EMOTION RECOGNITION FROM FACIAL EXPRESSIONS
Group-9

Srikar Chintha
chintha@buffalo.edu
50511271

Aakanksha Bannuru
abannuru@buffalo.edu
50539271

Lahari Jagarlamudi
govindal@buffalo.edu
50539692

Abstract

Facial Emotion Recognition (FER) holds significant importance across various domains, from human-computer interaction to mental health evaluation. Despite the complexities introduced by facial diversity and image variability, Convolutional Neural Networks (CNNs) have emerged as highly effective tools for FER due to their innate ability to extract features automatically. In our investigation, we introduce a novel approach that capitalizes on a Five-CNN architecture, meticulous optimization of hyper parameters, and exploration of diverse optimization techniques. Through our methodology, we achieve an impressive single-network accuracy of 64.10 on the FER2013 dataset, surpassing prior benchmarks without the necessity for additional training data. This study marks notable advancement in FER, promising enhanced computational efficiency and precision in recognizing emotions.

1 Dataset

In today's era of increasingly seamless human-computer interaction, the ability to discern and interpret emotions from facial expressions has become a subject of great interest. This interest doesn't just arise from its practical applications in fields like human-computer interaction, marketing, and healthcare, but also from its potential to enrich user experiences and comprehension across diverse domains.

1.1 Introduction

This report focuses on delving into the intricacies of emotion detection using facial expressions, centering its analysis on the FER2013 dataset. Despite the inherent complexities posed by the wide array of human facial features and variations in image quality, Convolutional Neural Networks (CNNs) have emerged as potent tools in this area, thanks to their capacity for automated feature extraction. This project seeks to innovate within the realm of FER by proposing a fresh methodology that harnesses a Five-CNN architecture, alongside rigorous optimization of hyperparameters and experimentation with various optimization techniques.

1.2 Data Engineering

We utilized the FER2013 dataset, which consists of over 35,000 grayscale images standardized to 48x48 pixels, each labeled with one of seven emotional expressions. To enhance the dataset's variability and robustness, we applied data augmentation techniques like horizontal flipping and random rotation. These preprocessing steps, combined with resizing and normalization, were critical in preparing the data for effective training and testing of our emotion recognition models.

2 Model Description

2.1 Designed model FIVE CNN Architecture

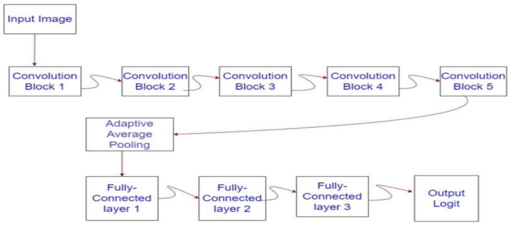


Figure 1: Designed model FIVE CNN Architecture

The neural network, designed for image classification, comprises an input layer, multiple convolutional blocks, adaptive average pooling, and several fully connected layers.

2.1.1 Description and Understanding of the designed algorithm of the model

Input Layer: The model accepts an input image which should be pre-processed to a standardized size, typically 224x224 pixels with three color channels (RGB). **Convolutional Blocks:** The core of the network consists of five convolutional blocks. Each block is designed to capture spatial hierarchies in the image data through filters of increasing complexity.

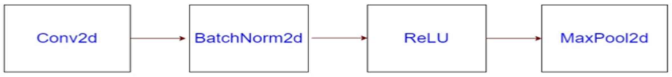


Figure 2: Layers in the model

Every convolution block is followed by batch normalization and ReLU activation. A max pooling layer with a kernel size of 2x2 and a stride of 2 reduces the spatial dimension. Batch Normalization is used here to accelerate neural network training by enabling higher learning rates and stabilizing activation distributions, enhancing model convergence. It addresses the vanishing and exploding gradients. The architecture progresses through five convolution blocks: Block 1 with 64 filters, Block 2 with 128 filters, Block 3 with 256 filters, Block 4 with 512 filters, and Block 5 with 1024 filters. Each uses 3x3 filters, a stride of 1, and padding of 1, increasing capacity to capture more abstract features at each stage.

Adaptive Average Pooling: Following the convolutional stages, an adaptive average pooling layer reduces the output of the last convolutional block to a fixed size of 1x1 per channel, which assists in making the model adaptable to different image sizes.

Fully Connected Layers:

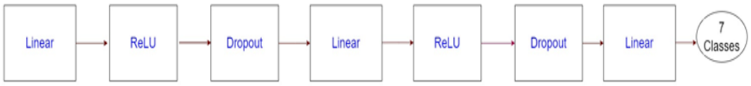


Figure 3: Fully Connected Layers

First layer consists of 4096 neurons, equipped with ReLU activation and a dropout of 0.5 to prevent overfitting. The second fully connected layer has 4096 as input and output.

Output Layer: The final layer has 7 neurons, each representing a class label in the classification task. The softmax activation function could be used here to derive the probability distribution over the classes.

Conclusion: This architecture leverages the power of deep convolutional networks to effectively learn and predict from visual data. The use of increasing filter depths allows the network to extract and learn from a wide range of features, from simple edges to complex objects within the image, making it suitable for diverse image recognition tasks.

2.2 List of Models Tried and why FIVE CNN is the best model

In developing a facial emotion detection system, we evaluated various models and techniques, including ResNet, VGG, and GoogleNet, along with transfer learning using VGG and different multi-class classification algorithms. ResNet, with its deep network and residual connections, offers high accuracy but requires significant computational resources and lengthy training times. VGG is simpler and effective in feature learning but is computationally intensive due to its deep layers. Although transfer learning with VGG aimed to utilize pre-trained features to cut training time and enhance generalization, it often fell short without extensive fine-tuning due to the unique characteristics of emotion data. Consequently, a custom-designed five-layer CNN model proved most effective. This model balances efficiency and depth, incorporating batch normalization to prevent vanishing gradients and avoiding the high computational costs of VGG and the complexity of ResNet. This tailored architecture provides quick training, efficient inference, and robust performance, making it ideal for real-world applications in facial emotion detection.

3 Loss Function

A loss function, also known as a cost function, is essential for evaluating how well a machine learning model's predictions align with actual target values. In multiclass classification tasks, selecting the appropriate loss function is crucial for successful model training. Through thorough evaluation of various options, our goal was to enhance performance by identifying the most suitable choice. Among the options considered, cross-entropy loss emerged as the top contender because of its inherent ability to accurately assess the difference between predicted and true probability distributions.

3.1 Selected Loss Function

Cross Entropy Loss: The adoption of Cross Entropy Loss played a vital role in evaluating alignment between predictions and actual emotion probabilities. It penalized disparities and enabled precise predictions. Integrating Softmax layer streamlined the models accuracy and performance. There was a significant increase in accuracy, reaching 64. Continuous parameter adjustments minimized loss, highlighting the effectiveness of Cross Entropy Loss in refining emotion detection model performance.

3.2 Tried Loss Functions

In our study on loss functions for emotion detection, we faced the challenge of class imbalance among seven emotions in our dataset. To tackle this issue, we employed focal loss, which reduces the emphasis on well-classified classes. However, its implementation only marginally improved model accuracy, reaching around 56. Kullback-Leibler (KL) Divergence Loss, measuring the gap between predicted and actual emotion probabilities, resulted in a slight accuracy boost to 60. Additionally, Sparse Categorical Cross-Entropy Loss was utilized to handle cases where target labels are encoded as integers, aiding training optimization and improving the model's ability to accurately predict emotions.

3.3 Innovation on Loss Function

Innovations in loss functions have notably boosted model accuracy, with focal loss being one such advancement. It addresses class imbalance by adjusting the weighting of well-classified examples during training, prioritizing hard-to-classify instances to capture nuances in less prevalent classes, thus enhancing overall accuracy. Additionally, exploring Kullback-Leibler Divergence (KLD) loss provides insights into disparities between predicted and true probability distributions, refining the training process and contributing to improved model precision and emotion detection performance.

4 Optimizers

Neural networks are trained using optimisation algorithms, which are essential tools that modify model parameters iteratively in order to minimise a predetermined loss function.Stochastic Gradient Descent (SGD) is a popular optimisation method that works by using a subset of data (mini-batch) in each iteration to compute gradients of the loss function with respect to the model parameters. Conversely, more sophisticated algorithms like as Adam dynamically adjust the learning rate for each parameter in response to previous gradients, leveraging momentum to improve resilience and speed up convergence, especially when the gradients are noisy or sparse.

4.1 Selected Optimizer

Adam: Implementing the Adam optimizer significantly boosted our emotion detection system's effectiveness. Its adaptive learning rate and momentum integration allowed efficient navigation through complex optimization landscapes, crucial for swift convergence in scenarios with noisy gradients. Moreover, Adam's reduced need for hyperparameter tuning streamlined our optimization process, enhancing the model's robustness and efficiency while freeing up resources for further refinement. Overall, Adam's integration notably improved the accuracy and reliability of our system, enabling more effective recognition and interpretation of emotions from input data.

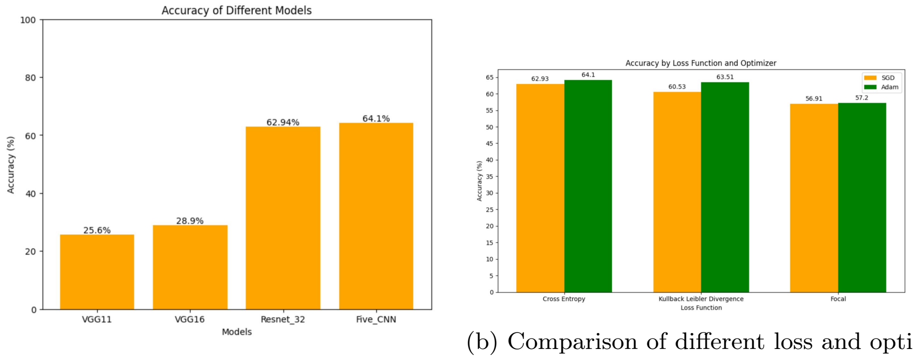
4.2 Tried Optimizers

To enhance performance in emotion detection tasks, three unique optimization methods were tested. Stochastic Gradient Descent (SGD) was utilized as an iterative technique to adjust model parameters by randomly selecting data subsets, resulting in a substantial 62 enhancement in model accuracy. Adagrad implemented an adaptive learning rate approach, tailoring rates to individual parameters based on observed gradient magnitudes, effectively navigating the parameter space and reducing risks of divergence or oscillation during training, particularly beneficial for sparse data scenarios in emotion detection. Additionally, RMSProp contributed by adapting learning rates according to recent gradient magnitudes, improving convergence efficiency and ensuring stable optimization, a critical aspect for enhancing model performance across various applications, including emotion detection.

4.3 Innovation in Optimization Algorithm

By enhancing adaptive learning rate , aiming to address challenges such as varying feature scales and sparse datasets commonly encountered in our model. Algorithms like Adagrad and RMSprop introduced dynamic adjustment mechanisms for learning rates, tailoring them to individual parameters based on gradient magnitudes. This ensures efficient navigation of the parameter space while minimizing the risk of divergence or oscillation during training. These innovations have significantly contributed to stable optimization and improved model performance.

5 Metrics and Experimental Results



(a) Different Models

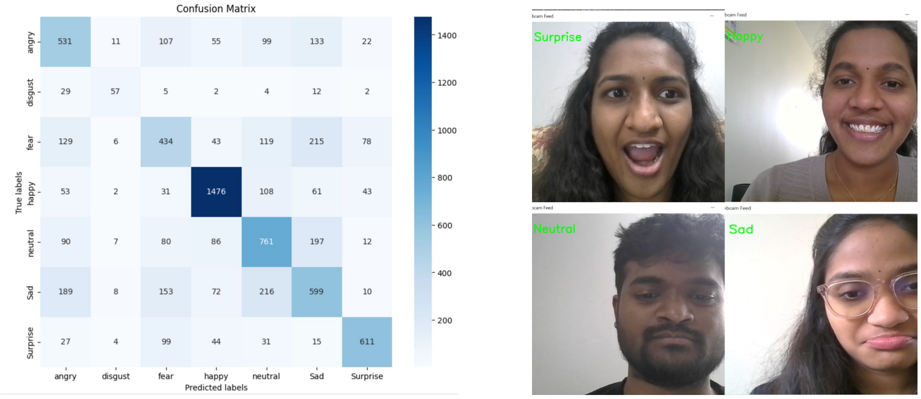
(b) Comparison of different loss and optimizer combinations

Figure 4: Graphical comparison

The first graph(a) showcases the accuracy of various convolutional neural network models on the FER-2013 dataset. The models VGG11 and VGG16 show much lower accuracy, achieving around 25.6 and 28.9 respectively. In contrast, Resnet-32 and a custom Five-CNN model perform significantly better, with accuracies of 62.94 and 64.1.

The second graph(b) compares the performance of different loss functions and optimizers. Adam optimizer consistently outperforms SGD across all loss functions. Among the loss functions, using Adam with Kullback Leibler Divergence yields the highest accuracy at 64.1, while Cross Entropy and Focal loss functions also show good performance with Adam, suggesting its suitability for optimizing this task.

In our testing of the facial emotion detection model using the FER-2013 dataset, the confusion matrix reveals detailed performance insights. The model



	angry	disgust	fear	happy	neutral	Sad	Surprise
angry	531	11	107	55	99	133	22
disgust	29	57	5	2	4	12	2
fear	129	6	434	43	119	215	78
happy	53	2	31	1476	108	61	43
neutral	90	7	80	86	761	197	12
Sad	189	8	153	72	216	599	10
Surprise	27	4	99	44	31	15	611

(a) Overview of the confusion matrix

(b) Real-time Expressions

Figure 5: Results

shows strong accuracy in recognizing ’happy’ emotions with 1476 correct predictions, indicating robust detection capabilities for this category. However, it exhibits challenges in accurately classifying ’fear’ and ’sad’, often misclassifying them as ’angry’ and ’neutral’, respectively. ’Surprise’ is also well-recognized with 611 correct predictions, but like ’fear’ and ’sad’, it is occasionally confused with ’angry’. Overall, while the model excels in some areas, it demonstrates the need for further refinement in distinguishing between emotions with similar facial expressions.

References

[1] Ko, ByoungChul. "A brief review of facial emotion recognition based on visual information." Sensors 18.2 (2018): 401.

[2] Li, Shan, and Weihong Deng. "Deep facial expression recognition: A survey." IEEE Transactions on Affective Computing 13.3 (2022): 1195-1215.

6 Contributions and Github

We have contributed equally for developing the project, which is 33.33 each. Below is the Github link.

https://github.com/ChinthaSrikar/Emotion_Recognition_From_Facial_Expressions