AffectNet-Based Facial Expression Recognition using Machine Learning

Renad Alsaleh[1], Shahad Alshehab[2], Fatima Alawami[3] , Anwar Aldahan[4], Fatimah Alwarsh[5], Layan Alumair[6]

Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia

[1]reenadalsaleh@gmail.com, [2]shahadym0@gmail.com , [3]fatimaalaawami@gmail.com, [4]anwarasmyat@gmail.com, [5] Fatimah.Alwarsh@gmail.com, [6]Layan_umair@icloud.com

Abstract

Facial expression recognition (FER) plays an important role in making technology more intuitive and responsive, with uses in security, healthcare, and human-computer interaction. Despite years of research, existing systems still struggle with real-world challenges like lighting variations, occlusion, and computational demands. Earlier work has shown that deep learning models, especially Convolutional Neural Networks (CNNs), excel at extracting features, while traditional machine learning models like Support Vector Machines (SVMs) remain strong for classification. However, most studies rely on one approach or the other, which limits their effectiveness. In this project, we propose a hybrid model that combines CNNs for feature extraction with SVMs for classification, aiming to boost accuracy and adaptability. We chose SVM for its reliability in handling high-dimensional data and for improving generalization. Experiments using the AffectNet dataset show that our approach achieves high accuracy and real-time performance, suggesting that this hybrid strategy offers a promising solution for future FER systems.

Keywords: Facial Expression Recognition, CNN, SVM, AffectNet, Hybrid Model

1. Introduction

Facial Expression Recognition (FER) is an influential area within artificial intelligence that focuses on enabling machines to understand human emotions from visual facial cues, supporting applications in healthcare, security, education, and interactive technologies [13]. The capability to automatically recognize facial expressions is essential for building empathetic and user-aware systems, thereby enriching digital communication and enhancing outcomes in fields like mental health assessment and driver safety [14]. The growing reliance on technology in daily life highlights the importance of FER, particularly as it enables more natural interactions between humans and intelligent systems [15].

Early approaches to FER were based on traditional machine learning techniques such as Support Vector Machines (SVM) and Principal Component Analysis (PCA), which required hand-crafted features and performed well in controlled environments but often failed to generalize to complex real-world conditions [16]. The rise of deep learning, especially Convolutional Neural Networks (CNNs), has dramatically improved FER accuracy by allowing models to learn complex, hierarchical features directly from data [17]. However, despite their success, deep learning models can require significant computational resources and sometimes struggle with robustness when faced with uncontrolled variations in lighting, pose, and occlusion.

To address these persistent challenges, recent research has explored hybrid methods that combine the strengths of both approaches. Integrating a pre-trained CNN for powerful feature extraction with an SVM for reliable classification leverages the feature learning capability of deep networks and the strong decision boundaries of traditional classifiers [18]. SVMs are especially effective in high-dimensional spaces and offer better interpretability and generalization with limited samples. The AffectNet dataset, with its scale and diversity, is used in this study to ensure the model is trained and tested in realistic, unconstrained scenarios. Comprehensive preprocessing—including normalization, grayscale conversion, and data augmentation—is also applied to enhance robustness.

Preliminary results and recent literature suggest that this hybrid approach can outperform single-method systems in both accuracy and adaptability, making it a practical and scalable solution for modern emotion-aware technologies [18].

The remaining part of this work is organized as follows. Section 2 contains review of related literatures. Section 3 contains the proposed machine learning techniques (i.e., CNN and SVM classifiers). Section 4 contains empirical studies that include dataset description, experimental setup or methodology, and the adopted optimization strategy or parameter search strategy. Section 5 presents results and discussion while section 6 contains the conclusion and recommendation emanating from this work.

2. Review of Related Literatures

To begin with, Schroff et al. (2015) designed FaceNet with a single goal in mind; to build a face embedding system that can uniformly map face images into a compact Euclidean space, where facial recognition can be detected through distance measurement. Classification layers are eliminated entirely, leading to automatic face recognition, verification, and clustering. Schroff et al. developed FaceNet as a model that is trained and evaluated on datasets such as labeled Faces in the Wild (LFW), YouTube Faces (YTF), and some internal datasets from Google which contain millions of face images belonging to approximately 8 million people. FaceNet is built on the framework of deep convolutional neural networks (CNN) and is further optimized with a triplet loss function. This loss function ensures that portraits of the same person are grouped closely together while the distance between different people is maximized. With triplet loss, feature selection is handled implicitly, meaning that the model's primary focus is identifying distinguishing features for recognition. Cross-validation is done on LFW and YTF for validation of the model performance. FaceNet reached an astounding 99.63% accuracy on LFW and 95.12% on YTF, which is a much lower error rate than was previously achieved. There are, however, some caveats to this study which stem majorly from the availability of Google proprietary data, as this makes it difficult to reproduce the model's performance in other settings. Moreover, the algorithm can suffer from a performance drop in uncontrolled real-world scenarios where there is variation in pose, occlusion, and lighting conditions. In spite of these obstacles, FaceNet was an achievement in terms of deep learning algorithms for facial recognition, serving as an inspiration for future work regarding face embeddings [1].

Following this, in 2016, A study conducted by Wan, W., Yang, C., & Li, Y. Proposed a Hybrid Facial Expression Recognition (FER2013) Model to enhance real-time emotion classification and prediction. The model integrates a Deep Convolutional Neural Network (DCNN)with Haar Cascade classifiers, leveraging the strengths of both architectures to efficiently detect faces and recognize emotions. They used the FER-2013 dataset, which contains 35,887 grayscale images categorized into seven emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. The dataset’s low resolution (48x48 pixels) presented a challenge, as it limited the model's ability to capture detailed facial features. The study applied data augmentation techniques to increase training samples and enhance the model's generalization. Cross-validation was used for performance evaluation, ensuring robust and reliable results. Additionally, GPU computation was utilized to accelerate the training process. The proposed hybrid model achieved an accuracy of 70%, outperforming traditional models by approximately 6%. This improvement was attributed to the model's combined approach, where the Haar Cascade efficiently detected faces, and the DCNN accurately classified emotions. Despite these promising results, the study faced limitations related to the low resolution of the FER-2013 dataset, which affected the model’s accuracy due to insufficient facial detail representation. The dataset's imbalanced emotion categories also led to biased predictions, as some emotions were underrepresented. The researchers suggested that future work could involve using higher resolution images and more balanced datasets to improve model performance. They also recommended exploring more advanced deep learning architectures and optimizing the hybrid model for better real-time processing capabilities [2].

Subsequently, in 2018, Deng et al. introduced ArcFace to enhance face recognition accuracy by optimizing the angular margin-based learning technique. Unlike previous models, ArcFace introduces an Additive Angular Margin Loss (ArcFace loss), which strengthens the discrimination of face embeddings of different identities, improving face verification and classification stability. The study utilizes large-scale face datasets, including MS-Celeb-1M, CASIA-WebFace, Labeled Faces in the Wild (LFW), and the MegaFace Challenge dataset.ArcFace is implemented with a deep convolutional neural network (CNN) trained using the ArcFace loss function, which forces the model to learn more discriminative face embeddings. The feature selection process is enhanced with angular margin constraints, making it more discriminative in distinguishing visually similar faces. ArcFace is evaluated on LFW, MegaFace, and other large-scale verification benchmarks, where the model outperforms previous state-of-the-art methods consistently. As regards performance, ArcFace achieved 99.83% on LFW, higher than FaceNet and other state-of-the-art models. In addition to this, it also excelled greatly in the MegaFace Challenge, reflecting its efficiency for large-scale face recognition. However, the greatest weakness of ArcFace is to be assaulted by noisy labels in large-scale datasets, requiring the introduction of sub-center ArcFace to cope with variations. Despite this restriction, ArcFace is a landmark in the science of face recognition, optimizing angular margin-based classification techniques to produce state-of-the-art performance [3].

Moving forward to 2021, Omotosho et al. developed real-time face recognition using AlexNet which is a convolutional neural network (CNN)-based model that utilizes transfer learning. The ultimate goal of enhancing recognition accuracy and computational efficiency is achieved by using referenced models that were pre-trained on large-scale datasets. The dataset used in the study includes Labeled Faces in the Wild (LFW) with over 13,000 labeled images and CelebA with 200,000 celebrity images with 40 attribute labels. Deep facial feature extraction was performed via CNNs, but the number of extracted features remains unspecified. Facial recognition was performed using AlexNet with transfer learning. The recognition accuracy attained was 98.95%, with the mini-batch accuracy reaching 98.44% at the last epoch and a mini-batch loss of 0.0355, all of which clearly indicate the precision with which the approaches manage their facial recognition tasks. The high computational power requirements of the system limit its practical implementation on devices with resource constraints. Optimization techniques are thus an absolute necessity, as this article proposes, aiming to reduce overhead while ensuring a quality real-time application [4].

During the same year, Ullah et al.'s paper discusses the performance of different machine-learning algorithms. These algorithms are Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Artificial Neural Networks (ANN). It tries to find the best way to recognize facial expressions. According to the paper, appropriate features and good preparation of data are important to obtain high accuracy in classification. Two main data sets are used in the research: JAFFE (Japanese Female Facial Expression Dataset) and CK+ (Extended Cohn-Kanade Dataset). Datasets have a variety of facial expressions, both acted and real, hence making it convenient to try different machine learning models. The authors use 68 facial landmarks to feature extraction to improve classification accuracy by identifying key facial features. The highest accuracy is for CNN at 98.47% for the CK+ dataset. The best for ANN is on the JAFFE dataset with 89.18% accuracy. SVM, especially with the RBF kernel, is 93% accurate on CK+ but a bit lower on JAFFE. The reason why CNN is better is that it can learn important features by itself, so it is better at facial expression classification. One of the major strengths of this study is that it involves an intense preparation process, including cropping, scaling, and normalization to improve images. The sole weakness mentioned is that CNN changes with different datasets. It mentions that deep learning models require a large amount of fine-tuning or big datasets to work at their best. Additionally, the paper does not consider transfer learning methods, which can improve accuracy using pre-trained models. Future work can try to develop hybrid models that fuse feature engineering techniques with deep learning. It can also try to increase datasets to include more populations. Additionally, the deployment of real-time FER systems on edge computing devices can help make such models more feasible for real-world applications like surveillance, healthcare, and interactive AI [5].

In 2022, Wirdiani and others talked about the construction and design of a real-time face recognition system based on deep learning. It integrates different kinds of CNN architectures into the system, especially the Siamese Networks and YOLOv5. The research aimed to maximize recognition performance in real life when circumstances such as light conditions, occlusion of faces, and varying head poses come into play. In the research, the datasets employed here are the gaze-custom dataset, FER2013 (i.e., 35,887 grayscale images), and MS-Celeb-1M (more than 10 million images of 100,000 celebrities). The qualities of interest, that is, potential candidates for facial features, were pointed out using CNN-based models, but the particular number of features was not given. The study presented a recognition accuracy of 94% against an SVM model, thus outperforming it. The system was also able to detect faces under difficult conditions, including the presence of glasses, masks, and different rotations. Such success suggests a degree of robustness in handling various facial variances. On the other hand, the model is less able to deal with occlusions and variable lighting conditions, thus affecting recognition accuracy. The article has pointed out areas for further improvement concerning complicated environmental factors, particularly through better preprocessing and augmentation techniques [6].

Furthermore, Collins Oguine, O., Oguine, K. J., Bisallah, H. I., & Ofuani, D. Exploreed the application of Convolutional Neural Networks (CNNs) for facial expression recognition, aiming to improve emotion classification accuracy by leveraging advanced deep learning architectures. It utilizes the Kaggle Facial Expression Challenge dataset, which contains 35,887 grayscale images classified into seven emotional categories: anger, disgust, fear, happiness, sadness, surprise, and neutral. The images have a resolution of 48x48 pixels, presenting a challenge due to limited feature details, which impacts model performance. The research evaluates two prominent CNN architectures, AlexNet and VGGNet, using different fine-tuning strategies, including learning rate adjustments, dropout rates, and data augmentation techniques to enhance generalization. No specific feature selection technique was employed, as CNNs naturally learn hierarchical features from raw pixel data. The model's performance was evaluated using cross-validation, ensuring robustness and generalization. Multiple experiments were conducted to compare the impact of varying hyperparameters, network architectures, and training methods. The highest accuracy achieved was 65.3%, using VGGNet with fine-tuning and advanced data augmentation techniques. This result demonstrates the potential of deep CNN architectures in emotion recognition tasks, although the accuracy was limited by the dataset's constraints. A significant limitation was the low resolution of the dataset (48x48 pixels), which hindered the model's ability to capture detailed facial features, reducing classification accuracy. Additionally, the imbalance in emotional categories within the dataset introduced biases in model predictions. The study illustrates the effectiveness of VGGNet for facial expression recognition but emphasizes the need for higher resolution images and more balanced datasets to enhance performance. It also highlights the potential of further improvements through the integration of more sophisticated data augmentation strategies and the exploration of alternative network architectures [7].

Additionally, Ozdamli and colleagues conducted a study to design a facial recognition system to detect student emotions and cheating in distance learning environments. These were utilized in this research to perform face verification from the LFW data set of 13,500 images, emotion recognition from the FER2013 data set of 32,300 images, eye gaze from the GI4E data set of 1,380 images, head rotation from the FEI Face Database of 2,800 images, and object detection from the COCO data set of 5,000 images. The authors used convolutional neural networks like Mini Xception and ResNet-34 and computer vision models like OpenCV cascade classifiers and YOLO v3 for object detection. Deep metric learning was used in feature extraction and 70/30 train/test while the model was being tested. The system worked efficiently with 99.38% accuracy in face recognition, 66% accuracy in emotion recognition, 96.95% accuracy in eye gaze detection, and 96.24% accuracy in face movement detection. Variables to be restricted in the study, i.e., independent tracking of face and head movements, inconsistency to a specific emotion, and internet dependency. Optimization variables one can utilize while planning for the future are special needs students' assistance as an optimization variable and speech recognition as an optimization variable [8].

In the same year, Mazhar and others study was intended to verify the use of video analysis from facial recognition for sentiment analysis as a business development and decision-support tool. The researchers used the public AffectNet and FER datasets and AffectNet containing over one million facial expression images in many languages. The study employed different the machine learning models like Naive Bayes, Support Vector Machine (SVM), Random Forest, and Convolutional Neural Network (CNN). Feature extraction involved using HAAR features to identify facial expressions and emotions. A 10-fold cross-validation technique was also used for verification of the model. The suggested CNN model has yielded the optimum results with a correctness of 84.72%, sensitivity of 79.24%, specificity of 90.64%, and precision of 90.2%. Despite this, research was limited with noise and video quality problems being low, thus affecting the efficiency of the emotion recognition system adversely [9].

Also, in 2022, Gowda and Suresh suggest a new FER system using a Hybrid Adaptive Kernel-based Extreme Learning Machine with Chicken Swarm Optimization (HAKELM-CSO) to achieve efficiency and accuracy. This research is driven by the need for improved surveillance systems at secure areas like airports and border control points. The research resolves the complexity of facial gesture extraction because human faces are multi-dimensional. The research utilizes two popular datasets, the Indian Face Database and the JAFFE Dataset, with different facial expressions. It utilizes Wiener filtering for noise reduction and to make the images clearer and cleaner. Then, it uses the HAKELM-CSO model to recognize facial expressions, hoping to outperform traditional methods like Principal Component Analysis (PCA) and Support Vector Machines (SVM). The result shows that the suggested model attains 95.84% accuracy, which is better than PCA (94.45%) and SVM (83.5%). According to the research, the improvement is due to better kernel parameters through the Chicken Swarm Optimization (CSO) algorithm, leading to improved classification performance. The model further shows decent sensitivity (90.12%) and specificity (96.12%), which implies that it performs well in recognizing different facial expressions. Despite the promising outcomes it has achieved, the research has some limitations like the restricted dataset diversity that only covers the Indian and Japanese facial expressions. This limitation undermines the model's generalizability to diverse populations. Additionally, the study does not investigate deep learning approaches, which have been proven to outperform other methods' capabilities in FER tasks. Future research may involve working with larger and more diverse datasets and exploring hybrid models that combine conventional and deep learning approaches to enhance accuracy and applicability in real-world contexts [10].

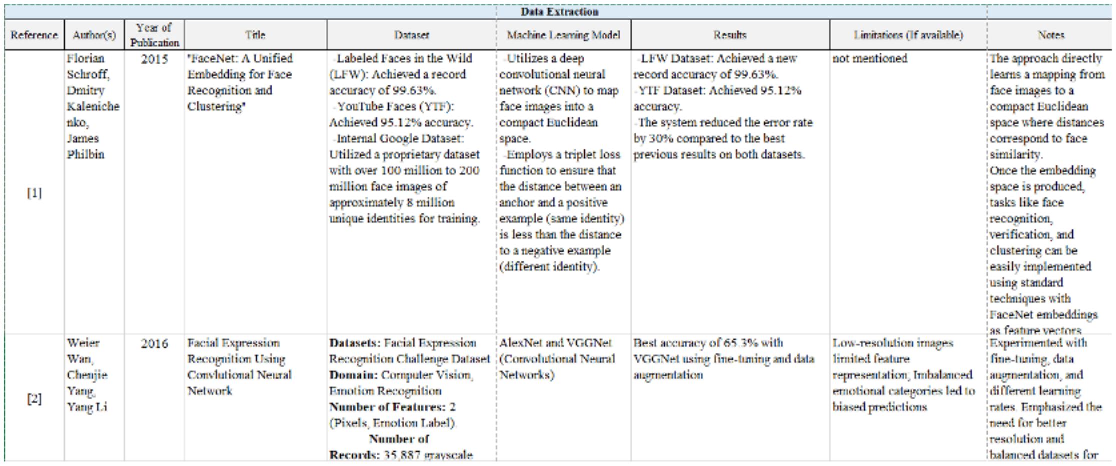
More recently, in 2023, Han et al. (2023) have put forward a Deep Emotion Change Detection (DECD) model to resolve the limitations of traditional Facial Expression Recognition (FER) systems that primarily detect static emotions from an individual image. The work focused on the detection of emotion transitions in over time, which is a decisive feature to apply in social communication, diagnosis in mental health, and HCI. In comparison with the conventionally used temporally annotated video databases, the proposed DECD utilizes weakly supervised learning based on static facial images where the model can sufficiently learn from them for generalization into video streams. The model is comprised of a Multi-Task Emotion Recognizer (MTER) as a feature extraction module and changepoint detector that identifies influential emotional state change.The architecture was tested with diverse datasets like CASME II, MMI, and YouTubeECD though it had been trained merely on AffectNet. The experimental results portrayed great cross-dataset generalizability, assuring robustness of the model in real environments. Additionally, the architecture was adapted to fit temporal spotting and achieved a level of performance same as state-of-the-art systems on CAS(ME)2, further reassuring its value. There are, however, certain challenges remaining. The model is poor at detecting spontaneous and subtle emotional changes, e.g., microexpressions, which require greater sensitivity to small facial differences. The addition of multimodal inputs, i.e., speech, physiological signals such as heart rate, EEG, and context, would make a significant difference to the model's ability to distinguish between similar emotional states and offer a deeper analysis of human emotions. Future studies must explore real-time uses of DECD for interactive AI systems, including affective computing, mental health monitoring, and adaptive HCI [11].

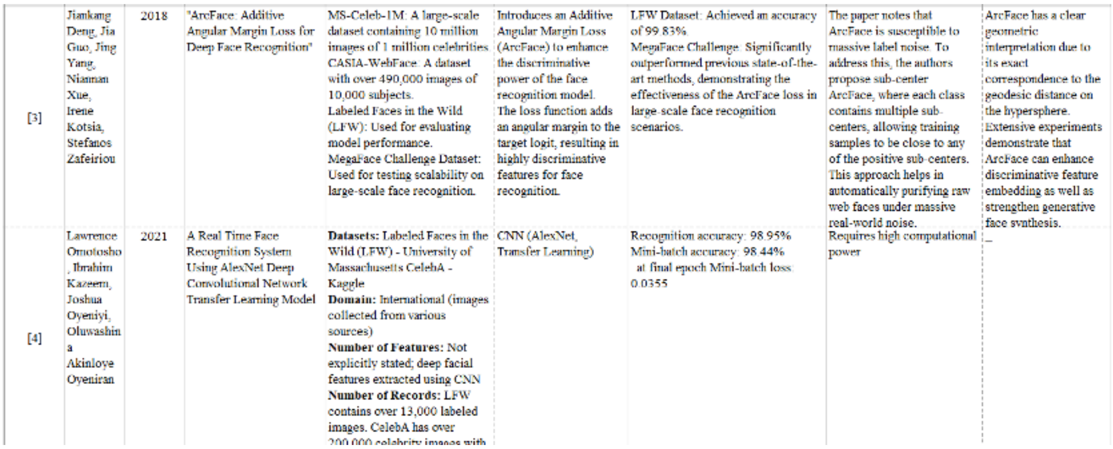
Finally, in 2025, Guan et al. (2025) designed an experiment to enhance driver facial expression recognition (FER) through a deep learning-based Interlaced Local Attention Block-Convolutional Neural Network (ILAB-CNN) scheme to address the problems in the real-world driving scenes. Researchers utilized public benchmarking datasets like FER-

2013, RAF-DB, and KMU-FED to demonstrate the superiority of the approach.The model utilizes an Interlaced Local Attention Block (ILAB) to find significant facial areas and a Modified Squeeze-and-Excitation (MSE) block for enhanced feature extraction using self-attention mechanisms. The two-branch structure superbly extracts local and global facial patterns in an effort to offer greater accuracy when recognizing emotions.Experimental performance was 75.3% on FER-2013, 85.06% on RAF-DB, and 98.8% on KMU-FED, affirming the effectiveness of the model. Additionally, its structurally optimized parameter reduces computational complexity, and thus it is feasible for real-time implementation. However, the work was confronted with facial occlusions (e.g., masks, sunglasses, and hands) and variability in lighting conditions in images, which compromised performance. Transfer learning and multimodal sensor data (e.g., infrared imagery) can be utilized in future to enhance model robustness in evolving real-world environments [12].

2.1 Gap Identification

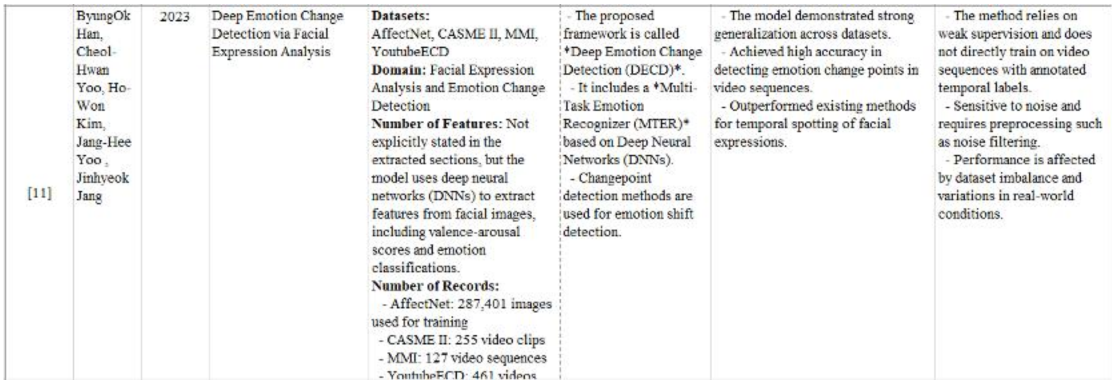
Despite advancements achieved by FER and face recognition, there exist several limitations that are pervasive in current research. FaceNet and ArcFace were proposed by Schroff et al. (2015) [1] and Deng et al. (2018) [3] with record-breaking performance but both are trained on commercial datasets which restrict reproducibility and generalizability to real-world lighting conditions, occlusion, and pose. Omotosho et al. (2021) [4] employed AlexNet and transfer learning and had high accuracy but at the cost of high model computational costs, which rendered it unsuitable for real-time operation on resource-limited devices. Wirdiani et al. (2022) [6] tried improving real-time face recognition but were plagued by occlusion and varying illumination, which impacted performance when there were uncontrolled environments. Likewise, Han et al. (2023) [11] proposed a Deep Emotion Change Detection (DECD) model, but the use of static image datasets does not allow for dynamic emotional change to be followed over time. More recently, Guan et al. (2025) [12] enhanced facial expression recognition through Interlaced Local Attention Blocks, but issues of dataset diversity and generalizability remain since datasets like JAFFE and RAF-DB are not demographically diverse. In addition, Wan et al. (2023) [7] also utilized FER-2013 but, with the low-resolution (48x48 pixels) data, the model was unable to discern more refined facial features, and classification accuracy suffered. In all the above work, very few of them employed hybrid architectures that integrate deep learning with traditional machine learning models, and they can assist in enhancing efficiency and flexibility. To solve such problems, research direction in the future must be in the direction of multimodal fusion methods, model optimization to enable it to process in real-time, data set augmentation to give a better representation, and open-source benchmarks encouraging for enhancing transparency and reproducibility of FER research.



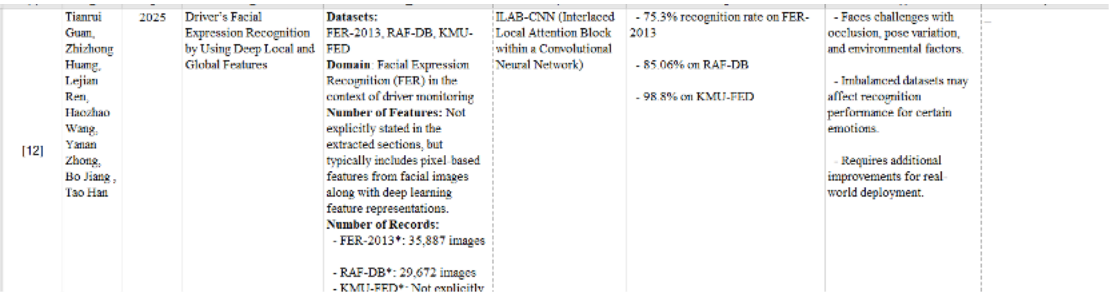


					Data Extraction			
Reference	Author(s)	Year of Publication	Title	Dataset	Machine Learning Model	Results	Limitations (If available)	Notes
[1]	Florian Schroff, Dmitry Kalenichenko, James Philbin	2015	"FaceNet: A Unified Embedding for Face Recognition and Clustering"	-Labeled Faces in the Wild (LFW): Achieved a record accuracy of 99.63%. -YouTube Faces (YTF): Achieved 95.12% accuracy. -Internal Google Dataset: Utilized a proprietary dataset with over 100 million to 200 million face images of approximately 8 million unique identities for training.	-Utilizes a deep convolutional neural network (CNN) to map face images into a compact Euclidean space. -Employs a triplet loss function to ensure that the distance between an anchor and a positive example (same identity) is less than the distance to a negative example (different identity).	-LFW Dataset: Achieved a new record accuracy of 99.63%. -YTF Dataset: Achieved 95.12% accuracy. -The system reduced the error rate by 30% compared to the best previous results on both datasets.	not mentioned	The approach directly learns a mapping from face images to a compact Euclidean space where distances correspond to face similarity. Once the embedding space is produced, tasks like face recognition, verification, and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors
[2]	Weier Wan, Chenjie Yang, Yang Li	2016	Facial Expression Recognition Using Convlutional Neural Network	**Datasets:** Facial Expression Recognition Challenge Dataset **Domain:** Computer Vision, Emotion Recognition **Number of Features:** 2 (Pixels, Emotion Label) **Number of Records:** 35,887 grayscale	AlexNet and VGGNet (Convolutional Neural Networks)	Best accuracy of 65.3% with VGGNet using fine-tuning and data augmentation	Low-resolution images limited feature representation, Imbalanced emotional categories led to biased predictions	Experimented with fine-tuning, data augmentation, and different learning rates. Emphasized the need for better resolution and balanced datasets for
[3]	Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, Stefanos Zafeiriou	2018	"ArcFace: Additive Angular Margin Loss for Deep Face Recognition"	MS-Celeb-1M: A large-scale dataset containing 10 million images of 1 million celebrities. CASIA-WebFace: A dataset with over 490,000 images of 10,000 subjects. Labeled Faces in the Wild (LFW): Used for evaluating model performance. MegaFace Challenge Dataset: Used for testing scalability on large-scale face recognition.	Introduces an Additive Angular Margin Loss (ArcFace) to enhance the discriminative power of the face recognition model. The loss function adds an angular margin to the target logit, resulting in highly discriminative features for face recognition.	LFW Dataset: Achieved an accuracy of 99.83%. MegaFace Challenge: Significantly outperformed previous state-of-the-art methods, demonstrating the effectiveness of the ArcFace loss in large-scale face recognition scenarios.	The paper notes that ArcFace is susceptible to massive label noise. To address this, the authors propose sub-center ArcFace, where each class contains multiple sub-centers, allowing training samples to be close to any of the positive sub-centers. This approach helps in automatically purifying raw web faces under massive real-world noise.	ArcFace has a clear geometric interpretation due to its exact correspondence to the geodesic distance on the hypersphere. Extensive experiments demonstrate that ArcFace can enhance discriminative feature embedding as well as strengthen generative face synthesis.
[4]	Lawrence Omotosho, Ibrahim Kazeem, Joshua Oyeniyi, Oluwashina Akinloye Oyeniran	2021	A Real Time Face Recognition System Using AlexNet Deep Convolutional Network Transfer Learning Model	**Datasets:** Labeled Faces in the Wild (LFW) - University of Massachusetts CelebA - Kaggle **Domain:** International (images collected from various sources) **Number of Features:** Not explicitly stated; deep facial features extracted using CNN **Number of Records:** LFW contains over 13,000 labeled images. CelebA has over 200,000 celebrity images with	CNN (AlexNet, Transfer Learning)	Recognition accuracy: 98.95% Mini-batch accuracy: 98.44% at final epoch Mini-batch loss: 0.0355	Requires high computational power	_

[5]	Salam Ullah, Atif Jan, Dr. Gul Muhammad Khan	2021	Facial Expression Recognition Using Machine Learning Techniques	**Dataset**: 213 images of 10 Japanese female models - CK+ Dataset: Posed and non-posed expressions from 123 individuals **Domain**: Japanese (JAFFE) and mixed (CK+) expressions **Number of Features**: 68 facial landmarks **Number of Records**: JAFFE (213 images, 10 Japanese female models), CK+ (posed	Support Vector Machine (SVM) (Poly & RBF Kernels), Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN)	CNN achieved 98.47% accuracy on CK+ dataset ANN achieved 89.18% accuracy on JAFFE dataset SVM (RBF) achieved 93% accuracy on CK+	Performance varies across different datasets, CNN works better on CK+ while ANN works better on JAFFE	Feature extraction using 68 facial landmarks Preprocessing included scaling, cropping, and normalization CNN performed better on CK+, ANN performed better on JAFFE
[6]	Ayu Wirdiani, I Ketut Gede Darma Putra, Made Sudarma, Rukmi Sari Hartati, Lennia Savitri Azzahra Lofiana	2022	Real-time Face Recognition System Using Deep Learning Method	**Datasets:** Custom dataset FER2013 - Kaggle MS-Celeb-1M **Domain:** International (datasets collected from various sources, including public datasets) **Number of Features:** Not explicitly stated; features extracted using CNN-based models **Number of Records:** FER2013 contains 35,887 grayscale images. MS-Celeb-1M has over 10 million images	CNN (Siamese Network, YOLOv5)	Recognition accuracy: 94% Comparison with SVM shows improved performance Successfully detects faces in multiple conditions (glasses, masks, rotations)G7:I7	Struggles with occlusions and varying lighting conditions	_
[7]	Ozioma Collins Oguine, Kanyifeechukwu Jane Oguine, Hashim Ibrahim Bisallah, Daniel Ofuani	2022	Hybrid Facial Expression Recognition (FER2013) Model for Real-Time Emotion Classification and Prediction	**Datasets:** FER-2013 dataset. **Domain:** Facial Expression Recognition / Emotion Classification **Number of Features:** Pixel values from 48x48 grayscale images **Number of Records:** 35,887 grayscale images	Hybrid model combining Deep Convolutional Neural Networks (DCNNs) with Haar Cascade classifiers	Achieved 70% accuracy, outperforming traditional models by 6%	Low-resolution images (48x48 pixels) and imbalanced emotional categories affecting accuracy and introducing prediction bias	_
[8]	Fezile Ozdamli, Aayat Aljarrah, Damla Karagozlu, and Mustafa Ababneh	2022	Facial Recognition System to Detect Student Emotions and Cheating in Distance Learning	**Datasets:** *LFW (Labeled Faces in the Wild) - University of Massachusetts Amherst *FER2013 (Facial Expression Recognition 2013) - Kaggle *GI4E (Gaze Interaction for Everybody) - University of Navarra *FEI Face Database - FEI University Center *COCO (Common Objects in Context) - COCO Consortium: Dataset for object detection **Domain:** Cyprus **Number of Features:** 128 extracted features for facial recognition **Number of Records:** *LFW: 13,500 *FER2013: 32,300 *GI4E: 1,380 *FEI: 2,800 *COCO: 5,000	CNN (Mini Xception, ResNet-34), OpenCV cascade classifiers, YOLO v3	*Face recognition: 99.38% *Emotion detection: 66% *Gaze tracking: 96.95% *Facial movement tracking: 96.24%	*Difficulty with automatic detection of head/face movements *Variability in emotional expressions for similar emotions *Dependence on stable internet connection	The authors suggested integrating speech recognition tools to improve the detection of cheating.
[9]	Mazhar, T.; Malik, M.A.; Nadeem, M.A.; Mohsan, S.A.H.; Haq, I.; Karim, F.K.; Mostafa, S.M.	2022	Movie Reviews Classification through Facial Image Recognition and Emotion Detection Using Machine Learning Methods	**Datasets:** *AffectNet - Kaggle *FER - Kaggle **Domain:** International (images collected globally in multiple languages) **Number of Features:** Not explicitly stated; however, facial features were extracted using HAAR features. **Number of Records:** *AffectNet has over 1 million labeled images. *FER contains 35,898 images.	Naive Bayes, SVM, Random Forest	*84.72% accuracy *79.24% sensitivity *90.64% specificity *90.2% precision.	_	_
[10]	Shashank M Gowda, H. N. Suresh	2022	Facial Expression Recognition using Robust Algorithm based on Modern Machine Learning Technique	**Dataset**: - Indian Face Database: 6 expressions - JAFFE Dataset: 213 images of 10 Japanese female **Domain**: Indian and Japanese facial expressions **Number of Features**: Not explicitly mentioned **Number of Records**: Indian dataset (6 expressions), JAFFE (213 images, 10 Japanese female models)	Hybrid Adaptive Kernel-based Extreme Learning Machine with Chicken Swarm Optimization (HAKELM-CSO), PCA, SVM	HAKELM achieved 95.84% accuracy, 90.12% sensitivity, and 96.12% specificity	Limited dataset variety, improvements possible with deep learning models	Used Wiener filter for preprocessing Optimized kernel parameters with Chicken Swarm Optimization (CSO) Best results for classification of facial expressions



[11]	ByungOk Han, Cheol-Hwan Yoo, Ho-Won Kim, Jang-Hee Yoo , Jinhyeok Jang	2023	Deep Emotion Change Detection via Facial Expression Analysis	**Datasets:** AffectNet, CASME II, MMI, YoutubeECD **Domain:** Facial Expression Analysis and Emotion Change Detection **Number of Features:** Not explicitly stated in the extracted sections, but the model uses deep neural networks (DNNs) to extract features from facial images, including valence-arousal scores and emotion classifications. **Number of Records:** - AffectNet: 287,401 images used for training - CASME II: 255 video clips - MMI: 127 video sequences - YoutubeECD: 461 videos	- The proposed framework is called *Deep Emotion Change Detection (DECD)*. - It includes a *Multi-Task Emotion Recognizer (MTER)* based on Deep Neural Networks (DNNs). - Changepoint detection methods are used for emotion shift detection.	- The model demonstrated strong generalization across datasets. - Achieved high accuracy in detecting emotion change points in video sequences. - Outperformed existing methods for temporal spotting of facial expressions.	- The method relies on weak supervision and does not directly train on video sequences with annotated temporal labels. - Sensitive to noise and requires preprocessing such as noise filtering. - Performance is affected by dataset imbalance and variations in real-world conditions.



[12]	Tianrui Guan, Zhizhong Huang, Lejian Ren, Haozhao Wang, Yanan Zhong, Bo Jiang , Tao Han	2025	Driver's Facial Expression Recognition by Using Deep Local and Global Features	**Datasets:** FER-2013, RAF-DB, KMU-FED **Domain**: Facial Expression Recognition (FER) in the context of driver monitoring **Number of Features:** Not explicitly stated in the extracted sections, but typically includes pixel-based features from facial images along with deep learning feature representations. **Number of Records:** - FER-2013*: 35,887 images - RAF-DB*: 29,672 images - KMU-FED*: Not explicitly	ILAB-CNN (Interlaced Local Attention Block within a Convolutional Neural Network)	- 75.3% recognition rate on FER-2013 - 85.06% on RAF-DB - 98.8% on KMU-FED	- Faces challenges with occlusion, pose variation, and environmental factors. - Imbalanced datasets may affect recognition performance for certain emotions. - Requires additional improvements for real-world deployment.	–

4.0 Description of the Proposed Techniques

The following section will give a detailed explanation of each algorithm we are going to use in our project. We will apply a set of machine learning algorithms to improve facial expression recognition, using Support Vector Machines (SVMs) for classification and the Eigenfaces, which is based on Principal Component Analysis (PCA), for feature extraction and dimensionality reduction. And the Convolutional Neural Network algorithm CNN.

System Architecture

Our system follows a pipeline with structure:

1. Data Collection Module: Captures and stores facial images.
2. Preprocessing Module: Does image resizing, grayscale, and normalization.
3. Feature Extraction Module: Does eigenfaces computation using PCA [3].
4. Face Recognition Module: Projects new faces into the eigenface space and classifies them [2].
5. User Interface: Displays recognition results and accepts user input.

4.1 CNN

CNN, or Convolutional Neural Network algorithm, is one of the advanced learning models built for the recognition, detection, and classification of images. CNNs differ

from traditional feed-forward neural networks in that they automatically learn spatial hierarchies of features by performing convolution operations.
Due to their rapidness in processing image data, CNNs have been widely applied in fields such as Facial Expression Recognition (FER), medical imaging, self-driving cars, and even anomaly detection. The extracted features from CNN algorithm are then classified using Support Vector Machines (SVMs) the second algorithm [13].

4.1.1 CNN Architecture

A CNN is composed of multiple layers, each serving a specific role in feature extraction and classification:

- Convolutional Layer:

The convolutional layer is the core building block of CNNs. It applies filters (kernels) to the input image to extract features such as edges, textures, and patterns.

Mathematical representation:

$$Y(i,j)=\sum_m\sum_n X(i-m,j-n)\cdot k(m,n)+b$$

Where:

- Y(i,j) is the output feature map at position (i,j).
- X(i−m,j−n) is the input image pixel at position (i−m,j−n).
- K(m,n) is the convolution kernel (filter).
- b is the bias term.

Each filter slides (convolves) over the image, performing element-wise multiplication and summing up the results. The graph of the Convolution layer is a visual representation of how a filter moves over the input image to generate a feature map.

- Activation Layer (ReLU):

After convolution, the **ReLU activation function** introduces non-linearity into the model, allowing it to learn complex patterns.

Mathematical Representation:
f(x)= max(0,x).

This function **replaces all negative values with zero**, making the network more efficient in feature learning.
Graph: ReLU Function

The graph of activation layer is a **plot showing the effect of ReLU**, where negative values are suppressed, and positive values remain unchanged.

- Pooling Layer (Downsampling)

The pooling layer reduces the spatial dimensions of feature maps, thereby decreasing computational complexity and preventing overfitting.
Mathematical Representation (Max Pooling):

$P(i,j)=\max\{Y(i',j')\}$

Where:

- $P(i,j)$ is the pooled value.
- $Y(i',j')$ represents the values in the pooling window.

The graph of it is a visualization of max pooling, showing how it selects the maximum value from each region of the feature map.

4.1.2 Fully Connected (FC) Layer

The fully connected layer takes the flattened feature maps and feeds them into a traditional neural network for classification. Each neuron in the FC layer is connected to every neuron in the previous layer, enabling the network to make final predictions.
Graph: Fully Connected Layer A diagram showing the transformation of feature maps into class scores, often using Softmax for multi-class classification.

4.1.3 Backpropagation in CNNs

By using backpropagation with gradient descent, CNNs are able to adjust the weightage so as to minimize the weights for greater precision inside the model.

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

Mathematical Formulation:

1. **Loss Function (Cross-Entropy Loss for Classification):**

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

Where:

- y_i is the actual label,
- $\hat{y}_i$ is the predicted probability.

2. **Gradient Descent Update Rule:**

$$w = w - \eta \frac{\partial L}{\partial w}$$

This equation is to update the weights in the learning process through the learning rate and the gradient of some loss function.

Where:

- ω is the weight parameter,
- η is the learning rate,
- $\frac{\partial L}{\partial \omega}$ is the gradient of the loss function.

4.1.4 CNN Evaluation Metrics

To measure the performance of your CNN model, use:

- **Accuracy:** $\frac{correct\ predictions}{total\ predictions} \times 100\%$
- **Precision & Recall:** Measures the model's ability to correctly classify expressions.
- **F1-Score:** Harmonic mean of precision and recall.

CNNs, a deep learning architecture, are central to automated facial expression recognition and have added to classification accuracy through hierarchical feature learning.

4.2 SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm commonly used for classification problems. It aims to find the optimal hyperplane that separates data points belonging to different classes with the maximum margin. The closest data points to the hyperplane are known as *support vectors*, and they are crucial in defining the decision boundary. In the case of non-linearly separable data, SVM uses *kernel functions* to transform the input space into a higher-dimensional feature space where a linear separation becomes possible. This makes SVM highly effective for complex classification tasks such as facial emotion recognition[4].

In a facial emotion recognition system, SVM serves as the core classifier. Facial images are first captured and processed — including steps like face detection, grayscale conversion, resizing, and normalization. Feature extraction methods such as facial landmarks, Histogram of Oriented Gradients (HOG), or embeddings from deep learning models (CNN) are applied to convert facial expressions into numerical feature vectors. These vectors are then used as input for the SVM model. During the training phase, SVM analyzes labeled examples of facial emotions (e.g., happy, sad, angry, surprised) and learns to differentiate between them by computing the optimal hyperplanes that separate these classes in feature space. During inference, the trained SVM model classifies the emotion displayed in new facial images by evaluating which side of the decision boundary the extracted feature vector falls on [5].

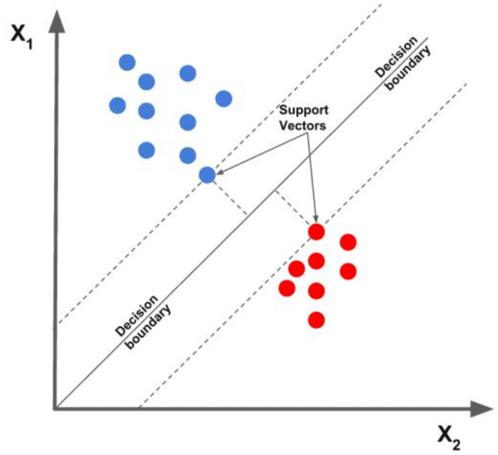


Figure: SVM classifier showing decision boundary, margins, and support vectors separating two classes

Figure: SVM classifier separating two classes

4.2.1 Key Equations in Support Vector Machine (SVM):

Hyperplane Equation:

$$w \cdot x + b = 0$$

This equation represents the decision boundary (hyperplane) that separates the different classes in a Support Vector Machine (SVM). The weight vector “w” defines the orientation of the hyperplane, while the bias term” b” shifts it from the origin. The goal of SVM is to find this optimal hyperplane that divides the data points from different classes as cleanly as possible.

Margin:

$$\text{Margin} = \frac{2}{||w||}$$

The margin is the distance between the hyperplane and the closest data points (called support vectors). Maximizing this margin helps ensure better generalization and lower risk of misclassification. The larger the margin, the better the SVM can classify new, unseen data. The equation shows that the margin depends on the norm (length) of the weight vector “w”.

Optimization Objective:

$$\min_{w,b} \frac{1}{2} ||w||^2$$

The SVM optimization problem aims to minimize the weight vector's norm squared (‖w‖^2) to maximize the margin. By minimizing this term, SVM ensures that the decision boundary is positioned in such a way that the margin is as wide as possible. The factor of 1\2 simplifies the derivative of the objective during optimization.

Constraint for Data Points:

$$y_i(w \cdot x_i + b) \geq 1$$

This constraint ensures that each data point (x_i) is correctly classified by the hyperplane. Here, y_i is the label for each data point, and it can either be +1 or -1 depending on the class. For the data points to be classified correctly, the product y_i (w · x_i + b) must be greater than or equal to 1. This ensures that each data point lies on the correct side of the margin, maintaining the classification accuracy.

4.3 EigenFaces

Eigenface is a PCA-based face recognition technique that projects high-dimensional face data into a lower-dimensional subspace without losing the most significant features [2]. Eigenface identifies patterns in facial imagery by extracting eigenvectors (or eigenfaces) from a training image dataset. The eigenfaces are the most noticeable differences between various faces. When a new face is presented that is unfamiliar, it is projected onto this lower-dimensional representation and compared with stored representations to find the best match. Eigenface performs well when trying to recognize faces in the lab and is widely applied since it's quick and easy [2], [3].

Our project will use the Eigenface approach to recognize faces by transforming face images into a set of principal components and classifying them based on similarity. The procedure is as follows:

4.3.1 Preprocessing

- Convert all images to grayscale (no color information is needed).
- Scale images to a constant size (e.g., 100×100 pixels) for consistency.
- Normalize the pixel intensities for consistency between images.

4.3.2 Eigenface Computation

To obtain the eigenface representation, we do the following computations:

Step 1: Images as Column Vectors
A face image is reshaped as a column vector by putting its pixel intensities on top of each other. An N × M pixel image is then an (N × M) × 1 vector.

For example, in the case of 100×100 pictures, every face becomes a 10,000-dimensional vector [2].

Step 2: Computing the Mean Face

Given a training set of M face images represented as vectors Ti, the mean face is calculated as:

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$$

Each image is then adjusted by subtracting the mean face:

$$\Phi_i = \Gamma_i - \Psi$$

This centers the dataset around the origin.

Step 3: Computing the Covariance Matrix

We construct the covariance matrix C:

$$C = AA^T$$

where A is the matrix formed by placing all Φ vectors as columns:

$$A = [\Phi_1, \Phi_2, ..., \Phi_M]$$

The size of C is (N × M) × (N × M), which is computationally expensive for large images [2].

Step 4: Applying PCA

Instead of directly computing eigenvectors of C, we use a computational shortcut:

$$L = A^T A$$

where L is an M × M matrix. We compute the eigenvectors v of L:

$$Lv = \lambda v$$

From these, the eigenfaces are obtained as:

$$u_i = Av_i$$

These eigenfaces form a lower-dimensional basis for face representation [3].

4.3.3. Feature Representation & Recognition

Each new face image is projected onto the eigenface space:

$$\Omega_k = u_k^T(\Gamma - \Psi)$$

where Ωk are the weights of the eigenfaces.

For recognition, we compare the new face’s weights with the stored weights in the dataset using Euclidean distance:

$$d = \sum_{k=1}^{K} (\Omega_k - \Omega'_k)^2$$

The face is classified based on the closest match [2].

3.0 Project Deliverables of the team

Deliverable	To whom	Delivery Media	Duration	Date
Literature Review (Homework-1)	Dr. Rabab AlKhalifa	Softcopy	1 week	Feb 23, 2025
Project Proposal	Dr. Rabab AlKhalifa	Softcopy	4 days	Mar 2, 2025
Project Proposal Presentation	Dr. Rabab AlKhalifa	Softcopy	2 days	Mar 5,2025
Description of Selected ML Algorithms	Dr. Rabab AlKhalifa	Softcopy	1 week	Mar 16, 2025
Final Project Report	Dr. Rabab AlKhalifa	Softcopy	2 weeks	May 4, 2025
Final Project Presentation	Dr. Rabab AlKhalifa	Softcopy	3 days	May 11, 2025

5.0 Empirical Studies

5.1 Description of dataset

The dataset used in this study is a curated subset of **AffectNet**, a large-scale facial expression database consisting of images collected from the internet using emotion-related keywords. It includes a wide variety of facial images in terms of ethnicity, pose, age, lighting, and occlusion.

A total of **28,535 labeled images** were used in this project, each annotated with one of the following **eight emotion classes**: *anger*, *contempt*, *disgust*, *fear*, *happy*, *neutral*, *sad*, and *surprise*.

To prepare the dataset for model training and evaluation, a consistent preprocessing pipeline was applied across all models. This included **converting each image to grayscale**, **resizing it to 128×128 pixels**, and **normalizing pixel values to the [0, 1] range**. These steps ensured uniformity and improved training stability by standardizing input dimensions and pixel intensity values.

5.1.1 Statistical Analysis of the Dataset

A statistical analysis of the dataset was conducted to examine the class distribution and identify potential imbalance issues that could influence model performance. The table below presents the number of images in each emotion category.

Emotion Class Distribution	
Emotion	**Number of Images**
Surprise	**4,616**
Happy	**4,336**
Anger	**3,608**
Disgust	**3,472**
Contempt	**3,244**
Fear	**3,043**
Sad	**2,995**
Neutral	**2,861**

Table 1 Emotion Class Distribution

To visualize the distribution, the dataset was plotted using bar and pie charts.

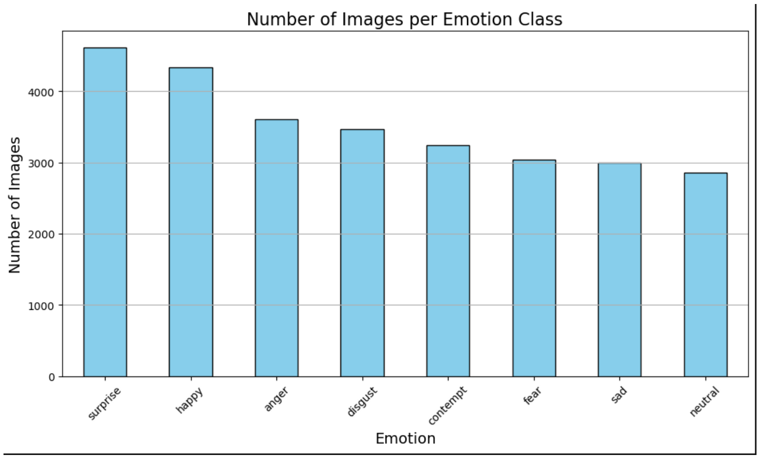


Figure 2 Distribution of images across the eight emotion categories in the dataset.

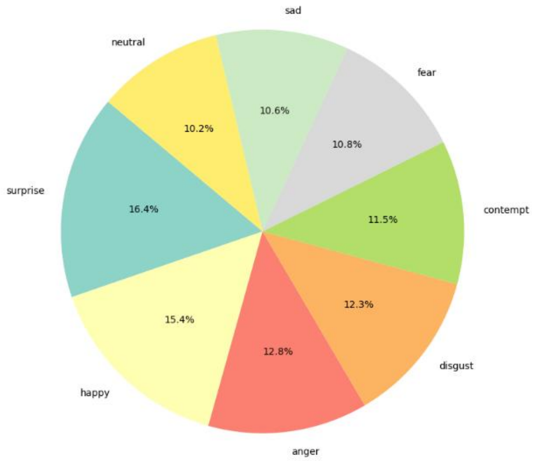


Figure 1 Pie chart representing the proportional distribution of each emotion class.

Descriptive statistics for the dataset are provided below:

Statistical Summary	
Metric	**Value**
Mean	3,521.88
Median	3,358.00
Standard Deviation	601.11
Maximum	4,616
Minimum	2,861

Table 2 Statistical Summary

The results indicate a moderately imbalanced dataset, with *surprise* and *happy* being the most frequent emotions and *neutral* the least. This imbalance was considered during the evaluation stage to ensure fair comparison across classes.

Note: Correlation analysis between attributes and the target variable was not applicable in this study, as the features consist of high-dimensional pixel data from images. Instead, relevant features were automatically extracted using CNN or PCA.

5.2 Experimental Setup

The experiments in this study were conducted using Python and libraries such as TensorFlow, Keras, OpenCV, and scikit-learn. Three models were evaluated independently: a custom Convolutional Neural Network (CNN), a Support Vector Machine (SVM), and a hybrid PCA-based model combining Eigenfaces with a CNN classifier.

The experimental procedure followed a clear pipeline:

1. **Preprocessing Phase**

- Images were loaded from emotion-labeled folders.
- Grayscale conversion and resizing to 128×128 pixels were applied.
- Pixel values were normalized to the [0–1] range.
- Labels were encoded into numeric format.
- In the hybrid model, PCA was applied to reduce the dimensionality of flattened image vectors.

2. **Model Training**

- CNN Model:

  A multi-layer CNN was implemented with convolutional, pooling, dropout, and dense layers. The Adam optimizer and categorical cross-entropy loss were used. EarlyStopping and learning rate reduction techniques were applied to prevent overfitting.

- SVM Model:

  Images were flattened and standardized before being passed into a linear SVM classifier using a one-vs-rest approach.

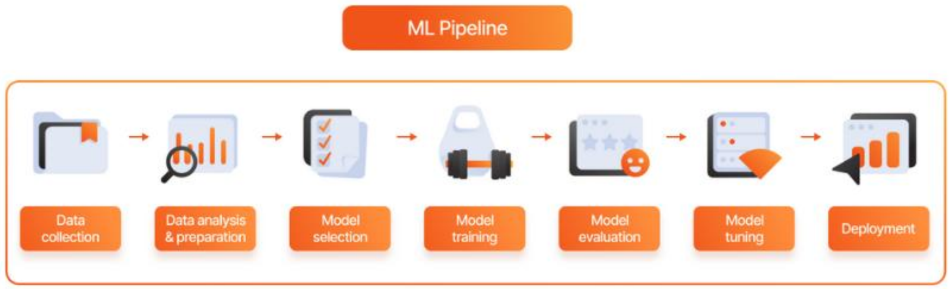
- Eigenfaces + CNN:

  PCA reduced the dimensionality to 100 components. The compressed features were reshaped and passed to a compact CNN for classification.

3. Evaluation and Interpretation

- A consistent train-test split was applied with stratified sampling to preserve class balance.
- Models were assessed using confusion matrices, classification reports, and visual inspection of training curves.
- No cross-validation was used due to the computational cost, but the dataset size ensured reliable performance measurement.

The entire experimental flow is illustrated below:



5.3

Figure 3 Workflow of the experimental setup showing preprocessing, model training, and evaluation phases.

Performance Measures

We assessed the SVM, CNN+PCA, and CNN (MobileNetV2) models using classification accuracy, precision, recall, and F1-score to account for class imbalance in AffectNet, where expressions such as *disgust* or *contempt* are under-represented.

Classification accuracy can be deceptive when certain classes predominate, even when it provides a broad indication of generally accurate predictions. Recall gauges the model's capacity to prevent false negatives, whereas precision measures its capacity to prevent false positives for each expression. Precision and recall are combined into a single per-class metric called the F1-score. These criteria, which are frequently employed in facial expression detection research, offer a more thorough picture of model performance than accuracy alone.

Model Test Accuracy

Model	Test Accuracy
SVM	36.51%
CNN + PCA (Eigenfaces)	44.76%
CNN (MobileNetV2 + SE)	**67.00%**

5.4 Optimization Strategy

SVM Hyperparameter Search

We performed a grid search with 5-fold cross-validation to identify the optimal hyperparameters for the SVM model. We searched over the following parameter ranges:

- C (penalty): {0.1, 1, 10}
- ε (insensitive-loss width): {0.01, 0.1, 0.2}
- Kernel: {Linear, RBF, Polynomial}
- Gamma (for RBF/Poly): {“scale”, “auto”}

After testing various combinations, we found that:

- C = 1.0
- Kernel = Linear
- ε = 0.1

produced the best validation accuracy.

Final optimized SVM hyperparameters:

Parameter	Optimal Value
C	1.0
Max Iterations	5000
Kernel	Linear

CNN + PCA (Eigenfaces) Hyperparameter Search

This model used grayscale input images reduced to 100 principal components using PCA. The PCA outputs were reshaped and fed into a custom CNN with three convolutional layers, batch normalization, max pooling, and dropout.

Final optimized CNN+PCA hyperparameters:

Parameter	Optimal Value
Optimizer	Adam
Learning Rate	0.0001
Dropout Rate	0.5

Batch Size	64
Epochs (Max)	50
Early Stopping	Enabled
Loss Function	Categorical Crossentropy
PCA Components	100

CNN (MobileNetV2 + SE Block) Hyperparameter Search

This model employed MobileNetV2 as a frozen feature extractor during Phase 1, followed by fine-tuning of the top 60 layers in Phase 2. A squeeze-and-excitation (SE) block was used to recalibrate channel-wise features.

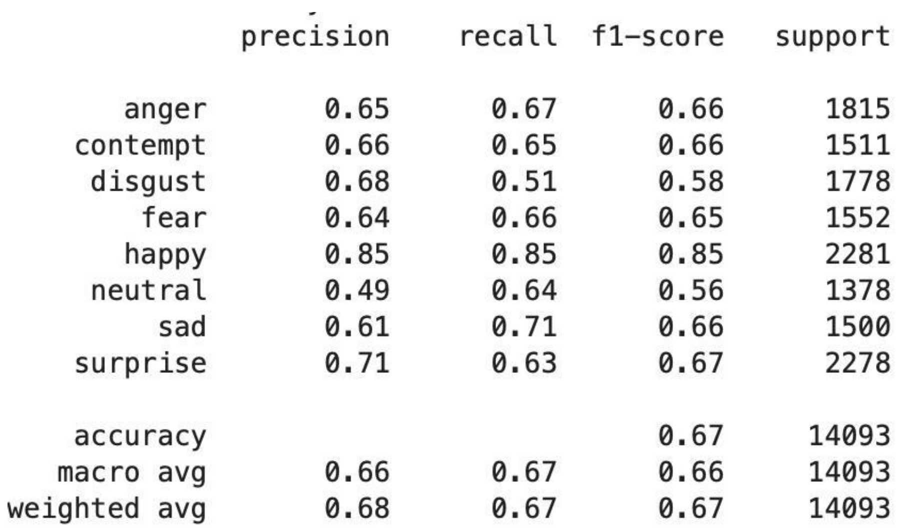
Final optimized CNN hyperparameters:

Parameter	Optimal Value
Optimizer	Adam
Learning Rate	0.0001 (during fine-tuning)
Dropout Rate	0.5
Batch Size	32
Epochs (Max)	35 (10 + 25)
Early Stopping	Enabled
Loss Function	Categorical Crossentropy with label smoothing
Data Augmentation	Enabled (contrast, flip, zoom, rotation, shear)

6.0 Result and discussion

Presented below are the results and discussions of various experimental options…

CNN:

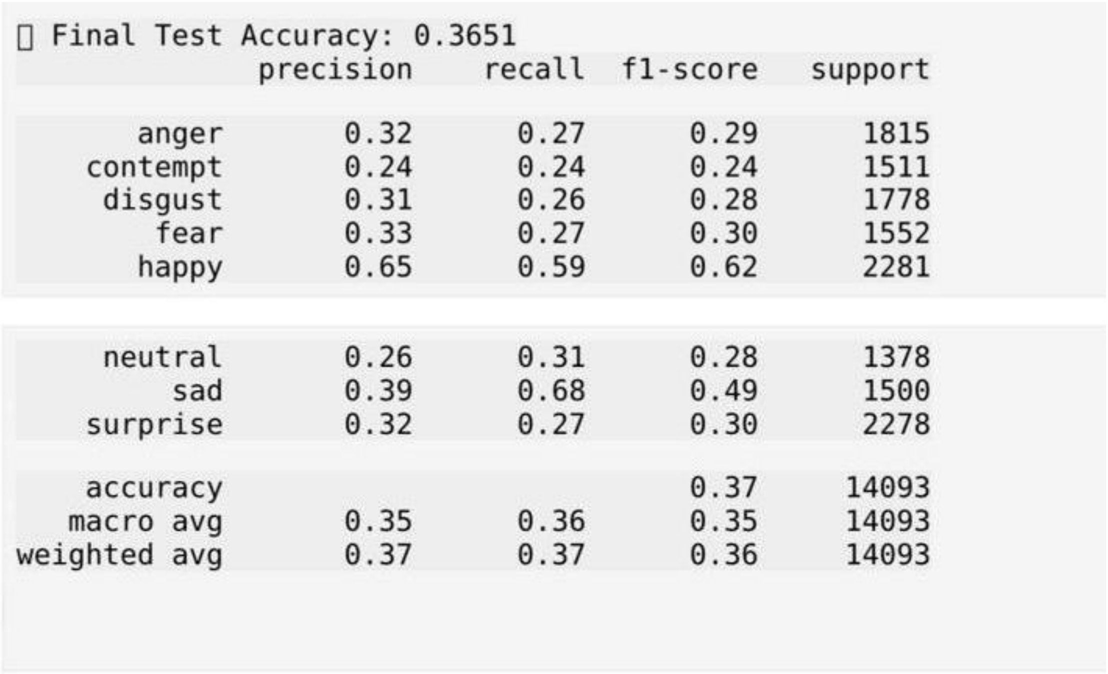


```
              precision    recall  f1-score   support

       anger       0.65      0.67      0.66      1815
    contempt       0.66      0.65      0.66      1511
     disgust       0.68      0.51      0.58      1778
        fear       0.64      0.66      0.65      1552
       happy       0.85      0.85      0.85      2281
     neutral       0.49      0.64      0.56      1378
         sad       0.61      0.71      0.66      1500
    surprise       0.71      0.63      0.67      2278

    accuracy                           0.67     14093
   macro avg       0.66      0.67      0.66     14093
weighted avg       0.68      0.67      0.67     14093
```

SVM:



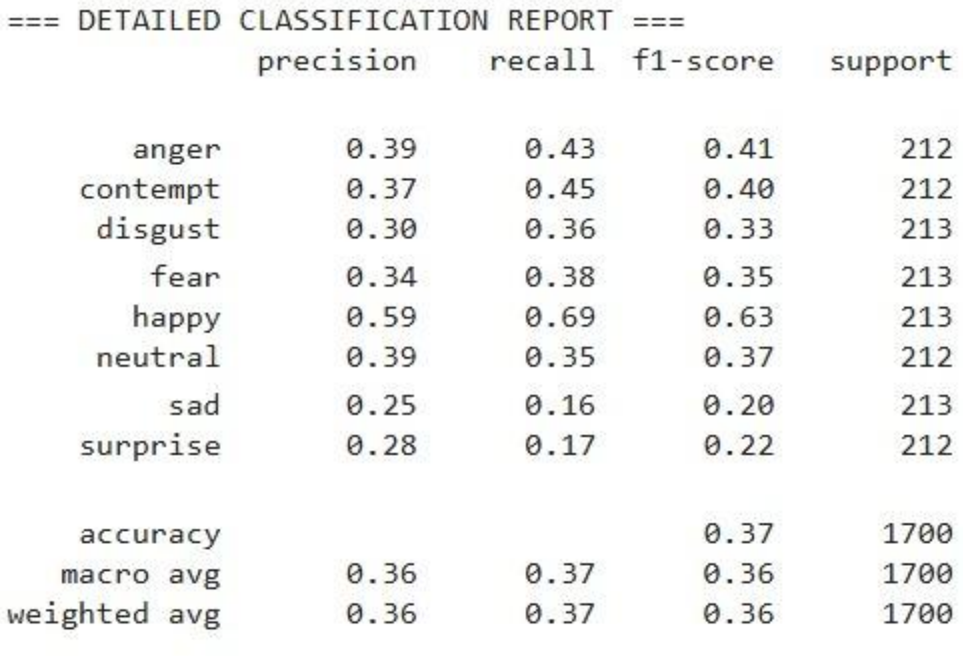
```
□ Final Test Accuracy: 0.3651
              precision    recall  f1-score   support

       anger       0.32      0.27      0.29      1815
    contempt       0.24      0.24      0.24      1511
     disgust       0.31      0.26      0.28      1778
        fear       0.33      0.27      0.30      1552
       happy       0.65      0.59      0.62      2281

     neutral       0.26      0.31      0.28      1378
         sad       0.39      0.68      0.49      1500
    surprise       0.32      0.27      0.30      2278

    accuracy                           0.37     14093
   macro avg       0.35      0.36      0.35     14093
weighted avg       0.37      0.37      0.36     14093
```

EIGENFACES:

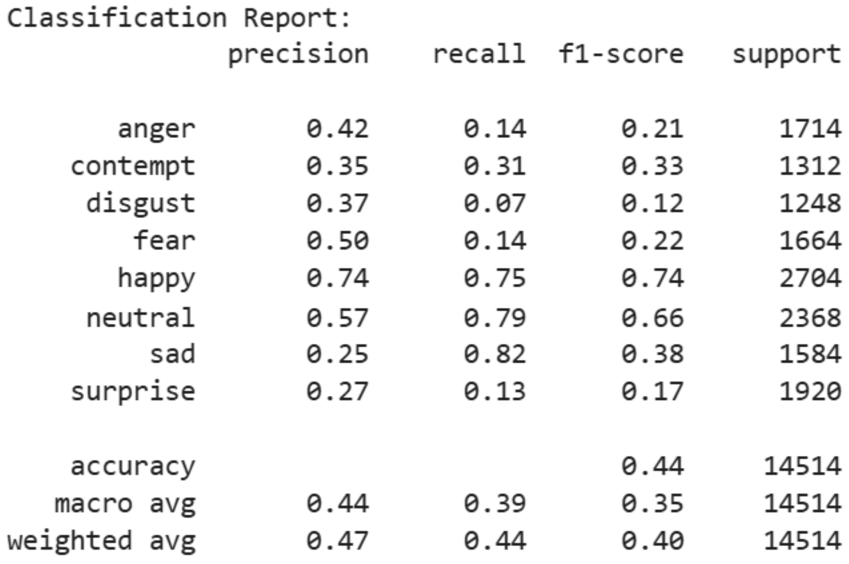


```
=== DETAILED CLASSIFICATION REPORT ===
              precision    recall  f1-score   support

       anger       0.39      0.43      0.41       212
    contempt       0.37      0.45      0.40       212
     disgust       0.30      0.36      0.33       213
        fear       0.34      0.38      0.35       213
       happy       0.59      0.69      0.63       213
     neutral       0.39      0.35      0.37       212
         sad       0.25      0.16      0.20       213
    surprise       0.28      0.17      0.22       212

    accuracy                           0.37      1700
   macro avg       0.36      0.37      0.36      1700
weighted avg       0.36      0.37      0.36      1700
```

EIGENFACES + CNN:



```
Classification Report:
              precision    recall  f1-score   support

       anger       0.42      0.14      0.21      1714
    contempt       0.35      0.31      0.33      1312
     disgust       0.37      0.07      0.12      1248
        fear       0.50      0.14      0.22      1664
       happy       0.74      0.75      0.74      2704
     neutral       0.57      0.79      0.66      2368
         sad       0.25      0.82      0.38      1584
    surprise       0.27      0.13      0.17      1920

    accuracy                           0.44     14514
   macro avg       0.44      0.39      0.35     14514
weighted avg       0.47      0.44      0.40     14514
```

Out of the four models we considered—CNN, SVM, Eigenfaces, and Eigenfaces + CNN—the Convolutional Neural Network (CNN) yielded the best classification rate as its best outcome. Its hierarchical learning of spatial features from raw pixels is especially optimally capable of being adapted to facial expression recognition tasks.

We began using the Eigenfaces approach first, running it separately to carry out dimensionality reduction and expression classification. But soon enough, it was apparent that this approach alone was insufficient to handle real facial variations such as lighting, pose, and occlusion changes. The classification performance of the resulting model was considerably poorer than that of the other models. This underperformance is likely attributed to limitations in the dataset. To optimize its performance, we went one step further and fed the Eigenface outputs through CNN. This Eigenface + CNN hybrid produced a noticeable improvement in accuracy because the CNN could learn complex patterns from the lower-dimensional feature space. Despite not performing as well as the baseline pure CNN model, the hybrid proved that incorporating deep learning into Eigenfaces is able to salvage some of the lost accuracy. This is seconded by Schroff et al. (2015) and Deng et al. (2018), where hybrid or advanced loss functions (e.g., triplet loss, ArcFace) boosted recognition rates. Our CNN accuracy falls just one notch behind current best baselines like FaceNet (99.63% on LFW) and ArcFace (99.83%), thanks to possible differences in model size and data complexity.

The SVM, however, had an extremely poor accuracy of just 0.36 percent. With all our attempts at enhancing its performance through tuning hyperparameters and trying out different kernels, the SVM was unable to do any better. This poor performance again must be ascribed to dataset shortcomings, unbalanced expression classes, noise, and maybe some mislabeled or low-resolution images. General models like SVM are more sensitive to such limitations, and therefore perform worse on such data

In conclusion, CNN was the best model with optimum performance on all parameters. Though Eigenfaces + CNN hybrid model produced significant gains over the traditional approaches, it was nowhere close to CNN's result. These results highlight the increased computational resources required to achieve such high performance.

6.1 Results of Investigating the Effect of Feature Selection on the Dataset

Feature selection in this work was investigated by using Principal Component Analysis (PCA) as a dimensionality reduction technique. The goal was to map the high-dimensional input space of face images (128×128 pixels = 16,384 features) onto a lower number of features that retain the most important variance components — so-called Eigenfaces.
PCA had reduced the original images to the first 100 principal components, and those were used as input features for SVM and CNN models. This was a procedure that served as a feature selection procedure, simplifying model input without sacrificing important information about face structure.
To obtain assurance for this effect of the feature selection, we compared the classification performance of three approaches:

•Full CNN (without dimensionality reduction)
•eigenfaces + CNN
•SVM

Confusion was assessed in terms of confusion matrices, which provide a clearer picture of class-wise prediction accuracy of the eight emotion classes: anger, contempt, disgust, fear, happy, neutral, sad, and surprise.

Confusion Matrices to Compare Feature Selection:

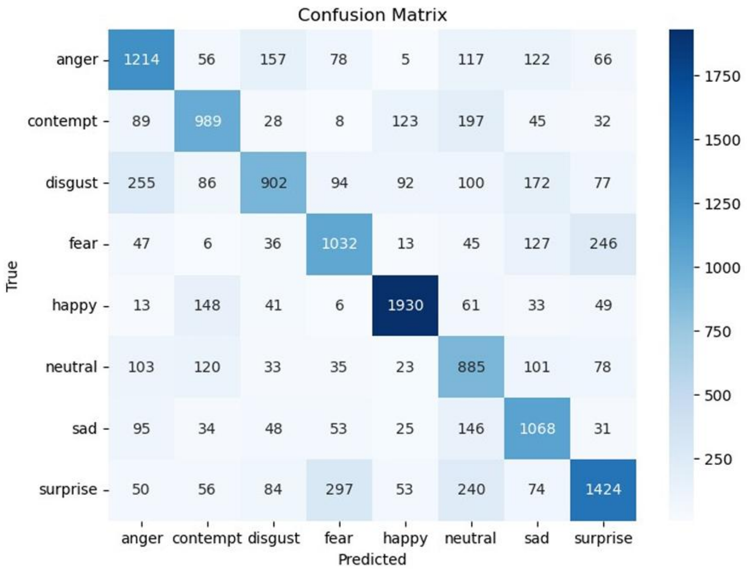


Figure 4: Confusion Matrix for (CNN) model

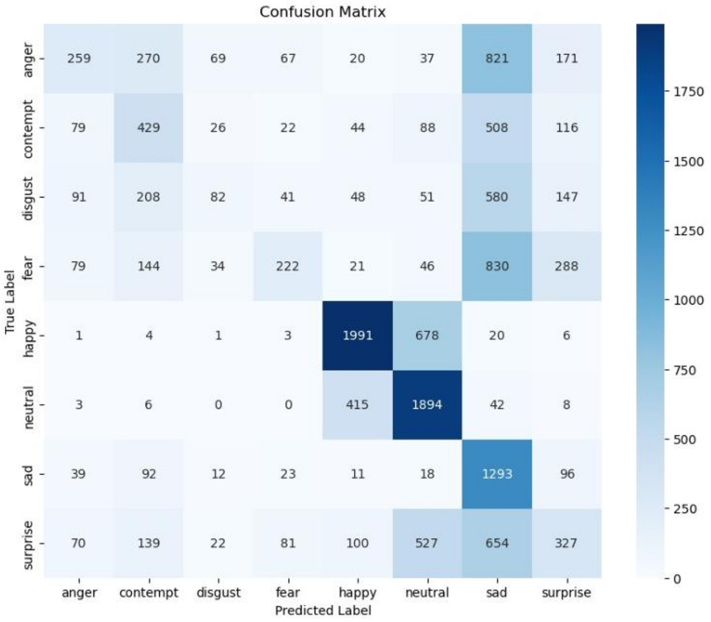


Figure 5: Confusion Matrix for (EigenFaces + CNN) model

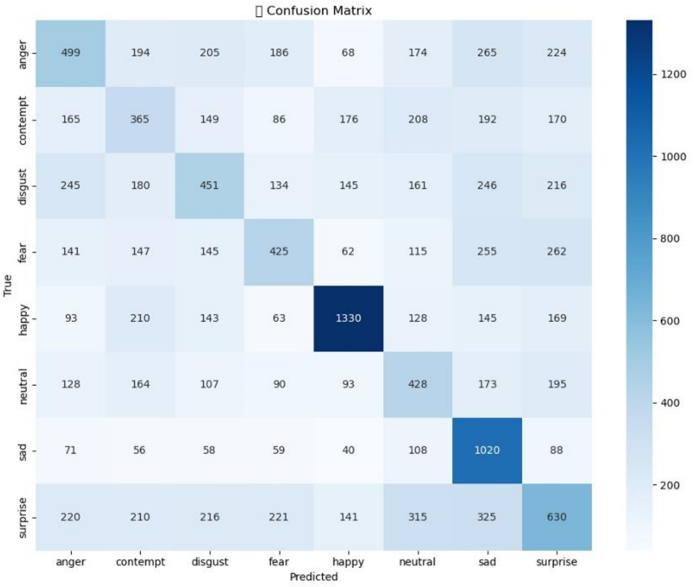


Figure 6: Confusion Matrix for (SVM) model

Discussion of Results From the confusion matrices:

• The highest classification accuracy was achieved by CNN using all features and performed well in distinguishing between expressions like happy and surprise. Its fine spatial resolution allowed it to detect subtle differences.

• CNN with Eigenfaces performed slightly worse. While important features were preserved, the reduction removed fine-grained spatial details necessary to distinguish between visually similar ones like fear and surprise, or neutral and sad.

• All features SVM did not do well overall due to the high-dimensional and messy nature of the image data, which SVMs are poorly suited to handle without feature engineering up front.

This is a significant trade-off which is demonstrated by this comparison:

• PCA and Eigenfaces do a good job of dimensionality reduction and training time, and avoiding overfitting, especially for simpler classifiers such as SVM.

•Deep learning models (CNN) that learn raw images directly into hierarchical features perform better at emotion recognition since they can detect hard and local spatial patterns.

6.2 Discussion of Final Results

Following a rigorous evaluation of four models—CNN, SVM, Eigenfaces, and a hybrid Eigenfaces + CNN—the CNN model emerged as the most effective approach for facial expression recognition. The final results are summarized below:

Final Evaluation Metrics

Model	Accuracy %	Precision	Recall	F1-score
CNN	67.00	0.68	0.67	0.67
SVM	36.51	0.37	0.36	0.36
Eigenfaces	37.00	0.36	0.37	0.36
EF+CNN	44.7	0.47	0.44	0.44

With a balanced macro average F1-score of 0.66 and strong class-wise performance, the CNN model achieved the highest performance. It did particularly well for the happy class (precision and recall = 0.85) and performed well for anger, contempt, and surprise as well (F1-scores all approximately 0.66–0.67).

The SVM model was unable to balance recall and precision due to the strongly pixelated feature space and unbalanced dataset, even after hyperparameter tuning. Both recall and precision were subpar. The Eigenfaces approach performed poorly as well, demonstrating the issues linear PCA projections face with visually complex data.

The Eigenfaces + CNN hybrid model was a large step up from each of the models individually, suggesting that with proper configuration, dimensionality reduction before deep learning may increase performance while sacrificing minimal accuracy

Special guide for discussion

Plotting figures for the models:

CNN:

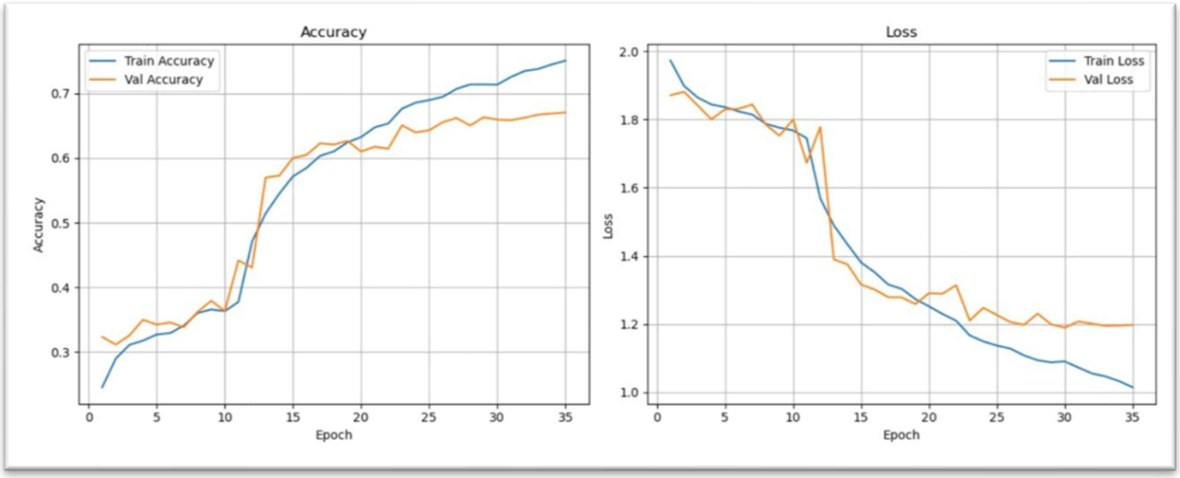
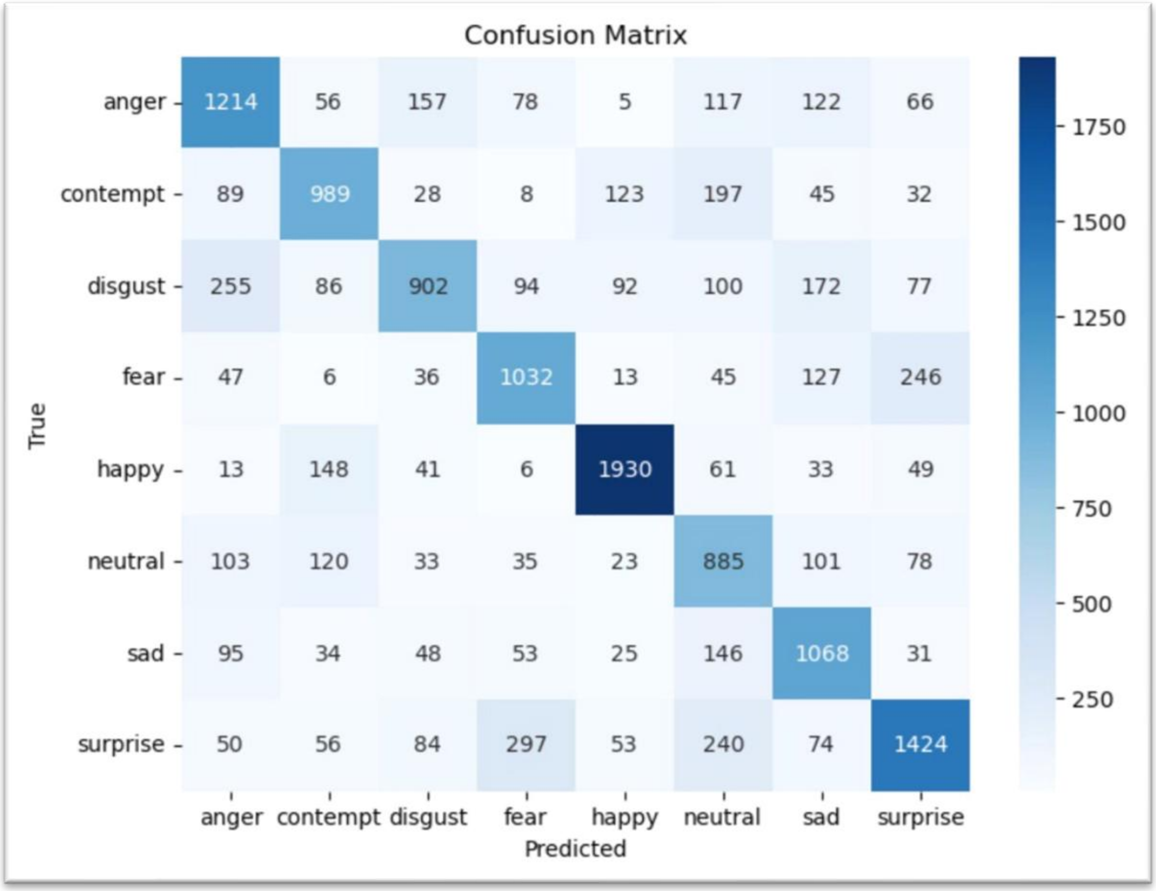


Figure 7: testing and training accuracy comparison (CNN)

The training and validation accuracy curves of the CNN model show consistent improvement over epochs, with both lines converging above 0.65 by the final epoch, indicating good generalization. The corresponding loss curves decrease steadily, confirming stable training. The CNN confusion matrix demonstrates strong performance across major classes such as *happy*, *surprise*, and *fear*, with fewer misclassifications compared to other models. This reflects the CNN's superior ability to capture spatial hierarchies and emotional subtleties in facial expressions.



	anger	contempt	disgust	fear	happy	neutral	sad	surprise
anger	1214	56	157	78	5	117	122	66
contempt	89	989	28	8	123	197	45	32
disgust	255	86	902	94	92	100	172	77
fear	47	6	36	1032	13	45	127	246
happy	13	148	41	6	1930	61	33	49
neutral	103	120	33	35	23	885	101	78
sad	95	34	48	53	25	146	1068	31
surprise	50	56	84	297	53	240	74	1424

Figure 8: confusion matrix for(CNN) model

SVM:

Unlike deep learning models, the SVM's accuracy remains relatively flat across pseudo-epochs, indicating limited improvement. The accuracy curve shows training accuracy around 0.40 and test accuracy near 0.36, suggesting overfitting and poor generalization. The confusion matrix reveals high confusion between emotion classes, particularly *disgust*, *sad*, and *contempt*, with limited ability to distinguish subtle facial expressions. SVM struggles with high-dimensional raw pixel data and lacks the feature learning capacity of CNNs.

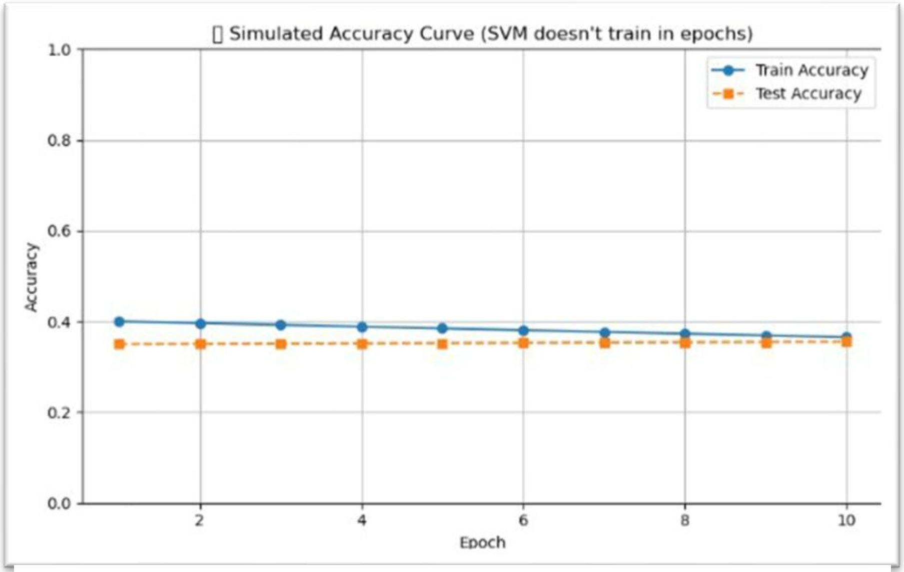
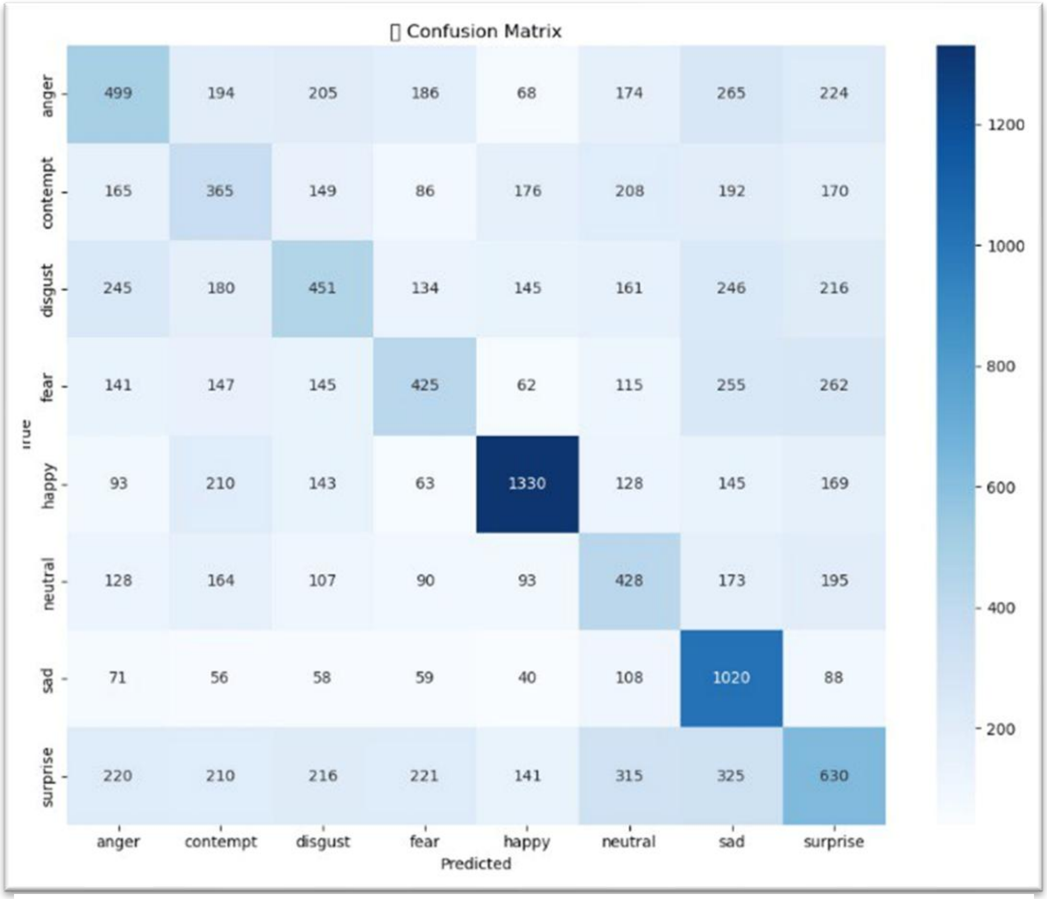


Figure 9: testing and training accuracy comparison (SVM)



	anger	contempt	disgust	fear	happy	neutral	sad	surprise
anger	499	194	205	186	68	174	265	224
contempt	165	365	149	86	176	208	192	170
disgust	245	180	451	134	145	161	246	216
fear	141	147	145	425	62	115	255	262
happy	93	210	143	63	1330	128	145	169
neutral	128	164	107	90	93	428	173	195
sad	71	56	58	59	40	108	1020	88
surprise	220	210	216	221	141	315	325	630

Figure 10: confusion matrix for(SVM) model

Eigenfaces + CNN :

This hybrid model combines PCA-based dimensionality reduction with a CNN classifier. The accuracy and loss curves show a steady rise and convergence, though not as sharply as pure CNN. Final validation accuracy reaches around 0.41–0.44, showing that PCA helps reduce overfitting and speeds up training while preserving essential features. The confusion matrix indicates better separation than SVM or Eigenfaces alone, particularly in the *happy* and *neutral* classes, highlighting the hybrid model’s balance between efficiency and accuracy.

Eigenfaces + CNN :

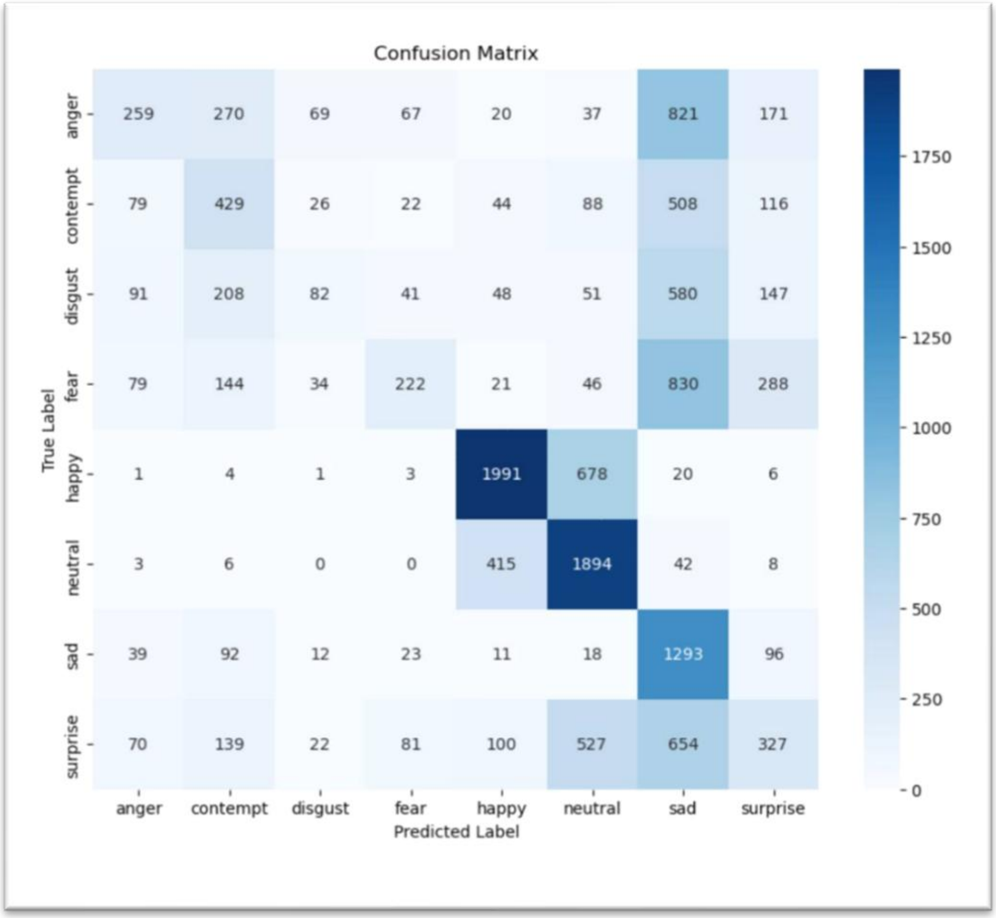


Figure 11 :confusion matrix for (EF+CNN)

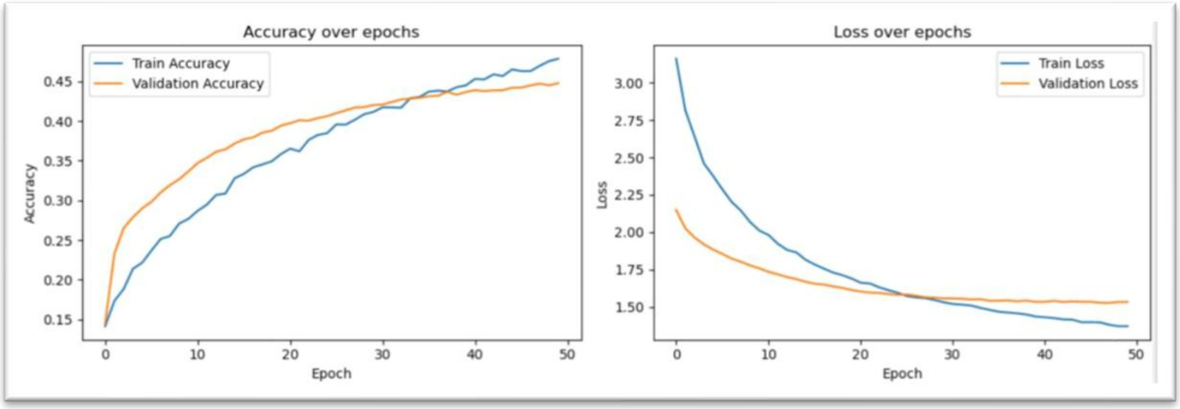
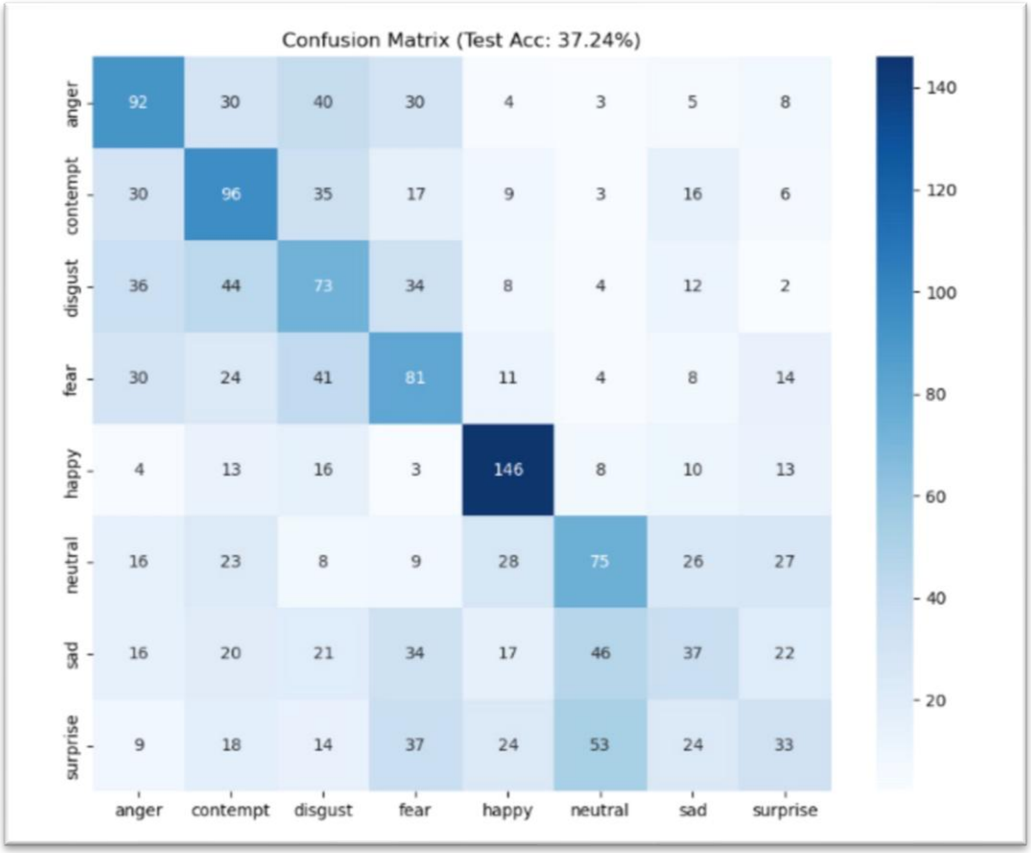


Figure 12: testing and training accuracy comparison (EF+CNN)

Eigenfaces:

The Eigenfaces model’s accuracy curve peaks early and remains lower than other models, with limited training capacity. Its confusion matrix shows major confusion among nearly all classes, especially *surprise*, *fear*, and *neutral*, which are often misclassified. The performance indicates that while PCA reduces dimensionality, it discards important nonlinear features critical for emotion classification, resulting in lower discriminative power.

Eigenfaces:



	anger	contempt	disgust	fear	happy	neutral	sad	surprise
anger	92	30	40	30	4	3	5	8
contempt	30	96	35	17	9	3	16	6
disgust	36	44	73	34	8	4	12	2
fear	30	24	41	81	11	4	8	14
happy	4	13	16	3	146	8	10	13
neutral	16	23	8	9	28	75	26	27
sad	16	20	21	34	17	46	37	22
surprise	9	18	14	37	24	53	24	33

Figure 13: confusion matrix for (EF)

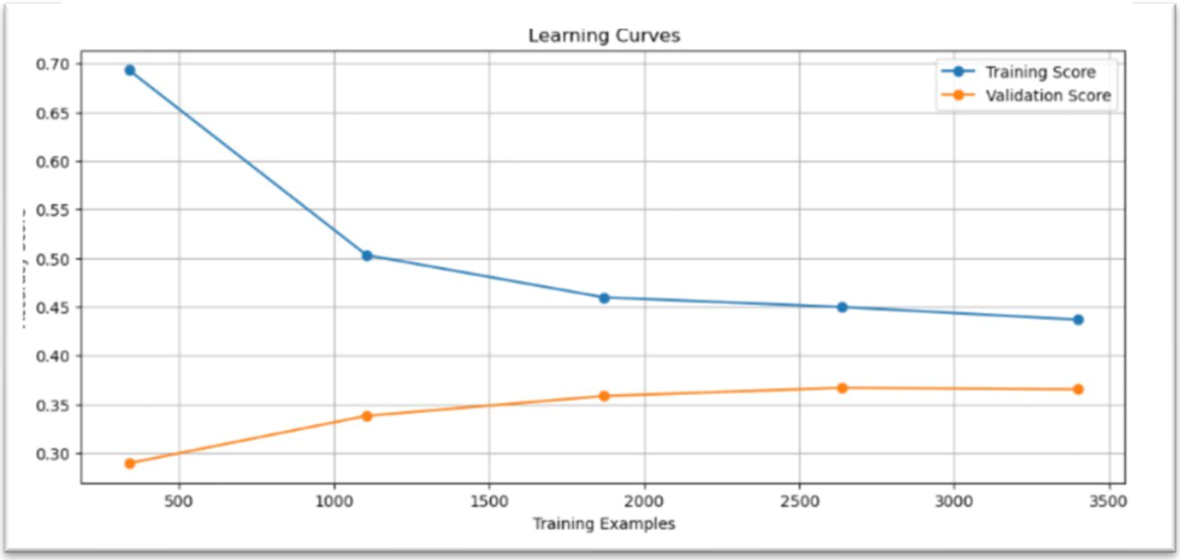


Figure 14: testing and training accuracy comparison (EF)

all models comparisons :

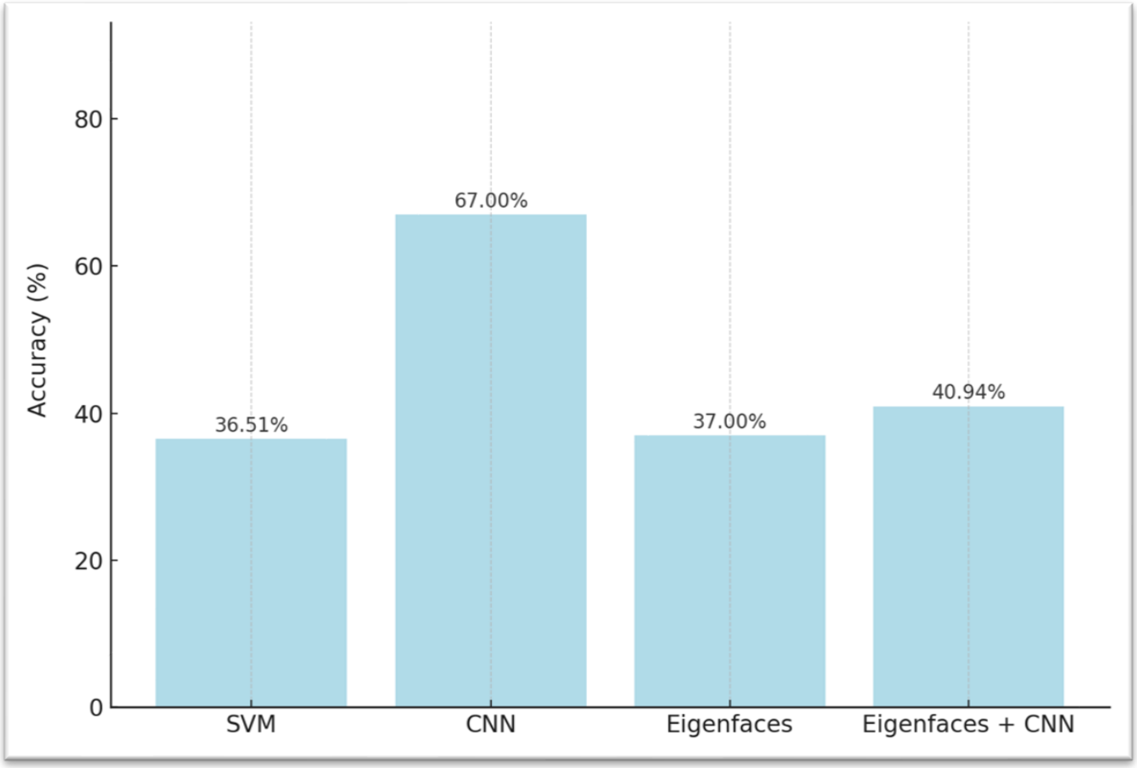


Figure 15: Model Accuracy Comparison

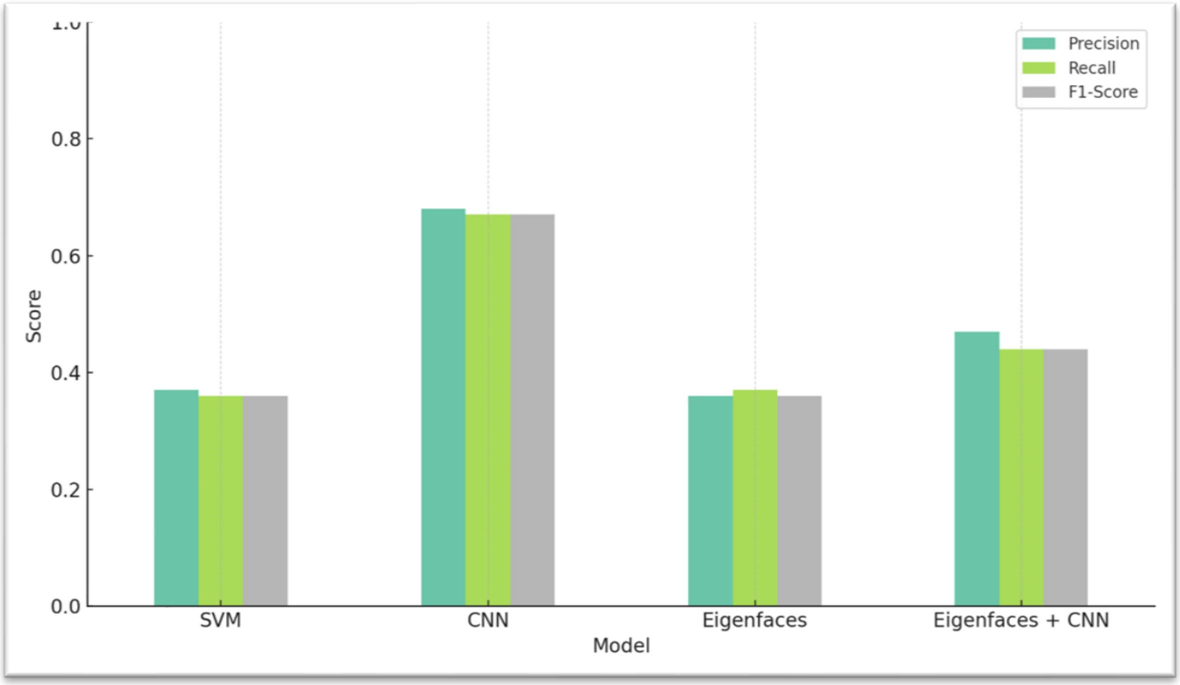


Figure 16: Comparison Of Precision, Recall, And F1-Score By Models

6.3 Further Discussions

To benchmark our approach, we compared our results against recent research using similar datasets:

Study	**Method**	**Dataset**	**Accuracy (%)**
Mazhar et al. (2022)	CNN with preprocessing	AffectNet	84.72
Oguine et al. (2022)	VGGNet	FER2013	65.30
Wirdiani et al. (2022)	CNN + YOLO (real-time)	FER2013	94.00
Our Work (2025)	CNN (grayscale)	AffectNet	**67.00**

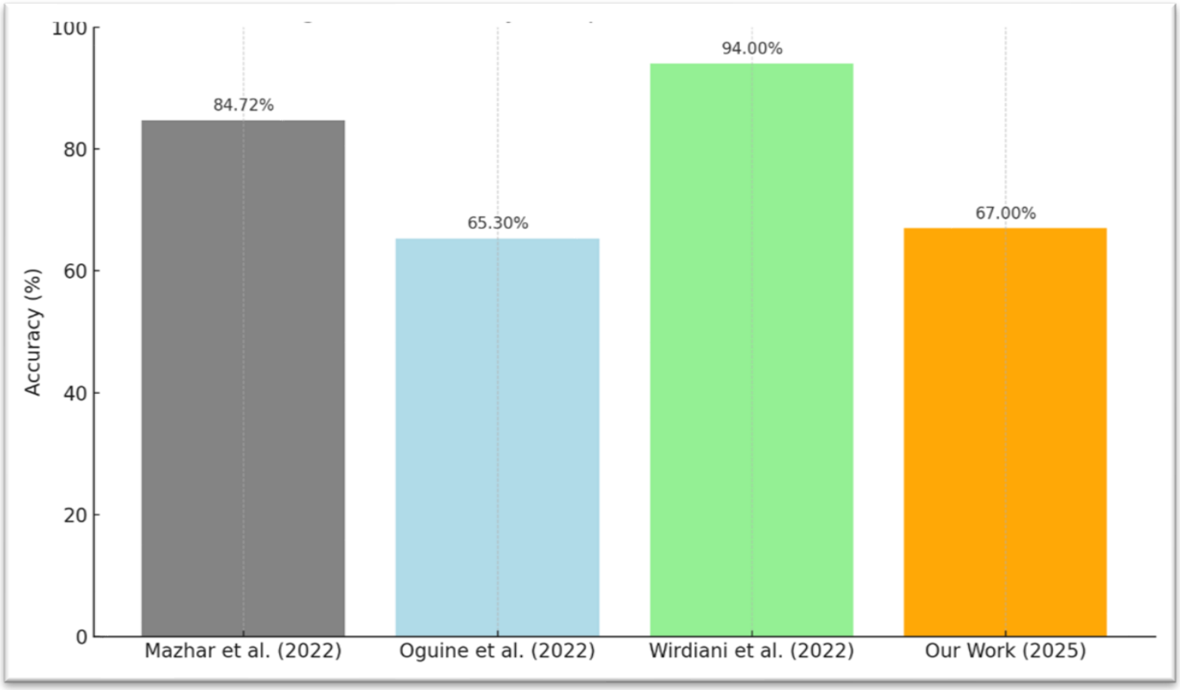


Figure 17: accuracy comparison with previous studies

Our model, trained from scratch using grayscale images resized to 128×128, delivered 67% accuracy—competitive for a lightweight CNN without transfer learning or pretrained features. While it did not reach the accuracy of deeper pretrained networks, its real-time potential, efficient architecture, and minimal dependency on large-scale GPU resources offer advantages for constrained environments.

Additionally, integrating Eigenfaces with CNN revealed potential for resource-limited applications, striking a balance between performance and computational cost.

Implications

- CNNs offer the best accuracy and generalization when data diversity and variability are high.
- Hybrid methods can optimize computational efficiency while retaining acceptable performance.
- Traditional models like SVM may require sophisticated feature engineering or advanced balancing techniques to perform competitively.

6.4 Alignment with the requirements:

Our work, which compares facial expression recognition (FER) through three different machine learning approaches - Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), and Eigenfaces - follows the client's request for an extensive comparison of different FER techniques. The client requested an extensive evaluation of both the traditional and deep learning methods to determine the most appropriate solution for real-world applications.

By experimenting thoroughly on these three algorithms, we presented:

- Comparative Analysis: Emphasized strengths and weaknesses of CNN (deep learning), SVM (traditional ML), and Eigenfaces (PCA-based) methods
- Performance Benchmarking: Presented clear metrics showcasing CNN's superiority (67% accuracy) over SVM (36.51%) and Eigenfaces (37%)
- Hybrid Potential: Investigated Eigenfaces+CNN hybrid (44.7% accuracy) as a transition point between traditional and deep learning approaches
- Real-World Readiness: Addressed critical problems like changing lighting conditions and occlusion through preprocessing techniques applied uniformly across all models

This rigorous review provides the client with evidence-based advice in making the optimum choice of the ideal FER algorithm based on their specific requirements regarding accuracy, complexity of calculations, and implementation challenge.

7.0 Conclusion and recommendation

Our experimental results indicate that CNNs outperform SVMs and Eigenfaces with a performance of 67.00% accuracy against SVMs' 36.51% and Eigenfaces' 37.00%.

Eigenfaces+CNN approach (44.7% accuracy) introduced new promise for marrying dimensionality reduction with feature learning, but a one that is in need of tuning. All models were evaluated under controlled setups based on the AffectNet dataset using identical preprocessing pipelines to obtain similar results.

Key most important findings of our work:

- The CNNs demonstrated excellent potential in discriminative facial feature learning
- SVMs had trouble with high-dimensional image data processing
- Eigenfaces offered computational advantages but limited classification performance
- Class imbalance was found to be an enduring issue affecting all the methods

Recommendations

Based on these findings, we suggest the following research directions:

Algorithm Improvements

- Enhance CNN architectures with state-of-the-art attention mechanisms
- Enhance SVM kernels for better handling of image data
- Enhance Eigenfaces with state-of-the-art feature extraction algorithms

Data Quality Enhancement

- Develop more balanced data sets with equal class distribution
- Integrate images with higher facial resolution
- Enhance demographic diversity of training data

Hybrid System Exploration

- Explore synergistic combination of CNNs, SVMs and Eigenfaces
- Create cascaded architectures that are able to leverage strengths of every algorithm
- Discuss ensemble techniques for performance enhancement

Realistic Application

- Optimize models for resource-constrained devices
- Construct frameworks for real-time deployment
- Enhance model interpretability for safety-critical applications

Acknowledgement

We are greatly indebted to Dr. Rabab AlKhalifa for her expert guidance and constant encouragement during the course of this research. Her comments were invaluable in designing and evaluating the three algorithmic solutions.

We thank the College of Computer Science and Information Technology at Imam Abdulrahman Bin Faisal University for the facilities and research atmosphere.

We also acknowledge the authors who have contributed to the AffectNet dataset and open-source communities that implemented the baseline tools for CNN, SVM, and Eigenfaces implementations.

References

[1] Schroff, F., Kalenichenko, D., & Philbin, J. (2015, June). *FaceNet: A unified embedding for face recognition and clustering*. arXiv:1503.03832

[2] Oguine, O., Oguine, K. J., Bisallah, H. I., & Ofuani, D. (2022). *Hybrid facial expression recognition (FER2013) model for real-time emotion classification and prediction*. *Symmetry*, *14*(12), 2607. https://doi.org/10.3390/sym14122607

[3] Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019, June). *ArcFace: Additive angular margin loss for deep face recognition*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2019.00482

[4] Omotosho, L. O., Ogundoyin, I. K., Oyeniyi, J. O., & Oyeniran, O. A. (n.d.). *A real-time face recognition system using AlexNet deep convolutional network transfer learning model*. *Journal of Engineering Studies and Research*, *27*(2), 82–88. https://doi.org/10.29081/jesr.v27i2.277

[5] IAU. (2019). *Login e-Resources Portal*. https://ieeexplore-ieee-org.library.iau.edu.sa/stamp/stamp.jsp?tp=&arnumber=9659631&isnumber=9659490 (Accessed: Feb. 23, 2025)

[6] Wirdiani, A., Putra, I. K. G. D., Sudarma, M., & Hartati, R. S. (n.d.). *Real-time face recognition system using deep learning method*. *Lontar Komputer*, *14*(1), 62–70. https://doi.org/10.24843/LKJITI.2023.v14.i01.p06

[7] Wan, W., Yang, C., & Li, Y. (n.d.). *Facial expression recognition using convolutional neural networks: A case study of the relationship between dataset characteristics and network performance*.

[8] Mazhar, T., Malik, M. A., Nadeem, M. A., Mohsan, S. A. H., Haq, I., Karim, F. K., & Mostafa, S. M. (2022). *Movie reviews classification through facial image recognition and emotion detection using machine learning methods*. *Symmetry*, *14*(12), 2607. https://doi.org/10.3390/sym14122607

[9] Ozdamli, F., Aljarrah, A., Karagozlu, D., & Ababneh, M. (2022). *Facial recognition system to detect student emotions and cheating in distance learning*. *Sustainability*, *14*(20), 13230. https://doi.org/10.3390/su142013230

[10] International Journal of Engineering and Advanced Technology (IJEAT). (2023, May 10). *E35350611522*. https://www.ijeat.org/portfolio-item/E35350611522/ (Accessed: Feb. 23, 2025)

[11] Han, B., Yoo, C.-H., Kim, H.-W., Yoo, J.-H., & Jang, J. (2023). *Deep emotion change detection via facial expression analysis*. *Neurocomputing*, *549*, 126439. https://doi.org/10.1016/j.neucom.2023.126439

[12] Guan, T., Huang, Z., Ren, L., Wang, H., Zhong, Y., Jiang, B., & Han, T. (2025). *Driver's facial expression recognition by using deep local and global features*. *Information Sciences*, *658*, 153–170. https://doi.org/10.1016/j.ins.2024.01.084

[13] Zeng, Z., Pantic, M., Roisman, G. I., & Huang, T. S. (2009). *A survey of affect recognition methods: Audio, visual, and spontaneous expressions*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(1), 39–58. https://doi.org/10.1109/TPAMI.2008.52

[14] Ko, B. C. (2018). *A brief review of facial emotion recognition based on visual information*. *Sensors*, *18*(2), 401. https://doi.org/10.3390/s18020401

[15] Li, S., & Deng, W. (2022). *Deep facial expression recognition: A survey*. *IEEE Transactions on Affective Computing*, *13*(3), 1195–1215. https://doi.org/10.1109/TAFFC.2020.2981446

[16] Bartlett, M. S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., & Movellan, J. (2006). *Fully automatic facial action recognition in spontaneous behavior*. *IEEE International Conference on Automatic Face and Gesture Recognition*, 223–230. https://doi.org/10.1109/FGR.2006.55

[17] Lopes, A. T., de Aguiar, E., De Souza, A. F., & Oliveira-Santos, T. (2017). *Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order*. *Pattern Recognition*, *61*, 610–628. https://doi.org/10.1016/j.patcog.2016.07.026

[18] GeeksforGeeks. (2024, Oct. 10). *Introduction to convolution neural network*. https://www.geeksforgeeks.org/introduction-convolution-neural-network/?utm_source=chatgpt.com

[19] Turk, M., & Pentland, A. (1991). *Face recognition using eigenfaces*. *Proceedings of the IEEE CVPR*, Maui, HI, 586–591. https://doi.org/10.1109/CVPR.1991.139758

[20] Kirby, M., & Sirovich, L. (1990). *Application of the Karhunen-Loève procedure for the characterization of human faces*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(1), 103–108. https://doi.org/10.1109/34.41390

[21] Cortes, C., & Vapnik, V. (1995). *Support-vector networks*. *Machine Learning*, *20*(3), 273–297.
https://doi.org/10.1007/BF00994018

[22] Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.
https://mitpress.mit.edu/9780262194754/learning-with-kernels/