DO YOU READ ME LIKE THE EMOJI?

COMPARATIVE ANALYSIS OF CUSTOM CNN AND TRANSFER LEARNING APPROACHES FOR CELEBRITY-SPECIFIC FACIAL EMOTION RECOGNITION

A PREPRINT

Joanne Micaela Dizo
Department of Mathematics
Ateneo de Manila University
Quezon City, Philippines
joanne.dizo@student.ateneo.edu

Caitlyn Sophie Lee
Department of Mathematics
Ateneo de Manila University
Quezon City, Philippines
caitlyn.lee@student.ateneo.edu

Dianne Yumol
Department of Mathematics
Ateneo de Manila University
Quezon City, Philippines
dianne.yumol@student.ateneo.edu

December 5, 2025

ABSTRACT

Facial emotion recognition (FER) is widely studied in human–computer interaction, yet most research focuses on generic faces rather than specific individuals. This project compares two approaches for analyzing facial expressions in K-pop idols: a custom Convolutional Neural Network (CNN) trained from scratch and a transfer learning pipeline using DeepFace with Facenet512. We implement a dual-task system that performs member identification and emotion classification. Emotion training uses the FER2013 dataset (35,887 images across seven emotions), while a curated Katseye dataset of six members supports identity recognition and serves as the test domain for emotion prediction. The pipeline includes Haar cascade face detection, normalization, data augmentation, and feature extraction. The custom models use four convolutional blocks with batch normalization and dropout, whereas the transfer learning approach relies on 512-dimensional embeddings and a pre-trained emotion model. We evaluate accuracy, F1-score, inference time, and confidence on a small proof-of-concept test set. In this setting, the transfer learning method substantially outperforms the custom CNN in accuracy and confidence, while the custom CNN offers faster inference. These results illustrate the challenges of training FER models from scratch in celebrity domains with limited labeled data.

Keywords Facial emotion recognition · Convolutional Neural Networks · Transfer learning · FER2013 · Celebrity recognition · K-pop · Deep learning

1 Introduction

1.1 Background and Motivation

Facial expressions are a primary channel for non-verbal communication, encoding emotional states that people routinely interpret in social interaction. Automatically recognizing these expressions has applications in human–computer interfaces, assistive technologies, media analytics, and human–robot interaction. With the rise of social media and

digital entertainment, particularly in the K-pop industry, there is growing interest in systems that can analyse celebrity facial expressions for fan engagement, content curation, and sentiment analysis.

1.2 Dataset Overview

This study utilizes two primary datasets:

The FER2013 dataset contains 35,887 grayscale images of size 48×48 pixels, labeled with seven emotion categories: angry, disgust, fear, happy, sad, surprise, and neutral. This dataset, introduced by Goodfellow et al. [1] in the "Challenges in Representation Learning" competition, has become a standard benchmark for emotion recognition research. Previous studies report test accuracies ranging from roughly 63% to 75%, depending on model architecture and training strategy. We curated a dataset of six Katseye members (Daniela, Lara, Manon, Megan, Sophia, and Yoonchae) containing RGB color images at higher resolution (128×128 pixels). This dataset serves as our domain-specific training set for member recognition and as the test domain where we ask whether a model trained on FER2013 can generalize to higher-quality, stylized celebrity photos.

1.3 Research Question

This study addresses the question: *Does a custom CNN trained from scratch outperform a transfer learning approach for celebrity-specific facial emotion recognition and member identification?*

1.4 Contributions

(1) We apply FER techniques to a K-pop celebrity context and build a dual-task system for identity and emotion; (2) we compare a from-scratch CNN approach to transfer learning under realistic data scarcity; and (3) we document the limitations and failure modes that arise when deploying FER2013-trained models on stylized celebrity images.

2 Related Work

2.1 Facial Emotion Recognition with CNNs

Convolutional Neural Networks have demonstrated strong performance in facial emotion recognition, often replacing hand-crafted features such as Local Binary Patterns (LBP) or ORB with learned representations. On FER2013, a wide range of CNN architectures have been explored, with test accuracies commonly reported in the 63–75% range. Data augmentation is frequently used to mitigate overfitting in the relatively small and low-resolution dataset. Recent work has reported accuracies beyond 80% by carefully tuning architectures such as CNN-10 and combining them with aggressive augmentation [6], illustrating how much performance can vary with design choices.

2.2 Transfer Learning in Face Recognition

Transfer learning has become a dominant paradigm in computer vision, especially when annotated data are limited. DeepFace and FaceNet architectures [2] learn face representations on large-scale datasets and then expose them as embedding spaces that can be reused in downstream tasks. Facenet512, which generates 512-dimensional embedding vectors, enables efficient face matching using simple distance metrics such as Euclidean distance. For tasks like celebrity recognition, these pre-trained embeddings often provide a stronger starting point than training an identity model from scratch on a small, group-specific dataset.

2.3 Research Gap

While there is substantial research on generic emotion recognition and on face recognition as separate problems, less work directly examines their intersection: recognizing emotions of *specific* individuals, particularly celebrities, in conditions that more closely resemble real social media content. Our study contributes to this gap by examining how a custom CNN and a transfer learning pipeline behave when asked to perform both identity recognition and emotion classification on images of a single K-pop group.

3 Methods

3.1 Overview of Methodology

Figure 1 presents our experimental pipeline, which consists of five main stages: data collection and preprocessing, model development along two parallel tracks (custom CNN and transfer learning), training and optimization, evaluation, and comparative analysis.

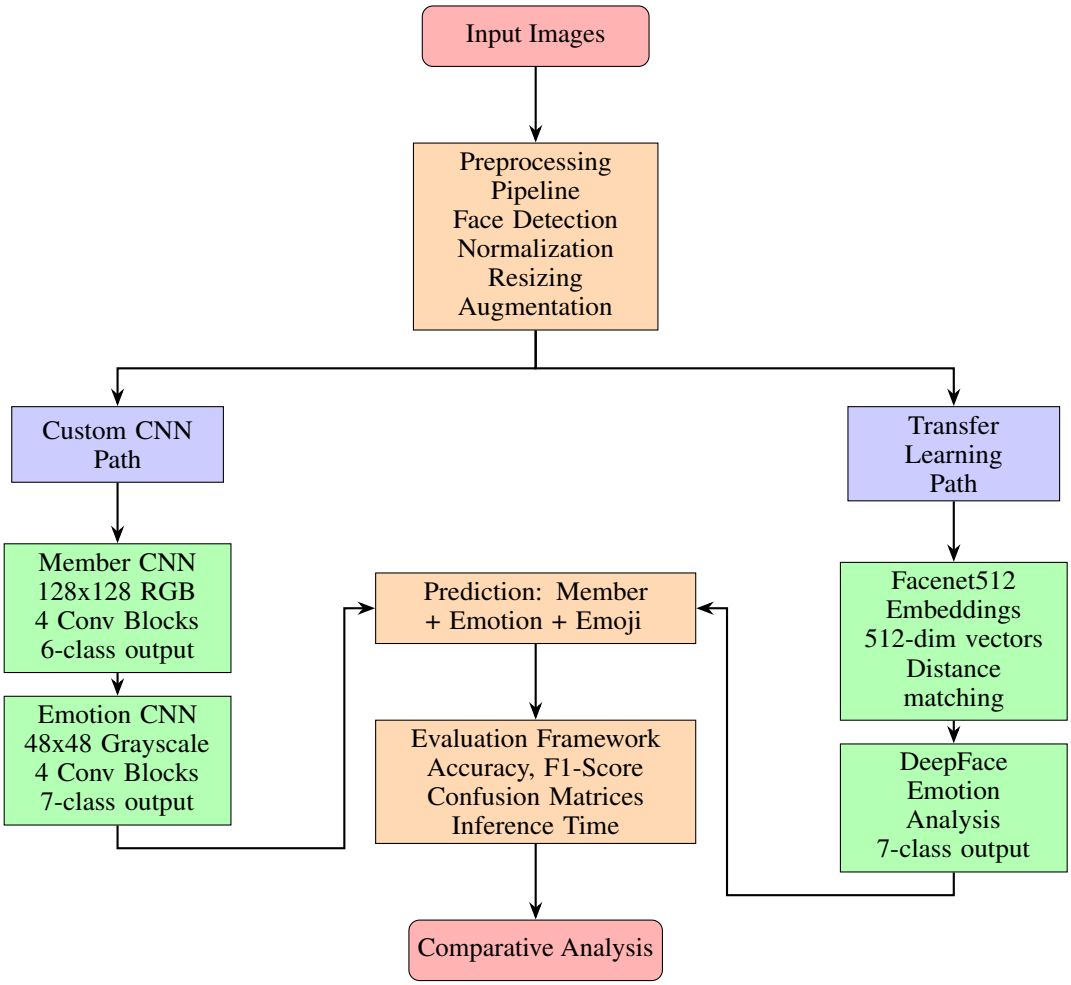

Figure 1: System architecture showing the complete pipeline from raw images to comparative evaluation.

3.2 Data Preprocessing

3.2.1 Face Detection

We employ Haar cascade classifiers [4] (Viola–Jones algorithm) for face detection, chosen for their computational efficiency and reliability for frontal faces in relatively controlled conditions. For each image, the cascade detector identifies face regions of interest, which are then cropped and passed down the pipeline.

3.2.2 Image Normalization and Resizing

For emotion recognition on FER2013, images are kept in grayscale and resized to 48×48 pixels. Pixel values are normalized to the $[0, 1]$ range by dividing by 255, and we add a singleton channel dimension to obtain tensors of shape $(48, 48, 1)$.

For member recognition on the Katseye dataset, we retain the RGB channels and resize images to 128×128 pixels to preserve more detail relevant to identity. Each channel is normalized to $[0, 1]$, resulting in tensors of shape $(128, 128, 3)$. This split reflects the different roles of the two models: the emotion model must be compatible with FER2013, while the member model is tailored to the higher-resolution domain.

3.2.3 Data Augmentation

To improve generalization and reduce overfitting, we apply data augmentation using Keras' `ImageDataGenerator`. The emotion model uses relatively mild augmentation: rotations of up to $\pm 10°$, width and height shifts of up to $\pm 10\%$, horizontal flips, zooms of up to $\pm 10\%$, and nearest-neighbor filling. The member model uses more aggressive augmentation—rotations up to $\pm 20°$, translations up to $\pm 20\%$, horizontal flips, zooms of up to $\pm 20\%$, and shear of approximately 15%—reflecting the fact that we have far fewer training samples per member (roughly 10–50 images) than per emotion class in FER2013.

3.3 Custom CNN Architecture

3.3.1 Member Recognition Model

Our custom member recognition CNN employs a four-block architecture.

Block 1: Conv2D(32 filters, 3×3, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ Conv2D(32 filters, 3×3, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling(2×2) $\rightarrow$ Dropout(0.25)

Block 2: Conv2D(64 filters, 3×3, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ Conv2D(64 filters, 3×3, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling(2×2) $\rightarrow$ Dropout(0.25)

Block 3: Conv2D(128 filters, 3×3, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ Conv2D(128 filters, 3×3, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling(2×2) $\rightarrow$ Dropout(0.25)

Block 4: Conv2D(256 filters, 3×3, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling(2×2) $\rightarrow$ Dropout(0.25)

Fully Connected: Flatten $\rightarrow$ Dense(512, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ Dropout(0.5) $\rightarrow$ Dense(256, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ Dropout(0.5) $\rightarrow$ Dense(6, Softmax)

The model has approximately 4.2 million trainable parameters. The progressive increase in filters ($32 \rightarrow 64 \rightarrow 128 \rightarrow 256$) allows the network to capture increasingly complex facial features, from edges and simple textures to more abstract identity cues. Batch normalization after each convolution stabilizes training and accelerates convergence, while dropout in both convolutional and dense layers helps mitigate overfitting on the relatively small member dataset. Two dense layers with 512 and 256 units provide sufficient capacity to model the six-class identity problem without exploding the parameter count.

3.3.2 Emotion Recognition Model

The emotion model uses a deeper architecture optimized for the seven-class FER2013 task.

Block 1: Conv2D(64 filters, 3×3, ReLU, padding='same') $\rightarrow$ BatchNorm $\rightarrow$ Conv2D(64 filters, 3×3, ReLU, padding='same') $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling(2×2) $\rightarrow$ Dropout(0.25)

Block 2: Conv2D(128 filters, 3×3, ReLU, padding='same') $\rightarrow$ BatchNorm $\rightarrow$ Conv2D(128 filters, 3×3, ReLU, padding='same') $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling(2×2) $\rightarrow$ Dropout(0.25)

Block 3: Conv2D(256 filters, 3×3, ReLU, padding='same') $\rightarrow$ BatchNorm $\rightarrow$ Conv2D(256 filters, 3×3, ReLU, padding='same') $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling(2×2) $\rightarrow$ Dropout(0.25)

Block 4: Conv2D(512 filters, 3×3, ReLU, padding='same') $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling(2×2) $\rightarrow$ Dropout(0.25)

Fully Connected: Flatten $\rightarrow$ Dense(1024, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ Dropout(0.5) $\rightarrow$ Dense(512, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ Dropout(0.5) $\rightarrow$ Dense(7, Softmax)

This model has approximately 7.8 million trainable parameters. Compared to the member model, the emotion model uses more filters per layer and deeper fully connected layers ($64 \rightarrow 128 \rightarrow 256 \rightarrow 512$ filters, then 1024 and 512 units) to capture subtle differences between facial expressions in the low-resolution FER2013 images. Padding='same' helps preserve spatial dimensions, which is important given the small input size. Higher dropout rates are used to cope with the noisiness and class imbalance known in FER2013.

3.3.3 Training Configuration

We train both models using the Adam optimizer, with an initial learning rate of 0.001 for the member model and 0.0001 for the emotion model. Adam's adaptive learning rate is well-suited to CNN training on image data.

The loss function for both tasks is categorical cross-entropy, the standard choice for multi-class classification. We employ several callbacks: early stopping (monitoring validation loss with patience of 10–15 epochs) to avoid overfitting, learning rate reduction on plateau (reducing the learning rate by a factor of 0.5 if validation loss stops improving), and model checkpointing based on validation accuracy to retain the best weights.

Batch sizes are set to 32 for the member model and 64 for the emotion model. We allow training for up to 50 epochs, with 20% of the training data reserved for validation in each case.

3.4 Transfer Learning Approach

3.4.1 Member Recognition with Facenet512

We employ DeepFace's Facenet512 model [3] for member recognition.

Process: We begin with feature extraction. For each member's images, we generate 512-dimensional embedding vectors using Facenet512. The embeddings for all training images are stored in a searchable database. Then, for a test image, we (1) generate a 512-dimensional embedding, (2) compute Euclidean distances to all stored embeddings, (3) assign the identity of the nearest neighbor (minimum distance), and (4) convert the distance to a heuristic confidence score: $\text{confidence} = 1 - \text{normalized\_distance}$.

The Euclidean distance is defined as

$$\text{distance} = \|\text{test\_embedding} - \text{stored\_embedding}\|.$$

A threshold of approximately 0.6 is used as an empirical cutoff for deciding whether a match is strong.

3.4.2 Emotion Recognition with DeepFace

For emotion detection, we use DeepFace's built-in emotion analysis. DeepFace returns a probability distribution over seven emotions; we take the dominant emotion (maximum probability) as the predicted label and log both the dominant confidence and the full distribution.

This pre-trained emotion model provides a strong off-the-shelf baseline and lets us ask how far a custom CNN trained from scratch on FER2013 can actually go in a realistic test scenario.

3.5 Evaluation

We evaluate both approaches on a small test set of three Katseye images with ground truth labels for identity and emotion. Metrics include overall accuracy, weighted F1-score, average inference time, and average confidence. Confusion patterns are examined qualitatively. Given the tiny test set, all results should be interpreted as descriptive and illustrative rather than statistically conclusive.

3.6 Implementation Details

We implemented the system in Python 3.9 using TensorFlow 2.15 (with Keras), OpenCV 4.8, DeepFace 0.0.79, and scikit-learn 1.3. Training the custom member CNN for up to 50 epochs took about 30–45 minutes, while the emotion CNN on FER2013 required roughly 2–3 hours; building the Facenet512 embedding database for transfer learning took around 10–15 minutes per member. Models were stored both as Keras `.h5` files and as separate weights-plus-architecture definitions for cross-environment compatibility.

4 Results and Discussion

4.1 Experimental Setup

We evaluated both approaches on a test set of three Katseye images with ground truth labels for identity and emotion. This is an extremely small test set and is best understood as a proof-of-concept sanity check rather than a statistically meaningful benchmark; the goal is to see how the two systems behave on real Katseye images, not to claim precise generalization performance. Our test set comprises of the following members and emotions: Sophia (happy), Manon (sad), Daniela (surprise).

4.2 Member Recognition Performance

Table 1 presents the member recognition results for both approaches.

Table 1: Member Recognition Performance Comparison

Metric	Custom CNN	DeepFace (Facenet512)	Winner
Overall Accuracy	33.33%	66.67%	DeepFace
F1-Score (Weighted)	0.22	0.56	DeepFace
Avg Inference Time	103.84ms	1093.96ms	Custom CNN
Avg Confidence	15.59%	44.58%	DeepFace

DeepFace achieves roughly double the accuracy of the custom CNN for member recognition on this small test. In per-member terms, both models correctly identify Sophia, only DeepFace correctly identifies Manon, and both misclassify Daniela as Sophia. The tendency to predict Sophia suggests that her features may occupy a central region in the learned feature space, making her a "default" class under uncertainty.

In terms of efficiency, the custom CNN is around $10.5\times$ faster, with a mean inference time of around 104 ms versus approximately 1094 ms for DeepFace. This reflects the overhead of loading and running the larger pre-trained models and computing embeddings. However, the custom CNN's low average confidence of 15.59% indicates that its predictions are often hesitant and poorly calibrated, limiting the practical value of its speed advantage in the current form.

4.3 Emotion Recognition Performance

Table 2 summarizes the emotion recognition results.

Table 2: Emotion Recognition Performance Comparison

Metric	Custom CNN	DeepFace	Winner
Overall Accuracy	0.00%	33.33%	DeepFace
F1-Score (Weighted)	0.00	0.22	DeepFace
Correct Predictions	0/3	1/3	DeepFace

The detailed predictions in Table 3 reveal that the custom CNN largely collapses onto "happy" or "sad", while DeepFace shows slightly more variety but still struggles.

Table 3: Emotion Prediction Details

Ground Truth	Custom CNN	DeepFace
Happy	Sad	Happy
Sad	Happy	Happy
Surprise	Happy	Fear

The custom CNN achieves 0% accuracy, predicting "happy" for two out of three images and "sad" for the remaining one. Despite training on FER2013, the model does not learn emotion boundaries that transfer well to the Katseye domain. Possible contributing factors include class imbalance in FER2013, insufficient regularization or hyperparameter tuning, and the domain shift from low-resolution grayscale faces to higher-quality RGB celebrity images. DeepFace correctly identifies one emotion (happy for Sophia) and misclassifies the other two, confusing sad with happy and surprise with fear. A qualitative inspection of the images suggests that both models are biased toward positive or high-arousal emotions, which are overrepresented in public images of idols.

4.4 Inference Speed and Efficiency

Table 4 reports inference time statistics for both approaches.

The custom CNN is significantly faster on average and has less variability in inference time, which is consistent with its smaller size and simpler runtime pipeline. DeepFace's higher mean and large standard deviation reflect the cost of loading and running a heavier architecture; the slowest calls likely correspond to first-use overhead or cold starts. In a real application, one could amortize this overhead by keeping the model in memory between calls, but the basic trade-off remains: transfer learning offers better accuracy at the cost of speed and resource usage.

Table 4: Inference Time Statistics

Model	Mean (ms)	Std Dev (ms)	Min (ms)	Max (ms)
Custom CNN	103.84	69.33	53.35	201.87
DeepFace	1093.96	1014.64	310.38	2526.79

4.5 Failure Modes and Qualitative Observations

Both models show systematic biases. For member recognition, the repeated prediction of Sophia suggests that either her embeddings are centrally located in the representation space or that her images share visual characteristics with other members (for example, similar styling, camera angle, or expression). With such a small test set, we cannot definitively attribute the bias, but the pattern points to the need for more diverse and balanced training data per member.

For emotion recognition, the custom CNN's failure to correctly classify any emotions, combined with its strong bias toward "happy", is a sign of poor generalization. Training on FER2013's grayscale faces does not translate smoothly to the Katseye domain, where lighting, makeup, and camera quality differ substantially. Moreover, FER2013 itself is noisy and imbalanced, and our training did not exhaustively explore the hyperparameter space. DeepFace's moderate performance, while better, still makes mistakes on subtle expressions and may reflect similar biases in its training data.

4.6 Summary of Comparative Findings

Several factors likely contribute to the custom CNN's underperformance compared to DeepFace. The member model is trained on only 10–50 images per Katseye member, which is extremely limited for a 4.2-million-parameter network; augmentation cannot fully compensate for the lack of diverse real examples. In contrast, Facenet512 has been trained on millions of identities, so even a simple nearest-neighbor classifier in its embedding space benefits from a strong prior.

The emotion model is trained on FER2013 but deployed on a different domain. The jump from 48×48 grayscale faces to higher-resolution, stylized RGB images introduces a distribution shift that we did not explicitly address. Without domain adaptation or fine-tuning on Katseye-like images, it is not surprising that the model struggles. The training process itself could also be improved: we did not deeply analyse training and validation curves for signs of overfitting or underfitting, and our hyperparameter search was limited.

DeepFace's stronger performance is consistent with its design and training history. Facenet512's embeddings encode identity information learned from a very large and diverse dataset, which makes them robust even when fine-tuning data are scarce. Similarly, DeepFace's emotion model draws on large-scale emotion corpora, giving it a broader notion of what different expressions look like across subjects and conditions. In our experiments, this translates into higher accuracy and more confident predictions, even though the model was not specifically trained on Katseye.

The most important limitation of our evaluation is the size of the test set: with only three images, a single misclassification changes accuracy by 33.3 percentage points, and there is no meaningful way to compute confidence intervals or perform robust statistical tests. The present results are best interpreted as a snapshot of model behavior on a few hand-picked examples, highlighting potential issues and failure modes rather than providing a final verdict.

Table 5: Overall Model Comparison

Criterion	Custom CNN	DeepFace	Winner
Member Recognition	33.33%	66.67%	DeepFace
Emotion Recognition	0.00%	33.33%	DeepFace
Inference Speed	103.84ms	1093.96ms	Custom CNN
Training Required	Yes ($\sim$3 hours)	No (pre-trained)	DeepFace
Model Size	$\sim$12 MB	$\sim$100 MB	Custom CNN
Confidence	15.59%	44.58%	DeepFace
Overall Winner	–	–	DeepFace

Overall, the transfer learning approach is clearly more effective for this task under our current constraints, although the custom CNN remains attractive from a latency and model-size perspective.

5 Conclusion and Recommendations

5.1 Summary of Contributions

This study compared custom CNN and transfer learning approaches for celebrity-specific facial emotion recognition in the context of the Katseye girl group. We built and trained two CNNs from scratch for member identification and FER2013-based emotion recognition, and contrasted them with a DeepFace pipeline using Facenet512 embeddings and a pre-trained emotion model. We evaluated both approaches on a small test set with respect to accuracy, F1-score, inference speed, and confidence, and documented practical challenges arising from limited data and domain shift.

5.2 Answers to the Research Question

Within the limitations of our evaluation, transfer learning via DeepFace substantially outperformed the custom CNN on both member and emotion recognition. DeepFace achieved roughly double the accuracy for member recognition and non-zero accuracy for emotion recognition, whereas the custom emotion model failed to correctly classify any of the test images. Although the custom CNN was about an order of magnitude faster at inference and more compact in terms of storage, its low confidence and poor accuracy make it unsuitable for deployment in its current form.

These findings support the broader lesson that pre-trained models are often a better starting point than training large CNNs from scratch when labeled data are scarce. Leveraging knowledge from millions of images provides a robustness that is difficult to match with a few dozen examples per class. At the same time, our results also suggest that with more data, better regularization, or fine-tuning strategies, a custom model could still be useful, especially where latency or deployment constraints make heavy models impractical.

5.3 Limitations

Several limitations should be considered when interpreting these results. The test set contains only three images, which severely limits statistical power and generalizability. The custom member model is trained on just 10–50 images per member, a tiny amount for a multi-million-parameter network. The emotion model is trained on 48×48 grayscale FER2013 faces but evaluated on higher-resolution RGB celebrity images, introducing a domain mismatch that we did not explicitly address. Our hyperparameter tuning was limited, and we did not systematically analyse training and validation curves for overfitting or underfitting. All experiments were conducted using CPU-based inference, which may underrepresent the real-time potential of the custom CNN on GPU hardware. Finally, our focus on a single K-pop group means that results may not generalize to other celebrities or demographics without further study.

5.4 Future Work

Given more time and resources, several extensions would be natural. First, expanding the Katseye dataset and including additional K-pop groups would allow more robust evaluation and reduce the impact of idiosyncratic images. Second, fine-tuning pre-trained backbones (e.g., ResNet or Facenet variants) on Katseye-specific data could combine the strengths of transfer learning with the flexibility of custom architectures. Third, incorporating attention mechanisms could help identify which facial regions are most informative for identity and emotion decisions, improving interpretability. Finally, extending the system to live video processing and integrating additional modalities such as audio or textual context could better reflect real-world use cases where emotion is read from more than just a single static frame.

References

[1] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing*, pages 117–124. Springer, Berlin, Heidelberg, 2013.

[2] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[3] Sefik Ilkin Serengil and Alper Ozpinar. LightFace: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 23–27. IEEE, 2020.

[4] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, volume 1, pages I–I. IEEE, 2001.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[6] Emmanuel Gbenga Dada, et al. Facial emotion recognition and classification using the convolutional neural network-10 (CNN-10). *Applied Computational Intelligence and Soft Computing*, 2023.

[7] Shubham Singh. Facial expression recognition using convolutional neural networks (CNNs) and generative adversarial networks (GANs) for data augmentation and image generation. UNLV Theses, Dissertations, Professional Papers, and Capstones. 4852, 2023.

[8] Shruti Deokar. Enhancing facial emotion recognition using image processing with CNN. Master's Projects. 1254. San José State University, 2023.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.