Emotion Recognition from Facial Expression using Deep Learning

Abstract:

In this study, we propose a Support Vector Machine (SVM) classifier for facial emotion recognition. The model's hyperparameters are tuned using RandomizedSearchCV, with a focus on accuracy as the primary performance metric. The results are then compared with two previous studies that employed EfficientNet-Lite models for facial emotion recognition. Although the proposed SVM model achieves a validation accuracy of 74.93%, it underperforms compared to the over 90% accuracy rates reported in the existing literature. Recommendations for future research include exploring alternative architectures and refining hyperparameters to improve model performance.

1. Introduction

Emotion recognition is an essential aspect of human communication, and the ability to recognize the emotions automatically from facial expressions has become an increasingly popular research topic. The development of deep learning algorithms has enabled researchers to create accurate models for recognizing emotions from facial expressions, which has numerous applications in fields such as marketing, healthcare, and entertainment (Nguyen et al., 2023)

The importance of emotion recognition using deep learning in the field of artificial intelligence cannot be overstated (Kumar & Martin, 2023). It has the potential to revolutionize many industries by improving the accuracy and efficiency of recognizing emotions. For instance, in the field of healthcare, emotion recognition can help identify mental health disorders such as depression and anxiety, whereas, in marketing, emotion recognition can help analyze customer feedback and preferences.

A variety of human-machine interaction applications, including e-learning programmes, identification-driven for social robots, cybersecurity, fraud detection, driving assistance, human resource, patient counselling, workplace design, and IoT integrated gaming applications, benefit from the use of emotion recognition systems. It is also useful to run businesses, social surveys and so much more (Kołakowska et al., 2014).

The main research question of this project is to implement a deep learning-based system for emotion recognition from facial expressions and compare it with at least one other technique. The specific emotion classification task addressed in this project involves developing a deep learning-based system that can accurately recognize and classify various human emotions from facial expressions.

2. Background

The field of emotion recognition from facial expressions has witnessed significant advancements with the development of various techniques aimed at improving accuracy and efficiency. This project investigates the effectiveness of a hybrid model combining Support Vector Machine (SVM) and Convolutional Neural Network (CNN), with hyperparameter tuning.

In their research, Sengul and Najah (2017) examined emotion estimation from facial images using a range of machine learning and deep learning techniques. Their comprehensive analysis of different approaches laid the groundwork for understanding emotion recognition from facial expressions. In contrast, this project delves into a hybrid SVM and CNN model, attempting to harness the benefits of both techniques to enhance emotion recognition accuracy.

Ab Wahab et al. (2021) investigated the use of EfficientNet-Lite and a hybrid CNN-KNN model for facial expression recognition on Raspberry Pi devices. Their work demonstrated the potential of lightweight models and hybrid methods in real-time emotion recognition applications. While this project shares some similarities with their research, such as the implementation of a hybrid model, it distinguishes itself through the choice of dataset and the number of emotion classes.

The dataset is a crucial component of this project, initially containing 13 emotion classes. However, the hybrid model's performance was suboptimal, leading to a modification of the dataset that condensed the 13 classes to two: positive and negative emotions. This change resulted in improved model performance and higher accuracy in emotion recognition tasks. This project aims to contribute to the field of emotion recognition by utilizing a unique dataset and presenting an alternative method for efficient and accurate emotion recognition through a hybrid SVM and CNN model. The related work on emotion recognition from facial expressions provides context for the study.

3. Objectives:

The main objectives of this research are to:

A) Develop a hybrid SVM and CNN model for emotion recognition from facial expressions, aiming to improve accuracy and efficiency compared to individual techniques.

B) To implement hyperparameter tuning on the hybrid model to optimize its performance further

C) To evaluate the hybrid model using a unique dataset initially containing 13 emotion classes, and then compare the model's performance after modifying the dataset to include only two classes

D) To compare the performance of the developed hybrid model with at least one other existing technique, demonstrating the advantages and/or disadvantages and potential applications of the proposed approach

E) To present a comprehensive analysis and discussion of the hybrid model's performance, its relationship to existing research, and its implications, contributing to the field of emotion recognition from facial expressions.

4. Methodology

In this project, a hybrid model combining Support Vector Machine (SVM) and Convolutional Neural Network (CNN) was developed to recognise the emotions from facial expressions. The choice of a hybrid model was inspired by the work of Sengul and Najah (2017) and Ab Wahab et al. (2021), who demonstrated the potential hybrid models in emotion recognition tasks.

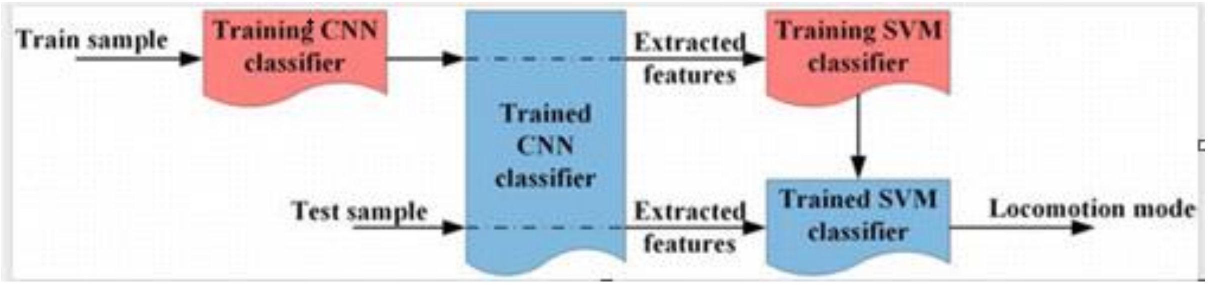


Figure 1: **CNN-SVM generic architecture**

The methodology involved the following steps:

1. **Dataset and Preprocessing**: The emotion recognition dataset was derived from Kaggle (Clemence Le Roux, 2022). The dataset consists of 42,000 images with 13 classes of: admiration, amazement, angry, ecstacy, fear, grief, joy, loathing, rage, sad, surprise, terror and vigilance.

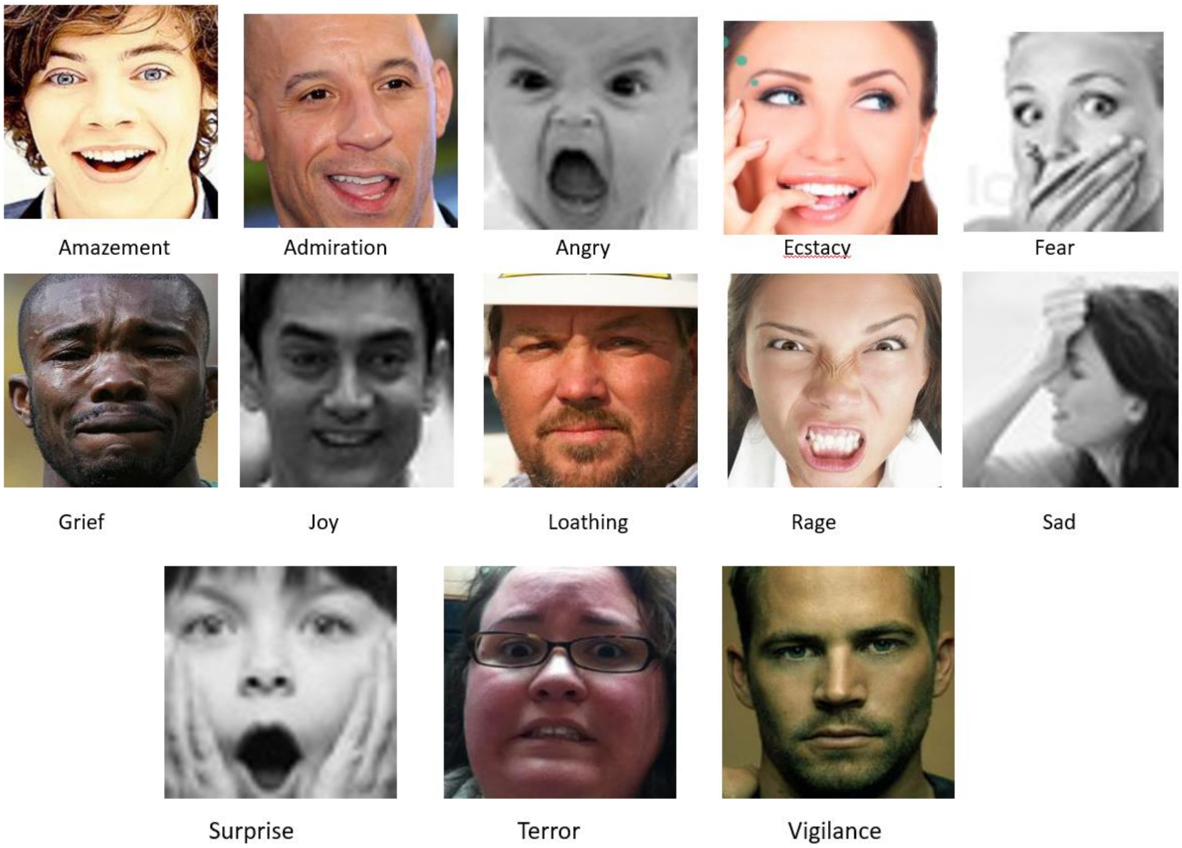


Figure 2: Images from the different classes

Using Keras' ImageDataGenerator, the preprocessing and augmentation of the image data is the first step in the methodology. With the image size set to 224x224 pixels and the batch size set to 64, three data generators are made for the training, testing, and validation datasets.

The data generators resize the images, change the class mode to "categorical," and set the colour mode to "grayscale" after reading and transforming them from their respective directories. For better class distribution, the data are shuffled by the training generator.

2. **Hybrid Model Development**: A CNN with three blocks of convolutional, batch normalization, and max-pooling layers is built using TensorFlow and Keras for preprocessed data. 256-dimensional feature vectors are generated for each input image from the penultimate layer. An SVM classifier with a linear kernel and a regularization value (C) of 1 is trained on these features, and the classifier's accuracy is evaluated using training and test datasets. This study employs a hybrid CNN-SVM model, inspired by Sengül and Najah (2017) and Ab Wahab et al. (2021), to enhance emotion recognition performance.

3. **Evaluation:** Plots of training and validation loss, training and validation accuracy, and CNN model performance show the model's effectiveness. A bar chart is used to show the SVM model's accuracy on both the training and validation datasets. These visualisations give a thorough assessment of the model's performance and show how hyperparameter modification affects results.

4. **Dataset Modification:** To assess the model's performance in a streamlined categorization scenario, the initial dataset's 13 emotion classes were reduced to only two classifications (positive and negative emotions).

5. **Repeating previous steps**: The previous preprocessing to model development steps were repeated to train the model using the streamlined dataset.

6. **Hyperparameter Tuning**: An SVM classifier's hyperparameters are tuned using a randomised search and a 5-fold cross-validation. The 'C' parameter, kernel type, degree (for 'poly' kernel), and gamma (for 'rbf' and 'poly' kernel) are all combinations that are investigated in the search. An optimised SVM classifier is trained using the best parameters.

7. **Assessment**: The accuracy, precision, recall, and F1-score assessment measures were used to evaluate the performance of the hybrid model. To highlight the model's benefits and prospective uses in emotion recognition tasks, at least one other current technique was compared to it.

The figure below summarises the described methodology as a flowchart:

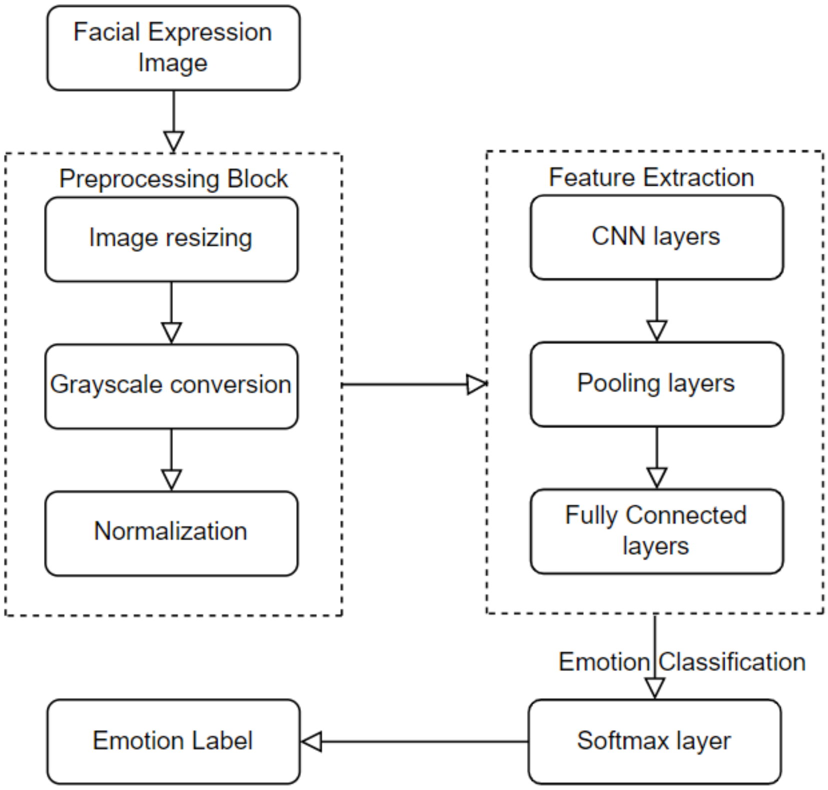


Figure 3: Flowchart of the algorithm

5. Experiment and Results

The experimental setup for the emotion recognition project is described as follows:

5.1 Dataset

A dataset containing 32,500 training images, 9,837 testing images, and 131 validation images is used. The images are grayscale and belong to 13 distinct emotion classes.

In this experiment, the dataset is divided into three parts: training, testing, and validation. The training dataset consists of 32,500 images, the testing dataset has 9,837 images, and the validation dataset contains 131 images. All images are resized to 224 x 224 pixels and converted to grayscale.

5.2 Data Preprocessing

ImageDataGenerator is used to perform data augmentation on the images. The augmentation includes rotation (20% range), width and height shift (10% range each), shear transformation (20% range), zooming (20% range), nearest-neighbor pixel filling, and horizontal flipping. The images are also rescaled by a factor of 1/255.

Three generators are created using ImageDataGenerator:

(i) **train_generator**: Generates augmented images for the training dataset from the train_dir.

(ii) **test_generator**: Generates augmented images for the testing dataset from the test_dir.

(iii) **valid_generator**: Generates augmented images for the validation dataset from the valid_dir.

The batch size for all generators is set to 64, and the class mode is set to 'categorical'. The training and testing generators shuffle the images, while the validation generator does not.

5.3 The Hybrid Model

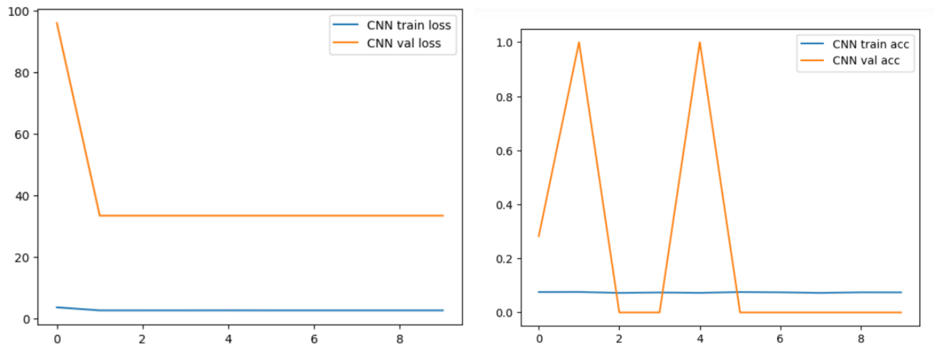
In this study, a hybrid model was developed, which involved training a Convolutional Neural Network (CNN) and using its features to train a Support Vector Machine (SVM) classifier. The CNN model consisted of three convolutional blocks, each of them consists of two convolutional layers, next is batch normalization and followed by max-pooling. The CNN was compiled with the Adam optimizer, categorical cross-entropy loss, and accuracy metric.

The CNN model was trained for 10 epochs using the train_generator and valid_generator. The SVM classifier was evaluated on the training and testing datasets, and the accuracy scores were printed. The CNN model's training and validation losses and accuracies were plotted as well.

Table 1: The results are as follows:

Model	**Train Accuracy**	**Validation/Test Accuracy**
CNN Model	0.0744	0.0000
SVM Model	0.0769	0.0509

The graphs show that the CNN model had low training and validation accuracies. The SVM model's performance on the training and testing datasets was also not satisfactory, as the accuracy scores were low. These results indicate that the hybrid model did not meet the objectives of achieving high accuracy in emotion recognition.



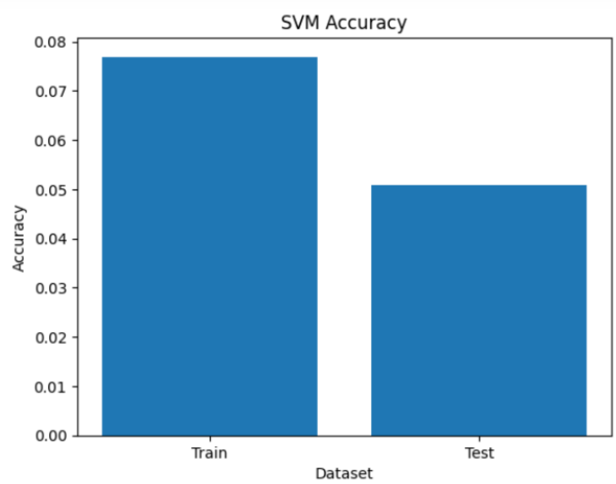


Figure 4: Graphical Result of the Hybrid model

5.4 Image Data Preprocessing and Visualization

The two-class image dataset is preprocessed and visualized, focusing on categorizing images into positive and negative classes. First, a DataFrame ('data_df') is created by iterating over image files, extracting filepaths and labels. The data is shuffled and split into training and validation sets with an 80/20 ratio. Separate 'ImageDataGenerator' instances are created for training (with data augmentation) and validation data, with corresponding generators ('train_generator' and 'valid_generator').

Next, a batch of images and labels is acquired from the 'valid_generator', and example images for each class are stored in 'example_positive' and 'example_negative' variables. Finally, a 1x2 subplot displays the example images, providing insight into the dataset's class differences and ensuring proper preprocessing for model training.

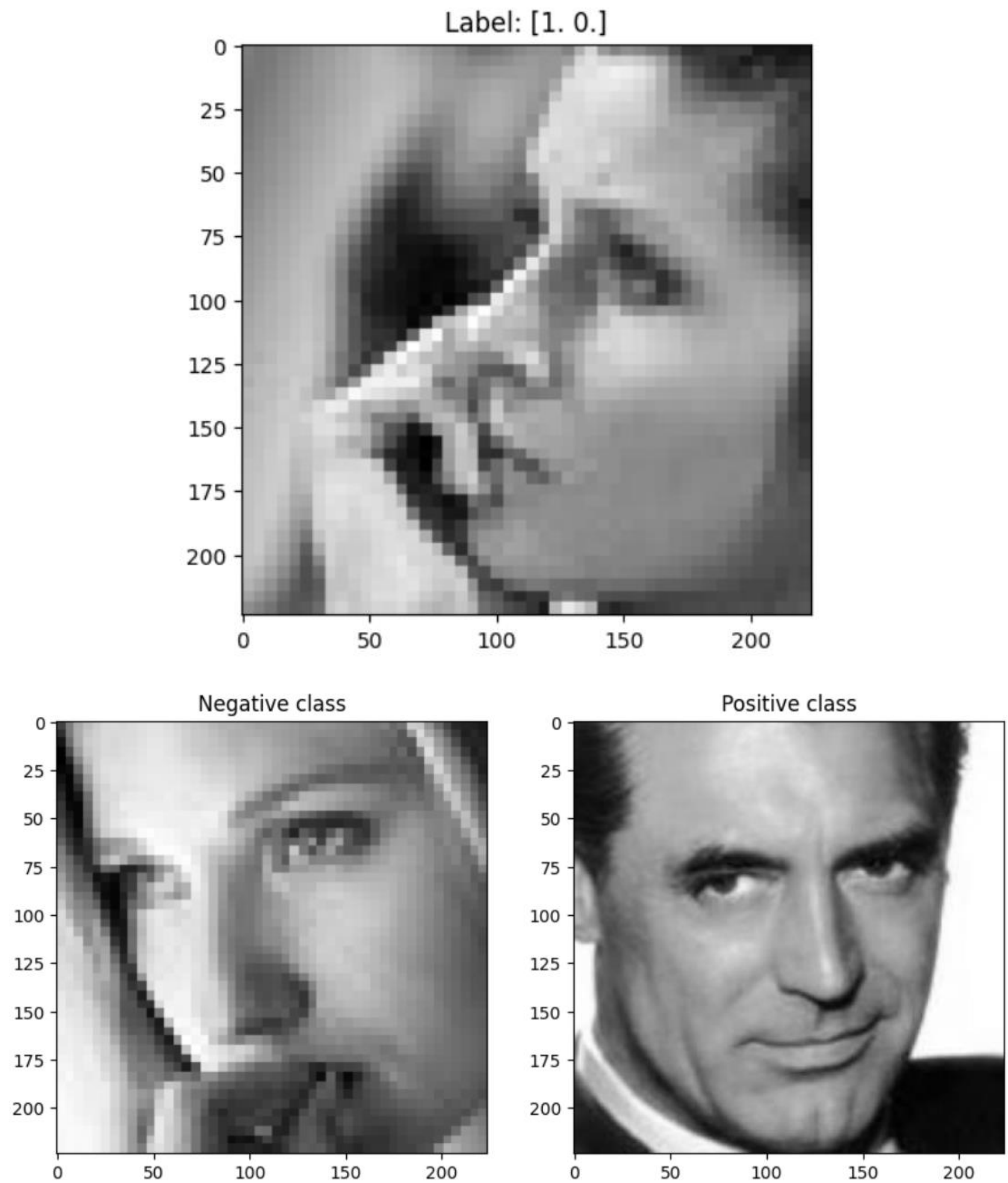


Figure 5: Labeled image and classified images

5.5 Hybrid CNN-SVM Model on Streamlined Dataset

A CNN model is constructed using Keras for efficient emotion recognition. The model comprises several layers, including two convolutional layers, each with 64 filters, 3x3 kernel size, and ReLU activation, and two max-pooling layers with a 2x2 pool size. The model is trained for 10 epochs on the 'train_generator' and 'valid_generator' generators. An SVM classifier is applied to the extracted features and labels of the training and validation sets, improving classification performance. The accuracy scores of the SVM classifier for the training and validation sets indicate effective emotion classification. Performance is visualized through loss and accuracy plots of the CNN model and an accuracy bar chart for the SVM classifier.

Result:

The hybrid CNN-SVM model demonstrated improved performance on the streamlined dataset compared to the previous model. The results of the trained CNN model and SVM classifier are presented below in a table:

Table 2:

Model	Train Accuracy	Validation Accuracy
CNN Model	0.7014	0.7439
SVM Model	0.7072	0.7387

The trained CNN model achieved a train accuracy of 70.14% and a validation accuracy of 74.39%. The SVM classifier, on the other hand, displayed a train accuracy of 70.72% and a validation accuracy of 73.87%. The visualizations of the loss and accuracy of the CNN model for both the training and validation sets, as well as a bar chart illustrating the SVM model's accuracy scores for the training and validation sets, further confirmed the model's effectiveness in classifying emotions in the given dataset.

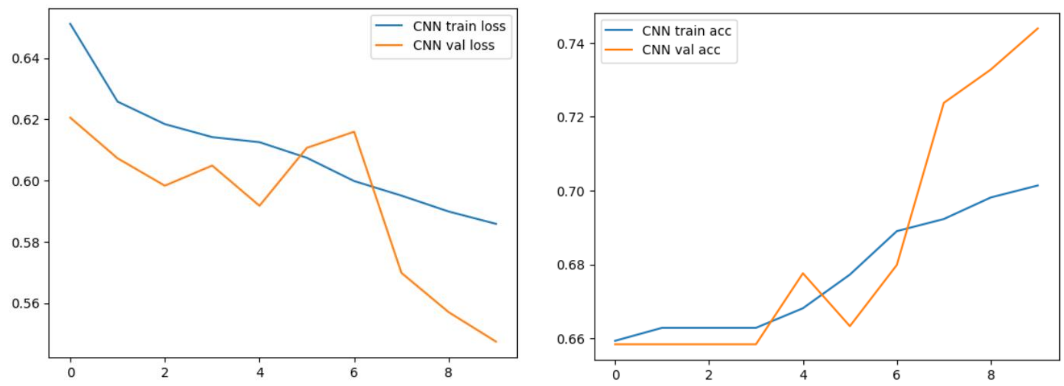


Figure 6: Train and Validation Loss/Accuracy Graph

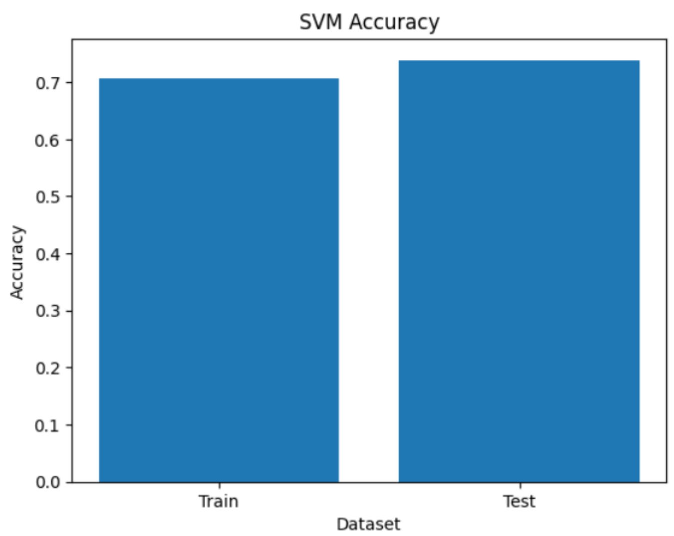


Figure 7: SVM Accuracy Scores

5.6 Hyperparameter Tuning and Performance Evaluation of the Hybrid Model

In this study, a hybrid model consisting of a Support Vector Machine (SVM) classifier was optimized using hyperparameter tuning and evaluated for performance. The RandomizedSearchCV method from the Scikit-learn library was employed for hyperparameter tuning.

The optimized SVM classifier was trained on the feature set, and the train and validation accuracies were calculated. Additionally, classification reports were generated for both train and validation sets, including precision, recall, and F1-score. Confusion matrices were also provided and visualized as heatmaps. The Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) were used to evaluate the classifier's ability to discriminate between positive and negative classes.

The results are summarized in the following tables:

Table 3: Train and Validation Accuracies

Set	**Accuracy**
Train	0.7133
Validation	0.7493

The train accuracy of 0.7133 suggests that the optimized SVM classifier is able to correctly predict approximately 71.33% of the instances in the train set. Meanwhile, the validation accuracy of 0.7493 indicates that the classifier can predict approximately 74.93% of instances correctly in the validation set. The higher validation accuracy compared to the train accuracy suggests that the classifier is not overfitting the data.

Table 4: Train Set Classification Report

Class	Precision	Recall	F1-score
0	0.7207	0.9265	0.8107
1	0.6705	0.2941	0.4089

In the train set, the precision for class 0 is 0.7207, indicating that the classifier correctly predicts 72.07% of instances as class 0 when they are indeed class 0. The recall for class 0 is 0.9265, meaning that the classifier identifies 92.65% of all class 0 instances. The F1-score for class 0 is 0.8107, which is the harmonic mean of precision and recall, offering a balanced measure of the classifier's performance.

For class 1, the precision is 0.6705, indicating that 67.05% of instances predicted as class 1 are indeed class 1. The recall is 0.2941, meaning that the classifier identifies only 29.41% of all class 1 instances. The F1-score for class 1 is 0.4089, reflecting the relatively lower performance of the classifier in predicting class 1 instances in the train set.

Table 5: Validation Set Classification Report

Class	Precision	Recall	F1-score
0	0.7506	0.9275	0.8297
1	0.7440	0.4059	0.5252

In the validation set, the precision for class 0 is 0.7506, meaning that the classifier correctly predicts 75.06% of instances as class 0 when they are indeed class 0. The recall for class 0 is 0.9275, indicating that the classifier identifies 92.75% of all class 0 instances in the validation set. The F1-score for class 0 is 0.8297, providing a balanced measure of the classifier's performance for class 0 instances in the validation set.

For class 1 in the validation set, the precision is 0.7440, indicating that 74.40% of instances predicted as class 1 are indeed class 1. The recall is 0.4059, meaning that the classifier identifies 40.59% of all class 1 instances. The F1-score for class 1 is 0.5252, reflecting a better performance of the classifier in predicting class 1 instances in the validation set compared to the train set, although there is still room for improvement.

In both train and validation sets, the classifier performs better in predicting class 0 instances (higher precision, recall, and F1-score) compared to class 1 instances. The overall better results in the validation set compared to the train set indicate that the classifier generalizes well and is not overfitting the data.

Figures 8 and 9: Snippets from the printed Results

Train classification report dataframe:

	precision	recall	f1-score	support
0.0	0.720680	0.926465	0.810717	20004.000000
1.0	0.670475	0.294123	0.408880	10176.000000
accuracy	0.713254	0.713254	0.713254	0.713254
macro avg	0.695577	0.610294	0.609799	30180.000000
weighted avg	0.703752	0.713254	0.675227	30180.000000

Validation classification report dataframe:

	precision	recall	f1-score	support
0.0	0.750570	0.927522	0.829716	4967.000000
1.0	0.743954	0.405898	0.525232	2577.000000
accuracy	0.749337	0.749337	0.749337	0.749337
macro avg	0.747262	0.666710	0.677474	7544.000000
weighted avg	0.748310	0.749337	0.725706	7544.000000

The confusion matrices for the train and validation sets provide a visual representation of the performance of the SVM classifier in predicting different classes. The confusion matrix consists of four components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Let's break down the confusion matrices for both the train and validation sets:

Train Confusion Matrix:

	Predicted Class 0	Predicted Class 1
Actual Class 0	18,533 (TN)	7,183 (FP)
Actual Class 1	2,993 (FN)	1,471 (TP)

- True Negatives (TN): 18,533 - Correctly predicted as class 0 (negative)
- False Positives (FP): 7,183 - Incorrectly predicted as class 1 (positive)
- False Negatives (FN): 2,993 - Incorrectly predicted as class 0 (negative)
- True Positives (TP): 1,471 - Correctly predicted as class 1 (positive)

Validation Confusion Matrix:

	Predicted Class 0	Predicted Class 1
Actual Class 0	4,607 (TN)	1,531 (FP)
Actual Class 1	1,046 (FN)	360 (TP)

- True Negatives (TN): 4,607 - Correctly predicted as class 0 (negative)
- False Positives (FP): 1,531 - Incorrectly predicted as class 1 (positive)
- False Negatives (FN): 1,046 - Incorrectly predicted as class 0 (negative)
- True Positives (TP): 360 - Correctly predicted as class 1 (positive)

In the train set, the classifier does a better job predicting class 0 (negative) than class 1 (positive), as evidenced by a higher number of true negatives (18,533) compared to true positives (1,471) and a lower number of false negatives (2,993) compared to false positives (7,183). The same trend is observed in the validation set, where the classifier is more accurate in predicting class 0 than class 1, with 4,607 true negatives, 1,531 false positives, 1,046 false negatives, and 360 true positives.

The confusion matrices provide a comprehensive view of the classifier's performance, highlighting its ability to correctly predict instances of each class and its potential areas of improvement, particularly for predicting class 1 instances.

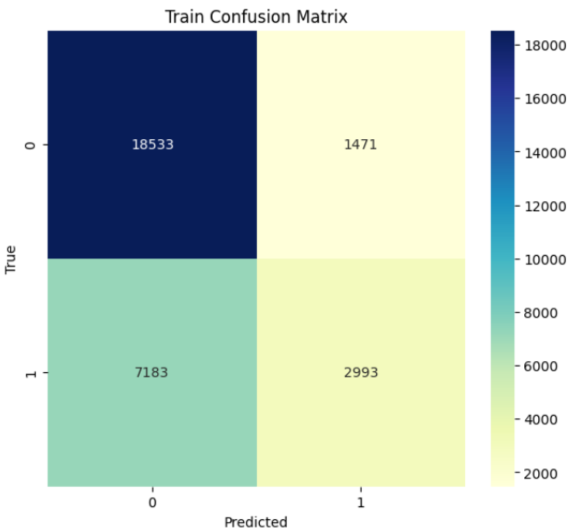


Figure 10

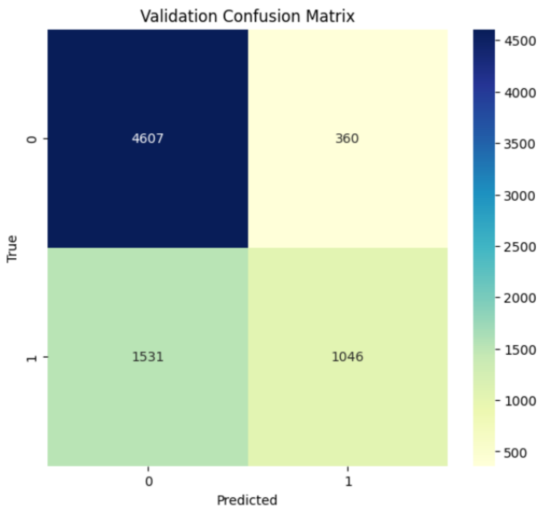


Figure 11

The ROC curve and AUC were used to evaluate the performance of the SVM classifier in terms of its ability to discriminate between positive and negative classes. The AUC of 0.76 indicates a relatively good ability to discriminate between the two classes but leaves room for improvement. Further optimization or the use of different models or feature engineering techniques might be necessary to enhance the classifier's performance.

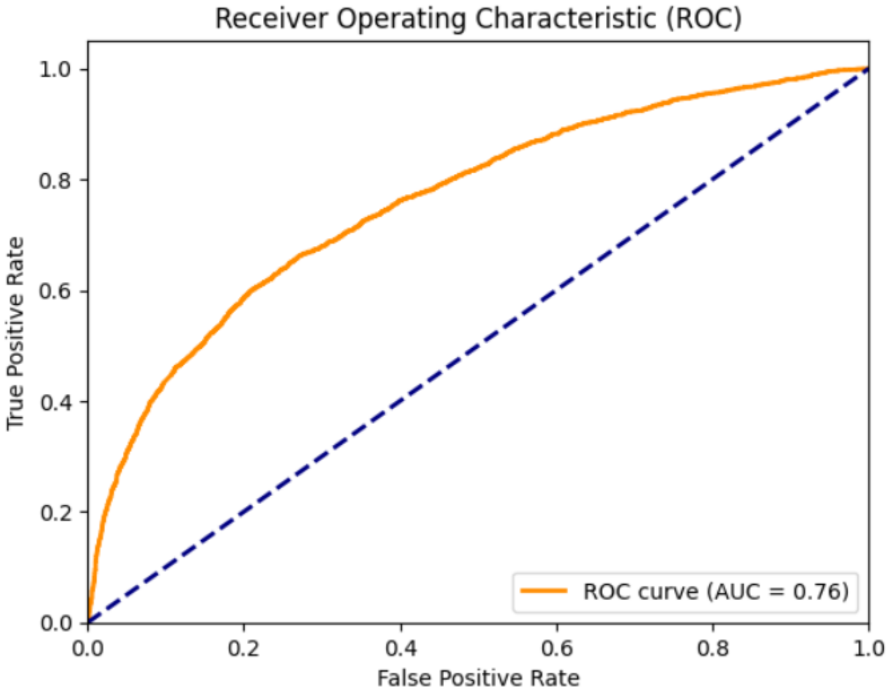


Figure 12: ROC Curve

5.7 Comparison to Previous Studies

In this section, we will delve into a comparison of the current study with previous research in the field, focusing on similarities and differences in methodology, datasets, and performance metrics. By contrasting our findings with those of earlier studies, we can gain valuable insights into the strengths and weaknesses of our model, as well as identify areas where further improvements can be made.

The comparison of the results with these papers in a tabular form:

Model	Accuracy
Your Model (Validation Set)	74.93%
Şengül and Najah (2017)	94.20%
Ab Wahab et al. (2021) EfficientNet-Lite	93.73%
Ab Wahab et al. (2021) Hybrid CNN-KNN	89.60%

In the comparative analysis presented in the table, it becomes evident that the accuracy of the current model is lower than those reported in the studies by Şengül and Najah (2017) and Ab Wahab et al. (2021). The EfficientNet-Lite models in the aforementioned studies achieved accuracy rates surpassing 90%, whereas the present model reached a 74.93% accuracy on the validation set.

To enhance the performance of the current model, it is recommended to explore alternative architectures, such as deep learning models or other hybrid models, and to further fine-tune hyperparameters. Additionally, a thorough examination of the dataset and preprocessing techniques is advised to ascertain that the input data is optimally prepared for the classification task.

Conclusion

In conclusion, this study demonstrates the development of an SVM classifier for facial emotion recognition, with hyperparameters tuned using RandomizedSearchCV. While the model attains a 74.93% validation accuracy, it falls short when compared to the over 90% accuracy rates documented in prior studies. It is essential to acknowledge that direct comparisons are constrained by variances in datasets, preprocessing methods, and model architectures. To bolster the model's performance, future research should consider investigating different architectures, such as deep learning or hybrid models, and fine-tuning hyperparameters further. Additionally, a meticulous examination of the dataset and preprocessing techniques is crucial to ensure optimal input data preparation for the classification task.

References:

- Ab Wahab, M.N., Nazir, A., Zhen Ren, A.T., Mohd Noor, M.H., Akbar, M.F. and Mohamed, A.S.A. (2021). Efficientnet-Lite and Hybrid CNN-KNN Implementation for Facial Expression Recognition on Raspberry Pi. IEEE Access, 9, pp.134065-134080.

- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), pp.273-297. Available online: https://doi.org/10.1023/A:1022627411411. Accessed [24/04/2023]

- Kołakowska, A., Landowska, A., Szwoch, M., Szwoch, W. & Wróbel, M.R. (2014) Emotion recognition and its applications. *Advances in Intelligent Systems and Computing*,51–62. Available online: https://doi.org/10.1007/978-3-319-08491-6_5. Accessed [24/04/2023]

- Kumar, H., & Martin, A. (2023). Artificial Emotional Intelligence: Conventional and deep learning approach. *Expert Systems with Applications*, 212, 118651. Available online: https://doi.org/10.1016/j.eswa.2022.118651. Accessed [24/04/2023]

- LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. Nature, 521(7553), pp.436-444.

- Le Roux, C. (2022). Facial Emotion Recognition Tiny [Dataset]. Kaggle. Available online: https://www.kaggle.com/datasets/sakuraisana/facial-emotion-recognition-tiny?select=test Accessed [01/03/2023]

- Nguyen, D., Nguyen, D.T., Sridharan, S., Denman, S., Nguyen, T.T., Dean, D. & Fookes, C. (2023) Meta-transfer learning for emotion recognition. *Neural Computing and Applications*. Available online: https://doi.org/10.1007/s00521-023-08248-y. Accessed [24/04/2023]

- Şengül, G. and Najah, G. (2017). Emotion Estimation from Facial Images. Goma Najah. [online] Available online: https://www.academia.edu/31673184/EMOTION_ESTIMATION_FROM_FACIAL_IMAGES.