Card Classification Using Deep Learning

by

Tamjidul Hasan
ID: 2104010202285

Rohed Deb Ove
ID: 2104010202296

Semester – 8th, Section – D, Session – Spring 2025

CSE 460 - Pattern Recognition Laboratory

Instructor :

Wazih Ullah Tanzim

Lecturer

Department of Computer Science and Engineering

Premier University

Signature

Department of Computer Science and Engineering

Premier University

Chattogram-4000, Bangladesh

Contents

1 Introduction

The challenge of automated image classification is a cornerstone of modern computer vision, with the specific task of recognizing playing cards holding significant practical value. Applications range from algorithmic gambling systems and real-time casino security to efficient manufacturing and inventory management. Traditional computer vision techniques, which rely heavily on manually defined features or simple template matching, prove inadequate when faced with real-world variables such as extreme lighting changes, partial card occlusion, arbitrary rotation, and diverse card designs.

This project's core objective is to build a highly accurate and resilient Playing Card Classification system capable of classifying images into one of 53 distinct categories (the 52 standard cards plus the Joker). We achieve this by leveraging Transfer Learning, a powerful deep learning methodology that utilizes models pre-trained on massive datasets (like ImageNet) and fine-tunes them for our specific domain.

We present a thorough comparative analysis of two contrasting state-of-the-art Convolutional Neural Networks (CNNs):

1. **MobileNetV2:** Chosen for its superior efficiency, low computational cost, and suitability for potential edge device deployment.

2. **DenseNet-201:** Selected as a high-performance benchmark due to its structural depth and strong feature reuse capabilities.

The report details the entire lifecycle, from data augmentation to model fine-tuning and a rigorous evaluation to determine the optimal architecture offering the best balance of accuracy and computational resources.

2 Literature Review

Research into object recognition has transitioned dramatically over the last decade, moving from reliance on handcrafted features to highly automated deep learning.

2.1 The Evolution of Recognition Systems

Early systems for card recognition were rooted in **classical image processing**. These methods involved isolating the card using color thresholds or edge detection and then employing techniques like **template matching** or invariant feature detectors (e.g., SIFT or SURF) to identify the card's suit and rank. These systems were fast but brittle, failing instantly when card visibility was compromised.

The breakthrough came with the advent of **Deep Convolutional Neural Networks (CNNs)**. Early architectures like LeNet and AlexNet demonstrated the ability to learn complex, hierarchical feature representations directly from pixels, outperforming traditional methods by wide margins.

2.2 The Paradigm Shift: Transfer Learning

The concept of **Transfer Learning** became pivotal. Researchers found that CNNs trained on the massive ImageNet database learned general, low-level visual features (edges, textures, shapes) that could be reused for virtually any new image task. By taking these pre-trained models (like VGG or ResNet) and fine-tuning the final few layers on a smaller, domain-specific dataset, near-state-of-the-art performance could be achieved with significantly reduced training time and data.

2.3 Contemporary CNN Architectures

The field has recently focused on developing models specialized either for maximum accuracy or maximum efficiency:

- **Dense Connectivity (DenseNet-201):** Introduced to address the vanishing gradient problem in very deep networks, DenseNet promotes **feature reuse** by connecting every layer to every subsequent layer in a feed-forward manner. This deep structure is ideal for maximizing accuracy but comes with high memory and computational demands.

- **Efficiency and Lightweight Design (MobileNetV2):** Driven by the need for on-device machine learning, MobileNetV2 employs a core innovation called **depthwise separable convolutions** and **inverted residual blocks**. This architecture drastically reduces the number of parameters and computation required while maintaining high performance, making it a cornerstone for efficient vision tasks.

The current project utilizes this transfer learning methodology, focusing the evaluation on the trade-offs between the efficiency of MobileNetV2 and the depth of DenseNet-201.

3 Dataset Description

The project leverages the **Cards Image Dataset-Classification 6k**, a publicly available collection specifically curated for this task. The dataset is meticulously partitioned to ensure a reliable training, validation, and testing environment.

The entire dataset comprises **6000** image samples categorized into **53** unique classes (52 standard cards plus one Joker). The critical characteristic of this dataset is its **class balance**: the training split contains an equal number of images for all 53 classes, which is crucial for preventing the model from developing predictive bias.

Split	Number of Samples	Distribution (Images per Class)
Training	5300	100
Validation	265	5
Test	265	5

Table 1: Dataset Split Summary

4 Data Preprocessing and Augmentation

Effective data preparation is non-negotiable for achieving stable and generalizable deep learning models.

4.1 Image Preprocessing and Normalization

An initial **data integrity check** was performed. For compatibility with the chosen Transfer Learning models (MobileNetV2 and DenseNet-201), all image samples were subjected to two core preprocessing steps:

1. **Resizing:** All images were uniformly resized to the standard input dimension of $\mathbf{224 \times 224}$ pixels.

2. **Normalization:** Pixel intensity values (0-255) were rescaled to the range [0, 1]. This normalization step is critical for accelerating model convergence during the training phase.

4.2 Robust Data Augmentation

To significantly boost the model's robustness and effectively combat **overfitting**, an extensive set of geometric data augmentation techniques were applied exclusively to the training set. This process synthetically expands the dataset and helps the model learn card features independently of their physical orientation or position. Key augmentations included:

- **Rotation:** Random rotations up to $\mathbf{20}^{\circ}$.

- **Shifting & Shearing:** Horizontal/vertical shifting and shearing to introduce realistic perspective distortions.

- **Flipping:** Random horizontal flipping.

4.3 Data Pipeline Management

The 53 categorical labels were converted using One-Hot Encoding. The final data flow was managed by the **ImageDataGenerator** utility, which handles batch loading and applies augmentation dynamically. This ensures efficient data delivery to the GPU while guaranteeing the model is constantly exposed to diverse, new image variations throughout training.

4.4 Image Filtering Analysis

As part of the exploratory visual inspection, various classic image filters (e.g., Canny edge detection and Gaussian blur) were systematically applied to sample images. This analysis served to:

1. **Feature Confirmation:** Visually confirm the clear detection and quality of the card's crucial features—the pips (suit symbols) and indices (rank numbers)—which are essential for classification.

2. **Complexity Insight:** Provide qualitative insight into the recognition task's complexity regarding texture, edges, and background noise, thereby justifying the adoption of deep learning.

This step helped ensure the core identifying features were robust enough for the subsequent preprocessing and augmentation pipeline.

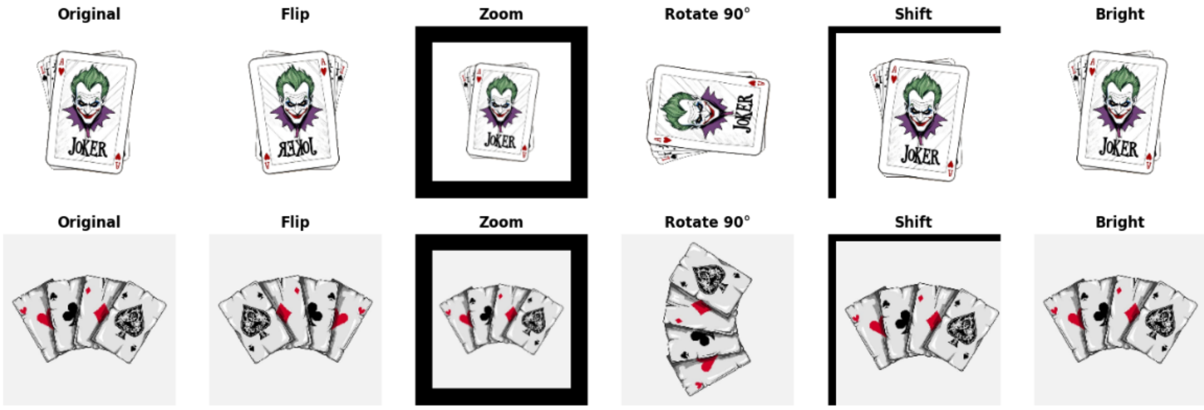

Figure 1: Augmentation Visualization.

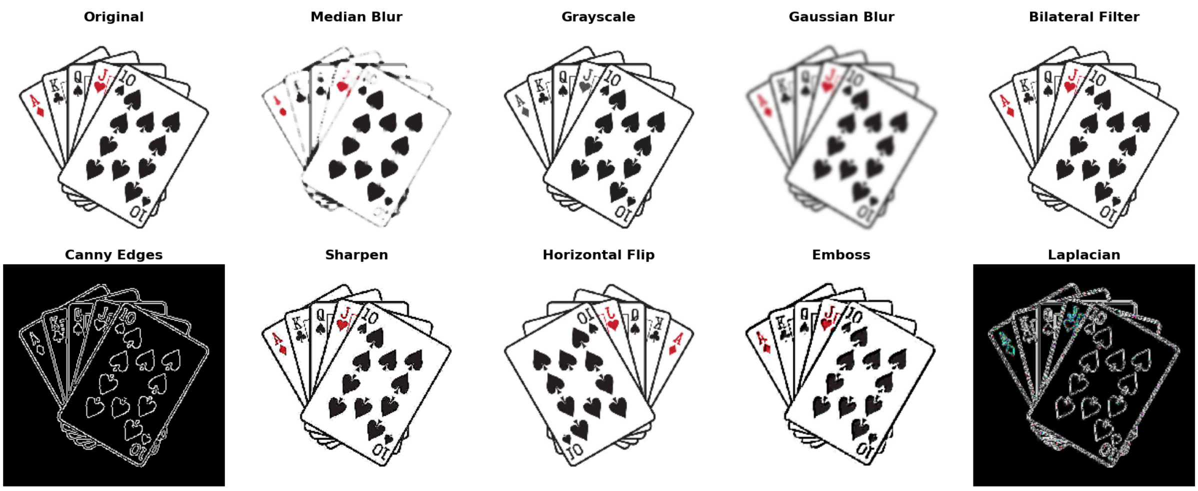

Figure 2: Visualization of various Image Filters.

5 Model Training

The training procedure was centered on the fine-tuning of the pre-trained MobileNetV2 and DenseNet-201 bases.

5.1 Transfer Learning Strategy

Both MobileNetV2 and DenseNet-201 were instantiated using ImageNet pre-trained weights. The training was conducted in two phases: first, the convolutional bases were frozen while only a custom classification head was trained. Second, the top **50** layers of the convolutional base were **unfrozen** for a delicate phase of fine-tuning, allowing the network to adapt generic features to card specifics. The custom head consisted of Global Average Pooling, a Dense layer (ReLU), a **Dropout** layer, and the final **53-unit Dense layer** (Softmax).

5.2 Model-Specific Training Parameters

Table 2: Model-Specific Training Parameters

Model	**Unfrozen Layers**	**Head Size**	**Dropout Rate**	**Epochs**	**Optimizer (LR)**
MobileNetV2	Top 50	128 units	50%	25	Adam (10^{-4})
DenseNet-201	Top 50	256 units	50%	20	Adam (10^{-4})

Table 3: Model-Specific Training Parameters

6 Experimental Results

The models were rigorously evaluated based on convergence speed, generalization ability, and final test set performance on the held-out test data. This section summarizes the quantitative findings and comparative analysis of the two transfer learning architectures.

6.1 Comparative Performance Summary

The final performance metrics after fine-tuning both models for their respective maximum epochs are detailed in Table 4. The results clearly establish MobileNetV2 as the superior model for this specific classification task, achieving a final test accuracy exceeding **85**%.

Model	**Train Accuracy**	**Validation Accuracy**	**Test Accuracy**
MobileNetV2	92.13%	86.04%	85.28%
DenseNet-201	85.66%	80.00%	73.96%

Table 4: Comparative Performance of Deep Learning Models

6.2 Convergence, Stability, and Generalization Analysis

Analysis of the accuracy and loss curves throughout the training process revealed critical differences in how the two architectures converged.

- **MobileNetV2 Stability:** The MobileNetV2 curves showed a tighter coupling between the training and validation metrics. The low divergence indicates a stable learning process where the model's performance on unseen validation data closely tracked its performance on the training data. This demonstrates superior generalization power and a successful application of regularization techniques like Dropout.

- **DenseNet-201 Overfitting:** Conversely, the DenseNet-201 curves displayed a noticeable divergence, with training accuracy significantly higher (85.66%) than its final test accuracy (73.96%). This large gap confirms a greater tendency towards overfitting despite the use of regularization. The model's complex, dense architecture appears to have captured high-variance noise specific to the training set rather than robust, generalizable card features.

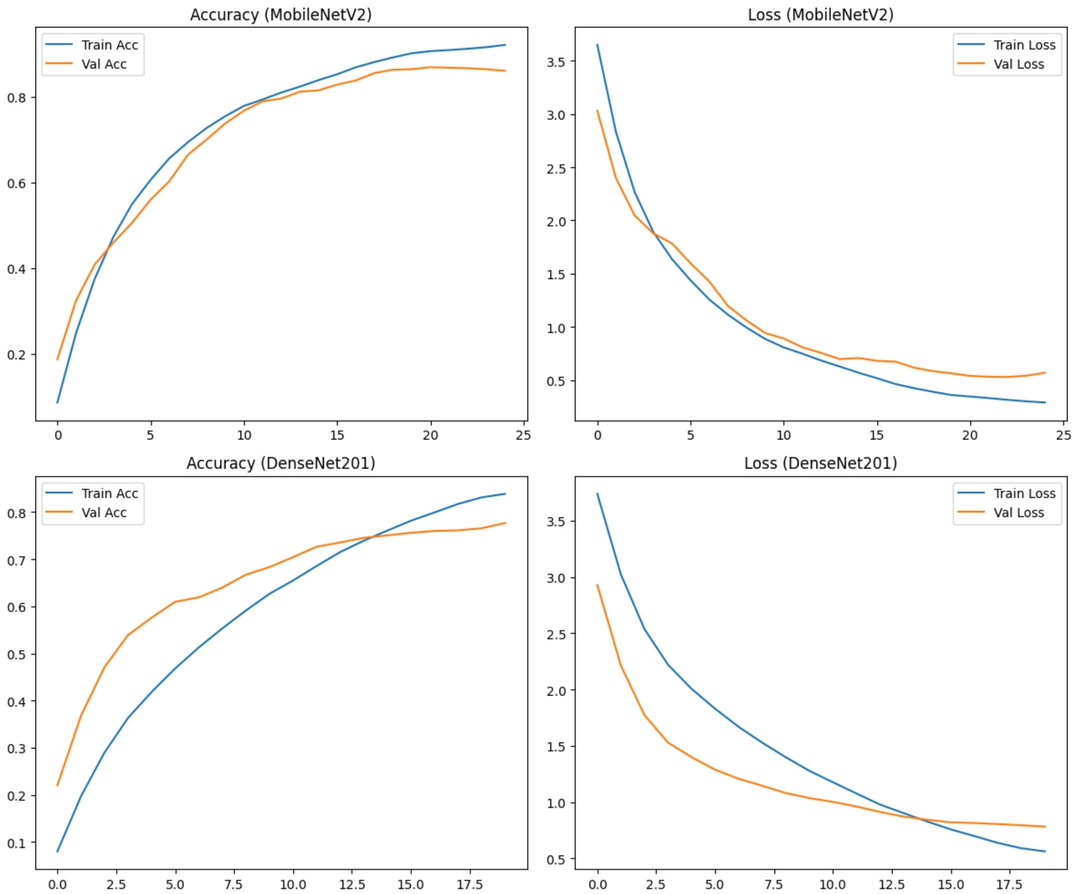

Figure 3: Accuracy and Loss Curves for MobileNetV2 and DenseNet-201.

6.3 Prediction on External Images

Qualitative testing was performed by feeding **unseen external card images** into both trained models to evaluate real-world prediction accuracy and robustness. This test is crucial as it confirms the model's ability to generalize to data outside the original dataset distribution. The models demonstrated successful and accurate classification on all tested external samples.

The images were preprocessed using the following steps before prediction:

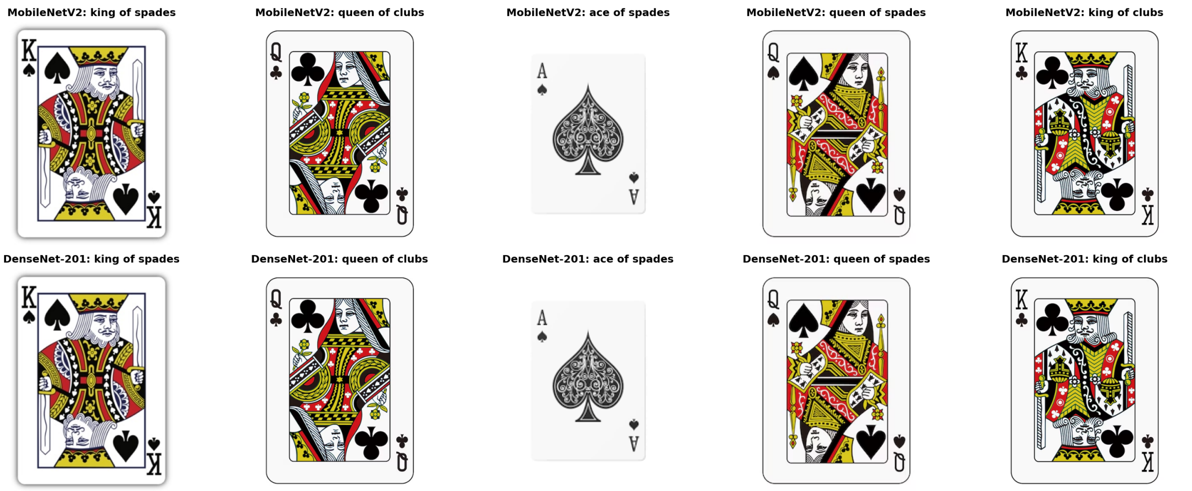

Figure 4: Dual Model Prediction on External Images. MobileNetV2 and DenseNet-201

6.4 Detailed Evaluation Metrics

The final evaluation on the held-out test set cemented MobileNetV2's superiority. The **Confusion Matrix** for MobileNetV2 exhibited a very clean, strong diagonal, signifying high true positive rates and few inter-class misclassifications.

Metric	**MobileNetV2**	**DenseNet-201**
Test Accuracy	**85.28%**	73.96%
Generalization	Excellent	Good (with high overfitting risk)

Table 5: Key Evaluation Metrics Comparison

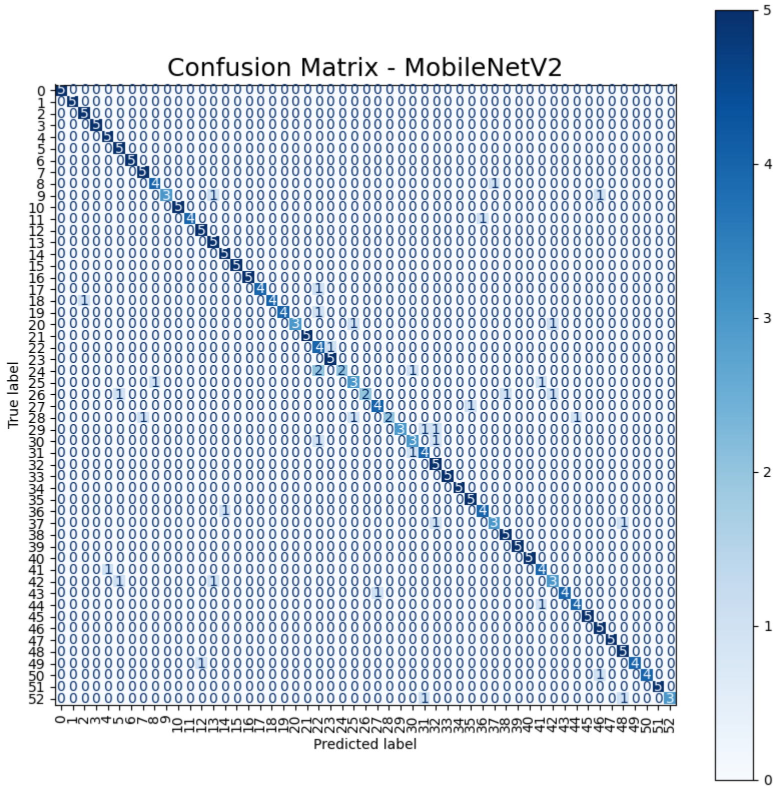

Figure 5: Confusion Matrix for MobileNetV2.

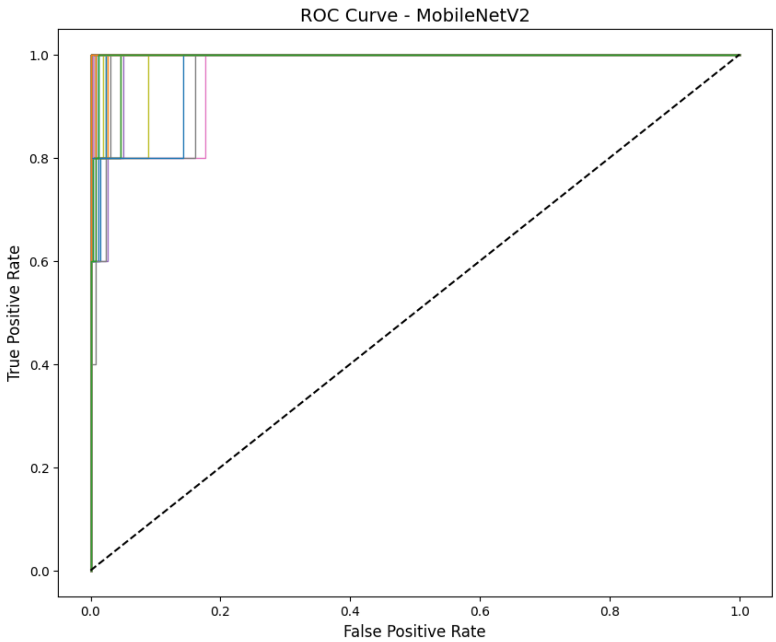

Figure 6: Multi-class ROC Curve for MobileNetV2.

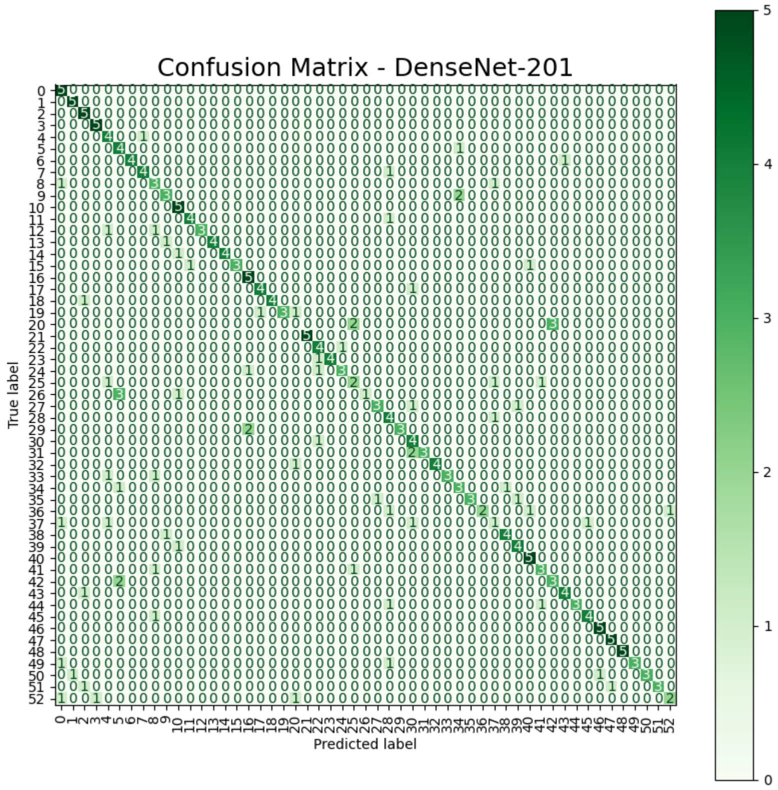

Figure 7: Confusion Matrix for DenseNet-201.

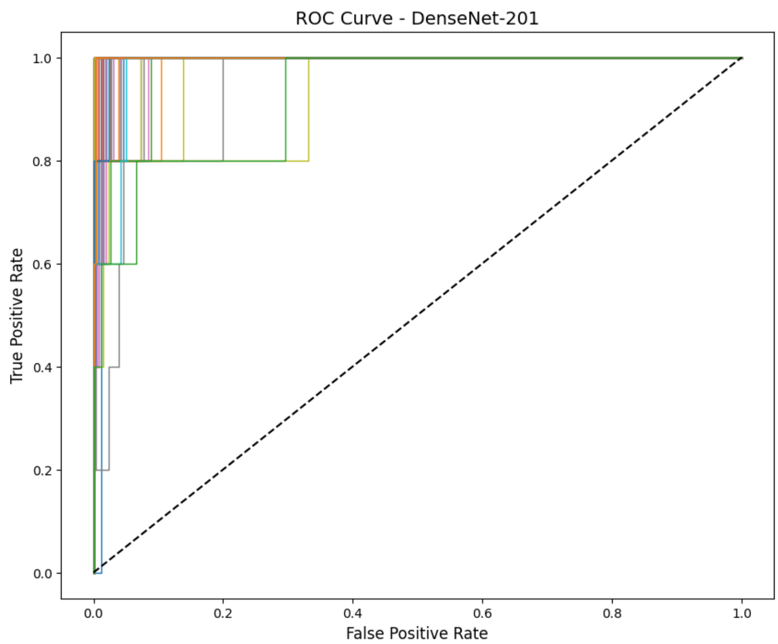

Figure 8: Multi-class ROC Curve for DenseNet-201.

S

7 Discussion

The project successfully achieved its goal, with the fine-tuned MobileNetV2 model delivering a high test accuracy of **85.28%**. The results unequivocally favor the MobileNetV2 architecture for this specific task.

The central finding is the significant underperformance of the more complex DenseNet-201 (Test Acc: 73.96%) compared to the lightweight MobileNetV2. This outcome is highly instructive: for visual classification tasks involving relatively clear, uniform objects like playing cards, the immense feature capacity and depth of DenseNet-201 are not only unnecessary but become detrimental. The model attempts to learn overly specific, high-variance details (noise) present only in the training images, leading to poor performance on unseen data (overfitting).

The **MobileNetV2** proved that an architecture focused on efficiency and streamlined information flow is perfectly sufficient for achieving state-of-the-art results when combined with robust data augmentation and fine-tuning. This efficiency makes MobileNetV2 the most practical choice, offering an optimal balance between accuracy and the low latency required for real-time applications. The successful classification of external, unseen images further validates the model's robust generalization capabilities.

8 Future Work

To build upon these successful results and move toward a deployable system, the following research and development steps are recommended:

1. **Ensemble Modeling:** Implement a soft-voting **ensemble** that combines the predictions of MobileNetV2 with other top-performing lightweight models (e.g., EfficientNet). This strategy often provides a marginal but critical boost in overall accuracy and stability.

2. **Advanced Augmentation for Robustness:** Further enhance the model's resilience by introducing more aggressive data augmentation simulating crucial real-world failures, such as high-degree **partial occlusion** (e.g., one card partially covering another) and severe non-uniform lighting effects.

3. **Real-Time Optimization:** Utilize optimization frameworks like TensorFlow Lite or OpenVINO to convert the superior MobileNetV2 model into an optimized format for low-latency inference on mobile or edge devices, enabling a true real-time application.

4. **Integrated Detection and Classification:** For complex, cluttered environments (e.g., a card game table), integrate a preliminary **Object Detection** module (such as YOLO or Faster R-CNN) to first locate and crop the card, ensuring the classifier only processes the target object.

9 References

[R1] Howard, A. G., et al. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.* arXiv preprint arXiv:1704.04861.

[R2] Huang, G., et al. (2017). *Densely Connected Convolutional Networks.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[R3] Sandler, M., et al. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[R4] Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition.* International Conference on Learning Representations (ICLR).

[R5] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks.* Advances in neural information processing systems, 25.

[R6] *Cards Image Dataset - Classification 6k.* (https://www.kaggle.com/datasets/gpiosenka/cards-image-datasetclassification).