Detect Real and Swapped face using Densenet

[1] Neppalli Bala Krishna Prasad, 11940730, neppallib@iitbhilai.ac.in
[2] Srilekha Kadambala, 11941190, kadambalas@iitbhilai.ac.in

Abstract. In recent years, with the rapid growth of generative adversarial networks (GANs), a photo-realistic face can be easily generated from a random vector. Moreover, the faces generated by advanced GANs are very realistic. Even a well trained or experienced viewer has difficulties to distinguish between real face(real in the sense that face exists in real world) and swapped face. Therefore, detecting the face generated by GANs is a necessary work. In this work we focus on the use of convolutional neural networks for classification of true versus fake faces obtained from a large online database. We try to explain in detail how we classify real and fake faces using Densenet architecture. We also compare our results with other models of fake face detection and prove that the results obtained by densenet are more accurate that other methods.

Keywords: Convolutional · Densenet · Random vector

1 Introduction

In the initial work we have done swapping of faces using GANs and we swapped a source face to a target face without changing the skin tone,expression and pose of the target. This method created a fake face which is not existed in the real world. Not only performing manipulation and other operations on existing face but there are many other methods that can even generate a complete fake face. This leads to many ethical problems with serious societal impact. The technology of Deepfakes has developed much in the recent years and a person even without a good knowledge of programming can swap the faces. This kind of things rises issues on privacy and anyone can easily create some viral videos on any celebrity. So in this work we propose a model that distinguish between real face and swapped face using densenet. A DenseNet is a type of convolutional neural network that utilises dense connections between layers, through Dense Blocks, where we connect all layers (with matching feature-map sizes) directly with each other. It is basically an extension of ResNet. ReLu activation function is also used to overcome the problem of non-linearity.

2 Problem Definition

Due to the increase in technological ways to swap faces and create fake fakes using Deepfakes, FSGAN, Face2Face and many other applications, it is getting difficult to know whether the image or video is real or not. It is leading to distortion of privacy and also leading to crimes without consent. This is the reason why we have to be able to detect the swapped and real faces by using the same tool 'technology' with which we have rooted those at first. The problem statement is to solve this by detecting between swapped and real faces.

The faces in the Figure-1 are generated by FSGAN. So these faces are considered as fake faces and our model will identify these kind of faces as fake faces.

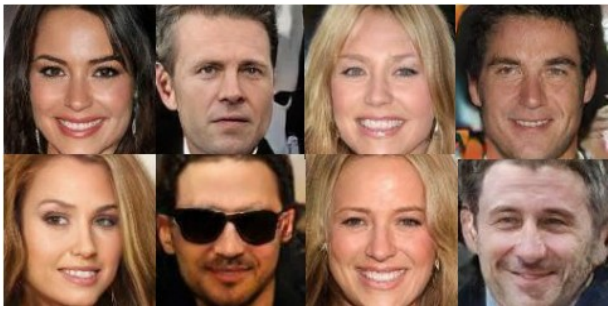


Figure 1: Fake faces generated by GANs.

3 Objective

To train a classifier to differentiate between a real face and swapped or fake face using DenseNet 121.

4 Technology used

The main model we used to solve this problem is Densenet. We also used support vector machine(SVM) to classify the retained images into two categories(real or fake).

Other Python Dependencies used:

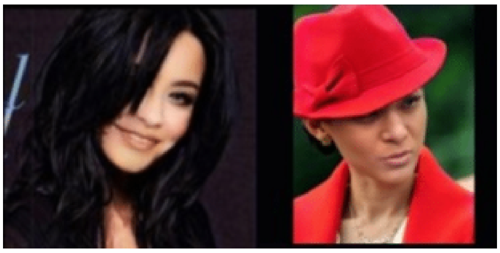
- Numpy
- Tensorflow
- Keras
- Matplotilab
- OpenCV
- Pandas
- Sklearn

5 Problems faced

The main problem we faced was detecting the faces which are covered or which have less portion of actual face. For instance, if a face is covered with a mask or eyes covered with some hair, the actual part of face is less, which makes it difficult to find the major differences (refer the figure below). If the face area is more, then the model can detect the flaws more accurately. Also, if the face is straight, then it detects more accurately. Greater the shift angle, more difficult the detection is.

6 Data Sets used

We used two datasets from Kaggle along with the images which we have generated using FSGAN. This dataset consists of nearly 1,40,000 images out of which 1,00,000 images are used for training, 20,000 images for validation and 20,000 images for testing purpose. A label of 1 was assigned to real images and a label of 0 was assigned to fake images. In every



set, there was an equal ratio of real and fake images. These real faces are obtained from flickr and mostly all the fake faces are generated by various GANs. These high-quality fake images contain a different face in each image, contain swapped faces and can have manipulations to the whole face or areas of the face such as the eyes, nose, or mouth.

7 Models Used

To detect whether a face in an image is real or swapped, we used the existing CNN framework Densenet. Basically densenet is an extension of widely known Residual CNN architecture(ResNet). In densenet each convolutional block receives input from all its preceeding blocks. It establishes a direct connection between each layer to all of the subsequent layers in the network through the use of dense blocks. DenseNet utilizes a transition layer made of convolution, average pooling, and batch normalization between each dense block to concatenate feature maps. So by concatenating feature maps, the collective data is shared by all layers, leading to a better flow of information. It also improves the gradient propogation bu connecting all the layers.

Additionally, using DenseNet poses additional benefits such as it requires fewer parameter, reduces number of feature maps needed, and is not computationally expensive compared to other CNNs. The model that we used is Densenet-121. This model contains 4 dense blocks with closely connected layers such as batch normalization (BN), ReLU activation, and 3 x 3 convolution. In addition to this, between each dense block, the model also contained a transition layer which included a 2 x 2 average pooling layer and a 1 x 1 convolution. After the last dense block, we added the custom dense layer with sigmoid activation.

8 Result and Performance

Neural networks are very efficient in classifying GAN generated images. Even though the faces in the data set have different poses or it can have manipulations on some parts of face like ears, eyes, then our model is able to detect them with more accuracy. By using Densenet we got an accuracy of greater than 90%. There are some other CNN architectures like VGGFace which can give more accuracy than Densenet-121. But VGGFace is computationally more expensive and requires much sophisticated processors when compared with that of Densenet-121. If we increase the number of epochs then we can probably get even more efficient results.

Our model is giving the results with an accuracy of 0.956 and precision of 0.94.

9 Conclusion and Further Work

This paper mainly introduces a detection method using Densenet architecture for detecting fake faces generated by GANs. From the results, it can be concluded these detection methods have high accuracy. Even with this model we can get accuracy as high as 90% is not enough. 10% of billions of images on Google or Facebook platforms would represent an immense loss of trust from users of these interfaces. Although the GANs are developing faster and faster. Various GANs may emerge in the future. They may generate higher quality fake faces. But we can get inspiration from the above methods. The GANs cannot completely describe many intrinsic properties of real images, such as differences in color spaces. So, trying to find the differences between fake images and true images. Then, according to the differences develop more advanced detectors. Furture work would include the use of unsupervised clustering methods such as auto-encoders to explore if true versus fake images cluster separately and also to add transparency and interpretability to our models by use of CNN visualization methods.

10 References

1. https://www.kaggle.com/xhlulu/deepfake-pretrain-densenet-on-face-classification

2. https://www.kaggle.com/ciplab/real-and-fake-face-detection

3. https://kcimc.medium.com/how-to-recognize-fake-ai-generated-images-4d1f6f9a2842#:~:text=Messy%20hair,thick%20stray%20hairs%20on%20foreheads

4. https://jis-eurasipjournals.springeropen.com/articles/10.1186/s13635-020-00109-8

5. Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2016.