FINAL PROJECT SYNOPSIS

CLASSIFICATION OF REAL AND AI-GENERATED SYNTHETIC IMAGES

MSDSP462 ComputerVision

Group 10

Abhinay Kumar Singh | Anubhav Bhatnagar | Vivek Mishra

March 2025 | Northwestern University MSDS

Introduction

The objective of this project is to develop a robust model capable of distinguishing between real and AI-generated (fake) images. Leveraging a comprehensive Kaggle dataset with 60,000 real images and 60,000 fake images, we trained and evaluated multiple models to achieve this goal.

Model Selection, Observations and Fine-Tuning

Our exploratory data analysis (EDA) revealed no significant features that could directly aid in classification. Therefore, we selected four models for evaluation: a custom Convolutional Neural Network (CNN) and three pre-trained models — EfficientNetV2, ResNet50, and MobileNetV3Small. Given the binary classification nature of the problem, we employed the Sigmoid activation function for the top layer and ReLU for the inner layers. For the CNN model, we implemented three convolutional layers with 64, 128, and 256 filters, accompanied by dropout and max-pooling layers. The outer layer comprised three dense layers. This architecture yielded promising results, with an accuracy of approximately 93% and an AUC of 0.93, positioning CNN as a strong contender.

In our initial approach with pre-trained models, we replaced only the top layer with a custom dense classifier while keeping the inner layers frozen. However, this approach resulted in poor accuracy and precision (~50%) for both training and test data.

To address this, we attempted to replace the classifier with a Random Forest model. While this improved the accuracy to over 90% on the training data, the test accuracy remained stagnant at around 50%, indicating overfitting. Recognizing that modifying only the top layer was insufficient, we opted to make the last 20 layers of the pre-trained models trainable. This adjustment significantly enhanced performance, with all models achieving accuracy above 90% and AUC above 0.93 as well.

Results and Evaluation

After experimenting with 10 model variations (one CNN and three variations for each pre-trained model), we summarized the key performance metrics in the table below. While additional parameters could further assess model efficiency, this concise summary effectively captures the results.

The pre-trained models, when fine-tuned for the last 20 layers, outperformed all others. EfficientNetV2 emerged as the top performer, boasting an accuracy of 97% and an AUC of 0.993. However, it comes with the drawback of a large file size and extended training time.

MobileNetV3Small closely followed, achieving an accuracy of 97.29% and an AUC of 0.98. Notably, its lightweight model file (.keras) is ideal for deployment on mobile devices, where memory constraints and user accessibility are critical factors.

In conclusion, while EfficientNetV2 excels in accuracy, MobileNetV3Small offers an optimal balance between performance and practicality for mobile applications.

Model	Model Variation	TrainData Accuracy	TestData Accuracy	AUC	.keras File Size(MB)
CNN					
	3 Convolution layer	94%	93%	0.93	4.00
RestNet50					
	Custom Classifier	49.74%	50.49%	0.501	283
	Random Forest Classifier	99.54%	50.06%	0.5	267
	Inner 20 Layer Training	98.69%	92%	0.97	158
EfficientNetV2					
	Custom Classifier	49.79%	50.00%	0.4987	74
	Random Forest Classifier	99.57%	50.04%	0.5	80
	Inner 20 Layer Training	99.31%	97%	0.993	242
MobileNetV3small					
	Custom Classifier	49.94%	50.43%	0.503	13
	Random Forest Classifier	99.43%	50.23%	0.502	14
	Inner 20 Layer Training	97.29%	94%	0.98	13

Detailed results and performance can be seen in the jupyter note books located at: https://github.com/vvkmshr-nwu/462-ComputerVision/blob/main/README.md

Conclusion

Considering a balance between accuracy and efficiency as resource requirements, MobileNetV3Small is clear winner.