

Deep Learning Using Forensic

[Fake Face prediction using VGG 16]

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfilment of the requirements to award the degree of

Bachelor of Technology/Master of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Mynam Karuna Sree - (AP20110010300)



Under the Guidance of

(Dr. Meenakshi Choudhary)

SRM University-AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

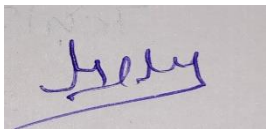
[December,2022]

Certificate

Date: 13/12/2022

This is to certify that the work present in this Project entitled “**Deep Learning Using Forensic**” has been carried out by **Bhavesh chanumolu, Gnaneswar Guddanti, Mynam Karuna Sree** under my supervision. The work is genuine, original, and suitable for submission to the SRM University - AP for the award of Bachelor of Technology in School of Engineering and Sciences.

Supervisor



(Signature)

Dr. Meenakshi Choudhary

Assistant Professor

Department of CSE

SRM University -AP

Acknowledgments

We are ever so thankful to all of those who have contributed to the successful completion of this project in any way. It is because of their dedication, hard work, patience and guidance that this has been achieved. A special thanks to our mentor who has been with us from the beginning, offering her advice and support assisted us in coordinating our project.

Table of Contents

Certificate 3

Acknowledgements..... 5

Table of contents 7

Abstract 9

Abbreviations 11

List of figures 13

Introduction..... 15

Methodology 18

How to train a VGG 16 model 22

Conclusion 25

Future work 27

Abstract

Artificial intelligence (AI), deep learning, machine learning and neural networks are the ones which represent extremely exciting and powerful machine learning-based techniques used to solve many day-to-day life problems. Deep learning is used for recognition, visual perception, decision-making and planning. Deep learning is a part of machine learning which is a part again in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabelled. Deep learning which is a technique used to generate face detection and recognize it for real or fake. Here in this paper, we used deep learning techniques to generate models for Real and Fake face detection. Here our goal is determining a suitable way for detection of real and fake faces. This model was designed and implemented, including both Dataset of images: Real and Fake faces detection through the use of Deep learning algorithms based on neural networks. We have trained dataset which consists of 6,000 images for total in 30 epochs, for our model VGG - 16 **giving 100% training accuracy, 99.18% validation accuracy, training loss 0.0003, validation loss 0.0265, and testing accuracy 99%.**

Keywords: Artificial intelligence, Deep learning, Human images, Real and Fake Face.

Abbreviation:

- CGFI - Computer generated face image
- GAN - Generative Adversarial Networks
- ReLU - Rectified Liner Unit
- F2F - Face-2-face
- VGG - Visual Geometric Group
- FC - Fully Connected
- GPU - Graphical Processing unit
- CNN - Convolutional Neural Network
- JPG - Joint Photographic Experts Group
- CV - Computer vision

List of Figures:

Figure 1. Architecture of VGG -16 20

Figure 2. Intuitive layer 20

Figure 3. Training loss and Training Accuracy 23

Figure 4. Examples of Real Images with predictions 23

Figure 5. Examples of the Fake images with predictions 24

1. Introduction

Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is in an unstructured manner. It is an effective technique and is used in various fields of natural language processing, image processing, computer vision, etc; facial recognition plays an important role in many areas such as security, camera surveillance, identity verification in modern electronic devices, criminal investigations, etc. which are quickly becoming integral parts of user security, allowing for a secondary level of user authentication. Deep fakes use deep learning techniques to synthesize and manipulate the image of a person.

It has become more and more difficult to differentiate between computer-generated and real-face images, even with human eyes. If the generated images are used with the intent to mislead and deceive readers, it would probably cause severe ethical, moral, and legal issues. This work presents the deep learning algorithm VGG-16 which is a special convolutional network designed for the classification of images. which is used in facial recognition for accurate identification and detection. The main objective of facial recognition is to authenticate and identify facial features. The reason for choosing the VGG-16 is because it is easy to understand which is a standard architecture of CNN and well-researched, available in all machine learning libraries.

1.1 Fake and Real Human Face

The advancement of Generative Adversarial Network (GAN) computers will generate vivid face images which will manipulate human beings easily. As a result, these generated fake faces will cause serious social risks, for example in fake news, fake evidence, and even cause threats to security.

Here powerful techniques are required as to detect these fake faces are highly desirable. However, in reality to these intensive studies in GANs, our knowledge of understanding these generated faces is fairly superficial and how to detect fake faces still remains a question mark. Moreover, these fake faces in real scenarios will be different from unknown sources, i.e., different GANs, and may undergo unknown image distortions such as down sampling, blur, noise, and JPEG compression, which makes our task even more challenging.

Here in our article, we aim to generate new insights on how important the understanding of fake faces and proposing a new architecture to handle these challenges.

1.2 Problem Statement

When where we use social media, we will find a fake identity of any person through using fake profile image. This fake profile image will be generated with using image editor, face effect, or any program which in turn the facial features of a person.

When we apply any effects on the face it can change the facial features and in turn making difficult us to know true identity of someone face. In this model, we did our best to train this model using the dataset to recognize of such fake faces.

1.3 Convolutional Neural Network (CNN)

These neural network uses a convolution operation instead of using matrix multiplication in any one of the layers. It is important for recognition and images classifications. Technically, these deep learning CNN models were used to train and test, where each of these input image will pass it through a series of convolution layers with filters (Kernels), polling, fully connected layers (FC) and apply SoftMax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

1.4 VGG-16 Model:

It is the most popular and widely used model for object detection and image classification proposed by A. Zisserman and K. Simonyan who belongs to the University of Oxford. It is a specific convolutional neural network designed for classification and localization. It is the most preferred model for extracting features from images. As it, names suggest it has 16 deep neural networks of 13- convolutional layers and 3- fully connected layers. which means it has a total of 138 million parameters. which can be a bit challenging to handle, but it is good-looking because of its uniform architecture. VGG-16 can be achieved through the transfer of Learning. In which the model is pre-trained on a dataset and the parameters are updated for better accuracy and we can use the parameter values.

2. Methodology:

My Proposed methodology includes gathering the dataset, identifying the tools and language to be used, pre-processing the images in the dataset, data augmentation, and construction of the model architecture, compiling the model, and training and validating the model.

2.1 Dataset:

The data set in this study consists of human faces of real-fake images. The numbers of images in this dataset are classified as follows: 1750 images that were real human faces, 1750 of fake-mid, 1750 of fake-hard, and 1750 images of fake-easy. The dataset of real-fake human faces images was collected from the Kaggle depository website.

2.2 Languages and tools used:

Python programming language is used for building the model. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, dynamically typed. The tools used are Google Colab a research tool best for deep learning models which is easy to use and does not require any practice for use. It can provide CPU or TPU for faster processing of the data and does not require installations.

2.3 Image format:

Dataset was collected from a set of real-fake human faces Images to detect whether an image is real or fake. The image format is (JPG) because it fits well within the model used to give the desired results.

2.4 Pre-processing:

The dataset is having a lot of images in which some of them having various dimensions. So, we need to resize all the images to 256*256 pixels which is the default size for this model to provide better accuracy.

2.5 Data augmentation:

The highest accuracy can be obtained in deep learning models by providing larger datasets. In another case, we can increase the model accuracy by augmenting the images of the dataset. Deep learning frameworks usually have a built-in library for data augmentation. In this model, we used two augmentation strategies to generate new training sets, (horizontal flip, and vertical flip).

2.6 Network Architecture:

We have trained our Real-Fake dataset using VGG-16. Which is a pre-trained models for deep learning.

2.6.1 Architecture of VGG-16:

Input Layer: It accepts images as input with the size 224×224 and 3 channels as the default size. These computerized applications will use deep learning techniques to increase the accuracy and efficiency of our diagnosis. Our datasets include human images, image processing techniques and data analysis.

Convolutional Layers: VGG's convolutional layers leverage a minimal receptive field of 3×3 filter with a stride of 1 which is the smallest possible size that captures up/down and left/right. Every convolution filter uses row and column padding. So, the size of the input, as well as the output feature maps, remains the same.

Max pooling: It is performed by a max-pool filter of size 2×2 with stride of 2, which means the max pool windows are non-overlapping windows. Not every convolution layer is followed by a max-pool layer but in some places, a convolution layer is following another convolution layer without the max-pool layer in between.

Hidden Layers: All the hidden layers in the VGG network use ReLU as their activation function.

2.6.2 Layers of VGG-16:

1. Convolution using 64 filters
2. Convolution using 64 filters + Max pooling
3. Convolution using 128 filters
4. Convolution using 128 filters + Max pooling
5. Convolution using 256 filters
6. Convolution using 256 filters
7. Convolution using 256 filters + Max pooling
8. Convolution using 512 filters
9. Convolution using 512 filters
10. Convolution using 512 filters+ Max pooling
11. Convolution using 512 filters
12. Convolution using 512 filters
13. Convolution using 512 filters+ Max pooling
14. Fully connected layer.
15. Fully connected layer.
16. Output layer.

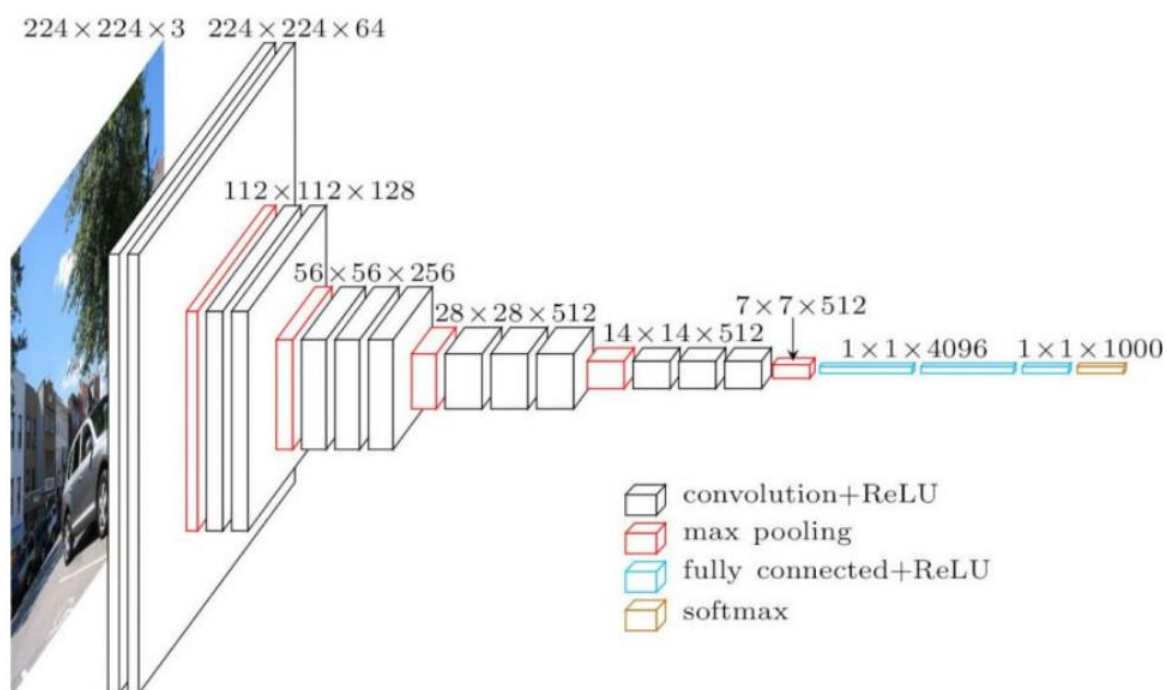


Figure 1: Architecture of VGG-16

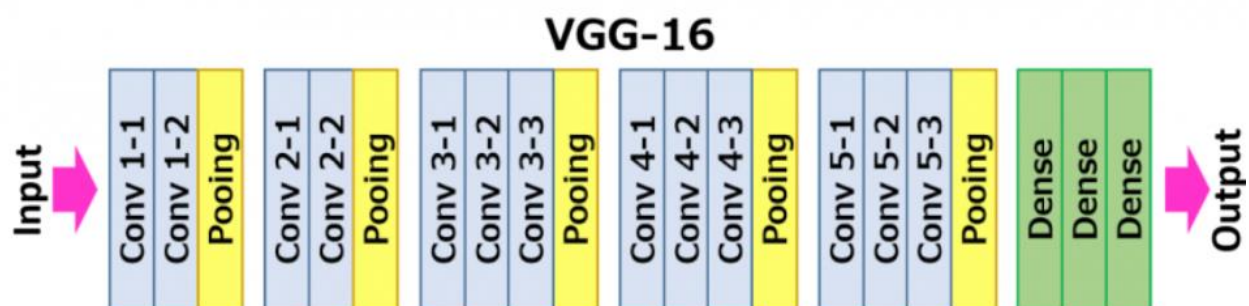


Figure2: Intuitive layout of the VGG-16 Model.

The following are the layers of the model:

- Convolutional Layers = 13
- Pooling Layers = 5
- Dense Layers = 3

3. How to train a VGG-16 model:

Step 1:

Importing the libraires

```
import numpy as np
import pandas as pd
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dropout, Dense, BatchNormalization, Flatten, MaxPool2D
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau, Callback
from keras.layers import Conv2D, Reshape
from keras.utils import Sequence
from keras.backend import epsilon
import tensorflow as tf
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from tensorflow.keras.layers import GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
import cv2

from tqdm.notebook import tqdm_notebook as tqdm

import os
img_width, img_height = 224, 224
```

providing the path for the dataset as well as real and fake images.

```
print(os.listdir("/content/drive/MyDrive/dataset-20221209T060920Z-001/dataset/training"))
real = "/content/drive/MyDrive/dataset-20221209T060920Z-001/dataset/training/real"
fake = "/content/drive/MyDrive/dataset-20221209T060920Z-001/dataset/training/fake"

real_path = os.listdir(real)
fake_path = os.listdir(fake)
dataset_path = "/content/drive/MyDrive/dataset-20221209T060920Z-001/dataset/training"
```

Step 2: Augmentation of train and test dataset.

Step 3: Download weights from a pre-trained method.

Step 4: Creation of model and compiling the model.

Step 5: Fitting the model.

Step 6: The output of the fitting is

```
Epoch 19/30
12/12 [=====] - 20s 2s/step - loss: 0.4314 - accuracy: 0.8123 - val_loss: 0.7720 - val_accuracy: 0.5804 - lr: 1.0000e-05
Epoch 20/30
12/12 [=====] - 20s 2s/step - loss: 0.4250 - accuracy: 0.8262 - val_loss: 0.7809 - val_accuracy: 0.5909 - lr: 1.0000e-05
Epoch 21/30
12/12 [=====] - 20s 2s/step - loss: 0.4302 - accuracy: 0.8149 - val_loss: 0.7747 - val_accuracy: 0.5839 - lr: 1.0000e-05
Epoch 22/30
12/12 [=====] - 20s 2s/step - loss: 0.4269 - accuracy: 0.8175 - val_loss: 0.7796 - val_accuracy: 0.5979 - lr: 1.0000e-05
Epoch 23/30
12/12 [=====] - 20s 2s/step - loss: 0.4259 - accuracy: 0.8210 - val_loss: 0.7721 - val_accuracy: 0.5769 - lr: 1.0000e-05
Epoch 24/30
12/12 [=====] - 20s 2s/step - loss: 0.4224 - accuracy: 0.8245 - val_loss: 0.7792 - val_accuracy: 0.5979 - lr: 1.0000e-05
Epoch 25/30
12/12 [=====] - 21s 2s/step - loss: 0.4269 - accuracy: 0.8193 - val_loss: 0.7979 - val_accuracy: 0.5734 - lr: 1.0000e-05
Epoch 26/30
12/12 [=====] - 20s 2s/step - loss: 0.4209 - accuracy: 0.8245 - val_loss: 0.7753 - val_accuracy: 0.5804 - lr: 1.0000e-05
Epoch 27/30
12/12 [=====] - 20s 2s/step - loss: 0.4143 - accuracy: 0.8280 - val_loss: 0.7902 - val_accuracy: 0.5455 - lr: 1.0000e-05
Epoch 28/30
12/12 [=====] - 20s 2s/step - loss: 0.4219 - accuracy: 0.8271 - val_loss: 0.7929 - val_accuracy: 0.5664 - lr: 1.0000e-05
Epoch 29/30
12/12 [=====] - 20s 2s/step - loss: 0.4246 - accuracy: 0.8262 - val_loss: 0.7869 - val_accuracy: 0.5734 - lr: 1.0000e-05
Epoch 30/30
12/12 [=====] - 20s 2s/step - loss: 0.4213 - accuracy: 0.8184 - val_loss: 0.7666 - val_accuracy: 0.5804 - lr: 1.0000e-05
```

3.1 Result:

After Running the Epoches for 30 with 53 Convolution layers we got Training layers and Training Loss graphs.

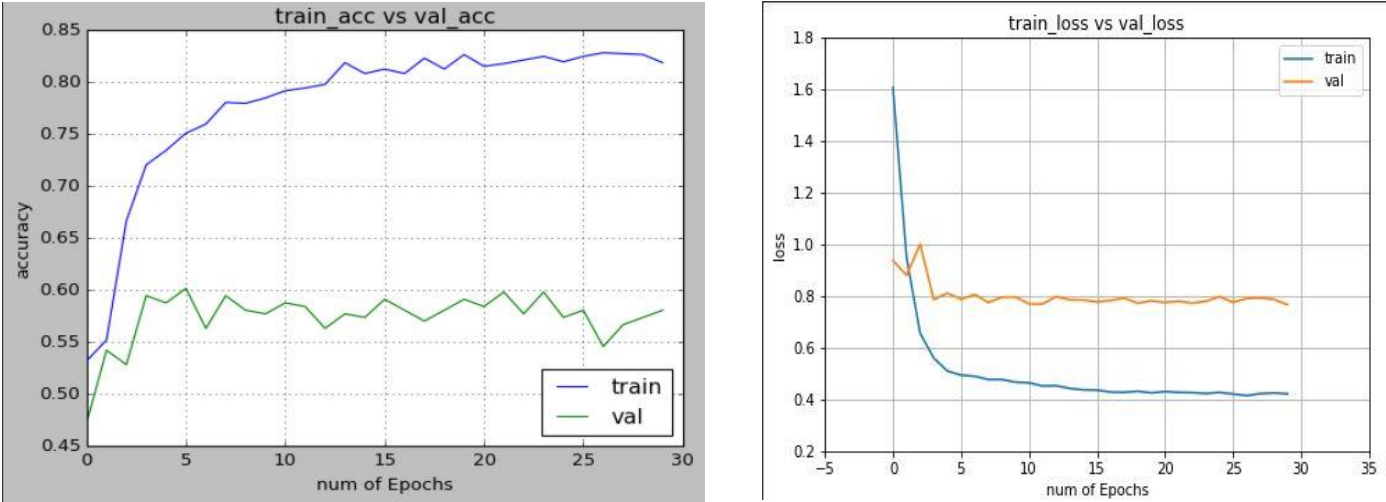


Figure 3: Training loss and Training Accuracy

Output:

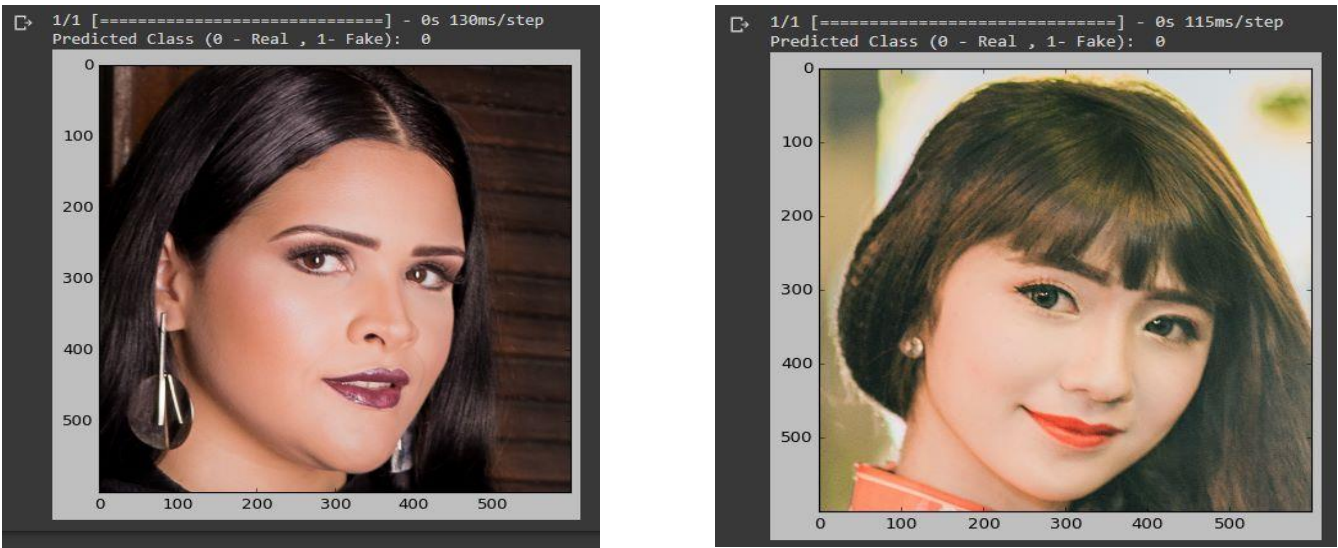


Figure 4: Examples of real images with predictions

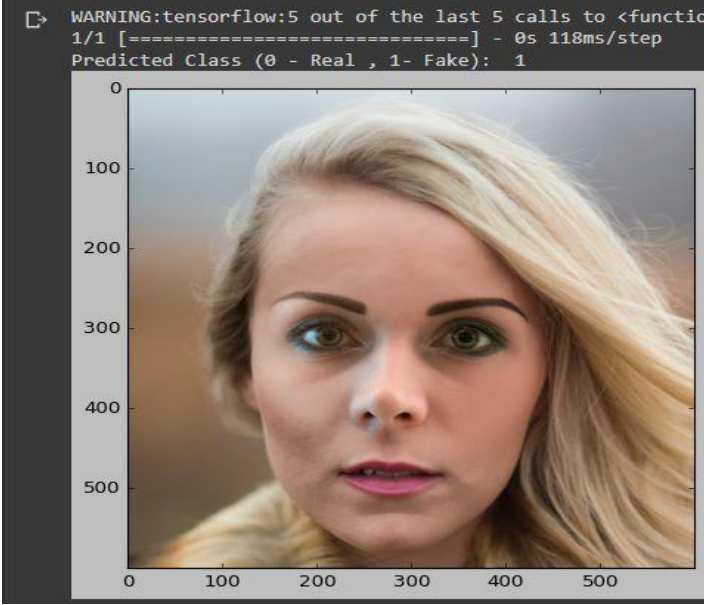


Figure 5: Examples of fake images with predictions

4. Conclusion:

This rapid advancement of image capability and image generation techniques has given us opportunity to create security-less and convincing fake face images. Most risky nature of data, whether it may be in terms of visual perception or algorithm discovery, was present in recent works. Here our main issue which has yet been taken into consideration for understanding the overall capability of existing fake face detection techniques. To do so we must answer the question of generalizability, of this work, which had been trained from dataset using 6,000 images for total of 30 epochs, and got to know that the ReLU model is the best model of network architectures with 100% training accuracy, 99.18% validation accuracy, training loss 0.0003, validation loss 0.0265, and testing accuracy 99%. We developed this by using Google Colab and Python language which supports for a deep learning machine. Thus, by concluding that for advancement of real-fake face detection technology requires the correct validation of datasets and also accuracy which will be included in all future research as a condition for publication as a requirement.

5. Future work:

1. Designing the models which take less time to train.
2. The model needs to have the capability to train more complex/deeper models.
3. The model needs to extract more features from the dataset which increases the accuracy of prediction.
4. The model needs to have fewer parameters and occupies less space but needs to be highly efficient.
5. The model needs to work perfectly even with less computational power and the ability to run on small devices.