Facial Expression Recognition Project

Alex Rowson, Alex Slotter

Introduction

The primary objective of this project was to achieve the highest possible accuracy in facial emotion recognition using deep learning techniques which we learned during our MSDS Deep Learning course. Facial expression classification remains a complex and critical task within the field of deep learning, with applications in psychology, healthcare, education, and human-computer interaction. While numerous approaches exist, achieving state-of-the-art performance on real-world datasets like FER2013 requires careful architectural choices and systematic evaluation.

Our approach involved progressively designing and refining a series of convolutional neural network models, each aimed at improving accuracy through increasingly advanced techniques. Starting from a simple baseline CNN, we introduced handcrafted preprocessing filters, residual connections, and ultimately an attention-enhanced architecture using Convolutional Block Attention Modules. Each model was rigorously evaluated through 3-fold cross-validation to ensure fair comparison and robust performance measurement. This paper outlines the architecture, performance, and challenges of each model, culminating in the identification of the most effective design which we could produce for facial emotion classification.

Benchmarks from Kaggle

When looking to address the classification problem associated with the FER dataset, we wanted to have a performance benchmark in order to see how our own methodologies compare to other approaches. Kaggle user ananthu017 made a significant contribution to this domain by releasing a curated FER2013 dataset and a clean CNN-based solution architecture. While we don’t intend to use anathu017s code as a starting point, there were some key ideas adapted from their implementation, including:

- Data normalization and augmentation pipeline: We adopted their use of rotation, flipping, and cropping to improve data diversity.

- CNN initialization strategy: Our baseline and custom kernel models were structurally inspired by their early convolution-pooling-dense pipeline.

- Training procedure: Their emphasis on batch normalization and dropout as anti-overfitting measures was integrated throughout our deeper models.

While their architecture was limited to traditional CNNs, our work extended it by incorporating residual connections, custom fixed filters, and attention-based modules to achieve higher accuracy and more robust generalization.

Methodology

Our overall methodology for this project was focused on finding the best model to address the FER challenge, and we aimed to test that using the accuracy of the model. We aimed to incorporate many of the key areas of this year's Data Science Deep learning class, so we created 4 separate models to compare the performance and evaluate better which model - or types of model - do best with facial expressions.

Dataset and Preprocessing

We utilized the FER2013 dataset, a widely used benchmark for facial emotion recognition introduced in an ICML Kaggle competition. It contains 35,887 grayscale images (48×48 pixels), each labeled with one of seven emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. The dataset is divided into training, validation, and test sets.

To improve generalization and mitigate overfitting, we applied data augmentation, including random horizontal flipping, rotation of ±10°, and random resized cropping. All images were normalized to a mean of 0.5 and standard deviation of 0.5, ensuring stable and efficient model convergence. These transformations help the models learn more robust features despite the low resolution and class imbalance of the dataset.

Loss Function and Optimizer

For each model tested, we used the same loss function and optimizer. Cross-entropy loss and the Adam optimizer were chosen for efficiency and reliability, as well as cross-entropy loss being a good choice for classification tasks like ours.

Model Architectures

1. Baseline CNN (EmotionCNN)

A standard three-layer convolutional neural network that establishes a foundation for comparison.

- Conv Layer 1 (32 filters): Captures basic edges and contours critical for initial facial structure detection.

- Max Pool + Dropout: Reduces spatial resolution and prevents overfitting on local pixel patterns.

- Conv Layer 2 (64 filters): Extracts mid-level features such as eyes, nose, and mouth shape, improving class separability.
- Max Pool + Dropout: Preserves essential activations while increasing invariance to translation and deformation.
- Conv Layer 3 (128 filters): Learns higher-level abstract features, supporting recognition of nuanced expressions.
- Flatten + FC Layers: Transforms the spatial feature map into a class-predictive vector, enabling emotion classification.

2. CNN with Fixed Custom Kernels

Introduces fixed filters at the input to enhance critical facial features before training.

- Custom Kernel Layer (5 filters): Applies edge and shape filters like Sobel and Laplacian to accentuate emotion-relevant facial boundaries (e.g., mouth corners, brow lines).
- Conv Layer 1 (32 filters): Builds on enhanced inputs to identify region-specific features like eyebrow lifts or frowns.
- Conv Layer 2 (64 filters): Learns spatial interactions between key regions, aiding in complex emotion mapping.
- Conv Layer 3 (128 filters): Integrates multi-regional patterns for improved class distinction in overlapping expressions.
- Flatten + FC Layers: Condenses extracted representations into emotion class probabilities.

3. SimpleResNetFER

Incorporates residual connections and global pooling for deeper, stable learning.

- Custom Kernel Layer: Pre-encodes facial structure by highlighting edges, reducing the burden on initial convolution layers.

- Initial Conv + BatchNorm (32 filters): Normalizes inputs and extracts refined low-level features under stabilized conditions.
- Residual Block 1 (64 filters): Learns shallow-to-intermediate features while preserving gradient flow via skip connections.
- Residual Block 2 (128 filters): Refines interactions among facial components (e.g., combining eye + mouth cues) for better emotion context.
- Residual Block 3 (256 filters): Encodes complex global patterns (e.g., asymmetry, intensity) critical for differentiating subtle emotions.
- Global Average Pooling: Compresses spatial data to retain only the most dominant features, improving robustness to alignment issues.
- FC Layer: Outputs the final emotion prediction, leveraging features that are both deep and generalizable.

4. CustomFERCNN with CBAM and Kernels (Best Model)

Leverages attention mechanisms to dynamically focus on emotionally salient regions of the face.

- Kernel Layer to focus model attention on key areas of the face: eyes, nose, mouth, jaw
- Conv Block 1 + CBAM (64 filters): Detects fine-grained local features such as eye shape or eyebrow position, with attention enhancing focus on expressive microregions like eye squints or brow raises.
- Conv Block 2 + CBAM (128 filters): Learns mid-level spatial compositions, emphasizing important channels related to cheek movement, lip curvature, and inter-region relationships crucial for emotion detection.
- Conv Block 3 + CBAM (256 filters): Captures high-level contextual patterns and emotional asymmetries (e.g., one-sided smiles, tension), refined further through spatial and channel-wise attention.
- Flatten + FC Layer (512 → 7): Converts refined spatial representations into class probabilities for the seven emotion categories, using dropout to mitigate overfitting.

Evaluation Strategy: K-Fold Cross-Validation

To ensure model robustness and avoid overfitting to any specific data split, 3-fold cross-validation was employed. The dataset was partitioned into three subsets, and in each fold, one served as validation while the others were used for training.

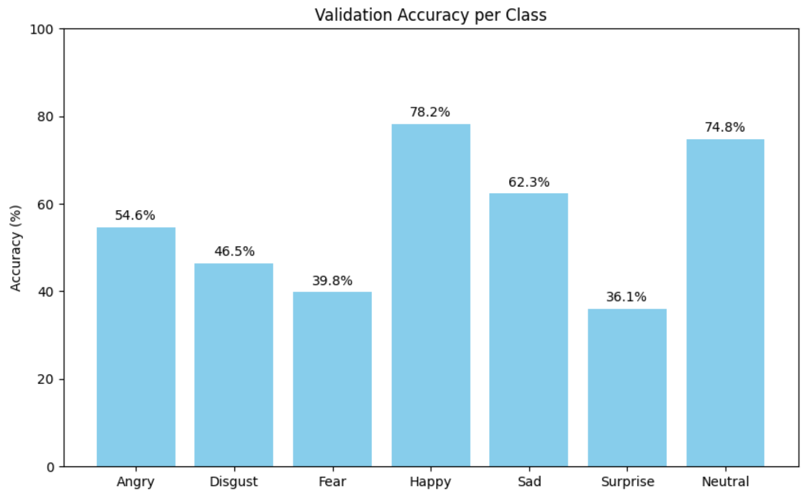
Each model was trained for 15–20 epochs using Adam optimizer and cross-entropy loss. Performance was tracked using validation accuracy, and test accuracy was reported on a held-out set. However, for the final results the built-in train and test set were used, but with the proper parameters and settings found from the K-fold training.

Results

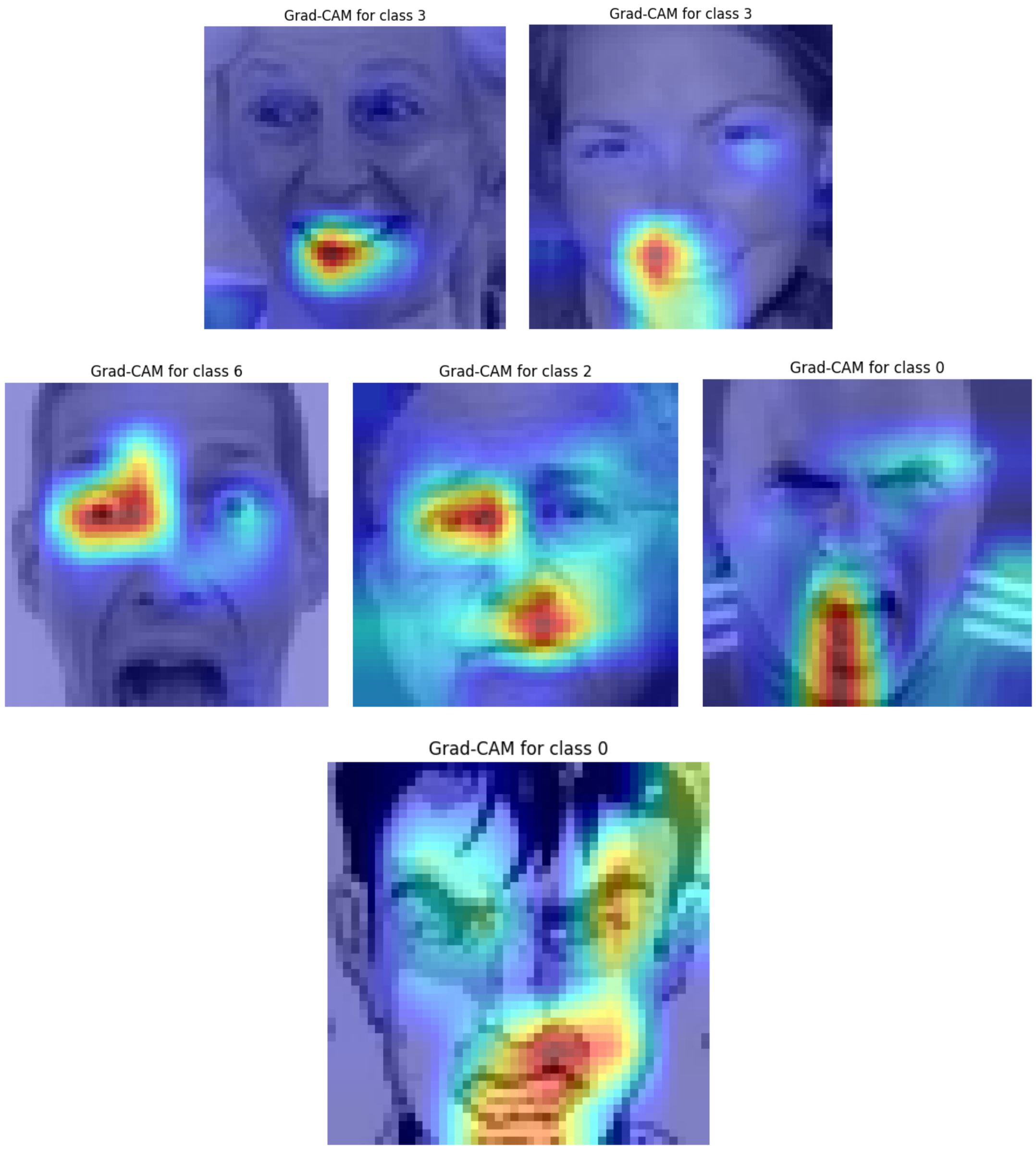
Model	Final Test Accuracy
EmotionCNN (baseline)	58.26%
CNN w/ Custom Kernels	59.72%
Simple ResNet	53.13%
Custom CNN + CBAM +Kernel	62.18%

The CBAM model showed the best results, significantly improving classification of more ambiguous expressions like “fear” and “disgust.” Misclassifications were most frequent between “sad” and “neutral,” which share visual overlap.

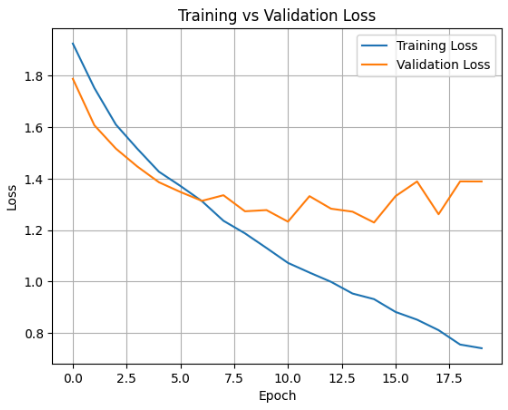
The below figures show results from our testing:

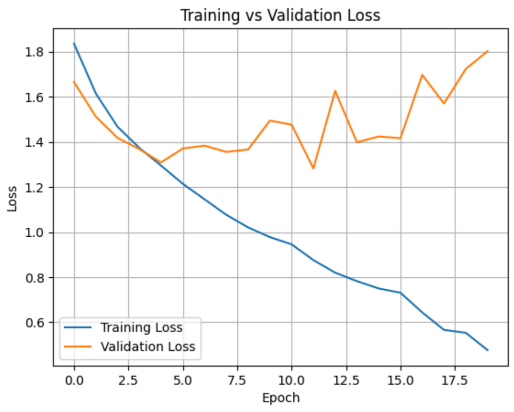


With our final model, class imbalance for validation accuracy was still quite high, with Neutral and Happy being the only two facial expressions able to reach into the 70%s for accuracy, and the lowest of Surprise sitting at 36.1% accuracy. We imagine that these results reflect a wider range of facial expression within the more complex emotions, whereas the simpler emotions like happy are easier for the model to detect. Distinct features like upward curve in lips seemed to play a large part in the identification of happiness, as shown by the facial expressions below:


Grad-CAM for class 3
Grad-CAM for class 3
Grad-CAM for class 6
Grad-CAM for class 2
Grad-CAM for class 0
Grad-CAM for class 0

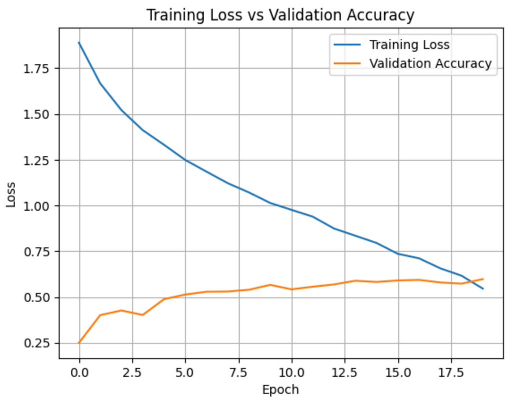
When looking at the effect of certain parameters on our model, we aimed to visualize the results into graphical representations in order to more easily see and be able to compare the models. The first comparison of these was between the use of L2 regularization and the lack of use:

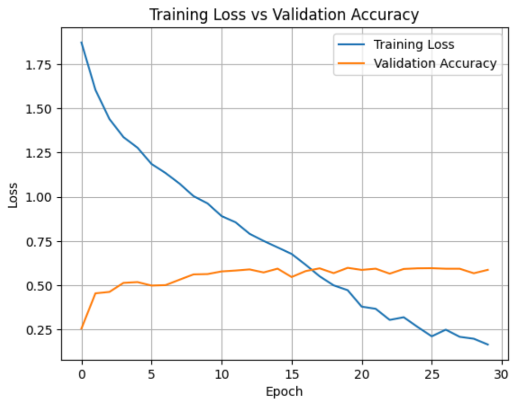




From the above, we can clearly see the impact of L2 regularization preventing overfitting through adding a penalty term to the loss function, which is proportional to the the sum of squares of the weights of the model. The prevention of overfitting reduced the validation loss greatly by the 20th Epoch from 1.8 to just shy of 1.4.

Looking now at the overall performance of our model, we are able to see the accuracy finally attained by our 'CustomFERCNN with CBAM and Kernels':





From the above left, we can see the optimal performance of our model at 20 Epochs where the accuracy of 62.18% was achieved, and the figure on the right shows what 30 Epochs of training would give for loss and accuracy - reduction in training loss but no further improvements in accuracy.

Challenges

- Data Imbalance: "Disgust" and "fear" were underrepresented. Class-weighting in the loss function partially corrected this, however a more balanced dataset would likely improve on some of the class imbalances seen in the earlier figures.

- Each emotion had different key facial areas which impacted their classification; happiness focused on mouth corners tilting, surprise focused on wide eyes, anger focused on open mouth. Getting the model to have kernels and diversified attention onto wherever the key features are was a challenge but the achieved accuracy shows that our approach was competent

Reflection

Discussion & Future Work

Our work with the FER dataset shows that even relatively simple CNN models can begin to understand the complexities of human facial expression, but it takes careful additions onto these models to properly have the model focus on the sections of the face which matter for each emotion. We were ultimately not able to surpass our goal of <68%, but we came respectably close with our final model. Our model behaved as expected; We did not expect to have such issue with overfitting the way we did but it is not entirely unexpected. Our approach changed several times. In the beginning, we did not foresee changing models, but rather tweaking the kernels we gave it. However our early results forced us to pivot and upgrade the model. We hoped ResNet would be able to help with the gradients and help it learn more, but this wasn't entirely the solution as it only slightly helped. Finally we went with the big guns and added an improved CNN with CBAM attention modules, which given the time and scope of the project was all we had left, and got us pretty close to our final goal. If we had to redo the project, we might start with a more capable model from the start, such as a transformer-based model. The overfitting was the main area we were not able to tackle and that we would like to have fixed if we had more time. The class imbalance also was not entirely solved by the optimizer weight change. Our biggest takeaway from this project is that nothing is as easy as it seems. We thought this would be a relatively simple project, given that we were starting with a well-known dataset and using CNNs which we had learned about in class already, but we ran into many roadblocks along the way that we had to address, a few which we were not able to end up completely fixing. Despite setbacks however, we were able to implement many areas from the course into our project, and ultimately are in a place now where given more time and research we could produce a more FER-tailored solution.

Future directions include:

- Integrating temporal data for video-based emotion recognition (computer vision)
- Exploring transformer-based models (e.g., Vision Transformers)
- Incorporating multi-task learning with additional labels like age or gender