a pre-trained ResNet model, generating a 512-dimensional feature vector for each image. These features are more robust to pose and illumination variations.

To improve feature richness, we concatenated LBP, HoG, and CNN features into a unified feature vector, which was then used for classification.Since the combined feature vector was high-dimensional, we applied Principal Component Analysis (PCA) to reduce the feature size to 100 dimensions. This dimensionality reduction not only improved training efficiency but also helped mitigate overfitting while retaining the most significant features for classification.

3.3 Classification Techniques

Various machine learning and deep learning models were employed to classify faces into one of the 50 identity classes using the extracted features. Traditional models like Support Vector Machine (SVM) performed best, effectively separating classes in high-dimensional space. Random Forest and XGBoost, though ensemble-based and powerful for tabular data, struggled with the complexity and dimensionality of facial features. Artificial Neural Networks (ANN) showed potential in learning non-linear relationships but required fine-tuning and more data.

Beyond these, we also trained Convolutional Neural Networks (CNNs) directly on raw images, using architectures like ResNet18, ResNet50, VGG16 . This end-to-end approach allowed models to learn features automatically and gave promising results.

3.4 Results and Evaluation

The results of our experiments highlight the impact of different feature extraction techniques, classical machine learning models, and deep learning approaches on face recognition performance. We evaluated individual feature sets—LBP, HoG, and CNN—as well as their combination, across multiple classifiers. Additionally, deep learning techniques such as directly training CNNs on raw images were explored to further enhance accuracy and robustness.To reduce feature dimensionality and improve model efficiency, we also compared Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) across various feature sets and classifiers. This helped identify the most suitable dimensionality reduction technique based on classification performance.The results obtained from various machine learning models are presented in the table below for clear comparison and analysis.

Features Used	**Accuracy (%)**
LBP	6
HoG	73
CNN	66
HoG+CNN	76
LBP+CNN	64
HoG+LBP	73
Face Landmarks	12

Accuracy of Different Features

Model	**Accuracy (%)**
Decision Tree	19
Random Forest	57
KNN	58
SVM	76
Logistic Regression	66
XGBoost	63
ANN	63
Naive Bayes	62
Perceptron	42

Accuracy using Concatenated Features (LBP + HoG + CNN and PCA)

Model Used	**Accuracy (%)**
ResNet18	93.8
ResNet50	92
VGG16	81.4

Accuracy of different CNN architectures (10 epochs)

Algorithm Used	**Accuracy (%)**
Majority Voting	72
Stacking	46
Boosting	30

Accuracy using ensemble methods

Model	**Accuracy (%)**
Decision Tree	12.62
Random Forest	33
KNN	52
SVM	51.3
Logistic Regression	50.3
ANN	32.4
Naive Bayes	49
Perceptron	24

Accuracy using Concatenated Features (LBP + HoG + CNN and LDA)

The performance of machine learning models in face recognition tasks using concatenated features (LBP, HoG, CNN, and PCA/LDA) can be attributed to their inherent strengths and limitations.SVM performed well because it is effective at handling structured, high-dimensional feature data and excelling at finding optimal decision boundaries in complex feature spaces. Random Forest, being an ensemble of decision trees, achieved moderate performance due to its robustness in handling feature variability. Simpler models like Decision Tree and Perceptron struggled because they lack the capacity to generalize well in the presence of high-dimensional and complex features. Ensemble methods like Majority Voting improved accuracy by combining the predictions of multiple models, effectively leveraging their complementary strengths.

4 Summary

4.1 Working Of System

This project presents a real-time face recognition-based attendance system designed to automate and streamline the attendance process. The system captures facial images using a webcam, detects and crops the face using OpenCV's Haar Cascades, and extracts features using a combination of LBP, HoG, and CNN (ResNet) methods. These features are then passed through trained machine learning models, with SVM achieving the highest accuracy , to classify the face into one of 50 predefined identities.Upon successful identification, the system uses Firebase to fetch the corresponding individual's records and mark them present in the attendance sheet. A time delay mechanism ensures that the same person is not marked multiple times in quick succession. To enhance usability, a fully deployed website was developed and hosted on Google Cloud, where students can seamlessly get their attendance marked through face recognition. Users simply access the site, allow camera permissions, and have their face detected automatically. Upon successful identification, attendance is marked in real-time, and all records are securely stored for future reference and analysis.

4.2 Future Work

Several enhancements are planned to improve the performance, scalability, and usability of the face recognition-based attendance system. First, we aim to expand the dataset to include more identities and diverse facial variations to improve generalization. We also plan to implement real-time model training and updating, allowing the system to dynamically add or delete individuals without requiring full retraining, enabling scalability in environments like schools or offices. To address the relatively lower accuracy of traditional machine learning models, we intend to replace them with fine-tuned deep CNN models, which are more capable of capturing complex facial features.

4.3 Use of Google Cloud

Google Cloud was integrated into the system to enhance scalability, reliability, and remote accessibility. Firebase, a cloud-based platform by Google, was used for real-time database management—allowing the system to fetch individual records instantly after successful face identification and update attendance logs securely. The cloud integration ensures that data is consistently backed up and accessible from multiple devices, making the system ideal for distributed environments like schools, offices, or organizations with multiple entry points. Furthermore, the project's web interface was hosted on Google Cloud and containerized using Docker for seamless deployment. The face recognition model was integrated directly into this web application, enabling real-time attendance marking via a user-friendly webpage accessible from anywhere. To demonstrate the system's functionality and showcase our work, the project page was deployed , allowing others to explore the application, see its features, and understand its implementation.

A Contribution of each member

1. Sandeep Soni (B23CM1035): Implemented ML techniques ,Prepared project page, report and frontend of the attendance system.

2. Japneet Singh (B23CS1022): Did Data Preprocessing , Implemented ML techniques(Using LDA and PCA) and CNN ,Backend of the attendance system , Prepared Project report.

3. Varchasva Raj Saxena (B23CM1062): Implemented ML techniques, Deployed the webpage using google cloud and docker, Linked the system to firebase database.

4. Atanu Kayal (B23EE1005): Implemented ML techniques , Prepared ReadMe , Frontend of the system.

5. Vishal (B23CM1048): Implemented ML techniques, Real time prediction , Prepared spotlight video.