Facial Emotion Recognition

Project Report

Submitted by

Keerthana S
B. Tech (3rd year, Mechanical Engineering)
Indian Institute of Technology Guwahati
E-mail: s.keerthana@iitg.ac.in

Under the guidance of

Dr. Deepak Mishra
Professor

Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram, India

July 2023

DECLARATION

I, **Keerthana S (Roll No: IS00140)**, hereby declare that, this report entitled **"Facial Emotion Recognition "** submitted to **Indian Institute of Space Science and Technology** towards the partial requirement of **Summer Internship 2023** in **Avionics**, is an original work carried out by me under the supervision of **Deepak Mishra** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold academic ethics and honesty. Whenever a piece of external information or statement or result is used then, that has been duly acknowledged and cited.

Thiruvananthapuram - 695 547 **Keerthana S**
August 2023

CERTIFICATE

This is to certify that the work contained in this project report entitled **"Facial Emotion Recognition"** submitted by **Keerthana S** (**Roll No: IS00140**) to Indian Institute of Space Sciences and Technology, Thiruvananthapuram towards the partial requirement of **Summer Internship 2023** in **Department of Avionics** has been carried out by him under my supervision. Throughout his tenure, he has exhibited a strong work ethic and a commitment to meeting and exceeding expectations. We wish him the best of luck in all of his future endeavors.

Thiruvananthapuram - 695 547
August 2023

Dr. Deepak Mishra
Project Supervisor

ACKNOWLEDGEMENT

Professor Deepak Mishra, my faculty supervisor, and mentor, for his invaluable guidance, support, and feedback that have helped shape the development of this project. His expertise and mentorship have been crucial in my learning and growth throughout this journey.

I am deeply thankful to all the interns at IIST, and their collaborative efforts, valuable insights, and willingness to share knowledge have contributed significantly to my learning and enriched my understanding of the subject matter. And thanks to my family and friends for their unwavering belief in me and constant support throughout this journey. Their encouragement and love have been a source of strength to me every time.

I want to give recognition to the academic institution for granting me the chance to work on this project and for creating a space that encourages education and development.

Thiruvananthapuram - 695 547 **[Keerthana S]**

ABSTRACT

*T*he research work focuses on Facial Emotion Recognition using the Supervised learning approach. The dataset used in this project is FER2013 and CK+ dataset. In FER2013, there were 28709 training images and 3589 testing images. In the CK+ dataset, there are 981 images. The dataset contains seven classified images of anger, disgust, fear, happy, neutral, sad, and surprise. Images contain 48x48 grayscale faces. The model used here is a novel CNN architecture, where neural-network had trained to detect facial emotions, and based on this model's weights, an Emotion recognition model had trained to detect emotions in the face. During the implementation, I learned about many techniques, such as HOG (Histogram of Oriented Gradients), LBP (Local Binary Pattern), and Facial Landmarks. I achieved an accuracy of 0.89 for my model(without using any of the mentioned techniques). The pre-trained model had accuracies like 0.8571(VGG), 0.5461(DenseNet169), and 0.8571(ResNet50) for the best epochs.

Contents

Chapter 1

Introduction

1.1 MultiModal Data Fusion

Multimodal data fusion in supervised learning refers to the integration of information from multiple sources or modalities to improve the performance and robustness of machine learning models. This approach has gained significant attention in recent years due to its potential to leverage complementary information from different data types, leading to better generalization, enhanced feature representation, and improved model accuracy. In traditional supervised learning, the models are trained on a single modality, typically represented by a feature vector derived from one type of data, such as images, text, or numerical data. For instance, in image classification, a Convolutional Neural Network (CNN) might be trained on raw pixel values as input. The limitation of traditional supervised learning lies in its inability to efficiently utilize information from multiple modalities. While some methods can handle multiple features, they often treat different modalities independently, ignoring potential correlations or dependencies between them. As a result, the models may not fully exploit all available information, leading to suboptimal performance and reduced accuracy.

1.2 Emotion Recognition

An image-processing method of emotion recognition from facial expressions includes identifying emotions and mainly classifying them into seven categories of images and aims to pinpoint their emotion correctly. Model Training, model Validation and Testing, Emotion Classification, and Feature Extraction are the fundamental methods used in this method, which classifies facial emotions into

happiness, sadness, disgust, anger, neutrality, fear, and surprise, where the model made here validates each feeling.

To generate accurate results, **pre-trained models used here:** VGG16, ResNet50, DenseNet169, SIFT, and Gabor filters.

The model used here is a **convolution neural network** that utilizes a series of convolutional, separable convolutional, and pooling layers to learn hierarchical features from the input images.

Chapter 2

Learning Path

2.1 Multimodal Techniques

Multimodal techniques, such as HOG and LBP, were thoroughly examined for the CK+ and FER2013 datasets. The CK+ was also reviewed with filters SIFT, and Gabor, and without feature extraction. Meanwhile, for FER2013, only HOG and LBP were used. It's important to note that in cases where pre-trained models were utilized, such as VGG16, ResNet50, and DenseNet169, filters were not employed, as the focus was on comparing the validation accuracies of the models.

2.1.1 `Histogram of Oriented Gradients(HOG)`

Histogram of Oriented Gradients (HOG) is a popular feature descriptor used in computer vision and facial emotion detection tasks. It is particularly useful for capturing local patterns and gradients in an image. In the context of facial emotion detection, HOG is employed to represent the facial texture and structure, providing valuable information about the distribution of edge orientations and intensity variations in different regions of the face.

It works in FER like:

Image Preprocessing: The facial image is preprocessed to enhance its quality and reduce variations due to lighting conditions or skin tone. Common preprocessing steps include normalization, contrast enhancement, and histogram equalization.

Gradient Computation: HOG computes the gradients (derivatives) of the image in both the horizontal and vertical directions. The gradients represent the change in intensity or color values at each pixel, indicating the presence of edges or boundaries.

Orientation Binning: The image is divided into small cells (e.g., 8x8 pixel blocks). For each cell, the gradient orientation and magnitude are calculated. The gradient orientation is quantized into bins (e.g., 9 bins covering 0 to 180 degrees), and the magnitude is used to determine the contribution of the gradient to each bin.

Cell Histograms: A histogram is created for each cell, which accumulates the gradient contributions from neighboring pixels. This histogram represents the dominant gradient orientations within the cell.

Block Normalization: To improve robustness against lighting variations and enhance the discriminative power, adjacent cells are grouped together to form blocks. Block normalization is applied to normalize the histograms within each block, reducing the impact of lighting changes and enhancing the discriminative power.

Feature Vector: The final feature vector is formed by concatenating the histograms from all the cells. This feature vector captures the distribution of oriented gradients in the facial image and represents the facial texture and structure.

2.1.2 Label Binary Pattern(LBP)

Local Binary Pattern (LBP) is a widely used feature descriptor in facial emotion recognition and computer vision tasks. It captures the local texture information in an image and has proven effective for analyzing facial patterns and expressions. LBP represents the relationship between the intensity of a central pixel and its surrounding neighbors in a compact binary format.

It works in FER like:

Image Preprocessing: The facial image is preprocessed to enhance its quality and reduce noise. Common preprocessing steps include converting the image to grayscale and performing histogram equalization to improve contrast.

Local Neighborhood Description: For each pixel in the image, a local neighborhood of neighboring pixels is defined. Typically, a circular region around the central pixel is used, and the number of neighbors (points on the circle) and the radius of the circle can be adjusted as parameters.

Thresholding and Binary Representation: The intensity value of the central pixel is compared with the intensity values of its neighboring pixels in the local neighborhood. If the intensity of a neighbor is greater than or equal to the intensity of the central pixel, a binary value of 1 is assigned; otherwise, a value of 0 is assigned. This comparison is done for all neighbors, resulting in a binary code representing the local texture pattern.

LBP Histogram: After computing the binary code for each pixel in the

image, a histogram of the LBP codes is constructed. The histogram represents the distribution of different local texture patterns in the facial image.

Feature Vector: The final feature vector is formed by concatenating the histogram bins. This feature vector encodes the texture information from different regions of the face, capturing unique patterns associated with facial expressions.

Machine Learning and Emotion Classification: The LBP feature vector is used as input to machine learning algorithms, such as support vector machines (SVM), k-nearest neighbors (KNN), or neural networks, for emotion classification. By learning from labeled samples, the model can associate specific LBP patterns with different emotions, enabling accurate emotion recognition from facial images.

2.1.3 Scale-Invariant Feature Transform (SIFT)

A feature descriptor commonly used in computer vision, including facial emotion recognition tasks. SIFT is designed to extract and represent key features from images that are invariant to changes in scale, rotation, and illumination, making it robust to image variations.

It works in FER like:

Keypoint Detection: SIFT first identifies keypoint locations in the facial image that correspond to distinctive features, such as corners, edges, or blobs. These keypoints are detected using a difference of Gaussian (DoG) algorithm to identify areas of significant intensity variations across different scales.

Orientation Assignment: For each keypoint, SIFT assigns an orientation based on the local gradient directions in the image. This step ensures that the extracted features are robust to image rotations.

Local Image Descriptors: A local image patch centered at each keypoint is extracted and described using a histogram of gradient orientations. The orientation of the gradient is quantized into bins, and the magnitude of the gradient contributes to the corresponding bin.

Keypoint Description: The histogram of gradient orientations from the local image patch is used to create a feature vector representing the keypoint. This feature vector captures the unique local texture information around the keypoint, providing a distinctive representation of the facial feature.

Feature Matching and Emotion Classification: In facial emotion recognition, SIFT features extracted from facial images can be matched with features from a database of labeled emotion samples. Machine learning algorithms, such as k-nearest neighbors (KNN) or support vector machines (SVM), can be used to classify emotions based on the matched features.

2.1.4 Gabor Filters

A type of linear filter used in image processing and computer vision, particularly for analyzing texture and spatial frequency patterns. They are inspired by the human visual system's response to different frequencies and orientations and have been widely used in facial emotion recognition tasks due to their ability to capture distinctive facial texture features.

It works in FER like:

Gabor Filter Construction: A Gabor filter is a sinusoidal plane wave modulated by a Gaussian envelope. It is defined by two main parameters: the frequency (which controls the number of oscillations) and the orientation (which determines the angle of the oscillations).

Filter Bank Generation: A set of Gabor filters with different frequencies and orientations is created. These filters act as feature detectors, each being sensitive to specific spatial frequency and orientation patterns in the input image.

Convolution: The Gabor filters are convolved with the facial image to obtain the Gabor response or feature maps. This process involves sliding the filter bank across the image and computing the response at each position, capturing the local texture information at different scales and orientations.

Feature Extraction: The responses obtained from the convolution process represent the magnitude of the detected texture features at various locations in the facial image. These Gabor response maps can be used as feature representations for the image.

Emotion Classification: The Gabor response maps are used as input to machine learning algorithms, such as support vector machines (SVM) or neural networks, to classify emotions based on the extracted texture features. By learning from labeled samples, the model can associate specific Gabor response patterns with different emotions, enabling accurate facial emotion recognition.

2.2 The model

This model is a deep neural network designed for facial emotion recognition. It takes 48x48 pixel facial images with three color channels as input and applies a series of convolutional, batch normalization, and separable convolutional layers to learn hierarchical features. The model utilizes skip connections and concatenations to capture important information at different scales. Finally, it employs global average pooling followed by dense layers with dropout for classification into seven emotion categories. This architecture enables the model to automatically learn and

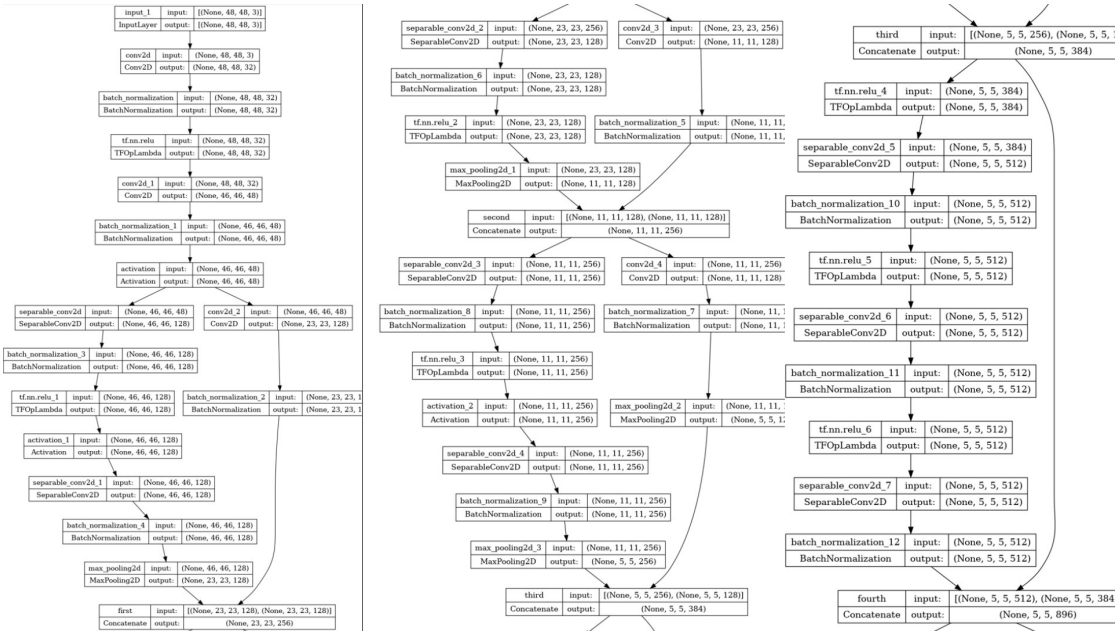

Figure 2.1: model upto 4th concatenation

distinguish facial patterns associated with different emotions, making it suitable for emotion recognition tasks.

2.3 How did I started?

After examining multiple pre-trained models and observing those who created their models before, I noticed a difference in the output shapes. For those developing custom models, they change the output shape from (None, 48, 48, 3) to (None, 7, 7, 256), while pre-trained models transform it to (None, 1, 1, 512). In response, I designed a model that maintains the output shape of (None, 1, 1, 256). This approach allows the model to retain higher-resolution information, which might benefit facial emotion recognition tasks.

2.3.1 `How many layers are in my model?`

some layer types have multiple instances in the model

1. **Input Layer:** 1 layer
2. **Conv2D Layers:** 3 layers
3. **BatchNormalization Layers:** 14 layers
4. **Activation Layers (ReLU):** 11 layers

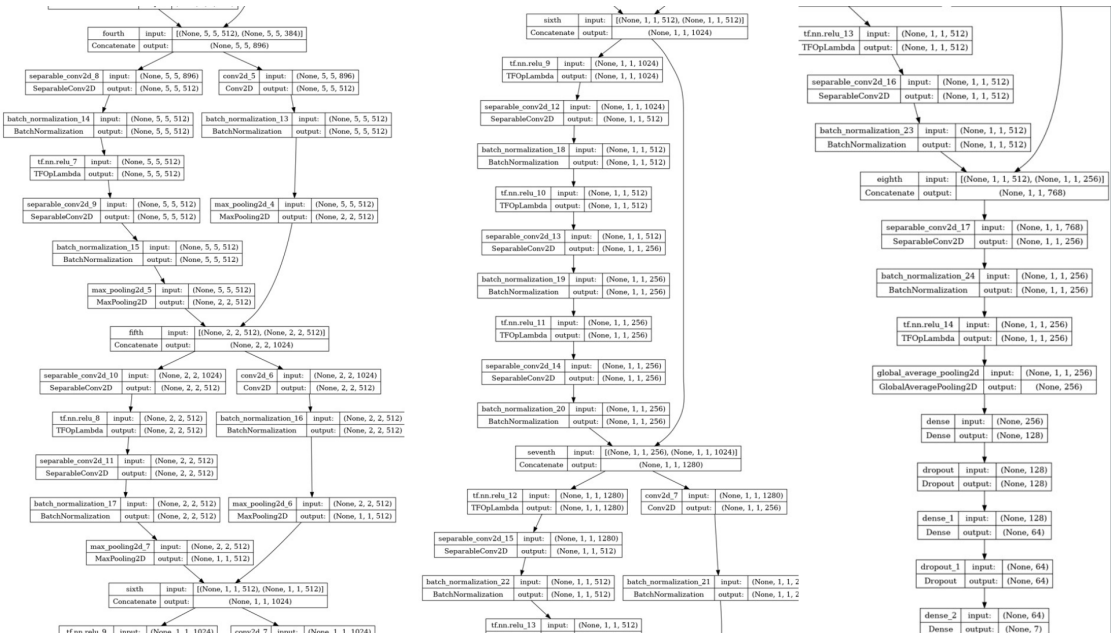

Figure 2.2: model upto 8th concatenation

```
==
Total params: 5,913,255
Trainable params: 5,897,223
Non-trainable params: 16,032
```

Figure 2.3: number of parameters in my model

5. **SeparableConv2D Layers:** 5 layers
6. **MaxPooling2D Layers:** 5 layers
7. **Concatenate Layers:** 6 layers
8. **GlobalAveragePooling2D Layer:** 1 layer
9. **Dense (Fully Connected) Layers:** 3 layers
10. **Dropout Layers:** 2 layers
11. **Output Layer:** 1 layer

2.3.2 What are the significance of these number of layers in this model?

Feature Learning: The multiple Conv2D and SeparableConv2D layers allow the model to learn a hierarchy of features, starting from low-level features like edges and textures and progressing to higher-level, more abstract features related to facial expressions and emotions.

Non-Linearity and Stabilization: The ReLU activation layers introduce non-linearity, which is essential for the model to learn complex relationships between the input data and the target emotions. BatchNormalization layers help stabilize and accelerate the training process, making it easier for the model to converge to a good solution.

Pooling and Concatenation: The MaxPooling2D layers perform downsampling, reducing the spatial dimensions of the feature maps and capturing the most important information. Concatenate layers merge feature maps from different parts of the network, allowing the model to learn multi-scale representations and combine information from different layers effectively.

Global Average Pooling: The GlobalAveragePooling2D layer condenses the feature maps to a single value per channel, providing a global summary of the learned features. This reduces the model's spatial dimensions and allows it to focus on the most relevant information.

Dense (Fully Connected) Layers: The fully connected layers process the global feature vector and learn higher-level representations that are used for emotion classification. The dropout layers help prevent overfitting and improve generalization.

Overall, the number and types of layers in the model have been carefully chosen to create an architecture that can effectively extract relevant features from facial images and make accurate predictions about the emotions expressed in those images. The model's significance lies in its ability to learn from data, generalize to unseen examples, and capture the complex patterns associated with different emotions, making it suitable for facial emotion recognition tasks.

2.3.3 why concatenation? is it making something big deal here?

it allows the model to combine information from different parts of the network, leading to improved feature representations and enhanced performance in facial emotion recognition. By merging feature maps from various convolutional layers, the model can capture multi-scale information and learn more comprehensive and

discriminative features.

Multi-Scale Information: Different convolutional layers capture features at different scales or levels of abstraction. Concatenating feature maps from various layers allows the model to retain multi-scale information, enabling it to detect both fine-grained and high-level patterns relevant to facial expressions.

Feature Reuse: Concatenation enables feature reuse, as the model can access and combine features from earlier layers with those learned from deeper layers. This enhances the model's ability to understand the relationships between different facial components and their impact on emotions.

Richer Representations: By combining information from multiple layers, the model obtains richer representations of the input data. This can lead to more expressive feature vectors, which are crucial for accurate emotion classification.

Learning Hierarchical Features: Concatenation helps create a hierarchical representation of the input data. As information from earlier layers gets concatenated with deeper layers, the model can build a more complex and informative understanding of facial expressions.

Improving Performance: Concatenation often leads to better performance in complex tasks like facial emotion recognition. It allows the model to learn diverse and complementary features, leading to improved generalization on unseen data.

2.3.4 `what should be the main reason for the better accuracy of my model?`

There could be several reasons contributing to its superior performance:

Domain-specific Representation: Pre-trained models like VGG16, ResNet50, and DenseNet169 are trained on large-scale datasets like ImageNet, which consist of a wide variety of objects and scenes. While they learn general visual features, they might not be optimized specifically for facial emotion recognition. My model, on the other hand, is designed with architecture and hyperparameters tailored to this task, enabling it to learn domain-specific representations more effectively.

Feature Relevance: The model architecture, including the selection of convolutional layers and concatenation, may be well-suited to capture essential facial expression features. The concatenation of feature maps from different layers allows it to combine multi-scale information and learn relevant features that are crucial for emotion recognition.

Dataset Size and Specificity: The size and specificity of the dataset used for training can play a vital role. If the training dataset for facial emotion recognition

	VGG16	DenseNet169	ResNet50
Accuracy	0.8591	0.6146	0.8820
Val_accuracy	0.8571	0.5461	0.8571
Total_params	14,735,879	13,860,679	23,885,383

Figure 2.4: Pre-trained models features

is large and contains diverse facial expressions, it allows the model to learn a broader range of features, leading to better generalization on the validation set.

Fine-tuning and Hyperparameter Tuning: Pre-trained models require fine-tuning to adapt to a specific task like emotion recognition. Proper fine-tuning and hyperparameter tuning are crucial to achieving optimal performance. If the pre-trained models are not fine-tuned effectively, they may not perform as well as your model on the validation set.

Overfitting Avoidance: Pre-trained models often have a large number of parameters, making them more prone to overfitting, especially with limited data. My model, with a specific architecture and appropriate dropout layers, may be better at avoiding overfitting the validation data.

Data Augmentation: Effective data augmentation techniques applied during training can significantly improve the generalization ability of my model. By artificially expanding the dataset through transformations like rotation, flipping, and scaling, the model learns to be more robust to variations in facial expressions.

Task-Specific Design Choices: The model's architecture and design choices may be well-aligned with the intricacies of facial emotion recognition, enabling it to learn more expressive and relevant features compared to the more general pre-trained models.

2.3.5 `why total parameters are less compared to the pre-trained models?`

Pre-trained models like VGG16, ResNet50, and DenseNet169 are deep convolutional neural networks (CNNs) that have been trained on large-scale datasets for general object recognition tasks. These models are designed to learn hierarchical representations of images by stacking numerous convolutional and fully connected layers.

The custom model I am using here has a more compact architecture with a smaller number of layers and parameters. The reduction in parameters can be

attributed to several factors:

Depth of the Network: Pre-trained models like VGG16, ResNet50, and DenseNet169 are much deeper, comprising numerous convolutional blocks and fully connected layers. In contrast, my custom model has a shallower architecture, resulting in fewer parameters.

Model Size and Complexity: Pre-trained models are designed for general object recognition and, hence, need to capture a wide range of complex features. This custom model may have a more specific focus on facial emotion recognition, and thus, it requires fewer parameters to capture relevant features for the task.

Model Width: The width of the network, i.e., the number of channels or neurons in each layer, also affects the total number of parameters. Deeper pre-trained models typically have wider layers to capture more diverse features, contributing to their higher parameter count.

Parameter Sharing and Regularization: Some pre-trained models use parameter-sharing techniques (e.g., ResNet's residual connections) and regularization methods (e.g., Dropout) to optimize the usage of parameters. These techniques increase the model's capacity without adding a proportional number of parameters.

Specialized Architectures: Pre-trained models may include specialized modules (e.g., attention mechanisms, skip connections) to handle specific challenges in general object recognition. This custom model may not require such modules, leading to fewer parameters.

Chapter 3

Future Plans

To improve the validation accuracy in both the CK+ and FER2013 datasets, I plan to incorporate Facial Landmarks, Histogram of Oriented Gradients (HOG), and Local Binary Pattern(LBP) as additional features in my model. Currently, I have only applied HOG and LBP to the CK+ dataset, but I believe that leveraging facial landmarks, HOG, and LBP on both datasets will enhance the model's understanding of facial expressions and lead to better accuracy. By doing so, I aim to achieve more robust and generalized emotion recognition capabilities across both CK+ and FER2013 datasets.

3.1 How can facial landmarks improve emotion recognition accuracy?

Facial landmarks play a crucial role in improving emotion recognition accuracy by providing valuable spatial information about the facial features

Localized Feature Representation: By identifying specific facial landmarks, the model can focus on localized regions around the eyes, nose, mouth, and other crucial facial areas. These localized regions contain vital cues for expressing emotions, such as eyebrow movements, eye widening, or lip curvature during smiling.

Facial Geometry and Expressions: Facial landmarks encode the facial geometry, including the distances and angles between landmarks. Different emotions are associated with distinct facial expressions, and the relative positions of landmarks can indicate specific emotional states.

Invariant to Pose and Scale: Emotion recognition based solely on pixel values can be sensitive to changes in pose and scale. Facial landmarks provide

pose-invariant information, allowing the model to recognize emotions consistently regardless of the face's orientation or size.

Robustness to Lighting Variations: Facial landmarks provide localized information that is less sensitive to variations in lighting conditions. This helps the model to extract emotion-related features accurately even in different lighting scenarios.

Reduction of Noise and Unrelated Features: By focusing on specific facial points, facial landmarks help the model to ignore irrelevant background details and noise in the image, leading to more efficient and accurate emotion recognition.

Enhanced Feature Extraction: Emotions are often conveyed through subtle changes in facial expressions. Facial landmarks enable the model to capture these subtle variations more effectively, leading to improved feature extraction for emotion recognition.

Improved Generalization: By using facial landmarks as a consistent representation, the model can generalize better across different individuals and demographics. It helps the model to recognize emotions in various faces, even if the individuals have diverse appearances.

Facial Dynamics: Facial landmarks can be used to analyze temporal changes in emotion expression, such as how the facial features move and evolve over time. This adds temporal dynamics to emotion recognition, making it more robust and accurate.

3.2 Conclusion

The main objective of the project is to categorize facial emotions through the utilization of pre-trained models, as well as comparing them to my own customized model. In the future, it is possible to improve accuracy by integrating features taken from facial landmarks, HOG, and LBP into the existing model. These may result in better emotion recognition abilities and more precise outcomes.

3.3 Citations and References

The References I have used for this project.

[1] **Huiyuan Yang, Umur Ciftci, and Lijun Yin. Facial Expression Recog- nition by De-expression Residue Learning.** In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 20291-20300

[2] Dan Zeng1, Zhiyuan Lin1, Xiao Yan1, Yuting Liu2, Fei Wang3, Bo Tang*1. Face2Exp: Combating Data Biases for Facial Expression Recog- nition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 20291-20300

[3] Han Yi1,2, Wang Xubin2, Lu Zhengyu2*. Multimodal data fusion and neural network. arXiv preprint arXiv:2109.12724

[4] Jun Yu, Zhongpeng Cai, Renda Li, Gongpeng Zhao, Guochen Xie, Jichao Zhu, Wangyuan Zhu. Exploring Large-scale Unlabeled Faces to Enhance Facial Expression Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2023, pp. 5802-5809

[5] Ewa Piatkowska. Facial Expression Recognition System. DePaul Uni- versity

3.4 Info

Fork me on GitHub and contribute to the project.

Appendices

Appendix A

Equations Using

In this case, we are utilizing the hyperparameters of the Convolution layer. These parameters are responsible for configuring the convolution layer.
- Kernel size(K): Small is better (But if it is on the first layer, it takes a lot of memory)
- Stride(S): How many pixels the kernel window will slide (Typically, there is one convolutional layer and two pooling layers.)
- Zero Padding(pad): Put zeros on the image border to allow the conv output size to be the same as the input size (F=1, PAD=0; F=3, PAD=1; F=5, PAD=2; F=7, PAD=3)
- Number of filters(F): the number of patterns that the conv layer will search for.

$$Nout = ((Nin + 2p - k)/s) + 1$$

where Nout = Size of the output feature map, Nin = size of the input feature map, p = padding, k = kernel size

Appendix B

Illustration of techniques using

The main idea behind LBP is to describe the neighborhood of image elements using binary codes. This method is usually used to study their local properties and identify the characteristics of individual parts of the image. This algorithm is a combination of statistical and structural methods.
When the LBP labeled image fl(x,y) has been obtained, the LBP histogram can be defined as

$$\sum_{x,y} Ifl(x,y) = i, i = 0, ..., n-1, \tag{B.1}$$

in which n is the number of different labels produced by the LBP operator, and IA is 1 if A is true and 0 if A is false.
When the image patches whose histograms are to be compared have different sizes, the histograms must be normalized to get a coherent description:

$$Ni = Hi/(\sum_{j=0}^{n-1} Hj) \tag{B.2}$$

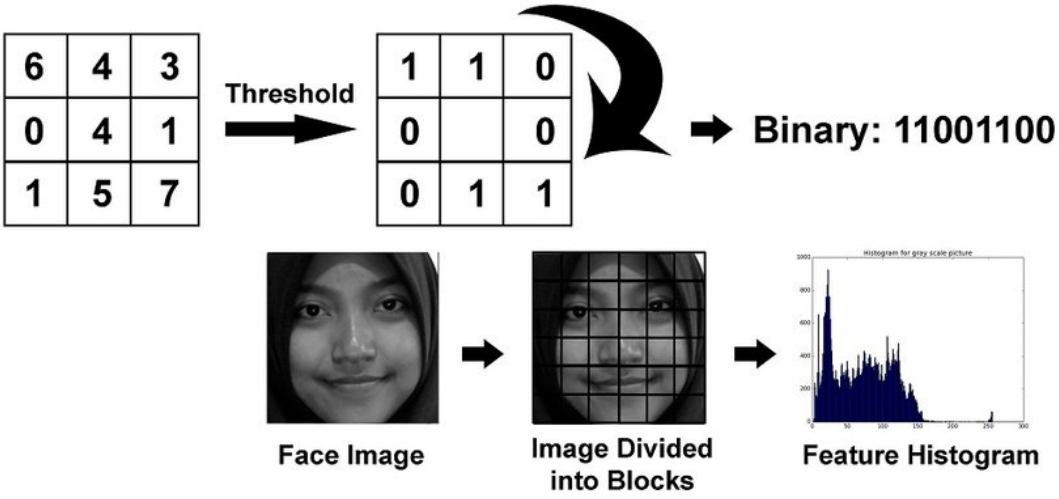

Figure B.1: LBP illustration