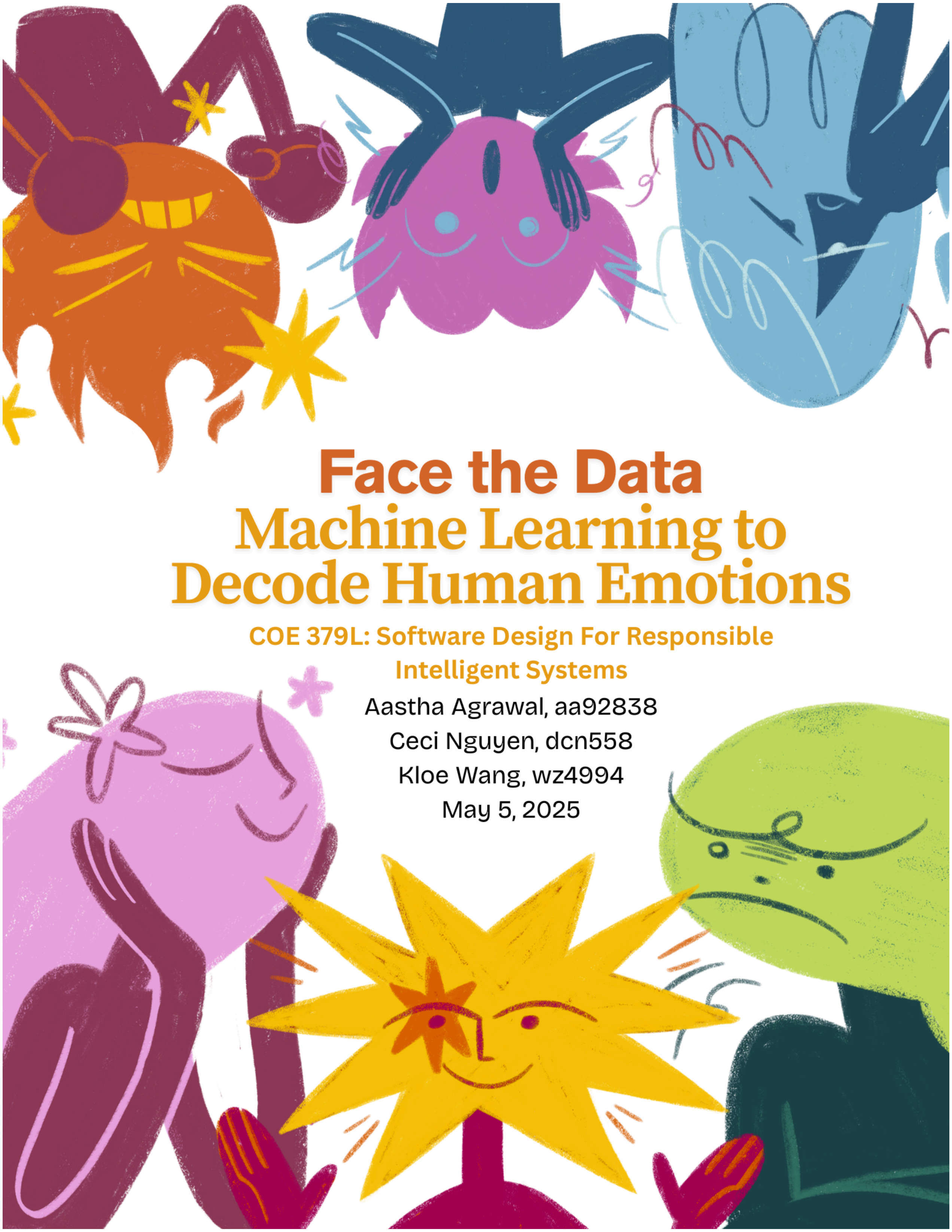


Face the Data

Machine Learning to Decode Human Emotions

COE 379L: Software Design For Responsible Intelligent Systems

Aastha Agrawal, aa92838
Ceci Nguyen, dcn558
Kloe Wang, wz4994

May 5, 2025

Table of Contents

Introduction and Problem Statement

Facial Expression Recognition (FER) is a branch of computer vision and affective computing that aims to enable machines to interpret and classify the emotional states displayed by people in images. The task involves taking an image of a human face and determining which one of seven fundamental emotions (Anger, Disgust, Fear, Happiness, Sadness, Surprise or Neutral) is being expressed. Although humans can generally easily recognize these expressions, developing an automated system that performs at a comparable level remains difficult. Real-world images may be low in resolution, partially obscured by objects such as glasses or scarves, or captured under uneven lighting conditions. At the same time, individual differences in facial structure, variations in head orientation, and the subtle nuances that distinguish, for example, a fearful expression from a surprised one, all contribute to the complexity of the problem.

In this project, we built an end-to-end FER pipeline using the FER2013 dataset, which provides tens of thousands of small grayscale face images labeled by emotion. Our objective was to train a model capable of achieving at least 70 percent classification accuracy. Beyond raw performance, we will pay careful attention to preprocessing steps, such as normalization, alignment, and data augmentation, to ensure that the model is resilient to common sources of noise. We will also explore different network architectures that balance accuracy with efficiency, recognizing that certain models with fewer parameters may be better suited for deployment on resource-constrained devices.

Data Sources and Technologies Used

For training and evaluating our models, we used the FER2013 dataset, a well-known public dataset hosted on Kaggle. The dataset consists of 35,887 facial images, each of size 48x48 pixels in grayscale format. Each image is labeled with one of seven emotion classes, namely Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The dataset is divided into training, validation, and test splits, although we manually verified and managed the splits within our code using Keras's ImageDataGenerator class. A major advantage of this dataset is its accessibility and relevance to the problem domain, but it also introduces challenges due to its low image resolution, potential label noise, and imbalanced class distribution, particularly the underrepresentation of the "Disgust" class. To support our workflow, we employed technologies such as Python, TensorFlow, and Keras for model development; NumPy and OpenCV for image processing and manipulation; and Matplotlib and Seaborn for visualization and performance analysis. Scikit-learn tools also played a role in computing performance metrics such as confusion matrices and classification reports.

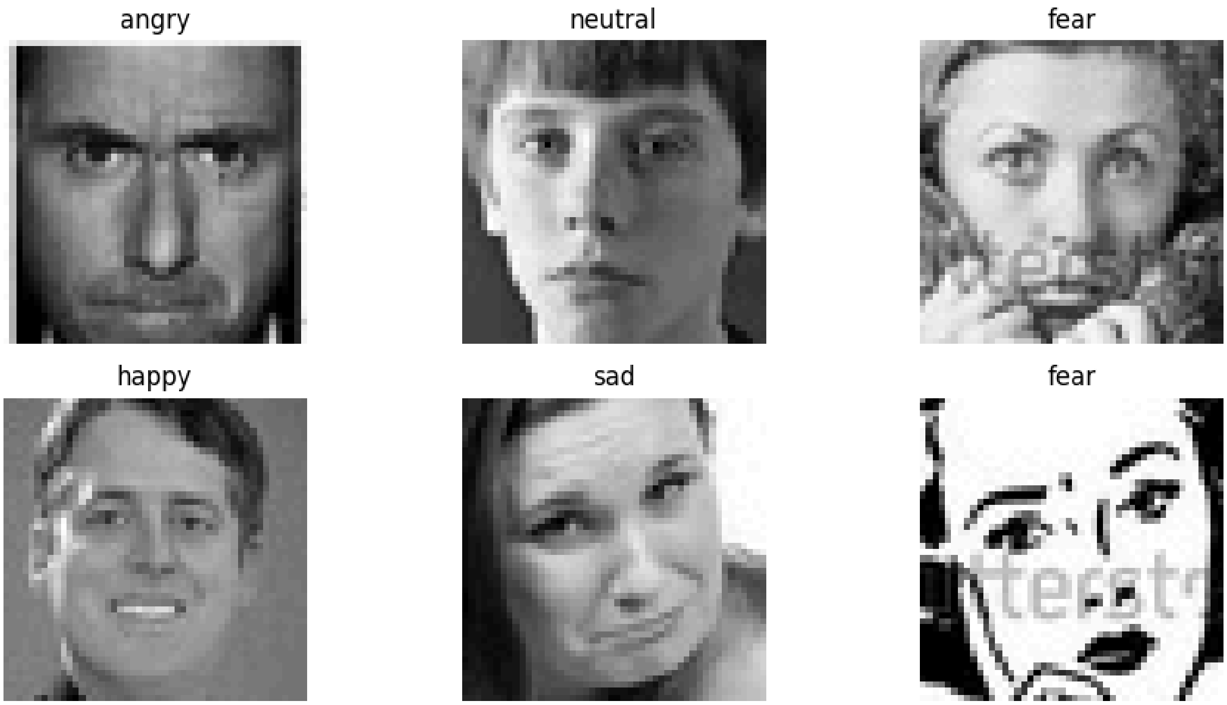


Figure 1: Examples of data for 7 of the emotions

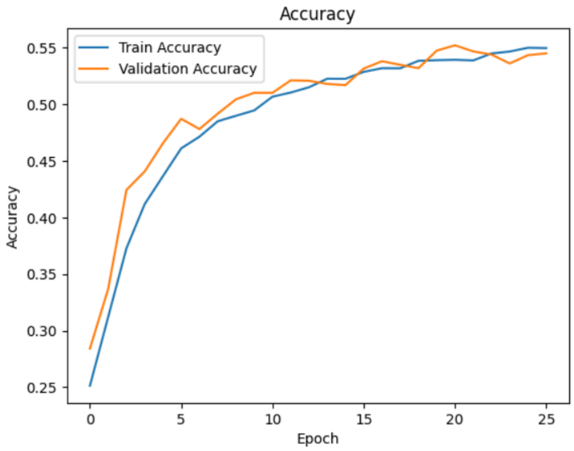
Preprocessing and Data Augmentation

The preprocessing stage involved normalizing the grayscale pixel values to the [0, 1] range and applying data augmentation techniques to enrich the training set. Given that the dataset includes a wide range of expressions with variations in pose, lighting, and facial structure, we implemented augmentation strategies to promote model generalization and reduce overfitting. Augmentations included random zooms, width and height shifts, horizontal flips, and brightness scaling. This approach not only increased the effective size of the training data but also exposed the model to more varied input conditions, simulating real-world unpredictability. Data augmentation was crucial in preventing the model from memorizing training samples and helped improve its ability to classify unseen images.

Model Architectures

We explored three primary deep learning architectures: a basic CNN, a ResNet-style model, and a VGG-style model. The basic CNN acted as our baseline and consisted of three convolutional layers with ReLU activations, max pooling, and dropout regularization. It was quick to train and reasonably effective, achieving a final test accuracy of 58%. The ResNet-style model, inspired by He et al.'s deep residual learning approach, introduced skip connections that enabled more effective training of deeper layers. This model slightly improved on the baseline with an accuracy of approximately 60%.

The best-performing model was the VGG-style architecture, which achieved a test accuracy of around 62%. This architecture used a deeper stack of convolutional layers with small (3x3) filters, interleaved with max-pooling layers. Its strength came from its ability to capture hierarchical spatial features, which allowed for better discrimination between similar facial expressions. However, this came at the cost of longer training times and a greater tendency toward overfitting, which we mitigated with data augmentation and dropout. The improved performance suggests that deeper convolutional hierarchies were more effective for this specific task, even with relatively low-resolution input.



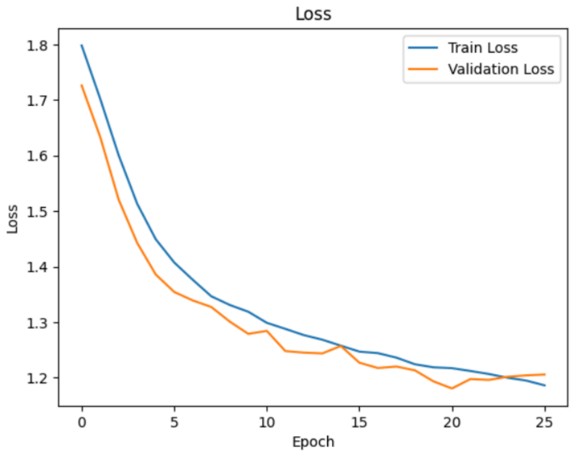
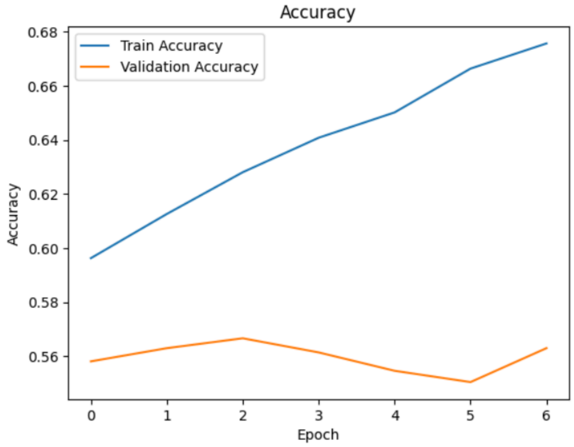


Figure 2 (CNN): Training and validation accuracy over 25 epochs. The curve rising steadily and converging indicates the model improving without overfitting

Figure 3 (CNN): Training and validation loss over 25 epochs. The smooth decline and stabilization reflects learning and generalization



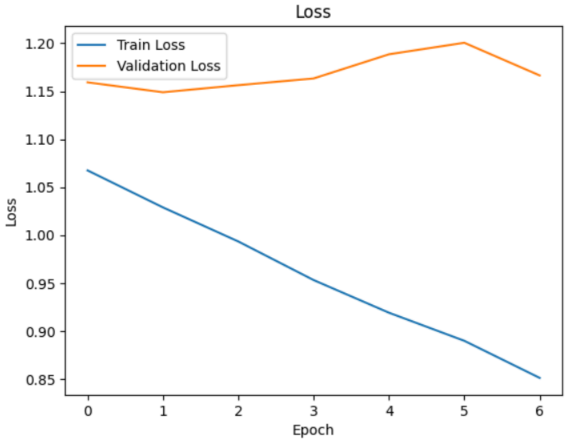
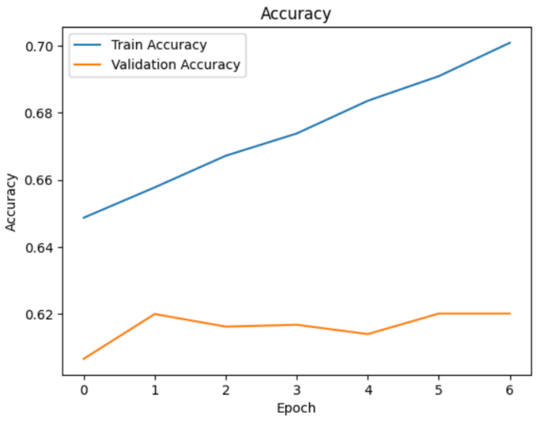


Figure 4 (ResNet): Training and validation accuracy over 6 epochs. Validation accuracy plateaus around 0.56.

Figure 5 (ResNet): Training and validation loss over 6 epochs. Validation loss drifts upwards.



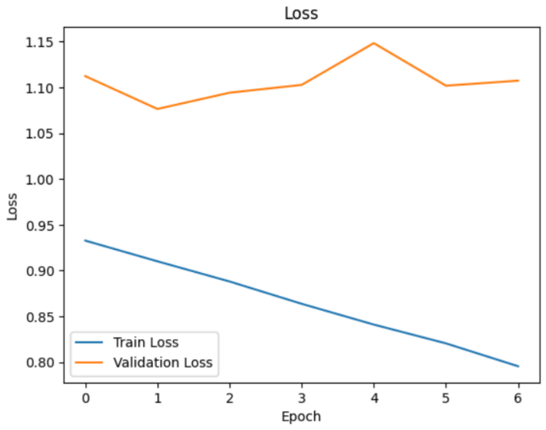


Figure 6 (VGG): Training and validation accuracy over 6 epochs. The model's training accuracy steadily increases.
Figure 7 (VGG): Training and validation loss over 6 epochs. Training loss decreases over 6 epochs

Results and Observations

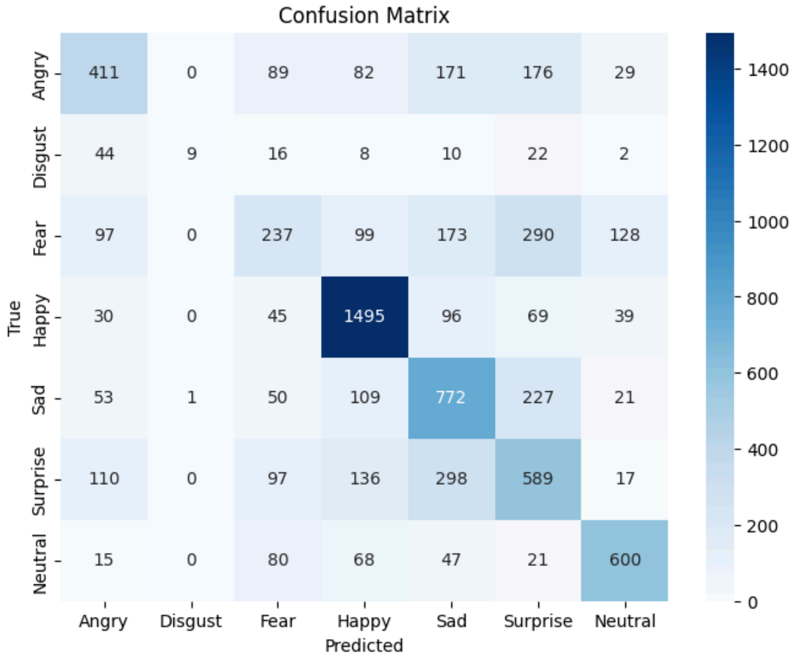
Our experimental results revealed a clear performance trend across the three model architectures. The basic CNN model achieved a test accuracy of approximately 58%, which set our performance baseline. Incorporating residual connections in the ResNet-style model boosted the accuracy to around 60%, likely due to the improved gradient flow during training. However, the highest test accuracy was achieved by the VGG-style model at roughly 62%. Despite having no residual connections, its architectural depth and fine-grained convolutional structure proved most effective for capturing the subtle variations between facial expressions.

Interestingly, while residual learning helped the ResNet-style model avoid degradation in deeper layers, it did not outperform the VGG-style network in this particular task. This outcome may reflect the fact that FER2013's image resolution is low enough that very deep architectures do not yield proportionally higher benefits unless carefully tuned. The VGG-style model struck a more effective balance between depth and feature expressiveness, resulting in better classification across most emotion classes.

Although the VGG-style model outperformed the others with a test accuracy of 62%, this still fell short of the 70% benchmark we initially set. Several factors contributed to this result. The 48x48 pixel image size significantly limited the amount of facial detail available for the model to learn from, reducing its ability to differentiate subtle expressions like fear versus surprise. Additionally, the dataset suffered from label noise and class imbalance, especially for less frequent emotions like disgust, which likely introduced bias into the learning process. While the VGG-style model had the most success due to its architectural depth and strong feature extraction capabilities, it still struggled with generalization on edge cases, likely due to data limitations rather than model capacity.

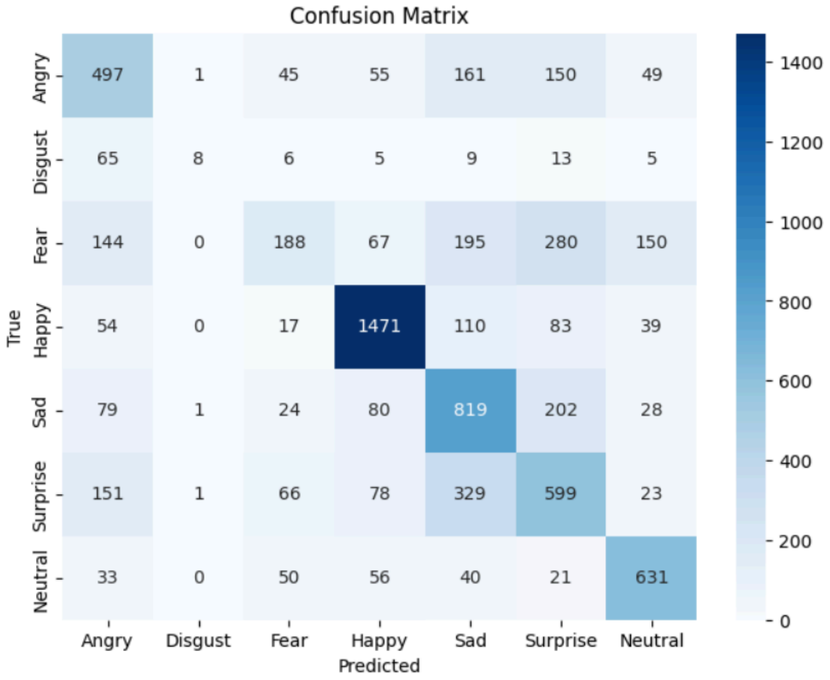
Moreover, we were constrained to training models from scratch without access to large-scale pretraining or transfer learning. Many top-performing systems in FER tasks rely on transfer learning from massive facial datasets or ensembles, neither of which we implemented.

Future work that incorporates these advanced techniques may close the remaining gap to 70% accuracy or beyond.



True \ Predicted	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	411	0	89	82	171	176	29
Disgust	44	9	16	8	10	22	2
Fear	97	0	237	99	173	290	128
Happy	30	0	45	1495	96	69	39
Sad	53	1	50	109	772	227	21
Surprise	110	0	97	136	298	589	17
Neutral	15	0	80	68	47	21	600

Figure 8 (CNN): Confusion matrix illustrating true versus predicted emotion counts. The model shows strong performance on "Happy" and "Neutral", while "Fear", "Disgust", and "Angry" are more frequently misclassified.



True \ Predicted	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	497	1	45	55	161	150	49
Disgust	65	8	6	5	9	13	5
Fear	144	0	188	67	195	280	150
Happy	54	0	17	1471	110	83	39
Sad	79	1	24	80	819	202	28
Surprise	151	1	66	78	329	599	23
Neutral	33	0	50	56	40	21	631

Figure 9 (ResNet): Confusion matrix illustrating true versus predicted emotion counts. The model shows strong performance on "Happy","Neutral", "Sad". and "Surprise", while "Fear" and "Disgust" are more frequently misclassified.

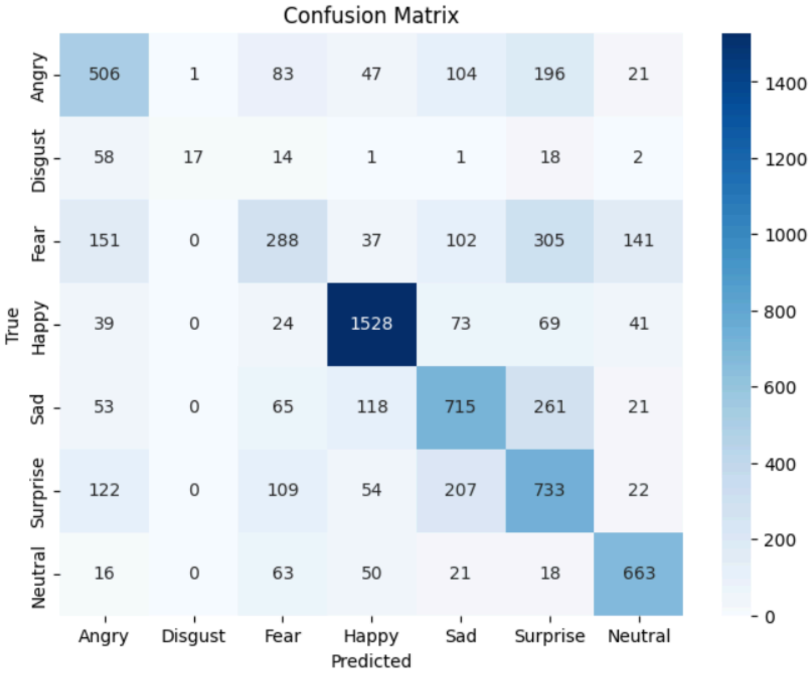


Figure 10 (VGG): Confusion matrix illustrating true versus predicted emotion counts. The model shows strong performance on "Happy","Neutral", "Sad". and "Surprise", while "Fear" and "Disgust" are more frequently misclassified as Angry or Surprise.

Inference Server

The inference server is implemented as it loads our trained FER2013 model at startup and exposes a single `/predict` endpoint. Clients submit images (via multipart form-data or base64), and the server automatically converts them into 48×48 grayscale face crops, normalizes pixel values, and feeds them through the neural network. The response is a concise JSON payload containing the top emotion prediction and confidence scores for all seven classes. All dependencies (TensorFlow/Keras, OpenCV, etc.) are specified in `requirements.txt` and containerized with a straightforward `Dockerfile`, enabling one-command deployment to any cloud VM or Kubernetes cluster. By decoupling training from serving, we can seamlessly roll out updated model checkpoints without modifying the API code, while ensuring low-latency, scalable inference behind a load balancer.

Reflections and Future Work

Although we did not meet our predefined accuracy goal, this project yielded valuable insights into the design and training of CNNs for facial emotion recognition. We confirmed that deeper and more structured models like VGG provide tangible benefits over simpler or residual-based networks for this dataset. We also learned that while data augmentation and regularization are helpful, there is a ceiling to what they can achieve when the data itself is limited. Moving forward, the most promising direction for improvement would involve using transfer learning with pre-trained models such as ResNet50 or MobileNet, which have already learned rich, generalizable visual features. Additionally, incorporating more expressive datasets like AffectNet or the CK+ dataset could provide better-quality labels and higher-resolution data. Other improvements could include the use of attention-based architectures to focus on key facial regions and ensemble methods to stabilize predictions. These steps would likely yield significant gains in accuracy and reliability, especially in real-world applications.

Conclusion

This project explored the development of a facial expression recognition pipeline using deep learning models trained on the FER2013 dataset. Our experimentation with CNN, ResNet, and VGG-style architectures demonstrated that deeper models with strong structural design, such as the VGG-style network, perform best on emotion classification tasks, achieving a final test accuracy of 62%. While this result falls short of the 70% benchmark, it reflects a meaningful outcome given the limitations of the data and the models' training from scratch. Overall, the project provided a rich opportunity to engage with real-world challenges in computer vision and build practical skills in model evaluation, design, and deployment for emotion recognition systems.

References

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. arXiv:1512.03385. https://arxiv.org/abs/1512.03385

Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv:1409.1556

FER-2013 Dataset. Kaggle. https://www.kaggle.com/datasets/msambare/fer2013

Presentation and Video

Here is our presentation slides and presentation video!

https://www.canva.com/design/DAGmR3EtofY/Jx141lduKX6qBmcJQ8h-7Q/edit?utm_content=DAGmR3EtofY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

https://utexas.zoom.us/rec/share/5FhY26xxM1-WWFTe6mno0ZCiAxg5o4_3cWQ_-9Zr9hlL8kCjl4l8fTP5eagnYzo.vw-nKEueFgDBjWQX