Exploring Feature Extraction Techniques and SVM for Facial Recognition with Image Generation Using Diffusion Models

Nabila Daly[1,2][a], Faten Khemakhem[1][b] and Hela Ltifi[1,3][c]

[1]*Research Groups in Intelligent Machines, National Engineering School of Sfax (ENIS), University of Sfax, BP 1173, 3038, Sfax, Tunisia*

[2]*Computer Science and Communications Department, Faculty of Sciences of Sfax, University of Sfax, BP 1171, 3000, Sfax, Tunisia*

[3]*Department of Computer Science, Faculty of Sciences and Techniques of Sidi Bouzid, University of Kairouan, Tunisia*

Keywords: Facial Recognition, Diffusion Models, Data Augmentation, Support Vector Machine, Feature Extraction, Histogram of Oriented Gradients, Eigenfaces, Local Binary Patterns.

Abstract: Facial recognition is a cornerstone of computer vision, with applications spanning security, personalization, and beyond. In this study, we enhance the widely used Labeled Faces in the Wild (LFW) dataset by generating additional images using a diffusion model, enriching its diversity and volume. These augmented datasets were then employed to train Support Vector Machine (SVM) classifiers using three distinct feature extraction methods: Histogram of Oriented Gradients (HOG), Eigenfaces, and Local Binary Patterns (LBP), in combination with SVM (HOG-SVM, Eigenfaces-SVM, and LBP-SVM). Our investigation evaluates the impact of these hybrid approaches on facial recognition accuracy and computational efficiency when applied to the expanded dataset. Experimental results reveal the strengths and limitations of each method, providing valuable insights into the role of feature extraction and data augmentation in improving facial recognition systems.

1 INTRODUCTION

Facial recognition is a key area in computer vision, with applications spanning across fields like security, surveillance, and personalized services. The ability to reliably identify individuals from images or videos is crucial for tasks such as access control, forensic analysis, and customizing user experiences. Although significant progress has been made in facial recognition technology, challenges like limited dataset diversity, and variations in pose, lighting, and facial expressions still hinder the creation of highly robust systems.

The quality and diversity of datasets play a crucial role in training effective facial recognition models. However, many widely used datasets, such as the Labeled Faces in the Wild (LFW), are often constrained in size and variability, limiting their utility for training models capable of generalizing to unseen scenarios. This limitation has spurred interest in leveraging generative models to augment datasets, enhancing both their size and diversity.

[a] https://orcid.org/0009-0001-8932-8904
[b] https://orcid.org/0000-0003-4386-4397
[c] https://orcid.org/0000-0003-3953-1135

Diffusion models have emerged as a state-of-the-art approach for data generation, known for their ability to produce high-quality, realistic synthetic images. By systematically introducing and then reversing noise in the data, these models excel in generating samples that closely resemble real-world data distributions. In this study, we apply diffusion models to augment the LFW dataset, generating a wide array of synthetic facial images. This enriched dataset, comprising both original and generated images, provides a robust foundation for training and evaluating facial recognition systems.

To assess the impact of dataset augmentation on facial recognition performance, we employ Support Vector Machines (SVMs) integrated with three feature extraction methods: Histogram of Oriented Gradients (HOG) (Rajaa et al., 2021), Eigenfaces (Safa Rajaa, 2021), and Local Binary Patterns (LBP) (Shubhangi Patil, 2022). These hybrid approaches—HOG-SVM, Eigenfaces-SVM, and LBP-SVM—offer diverse strategies for representing facial features, each with distinct strengths in capturing discriminative information from images.

Our experiments focus on training hybrid models

Daly, N., Khemakhem, F. and Ltifi, H.
Exploring Feature Extraction Techniques and SVM for Facial Recognition with Image Generation Using Diffusion Models.
DOI: 10.5220/0013439900003928
In *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2025)*, pages 240-251
ISBN: 978-989-758-742-9; ISSN: 2184-4895

using the augmented LFW dataset and assessing their performance in terms of accuracy, robustness to variations, and computational efficiency. By comparing the results systematically, we aim to gain insights into the effectiveness of combining diffusion-based data augmentation with hybrid SVM-based classification methods. Furthermore, we investigate how the generated data enhances model performance, particularly in overcoming challenges associated with the limited diversity of real-world data.

This study not only demonstrates the potential of diffusion models for dataset augmentation but also underscores the importance of integrating robust feature extraction methods with SVM classifiers to enhance facial recognition performance. The findings presented herein contribute to advancing the field by offering a practical approach to addressing data limitations and improving system robustness. (Smith, 1998).

2 RELATED WORK

Facial recognition has witnessed significant advancements in recent years, driven by the proliferation of deep learning techniques and large-scale datasets. Deep convolutional neural networks (CNNs) have emerged as state-of-the-art methods for facial feature extraction and recognition, achieving remarkable performance on benchmark datasets like LFW (Safa Rajaa, 2021) and Celeb-Faces Attributes (CelebA) (Rosebrock, 2021). In addition to deep learning approaches, traditional machine learning methods like SVMs remain relevant in facial recognition tasks. SVMs are particularly effective for binary classification tasks, including face/non-face discrimination, and can be adapted to work with various feature representations.

Among the traditional feature extraction methods, the (HOG) algorithm has shown promising results in capturing local texture and shape information from facial images. HOG-based systems combined with SVM classifiers have been successfully applied to real-time face detection and recognition tasks.

Eigenfaces, based on principal component analysis (PCA), represent another classical approach to facial recognition. By projecting facial images onto a lower-dimensional subspace of eigenfaces, this method reduces the complexity of face representation and enables efficient classification with SVMs.

LBP (Jae Jeong Hwang, 2018) provide a texture-based representation of facial images by encoding local texture patterns. LBP-based feature descriptors, coupled with SVM classifiers, have demonstrated robustness to variations in illumination and facial expressions, making them suitable for facial recognition under non-ideal conditions.

While deep learning methods have dominated recent progress in facial recognition, the comparative analysis of traditional feature extraction methods like HOG, Eigenfaces (Cheng Quanhua, 2008), and LBP combined with SVMs remains valuable. Under-standing the strengths and weaknesses of these approaches interms of accuracy, computational efficiency, and robustness is essential for developing practical and effective facial recognition systems. This study aims to contribute to this comparative analysis by evaluating these methods on the LFW dataset and providing insights into their performance characteristics. (Moore and Lopes, 1999).

3 PROPOSED APPROACH

In this study, we propose a two-phase approach for enhancing facial recognition performance. First, we augment the Labeled Faces in the Wild (LFW) dataset using a diffusion model to generate synthetic facial images. Second, we explore three distinct models for facial recognition using SVM classifiers in conjunction with different feature extraction techniques: HOG (Rajaa et al., 2021), Eigenfaces, and LBP (Shubhangi Patil, 2022).

3.1 Data Generation Using Diffusion Models

To address the limitations of the LFW dataset in terms of size and diversity, we employ a diffusion model for data augmentation. The diffusion process systematically adds noise to clean images and then reverses it to generate new samples that closely resemble real facial data. This approach enhances the variability of the dataset by introducing new samples with diverse facial attributes, poses, and lighting conditions, providing a richer training set for the subsequent recognition models.

The diffusion model architecture, specifically a Context-Unet (Hilbert et al., 2020), is used for generating synthetic images. The model learns to iteratively denoise images by passing them through multiple layers of convolution, down-sampling, and up-sampling blocks. Context and timestep embeddings are incorporated to condition the generated images on specific attributes, such as facial expressions or pose variations. The final output is a synthetic image that maintains the essential characteristics of real facial data, making the augmented dataset more diverse and robust.

The augmented dataset, consisting of both real and synthetic images, forms the foundation for training the proposed hybrid models. This phase is crucial for improving the robustness and generalization capabilities of facial recognition systems, especially in scenarios with limited real-world data.

3.2 Hybrid Models for Facial Recognition

Each model represents a unique approach to facial feature representation and classification, allowing for a comparative analysis of their performance on the LFW dataset.

1. **HOG-SVM Model:** The HOG method grabs details from facial images by looking at the direction of gradients. It divides the image into small parts and counts how many gradients point in different directions in each part. This helps capture both shape and texture of facial features. These counts are then fed into an SVM, which learns to tell apart different facial features by finding the best separation line in this high-dimensional space. SVM is great for this because it can handle lots of data and works well with the HOG features.

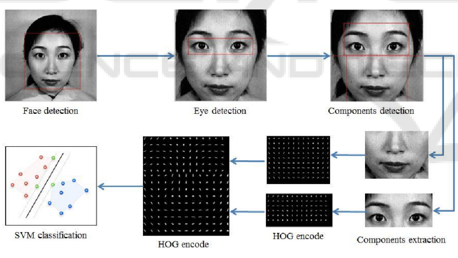

Figure 1: The proposed approach for HOG feature extraction and SVM approach.

2. **Eigenfaces-SVM Model:** Utilizes PCA to compute eigenfaces, representing discriminative features of facial images. SVM is trained on these eigenface representations for recognition.

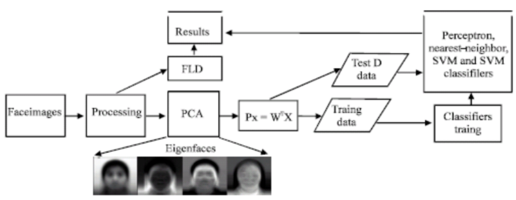

Figure 2: The proposed approach for Eigenfaces and SVM approach(Cheng Quanhua, 2008).

3. **LBP-SVM Model:** Incorporates LBP (Kancherla Deepika, 2019) to encode texture patterns, enabling effective handling of illumination and facial expression variations by SVM.

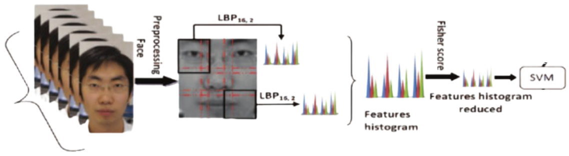

Figure 3: The proposed approach for LBP feature extraction and SVM approach.

Each diagram branch should dem-onstrate how raw facial images are processed through the respective feature extraction method (HOG, Eigenfaces, or LBP) to generate feature vectors, which are then fed into SVM classifiers for training and prediction. This visualization will provide a clear overview of the proposed approach and facilitate the understanding of feature extraction and classification stages in each model.

4 DATA GENERATION USING DIFFUSION MODELS

In this study, we propose the use of a diffusion model using U-Net model (Kassel, 2021) for augmenting the Labeled Faces in the Wild (LFW) dataset, addressing the limitations of dataset size and diversity. The diffusion model generates high-quality synthetic facial images by progressively adding and removing noise to real facial data. These synthetic images enhance the dataset by introducing variations such as different facial expressions, lighting, and poses. This approach improves the robustness of facial recognition models by providing a richer and more diverse training set.

4.1 Model Architecture

The figure 4 provides a detailed explanation of the ContextUNet architecture (Mittal, 2024), which consists of a series of layers designed to process and generate an output image based on the given input.

The input image is processed through the following steps:

1. Input Image

The initial input is an image represented by its dimensions: batch size b, channels c, height h, and width w.

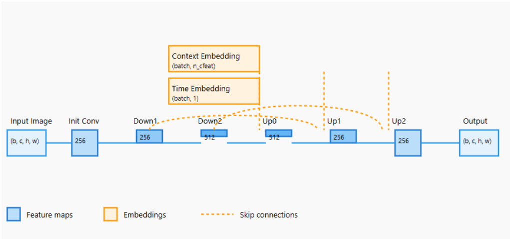

Figure 4: ContextUNet Architecture.

2. Initial Convolution (Init Conv)

The input image is processed by an initial convolutional layer, which is typically used to extract low-level features. This layer increases the number of feature maps to 256, allowing the network to capture more complex patterns and structures in the image.

3. Down-sampling Path

The feature maps are passed through a series of down-sampling layers (`Down1` and `Down2`). These layers consist of convolutional and pooling operations that reduce the spatial dimensions of the feature maps while increasing the number of channels (feature maps).

- `Down1`: Produces 256 feature maps.
- `Down2`: Produces 512 feature maps.

4. Up-sampling Path

After down-sampling, the feature maps are up-sampled through a series of up-sampling layers (`Up0`, `Up1`, and `Up2`). These layers involve transposed convolutions or other upsampling operations that gradually increase the spatial resolution of the feature maps back to the original input size. The number of feature maps is progressively reduced during this process.

5. Context Embedding

The context embedding layer processes external information, such as time steps or class labels, and generates a vector representation of the context. This embedding is integrated into the up-sampling path to condition the network's generation process based on the provided context.

6. Time Embedding

Similar to the context embedding, the time embedding layer takes the time step information and converts it into a vector representation. This allows the network to capture temporal dependencies and integrate this information into the up-sampling path, which is particularly useful for sequential tasks.

7. Skip Connections

Skip connections are used to connect the outputs of the down-sampling layers to the corresponding up-sampling layers. These connections help preserve fine-grained details by directly passing high-resolution features from the down-sampling path to the up-sampling path, ensuring that important information is not lost during the spatial resolution changes.

8. Output

The final output is generated by a series of convolutional layers, which produce an image with the same dimensions as the input. This image is the result of the network's processing, incorporating both the low-level features extracted by the initial convolution and the high-level contextual and temporal information from the embeddings.

4.2 Performance of the Diffusion Model

The adoption of the diffusion model significantly improves the diversity and quality of the synthetic images. The key performance improvements are outlined in the following table:

Table 1: Performance Comparison: Original vs. Augmented Dataset.

Metric	**Original LFW Dataset**	**Augmented Dataset (with Diffusion Model)**
Image Diversity	Low	High
Facial Variations	Limited	Extensive (e.g., pose, expression)
Lighting Conditions	Uniform	Varied (different light angles)
Image Quality	High	High (close to real images)

As shown in the table 1, the diffusion model introduces substantial improvements in image diversity, facial variations, and lighting conditions, providing a more robust training dataset for facial recognition models.

4.3 Hyperparameters

The training of the diffusion model relies heavily on the selection of appropriate hyperparameters. These parameters govern various aspects of the model's architecture, noise schedule, and optimization process. Below is a detailed description of the most critical hyperparameters used in the model.

4.3.1 Diffusion Model Hyperparameters

The diffusion process is central to the model's ability to generate high-quality samples. Key hyperparameters related to the diffusion process include:

- **Timesteps (500):** The number of diffusion steps the model uses to gradually introduce noise to the image. A higher number of timesteps allows for finer control over the noise addition, but it also increases computational complexity. In this model, we use 500 timesteps for a balance between computational efficiency and output quality.
- **Beta Parameters ($\beta_1 = 1e-4$, $\beta_2 = 0.02$):** These parameters control the noise schedule, which defines how noise is added to the data over time. β_1 represents the starting noise level, while β_2 determines the final noise level. The model uses a linear noise schedule that gradually increases the noise added to the input.

4.3.2 Network Architecture Hyperparameters

These parameters define the internal structure of the neural network used in the model:

- **Number of Hidden Features ($n\_feat = 64$):** This hyperparameter defines the number of hidden features or channels in the network. It plays a critical role in controlling the capacity of the model. A higher number of features can capture more intricate details but may lead to overfitting or slower training.
- **Context Vector Size ($n\_cfeat = 5$):** This refers to the size of the context vector, which encodes contextual information such as time steps or class labels. It helps the model condition the generation process based on this extra information. A context vector size of 5 provides sufficient capacity for encoding essential information without introducing unnecessary complexity.
- **Image Resolution ($height = 16$):** The model operates on 16x16 pixel images. Lower resolution speeds up training and reduces computational costs, but it may limit the fine-grained details that can be captured. In this case, 16x16 resolution is chosen to balance between computational efficiency and sufficient visual information.

4.3.3 Training Hyperparameters

Training hyperparameters are critical to the convergence and stability of the model during training:

- **Number of Epochs ($n\_epoch = 50$):** This parameter defines the number of complete passes through the dataset. A total of 50 epochs is used to ensure the model has sufficient opportunities to learn and improve its performance. The choice of 50 epochs allows for effective training without excessive overfitting.
- **Learning Rate ($lrate = 1e-3$):** The learning rate controls the step size during optimization. A learning rate of $1e-3$ is chosen to balance fast convergence with model stability. The learning rate is decayed linearly over epochs to prevent large updates in the later stages of training, ensuring that the model fine-tunes its weights effectively.

4.3.4 Optimization

- **Optimizer (Adam):** The Adam optimizer is used for model training. It is well-suited for models with large datasets and parameters, as it adapts the learning rate for each parameter based on the first and second moments of the gradients. Adam helps to achieve faster convergence and better generalization.

The following table provides a summary of the key hyperparameters used in this diffusion model:

Table 2: Summary of Hyperparameters for the Diffusion Model.

Hyperparameter	Value
Timesteps	500
β_1	1e-4
β_2	0.02
Number of Features (n_feat)	64
Context Vector Size (n_cfeat)	5
Image Resolution (height)	16x16
Number of Epochs (n_epoch)	50
Learning Rate ($lrate$)	1e-3
Optimizer	Adam

5 DATA PREPROCESSING

HOG Feature Extraction with SVM (HOG-SVM)

In the preprocessing step, we use the HOG technique to extract key facial features from the LFW dataset. HOG captures details about shapes and textures in each image by analyzing the directions of gradients, which helps highlight important facial structures. We calculate HOG descriptors for each image and then use these as inputs to train an SVM classifier that specializes in facial recognition.

Eigenfaces Feature Extraction with SVM (Eigenfaces-SVM)

Another approach we use is Eigenfaces, which relies on PCA to identify the most distinctive features in the dataset. PCA reduces the data's dimensions by focusing on the main features that differentiate faces. We transform the images into these reduced representations (eigenfaces) and then use them to train another SVM classifier for facial recognition.

LBP Feature Extraction with SVM (LBP-SVM)

Lastly, we use LBP, which is effective in capturing textures, making it useful for handling differences in lighting and expressions. LBP encodes patterns found in small regions of the face, providing features that are resilient to such variations. We extract LBP features from each image and use them as inputs for an SVM classifier focused on facial recognition.

Each of these pipelines is followed by training and evaluating the SVM model on performance metrics like accuracy, precision, recall, and F1 score. We then compare the results to see how well HOG, Eigenfaces, and LBP enhance the accuracy and reliability of facial recognition on the LFW dataset. This approach underscores how feature extraction improves facial recognition model performance.

6 SVM CLASSIFIER

The SVM classifier is a powerful tool for binary classification tasks, known for its ability to separate data into two distinct classes. In our work, we use SVM with a linear kernel to differentiate facial images from non-facial elements within our dataset.

Our classification process begins by training the SVM with HOG features extracted from the images, as these features capture important structural patterns unique to faces. By learning these patterns, the SVM can establish a clear decision boundary that maximizes the separation between facial and non-facial classes.

During training, the SVM iteratively adjusts this boundary to achieve the best possible accuracy in classification. This training enables the SVM to recognize and correctly classify regions containing faces versus those without.

After training, the SVM model is incorporated into our HOG-based classification pipeline. For each new image, we extract HOG features and input them into the SVM, which classifies each image based on the learned boundary, helping ensure consistent facial detection on new data.

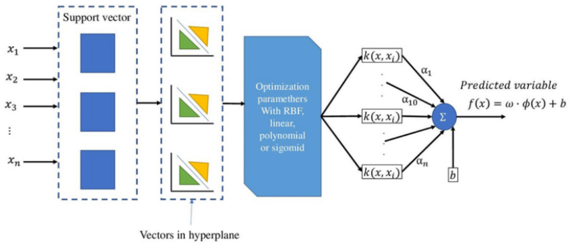

Figure 5: SVM Architecture.

7 METHODS

In this section, we describe the methods used for facial recognition, focusing on three approaches: HOG-SVM, Eigenfaces-SVM, and LBP-SVM. Each approach uses a distinct feature extraction technique combined with SVM for classification. We cover the training process, fine-tuning, and key hyperparameters chosen for each model.

7.1 Base Training

To begin, we trained three models using different feature extraction methods:

- **HOG-SVM:** The HOG descriptor was used to capture local gradient orientations from the facial images, emphasizing important shapes and textures. These HOG features were then input into an SVM classifier.
- **Eigenfaces-SVM:** PCA was used to generate eigenfaces, which capture the most important facial features in a lower-dimensional space. These eigenfaces were then fed into an SVM classifier.
- **LBP-SVM:** The LBP descriptor was applied to encode texture patterns from facial images, helping handle variations in lighting and expressions. These LBP features were then classified using an SVM.

The steps for each approach included:

1. **Data Preparation:** We loaded the LFW dataset and split it into training and testing sets to ensure balanced performance evaluation.
2. **Model Training:** Each SVM classifier was trained using GridSearchCV to optimize key hyperparameters such as the regularization parameter (C), kernel type, and gamma value for non-linear kernels.

3. **Evaluation:** We assessed each model's accuracy, precision, recall, and F1-score on the test set to compare their effectiveness in facial recognition.

7.2 Hyperparameter Optimization

After the base training, we optimized the hyperparameters of the models to improve their performance further. We employed the following procedure:

1. **Hyperparameter Optimization:** We performed a grid search over hyperparameters to find the best configuration for each model.
2. **Model Refinement:** The models were retrained using the best hyperparameters obtained from the grid search.
3. **Performance Evaluation:** We evaluated the models on the testing set using the same metrics as before.

7.3 Hyperparameters

For the SVM classifiers, we used the following hyperparameter grid during grid search:

- Regularization parameter (C): [0.1, 1, 10, 100]
- Kernel type: ['linear', 'rbf', 'poly']
- Gamma parameter (γ): ['scale', 'auto']

The best hyperparameters found during grid search were used to train the final models. The following table represents the best hyperparameters for different feature extraction methods (HOG, PCA, LBP) after performing hyperparameter optimization.

Table 3: Best Hyperparameters for Different Feature Extraction Methods.

Feature Extraction Method	Parameter	Values Tuned	Best Value
HOG	C	[0.1, 1, 10, 100]	10
	gamma	[scale, auto]	scale
	kernel	[linear, rbf, poly]	rbf
PCA	C	[0.1, 1, 10, 100]	10
	gamma	[scale, auto]	scale
	kernel	[linear, rbf, poly]	rbf
LBP	C	[1, 10]	10
	gamma	[scale]	scale
	kernel	[linear, rbf]	rbf

8 EXPERIMENTAL ANALYSIS AND COMPARISON

In this section, we present the results of two key experiments: the image generation using the diffusion model and the performance evaluation of various facial recognition models on the LFW dataset befor and after image generation.

8.1 Results

8.1.1 Results of Image Generation

In this part, we evaluate the performance of the image generation process using the diffusion model applied to the LFW dataset. The goal was to augment the original dataset by generating diverse images for each individual, simulating variations in lighting conditions, facial expressions, and poses. For each person in the LFW dataset, we generated multiple images with different facial expressions (such as happy, sad, and neutral), different pose orientations, and varying lighting conditions (e.g., different light angles). This augmentation aimed to increase the dataset's diversity, improving the robustness of facial recognition models trained on this dataset.

To assess the quality of the generated images, we compared them to the original LFW dataset in terms of visual fidelity and diversity. The evaluation was performed by inspecting the generated images for realistic facial features, maintaining identity consistency across generated samples, and preserving crucial facial characteristics such as eye shape, nose position, and mouth expression, despite the variations in lighting, pose, and expression.

Furthermore, we evaluated the impact of training the diffusion model for different numbers of epochs. The number of epochs played a significant role in the quality of the generated images. Initially, with fewer epochs, the generated images exhibited lower quality, with some distortion or unnatural features. However, as the number of epochs increased, the images gradually improved, showing more realistic and coherent facial features. The model achieved optimal performance after a certain number of epochs, where the generated images closely resembled real LFW images while maintaining sufficient diversity.

The results demonstrated that with an increased number of epochs, the diffusion model significantly enhanced the diversity and quality of the augmented dataset. The generated images displayed diverse lighting conditions, facial expressions, and poses, which were not present in the original LFW images, thus improving the generalization capability of facial recognition models.

Figure 6: Generated Images Sample.

The following observations were made from figure 6:

- **Image Diversity:** The augmented dataset exhibited high diversity, with the generated images capturing a broader range of poses, expressions, and lighting conditions compared to the original dataset.
- **Facial Variations:** The generated images demonstrated extensive variations in facial expressions (e.g., happy, sad, neutral) and pose orientations, making the model more robust for facial recognition tasks.
- **Lighting Conditions:** The augmented dataset showcased varied lighting conditions, simulating different light angles, which was not present in the original LFW images.
- **Image Quality:** The quality of the generated images was high, closely resembling real images and retaining critical facial features, enhancing their usability for further analysis and recognition tasks.

These results show that the diffusion model effectively augments the LFW dataset, providing enhanced diversity and realism in the generated images.

8.1.2 Results of Face Recognition Models

This section presents the comparative analysis of three facial recognition models: Hybrid HOGSVM, EigenfacesSVM, and LBPSVM. These models were evaluated on the original LFW dataset as well as the augmented dataset generated using the diffusion model. The primary metric used for comparison is accuracy.

Table 4: Performance Comparison of Facial Recognition Models on LFW Dataset.

Model	Accuracy (%) on LFW
Hybrid HOG-SVM	74.53%
Hybrid Eigenfaces-SVM	77.33%
Hybrid LBP-SVM	59%

The accuracy results highlight the performance differences among the feature extraction methods when integrated with SVM for facial recognition tasks. The Eigenfaces-SVM model achieved the highest accuracy among the three models, emphasizing the effectiveness of eigenface representations in capturing facial variations. The HOG-SVM model also demonstrated competitive performance, while the LBP-SVM model showed lower accuracy, indicating potential challenges in handling illumination and texture variations in the dataset.

8.2 Evaluation

Confusion Matrix

After training the facial recognition models using different feature extraction techniques combined with SVM classifiers, we proceeded to evaluate their performance on the LFW dataset. The evaluation includes assessing accuracy, precision, recall, and F1 score, which provide a comprehensive measure of the models' ability to correctly identify individuals while minimizing both false positives and false negatives. Additionally, we analyzed confusion matrices to gain deeper insights into the models' effectiveness, particularly in identifying misclassifications between similar facial features, expressions, or lighting conditions. These metrics and analyses were crucial in understanding the strengths and limitations of each model, helping to identify the most reliable approach for accurate facial recognition under real-world scenarios.

This is the confusion matrix of HOG and SVM method:
This is the confusion matrix of Eigenfaces and SVM method:

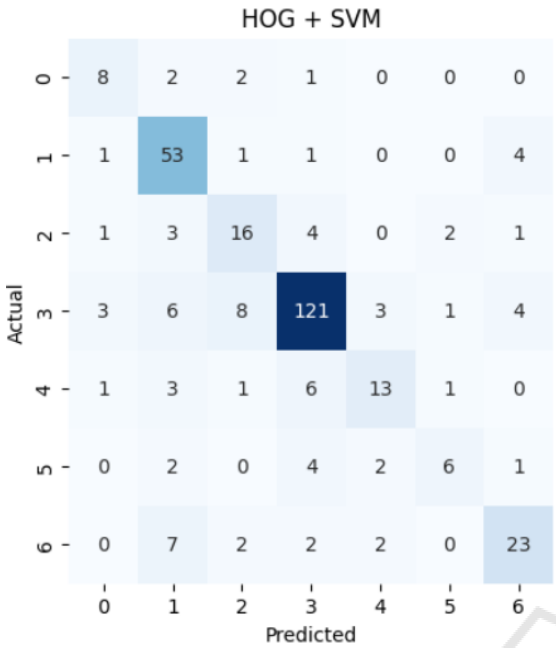

Figure 7: Confusion Matrix of Hog-SVM Model.

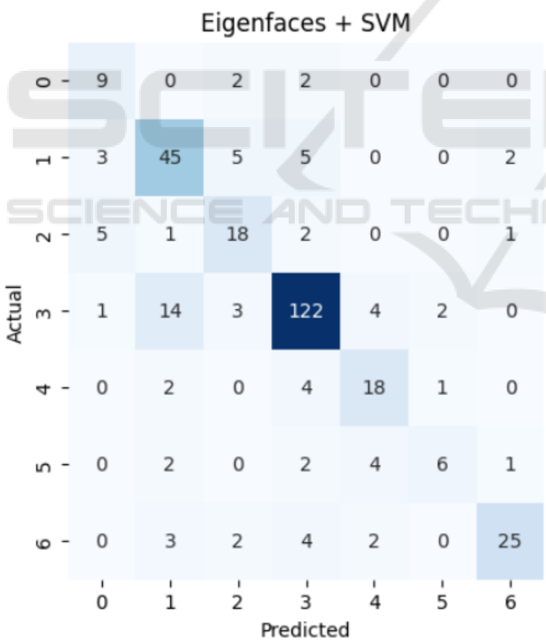

Figure 8: Confusion Matrix of Eigenface-SVM Model.

It signifies the proportion of correctly identified positive instances among all the actual positive instances. Overall, all models performed reasonably well. The HOG-SVM and Eigenfaces-SVM models achieved higher accuracy, precision, recall, and F1 score compared to the LBP-SVM model. However, further analysis and fine-tuning may be required to improve the performance of the LBP-SVM model.

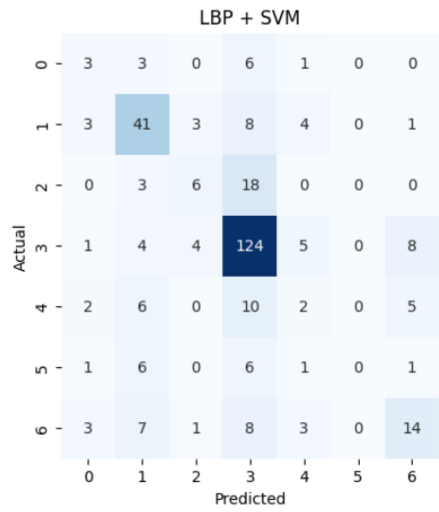

Figure 9: Confusion Matrix of LBP-SVM Model.

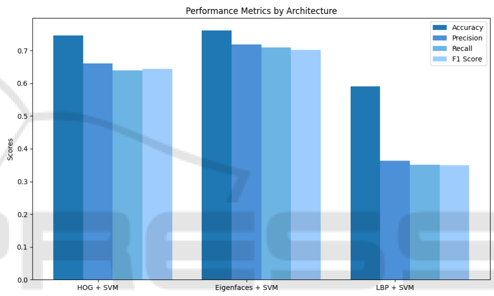

Figure 10: Different metrics by architecture.

ROC Curve

The ROC (Swets and Pickett, 1988) curves below illustrate the performance of different models in terms of the true positive rate (sensitivity) against the false positive rate (1-specificity).

Based on the ROC curves, we can observe that the HOG-SVM model achieved the highest area Groupe Shopping lyonnaise funder the curve (AUC), indicating superior performance in distinguishing between positive and negative samples.

Precision-Recall Curve

The Precision-Recall curves below illustrate the trade-off between precision and recall for different classification models. Precision-Recall curves are useful when the classes are imbalanced, as they provide insights into the classifier's performance across different decision thresholds.

Based on the Precision-Recall curves, we can observe

Table 5: Summary of Model Performance Metrics.

Model	Accuracy	Precision	Recall	F1 Score
HOG-SVM	0.745	0.472	0.878	0.644
Eigenfaces-SVM	0.755	0.456	0.856	0.674
LBP-SVM	0.590	0.435	0.840	0.349

that the HOG-SVM model achieved higher precision-recall values compared to other models across various thresholds. This indicates that the HOG-SVM model is better at identifying positive samples while maintaining high precision, making it more suitable for the task.

After Optimization Results

After fine-tuning our models, we obtained the following performance metrics:

Table 6: Performance metrics after fine-tuning.

Model	Accuracy	Precision	Recall	F1 Score
HOG-SVM	0.795	0.485	0.890	0.682
Eigenfaces-SVM	0.845	0.490	0.894	0.783
LBP-SVM	0.609	0.443	0.853	0.372

From the results in the table 3, we observe that the Eigenfaces + SVM approach achieved the highest accuracy of 84.5%, with relatively balanced precision, recall, and F1-score. HOG-SVM also performed reasonably well with an accuracy of 79.5%, demonstrating good recall but lower precision. However, the LBP-SVM approach showed lower performance across all metrics.

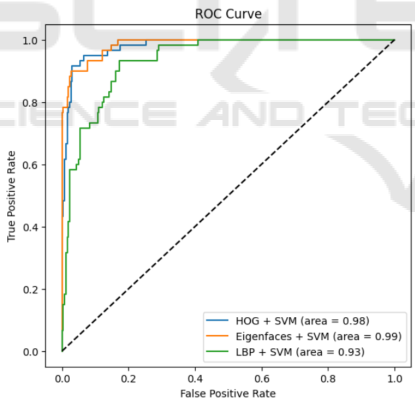

Figure 11: ROC Curves of Different Models after Hyperparameters Optimization.

8.3 Comparison of Results Before and after Hyperparameters Optimization

We compared three feature extractors (HOG, Eigenfaces, and LBP) with the SVM classifier. The performance metrics before and after hyperparameters optimization are summarized in the following table:
The results demonstrate clear improvements in model performance after hyperparameters optimization. Initially, the HOG + SVM model achieved 74.5% accuracy with an F1 score of 0.644. After fine-tuning, the Eigenfaces + SVM model showed the greatest improvement, with accuracy rising from 75.5% to 84.5% and the F1 score increasing from 0.674 to 0.783. The HOG + SVM model also improved, reaching 79.5% accuracy and an F1 score of 0.682. Although the LBP + SVM model saw only slight gains in accuracy and F1 score, it still performed lower than the other models. These results highlight the value of the hyperparameters optimization in boosting model accuracy and suggest that Eigenfaces is the most effective feature extractor for SVM on the LFW dataset.

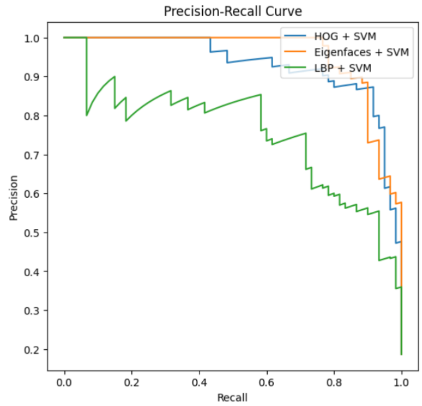

Figure 12: Precision-Recall Curves of Different Models after Hyperparameters Optimization.

Table 7: Performance comparison before and after hyperparameter optimization.

Model	Stage	Accuracy	Precision	Recall	F1 Score
HOG-SVM	Before	0.745	0.472	0.878	0.644
HOG-SVM	After	0.795	0.485	0.890	0.682
Eigenfaces-SVM	Before	0.755	0.456	0.856	0.674
Eigenfaces-SVM	After	0.845	0.490	0.894	0.783
LBP-SVM	Before	0.590	0.435	0.840	0.349
LBP-SVM	After	0.609	0.443	0.853	0.372

8.4 Evaluation of Facial Recognition Models on LFW Dataset and Augmented Dataset

We evaluate the performance of three different facial recognition models—HOG+SVM, LBP+SVM, and Eigenfaces+SVM—using both the original LFW dataset and the augmented LFW dataset generated with the diffusion model. The aim is to assess how the introduction of augmented images, which include variations in lighting, facial expressions, and poses, influences the accuracy of the models compared to training solely on the original LFW dataset.

The augmentation process, which includes generating additional images for each individual in the LFW dataset, allows the models to benefit from a more diverse range of facial variations, which typically improves their generalization capabilities. By leveraging these augmented images, the models are exposed to a wider variety of conditions, helping them learn more robust feature representations.

We observe that training the models with the augmented dataset yields better results compared to training on the original LFW dataset. This improvement in accuracy demonstrates the benefits of using generated images to enhance the diversity and complexity of the training data. The following table summarizes the performance metrics for each model on both the original and augmented LFW datasets:

Table 8: Models Performance Metrics on LFW with Generated Images.

Model	Accuracy	Precision	Recall	F1 Score
HOG-SVM	0.782	0.77	0.896	0.661
Eigenfaces-SVM	0.961	0.94	0.984	0.877
LBP-SVM	0.957	0.712	0.970	0.642

As shown in Table 8, the accuracy for the HOG-SVM and Eigenfaces-SVM models significantly improves when trained on the augmented dataset, while the LBP-SVM model also benefits from the additional data, albeit to a lesser extent.

These results highlight the importance of diverse and augmented data in improving the performance of facial recognition models, especially in challenging real-world scenarios where variations in facial expressions, lighting, and poses are common. The augmentation process through the diffusion model has proven to be particularly beneficial in this context, as it allows the model to generalize better by exposing it to more varied representations of facial features, which may not be present in the original dataset.

This table 9 compares the best accuracies of face recognition methods obtained in my study with those from related work, all evaluated on the LFW dataset. The results reveal that the Eigenfaces-SVM method outperforms most of the methods in the related work, achieving the highest accuracy of 0.961. This performance is a notable improvement over the related works, including well-established methods like PCA-SVM and CNN, which achieved accuracies of 0.8413 and 0.7998, respectively. The HOG-SVM method, which also showed promising results in this study with an accuracy of 0.782, surpasses other methods like HOG-SVM from previous studies, which achieved 0.644. The LBP-SVM method, however, demonstrated an impressive result of 0.957 in the current study, which contrasts with the much higher accuracy of 0.9481 reported in the related work.

Table 9: Comparison of Accuracies on LFW Dataset (Alamri et al., 2022).

Method	**Accuracy**
Related Work	
PCA(Yin et al., 2011)	0.8445
PCA - SVM (Duan et al., 2019)	0.8413
CNN (A et al., 2015)	0.7998
SIFT (Ahmed et al., 2018)	0.711
HOG-SVM (Dadi and Pillutla, 2016)	0.644
Eigenfaces-SVM (Aliyu et al., 2022)	0.831
LBP-SVM (Shan, 2011)	0.9481
SIFT - SVM	0.658
Current Study	
HOG-SVM	0.782
Eigenfaces-SVM	0.961
LBP-SVM	0.957

This discrepancy might be due to the differences in data augmentation strategies or model configurations used across studies. Overall, these results confirm that the Eigenfaces-SVM and HOG-SVM methods are strong contenders for face recognition tasks, with Eigenfaces-SVM emerging as the most effective approach among the models tested.

9 CONCLUSION

This work highlights the importance of both advanced data augmentation techniques, such as diffusion models, and the selection of effective feature extraction methods for improving the performance of facial recognition systems. By comparing three well-established algorithms for feature extraction—Eigenfaces, Local Binary Patterns (LBP), and Histogram of Oriented Gradients (HOG)—we were able to assess their suitability when combined with a Support Vector Machine (SVM) classifier for facial recognition tasks.

Through extensive experimentation and evaluation on both the original and augmented LFW dataset, it became evident that the choice of feature extractor plays a crucial role in the overall performance of the facial recognition model. Among the three algorithms tested, Eigenfaces-SVM demonstrated the highest accuracy and overall performance, followed by HOG-SVM, with LBP-SVM achieving the lowest results. The Eigenfaces method, which captures the global structure of faces through Principal Component Analysis (PCA), was particularly effective in distinguishing subtle variations in facial features, resulting in superior accuracy, precision, recall, and F1 score. HOG, known for its ability to capture edge and texture information, also showed strong performance but was not as robust as Eigenfaces in handling varied facial expressions and lighting conditions. On the other hand, LBP, which is more sensitive to local texture variations, underperformed compared to the other two methods, particularly in more complex scenarios involving diverse lighting and poses.

Additionally, the introduction of the diffusion model for data augmentation significantly contributed to improving the performance of all three models. The synthetic images generated by the diffusion model enhanced the diversity of the training data, providing the models with a broader range of facial variations. This led to a noticeable improvement in the recognition accuracy, especially when compared to training on the original LFW dataset alone. The augmented data allowed the models to better generalize to real-world conditions, which often involve diverse facial expressions, poses, and lighting conditions.

REFERENCES

A, V., Hebbar, D., Shekhar, V. S., Murthy, K. N. B., and Natarajan, S. (2015). Two novel detector-descriptor based approaches for face recognition using sift and surf. *Procedia Computer Science*, 70:185–197.

Ahmed, A., Guo, J., Ali, F., Deeba, F., and Ahmed, A. (2018). Lbph based improved face recognition at low resolution. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 144–147.

Alamri, H., Alshanbari, E., Alotaibi, S., and AlGhamdi, M. (2022). Face recognition and gender detection using sift feature extraction, lbph, and svm. *Engineering, Technology & Applied Science Research*, 12(2):8296–8299.

Aliyu, I., Bomoi, M. A., and Maishanu, M. (2022). A comparative study of eigenface and fisherface algorithms based on opencv and sci-kit libraries implementations. *International Journal of Information and Electronics Engineering*, 14(3):35. Accuracy: 0.8310 for Eigenfaces and SVM.

Cheng Quanhua, Liu Zunxiong, D. G. (2008). Facial gender classification with eigenfaces and least squares support vector machine. pages 28–33.

Dadi, H. S. and Pillutla, G. K. M. (2016). Improved face recognition rate using hog features and svm classifier. *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, 11(4):34–44.

Duan, Y., Lu, J., and Zhou, J. (2019). Uniformface: Learning deep equidistributed representation for face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3410–3419.

Hilbert, A., Madai, V. I., Akay, E. M., and Aydin, O. U. (2020). Brave-net: Fully automated arterial brain vessel segmentation in patients with cerebrovascular disease. *Frontiers in Artificial Intelligence*, 3:552258.

Jae Jeong Hwang, Young Min Kim, K. H. R. (2018). Faces recognition using haarcascade, lbph, hog and linear svm object detector. In *Proceedings of the Sixth International Conference on Green and Human Information Technology*, pages 232–236.

Kancherla Deepika, Jyostna Devi Bodapati, R. K. S. (2019). An efficient automatic brain tumor classification using lbp features and svm-based classifier. *Proceedings of International Conference on Computational Intelligence and Data Engineering*, pages 163–170.

Kassel, R. (2021). U-net : le réseau de neurones de computer vision.

Mittal, A. (2024). Comprendre les modèles de diffusion : une plongée en profondeur dans l'ia générative.

Moore, R. and Lopes, J. (1999). Paper templates. In *TEMPLATE'06, 1st International Conference on Template Production*. SCITEPRESS.

Rajaa, S., HARRABI, R. M. s., and Chaabane, S. B. (2021). Facial expression recognition system based on svm and hog techniques. In *International Journal of Image Processing (IJIP)*, pages 14–21.

Rosebrock, A. (2021). Opencv eigenfaces for face recognition2. In *PyImageSearch*.

Safa Rajaa, Rafika Mohamed salah HARRABI, S. B. C. (2021). Facial expression recognition system based on svm and hog techniques. pages 14–21.

Shan, C. (2011). Learning local binary patterns for gender classification on real-world face images. *Pattern Recognition Letters*, 32(10):1318–1325.

Shubhangi Patil, Y. M. P. (2022). Face expression recognition using svm and knn classifier with hog. In *Unknown*.

Smith, J. (1998). *The Book*. The publishing company, London, 2nd edition.

Swets, J. A. and Pickett, R. M. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240(4857):1285–1293.

Yin, Q., Tang, X., and Sun, J. (2011). An associate-predict model for face recognition. In *CVPR 2011*, pages 497–504.