DeepFake Detection Model Comparisons And Analysis

Yanjie Chen, Yu Xia
Department of Computer Science
Boston University
yanjie17@bu.edu, xiayu@bu.edu

Abstract

As our understanding of neural networks improves over the past decade, various implementations of GANs have emerged into our daily lives. Another well-known name of this network is DeepFake, which is the combination of "Deep learning" and "fake". Ever since Deepfake made an appearance, it is well acknowledged by everyone that goes on the internet. With this GAN your fantasy comes true – you can put your face on whoever you want to be, such as celebrities, or put other people's faces on someone else's body to mock them. While it does have an entertaining side, problems also arise as DeepFake becomes popular among the crowd. DeepFake is soon abused in making fake news, creating disturbing images that include explicit content, and even bypassing security systems that depend on face recognition.

Misleading information and unsettling pictures alert most of its audiences that DeepFake is not as pleasant as it seems. As the problem gets serious, it is now urgent for us to come up with a validation method for these materials that might be processed by the GAN. Fortunately, as of recent years, there are countless neural networks proposed to settle this issue. Open source datasets are also available on multiple platforms. In this paper, we are focusing on comparing three different CNNs: 1).DenseNet; 2).VGG16; 3).a Custom CNN; against each other in their efficiency and accuracy of detecting if an image is handled by GAN(Generative Adversarial Network) in hopes of bringing forward more discussion on this topic.

1 Introduction

DeepFake is most favored by the media industry, such as social media platforms and Film and TV production. It creates fabricated images, videos, or even audio. DeepFake achieves this by utilizing supervised and unsupervised machine learning such as CNN(convolutional neural networks), GAN(generative adversarial networks), and autoencoders. With all these neural networks it layers images or videos of a person A to another person B to make it looks like person A is doing something person B is doing. In this process, the dimension of the image or video provided is first reduced by the encoder while reserving the critical information. Then the decoder is applied to reconstruct the image/video with another person's information. A famous example of the application of DeepFake is the movie Fast and Fury series[7].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

On the other hand, the non-physical way of detecting DeepFake technology mostly involves different variations of CNNs. The recent VGG series and the DenseNet are both known for their accuracy in image classification. They all start with compressing the input image and then assign importance to certain parts of the image. This way CNNs are able to classify images with high precision. After much consideration, we decided to work with VGG16, DenseNet-169, and a custom CNN which we took inspiration from VGG networks and compare their accuracy to see which one will be the better fit for securing the cyber community.

2 Data

We initiated our models with a dataset collected from Kaggle. It contains in total 140,000 images of real human faces and A.I generated images. For dataset anatomy, the entire dataset is split into three subsets: training, testing, and validation with the amount of 100k, 20k, 20k images respectively. Within each subset, it is composed of real and fake images evenly. This dataset structure enables us to conduct our model training and analysis in a relatively smooth manner in which pre-proceedings thus became unnecessary. The 140k face image dataset contains people of highly diverse backgrounds from different races, ages, colors, and genders. More specifically, the lighting, face direction, and facial expressions also varied vastly. This highly diversified dataset allows us to attempt multiple detection models.

3 Models

3.1 DenseNet

A DenseNet is a type of convolutional neural network that utilizes dense connections between layers, through Dense Blocks, where all layers are connected directly with each other. DenseNet is able to achieve better performance than conventional neural networks such as ResNets and Highway Networks while requiring fewer parameters. The latter networks could easily surpass 100 layers causing tremendous time to train. In our experiment, we used DenseNet 169 due to its balanced performance in both accuracy and loss rate.

DenseNet has a unique feature that connects all layers directly to each other. In such feed-forward nature, all the information from preceding layers is passed into the next layer. In such a manner, each layer is able to access gradients from its previous loss functions such that it helps to reach the parameter efficiency and make them easy to train. For our own DenseNet, we added three dropout layers and batch normalization in order to decrease the chance of overfitting.

3.2 VGG16

VGG-16 is a convolutional neural network that is 16 layers deep and according to Neurohive, its 16 layers are composed of mostly convolutional layers and pooling layers.

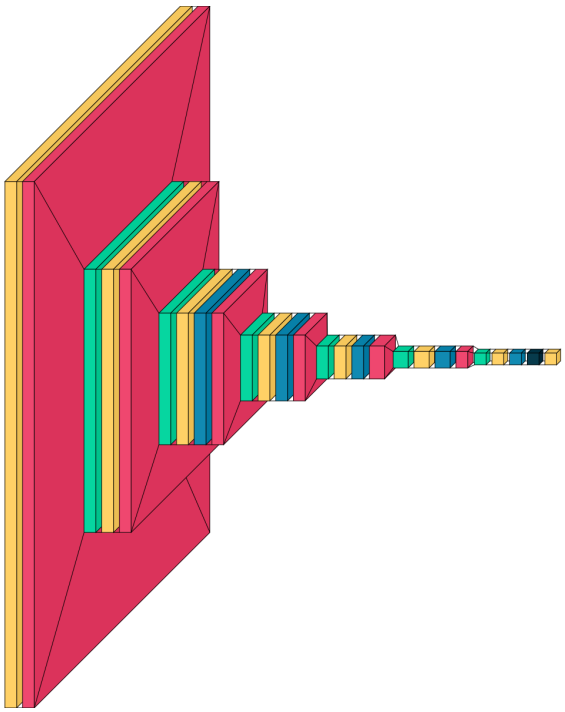
In the figure below x-x, the input to cov1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional layers, where the filters were used with a very small receptive field: 3×3. In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels . The convolution stride is fixed to 1 pixel; the spatial padding of convolutional layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 convolutional layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the convolutional layers. Max-pooling is performed over a 2×2 pixel window, with stride 2.[9]

This neural network is very accurate and capable of classification, but it has large network architecture weights and is difficult to train.

3.3 Custom-CNN

In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

We build our custom CNN model from a well-known model structure that includes conv2D, max pooling, and Batch Normalization of a total of six layers. Our team also made a modification to the dropout rate. That is to set two drop rates of 0.05 and 0.15. The reason behind this is to resemble the pseudo-randomness of false-positive detection dropout samples. With an average dropout rate of 0.1, the model successfully remains relatively steady and performs well.



4 Results and Analysis

4.1 Model Performances

Table 1: Model Performances

	Accuracy	RUC AUC SCORE	AP SCORE
Dense169	0.5649	0.5079	0.5089
VGG16	0.8610	0.9425	0.9419
CustomCNN	0.9177	0.9950	0.9946

Table 2: Model Performances on DeepFake Image

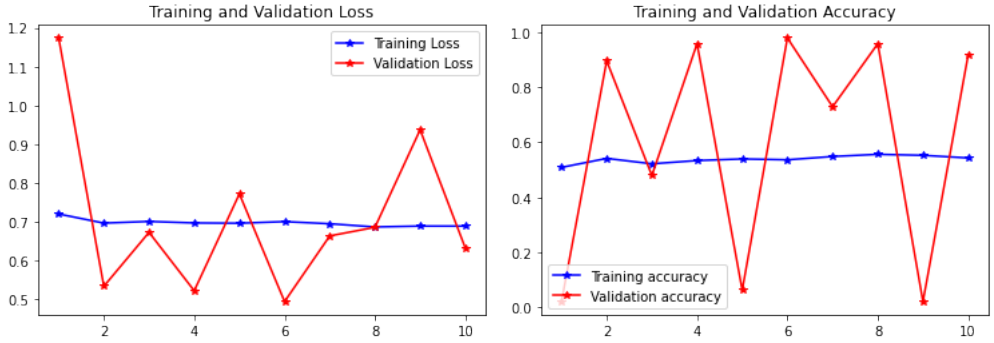
	Precision	Recall	f1 Score
Dense169	0.46	0.18	0.26
VGG16	0.82	0.92	0.87
CustomCNN	0.99	0.84	0.91

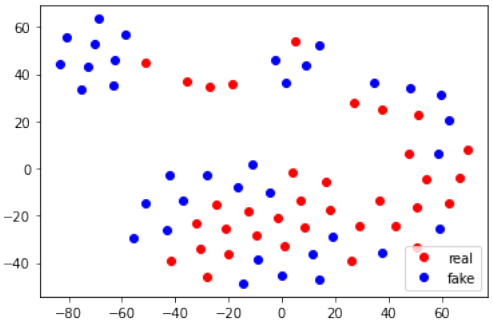
Table 3: Model Performances on Real Image

	Precision	Recall	f1 Score
Dense169	0.49	0.79	0.61
VGG16	0.91	0.80	0.85
CustomCNN	0.89	0.99	0.92

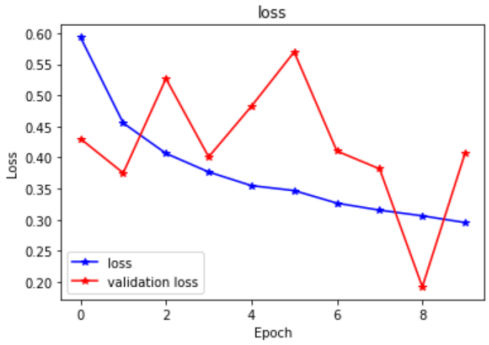
The performance of each model is shown as above. Observations include that the CustomCNN has the highest accuracy. It is also necessary to point out that the Dense 169 model we trained is not able to run on the SCC with the default setting batch size = 64. This situation forced us to shrink the training dataset and thus reached slightly above 0.5 accuracies. As for the VGG model, it performed well and has high scores in both RUC AUC and AP scores.

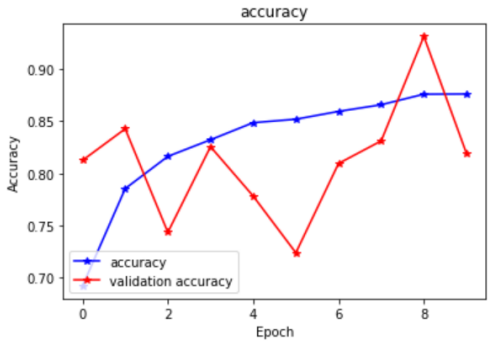
4.2 DenseNet

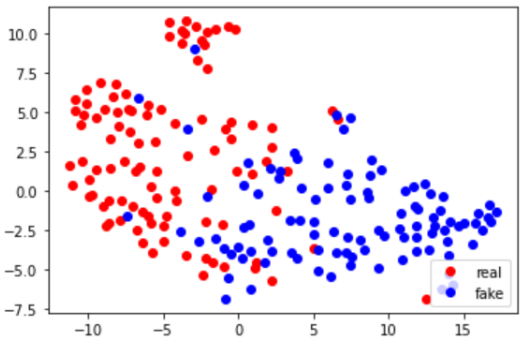



real
fake

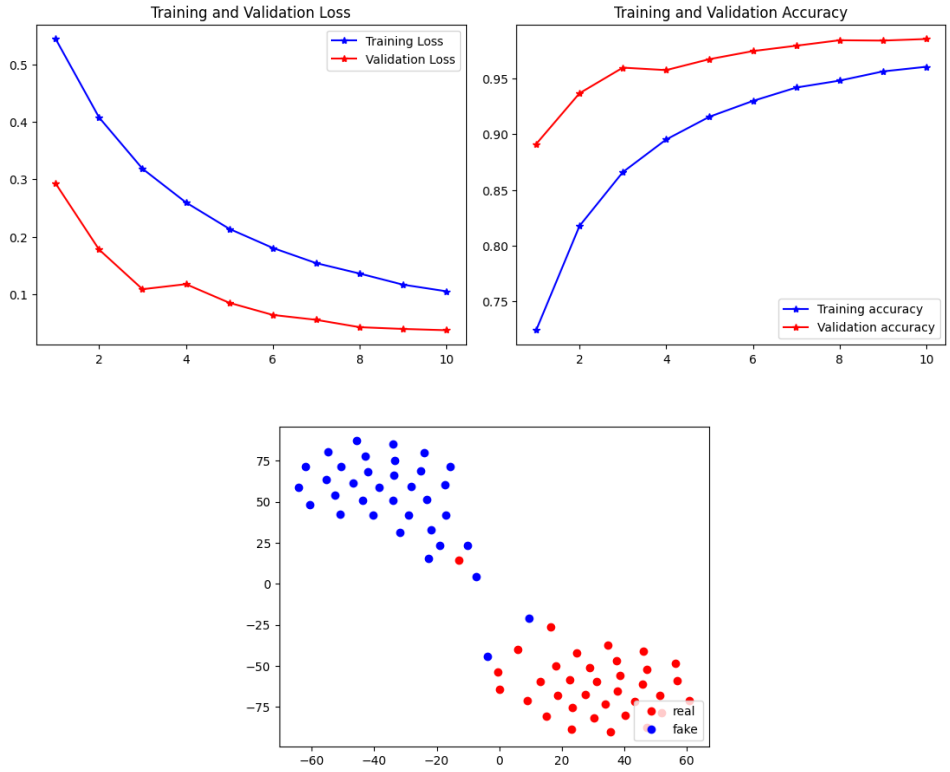
4.3 VGG16


loss
Loss
Epoch
loss
validation loss


accuracy
Accuracy
Epoch
accuracy
validation accuracy


real
fake

4.4 Custom-CNN



5 Discussion

From an analytical perspective, the custom CNN performed the best as it reached the lowest loss as well as higher accuracy. While DenseNet and VGG hold their own unique advantages: one requires fewer input data and parameters. Also, it could be inferred that if some larger dataset would be fed into all three models, the outcomes would be drastically different.

Furthermore, we will conduct discussions for each model as follows. When looking into our results, we find that VGG16 has the second-highest accuracy even with limited amounts of data. It has the best valid final test accuracy but requires a lot of resources and time to train.

DenseNet requires the least amount of parameters to reach a similar performance as the other two. However, its training time was tremendously longer than the other two and the model crashed multiple times even with 4 GPUs in the Boston University's Shared Computing Clusters(SCC).

First, the custom CNN model may be time-consuming. From our 140k dataset, it is trained for six minutes per epoch and in total one hour for the whole model. We configured it with Despite that it reached a relatively high accuracy, the speed of training may not be as efficient as the other two pre-configured models. Custom CNN reached the highest accuracy of 91.77 percent which performed the best among the three. The training time was relatively less demanding. The parameter optimization process was smooth due to its friendly training resources requirements, we even managed to train it locally with RTX 3080 GPU. The results we obtained may improve more if more datasets are available.

6 Conclusion

The era of DeepFake is inevitable. Numerous media and entertainment corporations now rely on this technology and no amount of effort would be enough to stop it's spreading in unethical and even criminal use. We now have to face the risks as it poses threats to our community.

However, no matter how precise the models are, a model with 100 percent accuracy is yet to be discovered. Any accuracy below 100 will serve since there are more than overflowing manipulated images and videos, and even 1 percent of them is an unignorable quantity. Another worth noting precaution is to layer credential images or videos with encrypted noise patterns or watermarks to make it easier for neural networks to detect the difference between real and fake information. As law enforcement is not enough for restraining the immoral use of DeepFake, the fight for a healthy and secure community will end up being the ultimate war between machines.

Future works would include training our models on more datasets, exploring more CNN models, and customizing these models to reach better accuracy. We would also do more research on applying other methods including unsupervised learning to achieve better performance on DeepFake detection.

7 References

[1] Chih-Chung Hsu & Yi-Xiu Zhuang & Chia-Yen Lee (2021) *Deep Fake Image Detection Based on Pairwise Learning* $Appl.Sci$.

[2] https://docs.openvinotoolkit.org/latest/omz_models_model_densenet_169.html

[3] https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803

[4] https://github.com/Codfish71/DeepFakeDetection

[5] https://www.kaggle.com/zohaib30/fake-vs-real-tensorflow-keras

[6] https://github.com/huangshiyu13/deepfake_detection/tree/master/timm/models

[7] https://www.hollywoodreporter.com/movies/movie-news/how-furious-7-brought-late-845763/

[8] https://arxiv.org/pdf/1608.06993.pdf

[9] https://neurohive.io/en/popular-networks/vgg16/