Facial Emotion Recognition System

Yuying Li
EECS
University of Kansas
Lawrence, KS
yuyingli@ku.edu

Abraham Pascoe
MATH
University of Kansas
Lawrence, KS
abrahampascoe@ku.edu

Aidan Schmelzle
EECS
University of Kansas
Lawrence, KS
schmelzle@ku.edu

Jigyas Sharma
EECS
University of Kansas
Lawrence, KS
engineer@ku.edu

Sungmin Won
MATH
University of Kansas
Lawrence, KS
goretti.won@ku.edu

Bakhbyergyen Yerjan
EECS
University of Kansas
Lawrence, KS
yerjanb@ku.edu

***Abstract*— Advances in technology have enabled scientists to implement facial recognition into real-world scenarios. One such type of facial recognition, emotion recognition, has found applications in healthcare, surveillance, and entertainment. This new technology has been brought to fruition through a multitude of different models and approaches. This project explores the performance of several different methodologies including AlexNet, CNN, ResNet, DeiT, and finally DenseNet on a dataset consisting of over 28,000 images. Further, the project includes an analysis of the benefits and limitations of each approach.**

***Index Terms*— convolutional neural network, data-efficient image transformer, emotion recognition, facial recognition, computer vision**

I. Introduction

Ingraining facial recognition into our society has been rapid, and its permanence is inevitable. While there are many benefits to advancing this technology (airline security, suspect/ criminal investigation, use in the healthcare field, etc.), there are also associated privacy and consent concerns of high significance. Learning about this technology and how it works enables the next generation of research and industry professionals to be better equipped to face the technological and ethical challenges of furthering facial recognition. For this project, our team decided to use pre-existing models and methodologies as a foundation for our analysis. Namely, deep-learning in the form of a custom convolutional neural network (CNN), AlexNet, the data-efficient image transformer model (DeiT), VGG, ResNet, and advancements to the CNN model known as DenseNet were utilized in our project and will be analyzed here. Emphasis will be given to the AlexNet model as it outperformed the other models making it the subject of focus in this paper.

The remainder of this paper is organized as follows: Section II provides background on the models utilized in this study. Section III describes our data cleaning and image preprocessing procedures, which are critical for preparing the dataset for effective model training. In Section IV, we introduce our selected model, AlexNet, and begin our analysis. Section V delves into other methods we explored, offering a comparative insight into their performance and suitability. Section VI redirects to an evaluation of our proposed method, focusing on its effectiveness and the results obtained. Section VII discusses the limitations encountered during the project, and Section VIII sheds light on areas for future improvement. Finally, Section IX concludes the paper with a summary of our findings and their implications for the field of emotion recognition.

II. Background

A. ImageNet

ImageNet is a dataset that includes various classes such as nature, dogs, and other non-human categories. While our dataset consists exclusively of human faces classified into 7 different emotions, ImageNet was utilized in our project and will be referenced in our paper.

B. DenseNet

The Dense Convolutional Network, or DenseNet, model was proposed in 2018 as an advancement of classic convolutional neural network models [1]. It embraces the increasing depth of CNNs and leverages feature reuse through its connections between all layers. It may be favored over other models for its narrow layers, low computational demands, and dismissal of redundant feature maps. The layer structure is implemented in such a way that feature maps are very selectively introduced to the network's collective knowledge. DenseNet architecture boasts optimized parameter efficiency and an advanced flow of information which allows it to be trained easily. Historically, DenseNets produce compact and accurate models. In this project, DenseNet was not the most high-performing model. We considered multiple reasons for why this may be the case and attempted to implement fixes to boost accuracy. Though DenseNet was ultimately outperformed by other models, we hypothesized future work that could potentially allow the model to produce better results. These factors will all be discussed here.

C. DeiT

DeiT (Data-efficient Image Transformers) [2] developed by Facebook AI, is an optimized version of the Vision Transformer (ViT). It is designed to work efficiently with less data, using mechanisms similar to those in language processing rather than traditional convolutional layers. This approach makes DeiT particularly suitable for scenarios with limited resources and smaller datasets. Unlike conventional models that require extensive computational power and vast image libraries, DeiT achieves competitive results with accessible resources. By training on ImageNet with just a single computer, a programmer could expect to see such results in less than three days. DeiT introduces a novel teacher-student strategy with a 'distillation token,' allowing the student model to learn effectively from the teacher through attention mechanisms. This model's efficiency and performance make it a viable alternative to traditional CNNs, especially in resource-constrained environments. Consequently, we chose to include DeiT as one of the methods in our Facial Emotion Recognition project for this project.

D. ResNet50

ResNet-50 (Residual Networks 50) [3] is a deep convolutional neural network developed by Microsoft Research. The model is designed to overcome the vanishing gradient problem by incorporating residual blocks with skip-connections, allowing for efficient training of much deeper networks. Our project integrates this model to classify emotions from facial expressions, leveraging its robust feature extraction capabilities.

E. CNN

Convolutional Neural Networks (CNNs) [4] are a class of deep neural networks commonly used in analyzing visual imagery. Developed initially by researchers, including those at Microsoft, CNNs have been instrumental in advancing the field of computer vision. CNN image-processing through layers simulates the object and pattern recognition process of the human brain. CNNs automatically detect and use relevant features from images without human intervention. This makes them especially effective for tasks like image classification, facial recognition, and object detection. Their structure is designed to take advantage of the 2D layout of images, significantly reducing the number of parameters needed compared to fully connected networks of similar size.

F. AlexNet

The AlexNet [5] model is a convolutional neural network designed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. It was introduced in 2012 and quickly became a seminal work in the field of deep-learning for computer vision. AlexNet outperformed other models in the ImageNet Large Scale Visual Recognition Challenge by a large margin, showcasing the potential of deep neural networks in image classification tasks. AlexNet's design is noted for its use of dropout layers that help to prevent overfitting, making it robust for various vision-based tasks. The success of AlexNet has led to the widespread adoption of convolutional neural networks across many areas of computer vision.

III. Data Cleaning and Pre-processing

Upon inspection of the given dataset prior to model-training, we identified two major observations:

- The database has not been well-cleaned.
- The image size is not qualified for most state-of-the-art models.

To enhance model performance and to combat limitations imposed by these aspects of the data, we performed some data cleaning and image pre-processing. More details will be discussed in the following sub-section.

Fig. 1: Bad data samples. From left to right: sad, disgust, sad and neutral

A. Data Cleaning

In the given training dataset, each emotion category contained a certain amount of inapplicable images. We identified 4 types of images that were considered for deletion based on our own discretion and depending on severity:

- *Completely Blank Images:* These contribute no informative value and do not aid in training the model.
- *Heavily Watermarked Images:* Watermarks obscure facial features, which are crucial for accurate emotion recognition.
- *Emojis and Cartoons:* Illustrations are not appropriate for models being trained to recognize human facial expressions, as they do not represent real human features or expressions.
- *Indiscernible and Misclassified Images:* This category includes low-quality images that hinder the model's ability to learn as well as images incorrectly labeled as the wrong emotion, which could mislead the training process.

While we aimed to create a clean training set, we also recognized the importance of preserving the model's ability to handle similar images in future testing scenarios. Figure 2 illustrates the number of images before and after cleaning.

B. Pre-processing

The original images in the dataset are 48×48 pixels in grayscale, unsuitable for the 224×224 pixel RGB input required by our models. To address this, we resized the images to 224×224 pixels and converted them from grayscale to RGB by replicating the grayscale values across the three color channels. Additionally, we employed data augmentation techniques such as random horizontal flips, rotations, and affine transformations. These methods enhance the diversity of our training dataset and improve the model's ability to generalize across different facial expressions.

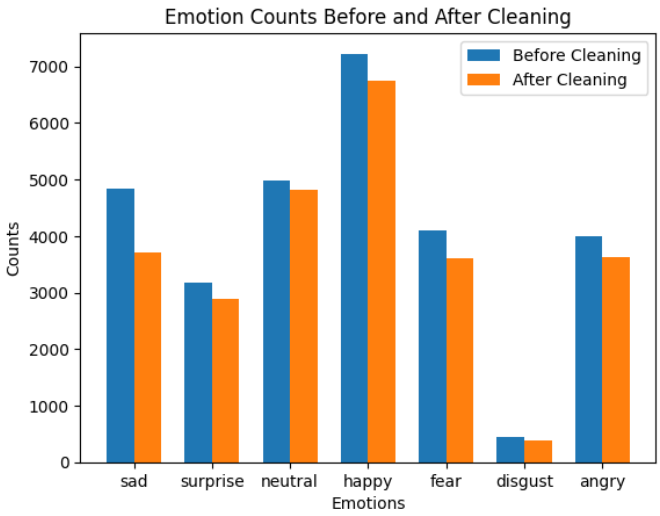

Fig. 2: Data Amount Before and After Cleaning

Layer (type)	Output Shape	Param #
conv2d_40 (Conv2D)	(None, 48, 48, 16)	160
batch_normalization_23 (BatchNormalization)	(None, 48, 48, 16)	64
max_pooling2d_27 (MaxPooling2D)	(None, 23, 23, 16)	0
conv2d_41 (Conv2D)	(None, 23, 23, 32)	4,640
batch_normalization_24 (BatchNormalization)	(None, 23, 23, 32)	128
max_pooling2d_28 (MaxPooling2D)	(None, 11, 11, 32)	0
conv2d_42 (Conv2D)	(None, 11, 11, 64)	8,256
conv2d_43 (Conv2D)	(None, 10, 10, 128)	32,896
max_pooling2d_29 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout_17 (Dropout)	(None, 8, 8, 128)	0
conv2d_44 (Conv2D)	(None, 8, 8, 128)	65,664
flatten_8 (Flatten)	(None, 8192)	0
dense_17 (Dense)	(None, 256)	2,097,408
dropout_18 (Dropout)	(None, 256)	0
dense_18 (Dense)	(None, 7)	1,799

Fig. 3: AlexNet model structure

IV. Proposed Method

The AlexNet model is a convolution neural network designed by Alex Krizhevsky et al [6], [7]. While this model is high-performing, it is also extremely expensive and demands a considerable amount of GPU power. We chose to select the AlexNet model as one method for this project because of its rich history as a catalyst that has spurred several other methods forward even though it has since been outperformed by other methods on other projects. The AlexNet architecture was made available via Kaggle and our group modified the foundational code to meet the requirements set forth by the project specifications and objectives [8].

In training the AlexNet model, we used max pooling on a 2D convolution of the images. For the convolution, we used layers with kernel sizes 2×2 or 3×3. In the model, we used an activation function of ReLu with a categorical cross-entropy loss function. For optimization, we used an Adam optimizer with a learning rate of 0.0001.

Figure 3 displays the specifics of the AlexNet model structure and demonstrates aspects of the model architecture.

V. Methods Attempted

In this section, we will introduce four other methods we used besides AlexNet. It is worth noting that for all methods, we used 80% of the data to train our respective models. Within the training data, 20% is reserved for validation, which comprises 16% of the total data amount. The remaining 20% serves as testing data.

A. DenseNet121

1) Architecture: The DenseNet121 model architecture follows a dense connectivity pattern, comprising multiple dense blocks and transition layers. It accepts single-channel 48x48 input images. Convolution layers followed by pooling layers create feature maps and reduce dimensions to prevent overfitting. Dense blocks concatenate input features, maintaining constant feature map dimensions within each block while altering filters for learning diversity. Transition layers manage feature-map reduction and spatial dimensions between dense blocks. The model consists of an input layer, convolution layer, four dense blocks (with 6, 12, 24, and 16 repetitions), and transition layers between each dense block.

2) Training: The training process consisted of three stages. In the first stage, the basic model was trained using all the data to observe learning rates and evaluate loss. Two training sessions were conducted: one with a batch size of 32 and 50 epochs, and another with a batch size of 128 and 250 epochs. The second stage involved freezing the first 15 layers to focus on learning deeper features and prevent overfitting on common features. Training and validation were conducted with a batch size of 16 and 128, and epochs of 50 and 100 respectively. In the third stage, 62 layers after the first two convolution layers were frozen. Training was carried out with batch sizes of 16 and 128, each for 40 epochs. Throughout each stage, two models were maintained—one trained with larger batch sizes and the other with smaller batch sizes—to determine which was more effective for the dataset.

3) Evaluation of DenseNet121: The evaluation of the model unfolds across three distinct stages. In the initial stage, testing accuracy gradually improves until it plateaus at the 40^{th} epoch with a value of approximately 0.53, while the corresponding loss stabilizes around 1.8. Concurrently, the training accuracy continues to ascend, and the associated loss progressively diminishes. During the second stage, the model demonstrates enhanced performance due to its prior exposure to the dataset. Although the accuracy initially shows an upward trend, it eventually levels off at 0.59. Similarly, the loss for validation data stabilizes at 1.4. The third stage exhibits a similar pattern in accuracy trends; however, a notable divergence is observed in the behavior of the loss for validation data, which continues to rise without impacting the model's accuracy significantly.

4) Limitations and Future Work: Limitations of DenseNet when related to the project database are both the single-

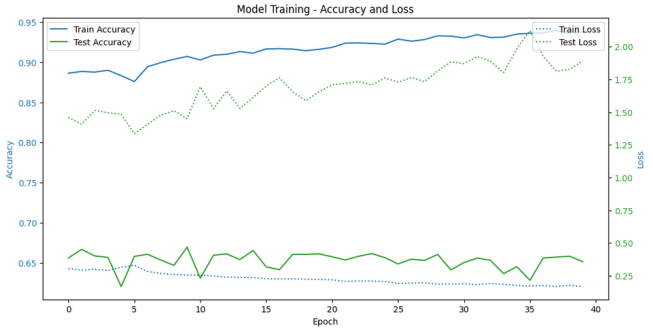

Fig. 4: Accuracy Post Training

channel image property and the 48x48 image pixel size of the supplied images. The single-channel aspect affects all models, but has specific detriment to DenseNet as it would ideally be able to extract information from 3-channel images. Further, DenseNet is based on 224x224 resolution for input images. These components together form less-than-ideal conditions for DenseNet's performance. The model will get a good picture in deeper layers, but not in the first few shallow layers.

Future work consists of training a model without transfer-learning on a larger dataset. We expect that attempting to implement the model with a larger supply of convolution layers before going inside dense blocks would enhance performance. Currently, the model we trained only has 1 convolution layer. Implementing this change may result in better performance, but of course the model size would also increase.

5) Conclusion: In conclusion, DenseNet was not an overwhelmingly poor match for this application and some small tweaks to factors outside of the model may have enhanced performance significantly. As mentioned in the Evaluation section, the model displayed an overfitting problem that is disproportionate to the size of the data. As the model consistently peaked at around 64% testing accuracy, it did not perform as well as other models we tested.

B. DeiT

1) Architecture: DeiT processes images by breaking them down into patches, similar to how words are treated in sentences for NLP. Specifically, we utilized deit_base_patch16_224, which divides each 224×224 image into patches of 16×16 pixels. Each patch is then flattened, linearly projected, and given positional encodings to retain spatial information. This approach allows DeiT to focus on the relationships between different image segments, enhancing its ability to process and understand the overall image structure.

The model's architecture comprises multiple transformer layers, each consisting of multi-head self-attention and feed-forward networks. The self-attention mechanism assesses the importance of different patches relative to one another, enabling the model to concentrate on the most informative parts of the image for classification tasks. This method is particularly effective for DeiT, as it can capture intricate details without requiring large-scale image inputs.

A unique feature of DeiT is the distillation token, part of its teacher-student training strategy. This token is treated as an additional input alongside the standard image patches and learns directly from the outputs of a pre-trained teacher model. The information gathered by the distillation token is then utilized by the classifier head to enhance prediction accuracy. This method effectively leverages the teacher model's knowledge, optimizing learning efficiency and making DeiT particularly effective in scenarios with constrained data and computational resources.

2) Training: Our training strategy leveraged the unique teacher-student learning approach of DeiT. Both the teacher and student models are versions of DeiT_base_patch16_224. We utilized a custom DistillationLoss function, which integrates Kullback-Leibler Divergence to minimize the difference between the teacher's and the student's softened outputs, with a traditional Cross-Entropy Loss that ensures the student model accurately predicts the actual labels. We set the distillation parameters with an alpha of 0.7 and a temperature setting of 1.0, optimizing the balance between following the teacher's guidance and learning from the actual data.

For optimization, the student DeiT model used an Adam optimizer with a learning rate of 1e-4 and weight decay of 1e-5. A ReduceLROnPlateau scheduler adjusted the learning rate based on the plateauing of the validation loss, maximizing learning efficiency. We also employed early-stopping to terminate training if no improvement in validation loss was observed over 20 epochs. This technique prevents overfitting and ensures that our training is computationally efficient. These methodologies together fortify the capabilities of our DeiT model in recognizing facial emotions effectively, even with our dataset's limitations.

3) Evaluation of DeiT: As shown in Figure 6, our training process demonstrated effective learning, with training accuracy peaking at approximately 98.5% and training loss stabilizing after initial improvements. In contrast, validation accuracy achieved a high of around 71.3%, and the validation loss displayed some fluctuations, indicating a reasonable but imperfect generalization to unseen data. This suggests that while the model learns well from the training set, there are challenges in achieving similar performance on the validation set. The training was terminated after 28 epochs due to early stopping. The subsequent testing phase resulted in a final accuracy of 71.21% on the test dataset.

4) Limitations and Future Work: The limitations of our model are primarily seen in its overfitting, as indicated by the disparity between training and validation results. The model's performance on underrepresented emotions shows an inadequate generalization across diverse emotions. For future work, addressing the imbalance in the training data and employing more different regularization techniques could enhance the model's robustness and accuracy across all classes.

5) Conclusion: In summary, the DeiT model can recognize data frequently represented in the training set. However,

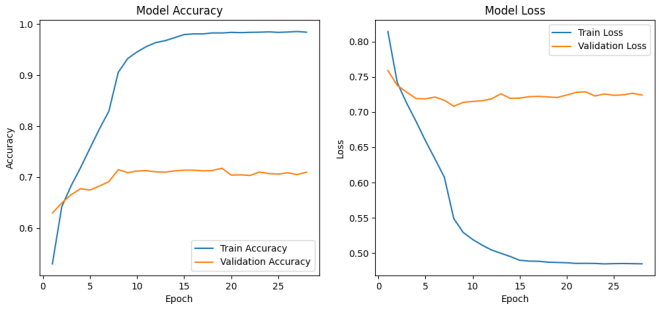

Fig. 5: DeiT Transfer Learning Accuracy and Loss Plots

it faced challenges with overfitting and effectively generalizing to less common emotions, evidenced by a significant discrepancy between training and validation accuracies. Therefore, given the superior performance on the testing set and other confusion matrix evaluations, we opted for the AlexNet model, which demonstrated better overall robustness and accuracy.

C. ResNet-50

1) Architecture: We adopted a transfer learning approach, utilizing a pre-trained ResNet-50 model trained on 1.2 million images from ImageNet. ResNet-50 follows a deep convolutional architecture, consisting of 50 layers including convolutional layers, residual blocks, and fully connected layers:

Residual Blocks: These blocks consist of three convolutional layers, accompanied by batch normalization and ReLU activations. The skip connection bypasses these convolutional layers, adding the original input directly to the output, effectively preventing gradient vanishing and promoting efficient learning. Global Average Pooling: After processing the input through a series of residual blocks, a global average pooling layer compresses the feature maps into a fixed-size vector, which is then passed to a fully connected layer for classification. The final fully-connected layer maps the global average-pooled vector to one of seven emotional classes, producing the final classification output.

2) Training: We applied various data augmentation techniques, including random flips, rotations, and zooms to diversify the dataset and enhance generalization.

I experimented with different hyperparameters and model variations:

- *Batch Sizes:* I tested batch sizes of 32, 48, 64, and 128, ultimately selecting 48 for the final report.
- *Learning Rate:* The model was trained with a learning rate of 0.0001, dynamically adjusted with a patience of 2 epochs, down to a minimum of 0.00001, using the Adam optimizer.
- *Dropout Layers:* These were incorporated to prevent overfitting.
- *Data Imbalance:* I attempted to address class imbalance by removing one class, "disgust," but this did not yield significant improvements.
- *Loss Function:* A categorical cross-entropy loss function minimized the difference between predicted labels and actual labels.

3) Evaluation of ResNet-50: After training, the ResNet-50 model achieved a training accuracy of 65.3% and a validation accuracy of 55.1%. The model's accuracy remained consistent across training epochs, indicating smooth convergence without oscillations.

Testing resulted in a test accuracy of 55.6%. The confusion matrix showed substantial accuracy in classifying 'happy', 'neutral', and 'surprise' emotions, while performance in distinguishing 'sad', 'angry', and 'fear' emotions lagged behind. These discrepancies reflect the dataset's imbalance and differences from ImageNet.

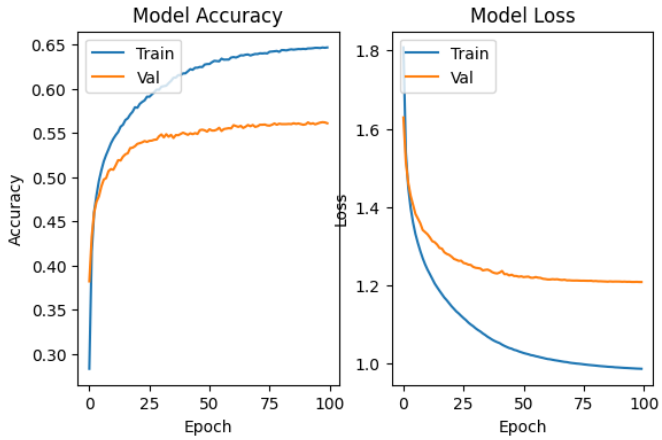

Fig. 6: ResNet-50 Transfer Learning Accuracy and Loss Plots

4) Limitations and Future Work: ResNet-50's limitations are seen in its inability to handle imbalanced datasets effectively, as reflected in its classification discrepancies. Future work includes addressing dataset imbalance and exploring additional regularization techniques to improve generalization across diverse emotions.

5) Conclusion: In summary, ResNet-50 demonstrated robust performance in facial emotion recognition, particularly for frequently represented emotions. However, its difficulty in distinguishing less-common emotions reveal its sensitivity to dataset imbalance. Given its overall accuracy, which remains competitive compared to state-of-the-art models, we opted not to propose it as the selected model due to its limited generalization.

D. Custom CNN

1) Architecture: In this project, we developed a custom Convolutional Neural Network (CNN) specifically designed for facial emotion recognition tasks. This model aims to balance complexity and performance, offering a tailored solution to the classification of emotions from a dataset of seven classes. The model's design is optimized to handle images efficiently and achieve high accuracy, making it a competitive choice for the project's requirements. The custom CNN consists of a multi-layered architecture, including convolutional, pooling, and fully connected layers:

- *Convolutional Layers:* The model starts with a series of convolutional layers, each followed by a ReLU activation function, which helps extract hierarchical features from the input images. These layers have varying kernel sizes, capturing features at different granularities.
- *Pooling Layers:* Max-pooling layers are introduced after every few convolutional layers, downsampling the feature maps and reducing dimensionality. This helps retain essential features while maintaining computational efficiency.
- *Fully Connected Layers:* The final convolutional block is followed by one or more fully-connected layers, transforming the flattened feature maps into a feature vector. This vector is then passed through a final fully connected layer that maps it to one of the seven emotion classes, producing the model's output.

2) Training: To train the model, we employed various data augmentation techniques, including rotations, flips, and color jittering, to diversify the dataset and enhance generalization.

The model was trained using a categorical cross-entropy loss function, minimizing the difference between predicted and actual labels. An Adam optimizer with a learning rate of 1e-4 and weight decay of 1e-5 was utilized for optimization. Furthermore, a learning rate scheduler was incorporated to adjust the rate upon plateauing validation loss, ensuring efficient training.

3) Evaluation of Custom CNN: After training, the custom CNN achieved a training accuracy of 90% and a validation accuracy of 55%. This indicates a sub-optimal balance between learning from the training set and generalizing to unseen data.

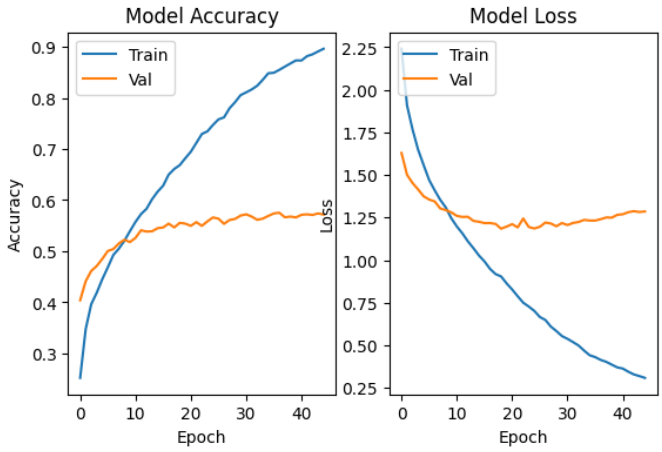

Fig. 7: Custom CNN Accuracy and Loss Plots

The testing phase resulted in a test accuracy of 55.8%.

4) Limitations and Future Work: The custom CNN's limitations are primarily seen in its inability to handle imbalanced datasets effectively. Future work includes addressing dataset imbalance and exploring additional regularization techniques to improve generalization across diverse emotions.

5) Conclusion: In summary, the custom CNN demonstrated robust performance in facial emotion recognition, particularly for frequently represented emotions. However, its difficulty in distinguishing less-common emotions demonstrates its sensitivity to dataset imbalance. Nonetheless, its overall performance makes it a viable option for this project, but improvements are needed to address its limitations.

VI. Evaluation

In this section, we discuss the evaluation of our proposed model, the AlexNet. As shown in Figure 8, our training process demonstrated effective learning, with training accuracy peaking around 80%. In contrast, the validation accuracy peaked around 75% which suggests that the model learns well from the training set. We halted the model after 200 epochs which resulted in a final accuracy of 75% on the training data and 69.05% accuracy on the validation set.

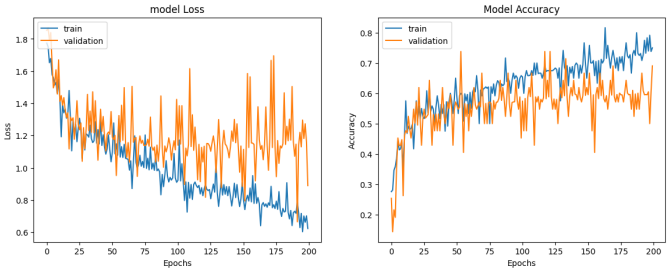

Fig. 8: Loss and Accuracy Post Training for AlexNet model

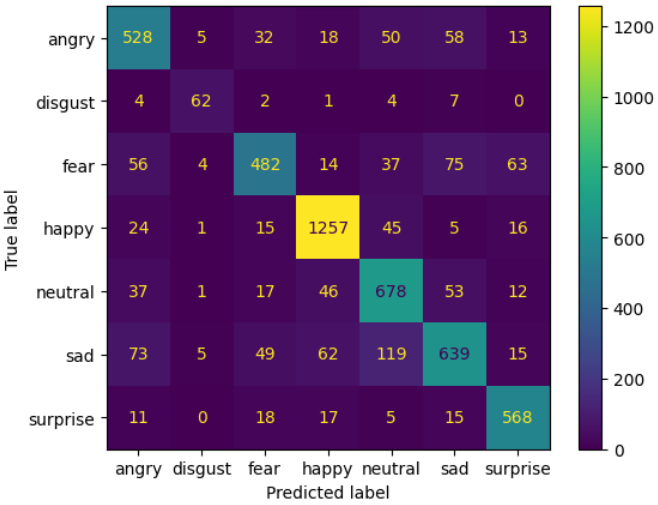

Fig. 9: Confusion Matrix of AlexNet model

When we applied the model to the test set, we observed an accuracy of 79.24%. The confusion matrix based on the test set (Figure 9), shows that the AlexNet model performed decently in classifying the various emotions. The entries along the diagonal of the confusion matrix are the largest in each of their respective rows and columns, which says that the model has a higher probability of correctly classifying an emotion than mislabelling an emotion. The 'happy' emotion was correctly classified more than any of the other emotions which aligns with what our other models determined. The 'sad' emotion was classified incorrectly most commonly with

the most confusion happening between 'sad' and 'neutral.' As mentioned in subsections of earlier models, the imbalance between emotion sets certainly contributes to misclassification.

VII. Limitations

1) Database Limitations: One of the drawbacks that contributed to lowering the performance of all models was related to the given database. Firstly, the 7 different emotion categories showed a highly skewed representation of data. Namely, the "disgust" category contained notably fewer images than the other emotions. Disgust had to be trained and tested on a total of only 393 images while the other 6 categories averaged 3,685 images. Even amongst those categories, there was notable variance in image count.

Class	Training	Validation	Testing	Class total	Class %
angry	2328	582	727	3637	14.1%
disgust	252	63	78	393	1.5%
fear	2310	578	721	3609	14.0%
happy	4314	1079	1348	6741	26.1%
sad	2371	593	741	3705	14.4%
surprise	1853	463	578	2894	11.2%
neutral	3078	770	962	4810	18.7%
TOTAL	16506	4128	5155	25789	100%

Fig. 10: Breakdown Table of Dataset

2) Hardware Limitations: Our project encountered significant hardware limitations, initially relying on the free version of Google Colab for training Convolutional Neural Networks (CNNs) and other Machine Learning (ML) models. The limited computational capacity available necessitated testing on personal laptops, with the lowest-end devices featuring an Intel i5 11th Gen processor, 20GB RAM at 2.5 GHz (8 CPUs), and shared memory with no integrated GPU. We sought assistance from the ITTC, who generously provided access to their Ampere machine. However, despite using an isolated Conda environment, the NVIDIA driver was outdated, hindering our progress. To overcome this hurdle, we upgraded to Google Colab Pro, which resolved the issue but limited us to 100 compute units, restricting further testing opportunities.

VIII. Conclusion

For this project, we attempted several models, the DenseNet121, the DeiT, the ResNet-50, a Custom CNN, and the AlexNet. In conclusion, the AlexNet model provided the highest overall accuracy and has the highest populated diagonal of the confusion matrix compared to the other models. Therefore, we propose the use of the AlexNet model. The DeiT model had the second-best accuracy, but had an issue with overfitting. We understood this to mean that under image processing, transformers do not perform as well as traditional CNNs.

While the accuracy of the AlexNet model is around 80%, which matches industry standards, we would like to build a model that performs better. Looking at future work, we would like to improve on the AlexNet model and truly understand why the model is misclassifying some images. In particular, we would like to gather extra data for these emotions (specifically disgust) to see if this increases accuracy for the AlexNet model.

Facial recognition systems have rapidly become a norm in our society. However, we have observed that the accuracy is not very high even for industry standards. As this is an extremely important topic in regards to privacy and safety, we want to continue looking at ways to improve the facial recognition system we have developed. Upon completion of this project, we are now more equipped to understand the technological and ethical aspects of facial recognition systems in terms of both challenges and benefits and are hopeful for the future advancements in this area of research.

IX. Future Work

To reiterate, future work may consist of training the model over alternative data sets. Additionally, the utilization of imitation-learning in combination with human input in the form of data aggregation may be employed in the creation of an optimized model that is able to combat over- and underfitting. Better resources could be used to train deeper layers on more valuable features.

References

[1] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.

[2] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," 2021.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[4] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, p. 84–90, may 2017.

[7] sanya9, "Emotion detection using alexnet." `https://www.kaggle.com/code/sanya9/emotion-detection-using-alexnet/notebook`, 2022. Accessed: 2024-05-02.

[8] K. A. Raj, "Alexnet cnn architecture on tensorflow (beginner)." `https://www.kaggle.com/code/vortexkol/alexnet-cnn-architecture-on-tensorflow-beginner/notebook`, 2020. Accessed: 2024-05-01.