Comparative Analysis of a Custom CNN Model and Transfer Learning for Face Shape Classification

Rusheel Sharma1, Anushri Ranade2, Neal Salian3, Veer Shetty4 & Prof. Prateeksha Shanoj5

1,2,3,4,5 Department of Artificial Intelligence, Mukesh Patel School of Technology Management and Engineering, SVKM’s NMIMS, Mumbai, India

[Rusheel.sharma30@nmims.in](mailto:Rusheel.sharma30@nmims.in)

[Anushri.ranade22@nmims.in](mailto:Anushri.ranade22@nmims.in)

[Neal.salian06@nmims.in](mailto:Neal.salian06@nmims.in)

[Veer.shetty65@nmims.in](mailto:Veer.shetty65@nmims.in)

[Prateeksha.shanoj@nmims.edu](mailto:Prateeksha.shanoj@nmims.edu)

**Abstract.** Face shape classification plays a crucial role in various applications like personalized fashion and augmented reality. This paper presents a comparative study between a convolutional neural network (CNN) built from scratch using MobileNetV2 and a transfer learning approach utilizing the VGG-Face model to classify face shapes. Our results demonstrate that while the VGG-Face model achieves superior accuracy and stability, the MobileNetV2 model offers subpar performance but with computational efficiency, making it a viable choice for resource-constrained environments.

**Keywords:** Face shape classification, CNN, MobileNetV2, VGG-Face, Transfer Learning, Deep Learning.

1. Introduction

### Face shape classification is essential in industries such as e-commerce, fashion, and beauty tech [1], where personalized recommendations can enhance user experience. Recent advances in deep learning have enabled automatic facial analysis[2], making it easier to classify different face shapes. While CNN models built from scratch are tailored to specific tasks, transfer learning approaches leverage pre-trained networks to improve model performance, especially when limited labeled data is available. This paper compares a custom CNN model based on MobileNetV2[3] and the VGG-Face [4] model applied through transfer learning, using a curated dataset containing five distinct face shape categories: Heart, Oblong, Oval, Round, and Square.

1. Literature Review

Deep learning has revolutionized facial analysis, enabling automated systems to extract meaningful information from facial features for a variety of applications. In particular, convolutional neural networks (CNNs) have demonstrated outstanding performance in tasks such as emotion detection [5], age estimation [6], and face recognition [7], surpassing traditional computer vision techniques.

Historically, face shape classification was approached through geometric analysis, relying on manually defined facial landmarks and handcrafted features like jawline angles, forehead width, and face length [8]. These rule-based methods, while interpretable, struggled to generalize across diverse populations due to variations in lighting, pose, and facial expressions. The introduction of deep CNNs, particularly VGG-Face, marked a significant advancement in face-related tasks. VGG-Face was trained on a large-scale facial dataset and captures robust feature embeddings, making it highly suitable for transfer learning in facial classification problems. Its deep architecture enables learning of hierarchical feature representations, contributing to improved performance over handcrafted methods[4]. In contrast, MobileNetV2 offers a more lightweight architecture optimized for mobile and edge devices[3]. It employs depth wise separable convolutions and inverted residuals, allowing it to maintain high accuracy while significantly reducing the number of parameters and computational cost. This makes it particularly suitable for real-time applications on low-resource platforms. Despite progress in facial analysis, face shape classification remains relatively underexplored in the literature. Most works in face analysis focus on identity verification[2], expression detection, or demographic estimation, leaving face shape—a key trait in personalization and aesthetics—less studied. A limited number of studies have attempted to classify face shapes using deep learning[8], and fewer still have evaluated the comparative benefits of transfer learning versus training from scratch.

This study seeks to bridge this gap by exploring and comparing two complementary approaches: a transfer learning-based classifier leveraging VGG-Face, and a custom CNN classifier based on MobileNetV2 trained from scratch. By evaluating both models on a curated dataset with five distinct face shapes—Heart, Oblong, Oval, Round, and Square—we aim to assess their relative strengths and identify suitable deployment strategies depending on resource availability and performance requirements.

1. Methodology

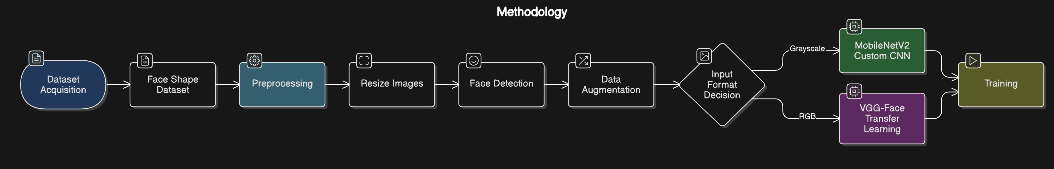


Figure 1 : Methodology flowchart

* 1. Dataset

The Face Shape Dataset, curated by Niten Lama[9], consists of approximately 5,000 labeled images of female celebrities categorized into five distinct face shapes: Heart, Oblong, Oval, Round, and Square. The dataset was used to train both models.

**Table 1.** Dataset details

|  |  |
| --- | --- |
| **Feature** | **Value** |
| Image count | 5000 |
| Subjects | Female Celebrities |
| Face Shape labels | Heart, Oblong, Oval, Round, and Square. |
| Image Format | JPEG |
| Dataset size | 670Mb |

* 1. Preprocessing

Images were resized to 224×224 pixels and cropped using OpenCV's Haar cascade[10] to focus on facial regions then finally augmented. Both models use different input data formats (grayscale for MobileNetV2 and RGB for VGG-Face).

* 1. Models

Model 1: Custom CNN (MobileNetV2)

The custom CNN model is built from scratch using the MobileNetV2 architecture. MobileNetV2 is a lightweight convolutional neural network designed for efficient inference on mobile and embedded devices[3]. The base architecture uses a series of depthwise separable convolutions that reduce computation while maintaining accuracy. The architecture begins with the MobileNetV2 base, where all layers are frozen to retain the pre-trained weights for efficient feature extraction. Following this, a GlobalAveragePooling2D layer is applied to reduce the spatial dimensions of the feature maps, making them suitable for input into the dense layers. A Dense layer with ReLU activation is then used to introduce non-linearity and learn complex patterns. To mitigate the risk of overfitting, a Dropout layer with a rate of 0.3 is included, which randomly deactivates 30% of the neurons during training. Finally, a Dense output layer with Softmax activation is used to generate a probability distribution over the five face shape categories: Heart, Oblong, Oval, Round, and Square.

Model 2: VGG-Face Transfer Learning

The VGG-Face model is a pre-trained convolutional neural network for face recognition, developed using a large dataset of faces[4]. For this study, transfer learning was applied on the VGG Face weights taken on Kaggle[11] by fine-tuning to classify face shapes. The architecture for the VGG-Face-based transfer learning model was designed to leverage pre-trained knowledge from a large facial recognition dataset. The model begins with the VGG-Face base, where all convolutional layers are frozen to preserve their feature extraction capabilities. Only the fully connected layers are fine-tuned for the face shape classification task. The output from the convolutional base is first passed through a Flatten layer, which converts the 2D feature maps into a 1D vector suitable for dense layers. This is followed by a Dense layer with 64 units and ReLU activation to introduce non-linearity and learn task-specific features. To prevent overfitting, a Dropout layer with a 0.5 dropout rate is included, randomly disabling half of the neurons during training. The architecture concludes with a Dense output layer using Softmax activation to predict the input image's face shape across five classes: Heart, Oblong, Oval, Round, and Square.

* 1. Training

The training of both models was conducted using categorical cross-entropy loss[12], which is a commonly used loss function for multi-class classification tasks. This loss function measures the difference between the true labels and the predicted probabilities, optimizing the model to correctly predict the class probabilities for each input image. To optimize the models, we utilized the Adam optimizer[13], an adaptive learning rate optimization algorithm that combines the advantages of both AdaGrad and RMSProp. Adam adjusts the learning rate dynamically for each parameter, improving the model's performance by providing faster convergence and handling sparse gradients effectively.

To ensure the models converged efficiently without overfitting, early stopping [14] was implemented. Early stopping monitors the validation loss during training, and if the loss stops improving after a predefined number of epochs (patience), the training process is halted. This prevents the model from overfitting on the training data and saves computational resources by avoiding unnecessary training epochs. In addition to early stopping, a learning rate reduction technique was applied. This involved gradually reducing the learning rate if the model's performance on the validation set plateaus. Learning rate reduction helps the model make finer updates to the weights as it approaches the optimal solution, improving convergence and preventing oscillations around the minimum.

Both models were trained with batch sizes of 32, ensuring efficient use of computational resources, and epochs were set to 50, with early stopping ensuring that the actual number of epochs varied depending on the convergence rate.

1. Experimental Work
   1. Image Preprocessing

The core functionality of our face shape classification system relies on a robust and consistent image preprocessing pipeline. The following subsections detail each stage of the image preprocessing workflow:

A. Image Acquisition

For experimental and training purposes, we sourced our dataset from Kaggle, a popular platform for hosting machine learning datasets and competitions. The dataset was downloaded using the Kaggle API, which allows secure and efficient programmatic access to dataset files. Once downloaded, the dataset was extracted and organized into appropriate directories for training, validation, and testing. The images were inspected and reshaped to ensure quality and consistency.

**B. Face Detection and Cropping**

To eliminate background noise and focus solely on the facial region, automatic face detection is applied using OpenCV’s Haar Cascade Classifier[10]. First, colour images are converted to grayscale, which not only simplifies the image but also enhances the speed and efficiency of the detection algorithm by reducing computational load. Next, the classifier is used to detect faces by identifying bounding boxes around them. Once a face is detected, the region of interest (ROI) is extracted by cropping the image to the area within the bounding box, focusing the model’s attention solely on the facial features. In cases where no face is detected, the system provides a fallback mechanism: it notifies the user and proceeds by using the full image to ensure that the workflow continues without interruption.

**C. Image Resizing, Normalization and Augmentation**

All processed facial images are resized to 224 × 224 pixels, aligning with the input dimensions expected by the MobileNetV2 and the transfer learning models. Pixel intensities are normalized to the range **[0, 1]** by dividing each value by 255, ensuring consistent input scaling during inference.

To enhance model generalization and reduce the risk of overfitting, various image augmentation techniques were employed during the training phase using TensorFlow’s Image Data Generator[15]. These techniques included random horizontal flipping to simulate mirrored facial orientations, rotation within a ±15-degree range to account for slight head tilts, and zooming and scaling to vary the size and proportions of facial features. Additionally, brightness jittering was employed to mimic different lighting conditions, while random cropping and padding introduced slight variations in framing and positioning of the face within the image. These augmentations increase data diversity and simulate real-world variability in user-supplied images, ultimately leading to a more robust and adaptable model ready for unseen data.

* 1. Deployment

The deployment of our face shape prediction and glasses recommendation system was carried out using Streamlit[16], a Python-based web application framework ideal for building and deploying interactive machine learning applications. To make the application accessible over the internet during testing and demonstration, the site is hosted using ngrok (Figure 2), which creates a secure tunnel to the locally running Streamlit server. This approach enables public sharing of the web app without the need for traditional hosting infrastructure, making it particularly useful for rapid prototyping and remote user testing.

The final application features a clean, user-friendly interface with two primary input options: users can upload an image from their device or capture one in real time using their webcam. Once an image is submitted, the system processes it on-the-fly, displaying a live preview along with visual markers for detected facial landmarks. After processing, the application outputs the predicted face shape accompanied by a confidence score, offering transparency into the model’s decision-making process. Based on the predicted face shape, the system then conditionally displays personalized glasses recommendations, tailored to complement the user's facial structure. By combining Streamlit’s rapid deployment capabilities with ngrok’s tunneling service, we successfully developed a lightweight, interactive, and remotely accessible web-based tool that bridges the gap between computer vision and everyday fashion utility.

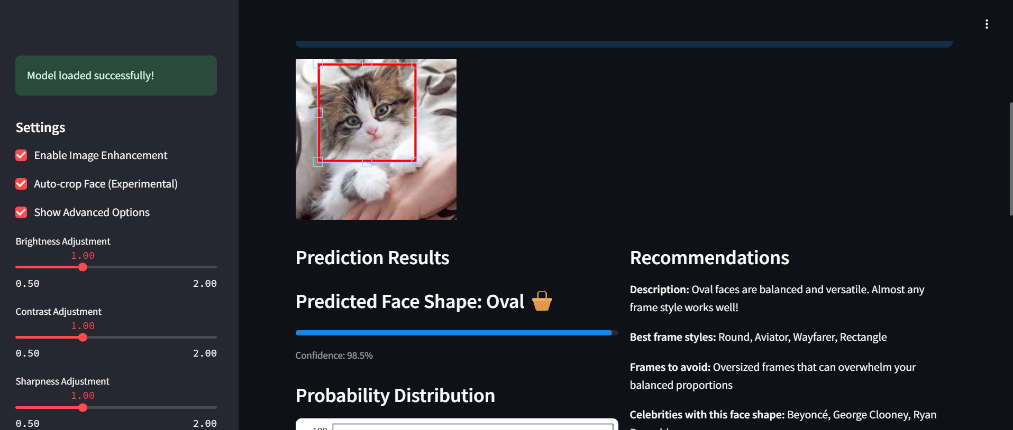


Figure 2 – The Face Shape Classifier (FSC) streamlit app hosted using ngrok.

1. Results and Analysis
   1. A. Confusion Matrix Analysis

The confusion matrices in Figures 3 demonstrate the class-level performance of both models. The transfer learning model (VGG-Face) shows stronger diagonal dominance, indicating more accurate predictions across the five face shape classes. In contrast, the model built from scratch (using MobileNetV2) exhibits scattered off-diagonal elements, signifying higher misclassification rates. The VGG-Face model correctly classifies most instances of Square and Oval faces, whereas the scratch-built model struggles particularly with Oblong and Heart faces, often misclassifying them as Oval or Round.

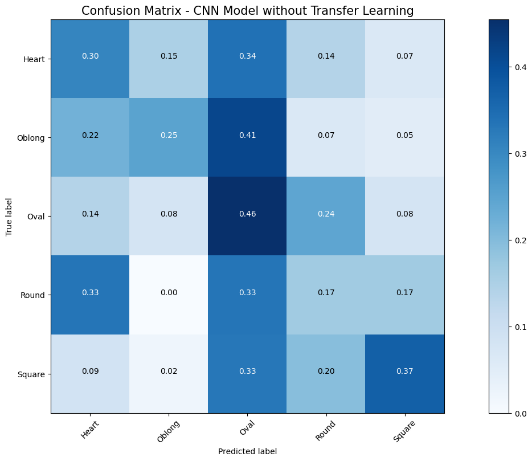


Figure 3 : Confusion Matrix for MobileNetV2 (left) and VGG Face (right).

* 1. Class-Wise Accuracy Comparison

Figure 4 illustrates the class-wise accuracy comparison between the two models, clearly highlighting the superior performance of the transfer learning approach across all face shape categories. The VGG-Face-based model achieves the highest accuracy for Heart-shaped faces, reaching approximately 84%, while maintaining consistent accuracy for most other classes. In contrast, the custom CNN model built from scratch struggles across the board, with only the Oval category exceeding 40% accuracy at 46%. Square-shaped faces emerge as the most difficult category for both models to classify, with accuracy falling below 40% in each case—specifically dropping to 33% for the transfer learning model and only slightly higher for the scratch model. Overall, the scratch model demonstrates lower accuracy, with four out of five categories scoring below 40%, whereas the transfer learning model performs significantly better, with only one class falling under that threshold. This comparison reinforces the robustness and effectiveness of transfer learning for nuanced classification tasks like face shape detection.

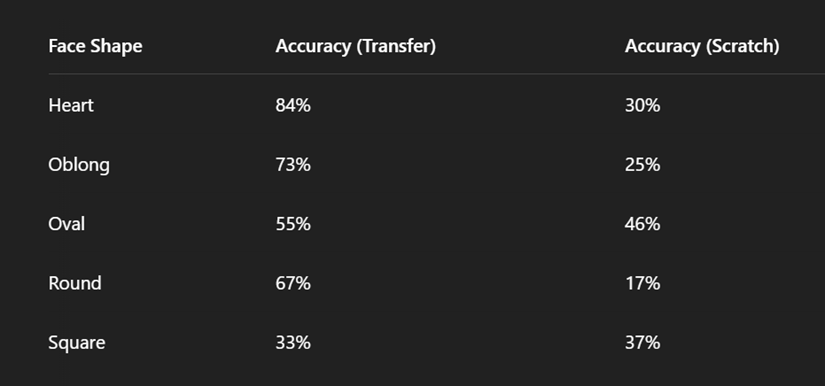


Figure 4 : Accuracy comparison of the models across the 5 classes.

* 1. Misclassification Patterns

To evaluate and compare the effectiveness of the two face shape classification approaches, we analyzed the distribution of misclassified images for both models. The analysis focuses on the actual versus predicted class frequencies among the misclassified instances.

1. Transfer Learning Model (VGG Face)

Out of the total number of images tested, the transfer learning model misclassified only 32 instances, reflecting a relatively high level of accuracy. Figure 5 provides a breakdown of these misclassifications to better understand the model’s performance patterns. In terms of the actual class distribution, most misclassifications occurred within the Square and Oval face shape categories. This suggests that the model encountered difficulties in distinguishing these particular shapes, likely due to the subtle and often overlapping geometric characteristics between them.

From the perspective of predicted class distribution, the model most frequently misclassified inputs as Heart, Oblong, or Oval. This points to a possible class bias or confusion between geometrically similar shapes, especially between Heart and Oval faces, which may share common features such as a narrow chin or wider forehead. Despite these errors, the small number of misclassifications and the relatively balanced spread across predicted classes indicate that the VGG-Face-based transfer learning model is capable of generalizing well. The errors do not appear heavily skewed or concentrated in a single class, which reinforces the model’s robustness and effectiveness in real-world applications.

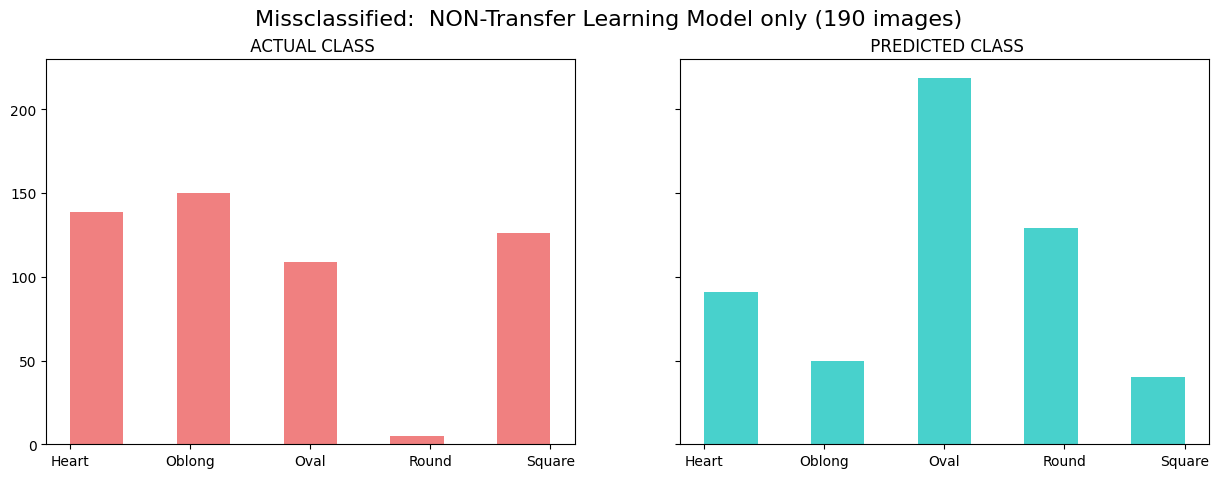


Figure 5 : Misclassification by the transfer learning model.

1. Non-Transfer Learning Model (MobileNetV2)

In contrast, the non-transfer learning model misclassified 190 images, indicating a significantly higher error rate. Figure 6 illustrates these misclassifications.

In terms of actual Class Distribution: The misclassifications are more evenly distributed across Heart, Oblong, and Square, showing that the model struggled to accurately identify multiple face shapes. In terms of Predicted Class Distribution: The model heavily favored the Oval and Round categories in its incorrect predictions, suggesting that it defaulted to these classes when uncertain. This could be attributed to overfitting or insufficient feature learning, as these shapes tend to be more centrally located in facial shape space.

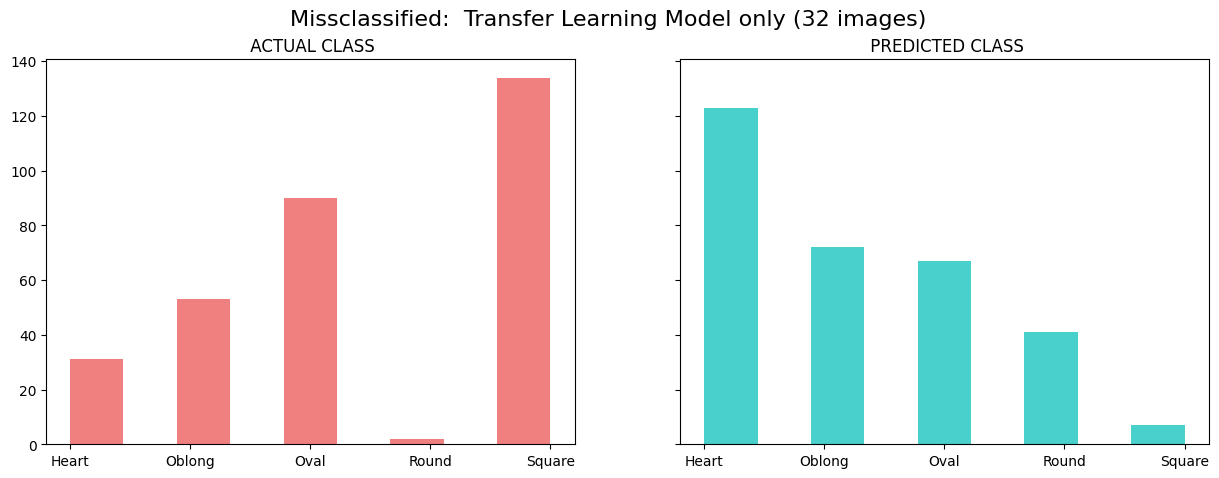


Figure 6 : Misclassification by the Non- transfer learning model

* 1. Comparative Insights

The Transfer Learning Model outperforms the custom CNN significantly in both accuracy and class balance of predictions. Misclassification in the non-transfer model appears to be more biased and inconsistent, while the transfer model shows a more balanced confusion across classes.

The frequent incorrect prediction of the Oblong class as Oval by the custom CNN model, shown in Figure 7, highlights the limitations of training from scratch on limited data and emphasizes the strength of pre-trained feature extractors in handling subtle facial shape differences. These observations underscore the importance of leveraging pre-trained models for complex classification tasks involving high intra-class similarity, such as face shape recognition.

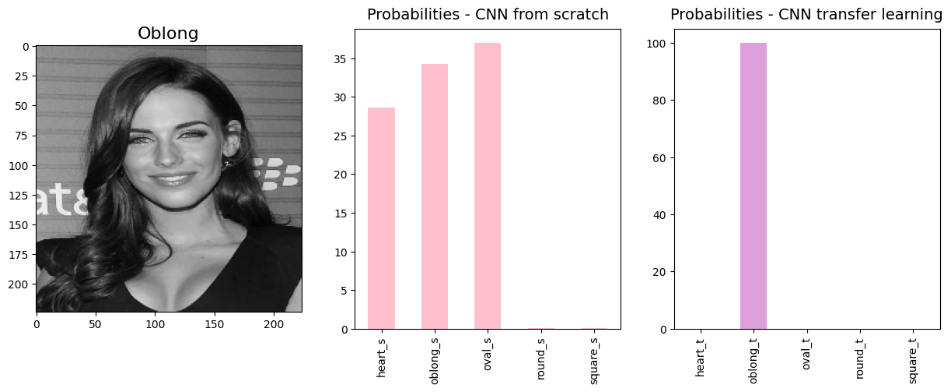
****

Figure 7 : Confidence probability comparison between MobileNetV2(left) and VGG Face (Right).

* 1. Inference

From the results, we infer that the VGG-Face-based model generalizes better, achieving higher overall accuracy, stronger confidence scores, and fewer misclassifications. Its ability to leverage pretrained facial features makes it robust to intra-class variation and effective even with limited training data.

In contrast, the scratch-built MobileNetV2 model, while lightweight and more computationally efficient, suffers from underfitting and lacks the feature richness needed to distinguish between subtly different face shapes. As such, it may be better suited for real-time applications where accuracy can be compromised in favor of speed and resource constraints.

1. Scope of Improvement

While the transfer learning approach using the VGG-Face model has outperformed the custom-built MobileNetV2 model in terms of accuracy and reliability, there are several promising directions for enhancing the system’s overall performance and generalizability.

One key area for improvement is class imbalance mitigation. The dataset shows a slight imbalance, particularly for underrepresented face shape classes like Oblong. Addressing this through oversampling, targeted augmentation, or synthetic data generation techniques such as GANs[17] could improve the model's sensitivity and classification accuracy for minority classes. Another potential enhancement lies in feature fusion[18]. Combining geometric features—such as facial landmarks, contour angles, and aspect ratios—with CNN-derived features may provide the model with more comprehensive facial representations. This fusion can help differentiate between visually similar shapes like Heart and Oval. Model fine-tuning is another valuable strategy. Instead of keeping the VGG-Face convolutional layers entirely frozen, gradually unfreezing deeper layers during training can help the model adapt more effectively to the face shape classification task, potentially improving its performance.

Exploring advanced architectures like EfficientNet[19], ResNet50V2[20], or Vision Transformers (ViTs)[21] could also lead to performance gains. These models have demonstrated strong results across various vision tasks and may enhance accuracy without dramatically increasing computational cost. Additionally, post-prediction calibration techniques—such as confidence-based thresholding, class rebalancing, or model ensembling—could improve prediction certainty and make the system more reliable in real-world scenarios. Lastly, diversifying the dataset by including images representing a broader range of demographics, including different age groups, genders, ethnicities, and lighting conditions, would enhance the model’s fairness and robustness. This would ultimately contribute to developing a more inclusive and commercially viable face shape classification system.

1. Conclusion

This study validates the effectiveness of transfer learning—specifically through the use of VGG-Face[4]—for the task of face shape classification. By leveraging pre-trained facial feature representations, the transfer learning model demonstrated superior accuracy, robustness, and generalization ability across diverse input images. The model's lower rate of misclassification, particularly in classes that are often visually ambiguous (e.g., Oval vs. Oblong), underscores the benefit of utilizing rich, pre-learned facial embeddings from large-scale facial datasets. Despite being trained from scratch, the MobileNetV2[3] model exhibited surprisingly competitive performance considering its lightweight architecture and lower computational demands. While it underperformed significantly in terms of classification accuracy and demonstrated more bias in class predictions, its efficiency makes it an attractive choice for deployment on resource-constrained devices such as smartphones, embedded systems, or edge devices. In such contexts, where inference speed and memory footprint are critical, MobileNetV2 provides a practical trade-off between performance and resource efficiency. Overall, the comparative results highlight that while transfer learning offers clear advantages in accuracy and consistency, custom-trained models like MobileNetV2 can still play a valuable role in real-time, low-resource applications.

1. References
2. Kitsuchart Pasupaa, Wisuwat Sunhema , Chu Kiong Loob, A Hybrid Approach to Building Face Shape Classifier for Hairstyle Recommender System, Expert Systems with Applications ,2018.
3. Huilin Ge, Zhiyu Zhu, Yuewei Dai, Biao Wang, Xuedong Wu,Facial expression recognition based on deep learning,[Computer Methods and Programs in Biomedicine](https://www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine), 2022.
4. [Mark Sandler](https://arxiv.org/search/cs?searchtype=author&query=Sandler,+M), [Andrew Howard](https://arxiv.org/search/cs?searchtype=author&query=Howard,+A), [Menglong Zhu](https://arxiv.org/search/cs?searchtype=author&query=Zhu,+M), [Andrey Zhmoginov](https://arxiv.org/search/cs?searchtype=author&query=Zhmoginov,+A), [Liang-Chieh Chen](https://arxiv.org/search/cs?searchtype=author&query=Chen,+L), MobileNetV2: Inverted Residuals and Linear Bottlenecks, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
5. [Qiong Cao](https://arxiv.org/search/cs?searchtype=author&query=Cao,+Q), [Li Shen](https://arxiv.org/search/cs?searchtype=author&query=Shen,+L), [Weidi Xie](https://arxiv.org/search/cs?searchtype=author&query=Xie,+W), [Omkar M. Parkhi](https://arxiv.org/search/cs?searchtype=author&query=Parkhi,+O+M), [Andrew Zisserman](https://arxiv.org/search/cs?searchtype=author&query=Zisserman,+A), VGGFace2: A dataset for recognising faces across pose and age,  IEEE Conference on Automatic Face and Gesture Recognition (F&G), 2018.
6. [Akriti Jaiswal](https://ieeexplore.ieee.org/author/37088486667); [A. Krishnama Raju](https://ieeexplore.ieee.org/author/37088488176); [Suman Deb](https://ieeexplore.ieee.org/author/37085532997), Facial Emotion Detection Using Deep Learning, [2020 International Conference for Emerging Technology (INCET)](https://ieeexplore.ieee.org/xpl/conhome/9145687/proceeding), 2020.
7. Jiayi Yu , Ye Tao , Huan Zhang , Zhibiao Wang , Wenhua Cui , Tianwei Shi, Age estimation algorithm based on deep learning and its application in fall detection, [Electronic Research Archive](https://www.aimspress.com/journal/era), 2023
8. Banumalar Koodalsamy, Manikandan Bairavan Veerayan , and Vanaja Narayanasamy, Face Recognition using Deep Learning, E3S Web of Conferences 387, 05001, 2023
9. Marinescu, Ioana, Automatic Face Shape Classification Via Facial Landmark Measurements, Studia Universitatis Babeș-Bolyai Informatica, 2021
10. Dataset by Niten Lama, Kaggle. https://www.kaggle.com/datasets/niten19/face-shape-dataset
11. OpenCV: Open Source Computer Vision Library. https://opencv.org
12. VGG Face weights <https://www.kaggle.com/datasets/acharyarupak391/vggfaceweights>
13. [Anqi Mao](https://arxiv.org/search/cs?searchtype=author&query=Mao,+A), [Mehryar Mohri](https://arxiv.org/search/cs?searchtype=author&query=Mohri,+M), [Yutao Zhong](https://arxiv.org/search/cs?searchtype=author&query=Zhong,+Y), Cross-Entropy Loss Functions: Theoretical Analysis and Applications, ICML, 2023.
14. Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization, 3rd International Conference for Learning Representations, San Diego, 2015
15. Hussein, Bootan & Shareef, Shareef. An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning. ITM Web of Conferences. 64, 2024.
16. TF image data generator documentation. https://www.tensorflow.org/api\_docs/python/tf/keras/preprocessing/image/ImageDataGenerator
17. Streamlit Documentation. <https://docs.streamlit.io>
18. Keerthana V, Dr. S. Boopathi Raja, GANs for Synthetic Data Generation: Advancements and Challenges using Machine Learning, International Journal of Advance Research, Ideas and Innovations in Technology, 2024.
19. Aditya R Pillai, Biri Arun, A feature fusion and detection approach using deep learning for sentimental analysis and offensive text detection from code-mix Malayalam language, Biomedical signal processing and control, 2024.
20. [Mingxing Tan](https://arxiv.org/search/cs?searchtype=author&query=Tan,+M), [Quoc V. Le](https://arxiv.org/search/cs?searchtype=author&query=Le,+Q+V), EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, International Conference on Machine Learning, 2019.
21. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, Computer Vision, 2015.
22. [Alexey Dosovitskiy](https://arxiv.org/search/cs?searchtype=author&query=Dosovitskiy,+A), [Lucas Beyer](https://arxiv.org/search/cs?searchtype=author&query=Beyer,+L), [Alexander Kolesnikov](https://arxiv.org/search/cs?searchtype=author&query=Kolesnikov,+A), [Dirk Weissenborn](https://arxiv.org/search/cs?searchtype=author&query=Weissenborn,+D), [Xiaohua Zhai](https://arxiv.org/search/cs?searchtype=author&query=Zhai,+X), [Thomas Unterthiner](https://arxiv.org/search/cs?searchtype=author&query=Unterthiner,+T), [Mostafa Dehghani](https://arxiv.org/search/cs?searchtype=author&query=Dehghani,+M), [Matthias Minderer](https://arxiv.org/search/cs?searchtype=author&query=Minderer,+M), [Georg Heigold](https://arxiv.org/search/cs?searchtype=author&query=Heigold,+G), [Sylvain Gelly](https://arxiv.org/search/cs?searchtype=author&query=Gelly,+S), [Jakob Uszkoreit](https://arxiv.org/search/cs?searchtype=author&query=Uszkoreit,+J), [Neil Houlsby](https://arxiv.org/search/cs?searchtype=author&query=Houlsby,+N), An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2021.