CONCORDIA UNIVERSITY OF EDMONTON
FACULTY OF SCIENCE
DEPARTMENT OF MATHEMATICAL AND PHYSICAL SCIENCES

Senior Project Capstone Report

IT 452 Senior Project Capstone II

The effectiveness of various Machine Learning Algorithms on detecting Synthetically Generated Imagery

Author:

Guillaume Comfort

Student, Concordia University of Edmonton

Academic Advisor:

Nasim Hajari

Assistant Professor, Concordia University of Edmonton

April 16, 2024

Abstract

In the face of a rise in A.I generated content, namely what it can generate in imagery, there is a new vector in which misinformation can be spread by. The prevention of misinformation via generative content ultimately lies in the creators of such machine intelligence via policies and regulations, as well as individuals being educated to fight against such misrepresentation of information. However, these methods does not stop the threat of it ultimately being used maliciously by individuals bypassing such. This research project intends on researching the effectiveness of various Machine Learning Algorithms using examples of A.I generated artwork obtained via generating said content through means such as DALL E, CIFAKE, etc. This A.I artwork will be contrasted with non-fabricated photography and artwork obtained via personal means or with credit attributed. The results obtained are then used to determine and explore the effectiveness of varying algorithms in detecting synthetically generated imagery from different datasets.

Keywords: A.I, Misinformation, DALL-E, CIFAKE, Synthetically-Generated.

1 Introduction

The popularity of generative models have skyrocketed in recent years, especially due to the spread of GANs and diffusion models.[1] [2] Because of their image generation abilities and their training, they have the ability to generate shockingly realistic images. [1] [2]. Because of this, they also risk being used by malicious actors to spread misinformation.[3]

This project explores a variety of Machine Learning Algorithms in detecting Synthetically Generated Imagery against Images and Artwork that are otherwise not generated through the use of A.I. It arose from a desire to further research into methods and the effectiveness of such methods in categorising A.I imagery and separating it from authentic media in an effort to combat against the potential misuse of generative models. Currently, these models have the potential of being used as methods of spreading misinformation regardless of any software policies and regulations.[1]

Without a clear and consistent way of detecting A.I generated imagery, this threat could grow exponentially as the use of Machine Learning and Generative A.I becomes more ingrained and widely available in the every day life. [3] As it does become more ingrained in the every day life of individuals, so too can it become ingrained with malicious actors trying to use it for their own gain Politically, Monetarily, Religiously, and so on.

By exploring the methods, effectiveness, and results of different Machine Learning Algorithms in detecting this imagery across different data sets such as the CIFAKE [4] dataset, this project aims to provide future insight into the use of Machine Learning to spot images generated using A.I, whether malicious or not.

2 Objectives / Research Questions

Some questions that have arose throughout the process and the drafting stage of this project.

The most important of which being, "What is metric for determining something that is created through the use of generative models that distinguishes itself from real photography." While methods such as spotting faults in the image, the addition of extra fingers for example, or faulty perspective may seem obvious to a human viewer, it could potentially lead to a model that wrongfully labels disabilities in people, or faults in drawn artwork, as content that has been generated by an A.I.

Having a model that also detects issues off of these errors can also become obsolete as generative models advance in technology to better generate images without these errors, and so then a sub-question would be "How do we create a model to detect A.I generated content that stays consistent in its detection regardless of advances in such technology."

3 Literature review (and theoretical framework)

The following words are defined in order to better understand this project's goals and intentions

A.I - Artificial Intelligence, any program that uses a machine learning framework in order to generate a desired output. DALL-E, Stable Diffusion, and so on.

GAN(s) - Generative Adversarial Models, models that compete against each other to become more accurate. Synthetically Generated - Content that has been generated through the use of A.I

Artwork - Any sort of images or collections thereof, created for the purpose of eliciting emotions. Photography, Digital Art, Traditional Art, etc.

Misinformation - Any sort of media created with the intent of deceiving an audience for any purpose, be it malicious, or for any sort of ulterior motives.

As noted by works cited previously, Machine learning and Artificial intelligence have been growing at an incredible rate,[3] [1] and while these developments can be a vector for good, there is the nigh-impossible to avoid threat of it being used for misinformation. [2]

As Walker et al. [5] notes, misinformation can be very dangerous, it can be used to undermine democracy, reduce the consensus on climate change, and further exacerbate crises. Because of the growth of machine learning algorithms, there is especially a need to curb the misinformation that may arise from it. While education in being able to spot misinformation effectively and to do the research necessary on the subject is a very effective strategy, ultimately not everyone has access to the resources

nor time to go through with it. This is where a free and open-source detection model is tantamount.

The proposed model employed by Chen et al. [2] first randomly crops 192 patches of the original image with the dimensions 32 x 32. These patches are then resized to 256 x 256 and uses a SRM to extract the noise pattern and records information using 3 high-pass filters.

4 Scope of Work

This project ultimately undergoes finding various machine learning algorithms throughout sources such as Kaggle and other parts of the internet. Using datasets collected through the same sources and through paperswithcode. By running the datasets through the code obtained, this will provide insight into how detection models work in regards to detecting synthetically generated imagery, and if the very same models can work on different data sets.

So far, my timeline for this project has been as followed:

January - The collection of the models and data, testing different models with a singular data set.

February - The collection of new data sets, testing those data sets with the existing models and noting any changes

March - Finishing the testing process and reading up more on the theory and literature surrounding the topic. Writing Project Report

April - Working and finishing the final report and presentation.

5 Project Design

This project uses images that were generated from Machine Learning Algorithms such as Dall-E, Midjourney, [6] and Stable Diffusion [4], This project also uses the open-source code found from the Kaggle page for the CIFAKE dataset in particular. [8] and the CIFAKE dataset itself [4][7][8] which can be found on Paperswithcode [9]

The code in particular used is a CNN model, and focuses on first training the model, and then testing the model. I decided to explore two different datasets, the DALL-E/Midjourney dataset, and the CIFAKE dataset, and ran it with 5, 10, and 20 epochs using the original amount of filters, and then also explored heightening the filters, running it with the original amount of filters first, and then doubling the filters with the hope that it would generate different results.

Two of the models used crops the image into 32 x 32 sized images, appends the model on to an array and then runs it. Another crops it into images 256 x 256 images.

6 Dataset

The datasets utilised for this project were:

CIFAKE: Real and AI-Generated Synthetic Images [9], utilising real images from Krizhevsky and Hinton's CIFAR-10 dataset [7], and synthetically generated images from Bird and Lotfi [4]. The synthetically generated images were generated to be equivalent to those found in the CIFAR-10 dataset and generated using Stable Diffusion 1.4. There are 100,000 images used for training (50,000 each of fake and real images) and 20,000 images used for testing purposes (10,000 each of fake and real images). The images in this dataset are all 32x32 pixels

DALL-E Dataset [6], this dataset comprises of real images taken from the public domain and synthetically generated images that were generated using DALL-E, that the person who made the dataset scraped from the internet. In total there are around 21,600 images, of which around 17,900 of which were synthetically generated. The file size of the images meant less were used than the CIFAKE dataset in order to offset memory usage and time spent. The images in this dataset are all in varying dimensions.

7 Running the Algorithms

All Algorithms were run on a local instance of Jupyterlab using my own linux laptop. with the tensorflow library.

How the project is run is that it is trained for an X amount of epochs, of which the X I picked was 20 for the CIFAKE datasets, and 10 for the DALL-E Dataset since it has images of larger file-size, this was to give it sufficient testing, but also to see if it suffered from damping returns. On the first algorithm it is then tested against 625 random images within the testing dataset, and it's final accuracy is the accuracy of that final test.

The first Algorithm was found on kaggle's code section for the CIFAKE dataset [10] and was slightly modified for the purposes of experimenting with the code and also to modify it for use with the DALL-E dataset. The Algorithm utilises a CNN model, and 5 layers, 3 are 2D convolution layers utilising a Rectified Linear Unit activation, 1 is a dense layer using the same ReLU activation, and the final layer is a sigmoid activation to classify the image as either a fake or real image. The optimisation is done using the Adam optimisation algorithm. This algorithm uses a sparse categorical cross entropy for the loss

Using the following parameters, the first algorithm was ran:

80 filters for the first layer, followed by 40, and then 20. It gave the following accuracy over 20 epochs for the CIFAKE. for the CIFAKE dataset, this took on average 206s (3m26s) per epoch, about 1 hour and 9 minutes in total. The final test evaluation

took 14 seconds. The change in accuracy from the last milestone epoch is also noted. On average using the DALL E dataset, it took 314s (5m13s) per epoch, about 1 hours and 45 minutes in total. The final test evaluation took 23 seconds.

For the purposes of the tables, the middle row is using the CIFAKE dataset, the bottom row is the DALL-E dataset.

Table 1: Accuracy of First Algorithm against the datasets.

-	5 epochs	10 epochs	15 epochs	20 epochs	Final Test
Accuracy	0.9260	0.9440	0.9529	0.9606	0.9101
Change	-	0.018	0.0089	0.0077	-0.0505
Accuracy	0.9043	0.9235	0.9338	0.9414	0.9018
Change	-	0.0192	0.0103	0.0076	-0.0396

As advised, I then changed the amount of filters in the algorithm to check if there was any different, and got this as a result with 160 filters for the first pass, then 80, then 40 (doubling the amount of filters in each layer). It took on average 383s (6m23s) per epoch, for a total time of about 2 hours and 8 minutes, or just shy of a few minutes for double the time of the algorithm with the original amount of filters. The final test evaluation took 21 seconds. With the DALL-E dataset, doubling the filters meant the algorithm took on average 608s (10m8s) per epoch, for a total time of 3 hours and 23 minutes.

Table 2: Accuracy of First Algorithm against the datasets, double filters

-	5 epochs	10 epochs	15 epochs	20 epochs	Final Test
Accuracy	0.9337	0.9540	0.9666	0.9755	0.9336
Change	-	0.0203	0.0126	0.0089	-0.0419
Accuracy	0.9184	0.9400	0.9534	0.9618	0.9198
Change	-	0.0216	0.0134	0.0084	-0.0420

Next is the 2nd algorithm I tested for this project. The code for this can be found on kaggle much like the first code. [11]. Much like the first code, this one utilises the adam optimiser. It uses 5 layers, 4 of which have an activation of ReLU, and the first 3 are 2d Convulation layers, followed by a Dense layer and the last one being a Dense layer with an activation of sigmoid to classify the image as either fake or real. It inputs the image to be 256 by 256 and it starts off first with a lower number of filters, as opposed to the previous which starts off at a high number. This algorithm also uses a binary cross entropy for the loss which differs from the first algorithm.

For the filters we start at 16 for the first layer, 32 for the 2nd, 64, and finally 128 for the last, with the cifake dataset, it took on average 2010s (33m30s) per epoch for a total run time of 11 hours and 10 minutes. The DALL E dataset took on average 2340s (39m) per epoch, for a total run time of 13 hours.

Table 3: Accuracy of Second Algorithm against the datasets

-	5 epochs	10 epochs	15 epochs	20 epochs
Accuracy	0.9220	0.9696	0.9833	0.9875
Change	-	0.0476	0.0137	0.0042
Accuracy	0.9132	0.9644	0.9757	0.9777
Change	-	0.0512	0.0113	0.0020

What surprised me the most was that in this algorithm it has a dramatic change between 5 epochs and 10 epochs, but then suffers very heavily from damping returns on further training in both datasets, although does reach accuracies higher than even the first algorithm with it's original amount of filters, although taking lot more time to run.

With double the filters, the time it took roughly doubled much like the first algorithm. In total it ran for almost an entire day with both datasets. While the initial accuracy was greater than the original, this algorithm suffered immensely with damping returns and almost seemingly reached its cap by the 13th epoch, however it managed to achieve an accuracy of 99% with the CIFAKE dataset after 20 epochs. Which is still an impressive metric although not in a feasible time frame for large datasets. With that in mind, I got these results:

Table 4: Accuracy of Second Algorithm against the datasets, Double filters

-	5 epochs	10 epochs	15 epochs	20 epochs
Accuracy	0.9348	0.9837	0.9875	0.9901
Change	-	0.489	0.0038	0.0026
Accuracy	0.9242	0.9715	0.9821	0.9852
Change	-	0.0473	0.0106	0.0031

Finally, the last algorithm I tested for this project. The code much like those before it can be found on kaggle. [**?**] This code uses the adam optimiser. It uses 4 layers, an Efficient Net V2 B0 layer, and 3 Dense layers, 2 of which are ReLU activation and 1 is a sigmoid activation to classify the image as either fake or real. The amount of layers in 2 of the dense layers are 64 and 128, and it is prioritised with the Efficient Net first, then 64, then 128. In this algorithm I noted had noted that doubling the amount of layers in the 2 dense layers had very little impact on the accuracy and time spent overall and so did not experiment with that further. On top of this, its accuracy seemingly reached the cap by around the 15 epoch mark and so did not go further than 15 epochs to save time further. The time the algorithm took with the CIFAKE dataset was about 8490s per epoch, (2h21m) for a total time spent running just at 35 hours, and with the DALL-E dataset, about 9320s per epoch, (2h35m) for a total time just over 38 hours. A vast increase in time spent over the previous two algorithms as the amount of trainable parameters was vastly higher as well. Most

definitely because of this, it managed to achieve an accuracy that far surpassed the other algorithms tested, with an accuracy of nearly 99.3% with the CIFAKE dataset, and 99.15% with the DALL-E dataset. The change in accuracy with the DALL-E Dataset was greater as well, which I had attributed it to the algorithm being able to learn more off of it with the increase in data and gain more than the CIFAKE which will maximise itself quickly already.

With all this, I got these results

Table 5: Accuracy of Third Algorithm against the datasets

-	5 epochs	10 epochs	15 epochs
Accuracy	0.9790	0.9881	0.9927
Change	-	0.0091	0.0046
Accuracy	0.9689	0.9831	0.9915
Change	-	0.0142	0.0084

Comparing all the Algorithms side by side, we get the following tables:

Table 6: Accuracy of all 3 algorithms - CIFAKE

Algorithm	5 epochs	10 epochs	15 epochs	20 epochs
Algorithm 1	0.9260	0.9440	0.9529	0.9606
Algorithm 2	0.9220	0.9696	0.9833	0.9875
Algorithm 3	0.9790	0.9881	0.9927	-
Algorithm 1 Dbl	0.9337	0.9540	0.9666	0.9755
Algorithm 2 Dbl	0.9348	0.9837	0.9875	0.9901

Table 7: Accuracy of all 3 algorithms - DALL E

Algorithm	5 epochs	10 epochs	15 epochs	20 epochs
Algorithm 1	0.9043	0.9235	0.9338	0.9414
Algorithm 2	0.9132	0.9644	0.9757	0.9777
Algorithm 3	0.9689	0.9831	0.9915	-
Algorithm 1 Dbl	0.9184	0.9400	0.9534	0.9618
Algorithm 2 Dbl	0.9242	0.9715	0.9821	0.9852

8 Conclusions

As I had initially expected in the tested algorithms with both datasets, there was a notable damping return on the accuracy in both the original amount of filters and with double the filters which does suggest there is both an ideal amount of epochs and an ideal amount of filters to train an algorithm off of to detect synthetically generated images to get the most accuracy with the least amount of time spent. If one's goal is to merely achieve a success accuracy rating of at least 90%, 5 epochs of training showed to be satisfactory in most of the algorithms against CIFAKE in generating an accuracy rating of 90%. 10 epochs yielded the most positive change in accuracy overall in all cases. In the case of filters, it was beneficial in this case to use a lower amount of filters as it yielded a satisfactory accuracy with the least amount of time spent.

Something that was also recorded was that in all algorithms, the time spent running the algorithms was greater and the accuracy was slightly lower overall using the DALL-E dataset although it had a higher gain in accuracy than the CIFAKE dataset as the various algorithms got accustomed to the dataset. However, it was not able to truly catch up to the CIFAKE dataset. I had expected this to be the case as all images used in the dataset are much larger than those in the CIFAKE dataset and so there would be more data to work with within the images.

The accuracy in the algorithms that had its filters doubled, scaled proportionally in time, taking nearly double the amount of time to complete the full training process. Although it did result in a higher accuracy overall during the training and a higher final accuracy, because of the time it would take to train the data using a large amount of filters, it would take an increasing amount of time with larger datasets. The initial amount of filters would be much more convenient on a mass-scale and large datasets to work with, but on a smaller scale where there is not as much to work with, the time would be definitely negligible to warrant the extra accuracy for if that is something desirable.

Something that also caught my attention was that the amount of trainable parameters influenced heavily the accuracy and time spent on the algorithms. Although this is a result of the amount of filters and layers the particular algorithm had, the algorithm that had the most parameters which was the 3rd algorithm tested vastly outperformed the other algorithms in accuracy with the least amount of epochs. Understandably it also took the most time out of any of the algorithms, nearly 35 times the amount of the first algorithm on the CIFAKE dataset alone.

With the success with the algorihtms in being able to determine accurately the A.I generated imagery from the real imagery in both the DALL-E and the CIFAKE datasets, there is also the possibility of even more datasets of being able to be used to train the models off of to accurately point out synthetically generated imagery and would only need a bit of tweaking in order to get the algorithms to function with the differing format in how the images are in the folders.

9 Acknowledgments

I would like to thank the following:

Dr. Apoorva Chauhan, for her invaluable feedback, teaching, and guidance throughout the first part of the Senior Capstone Project process in IT451

Dr. Nasim Hajari, for the invaluable feedback and guidance, as well as providing me with a wide range of resources and papers throughout the second part of the Senior capstone Project process in IT452

The varying sources to which I have referenced for providing me with a framework and theoretical knowledge surrounding the subject, as well as the algorithms to which I have used for the study involving this project.

References

[1] Roberto Amoroso, Davide Morelli, et al, 2023. *Parents and Children: Distinguishing Multimodal DeepFakes from Natural Images*

[2] Jiaxuan Chen et al, 2024. *A Single Simple Patch is All You Need for AI-generated Image Detection*

[3] Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, Paul Scharre, Thomas Zeitzoff, Bobby Filar, et al. 2018. The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation. arXiv preprint arXiv:1802.07228 (2018).

[4] Bird, J.J., Lotfi, A. (2023). *CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images.* arXiv preprint arXiv:2303.14126.

[5] Johanna Walker, et al. 2023. *AI Art and Misinformation: Approaches and Strategies for Media Literacy and Fact Checking.* In Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society (AIES '23). Association for Computing Machinery, New York, NY, USA, 26–37. https://doi.org/10.1145/3600211.3604715

[6] https://www.kaggle.com/datasets/superpotato9/dalle-recognition-dataset

[7] Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images.*

[8] https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images/data

[9] https://paperswithcode.com/dataset/cifake-real-and-ai-generated-synthetic-images

[10] https://www.kaggle.com/code/deveshupreti/92-real-and-ai-generated-synthetic-images-cnn
Accessed 15/04/2024

[11] https://www.kaggle.com/code/abdulhaq786/cnn-90-percent
Accessed 15/04/2024

[12] https://www.kaggle.com/code/wontonflambe/ai-generated-image-detection-cifake-97
Accessed 15/04/2024