Emotion Classification

DATASCI 281, Computer Vision
Students: Diego Moss, Subhasis Das, Priscilla Miller

1. Introduction

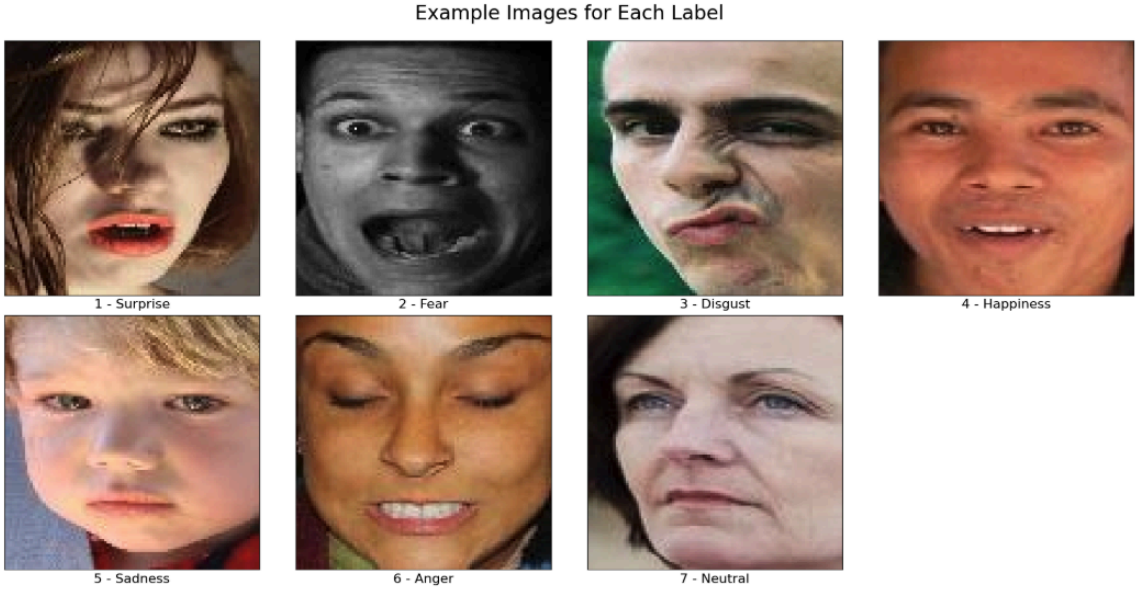
For many neurodivergent individuals, interpreting facial expressions and emotional cues presents a significant challenge in daily social interactions. Our project addresses this challenge through the development of a multi-class classification model capable of identifying expressions associated with seven distinct emotions: *surprise*, *fear*, *disgust*, *happiness*, *sadness*, *anger*, and *neutral*. This project serves as a preliminary proof-of-concept for an accessibility application that could be integrated into smart-glasses devices. The intended users are neurodivergent people who struggle with detecting emotions and social cues from nonverbal language. This project encompasses the initial component of a finalized video-to-label application, where real-world images are fed to an emotion classifier, the outputs of which are used to briefly explain emotional context to a user.

While the ultimate vision is to create a video-to-label application that provides real-time emotional context, the current project focuses specifically on the preliminary development of the core emotion classification component. To achieve this, we leverage the Real-world Affective Faces Database (RAF-DB), comprising approximately 30,000 diverse facial images. For this project, we used the single-label subset (basic emotions) of the RAF-DB.

2. Data Overview

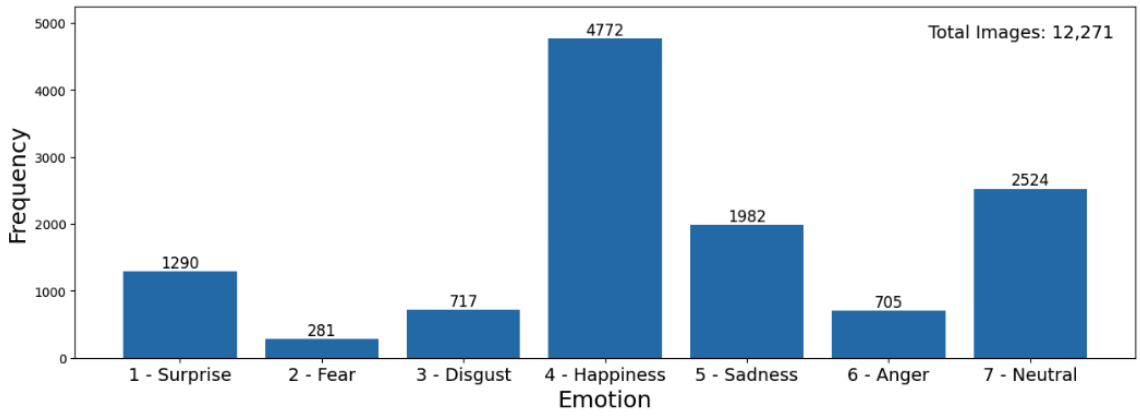
2.1 Chosen Dataset

The RAF-DB dataset was chosen because it provides diverse, naturally occurring facial expressions rather than posed ones in controlled settings, making it ideal for our future application aimed at helping neurodivergent individuals interpret real-world emotional cues (Li & Deng, 2019). The images vary greatly in subjects' age, gender, ethnicity, head poses, lighting conditions, and occlusions (e.g., glasses, facial hair). The images were pulled from URLs sourced from Flickr using its image search API, and an open-source batch downloader retrieved the images. Each image was independently annotated by approximately 40 contributors through a crowdsourcing process, which included reliability scoring via an expectation-maximization algorithm. This allowed for the generation of high-confidence emotion labels, which were determined via passing sufficient reliability score thresholds. This annotation approach acknowledges the inherent subjectivity of emotion recognition, and the real-world variability in the dataset should enhance the model’s generalizability to recognize facial expressions in diverse scenarios.



The single-label RAF-DB sample encompasses 12,271 training and 3,068 testing images, and facial images are annotated with the seven basic emotion labels as outlined above. The training sample has a skewed distribution of image classes, with the *happiness* label overrepresented (4772 images) and *fear* heavily underrepresented (281 images). To account for this heavy skew, we implement data augmentation on underrepresented groups to better equalize the distribution of classes. Data augmentation and other pre-processing steps are outlined below.

Number of Training Images per Category (pre-augmentation)

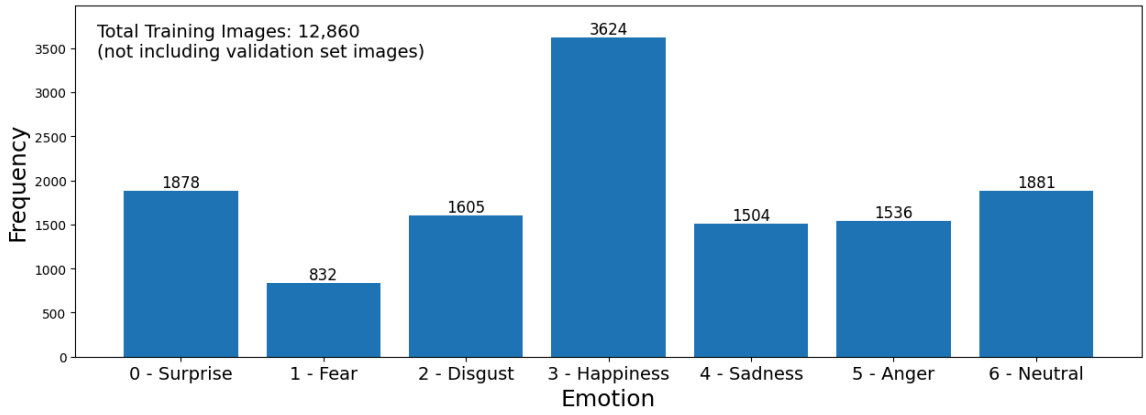


2.2 Preprocessing

All images underwent a series of preprocessing steps. During the creation of the dataset, image faces were aligned to standardize orientation across image samples. Each face was aligned based on the centers of the eyes and the mouth to ensure that facial features were consistently positioned across the dataset. Images were also sized to a uniform resolution of 100×100 pixels, maintaining consistent input dimensions for the model. To mitigate visual noise, we applied light denoising techniques, with an emphasis on preserving edges to maintain important facial structure while reducing irrelevant variation.

Finally, to improve generalization and address class imbalance, data augmentation techniques were applied. Prior to data augmentation, we took 25% of the original training data and set it aside as a validation set to use during model training. We then selectively applied transformations to the remainder of the training set, focusing only on images from underrepresented classes. These transformations included horizontal flips, controlled noise, and adjustments in brightness and contrast, effectively increasing the diversity of training samples. For each image, up to three augmented versions could be created—one per transformation.

Number of training Images per Category (post-augmentation)



3. Feature Extraction

3.1 Simple Features

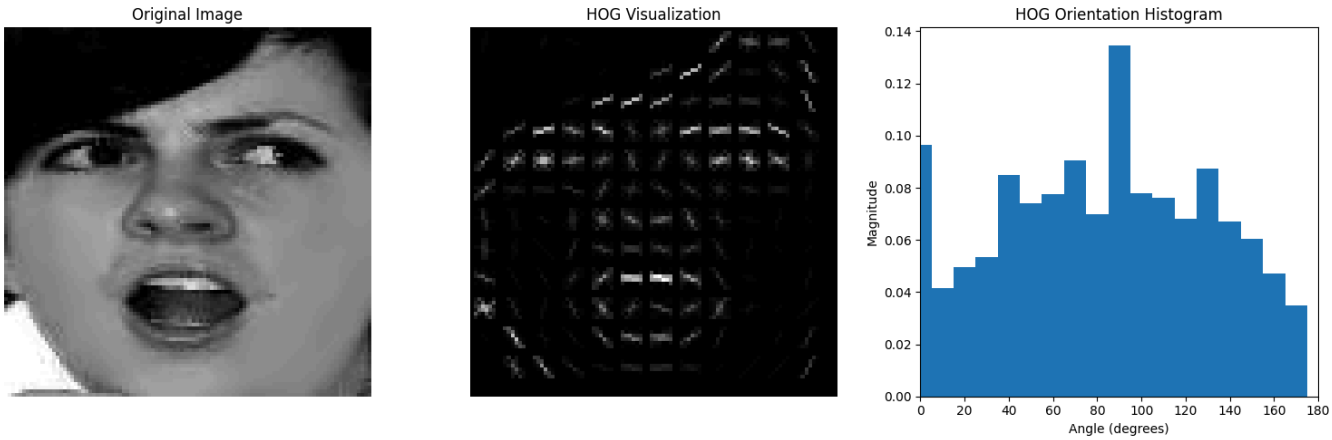
For feature extraction, we explored the following:

Histogram of Oriented Gradients (HOG): We utilized HOG features to extract the structural geometry of facial expressions. Our goal was to preserve the overall shape of the face—even when key regions were partially occluded due to hands, glasses, or shadows. Since HOG works by capturing the distribution of intensity gradients and edge directions, it remains effective even under varying lighting conditions and mild blurring. This made it a solid candidate for our dataset, which contains real-world variability in image quality and illumination.

To determine the optimal HOG configuration, we experimented with several combinations of cell sizes and orientation bins:

- Standard (8×8 cell, 9 orientations)
- Smaller cell sizes like 4×4 and 2×2
- Increased orientations (12, 18, and 20 bins)

Each configuration was visualized to assess clarity and gradient sharpness. While smaller cells and more orientations captured finer textures, they also significantly increased feature dimensionality and noise. Through visual inspection and evaluation of classification performance, we found that using 8×8 pixel cells with 18 orientation bins provided the best balance between expressive detail and computational efficiency.



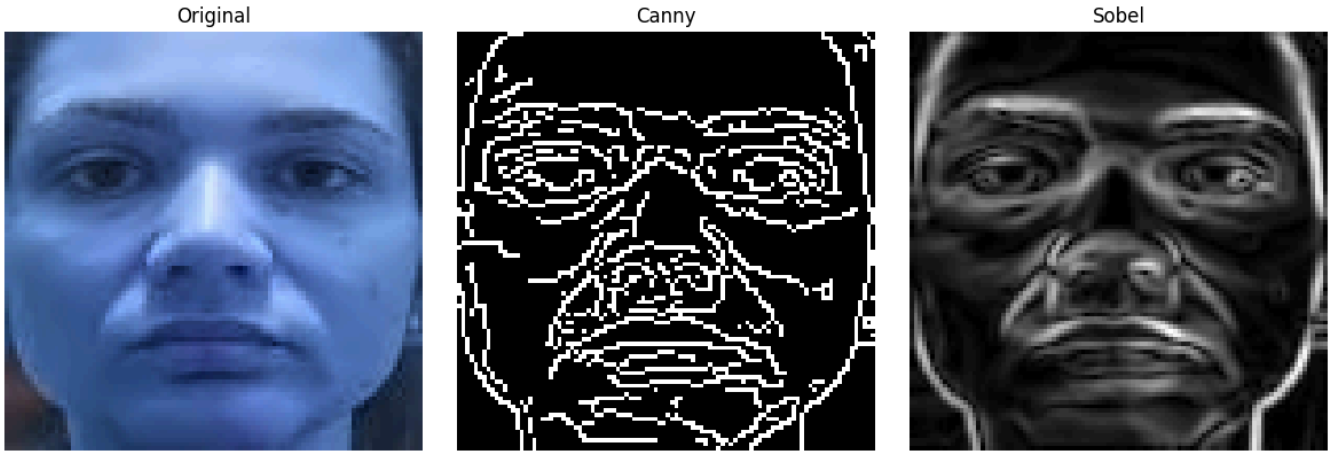
This final configuration allowed us to encode critical shape information, such as jawline, eyebrow, and cheek curvature, while minimizing redundancy, serving as a foundational low-level feature in our multi-feature fusion pipeline.

Edge Detection (Canny and Sobel): We incorporated edge detection to highlight sharp intensity changes in facial regions—such as the contours of the mouth, eyes, and brows—that are critical for expression analysis. Unlike HOG, which focuses on gradient orientation across regions, edge detection methods produce explicit boundary maps, helping isolate the outlines of facial components.

To evaluate their effectiveness, we experimented with both Canny and Sobel filters across multiple parameter settings:

- Canny thresholds: (50, 100), (100, 200), (150, 250)
- Sobel kernel sizes: 3, 5, and 7

For each setting, we applied edge detection to both the original grayscale images and their denoised counterparts. This helped us assess how noise and fine detail affected edge clarity. We visualized the results side by side, noting how Canny produced crisp, sparse edges while Sobel revealed smoother gradients and finer structures depending on kernel size.



After visual inspection and preliminary model evaluation, we selected Sobel edge detection with a 5×5 kernel. Prior to applying the filter, each image underwent grayscale conversion and histogram equalization to enhance contrast, improving edge clarity across varied lighting conditions. The filter consistently captured clean contours and localized boundary cues, which complemented the global structure from HOG and the texture sensitivity of Gabor filters. These edge maps were flattened and evaluated as part of the combined feature vector. However, after further experimentation, we found that the Sobel-derived features significantly overlapped with those of the ResNet features. As a result, we ultimately excluded Sobel edges from the final feature set to avoid redundancy and streamline the representation.

Gabor Filters: To capture fine-grained textures and spatial frequency information critical for emotion recognition, we incorporated Gabor filters into our feature extraction pipeline. Gabor filters are particularly effective at mimicking the early stages of human visual perception, making them well-suited for detecting micro-expressions, wrinkles, and localized muscle movements—such as frown lines or forehead tension—that are not easily captured by shape- or edge-based methods.

We applied Gabor filters across multiple orientations to account for directional variations in facial features. Specifically, we experimented with:

- Orientations (θ): 0°, 45°, 90°, 135°
- Frequencies and bandwidths: fixed across experiments for consistency
- Original grayscale vs. denoised grayscale images

Each filtered image captured textures aligned to a specific orientation. For example, 90° filters enhanced horizontal texture cues like smile lines, while 45° filters highlighted diagonally oriented wrinkles and creases. We visually compared these outputs and observed that denoised grayscale inputs produced more coherent filter responses with fewer false activations in flat regions.

The final configuration used responses from all four orientations, which were then flattened and concatenated to form a comprehensive texture descriptor. This approach allowed us to encode spatial patterns at different directions, increasing the expressiveness of our feature set.

Gabor features provided an important layer of complexity in our pipeline, bridging the gap between low-level edge detection and high-level deep features. When combined with other descriptors, they improved the model's ability to differentiate between similar expressions (e.g., *fear* vs. *surprise*) by focusing on subtle surface-level cues.

3.2 Complex Feature

While traditional feature extraction methods like HOG, edge detection, and Gabor filters capture low- to mid-level visual cues, they rely on handcrafted operations and are often limited in their ability to represent abstract, high-level patterns in the data. To complement these methods, we incorporated a deep learning–based feature extractor using a pretrained ResNet-18 model.

We used the ResNet architecture not for classification, but purely as a feature encoder. Specifically, we removed the final classification layer and extracted features from the last convolutional block. This produced a 512-dimensional embedding for each input image, capturing rich hierarchical representations learned from large-scale datasets (ImageNet). These features include patterns such as cheekbone shapes, eye-muscle tension, jaw movement, and other complex relationships that are hard to design manually.

To prepare images for ResNet input, we normalized them using the standard ImageNet mean and standard deviation values and resized them to the required resolution. We also ensured grayscale images were converted to 3-channel format to match the model's expected input.

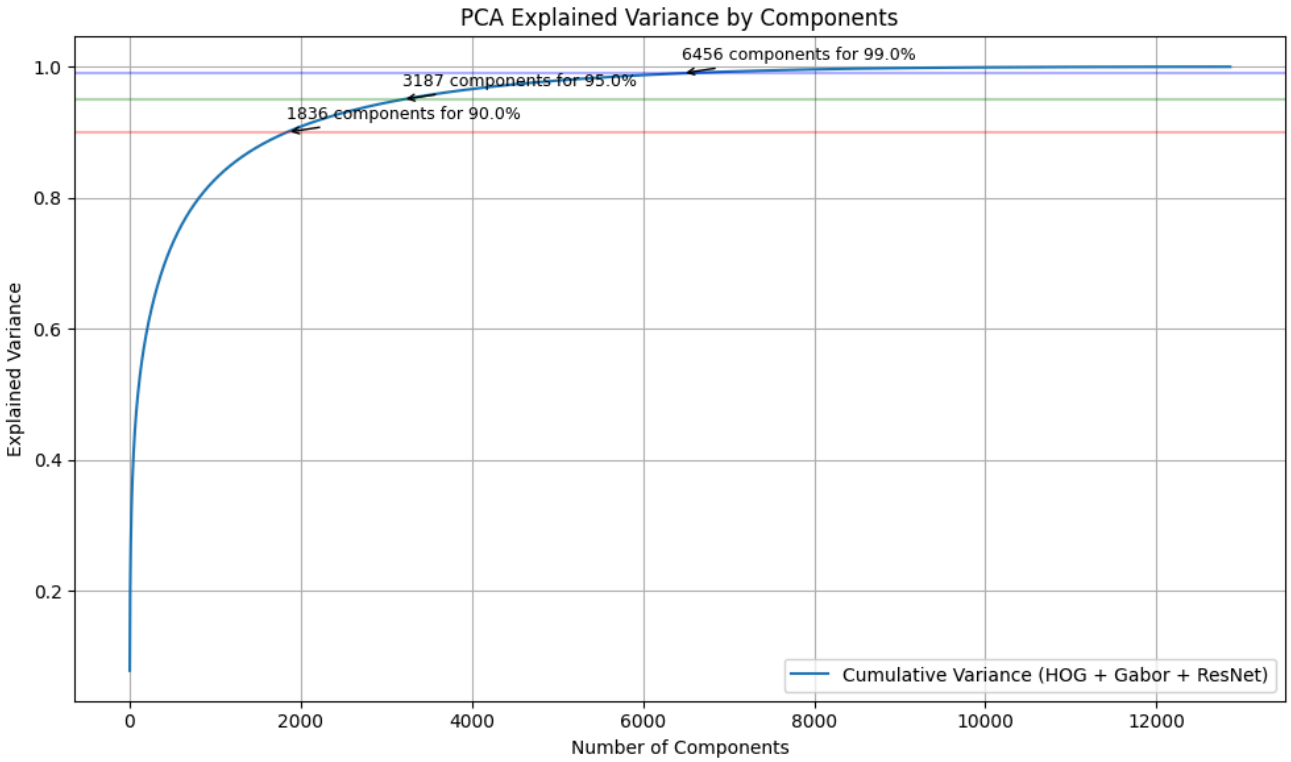
Unlike traditional methods, ResNet does not rely on explicit edge or texture filters. Instead, it learns features from data, making it capable of identifying subtle and distributed cues associated with emotional expressions, even when those cues are not visually obvious.

Including ResNet features in our final feature vector allowed us to blend handcrafted and learned representations, resulting in a more expressive and discriminative model. These deep features added a powerful layer of abstraction that significantly improved performance, especially in distinguishing visually similar but emotionally distinct expressions.

3.3 Feature Analysis

After combining all extracted features—including HOG, Gabor, and ResNet—the resulting feature vectors became very high-dimensional, exceeding 19,000 dimensions per image. Working with such high-dimensional data not only increases computational complexity but also introduces noise and redundancy, which can hurt model performance.

To address this, we applied Principal Component Analysis (PCA) as an unsupervised dimensionality reduction technique. PCA helped us retain the most significant variance in the data while reducing dimensionality. Specifically, we fit PCA on the training set (after scaling) and applied the transformation to the training, validation, and test data. This step reduced our feature space from ~19,000 dimensions to 1,836, capturing over 90% of the variance.



To gain further insight into how expressive our features were, we visualized the reduced data using t-SNE (t-distributed Stochastic Neighbor Embedding). t-SNE is a non-linear embedding technique that maps high-dimensional data into 2D or 3D space while preserving local neighborhood structure.

By plotting the PCA-reduced features using t-SNE:

- We color-coded each point by its emotion class label.
- Ideally, expressive features would form clear, distinct clusters by emotion.
- However, we observed significant overlap between certain emotions—especially *fear*, *disgust*, and *surprise*—highlighting the challenge of classifying subtle facial expressions.
- Emotions like *happiness* and *sadness* showed better separation, indicating more consistent visual cues across samples.

These visualizations confirmed that while our feature set captured useful information, there was still room to improve class separation—through additional fine-tuning, better balancing, or more expressive feature engineering.

4. Classification Models

We investigated the performance of three different machine learning approaches for emotion classification. This section details the implementation and evaluation of each model, along with insights gained from their performance across training, validation, and test sets.

4.1 Neural Network (Multi-Layer Perceptron)

Starting with a basic single-layer perceptron as a baseline for this model, we expanded to a more complex neural network architecture, incorporating dropout, batch normalization, and weight regularization in an effort to limit overfitting. We searched over architectural and training hyperparameters, and selected the best-performing configuration with early stopping based on validation loss.

To optimize the performance of the MLP model, we conducted a randomized hyperparameter search with the goal of identifying an architecture and regularization strategy that achieved strong generalization on the validation set while minimizing overfitting.

Across 16 different configurations, we tested the following hyperparameters:

- Learning Rate: .001, .0001
- Dropout Rate: 0.3, 0.5
- L2 Regularization (Weight Decay): 0.001, 0.01
- Hidden Layer Configurations: (32,), (64, 32)

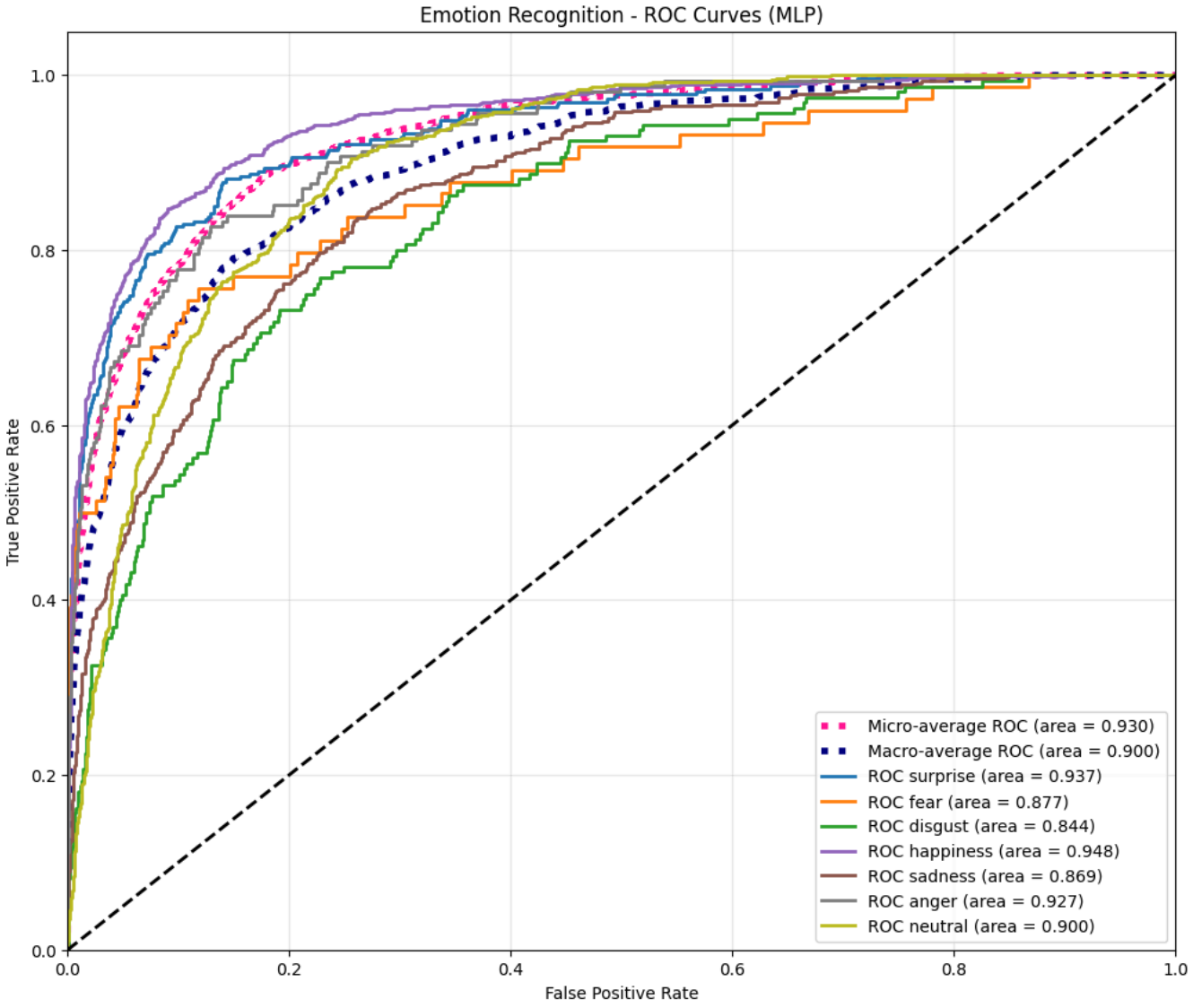
Hyperparameters were selected based on common best practices and insights from the performance of the MLP baseline, which had fairly high validation accuracy (63.62%) but overfitted after only two epochs. Dropout, batch normalization, and L2 regularization were added to stave off overfitting and improve generalization. We limited the search to single- and two-layer architectures to reduce the risk of overfitting associated with deeper models.

Each configuration was trained for up to 30 epochs with early stopping based on validation loss. Weighted cross-entropy loss was used to account for class imbalance, and accuracy on a held-out validation set was used as the primary performance metric for the hyperparameter search. The best model (based on validation accuracy) used a two-layer architecture with 64 and 32 hidden units, a learning rate of 0.0001, dropout rate of 0.5, and L2 regularization of 0.01.

4.1.1 Model Performance

The final MLP model showed consistent improvement over training for 28 epochs with early stopping, taking approximately 26 seconds in total. While the model achieved decent validation and test performance (~68%), the growing gap between training and validation accuracy indicates overfitting despite the applied regularization. The continued overfitting may be due to the subtle class differences and the class imbalance in the emotion dataset even after data augmentation.

Analysis of One-vs-Rest AUC scores revealed varying performance across emotion categories: *happiness* (0.9483) and *surprise* (0.9366) showed strong separability, while *disgust* (0.8440) and *sadness* (0.8694) did not perform as well. The model achieved a macro-average AUC of 0.9004, indicating good overall ranking capabilities even though classification accuracy was not as high.



4.2 Logistic Regression

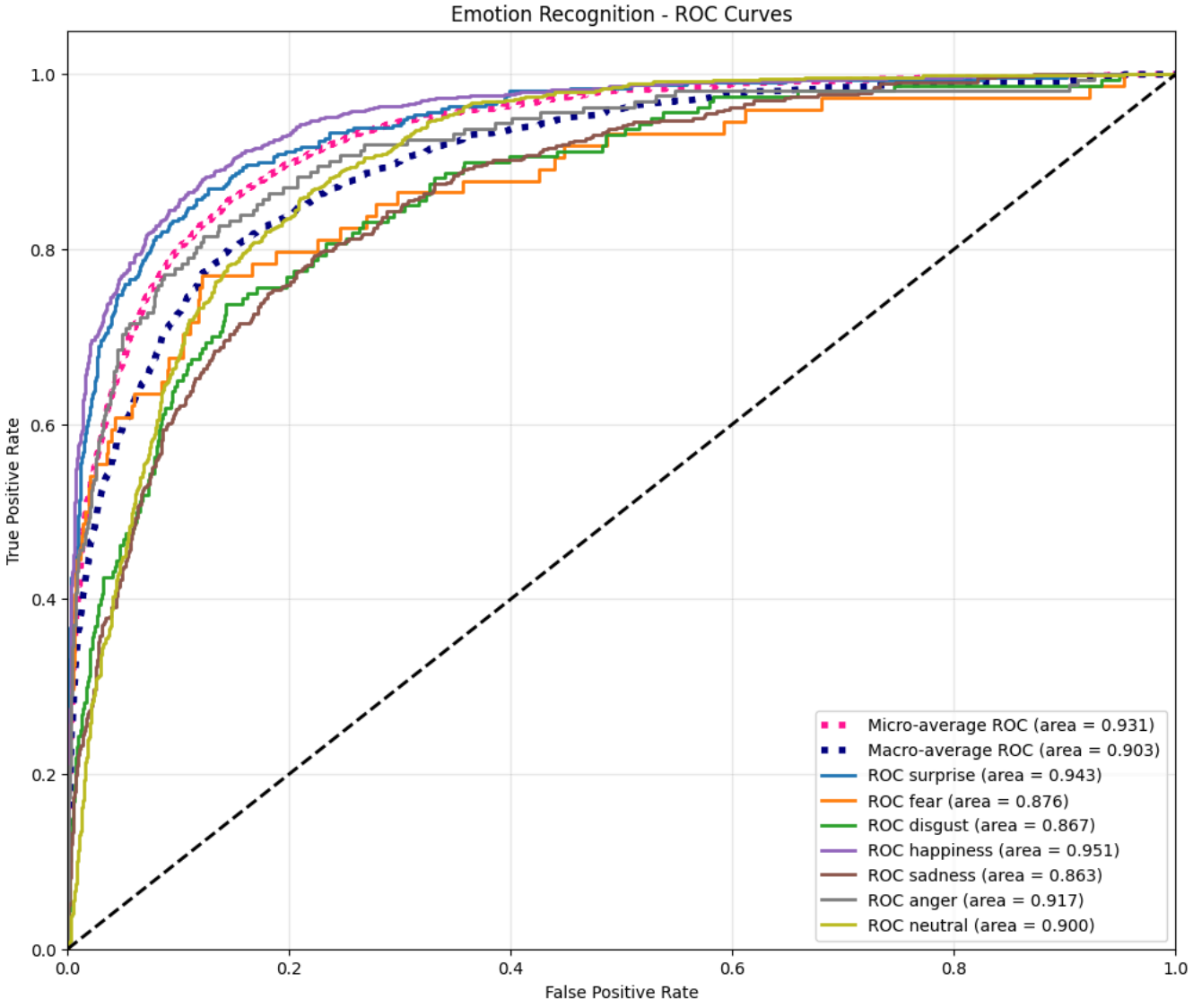
The logistic regression model was implemented using scikit-learn's LogisticRegression class with the "sag" solver, which supports L2 regularization and multi-class classification, and employed class weights to mitigate bias toward overrepresented classes.

For hyperparameter tuning, we tested three different regularization strengths (C values: 0.01, 0.1, 1.0) and selected the configuration that achieved the highest average validation accuracy across the subsets. The best-performing configuration used an L2 penalty with C = 0.01, achieving an average validation accuracy of approximately 66%. This configuration performed better in terms of generalization compared to higher values of C, which led to even more confident models with worse performance on unseen data.

4.2.1 Model Performance

The logistic regression model trained in just 27 seconds, achieving a test accuracy of 68.00% with the search-selected regularization parameter of C = 0.01. However, training accuracy reached 89.94%, indicating a substantial gap between training and test performance. Performance was strongest for frequent emotions such as *happiness*, with an F1-score of 0.85. Macro-average F1 scores were 0.56 on validation and 0.59 on test data, further highlighting imbalances in per-class performance.

AUC scores again revealed strong separability for *happiness* (0.9514) and *surprise* (0.9433), with moderate performance for *anger* (0.9172) and *neutral* (0.8997). The model struggled most with *sadness* (0.8631) and *disgust* (0.8667). Like the MLP model, the logistic regression model achieved a high macro-average AUC score of 0.9028.

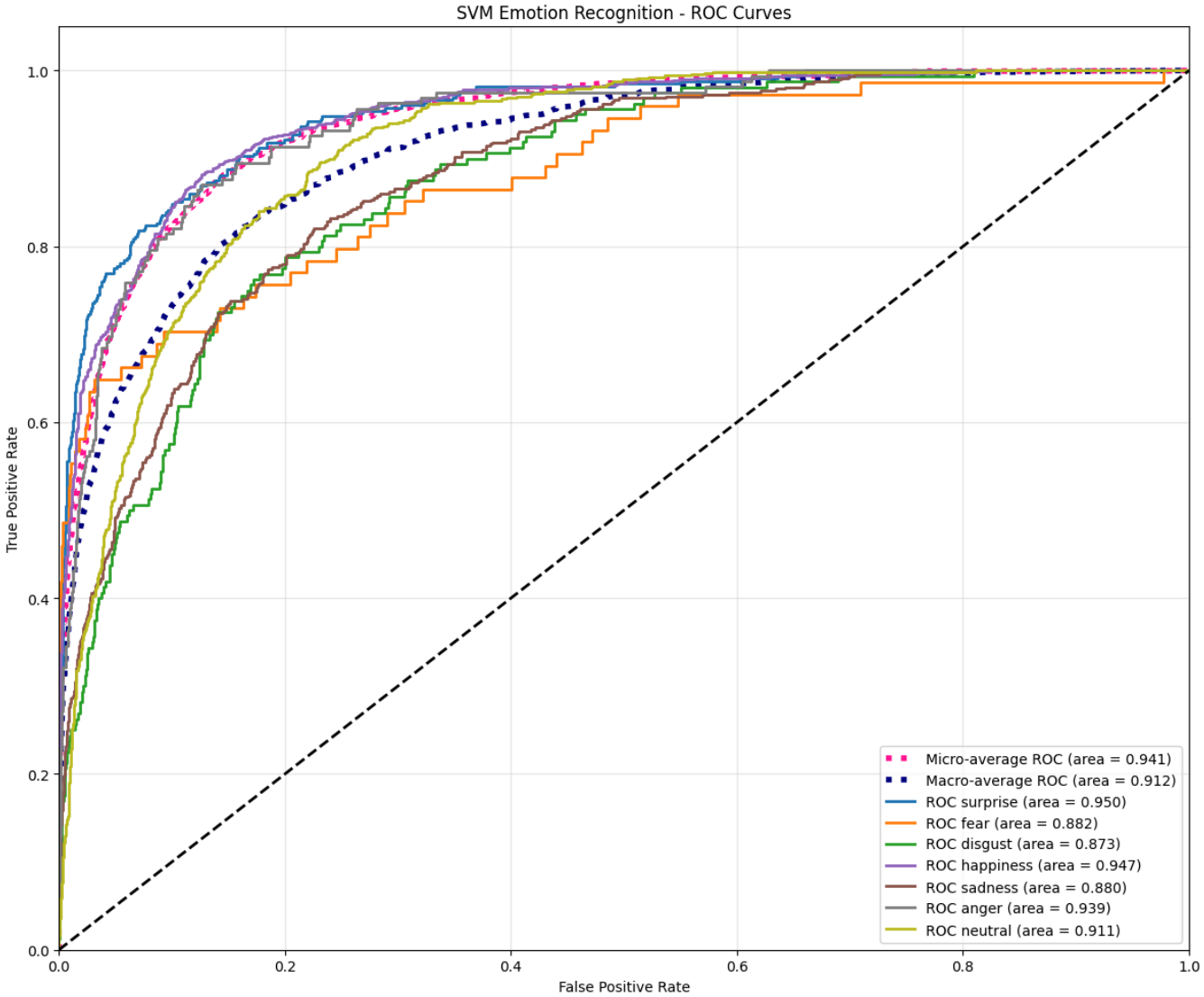


4.3 SVM

The third model implemented was a Support Vector Machine (SVM) classifier using scikit-learn's SVC with a radial basis function (RBF) kernel. As with the logistic regression model, a class-weighted approach was used to account for the class imbalance in the emotion dataset. Hyperparameter tuning was performed through a search, evaluating a grid of C values (regularization strength: 0.01, 0.1, and 1.0) and gamma values ('scale', 0.01, and 0.001). We created three validation subsets and evaluated each hyperparameter combination across all subsets to select the configuration that generalized best. The best configuration—C = 1, gamma = 'scale'—achieved an average validation accuracy of 69.08%, with a training accuracy of 93.81%, indicating a fairly high degree of overfitting.

4.3.1 Model Performance

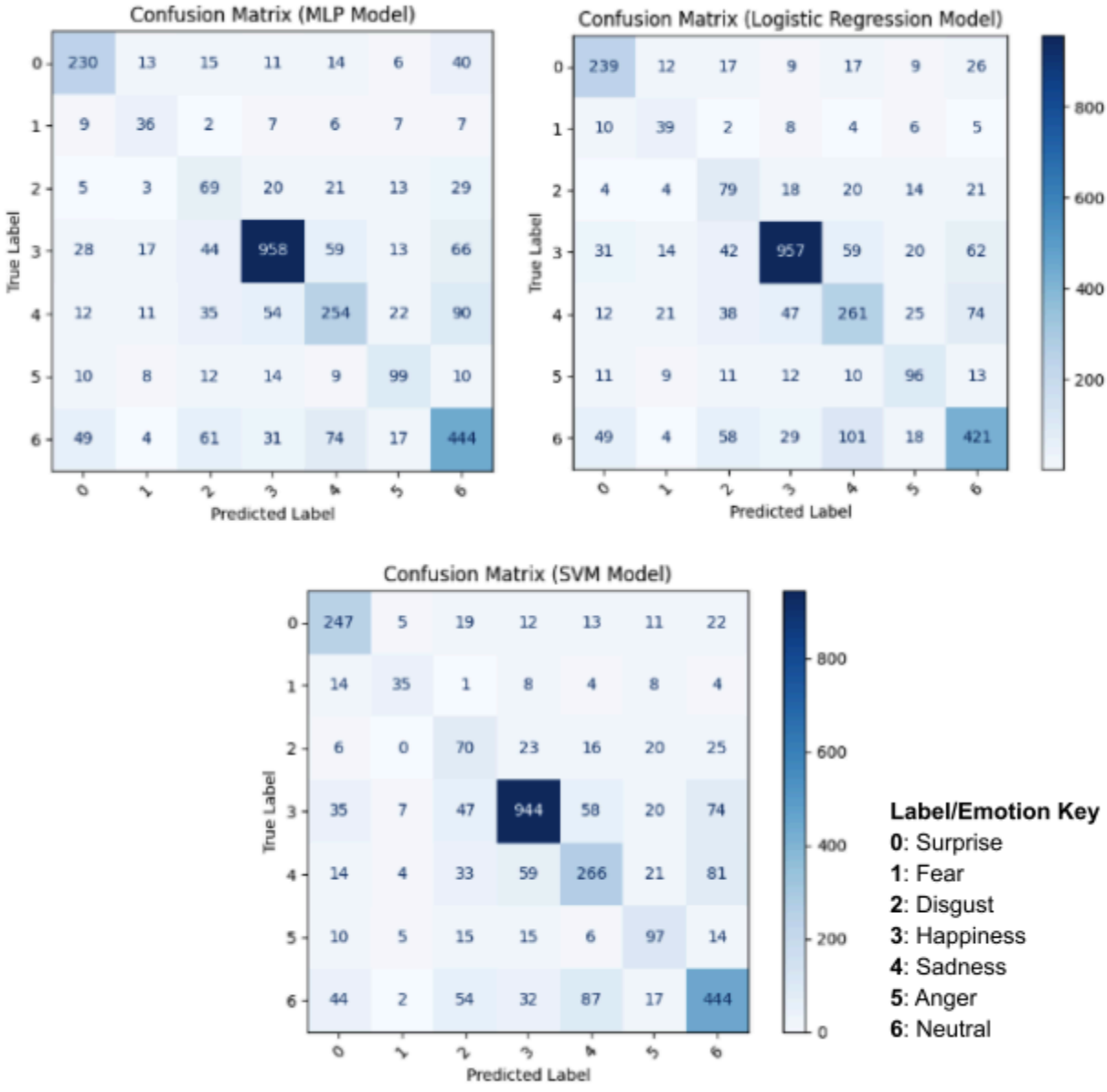
On the test data, the SVM achieved an overall accuracy of 69%, with a macro-average F1-score of 0.60. As with the previous two models, the SVM model performed well on dominant classes such as *happiness* (F1 = 0.84) and *surprise* (F1 = 0.70), while minority classes such as *fear* and *disgust* were more challenging, achieving F1-scores of 0.42 and 0.38, respectively. Additionally, AUC scores again showed strong separability between certain classes: the macro-average AUC was 0.9120, and the micro-average AUC was 0.9414.



The 25-point gap between training accuracy (93.81%) and validation/test accuracy (69%) suggests that the SVM fits the training data well but struggles to generalize as effectively to unseen data. This is further reflected in the per-class performance since minority and majority classes all have very high precision and recall for the training data, but on validation/test data, minority classes perform worse. Overfitting may be due to the small number of examples for *fear* and *disgust*, causing the SVM to memorize patterns in the training data that do not generalize to the validation/test samples.

4.4 Classification Strengths and Weaknesses

To understand the specific strengths and weaknesses of our emotion classification models, we analyzed their confusion matrices, as presented below.



All our models were excellent at identifying *happiness* (the most represented class), correctly classifying it about 80% of the time. However, they struggled with certain emotion pairs. The models frequently mixed up *neutral* expressions with *sadness*, and vice versa, and we observed moderate confusion between *neutral* and *surprise* expressions. *Disgust* proved particularly challenging for all models; *disgust* was often confused with *happiness*, *sadness*, *anger*, and *neutral* emotions. The models likely struggled to learn what *disgust* looks like or define the distinct characteristics of *disgust* since the emotion's features may overlap with the features of other emotions.

While all models shared the above common confusion patterns, each also had its own strengths and weaknesses. The neural network model had the most trouble with mistaking *sadness* as *neutral*. Logistic regression performed best at identifying *fear*, correctly classifying over half of the *fear* examples. The SVM model excelled at recognizing *surprise*, with the highest accuracy rate for the category, but also confused *happiness* as *surprise* more often than the other models.

4.5 Practical Implementation Considerations: Speed, Accuracy, and Generalization

Among our three classification models, the SVM achieved the highest test accuracy at 69%, demonstrating strong performance but also the longest training time. In contrast, the logistic regression model trained in just 27 seconds and delivered a comparable test accuracy of 68%, making it very efficient. The MLP model performed about the same as the logistic regression model, reaching 68% test accuracy after 28 epochs (~26 seconds of training).

All three models showed similar generalization patterns; despite using regularization, notable gaps persisted between training and test accuracy. This indicates that, while the models perform consistently on validation and test sets, they still face challenges in generalizing beyond the training data. To address this, future work could focus on exploring more advanced regularization techniques and increasing both the size and diversity of the dataset, especially for minority classes like *disgust* and *fear*.

For applications prioritizing classification accuracy, SVM is the best-performing option, though it comes with higher training/inference cost and a greater tendency to overfit. In contrast, for tasks requiring fast training or real-time deployment, logistic regression provides nearly the same accuracy with significantly lower resource requirements.

5. Conclusion

The motivation behind this project was to develop a classifier that can be integrated into a smart glasses device to assist in recognizing emotions through facial expressions. While our models demonstrate meaningful discriminability—particularly for well-represented emotions like *happiness* and *surprise*—they are not yet robust enough for real-world deployment. Performance across the MLP, logistic regression, and SVM models revealed strong AUC scores (all around 0.90), but all models suffered from limited classification accuracy, especially for minority classes such as *disgust*, *fear*, and *sadness*.

While each model performed similarly and suffered from their own patterns of classification mismatches, the use case of the classifier requires a model with reduced complexity, so we consider logistic regression as our tentative final model. However, all models showed signs of overfitting and struggled with emotions that were visually or semantically similar. These findings suggest that while the current modeling approach captures meaningful patterns, its effectiveness is limited by factors such as feature representation, dataset balance, and possibly the choice to label each expression with only a single emotion. Further refinement is needed—either through improved feature engineering or a re-evaluation of our labeling strategy. In the below section, we outline future directions for overcoming these limitations based on existing scientific literature on facial expression recognition, as well as future directions for implementation of the model into a smart glasses process.

5.1 Future Directions for Overcoming Model Training Limitations

Although our models demonstrated some ability to distinguish between facial expressions, several limitations in feature representation and outcome labeling may have constrained performance. First, our current features do not capture contextual cues that may be critical for emotion recognition—particularly when expressions share overlapping visual characteristics. Recent research suggests that a transformer-based architecture, which can model both local and global dependencies between facial regions and other features, offer substantial performance improvements in this area (Feng et al., 2023). Additionally, our t-SNE visualizations revealed limited feature separability between emotion classes, suggesting the need for more expressive or discriminative representations. Approaches such as regional-feature fusion have been shown to improve class distinction and reduce common misclassifications on RAF-DB (Huang & Liang, 2023).

Another promising direction involves rethinking the structure of the outcome variable. Facial expressions often represent a combination of emotions rather than a single, discrete category. Adopting an emotion-distributed learning framework, which assigns multiple emotion scores to each image, may better reflect this complexity and improve model performance (Zhou et al., 2015). This would be an easy implementation as the RAF-DB also contains a multi-labelled image sample. Similarly, implementing multi-tier classification systems—such as using PCA to reduce dimensionality followed by fine-grained classification with SVMs—has been shown to significantly enhance accuracy in facial expression recognition (Agrawal et al., 2014; Drume & Jalal, 2012). These approaches may prove especially useful when transitioning to real-time applications such as smart glasses, as model complexity reduction alongside performance enhancement is a key focus for model implementation.

5.2 Future Directions for Model Implementation

Looking ahead, the goal is to integrate the emotion classification model into a smart glasses platform that can deliver non-invasive, real-time feedback about others' facial expressions to the user. To enable on-device or edge computing, future iterations of the model will require significant reductions in complexity and computational demands while preserving classification accuracy. This may involve model compression techniques or the use of lightweight architectures optimized for embedded systems. Importantly, the full system encompassing both the hardware interface and emotional feedback mechanism will need to undergo usability testing with members of the intended user population. These evaluations will help ensure that the system is intuitive and non-disruptive while meaningfully enhancing emotional awareness in real-world social interactions.

6. References

Dataset/Code Links

- Dataset: https://drive.google.com/drive/folders/0B4E10azXECctRUgwVmFPblFUdUE?resourcekey=0-SCrrCMK2lc4IDmhDsDKhRw&usp=sharing or http://whdeng.cn/RAF/model1.html
- Github: https://github.com/Mossd-2/281-Final-Project

Works Cited

1. Agrawal, D., Dubey, S., & Jalal, A. (2014). Emotion recognition from facial expressions based on multi-level classification. *Int. J. Comput. Vis. Robotics*, 4, 365-389. https://doi.org/10.1504/IJCVR.2014.065571.
2. Drume, D., & Jalal, A. (2012). A multi-level classification approach for facial emotion recognition. *2012 IEEE International Conference on Computational Intelligence and Computing Research*, 1-5. https://doi.org/10.1109/ICCIC.2012.6510279.
3. Feng, H., Huang, W., Zhang, D., & Zhang, B. (2023). Fine-Tuning Swin Transformer and Multiple Weights Optimality-Seeking for Facial Expression Recognition. *IEEE Access*, 11, 9995-10003. https://doi.org/10.1109/ACCESS.2023.3237817.
4. Huang, X., & Liang, J. (2023). Facial expression optimal class separability for expression recognition. https://doi.org/10.1117/12.2684245.
5. Li, S., & Deng, W. (2019). Reliable Crowdsourcing and Deep Locality-Preserving Learning for Unconstrained Facial Expression Recognition. *IEEE Transactions on Image Processing*, *28*(1), 356–370. https://doi.org/10.1109/TIP.2018.2868382
6. Zhou, Y., Xue, H., & Geng, X. (2015). Emotion Distribution Recognition from Facial Expressions. *Proceedings of the 23rd ACM international conference on Multimedia*. https://doi.org/10.1145/2733373.2806328.