AI-Generated Anime Figure Detection

Yi'an Pei & Xiaotong Zha

Duke Kunshan University
COMPSCI 309
May 2024

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: Machine Learning · AI-Generated Content · AI Detection · Anime Images.

1 Introduction

AI-generated content (AIGC), which denotes the production or creation of content through artificial intelligence systems, has exerted a significant influence across various application scenarios, particularly with the emergence of large-scale models. The content produced by generative AI models comprises text, image, audio, video, and cross-modal transformations such as text-to-image and text-to-audio. Among these modalities, text-to-image conversion has garnered enormous attention in social media platforms, primarily owing to its potential to supplant human visual design. In this field, several representative models have emerged, one such model being DALL•E, which is trained using Variational Autoencoders, while another important model is VQGAN-CLIP based on Generative Adversarial Networks (GANs). Nowadays, the diffusion model has become the core approach in text-to-image generation, with popular models like Stable Diffusion, Disco Diffusion, MidJourney, and DALL•E2 etc. Users can cultivate realistic or artistic images significantly correlative with the prompt by typing the prompt containing expected image concepts, attributes, and styles.

Previous work in this domain has seen the utilization of familiar classification techniques prevalent in computer vision, such as ResNet, and EfficientNet, to discern between AI-generated and non-AI-generated images. Typically, the entire image serves as the input for such classification tasks. However, we are intrigued by the possibility of exploring specific regions within an image for this purpose. Particularly, we focus on anime picture generation, where the output of anime heads tends to be of higher quality compared to other regions, which may contain errors. This raises the question of which region of the image holds the most discriminatory information. In the context of anime picture generation, focusing on the quality of the generated anime heads could potentially enhance the accuracy of AI-generated image classification.

2 Background

2.1 YOLO

You Only Look Once (YOLO) is a famous machine learning model in field of object detection which is first introduced by Redmon et al. in 2015 [6]. This model applies advanced algorithms, increasing its speed and accuracy in analyzing images, which often plays an important role in real-time object detection.

Previous models usually need to look through the original picture several times, to find the box containing objects and decide the class of objects respectively. Different from those models, YOLO does not need to scan the figure multiple times. It solves the problem in view of a single regression problem directly from image pixels when bounding box coordinates and class probabilities at the same time, which ensures the process to be fast as it finishes all its works in one single pass. To be a tradeoff, the accuracy of YOLO is usually lower, but still generally high enough to be available, making it to be an overall better model to work on.

After several years of improvement, YOLO now has many different versions and branches for various scenarios. YOLOv5-anime is one of those versions, trained and specialized for anime style. This model can highlight the parts considered as "heads" by boxes and corresponding scores, similar to what other version of YOLO can do, while it is expected to have a better performance when dealing with databases like anime images.

2.2 Backbone

CNNs Convolutional Neural Networks (CNNs) have emerged as the cornerstone of many computer vision tasks, owing to their ability to effectively extract hierarchical features from images [1]. CNNs are composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers [2]. These layers collectively learn to identify patterns and features in images, enabling tasks such as object detection, classification, and segmentation. CNN architectures typically consist of a series of convolutional layers followed by pooling layers, which reduce the spatial dimensions of the feature maps while preserving important information [3]. The convolutional layers apply filters to the input image, extracting features such as edges, textures, and shapes [4]. These features are then passed through activation functions to introduce non-linearity, enhancing the model's capacity to capture complex patterns. One of the pioneering CNN architectures is AlexNet [5], which achieved breakthrough performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. Since then, numerous CNN architectures have been proposed, each offering improvements in terms of depth, width, and computational efficiency. Examples include VGGNet [6], GoogLeNet [7], ResNet [8], and EfficientNet [9].

Transformers Transformers have revolutionized various fields of artificial intelligence, particularly natural language processing (NLP), but their applicability has extended to computer vision tasks as well [10]. Originally introduced for sequence-to-sequence learning tasks, transformers have demonstrated remarkable success in capturing long-range dependencies and contextual information [11]. Unlike CNNs, which rely on convolutions for feature extraction, transformers utilize self-attention mechanisms to weigh the importance of different input tokens [12]. This mechanism allows transformers to capture global dependencies in the input sequence, making them well-suited for tasks that require understanding relationships between distant elements. In computer vision, transformers have been adapted for tasks such as image classification, object detection, and image generation. Notably, models like Vision Transformer (ViT) [13] have shown competitive performance on image classification benchmarks such as ImageNet. By treating images as sequences of patches, ViT applies transformer architectures to extract features and make predictions. Additionally, transformer-based models like DALL•E [14] and CLIP [15] have demonstrated impressive capabilities in generating and understanding images by leveraging text-based prompts. These models rely on the fusion of vision and language modalities, showcasing the versatility and power of transformer architectures in computer vision tasks.

3 Methods

3.1 Data Source

To form a database of anime images, we collected illustrations from the website called pixiv. Pixiv is an online community and platform centered around digital artwork with mostly anime style. This dataset is regarded to be different from those of most previous researches because anime figures have no biological limitations. This provides possibility of existing "unreal" components in non-AI images, which have potential to be a misdirection for regular detection models.

All the data is downloaded from pixiv website directly through chrome extensions. Only illustrations with clear mark of "AI" or "non-AI" were collected and separated into corresponding folders. Besides, images with less than 10 likes were ignored in case there are too many extreme outliers.

3.2 YOLOv5-anime

YOLOv5-anime is a variant of YOLO targeted at detecting and localizing anime-style characters within anime images and videos. This model will output the original image with highlighted boxes where containing objects considered as faces. It had a good performance at detection faces of anime characters since it is trained in similar situation.

In this project we decided to trust the model and applied a trained weight to establish a database with the help of YOLOv5-anime. All imaged from pixiv will be tested by this model to check whether there are character involved.

3.3 Architecture

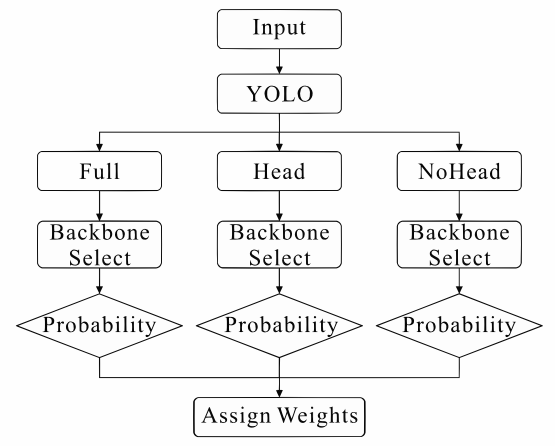

Fig. 1: Designed architecture using YOLO and Backbone Select

In our project, we aimed at combining YOLO and some of the backbones to create a better model on detecting AI-Generated anime images (see Fig. 3). Considering the common habit of focus on tags related to faces when generating AI anime pictures, we created an architecture to test the performance of dataset with, without, or even only head and made an ensemble accordingly.

For every image as an input, YOLOv5-anime will be applied. If YOLO successfully detected some faces, the image will be used to form three kinds of pictures: original, head-only and head-free. Head-only represents the cut of the highest face detected, and in the folder of head-free all faces are blocked in blank.

Afterwards, all these three kinds of data will be trained independently by different backbones. For each of these three folders, one model will be found to be the best backbone, and they will be combined to give out the finally prediction of AI or not.

The combination of backbones was regarded as a classification problem, where the confidence of an image to be AI by the three models forms three features. The weights was obtained from logistic regression, attempting to output 1 for AI and 0 for human during the iterations.

3.4 Backbone Select

We compared the performance of the following five backbones to select the best for each YOLO output category.

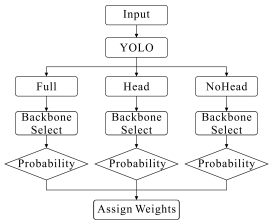

Fig. 2: Designed architecture using YOLO and Backbone Select

AlexNet A pioneering architecture in CNNs, AlexNet's eight-layer design, featuring rectified linear units (ReLUs) and overlapping max-pooling, achieved groundbreaking success in the ImageNet competition. In this study, AlexNet serves as a benchmark for evaluating subsequent architectures.

ResNet50 Addressing the challenge of vanishing gradients, ResNet50 introduced residual connections, enabling training of exceptionally deep networks. Its use of residual blocks enhances feature learning and prevents overfitting, making it valuable for capturing intricate patterns.

EfficientNet V1 Revolutionizing neural architecture design, EfficientNet V1 scales network width, depth, and resolution systematically to achieve state-of-the-art performance while maintaining computational efficiency. Its balance between model complexity and cost ensures high accuracy within resource constraints.

EfficientNet V2 An evolution of EfficientNet V1, V2 incorporates novel improvements like refined scaling methods and advanced training techniques. It surpasses previous benchmarks while maintaining computational efficiency, making it ideal for cutting-edge performance within resource constraints.

SWIN Transformer A novel approach tailored for visual recognition tasks, the Swin Transformer divides high-resolution images into non-overlapping patches, facilitating efficient processing and capturing long-range dependencies. Its hierarchical design and self-attention mechanisms align well with the objectives of our research, making it a suitable choice for analyzing complex visual patterns in anime images.

3.5 Evaluation Metrics

True Positives (TP) True Positives represent the number of correctly predicted positive observations.

True Negatives (TN) True Negatives represent the number of correctly predicted negative observations.

False Positives (FP) False Positives represent the number of incorrectly predicted positive observations.

False Negatives (FN) False Negatives represent the number of incorrectly predicted negative observations.

Accuracy Accuracy is the ratio of correctly predicted observations to the total observations. It measures the overall correctness of the model.

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

Precision Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It measures the accuracy of positive predictions.

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

Recall Recall (also known as sensitivity) is the ratio of correctly predicted positive observations to all the observations in the actual class. It measures the ability of the model to find all the relevant cases within a dataset.

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

Specificity Specificity (also known as the true negative rate) measures the proportion of actual negatives that are correctly identified.

$$Specificity = \frac{TN}{TN + FP}$$

F1 Score F1 Score is the weighted average of Precision and Recall. It considers both false positives and false negatives. F1 Score is best if there is some sort of balance between Precision and Recall.

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

4 Results

Backbones After weighing the pros and cons, we finally chose EffcientNet V2 to be the backbone for original full images for its good performance after several iterations. For head only dataset, EffcientNet V1 was applied for its coexistence of high accuracy and stability in low iteration. Although it seems to perform worse than EfficientNet V2 in the last several epoch, the overall effect of EfficientNet V1 in this dataset is worth considering. Furthermore, we picked EffcientNet V2 to apply to no head images for its outstanding performance of the best epoch.

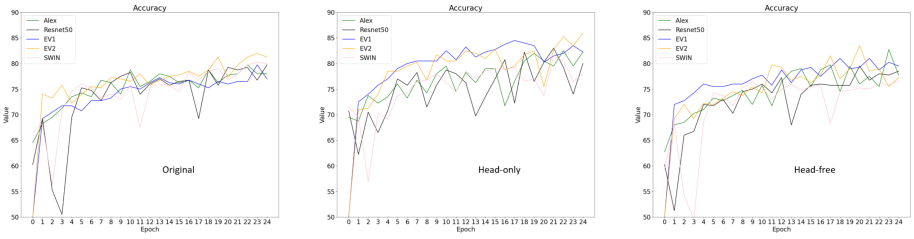

Fig. 3: Performance of different models through epochs

Weights The trained weights can be seen in the following picture. It was noticed that the head-only model weighted nearly two times of others, verifying the feasibility of testing a picture with respective of faces.

Table 1: Weights of three models for different type of datasets.

Dataset	**Model**	**Weight**
Original	EffcientNet V2	0.2931150682739578
Head-only	EffcientNet V1	0.4396615789134529
Head-free	EffcientNet V2	0.26722335281258935

Model Performance It can be seen in the following figure and charts that our model have a much higher performance on the datasets, forming a new way to detect AI-generated images.

Table 2: Performance Metrics of Different Models

Model	**Accuracy**	**Precision**	**Recall**	**Specificity**	**F1 Score**
AlexNet	0.818	0.827	0.800	0.835	0.813
ResNet50	0.815	0.820	0.803	0.826	0.811
EfficientNetV1	0.840	0.809	0.885	0.795	0.845
EfficientNetV2	0.833	0.808	0.870	0.796	0.838
SWIN	0.832	0.821	0.847	0.818	0.834
Our Model	0.873	0.882	0.867	0.867	0.874

Table 3: Performance Metrics of Our Model

Class	**Accuracy**	**Precision**	**Recall**	**Specificity**	**F1 Score**
AI	0.873	0.882	0.867	0.867	0.874
Human	0.873	0.865	0.880	0.880	0.872

References

1. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, http://www.springer.com/lncs, last accessed 2023/10/25
6. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).