AI-generated Image Detection using Deep Learning Methods (Track Number #1)

Group Number #12: Zheng, Kuang

Abstract

With the advancement of deep learning techniques, generative AI has enabled the creation of photorealistic images, leading to widespread concerns about deepfake content. Detecting AI-generated images is crucial to mitigating misinformation. This study evaluates various deep-learning models—CNN, Densenet, Augmented Neural ODE, and Vision Transformer (ViT)—by training them on the CIFAKE dataset and assessing their generalization ability on a blurred variant. Results indicate that the ViT model outperforms other convolution-based methods, demonstrating superior accuracy and robustness. Additionally, uncertainty estimation using deep ensemble methods reveals confidence disparities across models, with ViT exhibiting the least predictive uncertainty. These findings contribute to a fair comparative analysis of deepfake detection methodologies and highlight the strengths of transformer-based architectures.

1 Introduction

With the development of deep learning techniques, generative AI that is capable of producing photorealistic images is essentially accessible to everyone. This leads to the proliferation of so called "deepfake" contents where machine learning techniques are used to depict real people. Deepfake has gained widespread attention for their potential use in creating misinformation. Therefore, it is critical to develop a method that could robustly detect these AI-generated images.

Deep-learning based methods for detecting deepfake contents have also evolved, from conventional CNN, to RNN, VGG net, densenet, vision transformer, and hybrid models. However, there seems to be a lack of fair comparison among these models due to inconsistent training setting and datasets throughout literatures. For example, in Weir et al. [2024], the performance of a ViT model is similar or even slightly worse than a basic CNN model. While in Arshed et al. [2023], the ViT model performs much better than other convolutional models. Therefore, we want to add to the current literature by providing a fair comparison among these models. Specifically, we will train & test them on the CIFAKE dataset Bird and Lotfi [2023], testing the generalization ability on the blurred dataset, and adding uncertainty estimation to these models.

2 Related work

Conventionally, CNNs are used to detect AI-generated images due to their ability of revealing subtle features in these images. In Bird and Lotfi [2023], the authors created the CIFAKE dataset that contains 12000 images, half of which are generated from the Stable Diffusion version 1.4 (SDv1.4) using authentic images in CIFAR10 dataset, and the other half are real images from CIFAR10. Then, they used a network comprised of 2 convolutional layers with 32 filters each and 2 fully connected layers to classify the images, achieving an accuracy of 92.93%. In Jiang [2024], the author explored the generalization ability of the CNN structure proposed in Bird and Lotfi [2023] by training the network on the CIFAKE dataset but testing it on datasets that contains images generated using other models (e.g., SDv2.1 and SDv3.0). They found that the original network does not generalize well

Preprint. Under review.

across various models. To address this issue, hybrid methods that combines various models are proposed. In Kaddar et al. [2021], the authors combined a CNN with a Vision Transformer (ViT), and named the resulting architecture as HCiT. Specifically, they used the Xception CNN to learn low-level features of the cropped image, then fed these features into a ViT to get the binary classification. It achieved impressive accuracy (~99%) on both training and test sets (CIFAKE10), suggesting a better generalization ability than the CNN models. In Weir et al. [2024], the authors further modified HCiT by adding an additional attention mechanism layer between the ViT and dense layer to capture the interaction between local and global features and slightly improved the accuracy.

3 Methodology

3.1 Dataset

In 2023, Bird and Lotfi Bird and Lotfi [2023] introduced the CIFAKE dataset that contains 120K 32 by 32 RGB images, 60K of which are real images sourced from the CIFAR-10 dataset Krizhevsky and Hinton [2009], and the remaining 60K are fake images generated using stable diffusion 1.4 from the CIFAR-10 dataset. Each image is associated with a binary label, where '0' means fake and '1' means real. We chose this dataset because it was widely used in many other works and relatively easy to train.

By default, the train split contains 100K images and 20K for the test split. However, due to the large model size and hardware constraint, we randomly select 50K images as the training set, and 10K as the test set, which is achieved by first shuffling the dataset, then selecting the first 50K & 10K samples from the train & test split.

We did not apply any transformation on the dataset other than scaling the pixel values to [0,1] with the exception of ViT. The pretrained ViT model imposes an input image size of 224 by 224, so we resize the images using the preprocessor provided by the model after scaling its pixel values.

To test the generalization ability of the models, we created the CIFAKE Blur dataset by applying Gaussian blur on the CIFAKE dataset. The Gaussian blur was created using the *GaussianBlur* function in *torchvision.transforms* with a random kernal size of 3 or 5, a minimum sigma of 0.4 and a maximum sigma of 0.6 chosen uniformly.

3.2 Vanilla CNN

In Bird and Lotfi [2023], Bird and Lotfi proposed to a 2-layer CNN to classify CIFAKE dataset and achieved an accuracy of roughly 93%. Yet they didn't release the detailed architecture about their model and the training parameters. To reproduce their result, we employed the following architecture and used grid searching to find the appropriate parameters. This vanilla CNN is served as the baseline model in the experiment.

Figure 1: Architecture of vanilla CNN

Both of the two conv layers have 32 filters with kernal sizes equal to 3 and paddings of 1. The two max pooling layers have kernal sizes 2 and strides 2. The two linear layers are followed by a sigmoid function to regress the output between 0 and 1 for binary classification.

3.3 Densenet

Densenet Huang et al. [2018], proposed in 2017 by Hunag et. al, is an efficient convolutional architecture for computer vision tasks. In this project we used the pretrained Densenet121 network which is the smallest densenet in torchvision and slightly modified it to accommodate to the dataset. Specifically, the foremost conv0 block has a kernal size of 7, a stride of 2, and a padding of 3 by default. If the input has a pixel size of 32 by 32, it will shrink to 16 by 16 after this block. Therefore, we modify this block to have a kernal size of 3, a stride of 1 and a padding of 1, which will not change

the pixel size of the input. Also, the final linear classifier layer is modified to have output dimension equal to 1. Again, we have an extra Sigmoid function at the end to ensure the output lies between 0 and 1.

3.4 Augmented Neural ODE

In 2018, Chen et al. proposed the neural ODE Chen et al. [2019], an architecture that uses a neural network to parametrize the dynamics of an ODE. The inferencing is then just solving the ODE from t=0 to T with some black-box differential equation solvers. However, it was later shown that the neural ODE architecture is not a universal function approximator due to the fact that the flow of an ODE is a homemorphism, i.e., the flow of an ODE cannot change the topology of the input space. To address this, Dupont et al. proposed augmented neural ODE Dupont et al. [2019] in 2019, which augments the input to higher dimensional space by appending 'zeros' to the input and solve the ODE at the augmented space. By doing so, the trajectories of the ODE at different input points are less likely to intersect with each other, giving rise to simpler flow the ODE learns.

In our experiment, we used a convolutional architecture to parametrize the dynamics. The network involves 3 conv layers, with a hidden channel number of 64. The output of the ODE solver is then flattened and passed to a linear classifier to generate the final value. We augmented the input by 10 additional zero channels, so the input tensor shape increases from (3, 32, 32) to (13, 32, 32).

3.5 Vision Transformer

Vision transformer (ViT), introduced in 2020 Dosovitskiy et al. [2021], extends the transformer architecture, which was initially designed for NLP tasks, to computer vision tasks. The proposed network architecture is completely identical to the standard transformer encoder, with the only exception being the way they handle the inputs. Given an input tensor (image) of shape (C, H, W), we first split it in to N non-overlapping patches with shape (C, P, P), where $N=HW/P^2$. Then, we unflatten these patches, and stack them into a tensor of shape (N, CP^2), which is the effective token sequence to the embedding. The pipeline after this step is identical to the standard transformer encoder network: we embed the token sequence by a linear layer, add a positional encoding, and feed it through an encoder stack.

4 Experiments

4.1 Metrics

The metrics we used to evaluate the performance of the model are accuracy and loss. The loss criterion we used is the BCE loss as defined in the *torch.nn.BCEloss* class. For the uncertainty estimation, we used the NLL loss, which is given by the *torch.nn.GaussianNLLLoss* class.

4.2 Experiment 1

In this experiment we trained the vanilla CNN, densenet121, NODE and ViT on the CIFAKE dataset. The training parameters for the models in this experiment are specified in the following table. We used the Adam optimizer with a weight decay of 1e-4, and the StepLR scheduler with a gamma of 0.5. All models have the same precision of fp32, and are trained with an epoch number of 10.

	Batch size	Learning rate	#trainable para.	Training time
vannila CNN	32	1e-3	0.14M	8min
Densenet121	32	1e-3	6.95M	20min
NODE	32	1e-3	0.05M	45min
ViT	64	2e-5	86.57M	5h20min

Table 1: training parameters

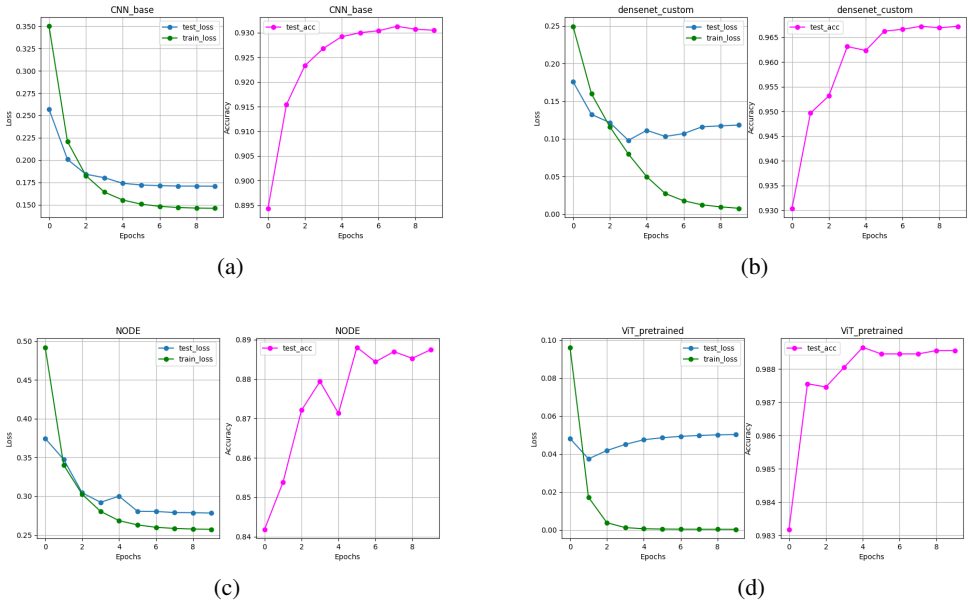


Figure 2: Learning curve of (a) vanilla CNN (b) Densenet121 (c) NODE (d) ViT

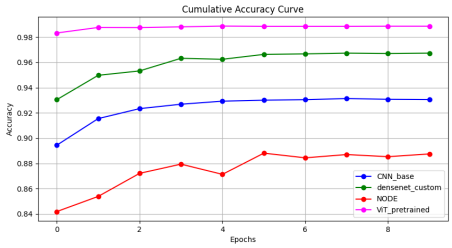


Figure 3: Cumulative accuracy curve

Several observations can be drawn from these figures. First, the vanilla CNN does successfully reproduce what was presented in Bird and Lotfi [2023], with a test accuracy of roughly 93% on the CIFAKE dataset. The accuracy and loss converge quite smooth, and the test and training loss decrease consistently over the whole 10 epochs, which implies a potential underfitting of the vanilla CNN model. This should not be surprising given the simple structure of the vanilla model.

An opposite story can be observed from the densenet and ViT model. After some epochs, the training loss and test loss start to diverge consistently, which indicates model overfitting. Despite this, their test accuracies converge smoothly and consistently outperform that of the CNN model. We can also see a clear gap between the accuracy curve of the densenet and the ViT, which one might attribute to the gap between the sizes of these 2 models. But, given the fact that even the smallest densenet is overfitting on the dataset, using a more complex architecture will not help. We fine-tuned the pretrained densenet201 network which is three times the size of densenet121 on the same dataset, and got exactly the same loss and accuracy curves. This showcases the superiority of the transformer-based architecture against the conventional convolutional architectures, where the model can exploit global connections among image patches.

For the NODE model, we can observe relatively bumpy and inferior accuracy curve compared to other models, which indicates the model does not learn the patterns very well. One can argue that the reason for this is that we used such a simple NN (only ~0.05M trainable parameters) that it is impossible to represent the true dynamics. However, further increasing the complexity of the NN does not offer better performance, and actually even worse. It is then obvious that using the flow of an ODE to transform the input image is not a good choice for classification task at least on this

dataset. We also want to highlight that our finding is aligned with what was shown in Dupont et al. [2019], where they use a similar architecture to classify the CIFAR10 dataset, and got a test accuracy of ~60%, which is far below other convolutional models.

4.3 Experiment 2

In this experiment we pick the state of the model that has the highest test accuracy during training, and test it on the blurred CIFAKE dataset. The purpose of this experiment is to test the generalization ability of these models.

	vanilla CNN	Densenet121	NODE	ViT
CIFAKE	0.93	0.97	0.89	0.99
CIFAKE+Blur	0.79	0.82	0.76	0.90

Table 2: Test accuracy on CIFAKE and blurred CIFAKE

We can observe that all models see a drop in accuracy when we test it on the blurred dataset, which is not surprising. What's notable is that the drop in accuracy of ViT is much lower than other models, which indicates that this model really learns the underlying features well.

4.4 Experiment 3

In this experiment we used the deep ensemble method presented in Lakshminarayanan et al. [2017] to train the vanilla CNN, densenet121 and ViT models. In general, we modify the final linear layer of these models to output two values, one is the posterior mean, and the other is the posterior variance. The positivity constraint of the variance is enforced by applying a softplus function to the second output. Then, instead of using the BCE loss, we used the NLL loss to capture the predictivity uncertainity. This is implemented in the *torch.nn.GaussianNLLLoss* class. We also employed adversarial training to smooth the predictive distribution.

Unfortunately, due to constraint on time and hardware, it is nearly impossible to train an ensemble of networks for each involved model. Therefore, we restrict ourself to just one network for each model. We trained the models on CIFAKE dataset and tested them on either the CIFAKE or the blurred CIFAKE dataset. The training procedure is summarized as follows:

Algorithm 1 An algorithm with caption

$\epsilon \leftarrow 0.01$
for $\mathbf{x_b}$ **in** $trainloader$:
$\mathbf{x_{adv}} \leftarrow \mathbf{x_b} + \epsilon \times sign\left(\nabla_{\mathbf{x}} l\left(\theta, \mathbf{x}, y\right)\right)$
$\theta \leftarrow \arg\min_{\theta} l\left(\theta, \mathbf{x}, y\right) + l\left(\theta, \mathbf{x_{adv}}, y\right)$

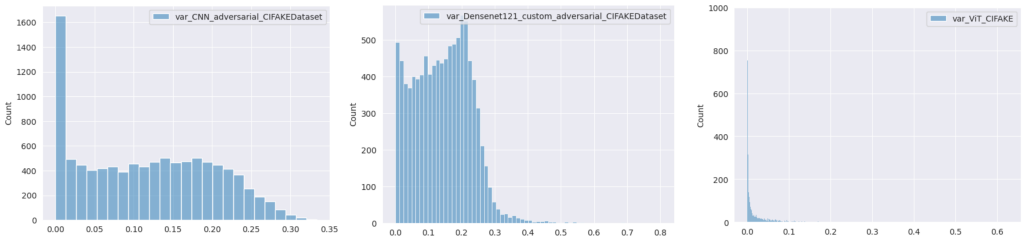
The following table and figures show the result of this experiment:

	vanilla CNN	Densenet121	ViT
CIFAKE	0.87	0.90	0.97
CIFAKE+Blur	0.79	0.80	0.87

Table 3: Test accuracy of the modified models on CIFAKE and blurred CIFAKE

We can see that instructing the model to optimize for NLL instead of MSE results to a decrease in accuracy. Among the three models, densenet sees the largest accuracy drop, while ViT almost remains the same accuracy.

What's more interesting are the figures showing the distribution of variance in the test dataset. There are 10000 points in the test set, so the length of bars in each figure adds up to 10000. We can see that for CNN, the variance is pretty much spread out uniformly from 0.01 to 0.25, and there is a long bar at the first bin. This means the model is very confident for a small portion of the points, and have a small degree of uncertainty for the majority of the points. For the densenet, the variance spread out uniformly from 0 to 0.3, which means the model show a moderate degree of uncertainty across the test set.



(a) (b) (c)

Figure 4: Distribution of variance across the test set of (a) vanilla CNN (b) densenet121 (c) ViT

For the ViT, however, we can see a completely different story. Over 90% of the points cluster in the interval 0 to 0.04, meaning that the model is extremely confident of its decision, another advantage of ViT over other models.

5 Conclusion

In this project, we trained the vanilla CNN, densenet121, augmented NODE and ViT models to on the CIFAKE dataset. When tested on the CIFAKE dataset, the ViT model exhibited exceptionally well performance with an accuracy of ~99%. The densenet121 also performs well, with an accuracy of ~97%. The NODE model, however, does not lend itself well to this classification task, with an accuracy of less than 90%.

To test the generalization ability of these models, we then tested them on the blurred CIFAKE dataset. The densenet121 sees the largest accuracy drop of ~15%, while the ViT is the least impacted, with an accuracy drop of ~9%.

Lastly, we modify the models to capture the predictive uncertainty by outputting the posterior mean along with variance, and optimizing for NLL loss. Adversarial training is also employed in the training procedure. We find that both the vanilla CNN and densenet121 see a drop in accuracy and show moderate degree of uncertainty across the test set, while the ViT remains a high accuracy and show very little uncertainty about its prediction.

Many questions still remain open. First, we ran our experiments on the relatively simple dataset CIFAKE, where the image size is only 32 by 32 and are all generated by Stable Diffusion 1.4. What if we choose a more complex dataset like Zhu et al. [2023] where there are over 1M samples that are generated by different generators? Second, we could include more modern models to our analysis. Third, for the uncertainty estimation part, we could train an ensemble of networks to better quantify the uncertainty. Also, we could have look at the uncertainty on out-of-distribution samples by testing the model on a completely different dataset from what we trained it on.

The following sections do not count towards the 8-Page Limits.

Appendix

Github Link

General

Team Member Contribution

All works are done individually by Zheng Kuang.

References

M. A. Arshed, A. Alwadain, R. Faizan Ali, S. Mumtaz, M. Ibrahim, and A. Muneer. Unmasking deception: Empowering deepfake detection with vision transformer network. *Mathematics*, 11 (17), 2023. ISSN 2227-7390. doi: 10.3390/math11173710. URL https://www.mdpi.com/2227-7390/11/17/3710.

J. J. Bird and A. Lotfi. Cifake: Image classification and explainable identification of ai-generated synthetic images, 2023. URL https://arxiv.org/abs/2303.14126.

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations, 2019. URL https://arxiv.org/abs/1806.07366.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

E. Dupont, A. Doucet, and Y. W. Teh. Augmented neural odes, 2019. URL https://arxiv.org/abs/1904.01681.

G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks, 2018. URL https://arxiv.org/abs/1608.06993.

J. Jiang. Addressing vulnerabilities in ai-image detection: Challenges and proposed solutions, 2024. URL https://arxiv.org/abs/2412.00073.

B. Kaddar, S. A. Fezza, W. Hamidouche, Z. Akhtar, and A. Hadid. Hcit: Deepfake video detection using a hybrid model of cnn features and vision transformer. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5, 2021. doi: 10.1109/VCIP53242.2021.9675402.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017. URL https://arxiv.org/abs/1612.01474.

S. Weir, M. S. Khan, N. Moradpoor, and J. Ahmad. Enhancing ai-generated image detection with a novel approach and comparative analysis. In *2024 17th International Conference on Security of Information and Networks (SIN)*, pages 1–7, 2024. doi: 10.1109/SIN63213.2024.10871900.

M. Zhu, H. Chen, Q. Yan, X. Huang, G. Lin, W. Li, Z. Tu, H. Hu, J. Hu, and Y. Wang. Genimage: A million-scale benchmark for detecting ai-generated image, 2023. URL https://arxiv.org/abs/2306.08571.