Evaluating Deep Learning Methods for Detection of AI Generated Images: A Study on GenImage

Isabela Iacob and Emilia-Maria Nuță

Babeș-Bolyai University, Cluj-Napoca, România

Abstract. In a world full of images and data, where artificial intelligence is now more powerful than ever and can generate complex and lifelike digital media, it is essential to maintain a well-defined line between real content and AI-generated creations. We can fight the misinformation where images generated by AI are used in malicious ways by training different AI models to identify and classify the real and the generated images. For this experiment, we can use different machine learning models to classify the fake and real images. Therefore, these authors propose a comparison between three machine learning techniques aimed at classifying images: a custom Convolutional Neural Network (CNN), a Residual Neural Network (ResNet), and a ConvNext architecture.
The custom CNN architecture relies on multiple convolutional layers to extract hierarchical features from the images, pooling layers for dimensionality reduction, and fully connected layers for final classification. On the other hand, the ResNet introduces residual connections, or skip connections, allowing the network to mitigate the vanishing gradient problem and train deeper architectures effectively. ResNet is particularly well-suited for capturing complex patterns in image data by preserving information across layers.
Our attention will be focused on the GenImage dataset, a large dataset containing both AI-generated images and real photographs, labeled as 'Fake' and 'Real'. We will present the final results of both architectures alongside the methodology and experiments conducted.
These experiments are crucial in an era dominated by artificial intelligence to maintain ethical and secure use of media across the internet and to address potential future legal implications.

Keywords: Computer Vision · Deep Learning · Generative AI

1 Introduction

Recently, the field of synthetic image generation through artificial intelligence (AI) has evolved rapidly, creating a critical need to detect these images to ensure authenticity and veracity. As AI-generated content becomes more sophisticated, detecting synthetic images will become increasingly challenging, posing significant risks in fields such as law, where the authenticity of visual evidence could influence the outcome of a case. This issue is particularly crucial when determining the veracity of images used in legal proceedings, where misidentification of AI-generated content could have serious consequences for justice.

Therefore, our research proposes the application and optimization of well-established deep learning architectures to detect AI-generated images. Additionally, we employ adversarial attacks on these architectures to assess their robustness and effectiveness in real-world scenarios, where attempts to deceive detection systems are becoming more prevalent.

We also take into consideration the fast development that the Transformers architecture brought into the machine-learning world, by leveraging the architectures inspired by the transformer's architecture and design, such as networks from the ConvNext family, that are now widely used in machine-learning tasks, especially classification.

The paper is structured into five main sections (excluding the introduction). Section 2 reviews the related work on detecting AI-generated images, as this area has become essential for ongoing research and development. We'll provide related work for the pre-trained architectures that we used, ResNet and ConvNext families. Even though they serve the same purpose and can be used in the completion of the same task, they are fundamentally different, which we will discuss in the Methodology section. Section 3 talks about the dataset used in the study: the GenImage dataset, a widely recognized collection of both AI-generated and real images. There are images from multiple sources and contain generated images and nature pictures. Section 4 outlines the methodology used in our experiments, detailing the deep learning models applied and the approach for testing their robustness through adversarial attacks. In Section 5, we present the results obtained from our experiments, followed by an in-depth discussion of their implications. Finally, Section 6 concludes the paper, summarizing the key findings and suggesting potential future research directions in this rapidly advancing field.

The goal of our study is to compare how well each model performs in classifying images. We'll use standard metrics to measure their performance, such as accuracy, precision, and F1-score, to compare the results of the three techniques.

2 Related work

Existing detection techniques. The accurate detection of AI-generated images is paramount, and as such there exist several deep learning approaches for this task. More prevalent are learning-based methods. Wang et al. [9] use a ResNet-50 architecture pre-trained with the ImageNet as a classifier and train it in a binary classification setting using a ProGAN-generated dataset. They find that the CNN model could generalize well in the detection of other GAN-generated images. While research suggests that learning-based methods are viable for this task, Ojha et al. [7] show that real-vs-fake image classification models trained on a specific generative model have limited generalizability to other generative models and that the learned features are biased towards recognizing patterns from one class disproportionately better than the other. Other works [4,10] present that AI-generated images have unique traces (called fingerprints) that depend on the architecture and training characteristics. Our work

tries to train neural networks on data generated by different generative models and assess their performance.

AI generated images. The field of AI-generated images has rapidly evolved in the last years, mainly with the use of GANs(Generative Adversarial Networks) and DMs (Diffusion Models). Although the mentioned approaches are still used and provide very good results, thanks to the transformers architecture we are now able to have models that reach capabilities that were not seen before. For example, the DALL-E 2 model provided by OpenAI is a transformer-based model that combines transformer architectures with diffusion models. It uses a combination of two techniques: CLIP (Contrasive Language-Image Pre-training), which connects text and images by understanding the semantic relationship between them, and diffusion models. DALL-E 2 uses random noise in the beginning, and it continues to refine the image until it produces a high-quality image, guided by a text prompt. [?] It's important to mention that even though transformers played a huge role in the development of today's AI application, other approaches involve the use of more classic techniques, that are still very important and used. For example, the autoregressive models generate images pixel-by-pixel in a sequential manner, and one pixel depends on the previously generated one. It can generate very qualitative images for small datasets, by capturing strong dependencies between the pixel. One of these models, for example, is PixelCNN which uses convolutional layers for different applications, such as generating new portraits of the same person with different facial expressions or lighting conditions. (Sursa: https://arxiv.org/abs/1606.05328)

In this study, we focus on this type of AI-generated images to determine whether different learning-based methods categorize them as fake or not, by combining and comparing classical ML techniques to more recent ones.

3 Dataset

In this study, we use the GenImage dataset [11], a comprehensive resource comprising over 2.6 million images, including 1.35 million AI-generated and 1.33 million real images. The dataset leverages 1,000 distinct labels from the ImageNet database, ensuring a diverse range of image categories that span various subjects such as animals, objects, and scenes. The image generation process for the GenImage dataset employs several state-of-the-art generative models. These include diffusion models such as Midjourney [5], Stable Diffusion V1.4 [8] and V1.5 [8], GLIDE [6], VQDM [3] and ADM [2], as well as GANs like BigGAN [1]. Since the original dataset proposed by the authors is very big in size, we use a tinier version of it available on Kaggle[1]. In the Kaggle version, just 5000 images are kept for 7 of the aforementioned generative models (Stable Diffusion V1.4 is excluded). For each model, data is divided into train (4000 images) and validation (1000 images), divided further into ai (2000 images for train and 500 images

[1] https://www.kaggle.com/datasets/yangsangtai/tiny-genimage

for validation) and nature (2000 images for train and 500 images for validation). It is not mentioned how many of the 1,000 labels are kept, but we will continue in a binary classification fashion, either 'nature' or 'ai'.

Fig. 1. Sample images from the GenImage dataset

Augmentation.

4 Methodology

In this paper, we will address the task of classifying images into two categories: images that an AI model and real images generated. For this experiment, we employed a comprehensive approach that involves three different models: a model that uses a custom Convolutional Neural Network (CNN), a pre-trained ResNet model, and a ConvNextSmall network. In this section, we will provide details about the dataset preparation, the network architectures we used, and the evaluation metrics that we used to measure the performance of the three models.

Data preparation As mentioned in a previous section, we used the GenImage dataset that contains several state-of-the-art generative models. The images were separated into different folders, with a training and a validation subfolder, both having two categories: ai and nature. We used all the training and validation images from all the folders, for better training. The images were pre-processed to maintain compatibility with the models, especially with the pre-trained networks, and to standardize input dimensions across the models. We didn't use any data augmentation techniques for this experiment. MODIFICAM DUPA DACA FACEM

Hardware and software resources The library employed for the detection of AI synthetic generated images was Keras (maybe link??). For reproducibility purposes, we set the seed to 42. All algorithms in this study were run on Google Colab resources.

Model Architectures As mentioned, for this task, we employed three model architectures: a custom CNN designed specifically for the classification task, a ResNet model fine-tuned from pre-trained weights, and a ConvNextSmall network adapted for binary classification.

The custom CNN used in this task was designed as a lightweight model, suitable for image classification tasks. We implemented this architecture using TensorFlow and Keras, and it consists of convolutional, pooling, and dense layers, organized to facilitate effective classification. The model uses images of dimensions 32x32x3, and the process begins with three convolutional layers. We use the ReLU(Rectified Linear Unit) activation, padding "Same", that ensures the output dimensions remain consistent with the input, and max-pooling operation, which reduces the spatial dimensions by half, to retain the most salient features. After the convolutional and pooling operations, the output feature maps are flattened into a one-dimensional vector, to prepare them for the dense layers, which consist of 128 neurons with ReLU activation and a dropout layer with a rate of 0.5, to mitigate overfitting by randomly deactivating half of the neurons during training. The output layer is a dense layer with neurons equal to the number of classes (2 classes) and it utilizes a SoftMax activation function. These techniques provide class probabilities for the binary classification task. The model is compiled using an Adam optimizer and we use a categorical cross-entropy loss function.

Resnet50 is a deep CNN architecture that introduced residual learning, a concept that was proposed to solve the vanishing gradient problem in very deep networks. It was developed as part of the ResNet family, and this specific version (ResNet50) became popular and a widely used architecture, especially in image classification. When it comes to fundamental elements, this architecture uses residual blocks, which introduce a shortcut connection, allowing the network to learn residual mappings instead of deep mappings. ResNet50 consists of 50 layers, 48 convolutional layers, 1 max-pooling, and 1 average pooling layer. Another important aspect of this architecture is its parameter number, approximately 25.6 million, a modest number compared to other architectures from the ResNet family. Moreover, the use of bottleneck blocks makes ResNet50 efficient, without losing accuracy points. For our experiment, the model was initialized with ImageNet weights to capitalize on pre-learned features, this dataset having over a million images across 1000 categories. This way, our model has a strong baseline of generalized features instead of learning from scratch, like in the first example. The top classification layer of ResNet50 was removed, so we can better adapt the network for our task, to append custom layers for our classification task. Now, the model accepts inputs of 224x224x3, consistent with the ResNet50 architecture. We wanted to ensure that the model's baseline remained intact, meaning that we wanted the model to keep the generic features. Therefore, we froze the convolutional layers of the base model. This way, the base layers will not be updated during training, saving a lot of computational costs and risk regarding overfitting. A custom classification head is appended to the base model, to fine-tune the base model for our task, a binary classification, consisting of

several elements: global average pooling, a fully connected layer, which is essentially a dense layer with 256 neurons, and ReLU activation, a dropout layer, with the same scope as the one used in the first experiment with the custom network, where we deactivated half of the neurons and an output layer, a dense layer with a single neuron and sigmoid activation, where the output values closer to 0 indicate AI-generated images, and values closer to 1, real photos. The model was compiled using binary cross-entropy as a loss function, as an optimizer we used Adam, and as a default metric, we selected accuracy.

ConvNext is a modern architecture, inspired by the design of transformer-based vision models. It was developed by Liu et al. (2022) when the paper "A ConvNet for the 2020s" appeared for the first time and introduced the new generation of models. To summarize, ConvNext tries to combine CNN's performance with the improvements brought by transformers. ConvNextSmall is a lighter version of the ConvNext family, that adopts design choices from the Swin Transformer (Liu et al, 2021), a multi-scale transformer architecture, that is well-designed for high-resolution images in computer vision. The implementation of the two architectures, using the Keras API and the processes that allow us to use the pre-trained network for a specific classification task, is quite similar, but the two architectures are fundamentally different. We used the GELU(Gaussian Error Linear Unit) activation function, which is very commonly used in Transformers. Gelu function weights inputs by their value and uses a probabilistic approach, using the Gaussian cumulative distribution function. In contrast, the ReLU function outputs the input directly if it is positive and 0 if it is not. While ReLU is much easier to compute, GELU works perfectly in reducing the changes in gradients, making the optimization more stable. Another important aspect is related to the building of blocks. If ReSnet50 used Residual BottleNeck Blocks, for ConvNextSmall we have Simplified ConvNext Blocks, modern blocks that use both CNN and transformer design principles.

5 Results and discussion

6 Conclusion

References

1. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis (2019), https://arxiv.org/abs/1809.11096
2. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: Advances in Neural Information Processing Systems. vol. 34, pp. 8780–8794 (2021), https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf
3. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10686–10696 (2022). https://doi.org/10.1109/CVPR52688.2022.01043

4. Marra, F., Gragnaniello, D., Verdoliva, L., Poggi, G.: Do gans leave artificial fingerprints? In: 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). pp. 506–511 (2019). https://doi.org/10.1109/MIPR.2019.00103
5. Midjourney: (2022), https://www.midjourney.com/home
6. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., Chen, M.: GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In: Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 16784–16804. PMLR (17-23 Jul 2022), https://proceedings.mlr.press/v162/nichol22a.html
7. Ojha, U., Li, Y., Lee, Y.J.: Towards universal fake image detectors that generalize across generative models. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 24480–24489 (2023). https://doi.org/10.1109/CVPR52729.2023.02345
8. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models . In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10674–10685. IEEE Computer Society (2022). https://doi.org/10.1109/CVPR52688.2022.01042, https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042
9. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot...for now. In: CVPR (2020)
10. Yu, N., Davis, L., Fritz, M.: Attributing fake images to gans: Learning and analyzing gan fingerprints. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7555–7565 (2019). https://doi.org/10.1109/ICCV.2019.00765
11. Zhu, M., Chen, H., YAN, Q., Huang, X., Lin, G., Li, W., Tu, Z., Hu, H., Hu, J., Wang, Y.: Genimage: A million-scale benchmark for detecting ai-generated image. In: Advances in Neural Information Processing Systems. vol. 36, pp. 77771–77782 (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/f4d4a021f9051a6c18183b059117e8b5-Paper-Datasets_and_Benchmarks.pdf