Get Real: Detector for AI Generated Product Images

15.077 Hands-on Deep Learning

Project Report
Group A25

Authors:
Lucas Goh
Vincent Tian
Scott Morgan
Brenda Ong

Table of Contents

1 Problem Description

1.1 Background

Today, Generative AI models have the capability to produce realistic product images swiftly, significantly accelerating the process of creating scam websites and enhancing social engineering strategies employed by scammers. Despite many AI platforms imposing restrictions on generating marketing material for fraudulent activities, scammers are adept at crafting prompts that bypass these limitations. This enables them to generate highly realistic images for e-commerce fraud, facilitating the rapid setup of entire product pages. Consequently, the ease and speed at which convincing counterfeit product sites can be established are greatly increased, lowering the hurdles for scammers to scale their operations.

In 2023, at least $172.8 million was reported lost to online e-commerce scams globally,[1] with the actual figure likely to be larger due to under-reporting. Scammers can set up fake storefronts to offer free trials or get visitors to make purchases, in order to obtain their personal information during the checkout process.[2] Scammers may order an actual product to be sent to the victim to evade detection, but once victims are lured into providing their financial details on the fake product site, scammers would be able to gain access to their bank accounts.[3] Platforms like e-commerce sites, domain hosts and search engines need to detect such fake sites containing generated images, to prevent them from scamming unsuspecting victims.

1.2 Objective

Our project aims to develop a customized model that can classify product images as real or AI-generated, with a focus on product images that can be used for e-commerce scams. Potentially, our product can be combined with other scam detection tools, e.g. scanning the registered domain owner, the activity of the IP address etc., we will be able to design an effective fake product scam detector.

[1] econsumer.gov, *International Fraud Report 2020-2023*, at: https://public.tableau.com/app/profile/federal.trade.commission/viz/eConsumer/Infographic

[2] Gerrit de Vynck and Alistair Barr, “Canada’s Shifty Battels the Scammers Behind Fake Web Stores,” Bloombery News (9 August 2018), at: https://www.bloomberg.com/news/articles/2018-08-09/canada-s-shopify-battles-the-scammers-behind-fake-web-stores (accessed 2 Mar 2024); FDIC Consumer News, “Avoid Scams While Shopping Online for Bargains,” (Dec 2022), at: https://www.fdic.gov/resources/consumers/consumer-news/2022-12.html (accessed 2 Mar 2024)

[3] Scammers can also take this further to generate product ranges, financial statements, company logos, and set up realistic company pages to lend credibility to investment scams when victims search for the company online. See: Ellen Sheng, “Generative AI financial scammers are getting very good at duping work email,” *CNBC News*, (14 Feb 2024), at: https://www.cnbc.com/2024/02/14/gen-ai-financial-scams-are-getting-very-good-at-duping-work-email.html (accessed 2 Mar 2024).

2 Project Approach

2.1 Data

Our dataset comprises 6,000 images. We randomly sampled 3,000 real product images from the Amazon Berkeley Objects dataset and ran it through Google's Gemini Pro to generate a one-line image caption of the object. These descriptions are then used to generate a "like-for-like" fake product image dataset comprising 3000 images using DALL-E 2, to create balanced classes for training our dataset. In creating our "real" product dataset, we made the effort to select a variety of product types against varying backgrounds. This is so that we are able to train a model on the diverse range of images a GenAI model can produce.

We built a "like-for-like" dataset so the model was trained on the same type of image and be able to pick up features for that particular product type to distinguish between real and fake images, and help us balance the variety of images in the dataset with similar image types in both positive and negative classes.[4] It was also important for us to build a balanced dataset for model training, to boost the model's ability to discriminate between real and fake GenAI images and reduce false negatives - if the training dataset comprises more real than fake images, the model may be skewed towards predicting images as real on unseen data, and we may miss out on flagging out genuine scam sites.

2.2 Challenges

The initial dataset that we had planned to train the model on was the CIFAR-10 dataset, which has 32x32 images. By training the fine-tuned VGG-19 model on a training dataset, we were able to achieve a 96.0% accuracy on the test set if the images were 32 x 32. However, when we fed larger images into the model, the model performed poorly. The 32 x 32 pixel constraint is also unrealistic for the images found on the web today. As such, we decided to build our model on a more realistic training set comprising larger images of 256 x 256 images instead for a real-world applicability. We downloaded the Amazon Product database and filtered the images down to a variety of products with varying zoom levels and backgrounds of size 256 x 256.

The costs associated with generating the images are also significant. It costs ~$50 to generate the 3000 Dalle-E 2 images. While we initially wanted to include Dalle-E 3 images for training, generating Dalle-E 3 images are significantly more expensive. Therefore, we constrained our training set to consist of just Dalle-E 3 images and having 129 of Dalle-E 3 images, which costs $5, to evaluate our model's effectiveness to generalize to newer image models.

[4] Our intention was to build a balanced dataset for training the model to have both the ability to discriminate between a real and GenAI image and reduce false negatives. However, we were constrained by our budget for generating fake images on the GenAI platforms. As such, the training dataset was imbalanced.

2.3 Model Training and Development

The 3,000 real and 3,000 fake images were split into 70% training, 15% test and 15% validation datasets respectively. Our base model was built using transfer learning on the VGG-19 model, with only the output layer trained. The VGG-19 model was selected as it is one of the best performing models for image classification. It comprises 20,155,969 parameters, of which only the final output layers with 131,585 parameters was trainable. We also fine-tuned the base VGG-19 model with all 20,155,969 trainable parameters, and compared its performance against a fine-tuned ResNet50 and EfficientNet models. The model training and development can be found in the Colab here. The base models fine-tuned are:

1. **VGG-19**, which consists of 19 layers (including 16 convolutional layers, 3 fully connected layers, 5 MaxPooling layers and 1 softmax layer), and it is known for its simplicity and depth, but may require more computational resources for training and inference for a larger images.[5]
2. **ResNet50**, which comprises "skip connections" that allow gradients to flow through the network directly without passing through non-linear activation functions, and alleviates the vanishing gradient problem and enable the network to converge faster, allowing for the training for very deep networks.[6] This helps in learning complex and subtle patterns, while improving training efficiency.
3. **EfficientNet series**, which are CNNs that use a compound scaling method to scale network width, depth and resolution. They use Mobile Inverted Bottleneck Convolution (MBConv) blocks in its architecture to focus on the most informative features of the input data, and are designed to achieve superior efficiency in accuracy and computational usage.[7] They are capable of generalizing well on unseen data.

3 Results

The primary metric used for evaluating model performance is accuracy, the percentage of predictions that are correct. The model that exhibited the highest accuracy on the test dataset was the fine-tuned EfficientNetB3 model that ran for 30 epochs. However, the best model that provided the best balance of accuracy, ROC-AUC score, False Negative Rate and Training Time was the ResNet50 model. The table below summarizes the performance for the **various fine-tuned models** on the test data of images generated from DALL-E 2 and Gemma:

[5] Karen Simonyan & Andrew Zisserman (2014), "Very Deep Convolutional Networks for Large-Scale Image Recognition.", arXiv preprint arXiv:1409.1556, available at: https://arxiv.org/abs/1409.1556

[6] ResNet50 paper. Kaiming He., Xaingyu Zhang, Shaoqing Ren and Jian Sun (2016), "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), available at: https://arxiv.org/abs/1512.03385

[7] Mingxing Tan, & Quoc V. Le (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." in *Proceedings of the 36th International Conference on Machine Learning* (ICML 2019), at: https://arxiv.org/abs/1905.11946

Table 3: Model Evaluation on Test dataset

Fine-tuned Model	Accuracy	AUC	False (-)ve Rate[8]	DALL-E 3 accuracy	Training Time (in secs)
VGG-19 model (20 epochs)	95.11%	99.03%	6.62%	24.00%	245
ResNet50 (20 epochs)	**99.00%**	**99.95%**	**0.45%**	**48.84%**	**270**
EfficientNetB0 (20 epochs)	99.22%	99.97%	0.45%	50.39%	293
EfficientNetB3 (20 epochs)	99.00%	99.98%	0.89%	46.51%	515
EfficientNetB7 (20 epochs)	99.44%	99.93%	0.88%	48.06%	1431

Source: Additional details of the models evaluated are in the Appendix. For further details on the model evaluation, see our GetReal Colab.

A closer look also shows that the fine-tuned VGG-19 model may not be as good at discerning the false images from the real images, as the number of False Negatives is still quite high. The confusion matrix for the 900 test set images are as follows:

Figure 3.1: Confusion Matrix for Fine-tuned VGG 19 model over 20 epochs

		Predicted Labels	
		Fake	Real
Actual Labels	Fake	**419** *True Negatives*: Model correctly identifies images as fake	**31** *False Negatives*: Model fails to flag out an image as fake.
	Real	**13** *False Positives*: Model erroneously flags out an image as false when it is actually real.	**437** *True Positives*: Model correctly identifies image as real

[8] False Negative Rate (FNR) or Miss Rate = FN / (FN + TP)

In comparison, although it flags out a few more False Positives, the fine-tuned ResNet50 and EfficientNetB0 models perform much better in discerning False Negatives, which provide the preferred outcome for our business case:

Figure 3.2: Confusion Matrix for Fine-tuned ResNet50 model over 20 epochs

		Predicted Labels	
		Fake	Real
Actual Labels	Fake	**448** *True Negatives*: Model correctly identifies images as fake	**2** *False Negatives*: Model fails to flag out an image as fake.
	Real	**7** *False Positives*: Model erroneously flags out an image as false when it is actually real.	**443** *True Positives*: Model correctly identifies image as real

Figure 3.3: Confusion Matrix for Fine-tuned EfficientNetB0 model over 20 epochs

		Predicted Labels	
		Fake	Real
Actual Labels	Fake	**448** *True Negatives*: Model correctly identifies images as fake	**2** *False Negatives*: Model fails to flag out an image as fake.
	Real	**5** *False Positives*: Model erroneously flags out an image as false when it is actually real.	**445** *True Positives*: Model correctly identifies image as real

Performance on newer image models: To mimic real-world conditions and test our hypothesis on whether the model performs better on older image models, we also created a variety of GenAI images on newer GenAI models like DALL-E 3, which are unseen to our model. Expectedly, our model performs poorly on the DALL-E 3 dataset. We compared the fine-tuned models fine-tuned over 20 epochs for consistency. EfficientNetB0 performs best on the unseen GenAI data, due to its advanced architecture features like the squeeze-and-excitation optimization and MBConv blocks that allow it to learn both coarse and fine features from the data and make them more adaptable to different types of images.[9] Of the 129 DALL-E 3 images tested, the following results were derived:

VGG-19 model:	*24.31%* accuracy (31 correct; 98 incorrectly predicted as real)
ResNet50 model:	*48.43%* accuracy (63 correct; 66 incorrectly predicted as real)
EfficientNetB0:	*50.39%* accuracy (65 correct; 64 incorrectly predicted as real)

This is testament to DALL-E 3's ability to generate convincing images, but also highlights the challenges in updating detection algorithms to keep up with the advancements in GenAI image generation.

4 Lessons Learned

Overall, a ResNet50 model trained on a large variety of image classes, was able to give a good accuracy with fine-tuning. Generally, it provided an excellent performance with a False Positive Rate (FPR)[10] of 1.54% and its True Positive Rate (TPR)[11] of 99.55%. The model performs less well on discerning GenAI images like DALL-3 that it is not trained on, so it would need to be re-trained periodically to incorporate the latest GenAI images so that it is updated to the latest developments in the image generation landscape and remains robust to detecting new trends in image generation.

Platforms are likely to prefer having False Positives over False Negatives as the model would be used to flag out suspicious sites for human review, and it would be better to raise a case for content review subsequently, than to miss out on a scam site. In the real-world setting, we can consider refinements to reduce the False Negatives:

1. ***Lower the decision threshold for classifying positive instances*** (a fake image). This will increase True Positives and False Positives, but we prefer a False Positive over a False Negative for our problem.

2. ***Train a model with more data using a balanced dataset***. Our current dataset comprises 3,000 images of each class as we were constrained by our budget to generate fake images. If we train the model on more images, it would likely improve its overall performance.

3. ***Increase the weight of the positive class*** (fake images) to penalize misclassifying fake images more than misclassifying real ones.

[9] Mingxing Tan, & Quoc V. Le (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks"
[10] False Positive Rate (FPR) = FP / (FP + TN)
[11] True Positive Rate (TPR) = TP / (TP + FN)

In terms of the model architecture, we believe the ResNet architecture is more suitable for our business problem as it performs better in discerning GenAI images from real product images, i.e. high Accuracy, lower False Negatives. ResNet's design allows it to be deeper without suffering vanishing gradient problems, and it is thus able to learn more complex and subtle patterns in images, which is crucial for discerning GenAI images that may have fine details that the model can pick up to to differentiate them from real images. Additionally, the skip connections in ResNet improve training efficiency and allow the network to converge faster, which is preferred when scaling this product to ingest new data or larger datasets. Lastly, ResNet is more adaptable to modifications and can be more easily fine-tuned to our GenAI image discernment tasks.

5 Further Improvements & Business Applications

5.1 Further Improvements

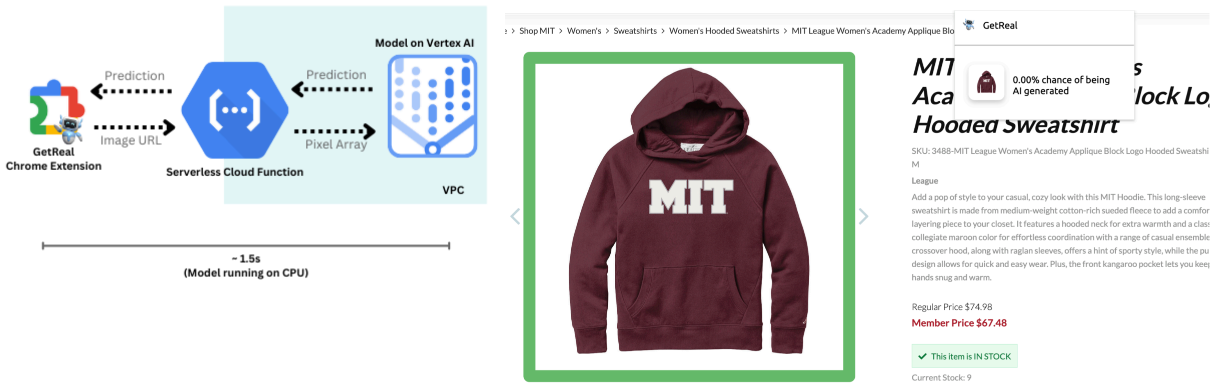
The high accuracy of the fine-tuned ResNet model indicates strong potential to further improve model performance with additional training data and fine-tuning. The accuracy of the model in detecting and flagging out a scam product site can be enhanced if we combine the fake product image detection model with other components common to fake product sites - such as GenAI marketing copy, IP addresses linked to fake sites, GenAI code detectors for the site and frequency of website maintenance activity - and there is potential to create a more accurate and useful scam website, to make the online shopping experience safe and trusted.

The current model is trained on "like-for-like" images, based on product descriptions of products that exist today. Scam websites may be created for products that have yet to be invented or are new to consumers. While the accuracy on such predictions is untested, we tried to simulate this with testing on the unseen DALL-E 3 dataset. The model does not perform well, but we believe it can be updated easily to account for such trends - we can use GenAI to ideate on prompts for possible new product types, and then use the latest image generators to produce the accompanying GenAI images to train our model on. Given that there are no existing "real" product types for this image to train the model on, we theorize that we are also less likely to get a positive prediction on these new, yet-to-be created image types.

5.2 GetReal Chrome Extension

To create real-world utility to users and to demonstrate the applicability of an application like this, we created a Google Chrome extension called "GetReal" to allow other users to test this model while they are doing their online shopping. The extension allows users to click on an image while shopping online to provide a probability that the image is AI generated. This also allowed us to demonstrate how a model like this would be served in the real world to generate online predictions. To do this, we hosted our final model on Google Cloud's Vertex AI, a model serving framework. We wrote a small serverless function that accepts an image URL as input, fetches the image and converts it into a 256 x 256 x 3 array, makes a request to our hosted model to get a prediction, and then returns this prediction back to the client. The system diagram and a screenshot of the Chrome extension in action are shown in Figure 5.1 below.

Figure 5.1: Get Real Chrome Extension incorporating model


Model on Vertex AI
Prediction
Prediction
Image URL
Pixel Array
GetReal Chrome Extension
Serverless Cloud Function
VPC
~ 1.5s (Model running on CPU)
Shop MIT > Women's > Sweatshirts > Women's Hooded Sweatshirts > MIT League Women's Academy Applique Blo
GetReal
MIT
Aca
lock Lo
Hooded Sweatshirt

0.00% chance of being AI generated

SKU: 3488-MIT League Women's Academy Applique Block Logo Hooded Sweatshi
M

League

Add a pop of style to your casual, cozy look with this MIT Hoodie. This long-sleeve sweatshirt is made from medium-weight cotton-rich sueded fleece to add a comfor layering piece to your closet. It features a hooded neck for extra warmth and a clas collegiate maroon color for effortless coordination with a range of casual ensemble crossover hood, along with raglan sleeves, offers a hint of sporty style, while the pu design allows for quick and easy wear. Plus, the front kangaroo pocket lets you kee hands snug and warm.

Regular Price $74.98
Member Price $67.48

✓ This item is IN STOCK

Current Stock: 9

6 Conclusion

The project tests one detection aspect of a scam site, and if combined with other suspicious indicators of a fake product site, we will be able to build a powerful fake product site detection engine that can be applied on: (i) e-commerce platforms to detect and remove fake product pages; (ii) search engines to block results to these sites; and (iii) domain hosting platforms to detect fakes sites which use their platform. With the application of the model alongside other in-house scam detection algorithms, companies like Shopify, Etsy, eBay, GoDaddy, bluehost, Google or Bing can better detect and prevent such sites from operating and scamming victims. In the immediate term, we have created a Google Chrome extension for users to apply to product images they come across online, if they need to determine if those images are real or fake.

With additional improvements to reduce False Negatives and in combination with other suspicious site indicators, our model has the potential to severely disrupt scammers' ability to scale e-commerce scam operations, and enhance consumer trust, safety, and the overall online shopping environment on the platforms.

7 Appendix

7.1 Details of Model Evaluation

The findings from the models trained and their training accuracy, validation accuracy, test accuracy and Test AUC are detailed below. The model with the best balance of the metrics is the ResNet50 model:

Results of model evaluation on various datasets

Model	Epochs	Training Accuracy	Validation Accuracy	Test Accuracy	Dall-E-3 Accuracy	Test AUC	Training* time (secs)
VGG19	10	0.9836	0.9356	0.9400	0.1550	0.9916	139
	20	0.9883	0.9289	0.9511	0.2400	0.9903	245
	30	0.9955	0.9289	0.9356	0.1163	0.9945	365
Resnet50	10	0.9979	0.9822	0.9878	0.5736	0.9994	150
	15	0.9990	0.9990	0.9889	0.4729	0.9991	210
	20	**1**	**0.9878**	**0.9900**	**0.4884**	**0.9995**	**270**
	40	0.9993	0.9867	0.9889	0.4806	0.9990	510
EffnetB0	10	0.9962	0.9867	0.9778	0.4729	0.9986	169
	20	0.9986	0.9878	0.9922	0.5039	0.9997	293
	30	0.9974	0.9889	0.9889	0.6047	0.9996	424
	40	0.9986	0.9933	0.9922	0.5039	0.9997	553
EffnetB3	10	0.9931	0.9789	0.9733	0.4341	0.9948	289
	20	0.9967	0.9900	0.9900	0.4651	0.9998	515
	30	0.9993	0.9993	0.9967	0.4884	0.9999	745
EffnetB7	10	0.9988	0.9911	0.9911	0.4651	0.9993	163
	20	0.9943	0.9889	0.9944	0.4806	0.9993	1431
	30	0.9976	0.9856	0.9956	0.4574	0.9997	2075

*on A100 GPU