Group 15 : Emotion Recognition from Facial Expressions

Leela Srija Alla
University at Buffalo (SUNY)
Buffalo, NY, USA
lalla@buffalo.edu

PavanKalyan Nayak Guguloth
University at Buffalo (SUNY)
Buffalo, NY, USA
pgugulot@buffalo.edu

Vishnu Teja Jampala
University at Buffalo (SUNY)
Buffalo, NY, USA
vjampala@buffalo.edu

Abstract

The goal of this project is to develop a model that can identify human emotions from facial expressions. Our model is VGG combined with Long Short-Term Memory (LSTM) networks and an attention mechanism, to detect facial emotions. These faces are part of a dataset, FER-2013, which includes several thousand images labeled with emotions like happy, sad, anger, disgust, fear, surprise and neutral. The project spans several phases, including pre processing the data, building and refining the model, and testing it to ensure it works well. This project enhances computer vision capabilities and could be useful in areas such as interactive technologies and mental health assessment.

1 Dataset

1.1 FER-2013

The FER-2013 dataset consists of 32,298 grayscale images of human faces, each image being 48x48 pixels. The images are standardized to ensure that each face is centered and occupies a consistent amount of space. This standardization is crucial as it enhances the uniformity of the input data for emotion classification. The dataset is divided into 28,709 training examples and 3,589 test examples. The training set is used for developing the model, while the test set allows for the evaluation of the model's performance in real-world scenarios.

1.2 Preprocessing Steps:

All images are resized to a uniform size of 48x48 pixels, which is necessary for consistent input into neural networks. Images are converted to grayscale to reduce the complexity of the model. Images are randomly flipped horizontally. This augmentation helps the model to learn facial expressions that are independent of the orientation of the face. Images are randomly rotated by up to 30 degrees. This step increases the robustness of the model by training it to recognize emotions across different facial orientations. Images are converted into tensors to be processed by neural networks. The pixel values are normalized using a mean of 0.5 and a standard deviation of 0.5. This normalization ensures that the input values are on a common scale and helps in speeding up the convergence during training by reducing issues related to the scale of different features. As shown in Figure 1, each image in the dataset is labeled with one of seven emotional expressions: 'happy' (0), 'sad' (1), 'angry' (2),

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

'disgust' (3), 'fear' (4), 'surprise' (5), and 'neutral' (6). These labels facilitate the supervised learning of emotion recognition models. Figure 2 shows a bar graph presents the distribution of samples across various emotions in the FER-2013 dataset. Notably, the 'happiness' category has the highest number of samples, indicating a significant focus on this emotion within the dataset. Conversely, 'disgust' has the fewest samples, suggesting less emphasis on this emotion.

Figure 1: Sample images

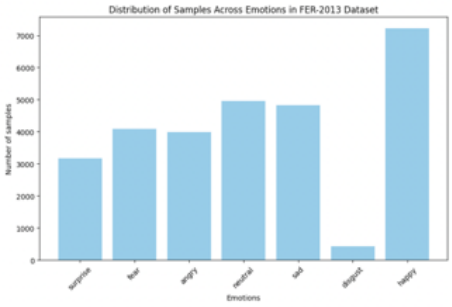

Figure 2: Data Distribution

2 Model

2.1 Model Architecture

Figure 3 score architecture of our model leverages the VGG network, which is renowned for its effectiveness in deep learning tasks involving image data. VGG's architecture is composed of sequentially stacked convolutional layers, which are designed to incrementally increase in depth from 64 to 512 channels. This structure allows the network to extract increasingly complex features from the input images.

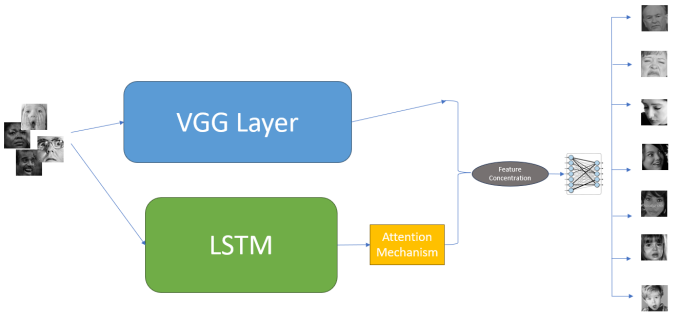

Figure 3: Model Architecture

Additionally, our model incorporates Long Short-Term Memory (LSTM) networks to capture temporal features from the image sequences. We further enhance the model with an attention mechanism, which allows the model to focus selectively on more relevant parts of the image sequence.

2.2 Description and Understanding of the Algorithm

The algorithm we've employed for facial emotion recognition is a combination of Convolutional Neural Networks (CNNs) with a VGG architecture, Long Short-Term Memory (LSTM) networks, and an attention mechanism. Each component plays a crucial role in processing and interpreting facial expressions from image sequences, ensuring robust and accurate emotion classification. Here's a detailed breakdown of how each part contributes to the overall functionality:

2.2.1 VGG Network:

The VGG network is a deep convolutional neural network known for its simplicity and high performance in image recognition tasks. The network consists of multiple convolutional layers stacked sequentially. As the network goes deeper, the number of channels increases. This progression allows the network to process more complex features. Max pooling is used following several convolutional layers to reduce the spatial size of the representation.

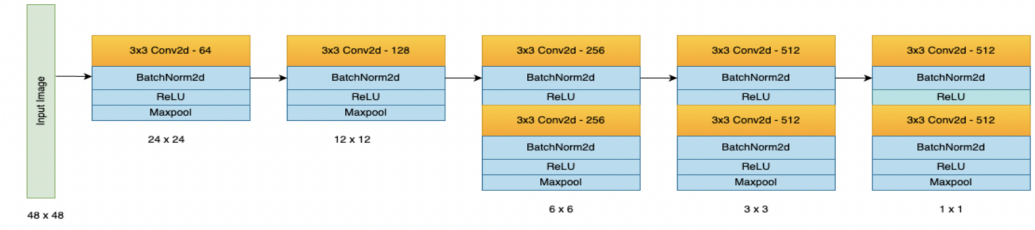

Figure 4: VGG Architecture

In our model, we introduced batch normalization to VGG network. This is applied after the convolutional layers to help stabilize the learning process by normalizing the inputs to a layer. It also speeds up training and improves overall performance.

2.2.2 LSTM Network:

LSTM networks are a type of recurrent neural network (RNN) suitable for sequence prediction problems. LSTMs are used to understand the temporal dynamics of facial expressions. Emotions can change rapidly, and understanding these changes over time is crucial for accurate classification. One of the key features of LSTMs is their ability to remember important information and forget non-essential data through gates (input, output, and forget gates). This capability allows the model to maintain relevant historical information across long sequences without being overwhelmed by data.

In our model, the images are directly fed into an LSTM to extract temporal features as shown in Figure 5. The images, which initially have a tensor size of 32, 1, 48, 48, are reshaped to 32, 48, 48. This means that we are treating each row of the image as a sequence input to each hidden layer in the LSTM. The LSTM is thereby trained to discern relationships between different rows of the image, aiding the model in classifying emotions that are visually similar, such as sad and neutral.

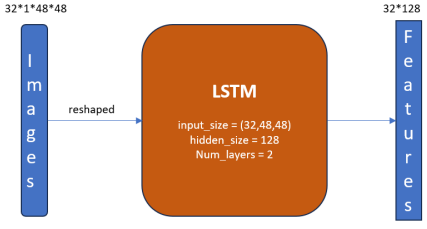

Figure 5: LSTM

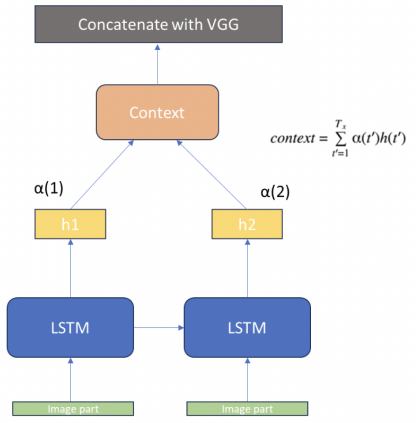

Figure 6: LSTM with Attention

2.2.3 Attention Mechanism:

The attention mechanism enhances the model's ability to focus on the most relevant parts of the input for making decisions. It allows the model to learn to concentrate on specific areas of an image that are more informative for emotion recognition, such as eyes. The context vector from the attention layer is combined with the output of the LSTM layers. This integration helps the model to not only consider the spatial features learned through CNNs but also to focus dynamically on the most emotionally expressive features over time.

In our model, we implemented an attention layer over the LSTM layer as shown in Figure 6. The attention mechanism can weigh the importance of each hidden state, thereby focusing on the most significant temporal features.

$$e_t = \tanh(W_{att} \cdot h_t) \tag{1}$$

$$\alpha_t = \text{softmax}(e_t) \tag{2}$$

The attention weights are computed using Equations 1 and 2 are used to calculate weights. W_{att} is the weight matrix

2.3 List of Models Tried and Why the Chosen One is the Best

In the development of our facial emotion recognition system, several models were experimented with to identify the most effective configuration for accurately classifying emotions from facial expressions. Below, we detail the models we tested, the rationale behind their selection, and the reasons why the final model was chosen as the best.

2.3.1 Baseline VGG Model:

- Architecture: VGG 19
- Rationale: We started with the VGG19 architecture as our base model and modified it by integrating batch normalization layers after each convolutional layer. This adaptation was intended to enhance the network's stability and efficiency during training by normalizing the activations. The addition of batch normalization helps in accelerating the convergence of the model, reducing the internal covariate shift, and allowing us to use higher learning rates safely.
- Performance: Provided reasonable accuracy in detecting basic emotions like happy , neutral but failed in cases of disgust and fear.

2.3.2 VGG with LSTM:

- Architecture: Combined the spatial feature extraction capabilities of VGG with the temporal processing strength of LSTM networks.
- Rationale: VGG19 excels in capturing spatial details from images, identifying key visual elements linked to emotions. LSTM complements this by analyzing the temporal progression of these features across image sequences. This combination allows our model to not only extract detailed visual cues from each frame but also track emotional transitions over time, significantly improving its ability to understand and predict complex emotional states. Upon experimentation, we observed that the combination of VGG11 with LSTM performs as well as VGG19 with LSTM.
- Performance: Showed a significant improvement over the baseline by recognizing emotional transitions.

2.3.3 VGG with LSTM and Dropout:

- Architecture: This iteration added dropout layers after max pooling layers of VGG11 model to combat overfitting. This is combined with LSTM parallelly.
- Rationale: To increase the model's generalization ability on unseen data by reducing overfitting, which is common in deep learning models trained on extensive data.
- Performance: The addition of dropout did not yield a significant improvement in performance. We observed that the performance of LSTM alone was not satisfactory, so we are considering the addition of an attention mechanism to enhance its capabilities.

2.3.4 VGG with LSTM and Attention (Final Model):

- Architecture: Integrates VGG11 and LSTM with an attention mechanism.
- Rationale: To enhance the model's focus on the most relevant features for emotion recognition, especially in sequences where certain frames are more expressive of the emotion than others.
- Performance: This model provided the highest accuracy by effectively combining the strengths of spatial and temporal feature recognition while also dynamically focusing on the most informative parts of the images. It excelled in differentiating between closely related emotions and performed robustly across various test scenarios.

3 Loss Function

3.1 Chosen Loss Function: Cross Entropy Loss

For our model, we chose Cross Entropy Loss as our primary loss function. Cross entropy loss, widely used in classification problems, measures the performance of a classification model whose output is a probability value between 0 and 1. It is particularly effective in scenarios where the classes are mutually exclusive, which applies to our task of classifying facial expressions into distinct emotion categories.

3.2 Other Loss Functions Tried

Focal Loss

To address the class imbalance present in the FER-2013 dataset as shown in Figure 2, where some emotions like 'Disgust' are underrepresented, we experimented with Focal Loss. Focal loss introduces a modulating factor to the cross entropy criterion, aiming to focus learning on hard misclassified examples and reduce the relative loss for well-classified examples.

3.2.1 Focal loss with Alpha Scalar Values:

we tried with alpha values = 0.01, 0.1, 0.25, 0.3.

3.2.2 Focal Loss with Alpha Vector Values

We tried with torch.tensor([1.0, 1.2 ,1.1,1.3, 1.1,1.0,1.1]). The alpha values were adjusted to weigh more heavily on the minority classes, such as 'Disgust', to counterbalance their fewer instances during training.

Innovation on the Loss Function To refine the model's performance further, we innovated on the application of Focal Loss by adjusting alpha values dynamically based on the performance feedback loop from validation results. This dynamic adjustment aims to continuously fine-tune the emphasis on classes according to their misclassification rates. But, We didn't observe any improvements compared to cross entropy loss.

4 Optimization Algorithm

Chosen Optimization Algorithm: AdamW

we selected AdamW as the optimization algorithm. AdamW is a variant of the Adam optimizer that separates the weight decay from the optimization steps, which helps in better regularization and consequently leads to improved training dynamics. Given the complexity of our model, involving VGG layers, LSTM, and an attention mechanism, AdamW provided a balance between efficient computation and effective handling of sparse gradients, which enhanced the overall training performance and stability.

Experiments - We tried with few other Optimizers of SGD, SGD with momentum and Adam without weight decay.

Innovations on the Optimization Algorithm While we did not introduce new optimization algorithms, we tried few innovative strategies to enhance the performance of the existing optimizers. We conducted extensive experiments with the learning rates and decay parameters within AdamW to find the optimal settings that minimized the validation loss and improved the generalization of the model on unseen data.

5 Metrics and Experimental Results

5.1 Metrics used

Accuracy - It is calculated as the number of correct predictions (both true positives and true negatives) divided by the total number of predictions made.

Confusion Matrix - A confusion matrix is a table used to describe the performance of a classification model on a set of test data for which the true values are known. It lists each class and the number of times the model predicted each class compared to the actual class, allowing detailed analysis of both correct and incorrect predictions.

5.2 Predicted results

Our final model, integrating VGG11, LSTM with 2 layers, and an attention mechanism, achieved a train accuracy of 80.51% and test accuracy of 65.99% on the FER-2013 dataset.

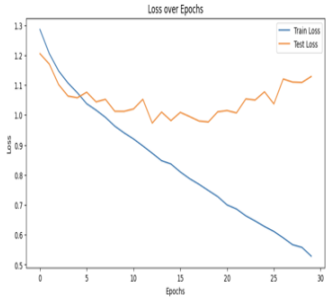

Figure 7: Loss over Epochs

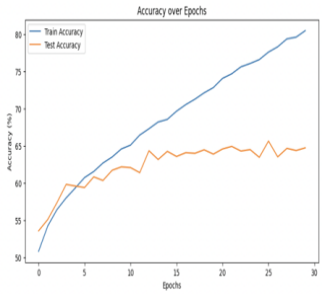

Figure 8: Accuracy over Epochs

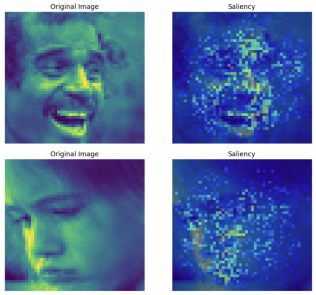

Figure 9: Saliency maps

The loss over epochs graph reveals a consistent decrease in training loss, whereas the test loss exhibits fluctuation and an increasing trend from around the 10th epoch onwards. This divergence hints at overfitting as shown in Figure 7. Confusion matrix as shown in Figure 13 tells us the high predictive accuracy for the emotions 'Happy' and 'Surprise', reflecting its strength in recognizing clear, distinct emotional expressions.

Saliency maps as shown in Figure 9 are included to show which regions of the face most significantly influence the model's predictions. These maps highlight the importance of facial features such as the eyes, nose, and mouth in the emotion recognition process.

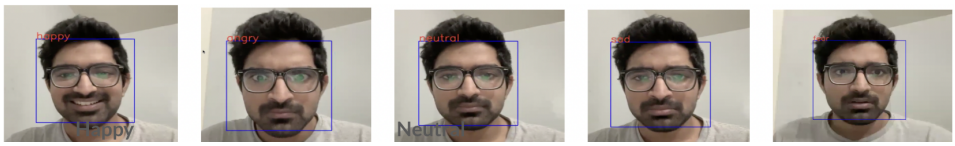

Figure 10: Live web cam Results

We integrated our model with a live webcam to evaluate its performance in real-time conditions. The model successfully detected emotions including 'Happy', 'Neutral', 'Sad', 'Angry', and 'Fear' with high accuracy, as demonstrated in the live feed. However, it struggled to correctly identify the emotion 'Disgust', indicating a need for more representative data for this particular emotion.

5.3 Comparing experiments

Table 1: Hyperparameters experiments on VGG + LSTM + Attention model

Loss Function	Optimizer	LR	Momentum	WD	Test Accuracy
Cross Entropy Loss	AdamW	0.001	-	0.001	63.85%
Cross Entropy Loss	AdamW	0.001	-	0.0001	65.01%
Cross Entropy Loss	AdamW	0.0001	-	0.001	65.99%
Cross Entropy Loss	AdamW	0.01	-	0.001	60.66%
Cross Entropy Loss	SGD	0.001	0.9	0.001	59.10%
FocalScalerLoss	AdamW	0.001	-	0.001	63.04%
FocalVectorLoss	AdamW	0.001	-	0.001	62.37%

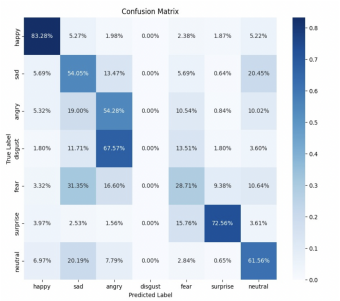

Figure 11: VGG19

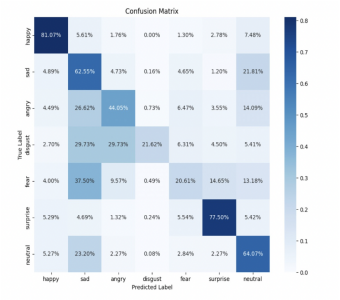

Figure 12: VGG11 + LSTM

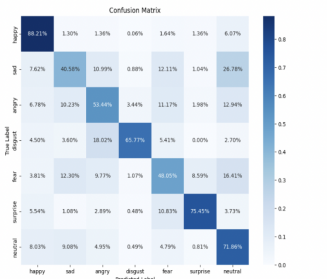

Figure 13: VGG11 + LSTM + attention

From the above confusion matrices, VGG19 model shows strong performance in recognizing clear expressions like 'Happy' and 'Surprise', with high accuracy in these categories. However, it struggles with more complex emotions like 'Disgust' and 'Fear', which are often confused with other emotional states.

With the integration of LSTM, the model improves in distinguishing temporal dynamics in facial expressions, leading to better handling of sequences and slight improvements in classifying emotions like 'Fear' and 'Sad'. Despite these enhancements, the model still shows confusion between closely related emotional expressions, reflecting the LSTM's ability to capture changes over time but still lacking in spatial feature discrimination.

The addition of an attention mechanism significantly refines the model's performance by directing focus towards the most relevant features of the input data for emotion recognition. This is evident in the reduced misclassification rates for 'Fear' and 'Sad', where the attention mechanism helps the model to differentiate between these closely related emotions more effectively. The final configuration using both LSTM and attention shows the best performance, highlighting its effectiveness in handling the complexities of emotion recognition from facial expressions.

6 Contributions

Our entire team have contributed equally for the project, which is 33.33% each. Github link for code - `https://github.com/VishnuJampalaUB/Emotion_Recognition_From_Facial_Expressions`

7 References

[1] Gahlan, Neha Sethia, Divyashikha "Emotion Recognition from Facial Expressions using Deep Recurrent Attention Network"