

Interpretable Emotion Recognition: Visualizing Neural Attention in Facial Emotion Classification with Grad-CAM

INTRODUCTION

Facial Emotion Recognition (FER) is a fundamental task in computer vision, with practical applications in areas such as mental health and human-computer interaction. While many convolutional neural networks (CNNs) have achieved strong performance on the FER2013 dataset (which includes 7 basic emotions), they often lack interpretability.

In this project, our goal is to combine accurate emotion classification with model interpretability. We will use Grad-CAM to visualize the key facial regions that influence the model's predictions. Our aim is not only to build an effective classifier and achieve competitive performance, but also to understand how the model processes facial features and whether it focuses on meaningful areas (such as the mouth or eyes). This dual approach of prediction and interpretation is essential for developing reliable and transparent AI systems for emotion recognition.

STATE OF THE ART

Facial emotion recognition (FER) has seen significant advancements with the adoption of deep learning techniques, particularly Convolutional Neural Networks (CNNs). The FER2013 dataset, introduced by Goodfellow et al. in 2013, has become a standard benchmark for evaluating FER models due to its challenging nature, including variations in facial expressions, lighting, and occlusions. Several studies have explored various CNN architectures to improve FER performance on the FER2013 dataset:

Pramerdorfer and Kampel (2016) reviewed the state of the art in image-based FER using CNNs. They highlighted that utilizing ensembles of modern deep CNNs can lead to substantial performance increases, achieving a test accuracy of 75.2% on FER2013 without requiring auxiliary training data or face registration. Khairuddin and Chen (2021) adopted the VGGNet architecture –similar to ours–, fine-tuned its hyperparameters, and experimented with various optimization methods. Their model achieved a single-network accuracy of 73.28% on FER2013 without using extra training data. Lamichhane and Karn (2024) introduced a hybrid CNN-BiLSTM model for FER, which achieved an accuracy of 79.4% when classifying all seven emotions on the FER2013 dataset.

These studies demonstrate the continuous efforts to enhance FER performance using deep learning techniques, with varying degrees of success on the FER2013 dataset. Facial Emotion Recognition (FER) has seen significant progress with the use of Convolutional Neural Networks (CNNs), with the FER2013 dataset serving as a standard benchmark.

METHODOLOGY

The project utilizes the FER2013 dataset, a standard benchmark comprising 35,887 grayscale facial images (48x48 pixels) categorized into 7 basic emotions. The dataset was divided, with 28,709 samples initially used for training (further split into 80% for training and 20% for validation), and a separate test set of 7,178 samples. Training data underwent both normalization and data augmentation, while validation and test sets received only normalization for consistent evaluation.

Our model is based on VGG19, a well-known CNN pre-trained on ImageNet, a large-scale dataset with over 14 million labeled images. This choice was driven by two key reasons:

First, transfer learning from ImageNet enables the model to leverage rich, generalizable visual features learned from a diverse set of images. This accelerates training and boosts performance, especially on smaller datasets like FER2013.

Second, VGG19's straightforward architecture, 16 convolutional layers and 3 fully connected ones using consistent 3×3 filters and 2×2 max-pooling, makes it ideal for interpretability. Its structured design facilitates techniques like Grad-CAM, helping us visualize how the model attends to facial features during emotion recognition.

We incorporated the pre-trained VGG19 layers into a custom PyTorch model, CustomVGG19, adapting the classifier head to output predictions for 7 emotion classes instead of the original 1000.

The architecture of the proposed CustomVGG19 model is composed of three core components: a feature extractor, an adaptive pooling layer, and a classifier head. The feature extractor leverages pre-trained VGG19 weights from ImageNet. Initially, all convolutional layers were frozen to preserve the general-purpose features learned from large-scale data; however, to better adapt to the FER2013 emotion recognition task, we implemented selective fine-tuning by unfreezing the last N convolutional blocks, controlled via a `fine_tune_layers` parameter in the model's constructor. The extracted feature maps are passed through an `nn.AdaptiveAvgPool2d((1, 1))` layer, which reduces spatial dimensions to produce a 512-dimensional vector. This vector is then fed into the classifier head, a sequential block comprising `nn.Linear(512, 256)`, `nn.BatchNorm1d(256)` for normalization, a ReLU activation, Dropout for regularization, and a final `nn.Linear(256, 7)` to output predictions over the seven emotion classes.

Preprocessing steps included: converting grayscale images to 3-channel RGB (to match VGG19's ImageNet input), resizing to 48x48 pixels, and normalization using ImageNet's mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225]. To enhance generalization, various data augmentation techniques (e.g., random rotation, horizontal flipping, random resized cropping) were applied exclusively to the training set.

The training process spanned up to 50 epochs, involving systematic optimization strategies:

-Optimizer: Evolution from Adam (initially with `weight_decay`) to the final choice of SGD with `momentum=0.9`, demonstrating systematic tuning for stability and performance.

-Loss function: Transitioned from `nn.CrossEntropyLoss()` to `LabelSmoothingCrossEntropy` (configured with `class_weights` to address class imbalance). Label Smoothing was adopted as a regularization strategy to prevent overfitting by smoothing target labels and reducing overconfidence.

-Early stopping: Consistently applied across all configurations to monitor validation loss and halt training when no significant improvements were observed, preventing overfitting.

The model was trained using a batch size of 32 and input images resized to 48x48 pixels. The classification task involved 7 emotion classes and was trained over 50 epochs. Key hyperparameters such as the learning rate (set within the optimizer), dropout rate (defined in the classifier head), and the number of fine-tuned layers (configured in the CustomVGG19 architecture) were treated as configurable to allow flexibility and optimization during experimentation.

To monitor model performance and guide architectural decisions, training and validation loss and accuracy curves were used throughout all experimental configurations. These curves allowed us to observe trends such as convergence behavior, underfitting, and overfitting, which informed decisions on regularization, fine-tuning depth, and learning rate adjustments.

In the final configuration, a more comprehensive evaluation was conducted using additional classification metrics. Cross-Entropy Loss remained the primary optimization objective, assessing the divergence between predicted probabilities and true labels. While overall accuracy was still reported, the imbalanced nature of the FER2013 dataset necessitated a deeper analysis. A detailed classification report was generated, including precision (measuring prediction exactness), recall (sensitivity to actual instances), and F1-score (the harmonic mean of precision and recall), all computed per class to better understand performance across the seven emotion categories. Furthermore, a confusion matrix was used to visualize misclassifications and assess whether the model disproportionately favored certain classes.

EXPERIMENTS

We designed four incremental experimental configurations to iteratively evaluate and enhance the performance of a facial emotion recognition model on the FER2013 dataset. Each experiment builds logically upon the previous one, testing specific hypotheses and incorporating architectural and training improvements based on observed outcomes.

Configuration 1: Baseline (VGG19 Feature Extractor with Simple Classifier)

We established a baseline using a pre-trained VGG19 as a fixed feature extractor. Only the final custom classifier (a linear layer with ReLU and Dropout) was trained, while all convolutional layers remained frozen. Images were resized to 48×48, normalized using ImageNet statistics, and augmented with random rotation, horizontal flip, and resized crops. The model was optimized using Adam with CrossEntropyLoss. Accuracy plateaued at ~52% (train), ~45% (val), and 44.22% (test), suggesting limited learning capacity due to the frozen backbone.

Configuration 2: Fine-tuning Deeper Layers and Class Weights

To improve performance, we unfroze the last 9 convolutional layers, allowing more task-specific adaptation. To address class imbalance, we applied balanced class weights in the CrossEntropyLoss. These weights were computed inversely proportional to the frequency of each class in the training dataset, effectively giving more importance to under-represented emotions during model optimization. The optimizer remained Adam (lr=1e-4). This yielded substantial improvement in training accuracy (>90%) but revealed overfitting, with validation peaking around 57% and test accuracy at 51.31%.

Configuration 3: Reduced Fine-tuning, SGD Optimization, and Learning Rate Scheduling

This experiment focused on generalization. We limited fine-tuning to the last 5 feature modules of VGG19 to reduce overfitting and switched to SGD with momentum (lr=0.01, momentum=0.9, weight_decay=1e-4). A learning rate scheduler (ReduceLROnPlateau) was added, monitoring validation loss. This scheduler dynamically reduced the learning rate by a factor of 0.1 if the validation loss did not improve for 10 consecutive epochs, helping the model converge more effectively. This configuration improved validation accuracy to ~74%, while training accuracy deliberately decreased from over 90% to approximately 77%. This narrowing of the gap, alongside a final test accuracy of 57.91%, indicated much better generalization and training stability.

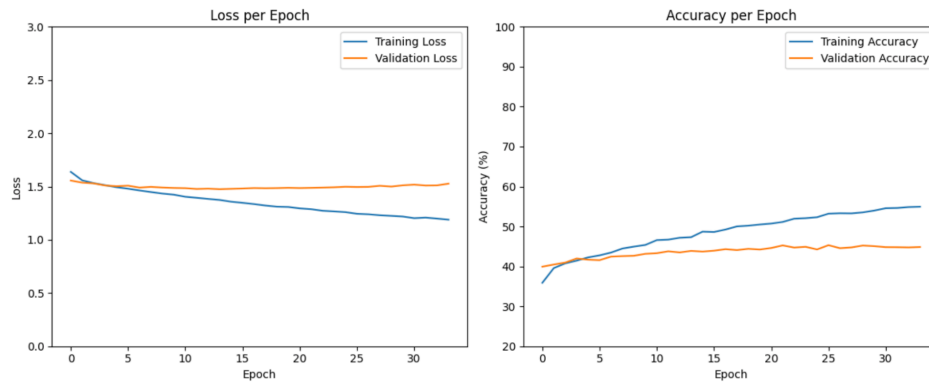
Configuration 4: Aggressive Augmentation, Label Smoothing, and Enhanced Regularization

Building on the previous setup, we expanded the augmentation pipeline: stronger rotations (15°), broader cropping scale, and added transforms like ColorJitter, ElasticTransform, RandomPerspective, and RandomAffine. To regularize further, L2 weight decay was increased to 1e-3, and CrossEntropyLoss was replaced by LabelSmoothingCrossEntropy to reduce overconfidence. This technique, by distributing a small portion of the target probability to all non-true classes, prevents the model from becoming overly confident in its predictions and enhances its generalization capabilities. Learning rate scheduling remained. This configuration significantly boosted robustness, reaching ~96% training accuracy, ~83% on validation, and 59.11% on test.

RESULTS

Configuration 1: Baseline (VGG19 Feature Extractor with Simple Classifier)

This baseline configuration showed limited learning capacity and severe underfitting, indicating the frozen backbone was insufficient for the task.

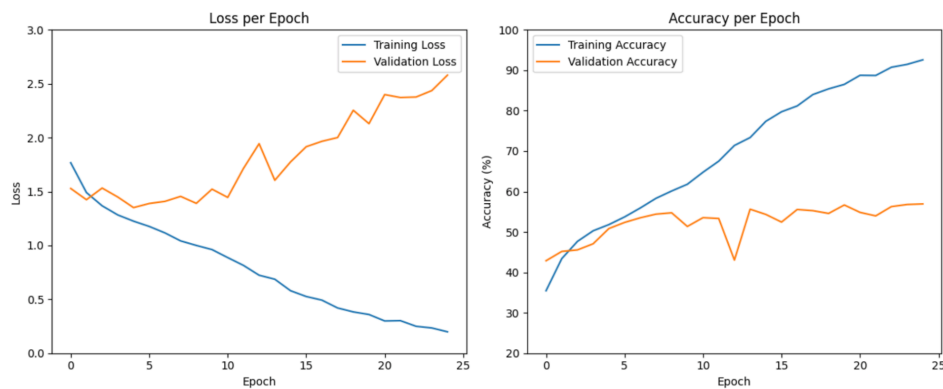


Loss and accuracy curves on training and validation set.

Accuracy of the model on the test set: 44.22%.

Configuration 2: Fine-tuning Deeper Layers and Class Weights

Despite improved training accuracy, this configuration exhibited significant overfitting, with a substantial performance gap between training and validation sets.

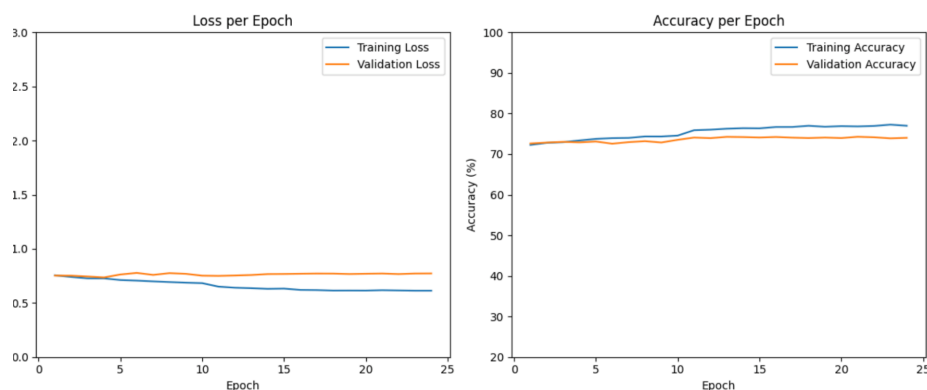


Loss and accuracy curves on training and validation set.

Accuracy of the model on the test set: 51.31%

Configuration 3: Reduced Fine-tuning, SGD Optimization, and Learning Rate Scheduling

This configuration achieved notably improved generalization and training stability, significantly narrowing the gap between training and validation performance.

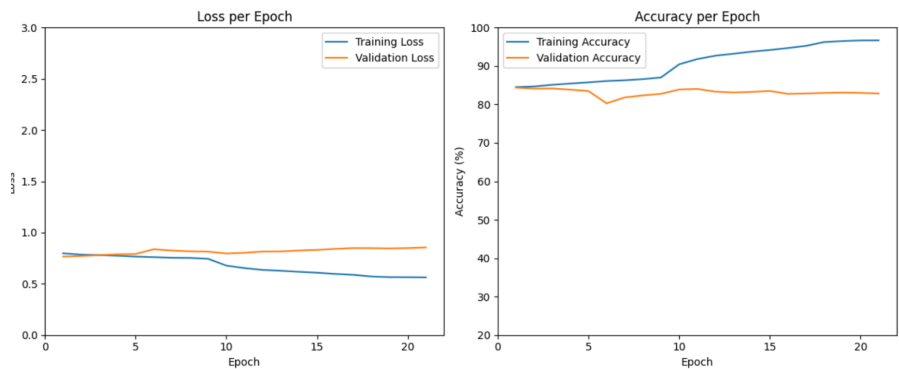


Loss and accuracy curves on training and validation set.

Accuracy of the model on the test set: 57.91%

Configuration 4: Aggressive Augmentation, Label Smoothing, and Enhanced Regularization

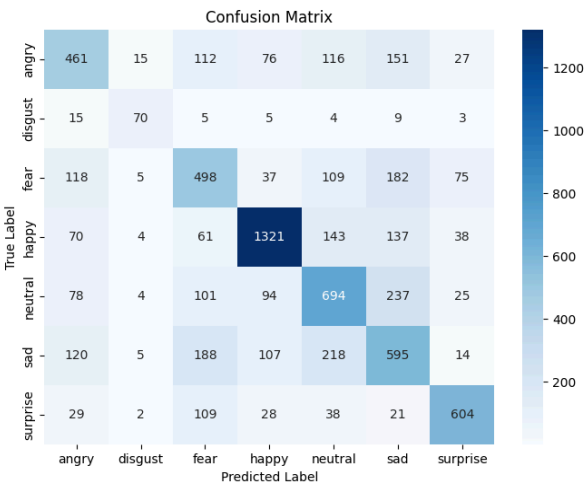
The final configuration demonstrated robust generalization and achieved the highest test accuracy, indicating enhanced model robustness through comprehensive regularization and augmentation.



Loss and accuracy curves on training and validation set. Classification report, confusion matrix on test set are in the results folder

Accuracy of the model on the test set: 59.11%

Label	Precision	Recall	F1-score	Support
angry	0.52	0.48	0.50	958
disgust	0.67	0.63	0.65	111
fear	0.46	0.49	0.47	1024
happy	0.79	0.74	0.77	1774
neutral	0.52	0.56	0.54	1233
sad	0.45	0.48	0.46	1247
surprise	0.77	0.73	0.75	831
accuracy			0.59	7178
macro avg	0.60	0.59	0.59	7178
weighted avg	0.60	0.59	0.59	7178



DISCUSSION

This section analyzes the outcomes of each experimental configuration, highlighting observed patterns, successes, and persistent limitations.

Configuration 1: Baseline Discussion

-Analysis of Outcomes: This initial configuration successfully established a preliminary performance baseline for the facial emotion recognition task. The model, leveraging a pre-trained VGG19 as a fixed feature extractor, achieved a test accuracy of approximately 44.93%, notably exceeding random chance for a 7-class classification problem. This outcome validated the fundamental premise of transfer learning, demonstrating that the VGG19's general image features could capture some discriminative information relevant to facial expressions

-Identified Limitations & Challenges: Despite this initial validation, the configuration exhibited significant limitations that restricted its overall effectiveness. A primary challenge was the model's severely limited learning capacity. Training accuracy showed a modest improvement from 47% to 52% over 25 epochs, which is notably low for a training set, indicating the model was fundamentally underfitting the complex patterns of facial expressions, even on data it had seen. Meanwhile, validation accuracy initially increased but then gradually declined to approximately 49%, suggesting that the model failed to learn transferable features and was beginning to overfit superficial patterns. The decision to keep the entire VGG19 convolutional base frozen proved overly restrictive, preventing the model from adapting its powerful feature extraction capabilities to the specifics of the FER2013

dataset. Additionally, the absence of explicit class weighting in the loss function led to a pronounced performance disparity across emotion categories, most notably, a recall of just 0.05 for the "disgust" class, highlighting the model's inability to learn from underrepresented classes.

-Key Takeaways for Future Iterations: The results from this baseline experiment provided crucial insights for subsequent iterations. They highlighted that the features extracted by the frozen VGG19 base were insufficient for emotion recognition, leading to both underfitting on the training set and poor generalization. This underlined the critical need for deeper fine-tuning of the convolutional layers to enable the model to learn more domain-specific features, thereby increasing its learning capacity. Also, the struggle with minority classes and the imbalance of the classes, emphasized the necessity of implementing robust regularization strategies and class weighting to ensure that any increased model capacity translates into improved generalization rather than simply memorization of the training set.

Configuration 2: Fine-tuning Deeper Layers, Class Weights, and Early Stopping

-Analysis of Outcomes: This configuration achieved a notable improvement in model performance compared to the baseline. By unfreezing the last 9 layers of the VGG19 convolutional base, the model gained significant capacity to adapt its feature extraction capabilities to the specific characteristics of facial expressions in the FER2013 dataset. The integration of class_weights in the CrossEntropyLoss effectively addressed the class imbalance issue, leading to more balanced learning across all emotion categories (though specific recall metrics for minority classes are not detailed here, the overall accuracy improvement suggests this). The validation accuracy peaked at approximately 57%, demonstrating a substantial gain over the baseline's 49%, and the final test accuracy reached 53.63%. The implementation of early stopping successfully prevented further degradation of validation performance, ensuring that the model saved its best state.

-Identified Limitations & Challenges: Despite these improvements, the primary and most significant limitation of this configuration remained severe overfitting. This was profoundly evident in the considerable gap between the training accuracy, which rapidly soared to over 90%, and the validation accuracy, which peaked at 57% before early stopping was triggered. This large discrepancy indicates that while the model had increased capacity to learn from the training data, it extensively memorized these patterns rather than learning robust, generalizable features. The Adam optimizer, used with a learning rate of $1e-4$, along with the existing basic data augmentation and dropout in the classifier, proved insufficient to bridge this generalization gap effectively.

-Key Takeaways for Future Iterations: The results from this configuration clearly validate the strategies of deeper fine-tuning and the use of class weighting, as both contributed to a significant boost in performance over the baseline. However, the pronounced overfitting emphatically highlights the urgent need for stronger and more diverse regularization techniques. Future experiments should prioritize the implementation of methods designed to explicitly reduce memorization, such as weight_decay and label smoothing, to force the model to learn more robust and transferable representations. Furthermore, exploring alternative optimizers like SGD with Momentum, which are sometimes more conducive to finding flatter minima that generalize better, would be a critical next step.

Configuration 3: Reduced Fine-tuning, SGD Optimization, and Learning Rate Scheduling

-Analysis of Outcomes: This configuration achieved a transformative improvement in model performance, fundamentally addressing the severe overfitting observed in Configuration 2. The most significant outcome was the dramatic reduction in the generalization gap: training accuracy peaked around 77.26%, while validation accuracy reached approximately 74.27%, resulting in a negligible discrepancy of only about 3%. This starkly contrasts the substantial 33% gap previously seen, unequivocally demonstrating that the model is no longer merely memorizing the training data but has learned robust and transferable features. Consequently, both validation and test accuracies saw

substantial gains; validation accuracy improved from 57% to 74.27%, and the final test accuracy reached an impressive 57.91% (up from 53.63%). These results validate the effectiveness of the introduced optimization strategy, where the shift to SGD with momentum and a well-tuned initial learning rate, coupled with weight_decay and the ReduceLROnPlateau scheduler, proved highly successful in achieving a more balanced and generalizable learning process. Furthermore, the reduction in fine_tune_layers to 5 likely played a crucial role in constraining the model's capacity, preventing excessive memorization while still allowing for effective feature adaptation.

-Identified Limitations & Challenges: Despite the substantial progress in generalization, the model currently appears to have reached a performance ceiling, as indicated by the plateauing of both training and validation accuracies after a certain number of epochs. While the generalization gap is now minimal, the absolute accuracy figures (particularly 57.91% on the test set) suggest that there is still room for improvement for this complex task. The model might now be slightly underfitting, meaning it has not yet fully extracted all the available patterns from the data, or it has converged to a local optimum given its current capacity and applied regularization. A contributing factor to this plateau could be the absence of more sophisticated regularization techniques, which were intentionally omitted in this configuration to isolate the effects of the optimizer and fine-tuning strategy.

-Key Takeaways for Future Iterations: The success of this configuration emphatically validates the combined strategy of employing the SGD optimizer with momentum and weight decay, integrating a learning rate scheduler, and carefully managing the number of fine-tuned layers for effective generalization. The paramount lesson learned is that robust optimization and capacity management are foundational for bridging the overfitting gap. Moving forward, the primary focus must shift from mitigating severe overfitting to boosting the absolute performance of the model. This necessitates the strategic introduction of additional regularization and robustness-enhancing techniques. Specifically, the next experiments should prioritize incorporating Label Smoothing to refine prediction calibration and applying more aggressive data augmentation techniques to further diversify the training data and enhance the model's ability to learn invariant features, thereby pushing the performance ceiling higher. Should these measures still result in a plateau, a cautious and slight increase in fine_tune_layers could be considered to provide the model with more capacity.

Configuration 4: Aggressive Augmentation, Label Smoothing, and Enhanced Regularization


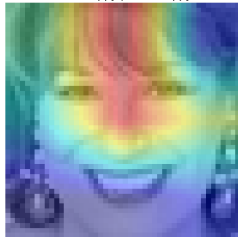
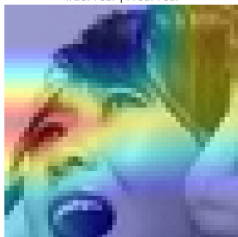
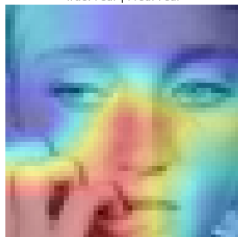

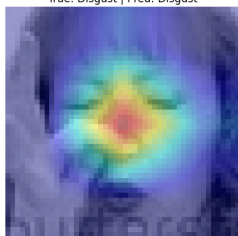
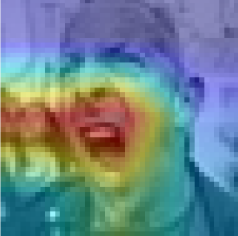
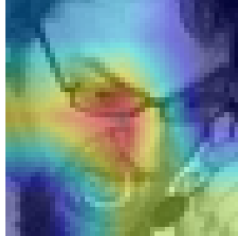
-Analysis of Outcomes: This configuration represents a significant improvement in model performance, successfully surpassing the limitations observed in Configuration 3. The strategic combination of highly aggressive data augmentation, Label Smoothing Cross-Entropy, and increased weight decay for stronger L2 regularization demonstrably enhanced the model's ability to learn robust and generalized features. Validation accuracy rose to approximately 83% (up from 74.27%), and test accuracy improved to 59.11% (from 57.91%), indicating a clear advancement in generalization despite maintaining a high training accuracy of around 96%. Aggressive augmentation forced the model to become more resilient to input variability, while label smoothing prevented overconfidence and encouraged better-calibrated predictions. The higher weight decay further supported generalization by penalizing large weights more heavily.

-Identified Limitations & Challenges: Despite exploring various hyperparameter combinations and aggressive regularization, test accuracy has consistently not exceeded 60%. This suggests that the model may have reached a performance ceiling inherently tied to the characteristics of the FER2013 dataset itself, such as its variability, noise, or potential ambiguities in labeling. The consistently high training and validation accuracies indicate the model is well-trained and capable of capturing complex patterns, but the nature of the test data likely poses an absolute constraint on achieving higher performance with current approaches.

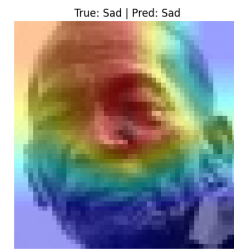
Grad-CAM

In this project, we aimed to combine accurate facial emotion classification with visual interpretability

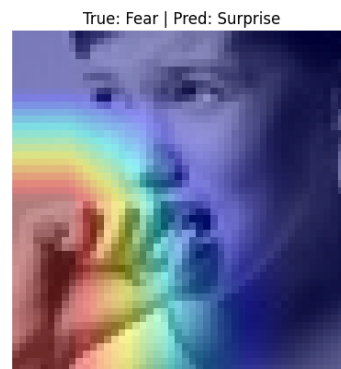
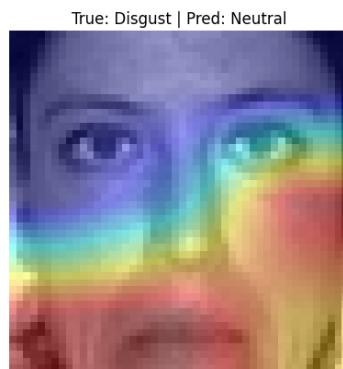
using Grad-CAM. The Grad-CAM heatmaps allow us to analyze which facial regions the model attends to when predicting emotions. We observe that:

<p>Happy faces often show attention around the mouth and cheeks, which is consistent with human perception of smiles.</p>	<p>True: Happy Pred: Happy</p> 	<p>True: Happy Pred: Happy</p> 
<p>For Fear, activations typically concentrate around the eyes and eyebrows, capturing widened eyes or raised brows, key features in fearful expressions.</p>	<p>True: Fear Pred: Fear</p> 	<p>True: Fear Pred: Fear</p> 
<p>In the case of Disgust, the model frequently focuses on the nose and upper lip area, which aligns with crinkling and muscle movements common in disgusted faces.</p>	<p>True: Disgust Pred: Disgust</p> 	<p>True: Disgust Pred: Disgust</p> 
<p>For Anger, attention often centers on the eyebrows and forehead, detecting furrowed brows or tension.</p>	<p>True: Angry Pred: Angry</p> 	<p>True: Angry Pred: Angry</p> 

For Sadness, the model focuses on the eyes and eyebrows. In some cases, it also highlights the mouth, especially when a frown is present.



However, we also observed some misclassifications. In particular, Disgust was often predicted as Neutral, and Fear was confused with Surprise. These errors may result from subtle facial similarities between emotions or from general noise in the dataset. In these cases, Grad-CAM still showed plausible attention areas, suggesting that the model was making informed, though imperfect, decisions.



During our analysis, we also found labeling errors in the dataset, specifically between Surprise and Neutral. Some images labeled as Neutral clearly showed features of Surprise, such as raised eyebrows or open eyes. These are not model errors, but mistakes in the ground truth labels. This label noise introduces bias during training, and may lead the model to learn inconsistent patterns, even when its predictions are visually reasonable.

Despite these issues, Grad-CAM confirms that the model attends to relevant facial regions, not background noise. This supports the interpretability and reliability of the model. Visualizing both correct and incorrect predictions helped us understand how the model behaves and where it struggles.

Overall, our findings highlight the importance of using Grad-CAM. It helps validate the model's reasoning and offers insights beyond accuracy. In future work, improving label quality and using noise-robust training methods could help reduce bias and improve both performance and interpretability.

Conclusions

This work has explored the development and progressive refinement of a deep learning model for facial emotion recognition using the FER2013 dataset. The starting point was a pre-trained VGG19 model used as a feature extractor with a simple classifier, which served as a baseline. From there, successive improvements were introduced through data augmentation, fine-tuning, class rebalancing, and deeper architectural modifications.

The results demonstrate that a thoughtful combination of transfer learning and data-specific adaptations can significantly improve model performance. Specifically:

- Fine-tuning the VGG19 backbone and incorporating Batch Normalization layers yielded noticeable gains in accuracy and generalization.
- Applying class weighting and early stopping effectively addressed the challenges of class imbalance and overfitting.
- Grad-CAM visualizations provided valuable interpretability, highlighting which regions of the face contributed most to each prediction.
- Despite improvements, the model still showed reduced performance in distinguishing between similar emotions (e.g., fear vs. surprise, sad vs. neutral), which is consistent with the known limitations of the FER2013 dataset.

In summary, the experiments show that transfer learning, when carefully applied, offers a powerful approach to emotion recognition, especially when the amount of data is limited. Future work may explore the use of attention mechanisms, more advanced data augmentation techniques (e.g., mixup, adversarial augmentation), or training from scratch on a more balanced and diverse dataset to further improve robustness and accuracy.

References

Pramerdorfer, C., & Kampel, M. (2016). Facial Expression Recognition using Convolutional Neural Networks: State of the Art. *arXiv preprint arXiv:1612.02903*. <https://arxiv.org/abs/1612.02903>

Khairuddin, M. A., & Chen, J. (2021). Performance Enhancement of Facial Expression Recognition System Using Deep Learning with Data Augmentation. *arXiv preprint arXiv:2105.03588*. <https://arxiv.org/abs/2105.03588>

Lamichhane, S., & Karn, R. K. (2024). Facial Emotion Recognition Using Hybrid CNN-BiLSTM Architecture. *International Journal of Engineering and Technology*. <https://www.nepjol.info/index.php/injet/article/view/72579>