Independent Study Report – Facial Emotion Recognition

Introduction:

In the landscape of computational challenges, a compelling issue emerges: the categorization of facial data into distinct expression classes. This necessitates the application of various machine learning models, among which Neural Networks and Convolutional Neural Networks (CNNs) stand out. This research endeavors to addressthe problem through the utilization of CNN architecture. The focus on CNNs aims to provide a methodical approach to deciphering the intricate nuances embedded in facial expressions. As we embark on this scientific exploration, the amalgamation of technology and the intricacy of human emotion holds the promise ofunlocking unprecedented dimensions within the domains of artificial intelligence and human-computer interaction.

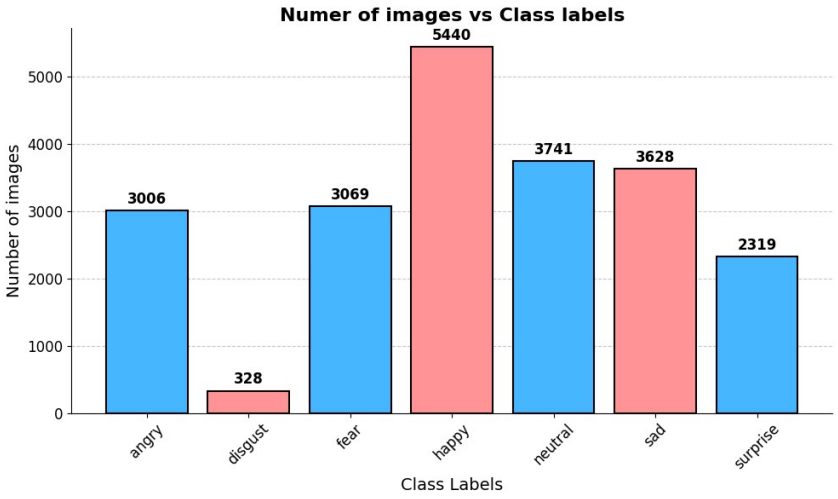
Dataset*:*

In this study, the FER-2013 dataset has been employed, a widely recognized standard dataset for facial expression recognition that encompasses labels for emotions such ashappiness, sadness, anger, and others. The dataset comprises both training and test datasets, with data for each expression class organized within the respective class folders. We further split the Train dataset into train and validation dataset with 3:1 ratio.To standardize the input images, we normalized them to have a mean of 0.485, 0.456, 0.406 for each layer of an RGB image, and a standard deviation of 0.229, 0.224, 0.225.

Furthermore, various augmentation techniques were implemented to mitigate overfittingin the training data, including:

1. Color jitter
2. Random rotation by 30 degrees
3. Random affine transformation by 0.1, 0.1
4. Random horizontal flip

The application of these data augmentation techniques to the training dataset yielded highly generalized results without succumbing to overfitting.



Training:

We included a common training and testing framework for all models. While training a model on each epoch we evaluate the model on the validation dataset and save the best models based on validation set accuracy and loss value. This ensures weare using the best version of the model. Here we can include regularization to the loss values as well. For testing the model, we test the performance of the model on a testdataset.

Models:

1) **Basic CNN model:**

The presented model, denoted as CNN Model, is a fundamental Convolutional Neural Network (CNN) architecture designed for image classification tasks. The model features a sequential arrangement of convolutional, batch normalization, rectified linearunit (ReLU) activation, dropout, and max-pooling layers, aiming to effectively learn hierarchical representations from input images.

The initial layer, conv1, convolves the input image with 16 filters of size 3x3, with subsequent batch normalization, ReLU activation, dropout, and max-pooling operations.The pattern is repeated with conv2, where the number of filters is increased to 32. The max-pooling layers reduce spatial dimensions, aiding in the extraction of salient features.

The subsequent fully connected layers, fc1 and fc2, contribute to feature aggregation and final classification. The flattened output from the convolutional layers is processed through a linear layer (fc1), followed by batch normalization, ReLU activation, and dropout. The final linear layer (fc2) produces logits for the classification task, and the logSoftMax activation function is applied to yield class.

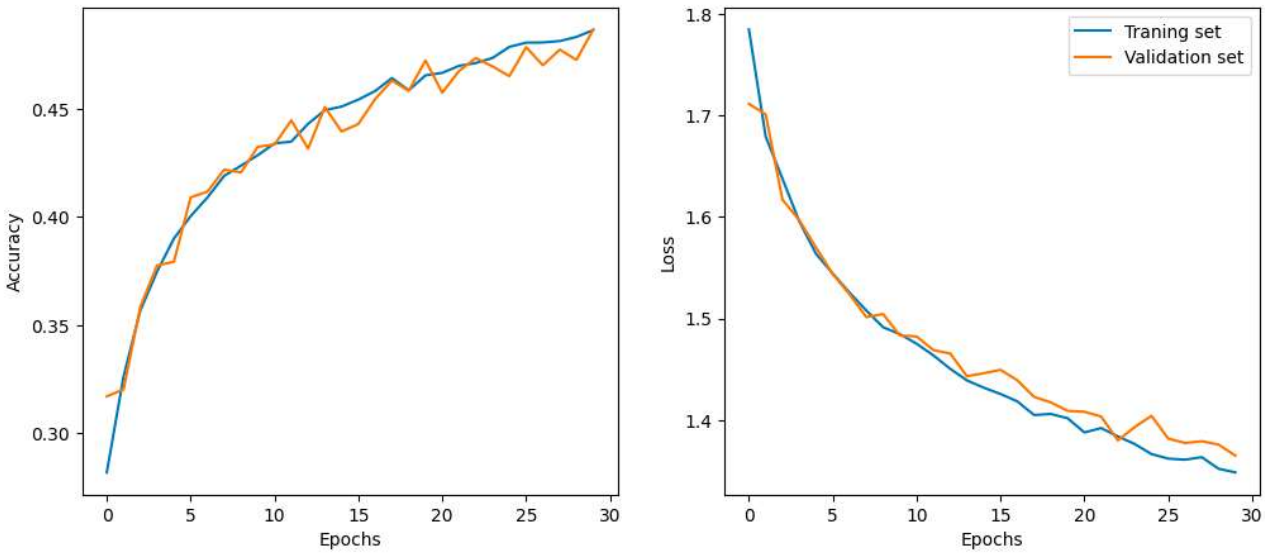


Fig 1: Test Accuracy is 53.9%

2) CNN model with attention block:

The proposed model, referred to as Attn_CNN, is a Convolutional Neural Network (CNN) architecture enhanced with an attention mechanism. This model isspecifically designed for image classification tasks, with a focus on capturing and highlighting salient features through the integration of an attention block.

The architecture comprises multiple layers, starting with two convolutional layers, each followed by batch normalization, rectified linear unit (ReLU) activation, dropout regularization, and max-pooling operations. The attention block, a key innovation in thismodel, is strategically inserted after the second convolutional layer. This attention mechanism, implemented as an Attention Block module, introduces adaptability by assigning weights to different regions of the feature map, emphasizing critical visual cues.

Following the convolutional layers, the network incorporates fully connected layers for further feature extraction and classification. The flattened output from the convolutionallayers is processed through a linear layer (fc1), followed by batch normalization, ReLU activation, and dropout. Subsequently, the final linear layer (fc2) produces the outputlogits for the classification task. The activation function used for the final layer is alogarithmic SoftMax to yield probabilistic predictions across the specified classes.

This Attn_CNN model presents a holistic approach to image classification, leveraging attention mechanisms to enhance the network's ability to focus on relevant information within the input images. The integration of attention introduces a level of adaptability, allowing the model to dynamically emphasize discriminative features during the learning process.

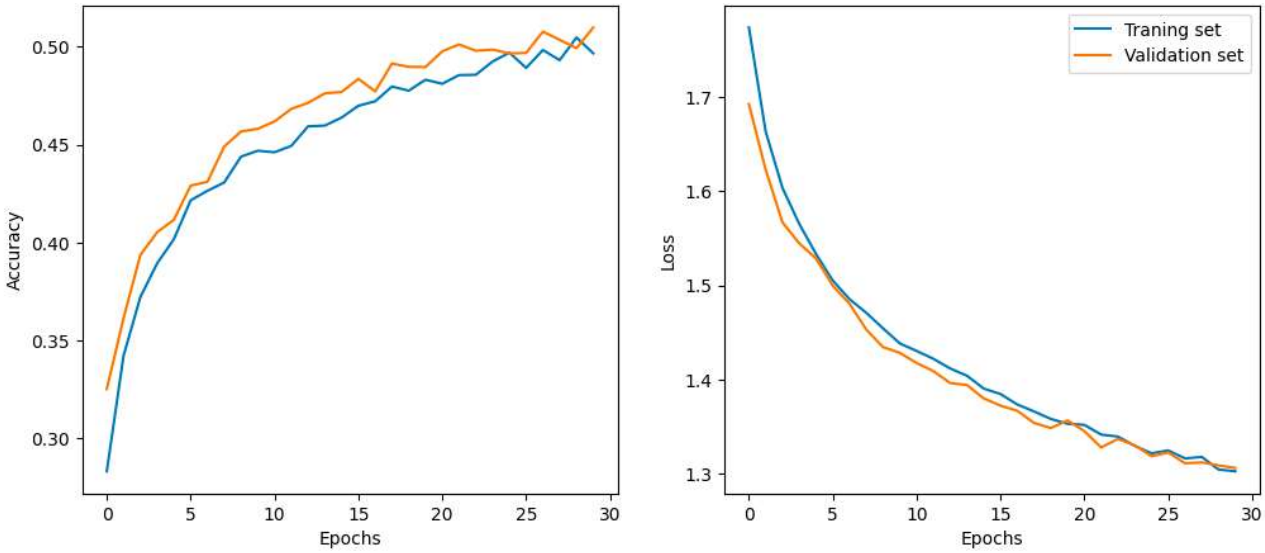


Fig 2: Test Accuracy is 55.2%

3) Fusion CNN (Combination of Unet, resnet and inception blocks and concepts):

The Fusion CNN model represents a comprehensive architecture designed for imageclassification, incorporating advanced concepts from Encoder-Decoder structures, ResNet, and Inception blocks. The model's objective is to capture intricate featuresthrough a hierarchical network that balances both local and global representations.

In the encoding phase, the model begins with the conv1 layer, utilizing 64 filters of size3x3 to transform the input image. The subsequent res_block1 introduces a Residual Block, enhancing feature extraction and down sampling with a stride of 2. Following this,the incept_block1 embeds an Inception Block, featuring 1x1, 3x3, and 5x5 convolutionalbranches, along with a 3x3 max-pool branch. This structure is repeated with res_block2and incept_block2, further refining features and expanding diversity.

The decoding phase initiates with deconv1, a transposed convolutional layer facilitatingup sampling. The res_block3 then applies a Residual Block to preserve learned featuresduring the up-sampling process. The subsequent incept_block3 mirrors the structure of the encoding phase, incorporating an Inception Block. Further up sampling is facilitated by deconv2, followed by the application of res_block4 to refine features. The decoding concludes with incept_block4, a final Inception Block.

The classifier component of Fusion CNN involves global_avg_pooling, implementing global average pooling to reduce spatial dimensions. The final classification logits areproduced through the fc (fully connected) layer, offering a comprehensive representation of the learned features.

The unique fusion of Residual Blocks, Inception Blocks, and Encoder-Decoder structures in Fusion CNN establishes a robust framework for image classification. Thisamalgamation of architectural elements is poised to enhance the model's adaptability and performance, allowing it to discern intricate patterns within diverse datasets. Themodel's capacity to capture both local and global features position it as a promisingcandidate for a wide range of image classification tasks.

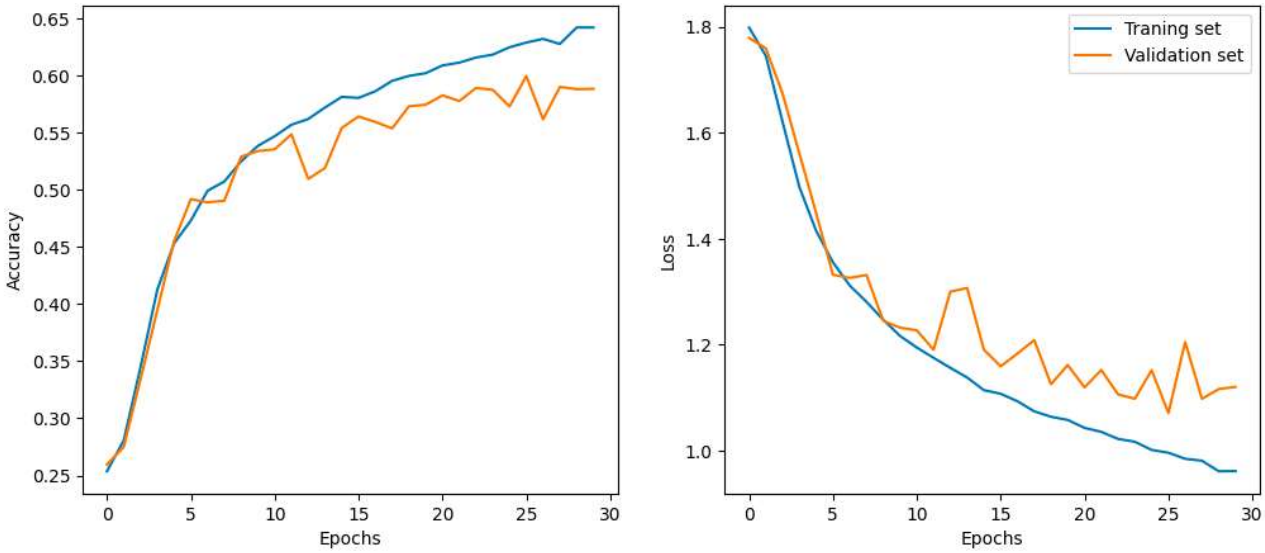


Fig 3: Test Accuracy is 60.8%

4) NasUnet (Combination of Shuffle net, resnet, Unet and Nasnet)

The NasUnet model is a sophisticated architecture tailored for image classification, combining concepts from Encoder-Decoder structures, ResNet, ShuffleNet, and a Reduction Cell inspired by NasNet. This amalgamation is designed to harness the strengths of each architectural element, providing a comprehensive framework for feature extraction and classification.

Encoder:

The encoding phase commences with a traditional convolutional layer, conv1, processing the input with 64 filters of size 3x3. This is followed by Residual Blocks, specifically res_block1 and res_block2, which facilitate hierarchical feature extractionwith down sampling to capture both local and global features. Shuffle Net Blocks, denoted as shufflenet_block1 and shufflenet_block2, contribute further to feature diversity. These blocks implement a combination of 1x1 and 3x3 convolutions within grouped convolutions to enhance feature representation. The model incorporates a reduction cell inspired by NasNet, which intelligently mergesspatial and channel-wise information through a series of convolutions and batch normalization.

Decoder:

In the decoding phase, the model employs transposed convolutional layers, such as deconv1 and deconv2, to upsample the features. Residual Blocks, dec_res_block1, areapplied to preserve learned features during the up-sampling process. A combination of convolutional layers, conv2 and conv3, further refines the features before arriving at thefinal classification. The model concludes with a fully connected layer, fc, producing the final classificationlogits. The activation function used is log SoftMax, providing probabilistic predictions across the specified classes.

The NasUnet architecture showcases a synergistic fusion of various proven architectural elements, demonstrating a holistic approach to image classification tasks.The incorporation of ResNet blocks, ShuffleNet blocks, and a NasNet-inspired Reduction Cell contributes to the model's adaptability, robust feature extraction, andenhanced performance in discerning intricate patterns within diverse datasets.

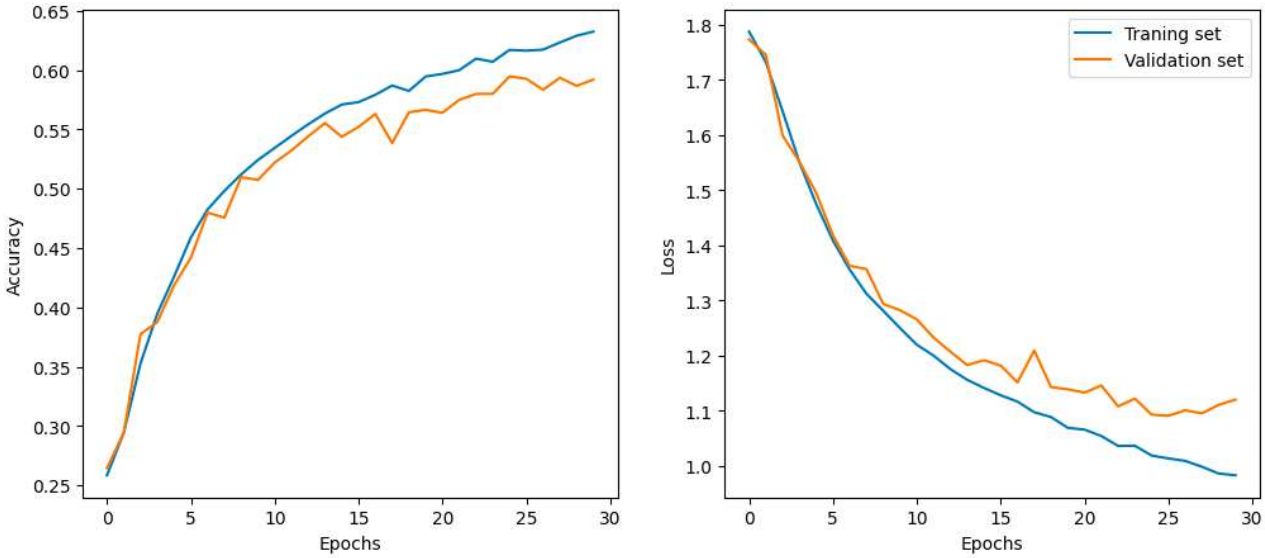


Fig 4: Test Accuracy is 61.4%

Loss Function:

We opted for the Cross Entropy loss, an optimal choice for multiclass classification tasks, given its effectiveness in minimizing the disparity between predictedand actual probability distributions. In contrast, other loss functions like Log loss and BCE were deemed unsuitable for our classification task.

The Cross Entropy loss aims to align predicted probabilities with actual label distributions. In addressing class imbalance, we implemented weighted cross-entropyloss, assigning different weights to classes based on their significance or frequency in the dataset. Given the observed class imbalance in our case, we assigned more weight to classes with fewer samples, resulting in a notable improvement in the recall and precision of these less-represented classes.

Optimization algorithm:

In our investigation, we conducted experiments with two prominent optimization algorithms, namely Stochastic Gradient Descent (SGD) and Adam. The Adam optimizeremerged as the preferred choice due to its exceptional convergence in minimizing loss over a reduced number of iterations. This efficacy is attributed to Adam's ability to sustain a moving average of gradients and square gradients, a process regulated by decay variables.

However, the Stochastic Gradient Descent (SGD) optimizer was strategically employedto achieve an optimal solution over an extended training period. This is particularly advantageous for more intricate models, where employing higher epochs becomes crucial for obtaining desirable outcomes. Remarkably, our optimal model was fine-tunedwith the SGD optimizer, resulting in commendable performance and robust results.

Metrics:

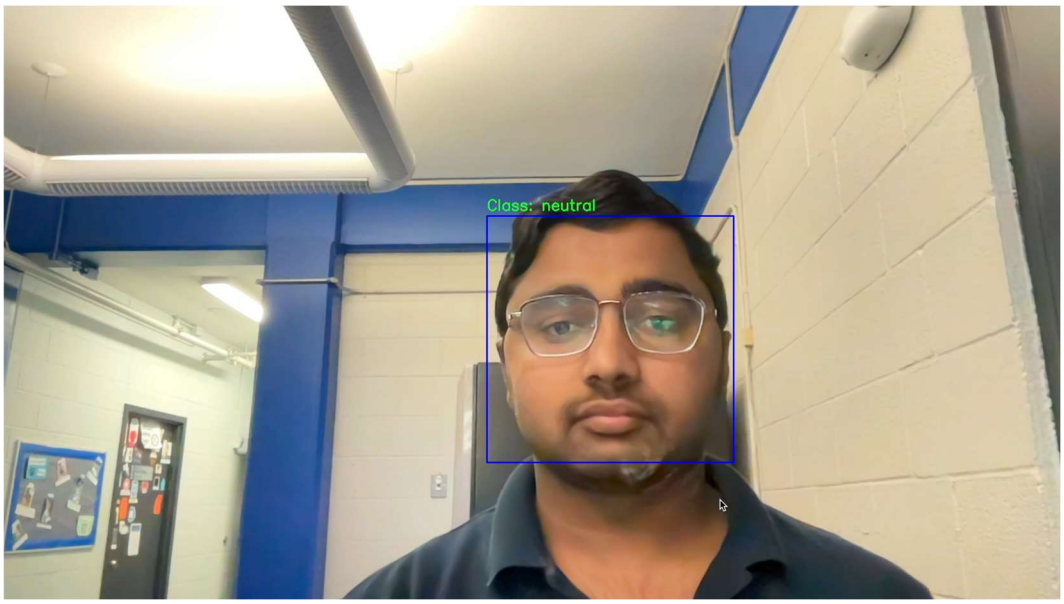


Fig 5: Model prediction from live web camera

Independent Study Report – Facial Emotion Recognition

In this study, we employed a diverse array of evaluation metrics, encompassing accuracy, precision, recall, F1 score, and the confusion matrix. Additionally, throughout the training process, we monitored key performance indicators, including accuracy and loss values for both the training and validation sets, tracked across epochs to gauge themodel's progression.

***Conclusion**:*

It is evident from our analysis that the classification of happy and surpriseexpressions is notably more accurate compared to other expressions, given their distinctiveness. Conversely, distinguishing between other expressions presents a more intricate challenge. Notably, our model demonstrates proficient classification of angry and neutral expressions. However, the analysis of the data indicates that our model encountered challenges in accurately discerning sad, neutral, and disgust expressions, as these were the expressions most frequently misclassified.

Github Respository Link:
https://github.com/dhanuhkumardk112/Independent-Study

References

1. Y. Ho and S. Wookey, "The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling," in IEEE Access, vol. 8, pp. 4806-4813, 2020, doi: 10.1109/ACCESS.2019.2962617.

2. Kaiming He,Xiangyu Zhang,Shaoqing Ren,Jian Sun," Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

3. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, https://arxiv.org/abs/1409.4842 //

4. Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, https://arxiv.org/abs/1505.04597//