Detecting AI-Generated Images using ResNet

Ziyang Zeng and **Zhehu Yuan** and **Yifan Jin**
Dept. of Computer Science
New York University
251 Mercer Street, New York, NY
zz2960@nyu.edu, zy2262@nyu.edu, yj2063@nyu.edu

Abstract

Recent advances in diffusion models have set an impressive milestone in image-to-image generation tasks. Trending works such as DALL-E2 and Stable Diffusion have attracted great interest in academia and industry. However, there are lack of evaluation about the images generated by these methods. In this paper, we generate synthetic images with the "Stable Diffusion" image diffusion generation model. The mixed dataset is mixed with the synthetic images and the original images and used as training data and test data in machine learning applications to investigate the capabilities of the Stable Diffusion model. Analyses show that the model training with ResNet could distinguish between the synthetic images and the original images with very high accuracy.

1 Introduction

Introduction goes here.

2 Related Work

Related work goes here.

3 Datasets

In order to train a classifier that can distinguish between real and generated images, we built a labeled dataset that consists of both real and generated images for supervised learning. Datasets of this kind is not common, and we'd like to experiment on the latest state-of-the-art image generation AI models, such as DALL·E 2 and Stable Diffusion. We started with real photos from existing public datasets as our raw datasets. Then, we use Diffusers-based image-to-image model to generate image variations from real photos. This way, we can get a labeled dataset that consists of both real and generated images.

3.1 Raw Datasets

The original real-world photos we experimented with are from public datasets: Indoor Scene Recognition Database and Weather Image Recognition Dataset. Additionally, we want to test how well our model performs on non-photo art work. A dataset that consists of pages of comic art (the Comic Books Images Dataset) is also used in our experiments.

3.1.1 Indoor Scene Recognition Dataset

Originally targeting at indoor scene recognition tasks, the Indoor Scene Recognition Database is a collection of 67 indoor categories (e.g. airport, living room, restaurant...) and at least 100 images per category [**?**]. There are 15,620 images in the dataset in total. Figure 1 shows some examples of the dataset.

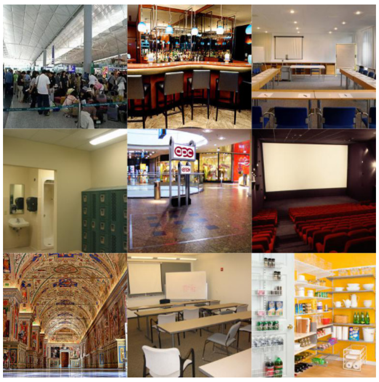

Figure 1: Indoor Scene Recognition Database Examples

This dataset is the biggest among the three datasets we used and serves as our main dataset. We use the AI-generated variations from these images and the images themselves to train our classifier.

3.1.2 Weather Image Recognition Dataset

The Weather Image Recognition Dataset contains labeled 6738 images of different types of weather, its catogories including dew, fog, frost, glaze, etc [**?**]. This dataset is used in our experiment to evaluate our model's generalization ability and performance on other kinds of photos than indoors.

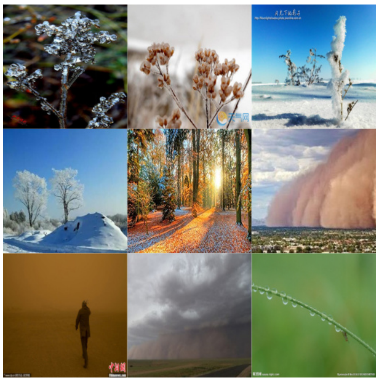

Figure 2: Weather Image Recognition Dataset Examples

3.1.3 Comic Books Images Dataset

The Comic Books Images Dataset is an open source dataset from Kaggle that contains 512 images from comic book pages [?]. Although our goal is to build a classifier to detect AI-generated photos versus real photos, and comic art seems to fall in neither category, we want to test whether our model can detect AI-generated comic art verses real comic art. This can further verify the generalization of our model. We what use to evaluation is a subset of this dataset, containing 513 images. Figure 3 shows some examples of the dataset.

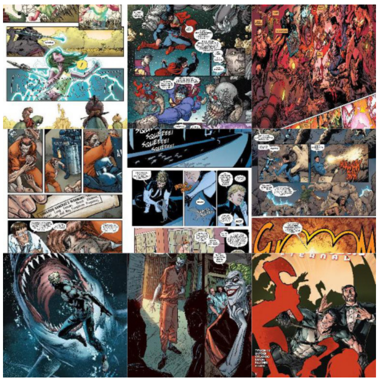

Figure 3: Comic Books Images Dataset Examples

3.2 Image-to-Image Generation with Diffusers

Instead of directly training on AI-generated images collected online, and compare them with the real photos from Section 3.1, we used an image-to-image (image-conditioned) generation pipeline to create AI-generated images from the real photos. The reason is that we think if we cannot make sure the AI-generated images and the real photos share the same themes, styles, content or objects, the model we train could learn that a certain theme/style of pictures containing certain things are from AI or real world without actually learning the general characteristics of the AI-generated photos. To mitigate this bias in the dataset, we came up with the solution of generating variations from given real world photos, so that our AI-generated images are of the same theme, similar style and content.

Even though diffuser-based image generation models have shown great success recently and achieved state-of-the-art results in conditioned image generation benchmarks, most of them are text-conditional, which means they take texts as an input to describe the images for the model to generate. In order for us to generate new variations of given real-world photos, we utilized an image-to-image generation pipeline that is present in DALL·E 2 APIs and a similar pipeline developed by Stable Diffusion community, which utilized image encoder output as conditioning.

Diffusion models, also known as DMs, are a type of generative model that sequenctially uses denoising autoencoders to synthesize new images. By decomposing the image generation process into a series of denoising steps, diffusion models can produce high-quality images that are similar to the ones they were trained on. To generate a new image, a diffusion model takes a noisy image as input, and starts with a certain step number. Then instead of directly predicting the denoised image, it uses denoising autoencoders to predict the noise added in the previous step, and then subtract that noise from the noisy image to get a less noisy image, because noise here is mathematically easier for the model to learn to predict. Then the slightly denoised image with a new step number that is smaller by 1 is fed into the autoencoder again to continue the denoising process. This process repeats until reaching step 1 where we have our final image.

3.2.1 CLIP guidance

CLIP (Contrastive Language-Image Pre-Training) is a neural network that has been trained on a large collection of (image, text) pairs. It has both an image encoder and a text encoder, and the goal of the model is to learn a representation of the images and texts that maximizes the cosine similarity between the representations of matching pairs [?]. As a result, CLIP model gives a great way to encode texts into vectors as embeddings, which is used in diffusion models like DALL·E 2 and Stable Diffusion to guide the image generation process so that the generated images fit the given texts. For every step of the diffusion process, the text encoder output from the CLIP model is used as image embedding into the denoising autoencoder to guide it to clear up noise. That way, the generated images are more likely to fit the description of the given texts, when the texts are used as conditioning.

A similar approach is to use the image encoder output from the CLIP model as image embedding into the denoising autoencoder. This is the approach we used in our project. We used the image encoder output from the CLIP model as image embedding into the denoising autoencoder to guide it to clear up noise. This way, the model appears to generate variations of the given image that are similar to the them.

3.2.2 Stable Diffusion

Stable Diffusion is one of the recent text-to-image models released in 2022 and is a latent diffusion model, a variety of deep generative neural network developed by the CompVis group at LMU Munich [**?**]. It combines the advantages of powerful pre-trained autoencoders and diffusion models, which are sequential applications of denoising autoencoders, to achieve state-of-the-art synthesis results on image data and beyond. Since Stable Diffusion is the only open-source one among the 3 most popular diffusion models and it is designed to reduce computational requirements while retaining visual fidelity can run on most consumer hardware equipped with a modest GPU, we can run it on Google Colab to generate part of our main dataset while paying no licensing fee for the model.

3.2.3 DALL·E 2

DALL·E 2, on the other hand, is a similar text-to-image generative model that also uses a diffusion model conditioned on CLIP image embeddings, and is developed by OpenAI as commercial product providing inference API services [**?**]. DALL·E 2 is trained on a large collection of proprietary captioned high-quality stock images other than scraped images from the internet like Stable Diffusion. As a result, DALL·E 2 is capable of generating more intricate and sophisticated images than Stable Diffusion or Midjourney, which is another closed-source Diffusion model that is more known for its artistic style [**?**]. DALL·E 2 comes with a image variation generation API which, under the hood, uses the image encoder from CLIP to condition the diffusion model, similar to what we use with Stable Diffusion. Because of our limited budget, we cannot afford to use DALL·E 2 to generate our main dataset, but we can use it to generate two small datasets for validation.

3.2.4 Image Variation Generation

As mentioned, we want to generate variations of the given real-world photos from our raw datasets, and then label both the generated images and the original real-world photos together to build a dataset for training our AI-generated image binary classification model. Our main generative model choice is Stable Diffusion since it's open-source and has a relatively small memory footprint.

Stable Diffusion is originally trained to be a text-to-image model, which means it takes texts as input to describe the images for the model to generate. In order for us to use it as an image-to-image model, we use a pipeline that is similar but swaps the text encoder with the image encoder from CLIP. Since the goal for training CLIP is to maximize the cosine similarities for the image presentation by the image encoder and the text presentation by the text encoder for matching (image, text) pairs, after the CLIP has been trained, the output presentations (vectors) from it for matching text and image are very similar, thus the swapped-in image encoder in the Stable Diffusion model we use could play the same role of guiding the diffusion model to generate images similar to the input images. The image encoder from CLIP is used to encode the image into a vector, which is then used as image embedding into the diffusion model to guide it during denoising steps, similar to DALL·E 2's image variation generation API. This way, the model appears to generate variations of the given image that are similar to the them.

3.3 Processed Datasets

By running the text-conditioned CLIP-guided Stable Diffusion mentioned in Section 3.2.4 on our raw datasets, we built our main dataset from the in-door dataset in Section 3.1.1 which contains 15,621 real-world in-door photos and combined with another 15,621 generated images that are similar to the real-world photos. Two examples from this dataset are shown in Figure 4. Aside from the main dataset, we also built two smaller datasets from the weather dataset in 3.1.2 and the comic book dataset in 3.1.3 for testing the model's generalization on other genres of photos. For each image in this dataset, we applied center-cropping and resizing postprocessing to make sure they are consistent in

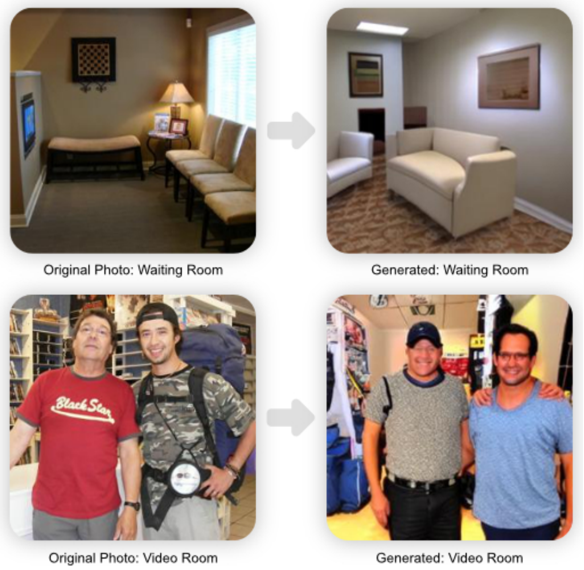

Figure 4: Stable Diffusion Image-to-Image Examples

image size, being 256x256. Apart from that, we also utilized OpenAI's DALL·E 2's image variation generation API to generate two small datasets from a subset of the indoor dataset and the weather dataset, the former containing 625 AI-generated images + 625 real-world photos, and the latter containing 301+301 images.

4 ResNet Classifier

The main task of this project is classification. Over the years, researchers tend to make deeper neural networks(adding more layers) to solve such complex tasks and to also improve the classification accuracy. But, it has been seen that as we go adding on more layers to the neural network, it becomes difficult to train them and the accuracy starts saturating and then degrades also. Therefore, we decided to use **ResNet**, which overcomes the above issue, to do the classification.

ResNet, short for Residual Network, provides another way for data to reach later regions of the neural network by skipping some layers. Consider a series of layers, from layer I to layer $I+n$, and the function F that these layers represent. Layer i's input is denoted by x. x will simply run through these layers one by one in a classic feedforward arrangement, and the result of layer $I+n$ is $F(x)$.

The Resnet conducts element-wise addition $F(x)+x$ after applying identity mapping to x. A residual block or a building block is the term used in literature to describe the entire architecture that takes an input x and creates an output $F(x)+x$. An activation function, such as ReLU applied to $F(x)+x$, is frequently included in a residual block.

The advantage of adding this type of skip connection is that if any layer hurt the performance of architecture then it will be skipped by regularization. So, this results in training a very deep neural network without the problems caused by vanishing/exploding gradient.The training process of a neural network with residual connections has been proven to converge significantly more readily.

5 Experiments

5.1 Experiment Settings

5.1.1 Model

As discussed above, we used ResNet as the classifier model. In order to compare the performance of different ResNet model, we experimented on both **ResNet34** and **ResNet50**.

Resnet-34 involves the insertion of shortcut connections in turning a plain network into its residual network counterpart. In this case, the plain network was inspired by VGG neural networks (VGG-16, VGG-19), with the convolutional networks having 3×3 filters. However, compared to VGGNets, ResNets have fewer filters and lower complexity. The 34-layer ResNet achieves a performance of 3.6 bn FLOPs, compared to 1.8bn FLOPs of smaller 18-layer ResNets.

5.1.2 Dataset

We split around 80% of the Indoor Scene Recognition Dataset with its generated pictures (using Stable Diffusion) for the training part.

5.1.3 Validation

We split around 20% of the Indoor Scene Recognition Dataset with its generated pictures (using Stable Diffusion) for the validation part. The validation data is not used in the training process.

5.1.4 Testing

The testing dataset includes several types of pictures:

Stable Diffusion model: In order to test the generalization ability of our Resnet34 model and to verify it really learns the features of Stable Diffusion generation, not any specific features of Indoor pictures, we use the same method to generate the following dataset which has different pictures contents than Indoor Scene Recognition Dataset:

(1) Weather Image Recognition Dataset: real-world weather pictures with generated pictures.

Dataset	Model	Accuracy	Precision	Recall	F1-Score
Indoor + SD (train)	ResNet34	98.99%	99.07%	98.72%	99.00%
Indoor + SD (train)	ResNet50	98.51%	98.90%	98.10%	98.50%
Weather + SD	ResNet34	90.63%	88.05%	94.02%	90.93%
Weather + SD	ResNet50	95.61%	95.94%	95.25%	95.59%
Comic + SD	ResNet34	92.87%	88.58%	98.44%	93.25%
Comic + SD	ResNet50	98.34%	100.00%	96.68%	98.31%
Small Indoor + DALLE2	ResNet34	51.04%	80.95%	2.72%	5.27%
Small Indoor + DALLE2	ResNet50	51.84%	82.86%	4.65%	8.80%
Weather + DALLE2	ResNet34	50.83%	53.33%	13.33%	21.33%
Weather + DALLE2	ResNet50	50.67%	56.67%	5.67%	10.30%

Table 1: Precision, Recall and F1-Score on the % datasets

(2) Comic Books Images Dataset: comic book images with generated pictures.

Dalle2 model: Besides, we want to test whether the model can distinguish AI pictures generated by other methods. Thus, we choose the same Weather Image Recognition Dataset but generate AI picutres using Dalle2 model to test the performance.

5.2 Evaluation Metrics

To evaluate the performance of the model, we used accuracy, precision, recall and F1-score as evaluation metrics. Below are the formulas for calculating the accuracy, precision, recall and F1-score. Here we assume the AI-generated pictures as positive labels while original pictures as negative labels. Thus, TP is the number of AI pictures being correctly predicted as AI pictures, FP is the number of original pictures being wrongly predicted as AI pictures. Similarly, TN is the number of original pictures being correctly predicted as original pictures, FN is the number of AI pictures being wrongly predicted as original pictures.

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

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

$$Recall = \frac{TP}{TP + FN}$$

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

6 Experiment Results

We experimented with two ResNet models, ResNet34 and XResNet50(ResNet50). We trained both models on our main dataset, which is the Indoor dataset with its generated pictures by Stable Diffusion. After the models have been trained, we tested them on the following datasets: Weather dataset with its generated pictures by Stable Diffusion (Weather + SD), Comic dataset with its generated pictures by Stable Diffusion (Comic + SD), Small Indoor dataset with its generated pictures by Dalle2 (Small Indoor + DALLE2), and Weather dataset with its generated pictures by Dalle2 (Weather + DALLE2). The result metrics including accuracies, precisions, recalls and F1 scores are shown in Table 1.

As shown in the table, on the validation split from the main dataset, the ResNet34 model has a higher accuracy, precision, recall and F1 score than the ResNet50 model. However, on the testing datasets, the ResNet50 model generally has a higher accuracy, precision, recall and F1 score than the ResNet34 model in most comparisons except for "Weather + DALLE2" dataset where ResNet50 is worse in F1 score. Even though all the margins are within 10%, this is still a statistically significant trend that ResNet50 seems to generalize better on the dataset that it hasn't seen. This is likely due to the fact that the ResNet50 model has more layers than the ResNet34 model, which makes it more complex and has more parameters to learn. Thus, it can learn more features from the training dataset and generalize better to the testing datasets. However, the ResNet34 model is simpler and has less parameters to learn. Thus, it is more likely to overfit the training dataset and perform worse on the testing datasets.

The models have shown surprisingly good results with the comic book dataset where both the synthetic images and the original images are not camera-captured photos and also very different from the training dataset in styles and themes. The

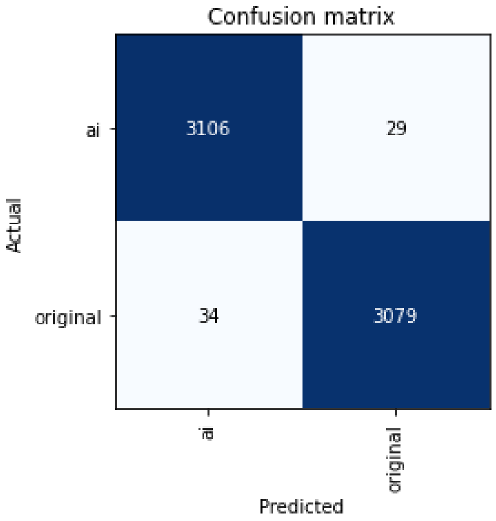

Figure 5: Training Confusion Matrix on Weather

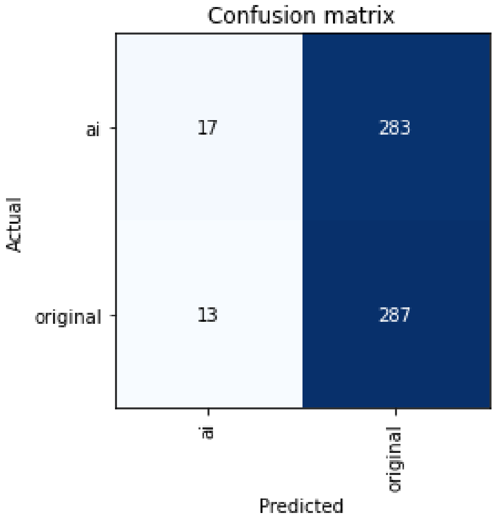

Figure 6: Testing Confusion Matrix on Weather + DALLE2

fact that the models generalize fairly well on this dataset indicates that the model may have learned some general features of the synthetic images that are from the Stable Diffusion model. But the much worse performance from the two models on the DALL·E-2-generated datasets indicates that the features the models have learned are very specific to the Stable Diffusion model and not generalizable to other image-generation models, even when those models are also based on diffusers. To human eyes, DALL·E 2 do generate more realistic and detailed images compared to the Stable Diffusion model. From the comparison of the confusion matrices shown in Figure 5 and Figure 6, we can see that for the validation split for the main dataset that we trained on, the models predicted rather balanced results. However, the models are more likely to predict the DALL·E-2 generated images as original images thinking that they are real photos.

7 Conclusion, Discussion and Future Work

In this article, we contributed:

- We generated labeled datasets with original pictures and AI-generated pictures.
- We trained models with ResNet and SD-generated pictures, which have a high accuracy on detecting the AI pictures generated by Stable Diffusion.
- We found that the pictures generated by Stable Diffusion have significant differences comparing to real pictures. Using the ResNet50 model trained with SD-generated pictures, we have more than 95% accuracy and F1-score on detecting such difference.
- However, both of the models generated by ResNet are not able to detect pictures generated by DALLE2, with nearly 50% accuracy and less than 15% recall.

We think there might be 2 reason for why our model cannot detect any pictures generated by DALLE2. 1) DALLE2 might have a great performance that the DALLE2-generated pictures are very similar to real pictures. 2) DALLE2-generated pictures have different characteristic comparing to the SD-generated pictures.

To verify our result, we can train models with DALLE2-generated pictures and test whether it have a high accuracy on detecting them in the future. This is currently limited by funding, because we didn't get free access to DALLE2.

References

[1] Cenk Bircanoğlu. Comic books images. https://www.kaggle.com/datasets/cenkbircanoglu/comic-books-classification, 2017. Accessed: 2022-12-10.

[2] Jonas Oppenlaender. The creativity of text-to-image generation. In *25th International Academic Mindtrek conference*, pages 192–202, 2022.

[3] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[5] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[7] Haixia Xiao. Weather phenomenon database (WEAPD), 2021.