Journal of Advanced Artificial Intelligence, Engineering and Technology

ISSN: 3139-4256

Research Article

WRJAAIET-25-007

Classifying Real and AI-Generated Images Using Fine-Tuned ResNet50

Khan Aabid Abdal Mashkoor*

Department of Engineering and Technology, BK Birla College, Kalyan, India

***Correspondence:** Khan Aabid Abdal Mashkoor, Department of Engineering and Technology, BK Birla College, Kalyan, India, E-mail: khanaabidabdal@gmail.com; DOI: https://doi.org/10.56147/aaiet.1.2.7

Citation: Mashkoor KAA (2025) Classifying Real and AI-Generated Images Using Fine-Tuned ResNet50. J Adv Arti Int Eng & Techn 1: 07.

Abstract

Due to advancements in technology, especially in Artificial Intelligence, it has become increasingly difficult for humans to distinguish between real and AI-generated images. Technologies like Generative Adversarial Networks (GANs) and Latent Diffusion Models (LDMs) can generate images so realistic that humans often cannot identify them as AI-generated. This paper addresses the challenge of identifying AI-generated and real images by presenting a Fine-Tuned ResNet50 model. The model is trained and tested on a dataset of 140K real and fake face images, demonstrating its effectiveness in image classification.

Keywords: Ai-generated images; Fake face image classification; Transfer learning; Deep learning; ResNet50

Received date: April 09, 2025; **Accepted date:** April 15, 2025; **Published date:** April 30, 2025

Introduction

In today's world, Artificial Intelligence (AI) is just a click away, significantly impacting human lives. However, it has also introduced modern challenges, particularly the misuse of Generative AI. Many models are now capable of generating images of human beings that are so realistic they could win competitions against real images.

Generative Adversarial Networks (GANs) are neural networks designed for image generation [1]. GANs consist of two Artificial Neural Networks (ANNs): The Generator and the discriminator. The generator creates images, attempting to fool the discriminator while the Discriminator tries to distinguish between images from the dataset and those generated by the generator. This competitive process results in highly realistic images. Deepfakes, which are fake images or videos generated using deep learning technology, can be created by either replacing a person's face with another's or using GANs.

AI-generated images have numerous benefits, such as in advertising, chemical industries, material science, education and astronomy. For instance, since it is impossible to capture an image of our own Milky Way galaxy from within, a generative AI model can create such an image based on features like the galaxy's diameter and the approximate number of stars. These applications demonstrate the advantages of GANs. However, like any technology, generative AI also has its drawbacks. For example, while nuclear energy can be used for power plants, it also poses the threat of nuclear bombs.

The misuse of generative AI is a significant concern, given its easy and free accessibility. AI-generated images can be used to conceal identities on social media, facilitating scams. These images can deceive facial recognition systems and AI-generated audio and video can be used for blackmail. Deepfakes can create false criminal evidence against innocent people [2].

This paper proposes a solution for classifying AI-generated and real images using a fine-tuned ResNet50 model [3].

Datasets

The dataset used for this research is the 140K real and fake faces dataset, consisting of 140,000 images of human faces, evenly split between 70,000 real faces and 70,000

Journal of Advanced Artificial Intelligence, Engineering and Technology

ISSN: 3139-4256

fake faces generated by Style GANs [4]. The dataset is divided into training, validation and test sets. The training set comprises 50,000 images of each category, while both the validation and test sets contain 20,000 images of each category. The data can be found at Kaggle [4].

The 70,000 real face images are sourced from the Flickr dataset, collected by Nvidia [5]. In contrast, the 70,000 fake face images are sampled from 1 million fake faces generated by Style GANs [6]. All images are resized to 256 pixels for consistency. Below are some examples from the dataset:

Figure 1: Real and fake face images.

Methods

Transfer learning: ResNet50

Transfer learning is a technique in deep learning where pre-trained models are used to solve new but similar problems. There are two ways to utilize transfer learning:

Fine-tuning: This involves not freezing the weights of the pre-trained model and using its structure to solve a new problem. This method requires more time and computational power as the entire model undergoes backpropagation. Sometimes some layers can be freeze while some are left unfreeze.

Feature extraction: In this method, some weights of the pre-trained model are frozen and only a few layers are adjusted to suit the new problem. This approach is faster and requires less computational power because the frozen layers do not update their weights.

For this paper, the ResNet50 model was used. ResNet50 or Residual Network, is a pre-trained deep learning model that uses residual connections to build deep networks and overcome the gradient vanishing problem. Gradient vanishing occurs when gradients become so small during backpropagation that weight updates are negligible. Residual connections help by providing the input from a previous block to the next block, ensuring the gradient remains considerable. Sometimes, these residual connections include convolutional layers for shape adjustment.

Experiment 1: Feature extraction

In the first experiment, minimal changes were made to the ResNet50 model. The fully connected (fc) layer was adjusted by changing the output class of the linear layer from 1000 to the length of my output classes (*i.e.,* 2). The structure of ResNet50 is given below:

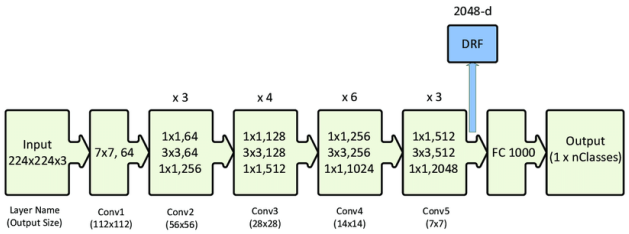

Figure 2: Feature extraction.

The code for feature extraction is:

```
model.fc=torch.nn.Sequential(torch.nn.Linear(2048,2).
```

Experiment 2: Fine-tuning

In the second experiment, additional changes were made to improve accuracy. A combination of Linear layers, ReLU() functions and Dropout layers was used. The fully connected (fc) layer of ResNet50 was modified as follows:

```
model.fc=torch.nn.Sequential(torch.nn.Linear(2048,
1000), torch.nn.ReLU(), torch.nn.Linear(1000, 500),
torch.nn.Dropout(), torch.nn.Linear(500, 100),
torch.nn.ReLU(), torch.nn.Dropout(), torch.nn.Linear(100,
2) #Number of classes)
```

Training setup

Since there were already 50,000 images for each class, no data augmentation techniques were used. The images were already resized to 256 pixels. Feature extractor model was trained for 15 epochs with a learning rate of 0.01, using the Adam optimizer. The batch size was set to 32. Trained the Fine-tuned Model with 5,10 and 15 epochs with same set up.

Results and Discussion

The best way to analyze a model's performance is to plot the graphs of loss and accuracy. Below, we discuss the results from the experiments conducted.

Experiment 1: Feature extraction model

Figure 3 shows the training loss, test loss, training accuracy and test accuracy for experiment 1. Both training

Journal of Advanced Artificial Intelligence, Engineering and Technology

ISSN: 3139-4256

and test accuracy increase with epochs, reaching up to 89%. Similarly, training and test loss decrease. However, after the 12th epoch, the model starts to overfit.

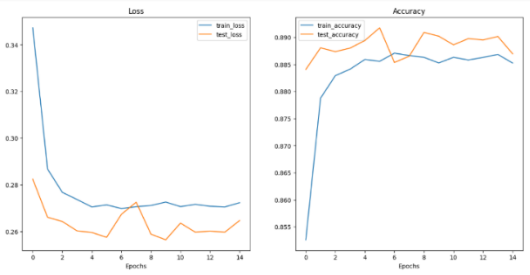

Figure 3: Feature extraction model.

Experiment 2: Fine-tuning

Figure 4 depicts the results for experiment 2 with 5 epochs. The training loss decreases consistently, while the test loss initially increases up to the second epoch before it also starts decreasing parallel to the training loss. Training accuracy increases steadily, while the test accuracy initially decreases. After the first epoch, the test accuracy begins to increase, reaching 98%. However, when the same experiment was repeated with the same setup, the accuracy did not exceed 94%.

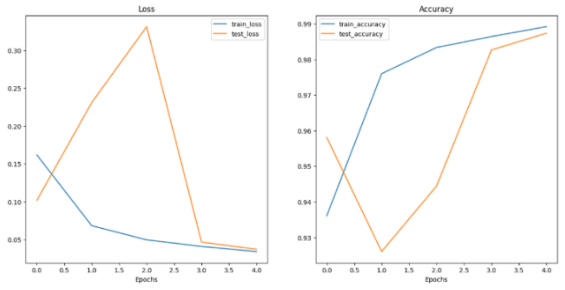

Figure 4: Experiment 2 with 5 epochs.

Figure 5 presents the results for experiment 2 with 10 epochs. Both training and test loss decrease, with training loss decreasing smoothly and test loss decreasing in a zigzag manner. The accuracy curves show both training and test accuracy increasing, with test accuracy reaching 95% and training accuracy around 97%.

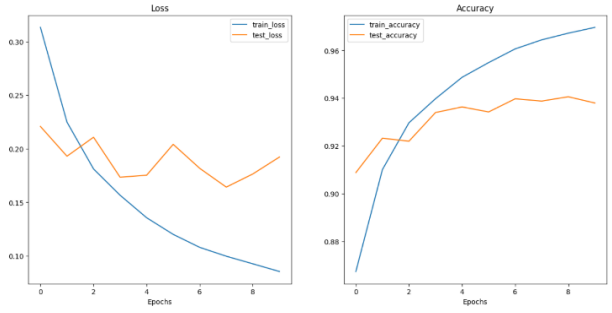

Figure 5: Results for experiment 2 with 10 epochs.

Figure 6 shows the results for experiment 2 with 15 epochs. Both training and test loss continue to decrease, though there is a noticeable difference between the two. Training and test accuracy increase, reaching a maximum value of 96%, but then trend in a zigzag manner.

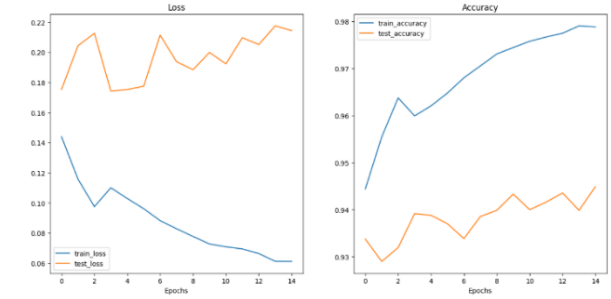

Figure 6: Experiment 2 with 15 epochs.

Performance metrics

Accuracy is defined as the number of correctly classified outputs divided by the total number of classes. The performance metrics used in this study include accuracy, precision, recall and F1 score, which are defined as follows:

True Positive (TP): The number of correctly predicted positive classes.

False Positive (FP): The number of incorrectly predicted positive classes.

True Negative (TN): The number of correctly predicted negative classes.

False Negative (FN): The number of incorrectly predicted negative classes

The formulas for these metrics are:

Accuracy

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall

$$\text{Recall} = \frac{TP}{TP+FP}$$

F1 score

$$\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The accuracy, precision, recall and F1 score for all models are summarized in below **table 1**:

Journal of Advanced Artificial Intelligence, Engineering and Technology

ISSN: 3139-4256

Table 1: Accuracy, precision, recall and F1 score for all models

Model	Accuracy	Precision	Recall	F1 Score
Feature extraction model	0.8997	0.9172	0.8505	0.8505
Fine-tuned model (5 epoch)	0.9776	0.9803	0.9947	0.9875
Fine-tuned model (10 epochs)	0.9542	0.9348	0.9425	0.9386
Fine-tuned model (15 epochs)	0.9685	0.9475	0.9416	0.9445

Discussion

The results indicate that fine-tuning the ResNet50 model significantly improves classification accuracy. While the feature extraction model achieved an 89% accuracy, the enhanced fine-tuning approach with 5 epochs reached up to 98% accuracy. However, the consistency of these results varied, as seen when repeating the experiment. Longer training (10 and 15 epochs) demonstrated stable yet slightly lower accuracy, suggesting potential overfitting or instability in training.

Future work could involve experimenting with different architectures or further fine-tuning to achieve more stable results. Additionally, incorporating data augmentation techniques might help improve generalization and performance.

Conclusions

This research addresses the problem of misuse of generative technology and proposes some solutions in the form of models for detecting AI-generated images. The feature extraction technique did not perform well initially, but by making adjustments to the fully connected layer and training for different epochs, better results were obtained.

The comparison of both models trained for different epochs was discussed, showing that fine-tuning the ResNet50 model can significantly improve classification accuracy. Future updates could involve applying data augmentation techniques and training with larger datasets to further enhance the model's performance.

References

1. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, et al. (2014) Generative adversarial nets. Advances in neural information processing systems 27. [Google Scholar]
2. Jovanović R (2022) Convolutional neural networks for real and fake face classification. In Sinteza 2022-International Scientific Conference on Information Technology and Data Related Research: 29-35. [Crossref] [Google Scholar]
3. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition: 770-778. [Google Scholar]
4. https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces.
5. https://www.kaggle.com/c/deepfake-detectionchallenge/discussion/122786.
6. https://www.kaggle.com/c/deepfake-detection-challenge/discussion/121173.