Contrastive Feature Extraction for Synthetic Image Detection

Calvin D.H. Leighton
University of Technology Sydney
Sydney, Australia
Calvin.d.leighton@student.uts.edu.au
ORCID: https://orcid.org/0009-0005-1004-5442

***Abstract*—Contrastive feature extraction has shown to have improved generalisability when differentiating between classes. The application of this onto the CIFAKE-10 dataset to distinguish between authentic and generated images (using Stable-Diffusion-1.4) has shown to have improved F1-scores and accuracy when compared to alternative supervised approaches including convolutional neural networks and vision-transformers. Supervised contrastive learning should be the basis for future image detection and especially in the detection of artificially generated images. All code for this research can be found here: https://github.com/nekovin/SupervisedContrastiveLearning_CIFAKE10; data is publicly available and can be found here: https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images.**

Keywords—Contrastive Feature Extraction, Synthetic Images, CIFAKE-10

I. Introduction

Any social-media platform is ripe for the spread of misinformation and trickery. In turn, synthetic images are becoming far more capable and deceptive (see Fig 1 for example). Not only that, but by August 2023, roughly 15.470 billion images were generated from DALL-E 2, Stable-Diffusion models, Adobe Firefly and Midjourney alone [1]. Therefore, the real-world application of reliable synthetic image detection models needs to be scalable and accurate, and generalisable.

Papers have explored synthetic image detection and contrastive feature learning [2] however, I believe that there is not enough literature on the exploration of how contrastive learning is a better option than traditional feature learning. Not only this, but the CIFAKE-10 dataset (a comprehensive dataset with 50,000 real and artificial images) has not been thoroughly (or at all) investigated with contrastive feature learning. Supervised contrastive learning has shown to have generalisable capabilities on discriminative tasks [3] and thus warrants more research and exploration in synthetic image detection. I apply a supervised contrastive feature extraction method for just this.

Fig 1: **Deceptive synthetic image**. A highly convincing synthetic image which circulated popular social media platforms. One giveaway if you look closely is that the street-lights are not correct.

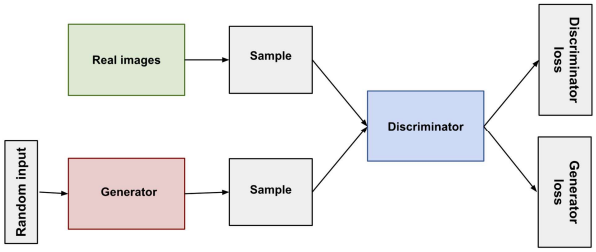


Fig 2: **Depiction of a GAN**. This entails training the discriminator to distinguish the generator's created samples and real image samples.

II. Background and Related Work

A. Generative Models and Synthetic Images

Generative models have the capabilities to generate high quality, artificial content, seemingly becoming unlimited in what they can create. Diffusion models [4] and Generative Adversarial Networks (GANs) [5] are some of the most prominent in artificial generation which are capable of generating such images.

Generative Adversarial Networks (GANs) are among the most popular and successful methods for generating synthetic images. GANs consist of two neural networks, a generator and a discriminator, trained in an adversarial manner (as shown in Fig 2). The generator tries to produce realistic synthetic images, while the discriminator tries to distinguish between real and synthetic images. Through this adversarial training process, the generator learns to capture the distribution of real images and generate highly realistic synthetic images.

Expanding upon this was the introduction of diffusion models. Diffusion probabilistic models, better known as diffusion models, are just variational Markov chains that learn transitions through reverse diffusion processes where each transition adds noise to the image until it has minimised the loss to a sufficient degree (see Fig 3 for visualisation). Simply, the steps in this are:

1. The forward diffusion process: The Markov chain gradually adds Gaussian noise to the images until the data is turned into pure Gaussian noise.

2. The reverse denoising process: A neural network learns to reverse the initial step by denoising, iteratively minimising the predicted noise.
3. The sampling process: Iteratively apply denoising to generate new data samples.

This was how the synthetic images for this project have been developed. The images which have been developed by Stable-Diff-1.4, a prominent and powerful diffusion model, capable of generating convincing synthetic images.

B. *Contrastive Learning*

Contrastive learning has shown greater potential in generalisable classification tasks. To implement contrastive feature extraction, an encoding step and a classifying step is required.

The encoding step aims to extract the feature vectors of the input images. It achieves this by passing the image into the pre-trained feature extractor to then flatten and pass into a projection head to reduce the dimensionality into a more 'digestible' batch. During training, the output of the encoder is then passed into a contrastive loss function in order to reduce the error between two feature vectors. Once the encoder is trained, the trained model is passed into a feed-forward neural network for classification training.

Contrastive learning comes down to learning the normalized embedded versions of the classes (see Fig 4 for visualisation). The encoder uses supervised contrastive feature extraction. A pretrained encoder has been used in order to extract the features of the pre-processed images, then uses the supervised contrastive loss function in order to create similar output vectors which are similar to similar classes [3].

$$\sum_{i=1}^{N} \frac{-1}{N_{y_i}-1} \sum_{j=1}^{N} 1_{i\neq i} 1_{j\neq j} \log \frac{\exp\left(\frac{z_i * z_j}{temp}\right)}{\sum_{k=1}^{N} 1_{i\neq k} \exp\left(\frac{z_i * z_k}{temp}\right)} \quad (1)$$

The supervised loss function (1) is comprised of many working parts:

- y_i is the ith sample
- N represents the total number of samples
- N_{y_i} is the number of samples with the same class with the i-th class
- $\frac{-1}{N_{y_i}-1}$ normalises the samples
- $\sum_{j=1}^{N} 1_{i\neq i} 1_{j\neq j}$ is essentially saying "do not consider the same sample with itself"
- $temp$ is manually set in hyperparameter tuning. Higher values mean that the cutoff between the different classes is higher; a lower temperature is the inverse.
- $\log \frac{\exp\left(\frac{z_i * z_j}{temp}\right)}{\sum_{k=1}^{N} 1_{i\neq k} \exp\left(\frac{z_i * z_k}{temp}\right)}$ is measuring the similarity, having higher values for same class prediction and lower values for different class predictions.

This is paramount in the extraction of features and learning in a contrastive manner.

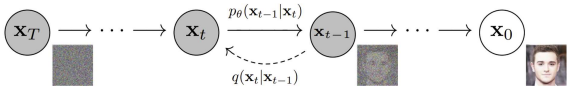


Fig 3: **Diffusion probabilistic Markov chain process**. This fairly basic diagram depicts the denoising process, where the model learns to denoise a seemingly nonsense, visually sporadic image, into a comprehensive one.

C. *Synthetic Image Detection*

Deep convolutional neural networks have paved the road for image classification tasks and have shown their effectiveness by [6] and has continued to be one of the most effective approaches. The classification of real and fake images, however, is not this simple- because these images are seemingly infinite, constantly being fed new training data and generating unique images, a simple deep classification network by itself may not prove to have the best results.

Various efforts have been made in order to address the issue of generative image detection and protection. Watermark technology [7] [8] was explored for this research, however, I believe this is not an effective approach. The proposed framework aims to embed watermarks into the GAN model while preserving its performance, enabling remote ownership verification for IP protection. I believe the continued alteration of these images further nullifies any sort of reliance we could have- what I mean by this is an real image could be labelled with this invisible watermark then passed off as fake which causes more problems if applied at scale.

More strides to address this challenge were made with [9] by introducing the Synthbuster model, a simple classifier that takes advantage of the residual noise left behind in diffused images in order to distinguish real and synthetic images. This was a promising approach and definitely warrants further, future research in tandem with contrastive learning.

Generalisable contrastive learning and has been done before [2] which introduces language-guided contrastive learning. I aim to contribute to this research field by demonstrating the immediate applicability of computationally effective and efficient a contrastive learning model, purely based on the features of the image.

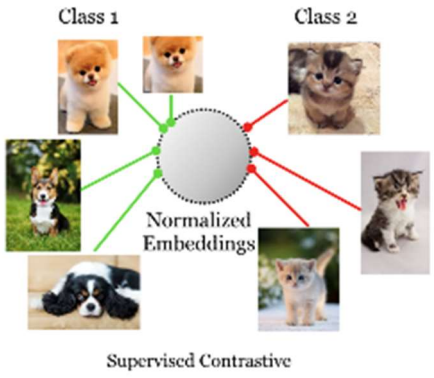


Fig 4: **Contrastive learning example.** Embeds the learned features to then differentiate classes.

III. Methodology

A. CIFAKE-10 Dataset

CIFAKE-10 is a dataset which is comprised of 100,000 images, half of which are real where the remaining are fake. This dataset contains 10 classes which contain 5000 real images, and 5000 fake images which were generated using Stable-Diffusion-1.4 [10]

B. Data Preprocessing

The dataset was pre-processed into 32x32 pixel RGB images and then normalized using standard normalization for pretrained models (mean being (0.485, 0.456, 0.406), and the standard deviation being (0.229, 0.224, 0.225) over the three RGB image channels). No other transformations or augmentations were made. Fig 5 (a) shows an example of a 32x32 image and (b) a 2912x1632 pixel image; the second is obviously of higher quality and would potentially have more capacity to capture more details, however, the computational cost would increase exceptionally.

All 50,000 pre-processed images were split into a 80/10/10 training/validation/test sets, which were then packaged into batches of 128.

C. Encoder Model Architecture

ResNet-50 was used as the backbone for the pre-trained encoder in-order to extract the features. This model was chosen as it has shown great capabilities in image feature extraction, especially for contrastive learning [11]. It has shown to be of state-of-the-art status when it comes to contrastive learning tasks. ResNet18 was also considered for this task, however, after initial experiments it was deemed that the 18-layer variant was far too simple and did not capture enough features in the training process.

D. Classifier Model Architecture

The classifier model is a simple, untrained feed forward network which is comprised of an input layer with 128 dimensions (the output vector of the encoder) and two linear layers with the Rectified Linear Unit (ReLU) activation function.

$$ReLU = \max(0, x) \tag{2}$$

E. Evaluation Metrics

The encoder and classifier use different evaluation metrics in-order to validate the training. The encoder uses supervised loss in order to minimise the representation of the images of the same class closer together while pushing away the rest. Therefore, the encoder takes in the image and processes it, normalises it, and then apply (1) to the batch.

F1 (3) and accuracy (4) were used in order to evaluate the effectiveness of the classifier. (3) is used as a metric to evaluate the accuracy of the model, while the accuracy metric computes the number of times that model accurately predicted a sample across the entire dataset [12].

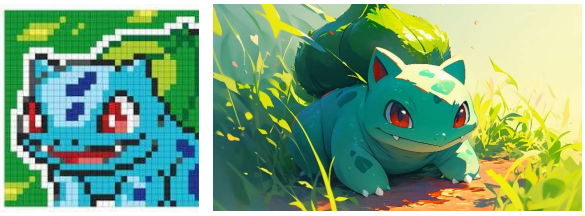
$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

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

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

$$Accuracy = \frac{TP+T}{FP+FN} \tag{4}$$

F1, precision, recall and accuracy are not included in the encoding stage as predictions are not being made, and instead uses the supervised loss function.



(a) 32x32 image. Sourced from [17].

(b) 2912x1632 image. Sourced from [16].

Fig 5: **Preprocessing example.** Depiction of 32*32 RGB image (a) and higher resolution image (b). The preprocessing was necessary in order to batch the images into a computationally reasonable amount. Any higher resolution images would cause the convolution embedding layer to proportionally increase in complexity, thus requiring more computation power.

IV. Experimental Settings

A. Hardware

The GPU used was the RTX 2080, with 368 tensor cores and 8 GB GDDR6 memory. 32GB of RAM was also utilised.

B. Hyperparameters

Hyperparameters were extensively tested and evaluated, with the aforementioned hardware limitations, these include:

- 20 epochs for both the encoder and classifier models.
- A learning rate of 0.0001
- Adam optimisation was used in order to iteratively adjust the learning rate for the parameters

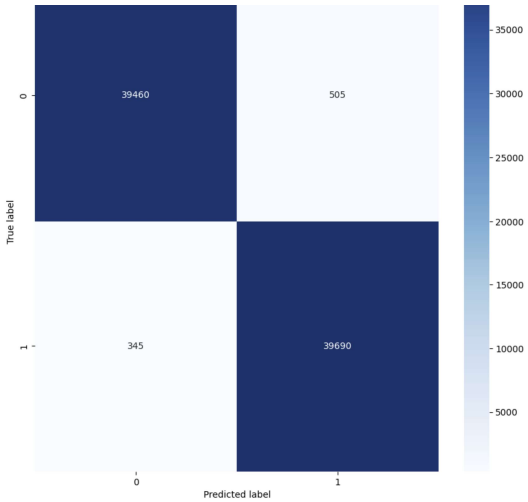


Fig 6: **Confusion matrix of predictions and true labels**. Reaching 98.9% accuracy in this experimental trial.

- Supervised contrastive loss temperature was set to 0.05, indicating that all the predictions made will be very confident.
- A dropout of 0.5 was used to regularise the network.

V. Results

As previously mentioned, the supervised contrastive loss function was adjusted to provide only the most reliable predictions. This makes the results all the more impressive, reaching accuracies of 99.20% (Fig 7 for confusion matrix of this). The model could encode the extracted features and the confidently separate the real and synthetic classes during classification. The precision, recall and F1 scores were all equally high as well, a strong indication that this model is not only accurate but robust and generalisable as well.

A. Comparative Study

From what was found, due to this being a relatively newer dataset, the publicly available dataset on Kaggle has many other submissions. I believe that the contrastive learner is actually the best performing. The three most highly rated models were a small CNN [13], a larger CNN [14] and a vision-transformer (ViT) [15].

A CNN uses convolution layers to extract features and the ViT uses self-attention to extract features from a tokenised representation of the image. The larger CNN did not give metrics other than accuracy.

As also shown in the results Table 1, out of all the approaches, the contrastive learner scored the best precision, recall and F1 scores.

Small CNN: Although this model was efficient, it still garnered worse results than all the other models.

Larger CNN: The larger CNN provided slightly better results thatn the small CNN, however was beat out by the contrastive learner and the ViT.

ViT: The ViT has earned it's reputation as being an effective learner by using self-attention, however this did not surpass the contextual learner either.

This comparative study has demonstrated the effectiveness of the contrastive learner in detecting synthetic images.

B. Ablation Study

In order to evaluate how well the contrastive learner was performing, an ablation study was conducted. The studies included (a) the removal of a single sub-class before training, then using only that class during the testing phase to evaluate how well this model can perform on completely unseen data; (b) the pairing and unpairing of data as a pre-processing step; (c) the removal of the projection head in the encoder.

1) Removal of a single class during pre-processing

Table 2: **Comparative study results**. The contrastive approach had not only the most consistent results among the alternative models but had the highest score in each respect.

Ref	Model	Precision	Recall	F1	Acc
	Contrastive	**0.9919**	**0.9919**	**0.9919**	**0.9920**
[13]	SmallCNN	0.9825	0.9479	0.9644	0.9464
[14]	CNN	-	-		0.9700
[15]	ViT	0.9225	0.9445	0.9825	0.9822

Table 1: **Results from Pairing and Unpairing**.

Pairing	F1 (weighted)	Accuracy
Paired	0.8354	0.837
Unpaired	**0.9919**	**0.992**

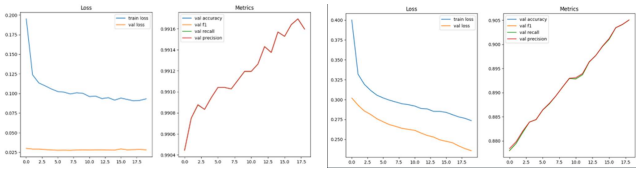
Before training, the 10th class was isolated and removed from the training set; then was used for testing. Removing this class resulted in an accuracy score of 78.19%.

2) Pairing and unpairing of data

Omitting the intra-class pairing was a purposeful design choice. By unpairing the data, instead of focusing on more traditional features among the classes (e.g. the point of the ears on a cat), the model can generally distinguish what a synthetic image is to a genuine image.

3) Removal of projection head in encoder

The projection head was removed to see the effect it has on the classifier. Fig. 7 shows with projection head, the figure below that shows without a projection head. The accuracy decreased by 9% after this. Although the accuracy decreased by 9%, down to 90.68%, this is still a very convincing accuracy score.



(a) performance of classification with included projection head

(b) performance of classification without a projection head

Fig 7: **Ablation results of projection head inclusion and exclusion**. The exclusion seems to have a more stable learning process, however, the measuring metrics clearly favour the inclusion; reaching accuracies of >99% than without, reaching >90%.

VI. Discussion

A. Interpretation of Results

The results were exciting and interesting. The removal of a class to then be used for testing is to gauge how well the model can predict on completely unseen data. With an accuracy of 78.19 %, we can presume that the model has very good generalisable synthetic image detection (at the very least, based on Stable-Diffusion-1.4 images).

Removal of the paired data improved the performance greatly. The paired data was designed to learn the features of similar and dissimilar images. However, unpairing the data meant that the model learned to distinguish between the classes as well, improving the generalisability of the detection.

In Fig 8, we can see that with the same parameters for with and without the head, the removal of the head slows convergence down and does not reach the same levels of accuracy. This underpins the importance of the projection-heads role in contrastive learning, by mapping the features onto a lower-dimensional space, it can generalise much better.

B. Implications

A real-world implication of this research is potentially a browser add on which segments a desired

image and can tell you if it is real or synthetic. This would be particularly useful on any social media site.

C. Limitations

There were two primary limitations that could not be addressed: (1) hardware constraints and (2) single-generative model images. The hardware constraints are unavoidable for something of this scope, however it should be noted that this research was done within the constraints of my GPU. The second limitation is that all the synthetic images are being generated by Stable-Diffusion-1.4; future research in this context should be inclusive of all/most of the popular generative models. I did not have the resources or time to self-generate a sufficient dataset, thus CIFAKE-10 was incredibly valuable.

D. Future Research

Future considerations include potentially training the model on an even larger corpus of publicly available data, where it may directly evaluate and self-supervise itself to generate more training data. Additionally, alternatively, supervised contrastive loss was used; other contrastive loss functions may wish to be considered. Alternate model architecture sizes could also be used.

VII. Conclusion

I have shown that contrastive learning is not only more generalisable but is also more accurate than deep learning and self-attention feature extraction methods like CNNs and ViT with detection of synthetic images. There was F1 and accuracy gains when using the contrastive method over the alternative models. This is significant as it opens up these methods into more synthetic detection applications which may increase the online safety for users. In conclusion, contrastive feature extraction should be used as a basis for all future synthetic image detection and future research should be focused on implementation in real-world applications.

VIII. Acknowledgement

Thank you to Dr Qiao and Zhe Luo for helping facilitate my learning throughout 2023 in this research and offering their guidance.

IX. References

[1] "AI Has Already Created As Many Images As Photographers Have Taken in 150 Years. Statistics for 2023," *Everypixel Journal. [Online]. Available: https://journal.everypixel.com/ai-image-statistics. (Accessed: May 16, 2024)* .

[2] J. Z. a. S. Z. H. Wu, " Generalizable Synthetic Image Detection via Language-guided Contrastive Learning," *arXiv preprint arXiv:2305.13800. [Online]. Available: https://arxiv.org/abs/2305.13800.*.

[3] P. T. C. W. A. S. Y. T. P. I. e. a. P. Khosla, " Supervised Contrastive Learning," *arXiv preprint arXiv:2004.11362. [Online]. Available: https://doi.org/10.48550/arXiv.2004.11362,* April 2020..

[4] A. J. P. A. Jonathan Ho, "Denoising Diffusion Probabilistic Models," *arXiv preprint. [Online]. https://arxiv.org/abs/2006.11239* .

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, Bing Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets," *arXiv preprint [Online]. https://arxiv.org/pdf/1406.2661.*

[6] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," 2012.

[7] A. a. O. B. Oh, "A Brief Survey of Watermarks in Generative AI," 2023.

[8] Y. M. N. Z. H. W. Y. C. M. X. X. L. Tong Qiao, "A novel model watermarking for protecting generative adversarial network," *ScienceDirect. https://doi.org/10.1016/j.cose.2023.103102,* 2023.

[9] Q. Bammey, "Synthbuster: Towards Detection of Diffusion," *IEEE Open Journal of Signal Processing,* 2023.

[10] J. J. Bird and A. Lotfi, "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images," *arXiv preprint arXiv:2303.14126, March 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2303.14126.*

[11] "Contrastive Learning on ImageNet-1K," *Papers with Code. [Online]. Available: https://paperswithcode.com/sota/contrastive-learning-on-imagenet-1k. (Accessed: May 16, 2024)..*

[12] R. Kundu, "F1 Score in Machine Learning: Intro & Calculation," 2022. [Online]. Available: https://www.v7labs.com/blog/f1-score-guide#:~:text=F1%20score%20is%20a%20machine%20learning%20evaluation%20metric%20that%20measures,prediction%20across%20the%20entire%20dataset..

[13] "Training a Small CNN on CIFAKE," *Kaggle. [Online]. Available: https://www.kaggle.com/code/birdy654/training-a-small-cnn-on-cifake. (Accessed: May 16, 2024)..*

[14] "Fine-tuning CNN 97.08%," *Kaggle. [Online]. Available: https://www.kaggle.com/code/guptaachal02/fine-tuning-cnn-97-08. (Accessed: May 16, 2024)..*

[15] "CIFAKE AI-Generated Image Detection ViT," *Kaggle. [Online]. Available: https://www.kaggle.com/code/dima806/cifake-ai-generated-image-detection-vit. (Accessed: May 16, 2024)..*

[16] "AlphaCoders-Wallpapers," *https://alphacoders.com/bulbasaur-%28pok%C3%A9mon%29-wallpapers.*

[17] "Pinterest. 32x32 Images.," [Online]. Available: https://www.pinterest.com.au/purple26peace/32x32/