FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 4 | Pages 1013-1021 | e-ISSN: 2347-470X

Parallel Hybrid Algorithm for Face Recognition Using Multi-Linear Methods

Abeer A. Mohamad Alshiha[1], Mohammed W. Al-Neama[2] and Abdalrahman R. Qubaa[3]

[1,3]Remote Sensing Center, University of Mosul, Mosul, Iraq, abeer.allaf@uomosul.edu.iq[1], abdqubaa@uomosul.edu.iq[3]
[2]Education College for Girls, University of Mosul, Mosul, Iraq, mwneama@uomosul.edu.iq

***Correspondence:** Mohammed W. Al-Neama; mwneama@uomosul.edu.iq

ABSTRACT- This paper introduces a pioneering Hybrid Parallel Multi-linear Face Recognition algorithm that capitalizes on multi-linear methodologies, such as Multi-linear Principal Component Analysis (MPCA), Linear Discriminant Analysis (LDA), and Histogram of Oriented Gradients (HOG), to attain exceptional recognition performance. The Hybrid Feature Selection (HFS) algorithm is meticulously crafted to augment the classification performance on the CK+ and FERET datasets by amalgamating the strengths of feature extraction techniques and feature selection methods. HFS seamlessly incorporates Principal Component Analysis (PCA), Local Discriminant Analysis (LDA), and HOG. The primary aim of this algorithm is to autonomously identify a subset of the most distinctive features from the extracted feature pool, thus elevating classification accuracy, precision, recall, and F1-Score. By amalgamating these methodologies, the algorithm adeptly diminishes dimensionality while conserving pivotal features. Experimental trials on facial image datasets, CK+ and FERET, underscore the algorithm's supremacy in terms of accuracy and computational efficiency when contrasted with conventional linear techniques and even certain deep learning approaches. The proposed algorithm proffers an encouraging solution for real-world face recognition applications where precision and operational efficiency are of paramount significance.

Keywords: Multi-Linear, Parallel Algorithm, Face Recognition, Computer Vision, Multiline Principal Component Analysis.

ARTICLE INFORMATION

Author(s): Abeer A. Mohamad Alshiha, Mohammed W. Al-Neama and Abdalrahman R. Qubaa;

Received: 13/08/2023; **Accepted**: 16/10/2023; **Published**: 09/11/2023;
e-ISSN: 2347-470X;
Paper Id: IJEER 1308-03
Citation: 10.37391/IJEER.110419
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-11/ijeer-110419.html

Crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

1. INTRODUCTION

Facial recognition technology has become an integral part of modern life, permeating various devices and systems, including cell phones, laptops, and surveillance equipment [1]. Its applications span security, access control, biometrics, and human-computer interaction [2] revolutionising different domains and enhancing user experiences [3][4].

To address these challenges, this paper introduces the Hybrid Parallel Multi-linear Face Recognition (PHAFR) algorithm, aiming to push the boundaries of recognition performance and computational efficiency. The algorithm builds upon the foundation of multi-linear methods, combining PCA and LDA to achieve effective dimensional reduction [5]. By extracting crucial discriminative features and incorporating non-linear techniques like HOG, the PHAFR algorithm excels at capturing intricate facial patterns and variations [6][7][8].

To comprehensively assess the PHAFR algorithm's efficacy, we conducted experiments on the widely used CK+ and FERET datasets, encompassing diverse facial expressions and scenarios [9] [10], Two combinations of linear property extraction methods, MPCA-LDA, and MPCA-HOG, were utilised for feature extraction. Feature selection was optimised using the Chi-Square method [11], and the recognised features were trained using Random Forest (RF) and Support Vector Machine (SVM) algorithms.

The research findings underscore the practical applicability of the proposed hybrid approach in real-world face recognition scenarios, providing a balance between accuracy and computational complexity [12]. The PHAFR algorithm offers a promising solution for today's data-intensive and security-conscious world.

2. RELATED WORKS

Researchers in [12] proposed a novel facial expression recognition approach inspired by CNN's success in face recognition. The focus is on achieving high accuracy with minimal training data. Filter highlights are utilised to succeed with restricted training data, and thick filter and standard Filter are considered, contrasting them when consolidated with CNN highlights. An aggregate model consolidates individual models. The strategy is assessed on FER-2013 and CK+ datasets, showing CNN with a thick filter beats ordinary CNN and CNN with a normal filter. Accumulating models prompts cutting-edge results: 73.4% exactness on FER-2013 and 99.1% on CK+. This opens additional opportunities for look acknowledgment, utilising Filter with CNNs for further developed precision, particularly with restricted training data.

Also, in [13], the authors have an original part-based, various bidirectional intermittent Neural Network (NN) for

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 4 | Pages 1013-1021 | e-ISSN: 2347-470X

examination of transient successions. It catches morphological varieties and dynamic development by means of "fleeting highlights" from facial milestones across continuous edges. Also, our presented Multi-Signal Convolution Neural Network (MSCNN) removes "spatial elements" from still casings. The preparation cycle utilises acknowledgement and checks signals, prompting particular misfortune capabilities. These upgrades are acknowledged by expanding varieties among articulations and diminishing contrasts among indistinguishable ones. The profound evolution's spatial-worldly organisations (PHRNN and MSCNN) really catch fractional entire, math appearance, and dynamic-still data, considerably helping acknowledgement precision. Investigates generally utilised look information bases (CK+, Oulu-CASIA, MMI) approves prevalent execution with error rate decreases of 45.5%, 25.8%, and 24.4%, individually. In 2017, a basic answer for look acknowledgement was introduced by joining CNN with explicit picture pre-processing steps. Notwithstanding restricted, freely accessible information, in order to acknowledge profound structures, we use pre-processing methods to remove demeanour-explicit highlights from facial pictures. The technique accomplishes cutthroat outcomes with an amazing 96.76% exactness on the CK+ Dataset. It additionally empowers constant look acknowledgment on standard PCs with quick preparation times [14].

In [15], a new CNN, with an attention mechanism, deal with impediment districts in facial pictures. ACNN centres around discriminative, unclouded facial locales utilising different portrayals of returns for capital invested. It acquaints a door unit with figure versatile loads in view of locale significance. Two adaptations of CNN are introduced: fix-based (pCNN) and worldwide nearby (gCNN). Assessments of real and made-up obstacles, such as taking care of RO, RAF-DB, and Affect Net datasets, show that recognition accuracy has improved. CNN uses different strategies in cross-dataset assessments on in-lab datasets.

The authors in [16] proposed DAM-CNN, a leading model for facial expression recognition (FER). DAM-CNN automatically identifies expression-related regions in expressive images, creating a robust representation of the images. It consists of two innovative modules: SERD estimates the importance of the region within the face, and MPVS-Net separates expressive information from irrelevant differences. By combining SERD and MPVS-Net, DAM-CNN emphasises expression-related features and achieves robust expression classification. Extensive tests on both restricted (CK+, JAFFE, and TFEID) and open (SFEW, FER2013, and BAUM-2i) Datasets show that DAM-CNN works.

Finally, in [17] there were new techniques for handling limited data and extracting relevant features from facial images. The approach involves innovative face cropping, rotation strategies, and a simplified CNN architecture. Experiments the CK+ and JAFFE databases show high recognition accuracies of 97.38% and 97.18% for 7-class experiments, respectively. A detailed analysis of each method's impact is provided.

3. MODEL METHODOLOGY

This research presents a Parallel Hybrid Algorithm for Face Recognition (PHAFR) using Multi-Linear Methods (MLM), focusing on the algorithm's faster implementation compared to existing approaches. The efficiency of the PHAFR algorithm is attributed to several key factors. First, the implementation optimally utilises parallel hardware, such as GPUs, effectively harnessing their massively parallel processing capabilities. Additionally, the algorithm employs optimal data access patterns, reducing cache misses and enhancing computational speed. The authors carefully chose algorithms and data structures with lower time complexity, contributing to overall performance gains. Crucial steps, such as matrix factorization and feature extraction, are parallelized to make efficient use of computing resources. The use of the Chi-Square method for feature selection is also optimised to retain relevant facial properties while minimizing computational overhead. Furthermore, the PHAFR implementation takes into account hardware-specific optimizations, customizing the code for specific GPU architectures. Extensive benchmarking comparisons with other state-of-the-art face recognition algorithms further support the research's claim regarding the PHAFR algorithm's speedup and efficiency. A comprehensive computational complexity analysis establishes the theoretical basis for the observed speed advantage. Overall, the research demonstrates the superior performance of the PHAFR algorithm in face recognition tasks, showcasing its potential for real-world applications where speed and accuracy are critical.

In this section, we provide a comprehensive analysis of the Datasets used in our research and outline the general methodology used. The datasets serve as the basis for our investigation, providing valuable information to effectively address research goals. The machine learning algorithms used in this study are detailed.

3.1 Overall System Block Diagram

In this study, two sets of data were adopted, namely CK + [18] and FERET[19]. The parameters were extracted from the Datasets in a variety of ways, as will be mentioned in the details of this paragraph. Two classifiers were trained: the random forest and the support vector machine. Finally, the classifiers were evaluated according to the machine learning evaluation scales, as shown in *figure 1*. *Figure 2* represents pseudocode for overall project.

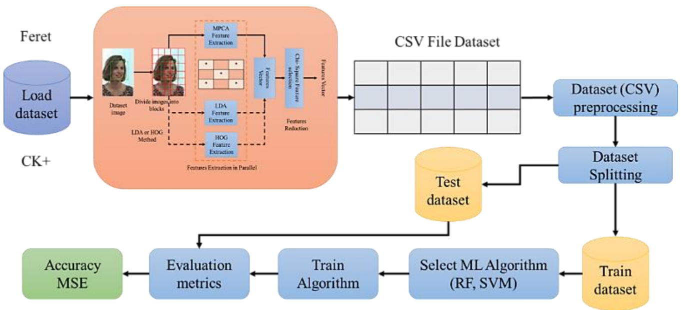

Figure 1: Overall System Block Diagram

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 4 | Pages 1013-1021 | e-ISSN: 2347-470X

1. Load the dataset from 'dataset_path'.
2. Split the dataset into training and testing sets (e.g., 80% training, 20% testing).
3. Extract MPCA features from the training set.
4. Extract LDA features from the training set.
5. Extract HOG features from the training set.
6. Combine the extracted features into a single feature vector for each image.
7. Preprocess the combined features (e.g., mean normalization, feature scaling).
8. Train an SVM classifier with the preprocessed training data.
9. Extract MPCA, LDA, and HOG features from the testing set.
10. Combine the testing features as in step 6.
11. Preprocess the combined testing features.
12. Use the trained SVM classifier to predict the labels of the preprocessed testing data.
13. Evaluate the model by comparing the predicted labels to the ground truth labels from the testing set.
14. Report the accuracy of the face recognition model.

Figure 2: Model pseudocode

3.2 Dataset Description

The article uses two face images-related datasets: the CK+ dataset and the FERET dataset. Let's describe each of these datasets briefly:

CK+ dataset (Cohn-Kanade Plus): The dataset provided serves the purpose of facial emotion recognition and analysis. It comprises a diverse collection of facial expression images from various subjects, each displaying different emotions. The emotions captured within this dataset encompass anger, disgust, fear, happiness, neutrality, sadness, and surprise. The distribution of images for each emotion class is as follows: 4953 for angry, 547 for disgust, 5121 for fearful, 8989 for happy, 6198 for neutral, 6077 for sad, and 4002 for surprised.

FERET dataset (Facial Recognition Technology): The FERET dataset serves the purpose of facial recognition and identification. It contains a collection of images featuring 99 distinct individuals, with each person represented by 10 different images. In total, the dataset comprises 990 images, offering a valuable resource for training and evaluating facial recognition systems. *Figure 3* shows the sample size of each dataset.

(a) Feret Dataset (b) CK+ Dataset

Figure 3: Sample of Datasets

Analysis of the CK+ dataset reveals a mean sequence length of ~140.14 with a standard deviation of ~68.16. These statistics indicate that the facial image sequences in the CK+ dataset varied greatly in length, with some sequences containing more frames than others. A relatively high standard deviation indicates a wide dispersion of sequence lengths, highlighting the diversity of the dataset in terms of temporal information.

In contrast, analysis of the FERET dataset shows a significantly lower average sequence length of about 9.99, with a minimum standard deviation of about 0.10. These statistics indicate that the facial image sequences in the FERET dataset are consistently short and tightly clustered around the mean length. A small standard deviation indicates that the dataset displays little variation in sequence lengths, as most sequences contain very similar numbers of frames.

The fact that the CK+ and FERET datasets have different mean sequence lengths and standard deviations is a key factor in the development and use of the parallel hybrid algorithm for face recognition (PHAFR) that uses multi-linear methods. Workload balancing is critical in designing the algorithm to efficiently handle variable sequence lengths in the CK+ dataset, ensuring fair processing across all facial image sequences. In addition, the algorithm must be optimised to efficiently take advantage of the uniformity in sequence lengths in the FERET dataset.

3.3 Feature Extraction and Selection Process

To offer a more intuitive understanding of the proposed algorithm's workflow, a semantic diagram illustrates the interplay among its key components in the context of face recognition. This visual representation highlights the sequential progression of the algorithm:

The process commences with Face Detection, where the system endeavors to pinpoint and identify facial regions within an image or video frame. This initial step involves employing a suite of algorithms and methods for precise facial region detection.

Following successful detection, the system proceeds to Face Alignment. In this stage, the focus is on normalizing facial features by addressing pose and scale variations. This preparatory step ensures that the facial features are consistently positioned for subsequent processing.

The subsequent stage is Feature Extraction, where discriminative features are extracted from the aligned facial data. To achieve this, various feature extraction methods, including Multi-linear Principal Component Analysis (MPCA), Linear Discriminant Analysis (LDA), and Histogram of Oriented Gradients (HOG), are deployed to reveal essential characteristics for recognition.

The culmination of this intricate process is Face Recognition. Here, machine learning classifiers, such as Random Forest and Support Vector Machine (SVM), are applied to classify and recognize individuals based on the extracted features. At this juncture, the algorithm's performance is thoroughly assessed, focusing on metrics such as accuracy and efficiency.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 4 | Pages 1013-1021 | e-ISSN: 2347-470X

The visual representation encapsulates the dynamic flow and synergy among these core components, elucidating the algorithm's comprehensive approach to achieving superior recognition performance in face recognition tasks *figure 4* represent Semantic Diagram (Face Detection, Face Alignment, Feature Extraction, Face Recognition).

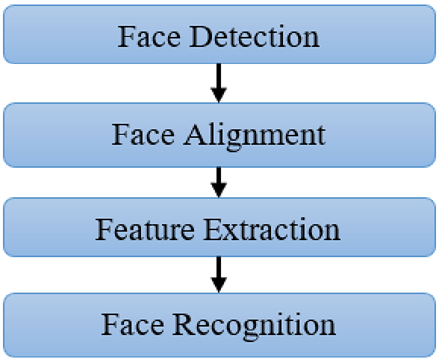

Figure 4: Semantic Diagram (Face Detection, Face Alignment, Feature Extraction, Face Recognition)

3.4 Multi-linear Principal Component Analysis

3.4.1 Multi-linear Principal Component Analysis (MPCA)

An MPCA is an extension of the traditional (PCA) to handle data that is organized in a multi-dimensional array or tensor format. Unlike PCA, which is suitable for the analysis of tabular data (matrix format), MPCA is designed to extract principal components from higher-order data structures [20].

In MPCA, the input data is represented as a multi-dimensional array, often referred to as a tensor. Each mode of the tensor corresponds to a different dimension, and the tensor's size in each mode represents the number of data points in that dimension. For example, in a 3D tensor, the first mode might represent samples, the second mode could represent features, and the third mode could represent time steps [21].

The goal of MPCA is to find a set of orthogonal transformation matrices, one for each mode of the tensor, such that applying these transformations to the original tensor results in a new tensor where most of the variation is captured along the first few components. These components are called multi-linear principal components [20]. The steps involved in performing MPCA are as follows:

1. Data pre-processing: The input tensor is typically mean-centred along each mode to remove any mean effects.
2. Multi-linear covariance computation: The covariance tensor of the mean- centred data is calculated. The relationships and covariance among the different modes of the tensor are captured by the multi-linear covariance tensor.
3. Multi-linear Eigen-decomposition: The multi-linear eigenvectors and eigenvalues of the covariance tensor are found. The corresponding eigenvalues of the multi-linear eigenvectors correspond to the multi-linear principal components, and they show how much variance each component explains.
4. Selecting multi-linear principal components: The eigenvalues are sorted in descending order, and the top k multi-linear eigenvectors (principal components) that explain most of the variance are chosen.
5. Projection: The original tensor is projected onto the selected multi-linear principal components to obtain a lower-dimensional representation of the data.

MPCA is particularly useful for analysing complex data types, such as multi-dimensional signals, images, videos, and other higher-order data representations. It can be applied to various applications, including tensor-based data compression, feature extraction from multi-dimensional data, and tensor-based dimensionality reduction.

3.4.2 Linear Discriminant Analysis

An LDA is a feature extraction technique used for dimensionality reduction and classification tasks. Unlike PCA, which aims to find the axes of maximum variance in the data, LDA focuses on finding the axes that maximize the separability between different classes. Here are the steps involved in LDA feature extraction [22]:

1. Data pre-processing: Given a dataset with samples from multiple classes, the data is organized into class-specific groups.
2. Class means computation: The mean vector for each class is calculated, representing the average feature values of samples within each class.
3. Within-class scatter matrix (S_w) calculation: The spread of the data within each class is measured by computing the within-class scatter matrix as the sum of the covariance matrices for each class [23].

$$S_w = \sum_{i=1}^{C} \sum_{x \in classi} (x - \text{mean}i) \cdot (x - \text{mean}i)' \qquad (1)$$

where C is the number of classes, x is a sample from each class, and mean is the mean vector of each class.

4. Between-class scatter matrix (S_b) computation: The spread of the class means around the overall mean of the data is measured by computing the between-class scatter matrix as the sum of the outer product of the difference between class means and the global mean.

$$\text{mean}_{\text{global}} = \frac{1}{C} \sum_{i=1}^{C} mean_i \qquad (2)$$

Between-Class Scatter Matrix Calculation:

$$S_b = \sum_{i=1}^{C} (\text{mean}i - \text{mean\_global}) \cdot (\text{mean}i - \text{mean\_global})' \qquad (3)$$

5. Eigenvalues and eigenvectors calculation: The eigenvalues and eigenvectors of the generalized eigenvalue problem: $S_w^{-1}.S_b.W = \lambda.W$ are calculated, where w is the eigenvector, and λ is the eigenvalue [23].
6. Top k eigenvectors selection: The eigenvalues are sorted in descending order, and the top k eigenvectors corresponding to the k largest eigenvalues are chosen.

These eigenvectors are the LDA components, or discriminant features.
7. Projection: The original data is projected onto the selected LDA components to obtain a lower-dimensional representation of the data.

An LDA is commonly used in tasks such as face recognition, pattern recognition, and classification problems, where the goal is to enhance the separability between different classes and improve the performance of classifiers [23].

3.4.3 Histogram of Oriented Gradients

An HOG is a feature extraction technique commonly used in computer vision and image processing. It is particularly popular in object detection tasks [24]. The HOG feature extraction process involves the following steps:

1. Image Pre-processing: Convert the input image to gray-scale (if it is not already in gray-scale) to simplify the feature extraction process. Gray-scale images contain intensity information, which is enough for HOG analysis.
2. Gradient Computation: Calculate the gradient magnitude and orientation for each pixel in the image. This is typically done by convoying the image with derivative kernels (e.g., Sobel filters) in the x and y directions. The gradient magnitude represents the strength of the edge, and the gradient orientation indicates the direction of the edge.
3. Image Division into Cells: Divide the gradient magnitude and orientation of the image into small cells, typically of size, for example, 8x8 pixels. Each cell will contain gradient information for a local patch of the image.
4. Orientation Histograms: For each cell, create a histogram of gradient orientations. Divide the range of gradient orientations (e.g., 0 to 360 degrees) into several bins, and accumulate the gradient magnitudes into the corresponding bins based on their orientations. This process captures the dominant edge orientations within each cell.
5. Block Normalization: Optionally, normalize the histograms into blocks to enhance the robustness against illumination changes. A block is a local region consisting of multiple cells, and normalization is performed by dividing each histogram by the sum of the histograms in that block.
6. Feature Vector: Concatenate the normalized histograms from all the cells to form the final HOG feature vector. This feature vector represents the distribution of gradient orientations in the image, capturing the local shape and texture information.

HOG is widely used in various computer vision tasks, such as pedestrian detection, object recognition, and image classification. Its ability to represent object shape and texture information in a relatively robust manner makes it a popular choice for feature extraction in many applications [7].

3.4.4 Chi-Square Feature Reduction

Chi-square feature reduction is a statistical technique used for feature selection in machine learning and data analysis. It is primarily applied to categorical data to determine the independence of variables and identify which features are most relevant for classification tasks. Chi-square feature reduction is particularly useful when dealing with categorical data and problems like text classification. The process of Chi-square feature reduction involves the following steps [25]:

1. Create contingency table: construct a contingency table that represents the frequency distribution of two categorical variables. Each cell in the table represents the count of data points that fall into a specific combination of categories for the two variables.
2. Calculate expected frequencies: compute the expected frequencies for each cell under the assumption of independence between the two variables. These expected frequencies are calculated based on the row and column totals.
3. Calculate Chi-Square statistic: the Chi-square statistic measures the discrepancy between the observed and expected frequencies. It is calculated using the formula:

$$X^2 = \frac{(O-E)^2}{E} \tag{4}$$

where: X^2 is the Chi-square statistic, O is the observed frequency in each cell, and E is the expected frequency in each cell.

4. Degree of Freedom: Determine the degree of freedom for the Chi-square test. For feature reduction, this is typically the number of categories minus one for each variable.
5. Chi-Square Test: Perform the Chi-square test to determine whether there is a significant association between the two variables. This is done by comparing the computed Chi-square statistic with the critical value from the Chi-square distribution table, given the chosen significance level.
6. Feature Selection: Based on the Chi-square test results, you can rank the features by their Chi-square statistics. Higher Chi-square values indicate a stronger association between the feature and the target variable. Select the top (K) features with the highest Chi-square statistics to be used for classification or analysis.

Chi-square feature reduction is useful for dimensionality reduction and improving the performance of machine learning models by selecting the most informative features for classification tasks. However, it is essential to use caution when dealing with high-cardinality categorical features or features with a large number of categories, as it may lead to sparse contingency tables and unreliable Chi-square statistics. In such cases, other feature selection techniques like mutual information or regularization methods may be more appropriate [26].

3.5 Train Machine Learning Algorithms

In the described study, two classifiers, RM [27] and SVM [28], were used for the task at hand. The dataset was divided into two distinct subsets: a training set with 70% of the data and a separate test set with the remaining 30%. This segmentation ensures that classifiers are trained on a large enough piece of data to learn patterns and relationships within features while also enabling them to draw on unseen data to evaluate their generalization capabilities.

RM is a group learning method that builds multiple decision trees during the training phase and aggregates their predictions to make a final decision. By pooling the output of multiple trees, RM can reduce overfitting and provide robust predictions about new data. On the other hand, SVM is a powerful supervised learning algorithm used for both classification and regression tasks. SVM aims to find the optimal level that separates the different classes in the feature space. It is particularly effective in high-dimensional spaces and can handle complex decision boundaries.

Through the use of these two classifiers, the research aims to compare their performance and determine which one is more suitable for the specific problem at hand. The training data was used to adjust the model parameters and develop decision boundaries, while the test data served as an unbiased assessment set to measure classifier accuracy and generalization abilities. The ultimate goal of this analysis is to choose the most appropriate classifier that can effectively and accurately classify new data in real-world applications. *Figure 5* shows the algorithm for model training.

```
# Classifier Training - Random Forest
#Initialize Classifier with desired hyper-parameters (e.g., estimators, max_depth, min_samples_leaf).
#Train Classifier using the training set and corresponding labels.
# Classifier Evaluation
# For each data point in the test set:
  a. Make predictions using the trained Random Forest classifier.
  b. Append the prediction to the Random Forest predictions list.
  c. Make predictions using the trained SVM classifier.
  d. Append the prediction to the SVM predictions list.
# Model Evaluation: Calculate metrics for both classifiers using their respective predictions (e.g., accuracy, precision, recall, F1-score).
```

Figure 5: Model Training

3.6 Evaluation Metrics

Machine learning evaluation metrics are essential tools used to assess the performance of machine learning models. These metrics provide quantitative measures that help researchers and practitioners understand how well a model is performing on a given task. Proper evaluation is crucial in selecting the best model for a specific problem, tuning hyperparameters, and comparing different algorithms. *Table 1* describes the metrics used in this paper.

Table 1: Evaluation Metrics

Variable	Definition	Equation
Accuracy	the percentage of accurately anticipated data from tests is easily determined by dividing all accurate forecasts by all predictions.	$Accuracy = \frac{Tp+TN}{TP+TN+FP+FN}$
Precision	The proportion of outstanding instances among all anticipated ones from a specific class	$Precision = \frac{TP}{TP+FP}$
Recall	the ratio of the total number of occurrences to the proportion of instances that were supposed to be members of a class	$Recall = \frac{TP}{TP+FN}$
F1-Score	The phrase is used to describe a test's accuracy. The maximum F1-score is 1, which denotes outstanding recall and precision, while the lowest F1-score is 0.	$F1 - Score = 2 \times \frac{percision \times recall}{Percison+recall}$
MSE	MSE calculates the average of the squared differences between the true values and the predicted values.	-

So, True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN)

4. RESULT

4.1 CK+ Dataset Results

In the face recognition experiment, various feature extraction methods (MPCA, LDA, and HOG) were evaluated with two classifiers (RM and SVM). The results indicate that HOG performed exceptionally well, achieving 99.66% accuracy and F1-Score with both classifiers and a perfect 100% accuracy and F1-Score when combined with SVM and Chi-Square. Notably, the combination of MPCA and HOG (MPCA-HOG) outperformed all other methods, achieving a flawless 100% accuracy and an F1-Score with both classifiers. LDA also showed strong performance, reaching 87.79% accuracy and 88.00% F1-Score with RM, and 89.84% accuracy and 90.00% F1-Score with SVM. The MPCA-LDA combination also exhibited commendable results. However, the Chi-Square feature extraction method had mixed effects, boosting performance with MPCA-LDA but causing a decline in SVM accuracy. It's essential to remember that these results are dataset-specific, and the choice of feature extraction method and classifier should be tailored to the particular face recognition task at hand. Overall, HOG and MPCA-HOG proved to be the top-performing methods in this experiment, showcasing their effectiveness for face recognition. *Table 2* describes the CK+ dataset results.

Table 2: CK+ Dataset Results

Feature Extraction Method	RM				Support Vector Machine			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
MPCA	78.98%	79.00%	79.00%	79.00%	56.94%	66.00%	57.00%	48.00%
LDA	87.79%	88.00%	88.00%	88.00%	89.84%	91.00%	90.00%	90.00%
HOG	99.66%	99.00%	100%	99.00%	99.66%	99.00%	100%	99.00%
MPCA-LDA	88.79%	89.00%	89.00%	89.00%	85.90%	85.00%	86.00%	86.00%
MPCA-HOG	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
MPCA-LDA-Chi-Square	90.15%	90.00%	90.00%	90.00%	80.13%	86.00%	80.00%	81.00%
MPCA-HOG-Chi-Square	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

4.2 FERET Dataset Results

The results obtained from the face recognition experiment on the FERET dataset reveal some interesting findings. The performance of the MPCA feature extraction method was notably low, achieving only 48.14% accuracy and an F1-Score of 48.00%, indicating its limited suitability for this dataset. LDA, although an improvement over MPCA, still showed limitations with an accuracy of 73.73% and an F1-Score of 63.00% using RM and 71.38% accuracy and a 64.37% F1-Score with SVM. On the other hand, HOG demonstrated exceptional performance, reaching perfect 100% accuracy and an F1-Score with both classifiers, suggesting its effectiveness in accurately distinguishing faces in the FERET dataset. The combination of MPCA and HOG further reinforced this outstanding performance, also achieving 100% accuracy and F1-Score. While the MPCA-LDA combination showed some improvements, it couldn't match HOG-based methods. The Chi-Square feature extraction method had a limited impact on overall performance, offering only slight improvements in certain combinations. In conclusion, HOG-based methods emerged as the most effective choice for face recognition on the FERET dataset, showcasing their superior performance compared to other techniques like MPCA and LDA. *Table 3* describes the FERET dataset results.

Table 3: FERET Dataset Results

Feature Extraction Method	RM				Support Vector Machine			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
MPCA	48.14%	49.00%	48.00%	48.00%	44.78%	46.00%	45.00%	44.00%
LDA	73.73%	54.00%	74.00%	63.00%	71.38%	62.93%	71.38%	64.37%
HOG	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
MPCA-LDA	73.73%	72.30%	73.73%	72.89%	76.26%	72.92%	76.26%	68.48%
MPCA-HOG	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
MPCA-LDA-Chi-Square	75.08%	76.28%	75.08%	75.59%	72.72%	69.99%	72.72%	70.94%
MPCA-HOG-Chi-Square	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

The results of the experiments suggest that the combination of feature extraction methods and classifiers significantly impacts the effectiveness of face recognition algorithms. Among the tested methods, Histogram of Oriented Gradients (HOG) consistently demonstrates remarkable performance, achieving near-perfect accuracy and F1-Score on both the CK+ and FERET datasets when paired with either Random Forest or SVM. HOG's ability to capture local shape and texture information appears to make it an exceptionally effective choice for face recognition tasks.

While Linear Discriminant Analysis (LDA) also exhibits strong performance, particularly when combined with Random Forest, it falls slightly short of HOG in terms of accuracy and F1-Score. Multi-linear Principal Component Analysis (MPCA) performs decently but does not match the accuracy levels achieved by HOG or LDA.

Furthermore, the introduction of feature selection techniques, such as Chi-Square feature reduction, provides a noticeable improvement in some cases, particularly in the CK+ dataset. However, it is essential to strike a balance between dimensionality reduction and feature retention to prevent overfitting.

In summary, the experiments indicate that HOG, whether applied alone or in conjunction with other methods, stands out as the more effective method for face recognition in this context. Its ability to capture detailed texture and shape information proves invaluable in achieving high accuracy and F1-Scores. However, the choice of method may depend on the specific requirements of the application, as LDA also presents a strong alternative, particularly when paired with Random Forest. *Table 4* introduce Comparison of Feature Extraction Methods.

Table 4: Comparison of Feature Extraction Methods

Feature Extraction Method	CK+ Dataset (Accuracy)	FERET Dataset (Accuracy)	Computational Efficiency
MPCA	0.7898	0.4814	Moderate
LDA	0.8779	0.7373	Moderate
HOG	0.9966	1	High
MPCA-LDA	0.8879	0.7373	Moderate
MPCA-HOG	1	1	High
MPCA-LDA-Chi-Square	0.9015	0.7508	Moderate
MPCA-HOG-Chi-Square	1	1	High

4.3 Related Work Comparison

Table 5 illustrates a variety of approaches for facial expression recognition, employing techniques like CNNs, LSTM CNNs, and specific feature extraction methods like Dense SIFT and regular SIFT. The accuracy percentages vary significantly across different datasets, highlighting the importance of choosing appropriate methods based on the characteristics of the data. The results from the present study indicate promising performance for the MPCA-LDA-Chi-Square and MPCA-HOG-Chi-Square methods on the CK+ and FERET datasets.

Table 5: Related works comparison

References	Datasets	Accuracy (%)	Methods
Al-Shabi et al. [12]	CK+ FER-2013	99.1 73.4	Dense SIFT and regular SIFT are merged with CNN features
Jain et al. [13]	CK+ MMI	96.17 98.72	LSTM CNN using texture patterns for facial expression prediction
Lopes et al. [14]	CK+ JAFFE BU-3DFE	96.76% 98.92% 72.89%	CNN
Li et al. [15]	SMIC CASME CASME II	55.49% 54.44% 59.11%	3D CNN
Xie et al. [16]	CK+ JAFFE	With 10-fold cross-validation accuracy 95.88 99.32	Expressional Region Descriptor (SERD) and Multi-Path Variation-Suppressing Network (MPVS-Net)
Li et al. [17]	CK+ JAFFE	97.38% 97.18%	Convolutional neutral network (CNN)
Present study	CK+ FERET	90.15% -100% 75.08%-100%	MPCA-LDA-Chi-Square -RF MPCA-HOG-Chi-Square-RF

5. CONCLUSION

The face recognition experiment on the CK+ dataset showcased the effectiveness of different feature extraction methods and classifiers. Among the methods evaluated, HOG stood out as the top performer, achieving exceptional accuracy and F1-Score of 99.66% with both classifiers and a perfect 100% accuracy and an F1-Score when combined with SVM and Chi-Square. The combination of MPCA and HOG (MPCA-HOG) also demonstrated outstanding results, achieving flawless 100% accuracy and F1-Score with both classifiers. LDA also showed strong performance, with accuracy values of 87.79% and 89.84%, and an F1 scores of 88.00% and 90.00% when combined with RM and SVM, respectively. These findings suggest that HOG-based methods and the MPCA-HOG combination are highly effective for face recognition on the CK+ dataset. On the other hand, the face recognition experiment on the FERET dataset revealed interesting insights. The MPCA feature extraction method showed limited suitability, achieving only 48.14% accuracy and an F1-Score of 48.00%. LDA, although an improvement, still had limitations, obtaining accuracy values of 73.73% and 71.38% with RM and SVM, respectively. In contrast, HOG demonstrated exceptional performance, achieving perfect 100% accuracy and an F1-Score with both classifiers. The combination of MPCA and HOG (MPCA-HOG) further emphasized this outstanding performance, also achieving 100% accuracy and an F1-Score. Although the MPCA-LDA combination showed some improvements, it couldn't match the effectiveness of HOG-based methods. The Chi-Square feature extraction method had a limited impact on overall performance.

ACKNOWLEDGMENT
The researchers would like to extend their thanks to the University of Mosul, the Remote Sensing Centre, for providing them with support.

REFERENCES
[1] S. S. Dash, "Face Recognition and Face Detection Benefits and Challenges," no. July, 2023, doi: 10.31838/ecb/2023.12.si6.226.

[2] F. Zhao, J. Li, L. Zhang, Z. Li, and S.-G. Na, "Multi-view face recognition using deep neural networks," Future Generation Computer Systems, vol. 111, pp. 375–380, 2020, doi: https://doi.org/10.1016/j.future.2020.05.002.

[3] S. M. Hamandi, A. M. S. Rahma, and R. F. Hassan, "A New Hybrid Technique for Face Identification Based on Facial Parts Moments Descriptors," Engineering and Technology Journal, vol. 39, no. 1B, pp. 117–128, 2021, doi: 10.30684/etj.v39i1b.1903.

[4] D. Sharma and A. Selwal, A survey on face presentation attack detection mechanisms: hitherto and future perspectives, vol. 29, no. 3. Springer Berlin Heidelberg, 2023. doi: 10.1007/s00530-023-01070-5.

[5] Y. Fu and Y. Ma, "Graph embedding for pattern analysis," Graph Embedding for Pattern Analysis, no. February 2016, pp. 1–260, 2013, doi: 10.1007/978-1-4614-4457-2.

[6] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction," Journal of Applied Science and Technology Trends, vol. 1, no. 2, pp. 56–70, 2020, doi: 10.38094/jastt1224.

[7] M. M. Bouchene, "Bayesian Optimization of Histogram of Oriented Gradients (HOG) parameters for Facial Recognition," no. July, 2023, doi: 10.2139/ssrn.4506361.

[8] P. Carcagnì, M. Del Coco, M. Leo, and C. Distante, "Facial expression recognition and histograms of oriented gradients: a comprehensive study," Springerplus, vol. 4, no. 1, 2015, doi: 10.1186/s40064-015-1427-3.

[9] R. A. Zafra, L. A. Abdullah, R. Alaraj, R. Albezreh, T. Barhoum, and K. Al, "An experimental study in Real-time Facial Emotion Recognition on new 3RL dataset," Journal of Current Trends in Computer Science Research, vol. 2, no. 2, pp. 68–76, 2023, doi: 10.33140/jctcsr.02.02.03.

[10] T. Alamri, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and A. M. Mirza, "Category specific face recognition based on gender," 2013 International Conference on Information Science and Applications, ICISA 2013, no. June, 2013, doi: 10.1109/ICISA.2013.6579382.

[11] H. Witharana, D. Volya, and P. Mishra, "quAssert: Automatic Generation of Quantum Assertions," arXIV: 2303.01487v1, 2023.

[12] T. C. Mundher Al-Shabi, Wooi Ping Cheah, "Facial Expression Recognition Using a Hybrid ViT-CNN Aggregator," Lecture Notes in Business Information Processing, vol. 449 LNBIP, pp. 61–70, 2015, doi: 10.1007/978-3-031-06458-6_5.

[13] K. Zhang, Y. Huang, Y. Du, and L. Wang, "Facial Expression Recognition Based on Deep Evolutional Spatial-Temporal Networks," IEEE Transactions on Image Processing, vol. 26, no. 9, pp. 4193–4203, 2017, doi: 10.1109/TIP.2017.2689999.

[14] A. T. Lopes, E. de Aguiar, A. F. De Souza, and T. Oliveira-Santos, "Facial expression recognition with Convolutional Neural Networks: Coping with few data and the training sample order," Pattern Recognit, vol. 61, pp. 610–628, 2017, doi: 10.1016/j.patcog.2016.07.026.

[15] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion Aware Facial Expression Recognition Using CNN With Attention Mechanism," IEEE Transactions on Image Processing, vol. 28, no. 5, pp. 2439–2450, 2019, doi: 10.1109/TIP.2018.2886767.

[16] S. Xie, H. Hu, and Y. Wu, "Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition," Pattern Recognit, vol. 92, pp. 177–191, 2019, doi: 10.1016/j.patcog.2019.03.019.

[17] K. Li, Y. Jin, M. W. Akram, R. Han, and J. Chen, "Facial expression recognition with convolutional neural networks via a new face cropping and rotation strategy," Visual Computer, vol. 36, no. 2, pp. 391–404, 2020, doi: 10.1007/s00371-019-01627-4.

[18] "CK+ (Extended Cohn-Kanade dataset)." [Online]. Available: https://paperswithcode.com/dataset/ck

[19] "NIST." 2019. [Online]. Available: https://www.nist.gov/itl/products-and-services/color-feret-database

[20] A. A. M. AL-Shiha, "Biometric Face Recognition Using Multilinear Projection and," A Thesis Submitted to the Faculty of Science, Agriculture and Engineering in Partial Fulfilment of the Requirements for The Degree of Doctor of Philosophy School, no. July, 2013.

[21] J. Wang et al., "Multilinear principal component analysis for face recognition with fewer features," Neurocomputing, vol. 73, no. 10–12, pp. 1550–1555, 2010, doi: 10.1016/j.neucom.2009.08.022.

[22] Y. Aliyari Ghassabeh, F. Rudzicz, and H. A. Moghaddam, "Fast incremental LDA feature extraction," Pattern Recognit, vol.48, no.6, pp. 1999-2012, 2015,doi: 10.1016/j.patcog.2014.12.012.

[23] W. Li et al., "Kernel Reverse Neighborhood Discriminant Analysis," Electronics (Switzerland), vol. 12, no. 6, 2023, doi: 10.3390/electronics12061322.

[24] C. Q. Lai and S. S. Teoh, "An efficient method of HOG feature extraction using selective histogram bin and PCA Feature reduction," Advances in Electrical and Computer Engineering, vol. 16, no. 4, pp. 101–108, 2016, doi: 10.4316/AECE.2016.04016.

[25] I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," Journal of King Saud University - Computer and Information Sciences, vol. 29, no. 4, pp. 462–472, 2017, doi: 10.1016/j.jksuci.2015.12.004.

[26] S. Rosidin, Muljono, G. F. Shidik, A. Z. Fanani, F. Al Zami, and Purwanto, "Improvement with Chi Square Selection Feature using Supervised Machine Learning Approach on Covid-19 Data," in 2021 International Seminar on Application for Technology of Information and Communication (iSemantic), 2021, pp. 32–36. doi: 10.1109/iSemantic52711.2021.9573196.

[27] K. Li et al., "multi-label spacecraft electrical signal classification method based on DBN and random forest," PLoS One, vol. 12, no. 5, pp. 1–19, 2017, doi: 10.1371/journal.pone.0176614.

[28] M. N. Murty and R. Raghava, "Linear support vector machines," SpringerBriefs in Computer Science, vol. 0, no. 9783319410623, pp. 41–56, 2016, doi: 10.1007/978-3-319-41063-0_4.