A Naive CNN Application to AI Generated Image Detection

Wesley Jones *Iowa State University*
CPRE Digital Forensics, 536
Ames, US
wesleyj@iastate.edu

***Abstract*—Detecting generated images is an increasingly prevalent need in forensics. Convolutional neural networks (CNN) are a model of AI training that predates the more popular usage of transformers in popular language models today (e.g., GPT-1 and newer). I will continue to expand the application of CNN models to image assessment by training a model that contains two categories: real and synthetic human faces. This model will then be utilized to assess a variety of images generated from various methods (unknown to the trained model) of image generation. By asses a naive implementation (one without in-depth filtering or other image contexts), I am attempting to establish the feasibility of a long-term solution to more simple pre-trained models that can be used to AI generated image detection. This simplification is made possible by both high quality datasets of real and synthetic images for training, as well as higher quality images for testing said model. One model utilized for generating testing images performed at chance (50%), the other achieved a 100% prediction rate. These scores could indicate that the issue that is causing one model to fail predictions might be causing another to be overly successful. More research into these types of consequences is needed. The current research of applying CNN models to generative image identification is focused on filters and sequential analysis, but if issues presented in this naive approach are worked out without relying on those approaches, a more robust path forward for quicker and more robust identification could become clear.**

***Index Terms*—artificial intelligence, convolutional neural networks, images, models, forensics, detection**

I. Introduction

CONTENT GENERATED by artificial intelligence (AI) includes text, pictures, sound, and videos. These can be wholly created utilizing models of AI intended for the purpose of generation or be a composition of mixed mediums: both "hand-crafted" by a human and augmented utilizing generative methods. All media benefits from relevant provenance – of course, this metadata should be accurate. Relying on human capabilities to decipher the trustworthiness of media metadata is not a feasible long term solution in the face of continually evolving generative AI models.

It is worthwhile to *constantly* explore methodologies and their feasibility for generated media detection. With each new iteration of generative techniques, various components will be vulnerable to detection. In the game of generative AI whack-a-mole, a winner does not survive long.

A. About Convolutional Neural Networks

Convolutional neural networks (CNN) are a model of AI training that predates the more popular usage of transformers in popular language models today (e.g., GPT-1 and newer). Networks trained with CNN models are capable of informing generative adversarial networks (GAN) – the predecessor to GPT-based generative methods commonplace today. To properly generate an image from text, AI relies on a deep set of feature labels, preferably so deep they would have only been generated using a neural network — CNN commonly. With these labels the appropriate approximation of an image can be built out, different models will vary during the generative process, but these models often relying on an existing feature list. A CNN operates as a layer within a neural network to apply a filter on an image. This filter works to determine whether a filter is met, first in a small part of the selected pixels, then scaling out. This granularity can lead to specific filters returning successfully [1], [2].

Reviewing the advantages of the CNN's granularity as well as it's "hands-off" approach to filtering this particular model should be an effective means of quickly detecting a synthetic image from a real image. This process should not rely on itemizing characteristics that individual generative models are vulnerable to, rather it should focus on a binary filter system in the CNN layer: is the image real or synthetic? To test the feasibility of a broad application of this method I will set up a simple CNN model filtered on a set of real and synthetic images. I will test this naive approach by providing a variety of potential real or fake images from different models, qualities, and compositions (e.g., synthetic components to an otherwise real image).

B. Objectives

I will continue to expand the application of CNN models to image assessment by training a model that contains two categories: real and synthetic human faces. This model will then be utilized to assess a variety of images generated from various methods (unknown to the trained model) of image generation. By assessing a naive implementation (one without in-depth filtering or other image contexts), I am attempting to establish the feasibility of a long-term solution to more simple pre-trained models that can be used to AI generated image detection. This simplification is made possible by both high quality datasets of real and synthetic images for training, as well as higher quality images for testing said model.

II. Related Works

I will walk through a few basic models of image generation to provide context for the choice of synthetic comparisons I provide during the testing phase.

A. Methods of Text to Image Generation

As discussed in the CNN background, a core element of each of these methods is the data set that depends on an accurately labelled list of features within sample images to build a trained model off of. In this way, a very simple explanation of all generative models would involve first the definition of features and second a process akin to splashing color at a variety of canvases until the third step could be passed, the evaluation of whether this image meets the requirements put forth.

GAN is an extension of a neural network implementation. The process for generating an image with a GAN works by generating noise (incorrect) images while simultaneously evaluating "real-ness". As this evaluation progresses a threshold is met and enough data has been added to a potential image that it could be construed as fulfilling the request. GANs rely on a few models to generate images, all neural network based [2].

Another method came from the development of transformers [3] as a means to compute AI models. An existing technology called discrete variational autoencoder (dVAE) was applied in a more efficient manner which allowed Ramesh et al. [4] to create a large enough set of transformers to give a pre-trained AI model a selection of qualified images to choose from that would accurate represent the prompt.

Arguably the lion's share of text to image generation today is coming from the diffusion models. This process solves for common limitations of traditional GANs such as downsampling (fuzzy or unclear components). In 2022, a group from CompVis [5] launched Stable Diffusion [6], a collaboration borne out of their research into diffusion models and Stability AI [7]. Diffusion models involve some convolutional processing, but do not build a network that previous GAN models required to generate images [8].

B. Methods of Image Detection

A straightforward approach to predicting the real-ness of an image is to identify vulnerabilities in the generative capabilities of the originating model. Amerini et al. [9] analyzed deepfake videos for frame-to-frame indicators. They looked specific at optical flow, a vector measurable across the frames that could create a quantifiable indicator of whether a video was real or fake. Their evaluation determined that facial features provided a reliable result — enough to justify continued research. In this paper, the analysis is restricted to images, but specifically facial features. This work is continued by Nassif et al. [10] who evaluated the efficiency of this method with modern hardware. They found optical flow to be a reliable indicator and were able to provide statistics for models' accuracy based on training time.

In 2019, Wang et al. [11] performed a study on CNN generated images that displayed an ability to detect images as they classified "easily". The authors relied on a generative model (ProGAN implemented with CNN) to synthesize fake images from a real source. They compared these in a binary process and searched for evidence of fingerprinting, or a reliable method that would offer a clear indicator that an image has been generated by a CNN model. They determined that their accuracy was reliable enough that CNN generation processes should be considered identifiable, at least in the short run. A shortcoming of this study is the narrowness of the fake image data set. Relying on exact originals to the faked images might make the outcomes unreliable when applied to an image that the neural network has never seen before.

This paper's approach is motivated by a similar process laid out in [12]. Hulzebosch et al. focus on "real world" applications. Their techniques avoids the short comings of a potentially limited data set and adds a human identification factor. They focus results on taking an image from an unknown model and also an unknown post-processing model. This research includes an element of human trials — where individuals are asked to identify a synthetic image from a group of 18 potential images. The researchers provide quantitative results indicating that human-based judgement is unreliable. Their weighted algorithmic performance (controlled across benchmarks and studies) only performed slightly better than humans.

Bird and Lotfi [13] used Stable Diffusion 1.4 (similar to here) to create synthetic images to test with. These researchers analyzed the CNN classification process using Explainable AI — a technique created to identify *why* an AI handled data the way it did. They used this to optimize their analyzing. The image dataset was a selection of random categories from their own generated dataset and analyzed at a resolution of 512x512. Their results show 92% accuracy.

III. Methodology

I am opting out of image pre-processing techniques aside from the required process to load data into a tensor for modelling. Based previous readings, I will also utilize a larger training image size (1024x1024). The final methodology that I am relying to set my results apart is utilizing a better selection of "real world" images both for training and for testing. In this case, a real world image is one that could be plausibly found online as passing for a person.

The process to test my naive implementation is as follows:

1) Identify testing images of real faces and previously generated fake faces. Train a CNN model using these.
2) Run and save the model. This process timing is dependent on the amount of images being analyzed.
3) Create images to test with. These are high quality generated images that would typically require scrutiny to identify as AI generated.
4) Load the saved model and gather prediction results.

A. Model Training Images

For real images, I am relying on the same common library utilized in many resources to-date, the Flickr Faces High Quality (FFHQ) Dataset [14] . It was generated by Nvidia

Purpose	*Image Count*	*Classes*
Training	112,952	2
Test	16,136	2
Validation	32,273	2

TABLE I: Image counts for stages of model training.

Labs for their own research into machine learning. This dataset has been made available for download and contains 70,000 images at 1024x1024 resolution of faces scraped from the Flickr website. FFHQ contains 70,000 images.

For synthetic images, I found a dataset offered that combined a common source of faces generated utilizing StyleGAN, but this set has been updated using Stable Diffusion 1.4 by Beniaguev through an extensive process (at no point were real images of faces utilized to create this dataset) [15]. Ideally this should better prepare the trained CNN on potential unknown model generated images. This dataset contains 92,151 images.

The total images utilized in training is 162,151, comparable to Bird and Lotfi's study of CIFAKE [13].

B. Hardware

The CNN model was trained with Nvidia A100 Tesla card in a high performance computer cluster. The training process required 14.45 hours and benefited from only negligible attention granted to optimization due to approaching this topic as a naive effort. Once the model is generated and saved, predictions can occur on indiscriminate hardware so long as the necessary Python packages are loaded and the CPU has enough power to provide predictions in a reasonable amount of time.

C. Naive CNN Model

The CNN model is built using Python and TensorFlow [16]. The model is trained using the Keras method [17] and uses the Adam optimizer. Two labels are defined: "real" and "synth". Three datasets per label ("real" and "synth") are defined, one for training, validation, and testing each. The trained model is saved as a `keras` object that the prediction query is evaluated with. The reported accuracy during the training from a subset of validation real and synthetic images was $> 99\%$. The training was preset to run over 15 epochs, but would have been sufficiently training starting around 10, see Fig. 1. For counts of the images used in each stage of building the model see Table I. More extensive training would have been necessary if more than two parameters were specified, as evidenced by [12]. For counts of the images used in each stage of building the model see Table I.

Once the run was complete, I saved the model using a feature of the `keras` library. This is loaded later to predict using – resulting in a very fast turnaround (prediction results averaged around 175 milliseconds per image on low frequency server CPU). The saved model size is around 18 megabytes.

D. Prompts for Generating Test Images

Through some research and proof-of-concept testing, a I determined a generic prompt to generate professional head shots, seen in Fig 2. I wrote a small program[1]. Since the breakdown of the training libraries was purported to be diverse [14], [15], I needed to verify with a diverse set of high quality face images. Both of the models discussed in sub section III-E. The selected genders and races are provided in table II. The produced results from both models are high quality, but the content relating to the prompt tells another story. Some of the concern images can be see in the provided GitHub repository,[1] but I will lay out the more obvious issues here.

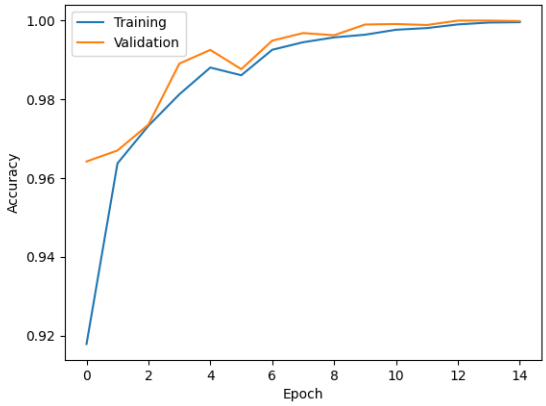


Fig. 1: Accuracy rates during model training over 15 epochs.

- Both models produced the same type of result for a prompt asking for "Native American". In this case, the models (both SDXL Turbo and Fast SDXL) are relying on the same parameters encoded in the underlying Stable Diffusion generation [18]. This means that the prompt generates a similar output, with the main difference between them relying on filters applied during post-generation processes. To this end: the cultural insensitive results *persist* across models. Refer to Figure 3 for examples. These same concerning results occurred through other prompts (notably again in the African man generation – where the shoulders are bare).
- Both models *rarely* generated either androgynous or masculine presenting faces when prompted for a "person". I had expected this to be a more equitable – tossup – between the more presumably common categories of "man" and "woman". I could not find a prompt that generate a cisgendered face, or even one that could be construed as such. There's clearly a categorization limitation in these models when it comes to gender identity.

The models have warnings about their applicability to reality [18]. In this case, the designers are correct to warn the users of these models that the results are a representative generation of the provided prompt. The Fast SDXL implementation of the model provided four responses per request. For these I simply saved all images and only ran the generalized "person" attribute instead of using "woman" and "man". This resulted in a total of 49 images to verify my CNN model.

[1]See the Jupyter notebook in my code base for this process https://github.com/iamwpj/naive-cnn-identifier

```
{race} {gender} looking directly at the camera
for a professional headshot taken using a
Sony A7 III camera with 1/250, f/1.4,
ISO 200 - FE 35mm 1.4 ZA - Portrait Style
and 6200 K
```

Fig. 2: Prompt for generating test images from listed AI models. The bracketed values (`race`, `gender`) are rotated to generate various race and gendered faces.

Genders	*Races*
Man	African
Woman	Asian
Person	Black
	Latinx
	Middle Eastern
	Native American
	White

TABLE II: List of genders and races itemized per prompt.

E. Models Generating Test Images

To generated test images I relied on popular technologies available to the public for either "beta" testing or as part of their licensing. Both technologies are continuations of the diffusion process by Stability AI [6] – SDXL [18]. SDXL enhances the underlying diffused images during the generative process to allow the model the ability to generate at a finer grain than previous training of Stable Diffusion model versions. As discussed above in section III-D, there are inherent issues with the faces produced in these models.

Both datasets were quick, averaging a result (or four) from the input prompt within seconds (typically less than 10). See Table III for the counts from each model.

1) SDXL Turbo: This model has been enhanced from the open model SDXL by Stability AI[2], generated on the hosting partner Clipdrop[3], utilizing the same underlying data as the first version, but optimized to provide low latency generation, even updating as the prompt is input. While the output was high quality in many ways that AI face generation has traditionally suffered, this speedup causes all generated faces to appear similar with each prompt, see Fig 4. The generated images from this model are watermarked. This was an initial concern, but I opted to wait for results and if I determined this marking had an adverse affect on the output (i.e. made it too easy for the model to predict it as synthetic), I would seek a better solution.

The images supplied by this modeled offered in a resolution of 512x512. Since the tensors expect images of 1024x1024, some pre-processing to adjust sizing is preformed during the prediction testing.

[2]https://huggingface.co/stabilityai/sdxl-turbo
[3]https://clipdrop.co/stable-diffusion-turbo

Model	*Count*
Turbo SDXL	21
Fast SDXL	28

TABLE III: Count of test images generated using each model

(a) "Native American woman" *via SDXL Turbo*

(b) "Native American person" *via Fast SDXL*

Fig. 3: The cultural representation of Native Americans presents the inherent biases in this categorization of human faces. In the process of generating faces specific to an indigenous group, the results are restricted the biases encoded in the model's parameters.

2) Fast SDXL: Fast SDXL[4] is based on the same core model provided in SDXL Turbo [19], [20]. This iteration instead benefits from the post-processing efficiencies of a new hardware generation of tensor processing units. The implication of the announcements for both the Fast SDXL and Google's TPUv5e is that inferences during the diffusion process should improve the resulting image quality. It's not clear that either party expects a better image in relation to the prompt.

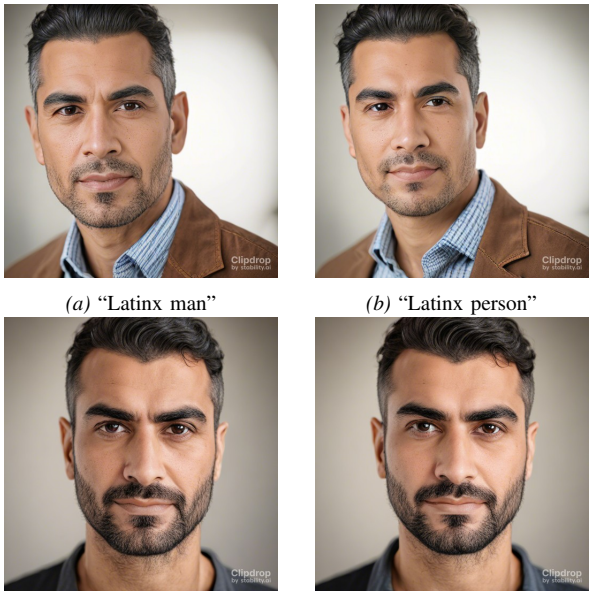
The images supplied by this model are offered at 1024x1024 and required no pre-processing.

IV. Results

I supplied 49 images (Table III) to the model for confirmation. Predictions in this model are made as a value between 0 and 1. I set up my output to provide an easy interpretation of anything over 0.5 to determine as *real* and anything under that determine as *synthetic*. The results of this run showed that a naive model is too unpredictable to provide any reliable role in the detection process aside from being a first-stage filter. Of the 49 images provided 25 were correctly predicted, and 24 were incorrectly predicted. Review accumulated results in Tables 5 and

Typical predictions were recorded confidently by the model – regardless of the accuracy. The raw data can be viewed on GitHub [5]. The majority of correct predictions occurred utilizing the test images created by Fast SDXL

[4]https://huggingface.co/spaces/google/sdxl
[5]https://github.com/iamwpj/naive-cnn-identifier/blob/main/stats.ipynb



(a) "Latinx man" *(b)* "Latinx person"

(c) "Middle Eastern man" *(d)* "Middle Eastern person"

Fig. 4: Collection of similar faces generated from SDXL-Turbo. Notice how (a) and (b) are nearly identical? There are small aberrations between each generation (see the light in the eyes, for example), but these are minute. The same result is true for prompt of (c) and (d).

III-E2. These are the images that relied on up-scaling techniques to match the expected tensors. This modification must have played a significant role in the reliability of output, but was not uniform. The Native American selections (`na_[man,woman,person].jpg`) were all flagged as *real*.

V. Discussion

A model that detects 50% correct should not immediately be dismissed. This matches an experience by Wang et al. [11], where they sought to isolate *shallow* methods of generated images. However in this case, there seems to be a processing flaw. After consulting the results, I preformed a series of predictions using known images (e.g., images I had taken) and attempted to estimate most likely cause of error. These tests performed closer what a "production" detector would expect. Of the eight provided images six were successfully identified as real or synthetic. Ultimately such a small sample size combined with the inclusion of two images *also* included in the model training data, makes these results irrelevant to any bigger picture conclusions.

Taking this improvised testing into consideration, I will outline a couple of issues with the larger image tests prediction outcomes.

A. Resizing

A defining difference between the test image data is sizing. All of the images supplied in 512x512 format were incorrectly predicted. The process undertaken to perform a resizing should have resulted in a lower quality image, but not one that was appreciably different in regards to how the CNN model would see it. The adjusted prediction images can be seen in GitHub[6]. To soften any approach the tensor resize function[7] might have exerted on image. I tested a run relying on general image library in Python (Pillow[8]) to resize and one with the default tensor library resizing. Neither changed the output predictions.

Conclusively determining the resizing process was a culprit contributing to mispredictions has been challenging. There are outliers in both my one-off (see Table IV and the aforementioned results). There is also the case of the three images from the otherwise correctly predicted library — the Native American faces from the SDXL Turbo model. These photos features significant aberrations from the trained model images, items such as feathers, extensive jewelry, and headbands, review the complicated examples in Figure 3.

In 2019, Monasse detailed the adjustments that occur in an image affected by a bilinear filter [21]. The likelihood that a shortage of pixels would cause this filter to upscale the image in a way that affects the prediction seems poor. If it was the case that pixel-by-pixel adjustments led to cascading failures the results might have had less confidence in the predictions. I removed the most confident results and isolated > 0.1 to < 0.99 in Figure 7. This chart isolates 9 of the 49, $> 20\%$ of the results are less than 99% confident, but only the Fast SDXL model had to undergo resizing. Isolated to the model of interest, Fast SDXL, this is 9 of 21, or just over 40% of the total are less confident.

Resizing could have play a role in these incorrect predictions, but it is likely only a factor in the big picture of utilizing a naive model for AI generated content detection.

B. Reliance on binary choice

The confidence of the model's predictions regarding Turbo SDXL is commendable, but likely will not scale beyond a narrow domain. This is because my trained CNN model is restricted to a binary choice by the input parameters. Common research in this field on filters [11], [12] or in the case of video, variations in an image sequence [22] to establish a non-binary choice. Adding filter provides layers — a composition of the prediction result would allow for measuring the output from each layer in order to draw a conclusion. The increased data offers the design more flexibility. In a case sequence of filters it is possible that the positive identification could be a smaller or broader range of the average returned by the filters.

In my CNN model, there is little opportunity to shift these numbers to tailor for the accuracy of the domain specific identification of a face. For example, moving the positive identification of a synthetic image to be < 0.33 does not significantly change the classification of my results (it affects none of the model's predictions). Because of the passive nature

[6]An example of an image provided at 1024x1024 and thus left untouched: https://github.com/iamwpj/naive-cnn-identifier/blob/main/predict_modified/asian_person.jpg. An example of an image initially loaded at 512x512 and then modified: https://github.com/iamwpj/naive-cnn-identifier/blob/main/predict_modified/asian_person_1.jpg

[7]https://www.tensorflow.org/api_docs/python/tf/image/resize – This function applies a filter to either downscale or upscale an image. The default filter is bilinear.

[8]https://python-pillow.org

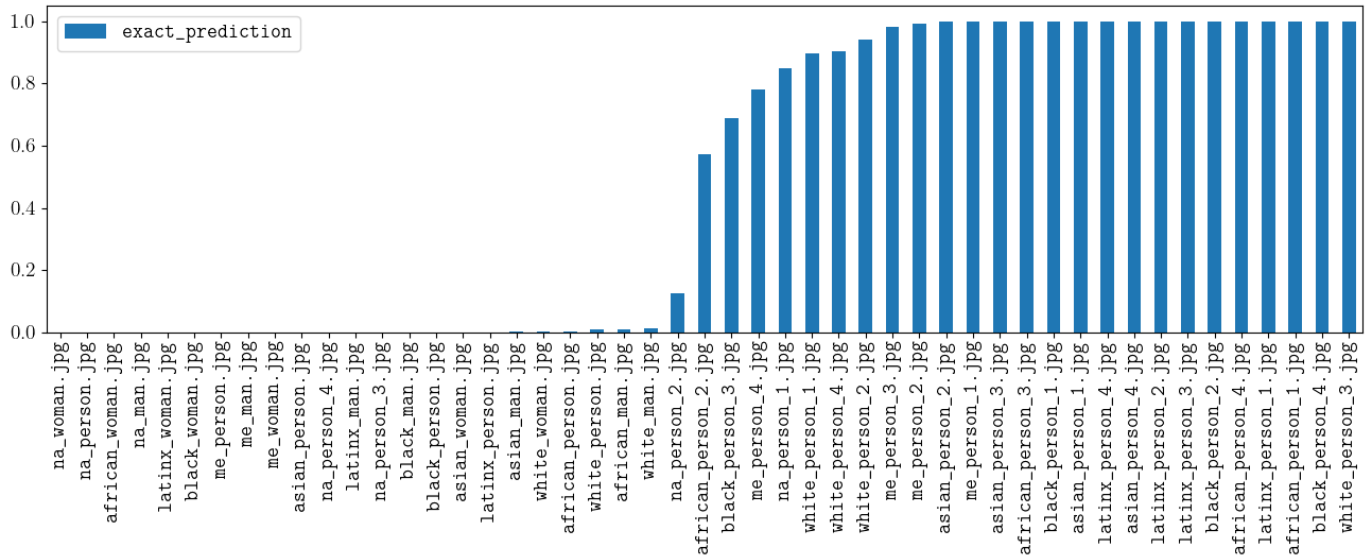


Fig. 5: Test prediction results overview. Notice the similarities in the exact prediction results. If the model determines an image is real or synthetic it is very confident in this result.

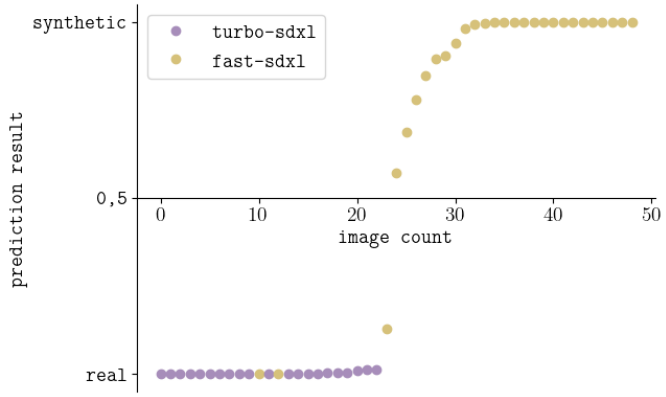


Fig. 6: This plot highlights the performance of each model. This is not an incremental model - the x-axis indicates the index number of an image, a counter that increases as subsequent images are tested. The x-axis indicates the dividing line. Prediction values < 0.5 are determined to be real. A perfect test would result in all predictions being > 0.5.

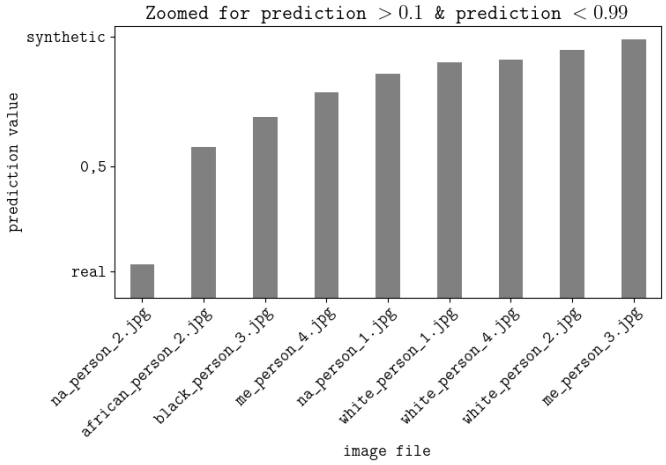


Fig. 7: Selected results focused on the middling predictions. Anything falling below o.5 is considered a real image by the prediction.

of feature abstraction in CNN it is well suited to be paired with enhancements to produce more granular results [1].

VI. Conclusion

A. Future Work

There is ample opportunity for tuning the CNN model training and prediction architecture. With a reliable set of model settings and better performance optimizations this model could be utilized as a "just in time" solution — where an image can be quickly sent to verification and the result returned as a user interacts with the resource.

As pointed out in Section V-B, there is room to identify methods of adding variables to the prediction confidence values. Such additions might look like:

- Positive identifications of non-domain subjects in the image. Using a sequence of publicly available object identifiers this method could lower the confidence of a false "real" prediction.
- General purpose fake image detection CNN models could also offer an adjustment to the confidence score. This is *similar* to the filter, however, since it relies on pre-trained models instead of just-in-time processing it should be more salable.
- The CNN model trained relied on high quality input images, but these are common in all models trained, potentially even in models trained to generate images. A study of the effect of this source pollution would be wise as these models continue to distillate. Even in this research one shared source model produced two generative models with significantly different prediction output scores. Alzubaidi et al. provides some context here — a deep CNN (DCNN) would extend to not only a deeper learning, but can also be a way to train source

Image Source	*Expected Prediction*	*Actual Prediction*	*Exact Prediction*	*Dimensions*
morgan_freeman-wikipedia.jpg	real	real	0.0000000000	3006x2253
burt_reynolds-wikipedia.jpg	real	real	0.0000042566	675x460
asian_woman-openjourney-v4.jpg	synthetic	real	0.0095512047	512x512
me_person-stabilityai-SD-XL-1.0.jpg	synthetic	synthetic	1.0000000000	1024x1024
latinx_person-dalle-mini.jpg	synthetic	real	0.0001257265	1024x1024
me.jpg	real	real	0.0128671583	985x1103
litmus_real.png	real	real	0.0000000001	1024x1024
litmus_synthetic.jpg	synthetic	synthetic	1.0000000000	1024x1024

TABLE IV: I performed a series of tests to verify results against known image sources and with various degrees of sizing. All of these images are disparate sizes, with cropping/resizing occurring in cases where the dimension do not match the expected size of 1024x1024. Image sources indicate where the data originated from. To easily decipher results, I have highlighted the matches in green and the mismatches in red.

models more efficiently [1]. By training many models and then connecting them together you can scale horizontally for more efficient training time and ideally provide more informed prediction scores.

Ultimately the deciding factor on utilizing CNN modelling technology for detecting generated images depends on processing of input images, either through filters or additional context. My results show a chance result across the board, but become more promising if the testing images sourcing is subdivided by model used to generate the images. One model remains poor — Fast SDXL. The other achieved a 100% prediction rate. These scores could indicate that the issue that is causing one model to fail predictions might be causing another to be overly successful. More research into these types of consequences is needed.

The current research of applying CNN models to generative image identification is focused on filters and sequential analysis, but if issues presented in this naive approach are worked out without relying on those approaches, a more robust path forward for quicker and more robust identification could become clear.

References

[1] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, Mar. 2021. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8

[2] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent Progress on Generative Adversarial Networks (GANs): A Survey," *IEEE Access*, vol. 7, pp. 36 322–36 333, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8667290/

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 2017, publisher: arXiv Version Number: 7. [Online]. Available: https://arxiv.org/abs/1706.03762

[4] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-Shot Text-to-Image Generation," 2021, publisher: arXiv Version Number: 2. [Online]. Available: https://arxiv.org/abs/2102.12092

[5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Apr. 2022, arXiv:2112.10752 [cs]. [Online]. Available: http://arxiv.org/abs/2112.10752

[6] "Stable Diffusion," Nov. 2023, original-date: 2022-08-10T14:36:44Z. [Online]. Available: https://github.com/CompVis/stable-diffusion

[7] "About." [Online]. Available: https://stability.ai/about

[8] R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva, "Intriguing Properties of Synthetic Images: From Generative Adversarial Networks to Diffusion Models," 2023, pp. 973–982. [Online]. Available: https://openaccess.thecvf.com

[9] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake Video Detection through Optical Flow Based CNN," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 1205–1207. [Online]. Available: https://ieeexplore.ieee.org/document/9022558/

[10] A. B. Nassif, Q. Nasir, M. A. Talib, and O. M. Gouda, "Improved Optical Flow Estimation Method for Deepfake Videos," *Sensors*, vol. 22, no. 7, p. 2500, Mar. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/7/2500

[11] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot... for now," 2019, publisher: arXiv Version Number: 2. [Online]. Available: https://arxiv.org/abs/1912.11035

[12] N. Hulzebosch, S. Ibrahimi, and M. Worring, "Detecting CNN-Generated Facial Images in Real-World Scenarios," *CoRR*, 2020, publisher: arXiv Version Number: 1. [Online]. Available: https://arxiv.org/abs/2005.05632

[13] J. J. Bird and A. Lotfi, "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images," Mar. 2023, arXiv:2303.14126 [cs]. [Online]. Available: http://arxiv.org/abs/2303.14126

[14] "NVlabs/ffhq-dataset," Oct. 2023, original-date: 2019-02-04T15:35:08Z. [Online]. Available: https://github.com/NVlabs/ffhq-dataset

[15] David Beniaguev, "Synthetic Faces High Quality (SFHQ) part 2." [Online]. Available: https://www.kaggle.com/dsv/4737578

[16] T. Developers, "TensorFlow," Nov. 2023. [Online]. Available: https://zenodo.org/doi/10.5281/zenodo.4724125

[17] F. Chollet and others, "Keras," 2015. [Online]. Available: https://keras.io

[18] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis," 2023, publisher: arXiv Version Number: 1. [Online]. Available: https://arxiv.org/abs/2307.01952

[19] "Accelerating Stable Diffusion XL Inference with JAX on Cloud TPU v5e." [Online]. Available: https://huggingface.co/blog/sdxl_jax

[20] "Announcing Cloud TPU v5e and A3 GPUs in GA." [Online]. Available: https://cloud.google.com/blog/products/compute/announcing-cloud-tpu-v5e-and-a3-gpus-in-ga

[21] P. Monasse, "Extraction of the Level Lines of a Bilinear Image," *Image Processing On Line*, vol. 9, pp. 205–219, Aug. 2019. [Online]. Available: https://www.ipol.im/pub/art/2019/269/?utm_source=doi

[22] J. Yang, S. Xiao, A. Li, G. Lan, and H. Wang, "Detecting fake images by identifying potential texture difference," *Future Generation Computer Systems*, vol. 125, pp. 127–135, Dec. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X21002387