Face Recognition: A Traditional Machine Learning Approach

Ghaith Chrit
Computer Science
University of British Columbia
Kelowna, Canada
ghaith.chrit@ubc.ca

Syed Aamir Ahmed
Computer Science
University of British Columbia
Kelowna, Canada
saamir@student.ubc.ca

Benjamin Norton
Computer Science
University of British Columbia
Kelowna, Canada
nortonb@student.ubc.ca

Justin Drenka
Computer Science
University of British Columbia
Kelowna, Canada
jdrenka@student.ubc.ca

***Abstract*— Facial recognition is an important component in computer vision, with extensive use in numerous industries. This paper presents the design and evaluation of an ensemble-based facial recognition system that combines multiple detection and recognition techniques to improve accuracy and resilience in uncontrolled conditions. The system uses a multi-stage preprocessing pipeline to detect and align facial features. A hybrid feature extraction method is then applied to capture diverse facial characteristics. Multiple classification models are used in parallel, and their combined outputs generate the final prediction. Experimental results demonstrate that the ensemble outperforms individual methods by better handling variations in lighting, pose, and occlusion. Our findings suggest that a generalized ensemble approach offers a more reliable solution for practical facial recognition applications compared to single-method pipelines.**

Keywords—Facial Recognition, Computer Vision, Ensemble Learning, Robust Recognition, Classification

I. Introduction

Computer vision, a subfield of artificial intelligence, empowers machines with the ability to interpret and analyze visual data [1]. This technology has transformed multiple industries, including security, healthcare, and autonomous systems. Within computer vision, face recognition is a task that involves identifying and verifying individuals based on their facial features. Face recognition plays a pivotal role in applications such as biometric authentication, surveillance, and personalized user experiences; however, achieving robust and accurate face recognition remains a significant challenge due to variations in lighting, pose, occlusion, and image quality [2]. These factors can dramatically alter the visual appearance of a face, making it tough for algorithms to extract features reliably. Changes in lighting can wash out or exaggerate face details. Additionally, variation in pose can cause misalignment of key features. Occlusions, from eyeglasses, hair, or masks can obscure important regions of the face. As a result of this inconsistency, facial recognition models must be capable of generalizing across a large range of unpredictable conditions to perform well in practical conditions.

Developing an effective face recognition system requires a strong understanding of both traditional and modern approaches. Earlier techniques such as gradient-based descriptors (e.g. SIFT [3]) and dimensionality reduction methods (e.g. Fisherfaces [4]) were computationally efficient but struggled individually in uncontrolled conditions. SIFT, while robust to scale and rotation, can produce inconsistent key points when facial features are obscured or when lighting changes suddenly. Similarly, Fisherfaces relies on linear projections that assume uniform lighting and front-facing faces, limiting its ability to generalize across unpredictable scenarios. More recently, deep learning techniques have brought improvements in accuracy and generalization but at a high cost of computation and increased complexity to implement. While these deep learning models dominate benchmark performance, they are often impractical for scenarios where data, hardware, or time is limited. In many real-world cases, especially when onboarding new individuals into a recognition system, only a few labeled images are available per person. This is a case where data intensive deep learning models may be less suitable.

To address these issues with facial recognition, we propose an ensemble-based system that strategically integrates multiple classical methods at various stages of the face recognition pipeline. This approach is designed to combine the strengths and efficiency of traditional techniques with the adaptability and robustness of a multi-model system. By utilizing the strengths of an ensemble approach, our face recognition system is effective across a diverse set of test conditions.

The remainder of the paper is organized as follows:
Section 2 provides background information and a comprehensive review of existing work in face detection and recognition, highlighting the strengths and limitations of prior approaches. Section 3 describes the design and implementation of our proposed system, including its architecture, inputs and outputs, and the algorithms used. Section 4 presents the results of our implementation, including test data and performance outputs. In Section 5, we discuss the overall performance of the system, compare it with existing methods, and analyze any limitations or unexpected results. Section 6 outlines possible improvements and future directions for the system. Finally, Section 7 provides all references cited throughout the paper.

II. Literary Review

Face recognition is a key area in computer vision and it consists of multiple stages. The general pipeline to recognize a face consists of: face detection, normalization, feature extraction, and classification. Each one of these steps contributes importantly to the performance of a recognition pipeline, especially when these systems are deployed in uncontrolled settings. Common challenges with face recognition include light variation, pose, facial expression, and occlusion, all of which often degrade the accuracy of traditional recognition systems. As face recognition systems are utilized more widely in real-world applications such as surveillance, access control, and personalization, robust handling of these challenges becomes essential [1], [2], [3].

Among traditional methods, the Viola-Jones algorithm [5] based on Haar cascades is widely used for real-time face detection. It leverages rectangular Haar features computed through an integral image representation and uses a cascade of weak classifiers. The method is very efficient and is well suited for controlled environments. However, it struggles with sensitivity to pose variation and occlusion. This often results in detection failing when faces deviate from frontal alignment or are obstructed. In contrast, the Histogram of Oriented Gradients (HOG) descriptor introduced by Dalal and Triggs [6] captures local edge structure by accumulating gradient orientations over spatial regions. When applied to face detection, HOG is more robust to variations in lighting and change of pose. Shu et al. [7] demonstrated HOG's effectiveness in face recognition pipelines. Although, it is more computationally expensive than Haar cascades and is still impacted by occlusion or extreme pose.

To mitigate pose variation, face normalization is often employed. Hassner et al. [8] introduced a method for 3D face frontalization using a generic reference model to synthesize frontal views of faces in unconstrained photos. This approach improves alignment without requiring 3D shape estimation on each image. While improving performance with pose variation, it relies on accurate facial landmarks and may create visual distortions when the face is not symmetrical. Still, normalization has shown to be a valuable step in boosting recognition performance, especially for traditional descriptors that assume frontal alignment of the face.

Early face recognition systems used global feature methods, treating the entire face as a single input; these systems include Eigenfaces and Fisherfaces. Turk and Pentland [9] demonstrated that Principal Component Analysis (PCA) could be used to reduce image dimensionality and represent facial images as projections in a low dimensional eigenspace. This approach is known as Eigenfaces and is efficient but sensitive to lighting and pose variations. Fisherfaces, introduced by Belhumeur et al. [4], combated these limitations by applying Linear Discriminant Analysis (LDA) to maximize class separability. As a result, discrimination between individuals improved despite changes in lighting and facial expression. However, both of these methods assume a linear relationship in data and perform relatively poorly in uncontrolled environments unless paired with preprocessing like normalization.

Feature-based methods extract local image descriptors which are more robust to small deformations and occlusions. Lowe's Scale-Invariant Feature Transform (SIFT) [3] identifies keypoints and computes descriptors that are not affected by scale and rotation. When used for facial recognition, SIFT suffers due to the smooth non-rigid nature of faces. Geng and Jiang [10] proposed modifications to the SIFT algorithm, such as Keypoint Preserving SIFT (KPSIFT) and Partial Descriptor SIFT (PDSIFT), to improve performance on facial data. These extensions to SIFT preserve useful boundary keypoints and increase the amount of features usable in face recognition but at the cost of increased computational complexity. The Bag of Visual Words (BoVW) model [11] extends SIFT by clustering local descriptors into a visual vocabulary, representing images as histograms of visual word occurrences. This approach captures local structure well and is relatively robust to minor geometric transformations, yet it ignores the spatial arrangement of features, which can negatively affect accuracy in a detailed task like face recognition.

Another traditional alternative is wavelet scattering, proposed by Mallat and extended by Oyallon et al. [12], [13], wavelet scattering introduces a mathematically grounded feature extraction framework that produces face representations which are stable to translations and slight shape changes. Unlike deep learning methods, wavelet scattering does not require training, which makes it very suitable for tasks with limited data. Its tolerance to noise and lighting variation makes it a strong candidate in face recognition, though its design limits how well it can adapt to highly specific datasets.

Given that no single method performs well across all conditions, ensemble learning has gained traction as a strategy to combine multiple models for improved performance. Polikar [14] presents a comprehensive overview of ensemble systems, emphasizing their ability to reduce error and variance by leveraging model diversity. In face recognition, Gupta et al. [15] proposed an ensemble approach combining multiple classifiers (LDA, SVM, KNN, etc.) using voting and bagging strategies. Their system, tested on the Olivetti Faces dataset [16], demonstrated high accuracy under controlled conditions. However, the Olivetti dataset is very limited in scale and variability, which restricts the generalizability of such approaches to real-world environments.

While ensemble learning has shown promise in combining classifiers, few systems explore this technique at both the feature extraction and classification stages. Additionally, many reported results are limited to small, idealized datasets.

III. System Design

Our proposed system employs a multi-stage pipeline for face recognition, starting with face detection using Viola-Jones (Haar cascades) and Histogram. Next, face normalization is performed using a 3D frontalization technique based on facial landmarks detected by an LBF model. For feature extraction, the system will ensemble Wavelet Scattering, Fisherface, and Bag of Visual Words with SVM and KNN classifiers using a vote aggregation strategy to further improve effectiveness in tough conditions. This design aims to improve generalization across pose, lighting, and occlusion.

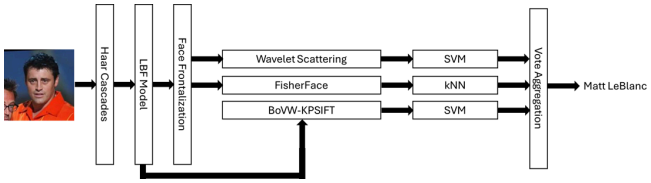

Figure 3.1: Proposed pipeline for face recognition

We will discuss the details of each step of the pipeline- face detection, face normalization, and face recognition - separately in the following subsections:

A. Face Detection:

For face detection, the system utilizes the Viola-Jones algorithm, which employs rectangular, two-valued Haar-like filters. These filters can be efficiently computed using an integral image representation, making this method widely adopted for its efficiency in real-time applications such as cameras.

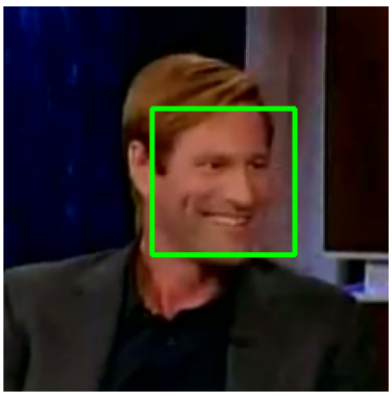

Figure 3.2: Haar Cascade true positive example

The algorithm uses a learning technique called AdaBoost to select a small, highly discriminative set of Haar features from a large pool. These selected features are then organized into a cascade of weak classifiers. Each stage in the cascade is designed to quickly reject the majority of non-face regions with minimal computation. Only image regions that successfully pass through all stages of the cascade are considered to contain a face. This cascading structure significantly contributes to the algorithm's speed and efficiency, making it highly effective in controlled environments where faces are typically frontal and well-lit. However, its reliance on specific Haar-like features makes it susceptible to performance degradation when encountering significant variations in pose, such as tilted or rotated faces, or when parts of the face are obscured by objects or other faces.

We also explored two other techniques for face detection: Naive PCA and Histogram of Oriented Gradients (HOG) for feature extraction, followed by an SVM for classification as face or non-face. HOG initially appeared to be a viable option, performing very well on the test split during initial testing. However, when applied to a webcam feed, it frequently yielded false positives. Efforts to improve performance by augmenting the training data with variable face resolutions, scenes, and other objects did not yield sufficient improvement. Consequently, we decided to proceed with the Viola-Jones approach, using a Haar Cascade classifier implemented by OpenCV, as it proved to be the most reliable.

Figure 3.3: Haar Cascade false negative example

We attempted to retrain the model using images from webcams and YouTube, as opposed to the pre-trained frontal face cascade file, to better align with the final testing distribution. However, this resulted in errors due to the training tools not being updated for the latest version of OpenCV. While the performance of the out-of-the-box classifier was not ideal for real-world use cases, it was sufficient for the specific requirements of our project.

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$
$$Cr \leftarrow (R - Y) \cdot 0.713 + \delta$$
$$Cb \leftarrow (B - Y) \cdot 0.564 + \delta$$

$$V \leftarrow max(R, G, B)$$
$$S \leftarrow \frac{V - min(R, G, B)}{V} \; if \; V \neq 0 \; else \; 0$$
$$H \leftarrow \begin{cases} 60 \frac{G-B}{V - min(R.G.B)} & if \; V = R \\ 120 + 60 \frac{B-R}{V - min(R.G.B)} & if \; V = G \\ 240 + 60 \frac{R-G}{V - min(R.G.B)} & else \end{cases}$$

Equation 3.1: YCbCr, following the JPEG conversion, and the HSV conversion equations. The value of δ depends on the image format

Additionally, we also tried to perform skin segmentation using YCbCr and HSV values. The process is summarized in Figure 3.4. First, the input image was converted to the YCbCr and HSV color spaces, as shown in Equations 3.1. Then, we applied thresholding to these values: Y (0, 255), Cb (133, 173), Cr (77, 127), H (0, 33), S (58, 255), and V (30, 255). These thresholds were chosen based on empirical observations and were influenced by prior work on skin detection, particularly in YCbCr space, as discussed in [20].

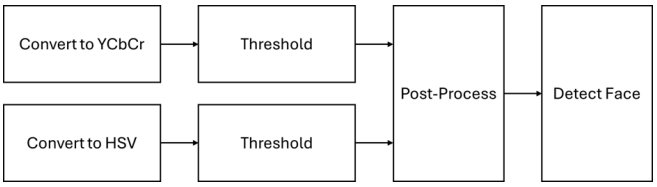

Figure 3.4: Process of the skin segmentation method

Following the initial segmentation, we performed post-processing to refine the detected skin regions. This included applying a minimum bounding box constraint to filter out small detections, where the bounding box was required to have an aspect ratio of approximately 1.5. Additionally, we experimented with cropping the detected region and applying Local Binary Features (LBF) and Haar-based classifiers to further refine the segmentation. Figure 3.7 illustrates the output of each individual channel on a sample image.

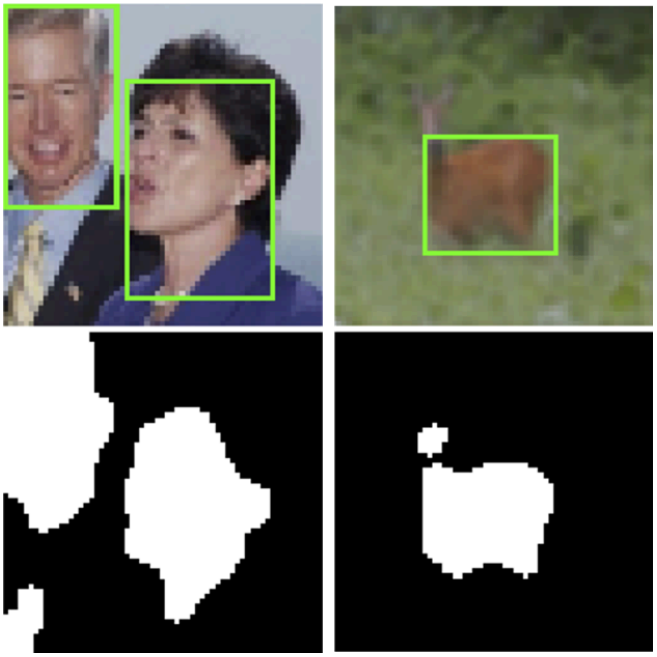

Figure 3.5: Example output of the skin segmentation. The left column shows a case where the skin segmentation was able to detect the face. The right column shows a false positive case. The bottom row shows the mask region where the skin is detected.

While this algorithm achieved a very high recall, it suffered from low precision due to a large number of false positives. Figure 3.6 presents an example of both a false positive and a true positive detection. The high false positive rate was primarily due to the broad threshold ranges, which caused non-skin regions with similar chrominance or hue values—such as certain fabrics, wooden surfaces, and warm lighting conditions—to be mistakenly classified as skin. Additionally, shadows and reflections often fell within the thresholded values, further increasing the number of false detections. Thus, this method was dropped from the pipeline for further comparisons.

B. Face Normalization:

Our pipeline uses a face normalization step which further includes face landmark detection, face frontalization, and face cropping. The overall aim is to systematically reduce variations in rotation and pose, thereby increasing consistency in the input images to improve model training.

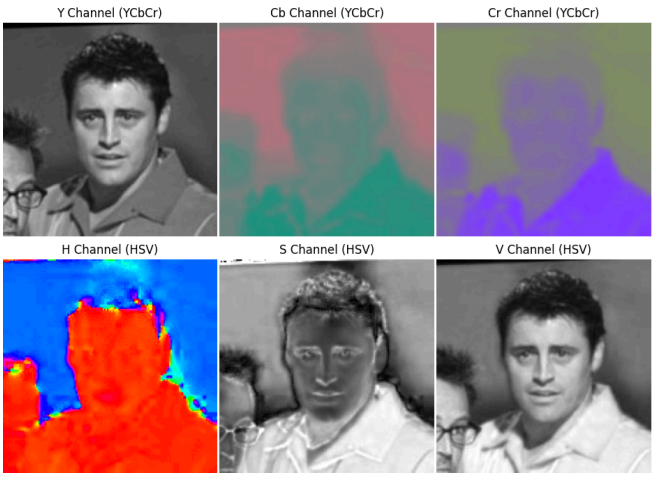

Figure 3.6: The YCbCr (top) amd HSV (bottom) channels

For face landmark detection, we use a pre-trained random forest that maps local binary features (LBF) to displacement vectors (offsets) [18], which iteratively adjust landmark positions on the face. This iterative approach helps to compensate for large pose variations in the early stages and then refines the alignment with finer adjustments. The output of this step is the face with landmarks close to their ground-truth locations.

Next, we align the face's pose using the Perspective-3-Point algorithm with a textured 3D model of a generic reference face, by estimating the camera's pose first then applying the necessary transformations to render the modelled face to the camera position. In order to fit the new alignment, the face usually needs to be resampled, so we back-project the appearance of the input face-photo onto this reference system, using the 3D model as a proxy to synthesize the frontalized view. This work followed the method introduced in [8]. Lastly, we also crop the face, using a convex hull, to ensure that the other parts of the image do not interfere with the classifier training. Figure 3.7 illustrates an example input to the face normalization step.

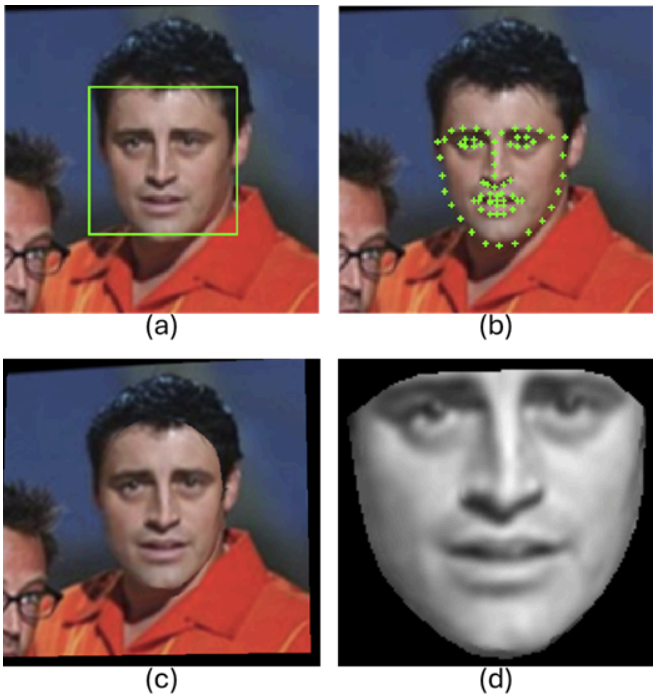

Figure 3.7: Output of every stage of the face normalization step. (a) represents the input of to the normalization step, (b) shows the output of the LBF step, (c) output of the face frontalization, and finally (d) represents the output after cropping the face

C. Face Recognition:

We use three separate methods of extracting features, with separate classifiers, as shown in Figure 3.1. These methods consist of FisherFaces, Wavelet Scattering, and BoVW + KPSIFT as mentioned above. Each feature extractor when combined complements each other to ensure optimal results. FisherFaces performs better at class separability and also has a better handle on within-class variability. Wavelet Scattering is robust to small transformations, noise, illumination changes while maintaining a fixed representation, limiting adaptability to highly specialised datasets. BoVW + KPSIFT is a flexible model that ensures no key-points are discarded which helps combat some of the weaknesses in Wavelet Scattering.

As shown in Section 4B, those methods were among the best when a single method was used. To obtain a final classification result, we aggregated the output of each classifier using majority voting. We also experimented with weighted voting based on their individual F1 scores. However, given the very close values, this approach did not make any noticeable difference. This is because we are using a one-vs-all approach where the output is binary, making simple majority voting just as effective.

IV. Results

This section discusses the results of our final experiments, including the intermediate results from our experiments on trying various feature extractors for face detection, as mentioned in Section III Subsection A. We also discuss the intermediate results from running the entire pipeline for the face recognition task using different face recognition feature extractors, that we used to compare the different approaches and decide on which extractors and corresponding classifiers to use in the final ensemble, along with whether or not to include face normalization in the final pipeline.

A. Face Detection

These are the results after training on ~3200 images from the Labelled Faces in the Wild (LFW) dataset [19] and the CIFAR-10 dataset for non-face images, with an even split between face and non-face images. These results are taken from testing on 400 face images and 400 non-face images.

As expected, the naive PCA classifier performed the worst on the dataset, but it achieved respectable classification performance when evaluating it on a live webcam feed.

Feature Extractor	**F1**	**Precision**	**Recall**	**Classifier**
PCA	0.84	0.83	0.85	RBF-SVM
HOG	0.99	0.99	0.99	RBF-SVM
Haar Cascade	0.95	0.98	0.92	AdaBoost

Table 4.1: The results of testing various face detection methods.

The HOG feature extractor achieved excellent results when testing on the YouTube faces dataset [17], however, the performance on live camera feeds proved to be suboptimal with the model often yielding excessive false positives despite performing well in tests. This was likely the model overfitting to the specific quality and dimensions of the feature vectors outputted by the HOG algorithm during training.

Lastly, for the Haar Cascade classifier, it should be noted that this model was not trained and/or fine tuned on the LFW data [19] due to the training issues mentioned earlier. However, it performed fairly well on our test set even despite not being trained on images from those distributions.

B. Face Recognition

This table includes the results from conducting the face recognition task using only 10 training images per target: 5 images of the target person and 5 images of random individuals. This procedure was repeated for each individual in the LFW dataset [19]. The results from these trials were then averaged to obtain a comprehensive evaluation for that trial. Experiments were first conducted with face normalization, and then rerun without it to assess the impact that face normalization can have on the classification.

The results for the pipeline with Haar Cascades for face detection, and incorporating face normalization are summarized in Table 4.2.

Feature Extractor	F1	Precision	Recall	Classifier
EigenFace	0.664	0.686	0.645	kNN
FisherFace	**0.712**	**0.745**	**0.682**	kNN
SIFT	0.196	0.141	0.322	kNN
KPSIFT	0.291	0.263	0.328	Linear SVM
BoVW-SIFT	0.698	0.728	0.670	Linear SVM
BoVW-KPSIFT	0.672	0.701	0.644	Linear SVM
Wavelet Scattering	<u>0.705</u>	<u>0.736</u>	<u>0.677</u>	Linear SVM
Wavelet Scattering	0.656	0.698	0.619	FC Layer

Table 4.2: The results of various face recognition methods when tested with face normalization. Aside from the Wavelet Scattering method, the classifier mentioned is the one that on average produced the best results (from an RBF-SVM, Linear-SVM, and kNN). Results are averaged over 10 trials. All methods were tested with the same random seeds (10, 99, 35, 8, 23, 81, 43, 77, 54, and 15). Bold indicates the highest performance among the methods, while underline indicates the second-highest performance.

These experiments revealed significant differences in performance across various face recognition methods, as summarized in Table 4.2. Classical linear methods like EigenFace and FisherFace serve as baselines, with FisherFace outperforming EigenFace (F1 scores of 0.712 vs. 0.664 with face normalization, FN). This improvement highlights FisherFace's ability to better capture discriminative features by maximizing between-class variance. In contrast, local feature-based methods such as SIFT and KPSIFT performed poorly, achieving F1 scores below 0.3, likely due to their limited ability to model subtle facial variations. Another possible explanation would be that there should be more hyperparameter tuning targeting those two methods. BoVW approach, when combined with SIFT or KPSIFT descriptors, showed much stronger results, particularly BoVW-SIFT (F1 score of 0.698 with FN), demonstrating the effectiveness of aggregating local features into a global representation. Wavelet Scattering also emerged as one of the top performers, achieving F1 scores of 0.705 and 0.656 with FN when paired with different classifiers, underscoring its robustness to variations in pose and illumination.

The results of the pipeline without incorporating face normalization into the pipeline are presented in Table 4.3. The impact of face normalization (FN) varied across methods, providing significant boosts for techniques like EigenFace, FisherFace, and Wavelet Scattering, but showing less influence on BoVW-KPSIFT, which performed comparably well without FN. This suggests that certain methods, particularly those retaining more keypoints or relying on aggregated features, may inherently tolerate misalignment better than others.

Additionally, the choice of classifier played a critical role, as seen in the performance gap for Wavelet Scattering when switching from a Linear SVM to a Fully Connected Layer. These results highlight the complementary strengths of different methods and the importance of aligning feature extraction techniques with appropriate classifiers.

Feature Extractor	F1	Precision	Recall	Classifier
EigenFace	0.637	0.673	0.604	kNN
FisherFace	0.666	0.711	0.627	kNN
SIFT	0.185	0.131	0.318	kNN
KPSIFT	0.260	0.217	0.325	Linear SVM
BoVW-SIFT	<u>0.708</u>	<u>0.733</u>	<u>0.676</u>	Linear SVM
BoVW-KPSIFT	**0.704**	**0.735**	**0.676**	Linear SVM
Wavelet Scattering	0.670	0.712	0.633	Linear SVM
Wavelet Scattering	0.640	0.687	0.601	FC Layer

Table 4.3: The results of various face recognition methods when tested without face normalization. Aside from the Wavelet Scattering method, the classifier mentioned is the one that on average produced the best results (from an RBF-SVM, Linear-SVM, and kNN). Results are averaged over 10 trials. All methods were tested with the same random seeds (10, 99, 35, 8, 23, 81, 43, 77, 54, and 15). Bold indicates the highest performance among the methods, while underline indicates the second-highest performance.

Overall, we observe that no single method dominates across all scenarios, which led us to try combining feature extraction techniques along with their best performing classifiers based on these results from our single-method testing. This is the final pipeline we decided to try for our experiments was the ensemble method that was described in Section 3. The results for the ensemble method is shown in Table 4.4

Feature Extractor	F1	Precision	Recall
Ensemble	0.734	0.762	0.708

Table 4.4: The results of the ensemble method. The result is average over 10 trails with the same random seeds as earlier

V. Discussions

The performance of the Histogram of Oriented Gradients (HOG) descriptor was found to be highly dependent on the resolution of the input images. Initial experiments trained a HOG-based model using a dataset comprising both faces and non-face objects. However, results indicated that the model struggled to reliably differentiate between the two categories when the resolution of the facial images was low. The gradient-based nature of HOG relies on the presence of well-defined edge structures, which are significantly diminished in low-resolution images. This led to a decrease in accuracy, as the model failed to capture the distinguishing features necessary for effective face detection.

Subsequent experiments involved refining the training dataset by incorporating more facial images of higher resolutions in an attempt to improve the model's ability to generalize. However, even with an increased number of face samples, the model's reliability remained suboptimal when applied to face cam inputs. Additional trials included training the HOG descriptor on datasets containing outdoor scenes with varying backgrounds and lighting conditions. The goal was to enhance robustness and reduce false positives, but these modifications did not yield significant improvements in performance. The core issue persisted: when facial images lacked sufficient resolution, the gradient-based feature extraction process was unable to effectively encode facial structures, leading to unreliable detection outcomes.

To mitigate these issues, a facial normalization algorithm was introduced as a preprocessing step before applying HOG. This step improved performance, as normalized faces provided more consistent and distinguishable gradient patterns. However, this approach presented a fundamental contradiction to the intended methodology. The primary goal was to use HOG as a preliminary detector to determine the presence of a face before applying facial normalization techniques. If normalization had to be applied beforehand to achieve satisfactory results, it effectively rendered the use of HOG as an initial detection step redundant. This outcome highlighted the inherent limitations of HOG for real-time face detection as inputs, suggesting that alternative feature descriptors may be better suited for such tasks.

Similarly, skin segmentation did not perform well due to the wide threshold range, which was designed to accommodate real-world variations in skin tones [20]. However, this broad range led to a high number of false positives, as many non-skin regions shared similar color characteristics, making the approach unreliable for robust face detection.

While the results shown in Section 4 might seem surprisingly low for face recognition techniques (i.e., face feature extraction), it is important to note that these results reflect the performance of the entire pipeline rather than just the feature extraction stage. Although, given that all other factors are controlled, the performance of different face feature extractors can be directly compared, as described in Section 4B, the earlier steps in the pipeline can introduce errors that negatively impact overall accuracy. These errors remain constant across the pipeline but can still degrade the effectiveness of the feature extraction stage. Specifically, inaccuracies in face normalization or cropping due to errors in face landmark detection can significantly affect recognition performance. Figure 5.1 illustrates several cases where the method fails to recognize a face, most likely due to mistakes in the face normalization step, emphasizing the cascading impact of early-stage errors on the final results.

Figure 5.1: The top row shows mistakes due to the LBF landmark detection algorithm while the bottom row shows mistakes during the face frontalization logic

While our primary focus in this study was on improving model performance, we did not specifically optimize for efficiency. As a result, while the ensemble method demonstrated the highest accuracy, it also introduced a trade-off in terms of computational complexity. The aggregation of multiple classifiers inherently increases inference time, making it less suitable for real-time applications.

VI. Future Work

Our ensemble system demonstrates strong performance using traditional computer vision methods. For future work, deep learning offers an exciting possible direction for further development of our pipeline. Deep neural networks, usually convolutional neural networks (CNNs), have reached state-of-the-art results in face recognition tasks by learning complex face patterns directly from massive datasets. They can automatically extract high-level features from images which allows for good performance under challenging conditions.

One direction would be to integrate deep learning into our pipeline for feature extraction. Pretrained deep learning models such as FaceNet, or ArcFace could be incorporated using transfer learning, enabling us to fine-tune facial embeddings on our dataset for improved performance. ArcFace, in particular, has shown superior performance by optimizing how distinct different faces are in the feature space using angular distance [21]. These embeddings could be fed into our existing ensemble of classifiers, potentially preserving the strengths of our current system while benefiting from deep feature representations.

Despite their strengths, deep learning methods come with trade-offs. They require large labeled datasets and powerful

hardware, making them difficult to deploy in low-resource or real-time settings. Additionally, retraining deep models to recognize new individuals with few sample images can be impractical. For these reasons, deep learning may not always be the best fit for our goals. While models like ArcFace offer strong feature representations, they introduce added complexity, resource demands, and retraining challenges.

Incorporating deep learning into our ensemble architecture represents a possible evolution of our system and may provide the adaptability needed for even higher performance. As deep models continue to improve in efficiency and accessibility, using them in a hybrid system like ours could be a fascinating avenue for future research.

Lastly, Siamese network architectures have been widely used for the task of face recognition, using convolutional networks for feature extraction and performing the final identification by finding which image in the database has features with the smallest distance to the target. Perhaps this approach can also be explored using the feature extractors discussed in this paper.

VII. Conclusion

In conclusion, our investigation into traditional face recognition techniques revealed the challenges of achieving robust performance across varying real-world conditions. While the Viola-Jones algorithm proved to be the most reliable choice for face detection among the explored methods, it had clear limitations in handling pose and occlusion. Furthermore, the performance of individual face recognition methods was significantly impacted by the accuracy of the preceding face normalization step, highlighting the compounding effect of errors within the pipeline.

Ultimately, our ensemble approach demonstrated the highest recognition accuracy by combining the strengths of different feature extractors and classifiers, albeit at the cost of increased computational complexity, indicating a clear trade-off between accuracy and efficiency that needs to be considered for practical use-cases.

VIII. Reference

[1] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2011.

[2] G. Huang et al., “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments,”
in Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, 2008.

[3] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711–720, 1997.

[5] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. I-511–I-518.

[6] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[7] Y. Shu, H. Li, and T. Wang, “Histogram of the Oriented Gradient for Face Recognition,” in Proceedings of the 2008 International Conference on Pattern Recognition (ICPR), 2008, pp. 789–794.

[8] T. Hassner, S. Harel, E. Paz, and R. Enbar, “Effective face frontalization in unconstrained images,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015.

[9] M. Turk and A. Pentland, “Eigenfaces for recognition,” Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, 1991, doi: 10.1162/jocn.1991.3.1.71.

[10] C. Geng and X. Jiang, “SIFT features for face recognition,” in Proceedings of the 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, 2009, pp. 598–602, doi: 10.1109/ICCSIT.2009.5234877.

[11] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” in Workshop on Statistical Learning in Computer Vision, ECCV, vol. 1, no. 1–22, 2004.

[12] S. Mallat, “Group invariant scattering,” Communications on Pure and Applied Mathematics, vol. 65, 2012.

[13] E. Oyallon, S. Mallat, and L. Sifre, “Generic deep networks with wavelet scattering,” arXiv preprint arXiv:1312.5940, 2013.

[14] R. Polikar, “Ensemble based systems in decision making,” IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, Third Quarter 2006, doi: 10.1109/MCAS.2006.1688199.

[15] A. Gupta, S. Sriram, and V. Nivethitha, “Harnessing Diversity in Face Recognition: A Voting and Bagging Ensemble Approach,” in Proceedings of the 2024 International Conference on Automation and Computation (AUTOCOM), Dehradun, India, 2024, pp. 249–255, doi: 10.1109/AUTOCOM60220.2024.10486188.

[16] AT&T Laboratories Cambridge, “The ORL Database of Faces,” [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[17] Beniaguev, David. “YouTube Faces with Facial Keypoints.” Kaggle, 6 Oct. 2021, www.kaggle.com/datasets/selfishgene/youtube-faces-with-facial-keypoints.

[18] L. Kurnianggoro and D. Passalacqua, “Kurnianggoro/GSOC2017,” Facemark API for OpenCV, https://github.com/kurnianggoro/GSOC2017 (accessed Feb. 20, 2025)

[19] Huang, G., et al, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," in Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, 2008

[20] Basilio, J., Torres, G., Pérez, G., Medina, L., & Meana, H., “Explicit image detection using YCbCr space color model as skin detection,” Proceedings of the 2011 American Conference on Applied Mathematics and the 5th WSEAS International Conference on Computer Engineering and Applications World Scientific and Engineering Academy and Society (WSEAS), pp. 123–128, 2011.

[21] J. Deng, J. Guo, N. Xue and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 4685-4694, doi: 10.1109/CVPR.2019.00482.