UCCD3074 Deep Learning for Data Science
Group Assignment
Classification of AI-Generated and Real Images using Deep Learning Techniques

Research-based □ Application-based ■

Name	Brandon Ting En Junn (Leader)	Chin Wai Yee	Loh Kin Ming
Programme	CS	CS	CS
ID	2101751	2103370	2102691
Contribution	1/3	1/3	1/3

1. INTRODUCTION

Done By: Loh Kin Ming 2102691

The advancements in Generative Adversarial Networks (GANs) and diffusion models have led to synthetic images becoming more photorealistic, which raises concerns regarding digital forgery, identity theft and the spread of false information. The main objective behind this project is to develop an Artificial Intelligence (AI) system with the ability to differentiate between real and AI-generated images.

Detection of AI-generated content is crucial to retain digital media integrity which makes this problem a matter of utmost importance. Present detection systems currently use artifacts together with metadata yet prove inefficient when handling images from various types and generation methods.

Modern systems also face difficulties in differentiating between minor differences between real and AI-generated images, handle fast-changing generative methods, and perform on different image sources. The main goal of this work is to develop a binary classifier that uses a curated dataset while emphasising on high accuracy, robustness and identifying key distinguishing features.

2. RELATED WORK

Done By: Loh Kin Ming 2102691

AI-generated image identification has become fundamental due to the developing generative models. Multiple research studies provide extensive information about detection strategies and difficulties in this field.

Mahara and Rishe [1] conducted an extensive review of methods that identify images generated by GANs and Diffusion Models and Variational Autoencoders (VAEs). Detection methods fall into three categories, namely artifact-based, model-based, and multimodal approaches according to the authors who stress the need for extensive training datasets for successful operations. The study points out that the rapid evolution of generative models remains ahead of detection method development requiring consistent development of adaptable and robust detection frameworks.

The research by Liu et al. [2] studies how deepfake detection strategies move from single-modal systems to multi-modal systems in facial forgery applications. The authors emphasize through their research that single-modal detection strategies based on visual artifact analysis are insufficient while promoting multi-modality detection techniques combining text-visual and audio-visual components for better recognition results. Research presents the potential to have the usage of modality correlation plus with transformer-based models for the aim of enhancing detection system reliability as well as preserving hostility.

Done By: Chin Wai Yee 2103370

From recent studies, [3] had trained GAN-generated images classifier by doing fine-tuning on pre-trained model. They first collected real images and generated AI-Generated image using different tasks such as image-to-image synthesis, sketch-to-image synthesis, and text-to-image synthesis. By doing so, they prepared 24,000, 12,000, and 12,000 images for training, validation, and testing respectively. To speed up the training process, they performed fine-tuning on 11 pre-trained models including Visual Geometry Group (VGG) 19, ResNet-50 and EfficientNetB4. For each training process, they used a batch size of 64, initial learning rate of 0.001 and 20 epochs. As a result, they successfully trained an EfficientNetB4 model with 100% accuracy. However, the study also pointed out some limitations of the model such as failing in classifying images when background or foreground is blurry, or when image is of low quality.

Furthermore, study in [4] demonstrated a reliable AI-generated image detection method using either Photo Response Non-Uniformity (PRNU) or Error Level Analysis (ELA) preprocessing combined with Convolutional Neural Network (CNN) classification. While both techniques were effective, ELA provided slightly better and more stable results. The system was designed to work with 512×512 image patches and could be adapted for smaller sizes with retraining.

Done By: Brandon Ting En Junn 2101751

Muthaiah et al. [5] conducted an extensive study on reviewing classification techniques for AI-generated and real images. They found that most papers use deep neural networks such as VGG, CNN, ResNet, DenseNet, and Swim Transformer. A comparative analysis was provided for the highlighted models of CNN, ResNet-50, and VGG16 in terms of performance, accuracy, complexity, memory requirements, best use cases, strengths, and weaknesses. ResNet-50 was a balanced architecture in terms of consistent accuracy and computational speed with its residual connections to overcome the vanishing gradient problem.

Jain and Kundra [6] proposed their approach on classifying deepfakes. Their approach involved the EfficientNet-based deep learning network that is a highly accurate and efficient classifier by being scalable and computationally efficient. Specifically, they proposed the EfficientNetB0-based approach with a total of 190,305 images as their dataset. The "Real" class contained 95,213 images, while the "Fake" class contained 95,092 images. Their proposed model achieved an overall accuracy of 82% and a balanced performance between precision, recall, and F1-Scores, with macro and weighted averages of 0.82.

3. SYSTEM DESIGN

Done By: Chin Wai Yee 2103370

Figure 3.1 shows the overall system workflow that includes data acquisition, image preprocessing, get pre-trained model, model fine-tuning, evaluation, and saving results for future use.

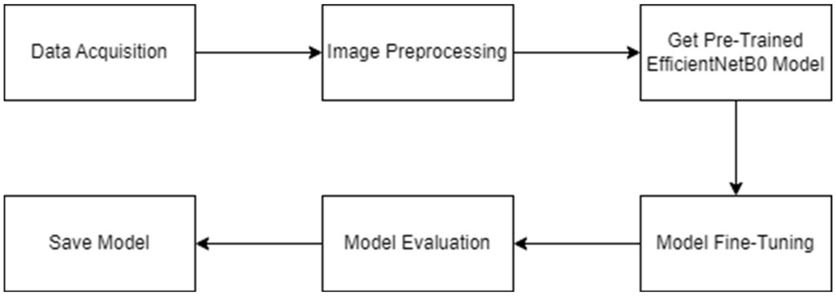


Figure 3.1: System Workflow

Data Acquisition

The dataset was obtained from the Kaggle dataset [7] which included 30,000 AI-generated images and 30,000 real images.

Image Preprocessing

The images had been split into 3, which were 72.25% training set, 12.75% validation set and 15% test set. The training set had been transformed and augmented by resizing to size of (224, 224), random horizontal flip, slight rotation, color jittering, and affine transformations. These augmentations helped the model become robust to minor variations and prevent overfitting. The images were then converted into tensors and normalised to a common scale. For validation and testing, the transformation only involves resizing and normalisation only, ensuring consistent evaluation without introducing randomness. The parameters for data preprocessing were summarized in Table 3.1 and Table 3.2.

Table 3.1: Parameter Settings for Training Transformation

Resize	(244, 244)
Random Horizontal Flip	-
Random Rotation	Degrees=10

Color Jitter	Brightness = 0.2,
	Contrast = 0.2,
	Saturation = 0.2

Random Affine	Degrees = 0,
	Translate = (0.05, 0.05)

Normalize	Mean = [0.5, 0.5, 0.5]
	Std = [0.5, 0.5, 0.5]

Table 3.2: Parameter Settings for Testing Transformation

Resize	(244, 244)

Normalize	Mean = [0.5, 0.5, 0.5]
	Std = [0.5, 0.5, 0.5]

Pre-Trained Model

The pre-trained model that was used in this system is EfficientNetB0. The architecture of EfficientNetB0 and MBConv are shown in Figure 3.2 and Figure 3.3 respectively.

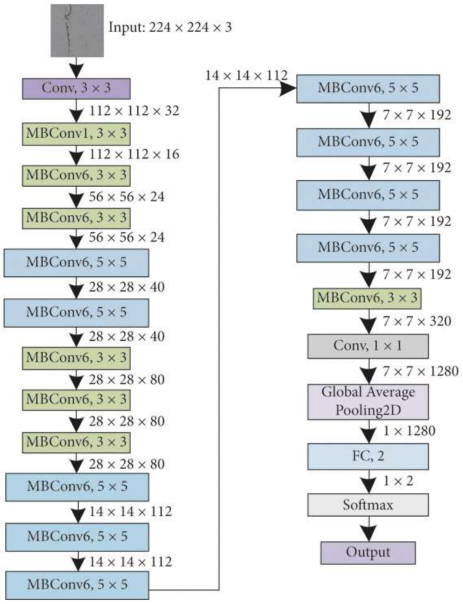


Figure 3.2: EfficientNetB0 Architecture [8]

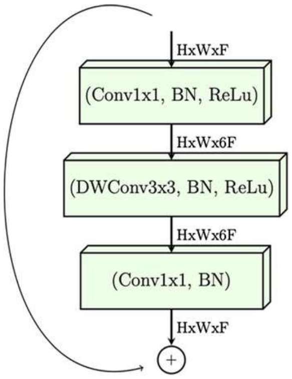


Figure 3.3: MBConv Architecture [9]

EfficientNetB0 [10] is part of the EfficientNet family of CNN developed by Google AI which has 5.3M parameters. It utilises Mobile Inverted Bottleneck Convolution (MBConv) which are efficient convolutional blocks that expand the number of channels, apply depth wise separable convolutions, and then project back to a lower-dimensional space. In this design, the EfficientNet pre-trained model from Pytorch Image Models was used.

Model Fine-Tuning

During the fine-tuning, all layers were trained. The optimiser selected for training was Adam with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5}. The model was trained for 5 epochs. In each epoch, the running loss and accuracy were calculated, and the model was evaluated on the validation set.

Model Evaluation

After completing the training and validation process, the model was evaluated on the test dataset with standard classification performance metrics, including accuracy, precision, recall, F1-score, and Cohen's kappa coefficient. Additionally, the Precision-Recall Curve and an AUC-ROC (Area Under Curve-Receiver Operating Characteristics) Curve were plotted.

Save Model

The fine-tuned model will be saved as .pth file for future reference.

4. EXPERIMENT & EVALUATION

Done By: Brandon Ting En Junn 2101751

This project was based on the dataset "ai-generated-images-vs-real-images" from Kaggle [7]. The dataset consisted of a total of 60,000 images. 30,000 were AI-generated images and 30,000 were real images. For the AI-generated images, 10,000 of Stable Diffusion, 10,000 of MidJourney, and 10,000 of DALL-E images. For the real images, 22,500 of Pexels, Unsplash, and 7,500 of WikiArt images. The dataset was split into train, validation, and test sets with stratified sampling for the experiment. Specifically, the train set consisted of 72.25%, validation set consisted of 12.75%, and test set consisted of 15% out of the total dataset. Data transformation such as image resolution resizing to 224x224 and normalisation were performed to the train, validation, and test datasets. Data augmentation techniques such as random horizontal flip, random rotation at 10 degrees, colour jitter, and random affine were performed on the train set only. The train, validation, and test sets were shuffled and loaded into data loaders with a batch size of 32. The evaluation of the experiment was done on the test set.

This project used the EfficientNetB0 architecture that was initialised with its pre-trained weights to train the classification model. Additionally, hyperparameter fine-tuning was performed for the model to optimise its performance. Hyperparameters such as batch size, shuffling, epoch number, pre-trained weights, layer freezing, loss function, optimiser, learning rate, and weight decay were considered for the model. Table 4.1 shows the summary of the hyperparameter settings for the model.

Table 4.1: Summary of Model Hyperparameter Settings

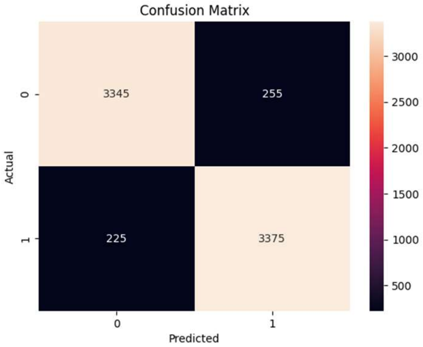
Hyperparameter	**Setting**
Batch size	32
Shuffling	True
Epoch number	5
Pre-trained weights	True
Layer freezing	None
Loss function	Cross entropy loss
Optimiser	Adam
Learning rate	1×10^{-4}
Weight decay	1×10^{-5}

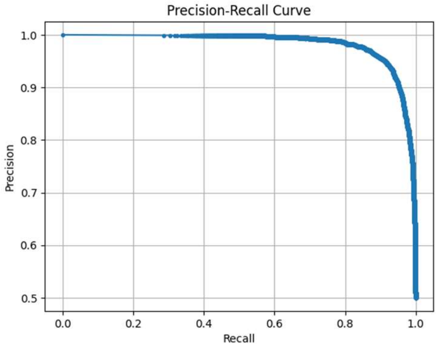
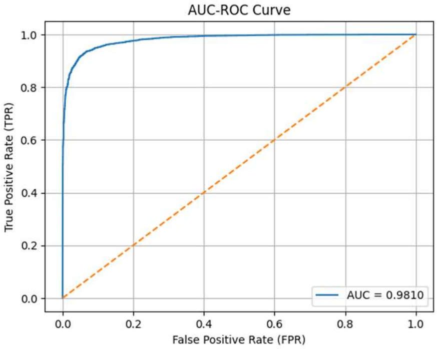
A batch size of 32 provided balance between computational speed and model generalisation. It is a suitable setting for the mid-sized dataset of 60,000 images. Enabling shuffling reduces overfitting of the model due to the random order of AI-generated and real images being trained to avoid memorization of data order. The epoch number of 5 is appropriate due to the mid-sized dataset combined with using previously well-established pre-trained weights of EfficientNetB0. All the layers were not frozen, allowing the weights of all layers to be trained for the classification task. This was effective only due to the sufficient size of the dataset. The Adam optimiser that was used provided an adaptive gradient descent optimiser. The learning rate and weight decay were set to 1×10^{-4} and 1×10^{-5} respectively. A small learning rate allowed for good adjustments of the weights without overwriting useful extracted features, while a small weight decay provided sufficient regularisation to prevent overfitting by penalising large weights.

An important note for this project is that AI-generated images were labeled as "0" (positive cases) while real images were labeled as "1" (negative cases).

Referring to the Zero Rule (ZeroR) assumption on the dataset, the baseline performance of the model was 0.5 (50%), as the expected performance benchmark. Performance metrics such as confusion matrix, accuracy, precision, recall, specificity, F1-score, precision-recall curve, AUC-ROC curve, and kappa coefficient were considered to evaluate the performance of the model on the test set. Table 4.2 shows the summary of the experiment results.

Table 4.2: Summary of Experiment Results

Performance Metric	**Experiment Result**	**Description**
Confusion matrix	 Figure 4.1: Confusion Matrix	True Positive (TP), predicted positive and actual positive. = 3,345 True Negative (TN), predicted negative and actual negative. = 3,375 False Positive (FP), predicted positive but actual negative. = 225 False Negative (FN), predicted negative but actual positive = 225
Accuracy	93.33%	High accuracy indicated a high number of correct predictions.

Precision	93.70%	High precision indicated a high number of correct predictions on positive cases and low FP errors.
Recall	92.92%	High recall indicated the high ability to capture most of the positive cases.
Specificity	$\frac{TN}{TN+FP} = 93.75\%$	High specificity indicated the high ability to capture most of the negative cases.
F1-score	0.9331	The F1-score was very close to 1, indicated that precision and recall were well-balanced. Correct predictions on both positive and negative cases were consistently high.
Precision-recall curve	 Figure 4.2: Precision-Recall Curve	The visualised curve was very close to the top right corner, indicated that high precision and high recall on each prediction.
AUC-ROC curve	 Figure 4.3: AUC-ROC Curve	The visualised area (0.9810) was very close to 1, indicated that a nearly perfect model performance.
Kappa coefficient	0.8667	The kappa coefficient was close to 1, indicated that good confidence in predictions rather than random guessing.

An ablation study on layer freezing was conducted for this experiment. All of the layers in the network were frozen except the final classification layer. This evaluated the impact of layer freezing on the performance of the models. Figure 4.4 shows a comparison between Model A and Model B in terms of their test accuracy. Model A had all layers unfrozen, while Model B only had the final classification layer unfrozen. Both of the models had exactly the same hyperparameter settings except for layer freezing.

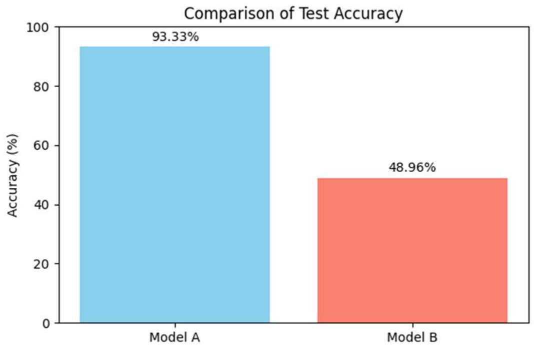


Figure 4.4: Comparison of Test Accuracy

Model B performed significantly worse compared to Model A, specifically test accuracy of 48.96% to 93.33%, given the same model hyperparameter settings. This proved that given a sufficient mid-sized dataset, it is better to unfreeze all the layers in the network rather than only the final classification layer for training and extracting new features for the classification task.

A comparative analysis was conducted to compare the performance of the proposed model with other selected state-of-the-art models adopted from [5]. Table 4.3 shows a comparison table of each model in terms of accuracy.

Table 4.3: Accuracy Comparison Table between Models

Author Name	**Dataset Used**	**Model Used**	**Accuracy**
Brandon Ting En Junn et al.	ai-generated-images-vs-real-images	EfficientNetB0	93.33%
Jordan J. Bird and Ahmad Lotfi	Synthetic CIFAR-10	CNN (InceptionV3 + ANN)	92.98%
Isabella Castiglioni et al.	Medical Images (The Cancer Imaging Archive - TCIA)	CNN	92-95%
Mingjian Zhu et al.	GenImage Dataset	ResNet-50, Swin-T	99.9% (ideal conditions), 54.9% (cross-generator)
Anug Badale et al.	900 Deepfake Videos	CNN + Dense Neural Networks	91% (Adam optimizer)
Riccardo Corvi et al.	COCO, ImageNet, UCID	ResNet50-based detectors	99.9% (ProGAN), 50% (diffusion models)
Md Shohel Rana et al.	FaceForensics++	CNN, RNN	99.65%
Yan Ju et al.	Custom AI-Synthesized Dataset (128,000 images)	Patch Selection Module (PSM) + CNN	76.25%
Saadaldeen Rashid Ahmed et al.	140k Real and Fake Faces	DenseNet, VGG16, VGG19	97%
Bojia Zi et al.	WildDeepfake	CNN	76.25%

Based on Table 4.3, the proposed model managed to outperform 4 out of 9 of the other models based on accuracy. This shows that the proposed model is a potential competitor for the classification task of AI-generated and real images.

5. CONCLUSION

Done By: Loh Kin Ming 2102691

This project has successfully demonstrated the use of EfficientNetB0 to classify AI-generated images and real images using the dataset from [7]. Initial experiments with frozen layers except the final classification layer produced a test accuracy of 48.96%, but with all layers unfrozen the EfficientNetB0 model was able to reach an accuracy of 93.33%.

EfficientNetB0's ability to balance depth, width and resolution led to better feature extraction, reduced overfitting and faster convergence. The results suggest that with further training on larger and better datasets, EfficientNet can be on par or surpass older models such as CNN, Recurrent Neural Network (RNN) or ResNet.

REFERENCES

[1] A. Mahara and N. Rishe, "Methods and Trends in Detecting Generated Images: A Comprehensive Review," arXiv preprint arXiv:2502.15176, Feb. 2025.

[2] P. Liu, Q. Tao, and J. T. Zhou, "Evolving from Single-modal to Multi-modal Facial Deepfake Detection: A Survey," arXiv preprint arXiv:2406.06965, Jun. 2024.

[3] S. S. Baraheem and T. V. Nguyen, “AI vs. AI: Can AI Detect AI-Generated Images?,” Journal of Imaging 2023, Vol. 9, Page 199, vol. 9, no. 10, p. 199, Sep. 2023, doi: 10.3390/JIMAGING9100199.

[4] F. Martin-Rodriguez, R. Garcia-Mojon, and M. Fernandez-Barciela, “Detection of AI-Created Images Using Pixel-Wise Feature Extraction and Convolutional Neural Networks,” Sensors (Basel), vol. 23, no. 22, p. 9037, Nov. 2023, doi: 10.3390/S23229037.

[5] Muthaiah U, A. Divya, T.N. Swarnalaxmi, and Vidhyasagar BS, “A Comparative Review of AI-Generated vs Real Images and Classification Techniques,” pp. 141–147, Dec. 2024, doi: https://doi.org/10.1109/icuis64676.2024.10866220.

[6] E. Jain and D. Kundra, “EfficientNet-Based Deepfake Detection: A Robust Approach for Real and Fake Media Classification,” 2024 Global Conference on Communications and Information Technologies (GCCIT), pp. 1–6, Oct. 2024, doi: https://doi.org/10.1109/gccit63234.2024.10862025.

[7] Z. Tristan, “ai-generated-images-vs-real-images.” Accessed: May 03, 2025. [Online]. Available: https://www.kaggle.com/datasets/tristanzhang32/ai-generated-images-vs-real-images/data

[8] C. Su and W. Wang, “Concrete Cracks Detection Using Convolutional Neural Network Based on Transfer Learning,” Math Probl Eng, vol. 2020, 2020, doi: 10.1155/2020/7240129.

[9] E. Luz et al., “Towards an effective and efficient deep learning model for COVID-19 patterns detection in X-ray images,” Research on Biomedical Engineering, vol. 38, no. 1, pp. 149–162, Mar. 2022, doi: 10.1007/S42600-021-00151-6.

[10] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” 36th International Conference on Machine Learning, ICML 2019, vol. 2019-June, pp. 10691–10700, May 2019, Accessed: May 03, 2025. [Online]. Available: https://arxiv.org/pdf/1905.11946