Mini Project Report on

Face Emotion Detection using Deep Learning

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

and

COMPUTER SCIENCE & ENGINEERING

Submitted By:

StudentName:
Aashish Subedi

UniversityRollNo:
2118052

Department of Computer Science and Engineering

Graphic Era (Hill University)

Dehradun , Uttarakhand

January 2024

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled**"Face Emotion Detection inPython"**in partial fulfillment of the

requirements for the award of the Degree of Bachelor of Technology in Computer Science

and Engineeringof the Graphic Era (Hill University), Dehradun shall be carried

out by the under the mentorship of **MS Himadri Vaidhya,AssistantProfessor**,

Department of Computer Science and Engineering, Graphic Era Hill University,

(Dehradun)**.**

Name:	**UniversityRollNo.**	**Signature**
Aashish Subedi	2118052	

Table of Contents

1. Introduction

2. Dataset

3. Python Libraries used

4. Methodology

5. Training

6. Evaluation

8. Conclusion

INTRODUCTION

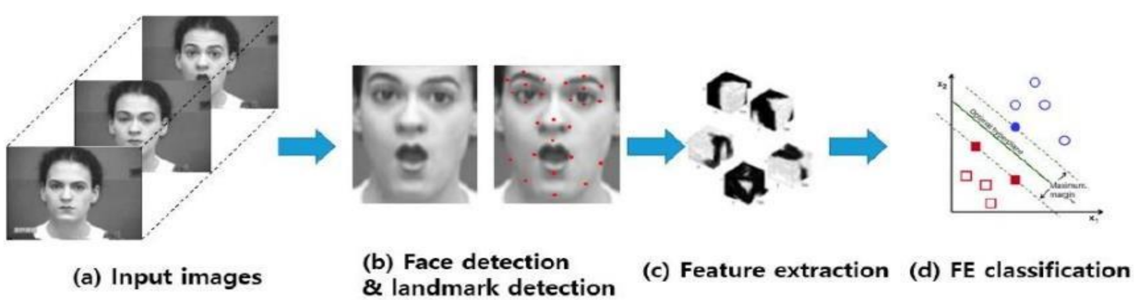
Abstract:

This report explores the implementation of a face emotion detection system using computer vision (CV2), NumPy, Keras, and TensorFlow. Emotion detection from facial expressions has numerous applications, ranging from human-computer interaction to sentiment analysis. The integration of these powerful Python libraries enables the creation of an accurate and efficient emotion recognition model.

Humans primarily transmit their own emotional states to others through their facial expressions. Numerous investigations have shown that more than 50% our facial expressions directly communicate how we are feeling. It represents a ten times greater percentage than the inflection of spoken words conveys feelings. Since The Networked Age of Technology, which we are currently living in, it is become more and more important in daily life to have intelligent monitoring. For instance, cameras and helper robots must comprehend how human emotions function. For humans, expression recognition is fairly simple to guess, but is a very challenging assignment for even the most intelligent AI technologies. The obstacles associated with automatic emotion detection range from the categorization of emotions to a more extensive investigation by psychologists

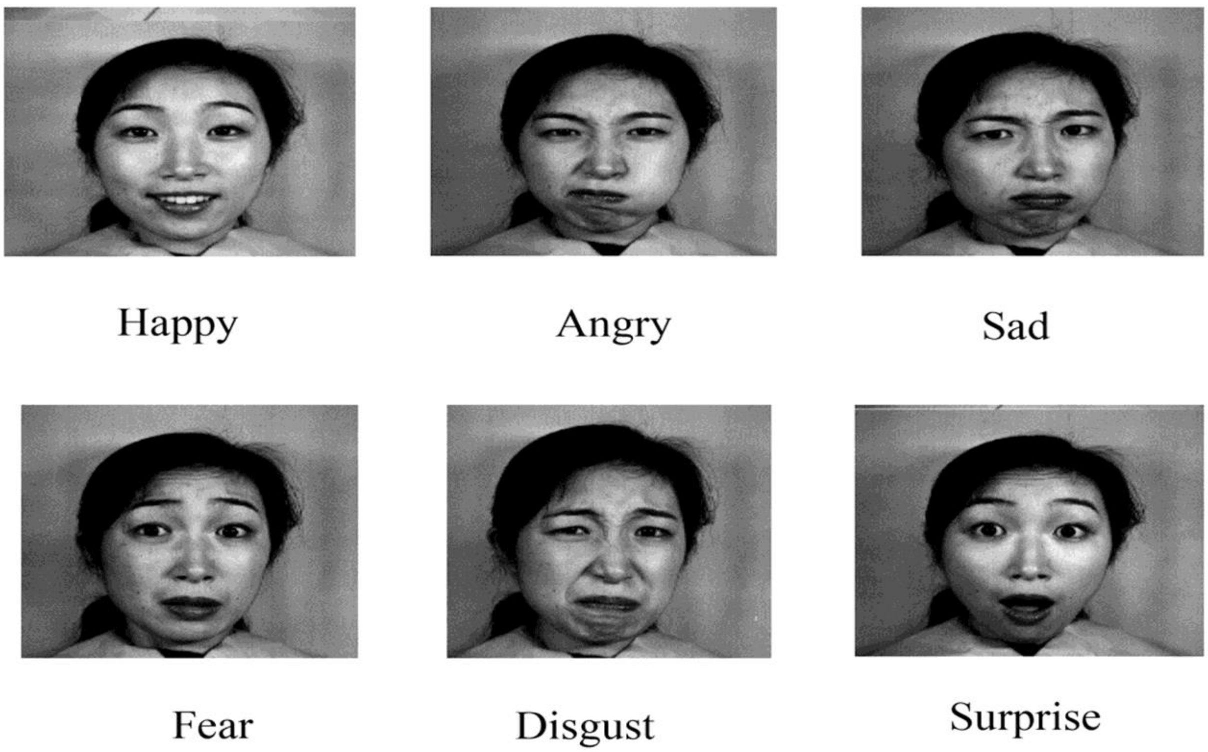
and their partnership with scientists.

Procedure of FER.



DATASET

The dataset contain 35,685 examples of 48x48 pixel gray scale images of faces divided into train and test dataset. Images are categorized based on the emotion shown in the facial expressions (happiness, neutral, sadness, anger, surprise, disgust, fear).



Notwithstanding these obstacles, the CK+ database has been widely utilized in the creation and assessment of machine learning models for facial expression identification, and has made major contributions to the field's advancement. The CK+ database has been used by researchers to investigate a variety of topics, including the impact of facial landmarks on emotion recognition accuracy, the efficacy of deep learning approaches for facial expression analysis, and the impact of cross-cultural differences on facial expression perception.

Python Libraries used

NumPy: Working with arrays and matrices is made possible via the free source Python package known as Numerical Python. NumPy can be used to transform photos into NumPy arrays so that matrix multiplications and other CNN operations can be carried out quickly. CNN inputs are numerical arrays.

TensorFlow: TensorFlow is a complete open-source machine learning platform. TensorFlow is a powerful system for controlling all parts of a machine learning system; however, this lecture concentrates on developing and training machine learning models using a specific TensorFlow API.

Keras: The neural network Application Programming Interfaces (APIs) TensorFlow and Python's Keras are closely related. TensorFlow is the software used to build machine learning models. Using Keras models, a neural network can be quickly defined, and TensorFlow will build the network for you.

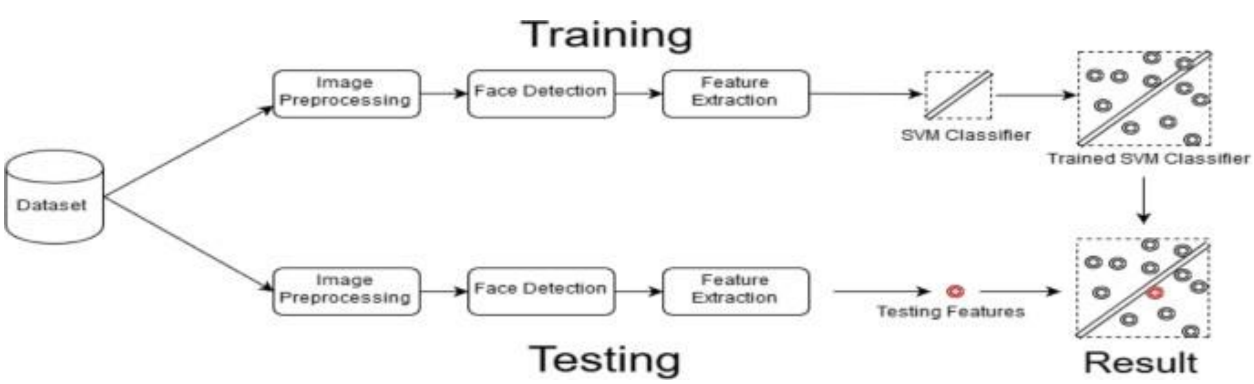
Open CV: An open-source library for image, machine learning, and CV is called OpenCV. OpenCV can process images and videos to recognize objects, faces, and handwriting. OpenCV can handle array structures for analysis when it is used with a library like Numpy. These array structures are subjected to mathematical procedures for pattern recognition.

Matplotlib: A Python tool called Matplotlib is used to produce excellent 2D and 3D data visualizations. It is made to be simple to use and very adaptable, and it offers a variety of tools for making plots, charts, histograms, and other kinds of visualizations. From straightforward line plots to intricate heatmaps and three-dimensional surface plots, Matplotlib may be used to produce a wide range of visualizations.

METHODOLOGY

Support Vector Machine (SVM)

SVMs are a form of machine learning technique that is used for classification and regression problems. By maximizing the margin between the decision border and the data points, SVMs discover the ideal hyperplane that optimally divides the data into multiple classes. By the application of kernel functions that translate the data into a higher-dimensional feature space, SVMs may deal with both linearly separable and non-linearly separable datasets. SVMs have a number of advantages, including the capacity to handle high-dimensional and noisy datasets. They are also suitable for binary and multi-class classification jobs. SVMs are also resistant to overfitting since they optimize the margin between the decision border and the data points.SVMs, on the other hand, can be sensitive to the kernel function and hyperparameter selection, which can be difficult to modify. Also, for big datasets, SVMs might be computationally costly. Image recognition, text categorization, and bioinformatics are just a few of the numerous practical uses for SVMs. SVM versions and extensions are still being investigated by researchers



in order to increase their performance and usefulness in real-world circumstances.

Kernel functions

Kernel functions are a critical component of Support Vector Machines (SVMs) in facial expression recognition (FER) tasks [2]. The selection of the kernel function plays a vital role in determining the SVM's ability to classify data accurately. There are three primary kernel functions used in SVM for FER: linear, polynomial, and radial basis function (RBF).

The linear kernel function is the simplest and most commonly used kernel function. It maps the input data into a higher-dimensional space using a linear function, which is suitable for linearly separable datasets. This kernel function is useful when the features are linearly related to the output, and there is no need for complex decision boundaries.

The polynomial kernel function maps the input data into a higher-dimensional space using a polynomial function. This kernel function is suitable for datasets with non-linearly separable data. It works by introducing additional polynomial features, which can increase the accuracy of the classification. However, selecting the right degree of polynomial can be a challenging task.

The radial basis function (RBF) kernel function is a popular choice for FER tasks as it can handle non-linearly separable data. It maps the input data into an infinite-dimensional feature space using Gaussian functions, which captures the similarities between the data points. This kernel function is useful when the data is not linearly separable, and there is a need for complex decision boundaries.

In summary, the choice of kernel function in SVM for FER tasks is crucial in achieving high classification accuracy. The linear kernel function is useful for linearly separable datasets, while the polynomial kernel function is suitable for non-linearly separable datasets with moderate complexity. The RBF kernel function is the most commonly used kernel function in FER tasks asit can handle complex, non-linearly separable data.

Data Pre-processing

Data preprocessing is a crucial step in preparing data for Support Vector Machines (SVMs). SVMs require high-quality input data, and data preprocessing techniques such as feature extraction and feature scaling can help improve the model's accuracy. Here's an overview of thesetechniques:

Feature Extraction: Feature extraction involves selecting and transforming relevant data into a set of informative features that can be used for classification. Feature extraction can help reduce the dimensionality of the data, which can improve the model's accuracy and reduce the computational burden.

For example, in facial expression recognition tasks, feature extraction can involve detecting and extracting the facial landmarks, such as the eyes, nose, and mouth, and representing them as numerical features.

Feature Scaling: Feature scaling involves normalizing the features to a similar range, which can help improve the model's performance. Different features may have different ranges, which can make some features more influential than others. Normalizing the features can ensure that all features contribute equally to the model's prediction.

There are various ways to scale features, including Z-score normalization and Min-Max scaling. Scaling the features to a range between 0 and 1 is known as min-max scaling. By scaling the features to have a mean of 0 and a standard deviation of 1, Z-score normalization achieves this.

TRAINING

Training SVMs:

Training Support Vector Machines (SVMs) for facial expression recognition (FER) involves several steps, including hyperparameter tuning and model selection [2]. Hyperparameter tuning involves selecting the optimal values for the SVM's hyperparameters, which can significantly impact the model's accuracy. Model selection involves evaluating and selecting the best-performing SVM model. Here's an overview of these steps:

Hyperparameter Tuning: Hyperparameters are parameters that are not learned by the SVM but are set prior to training. Examples of hyperparameters in SVMs include the kernel type, regularization parameter, and kernel-specific parameters. Selecting the optimal values for these hyperparameters can be challenging and requires careful experimentation.

Hyperparameter tuning involves evaluating the SVM's performance on a validation set for different combinations of hyperparameters. The hyperparameters that yield the best performance are selected for the final model. Techniques such as grid search and random search can be used forhyperparameter tuning.

Model Selection: Model selection involves evaluating and selecting the best-performing SVM model. This is typically done by comparing the SVM's performance on a separate test set for different models with different hyperparameters.To ensure a fair comparison, the same validation and test sets should be used for all models. Additionally, cross-validation can be used to evaluate the SVM's performance on different subsetsof the data.

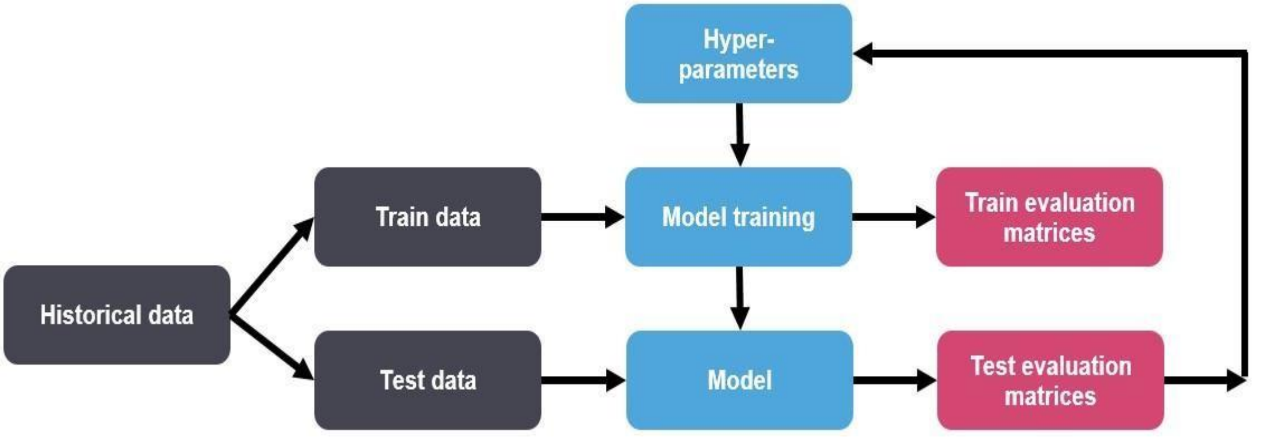


Fig .hyperparameter tunning.

EVALUATION

The evaluation phase is a critical component in assessing the performance and reliability of the face emotion detection model. This stage involves testing the trained model on a separate dataset (commonly referred to as the test set) to measure its ability to accurately recognize and classify facial expressions. Several metrics are employed to gauge the model's effectiveness in capturing diverse emotions.

Key Aspects:

a. Testing Metrics:

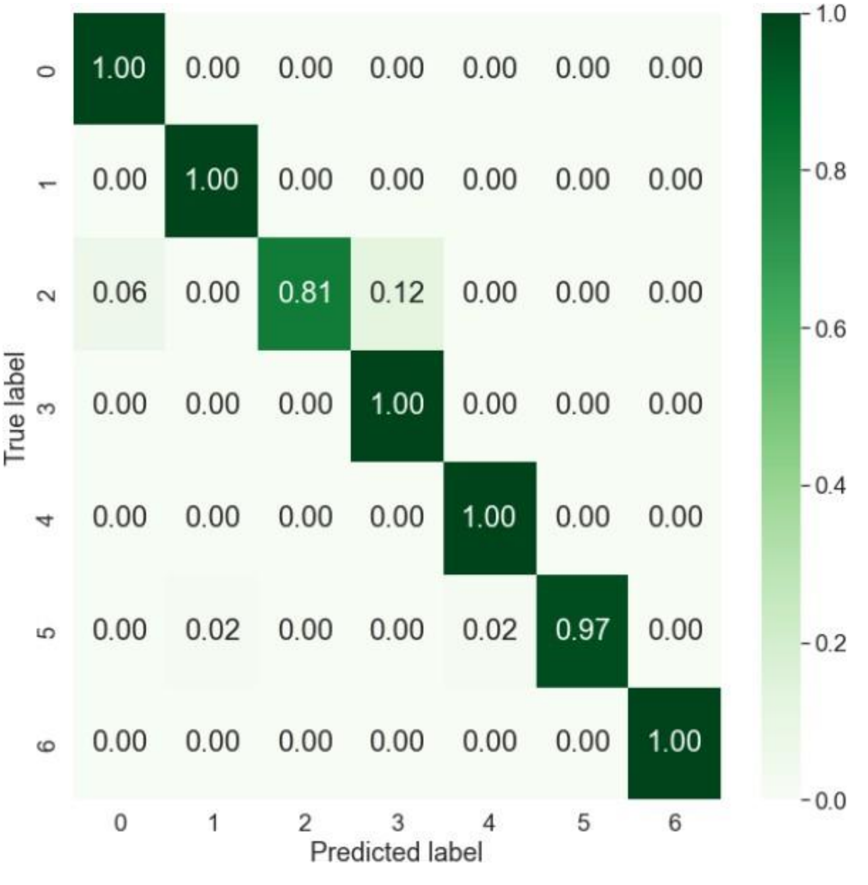
Accuracy: The overall correctness of the model in predicting emotions.

Precision: The ratio of correctly predicted positive observations to the total predicted positives, indicating the model's ability to avoid false positives.

Recall (Sensitivity): The ratio of correctly predicted positive observations to the total actual positives, representing the model's ability to capture all instances of a particular emotion.
F1 Score: The harmonic mean of precision and recall, providing a balanced measure between the two.

b. Confusion Matrix:

A visual representation of the model's performance, showcasing the number of true positive, true negative, false positive, and false negative predictions.



c. Cross-Validation:

To ensure the model's robustness, techniques like k-fold cross-validation can be applied. This involves dividing the dataset into multiple subsets, training the model on different combinations of these subsets, and averaging the evaluation metrics.

d. Overfitting and Generalization:

Analyzing the model's behavior on both the training and test sets helps identify overfitting (performing well on training data but poorly on unseen data) and ensures the model generalizes well to new data.

e. Fine-Tuning:

Based on evaluation results, fine-tuning the model may be necessary. Adjustments to hyperparameters, architecture, or the dataset may be made to enhance performance.

f. Real-World Performance:

Consideration of the model's performance in real-world scenarios, such as handling noisy or varied environments, is crucial for practical applications.

g. Future Improvements:

Identification of areas for improvement and potential research directions to enhance the model's accuracy and applicability.

The evaluation phase provides insights into the strengths and weaknesses of the developed face emotion detection model. By employing rigorous metrics and testing procedures, developers can ensure the model's reliability and effectiveness in various contexts.

Conclusion

This research study provides a deep learning approach that makes use of a unique CNN architecture and SVM model to recognize facial expressions of emotion. The FER system is expanded using the CNN architecture to boost performance on a specific database-based system. To generalize the CNN and SVM-based model for testing, the structure is selected and trained differently for each dataset.

During the testing process, the test samples is anticipated and contrasted with the genuine label. Applying the CNN model to the FER13 dataset yields an overall accuracy of 87.78%, just under the CK+ accuracy of 99%. With the FER13 dataset, we apply the SVM model and obtain an overall accuracy of 38%, which is lower than the CK+ accuracy of 97%. It is evident from comparisons to cutting-edge results that the generated custom CNN offers good accuracy while being less sophisticated in terms of network depth and parameters. However, there is room to experiment with the standard CNN design, deeper CNN networks, or other methods to increase accuracy.

Future work may employ unsupervised pre-training transfer learning methods to further cut down on recognition failures. Pre-processing and feature extraction techniques can be applied prior to training models. Transfer Learning can also be tested using these datasets. The CK+ Dataset should be balanced because it has few images and produces unbalanced test results. A cutting-edge study that can be reproduced has shown encouraging results for FER based on Facial Action Units (AUs). Advanced Models like DBN, VAE, and GANs can be utilized to increase accuracy.