IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Deepfake Detection of Images Using Deep Learning Techniques

Omkar Singh[1], Om Patel[2], Sahil Singh[3]

[1]H.O.D. (Data Science), [2,3]P.G. Student (Data Science)

[1,2,3]Thakur College of Science and Commerce, Kandivali East, Mumbai – 400101, Maharashtra, India

ABSTRACT

Deepfake detection is based on a deep learning model. Deepfake content is created with the help of artificial intelligence and machine learning to replace one person's face with another person's face in pictures. These manipulated pictures will undoubtedly have an enormous societal impact. Deepfake uses the latest technology like Machine Learning (ML), and Deep Learning (DL) to construct automated methods for creating fake content. We use the DFDC dataset for training the model, it contains 100,000 videos of more real and fake videos. To identify real and fake images using various Convolutional Neural Network (CNN) models, namely EfficientNetB0, VGG-16, and Xception. Our primary objective is to determine the model that yields the highest accuracy in identifying deepfake images. Initially, we preprocess videos from the DFDC dataset, extracting frames and preparing them for input into CNN models. Subsequently, we construct and train each CNN architecture on the preprocessed data, integrating techniques like data augmentation to enhance model generalization and resilience. Performance evaluation utilizes metrics such as accuracy, precision, recall, and F1 score.

Keywords: Deepfake Detection, Deepfake Image, Convolution Neural Network, DFDC Dataset

INTRODUCTION

Deepfake, which combines the terms "Deep Learning" and "Fake" is a method that uses artificial intelligence to produce modified media, including images, videos, and audio. In recent years, the usage of deepfakes has grown in popularity and raised serious concerns for society. Deepfake may be used for evil intent, including instigating violence, destroying reputations, and disseminating false information. Deepfake's most widely used app is FaceApp. Social media sites like Facebook and Twitter identify and remove deepfake content to ensure authenticity. Because deepfake makes it so easy to create fake scenarios, we can no longer trust any video footage at face value. It's less about the death of truth and more about the end of faith in the trust. Deepfake identification is an essential activity that may shield people from exploitation and stop false information from spreading. As a result, creating efficient deepfake detection techniques has become essential.

The most prominent method to create realistic images is by using GANs, Generative Adversarial Neural Networks which is a deep learning technique. GAN consists of two components, a generator and a discriminator. These two work in an adversarial nature where the generator generates the fake images and discriminators distinguish the image whether is real or fake.

Figure 1: Deepfake images

LITERATURE REVIEW

Yogesh Patel et.al conducted a study on deepfake image detection using a Dense CNN Architecture. The dataset used is from Deepfake Images Detection and Reconstruction Challenge which contains real images from CelebA and FFHQ and for deepfake images the dataset used are GDWCT, AttGAN, STARGAN, StyleGAN, and StyleGAN2. They have proposed a D-CNN model for binary classification to detect deepfake images. The augmented CNN model is presented to the D-CNN model to extract the deep features from input images using convolution layers. After performing the convolution operations over the images can be used to classify the input images into fake and real images. This proposed model has achieved 97.2% accuracy on the test dataset. The performance of the model goes down when implemented on other existing models like MesoNet and MesoInception network over the CelebDF dataset.[1]

Hanqing Zhao et.al have redefined deepfake detection as a fine-grained classification task, introducing a fresh approach to the field. Additionally, they presented a new multi-attentional network architecture designed to capture local discriminative features from various face-attentive regions. The datasets used are FaceForenscis++, CelebDF, and DFDC. The proposed architecture decomposes the single attentional structural networks into multiple regions which is more efficient to collect local features. They have used local attention pooling to capture textural patterns. Implementing the multi-attentional framework achieves good improvement on different datasets using extensive metrics.[2]

Yuval Nirkrin et.al has proposed a method for detecting deepfake images based on discrepancies between faces and their context. The datasets used for this method are FaceForensics++, CelebDF, and DFDC. Their proposed method involves two networks, one for detecting the face with its surrounding region and the other for detecting facial landmarks based on an Xception network that considers the face context. This new method outperforms the baseline Xception model by a significant margin. This method may not perform well when images have low contrast and blurry features.[3]

Sohail Ahmed Khan et.al have proposed a hybrid transformer network for deepfake detection of images. The datasets used are FaceForensics++ and DFDC. Two CNN architectures are used, XceptionNet and EfficientNet-B4 for feature extraction, and then a BERT-style transformer is used to learn the joint features. The model may not perform well on unseen data with different styles of forgery techniques applied to images.[4]

Ali Raza et.al has proposed a novel deep-learning approach for deepfake image detection. The dataset used in this approach is a deepfake dataset that is publicly available on Kaggle. The novel DFP approach is based on a hybrid of VGG16 and convolution neural architecture. The hybrid layers of both are used to create the architecture. The DFP approach performed better than other state-of-the-art techniques. This approach does not better generalize on different types of images generated by different types of techniques.[5]

Asad Malik et.al surveyed deepfake detection for human face images and videos. Different algorithms were studied that used different datasets. Some of the datasets used are Celeb-DF, DeepForensics, WildDeepfake dataset, and OpenForensics dataset. Different algorithms and CNN models were used for a comprehensive analysis.[6]

METHODOLOGY

CNN

CNN is a type of neural network in deep learning which is usually for computer vision tasks. CNN is mostly employed in image processing to classify and detect images by extracting features from images. CNN consists of three layers. The convolution layer does the computation of extracting the features from the image data by using filters and this operation is referred to as convolution. The CNN typically applies the ReLu activation function to introduce non-linearity to the model. The pooling layer also sometimes called the downsampling layer reduces the dimensions of the image by applying the filters. The filters use an aggregation function to reduce the features. There are two types of pooling, Average pooling takes the average of each pixel value and Max pooling takes the maximum value. The final layer is the Fully connected layer which performs classification tasks using the softmax function that classifies input and produces a probability score between 0 and 1. Figure 2. Represents the working of the CNN model.

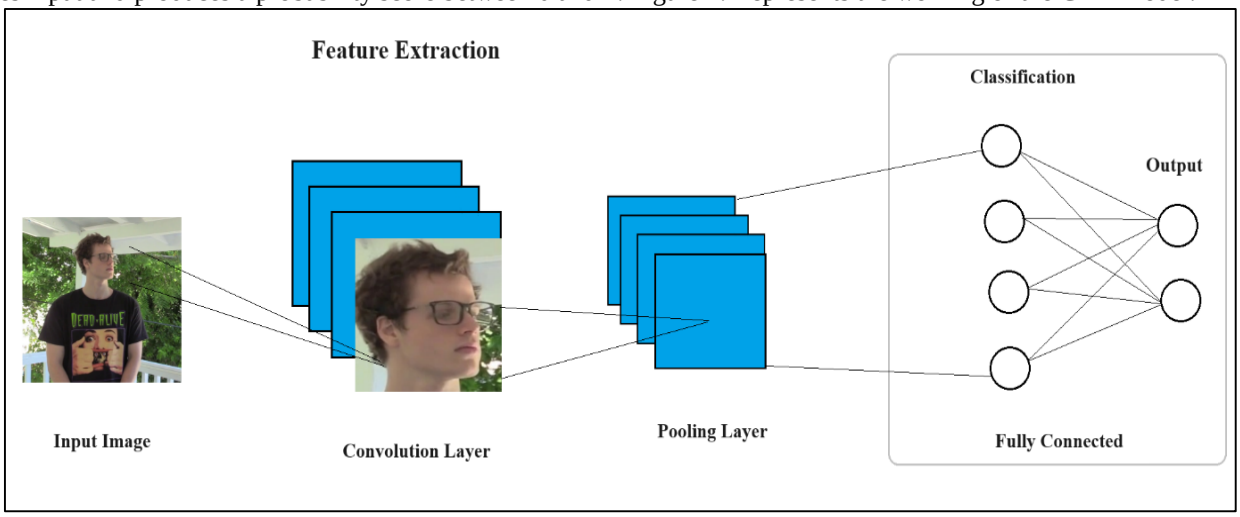

Figure 2: Overview of CNN Model

Dataset

The dataset used for training the model is DFDC, Deepfake Detection Challenge Dataset[7] which is publicly available on Kaggle. The DFDC dataset contains 100,000 videos each with a video length of around 10 seconds and the images were extracted from a randomly selected subset of videos. As there were more fake videos than real videos, so after the frame extraction at 1 fps, the dataset was balanced by randomly removing the excess number of fake images.

Figure 3: Some real & fake images extracted from the DFDC datasets

EfficientNetB0

EfficientNet is a class of CNN architecture proposed by researchers at Google [8]. EfficientNet's layers are based on a compound scaling method that uniformly scales the depth, width, and resolution of the network, with the stem layer being the initial part of the network that performs initial convolutions to process the input image before deeper layers. The "Stem" in a neural network, particularly in the context of architectures like EfficientNet, refers to the initial set of layers that process the input data before it passes through the main body of the network. The compound scaling method ensures that the network becomes more efficient as it grows larger, balancing the trade-off between model size and accuracy.

VGG-16

VGG-16 is a type of CNN architecture proposed by the Visual Geometry Group at the University of Oxford [9]. The 16 in VGG16 refers to 16 layers that have weights. In VGG16 there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers i.e., learnable parameters layer. VGG-16 has a fixed architecture with 13 convolutional layers and 3 fully connected layers. Each convolutional block contains multiple 3x3 convolutional layers followed by a max-pooling layer. It is computationally expensive and requires a large number of parameters.

Xception

Xception stands for "Extreme Inception", it is an improved version of Inception and it was developed by Google[10]. It is based on the idea of depthwise separable convolutions, which separate the process of learning spatial patterns from learning channel-wise relationships. The data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow. Xception architecture is an extension of the Inception architecture but replaces standard convolutional layers with depthwise separable convolutions. Depthwise separable convolutions consist of two steps: depthwise convolutions and pointwise convolutions. This separation reduces computational cost while maintaining expressive power.

EXPERIMENTATION

We have trained three pre-trained models EfficientNetB0, VGG-16, and Xception separately on randomly sampled 1000 images extracted from videos in DFDC. Tabe 1 Represents how many images were used in the training, validation, and testing set.

	Fake images	Real images	Total images
Training set	350	350	700
Validation set	100	100	200
Testing set	50	50	100

Table 1: Distribution of Dataset used

Before feeding the input images to the model, the images are pre-processed. Data augmentation techniques are applied to all the images. Data Augmentation brings diverse set images so that the model can generalize better. Two types of augmentation can be performed on images.

Spatial Augmentation includes:

- Scaling
- Cropping
- Flipping
- Rotation
- Translation

Pixel Augmentation includes:

- Brightness
- Contrast
- Saturation
- Hue

The same set of images is used as input to all three models. The shape of the image was set to (224,224,3), the height and width of images are set to 224 which is the standard input size to all the pre-trained CNN models and 3 represents the color channel RGB. The facial landmarks like eyes, nose, mouth, ears, and head are detected from the image, and the distance between each of them is calculated so that the model learns the difference between real and fake images. Figure 4. Represents the overview of the model to detect deepfake images.

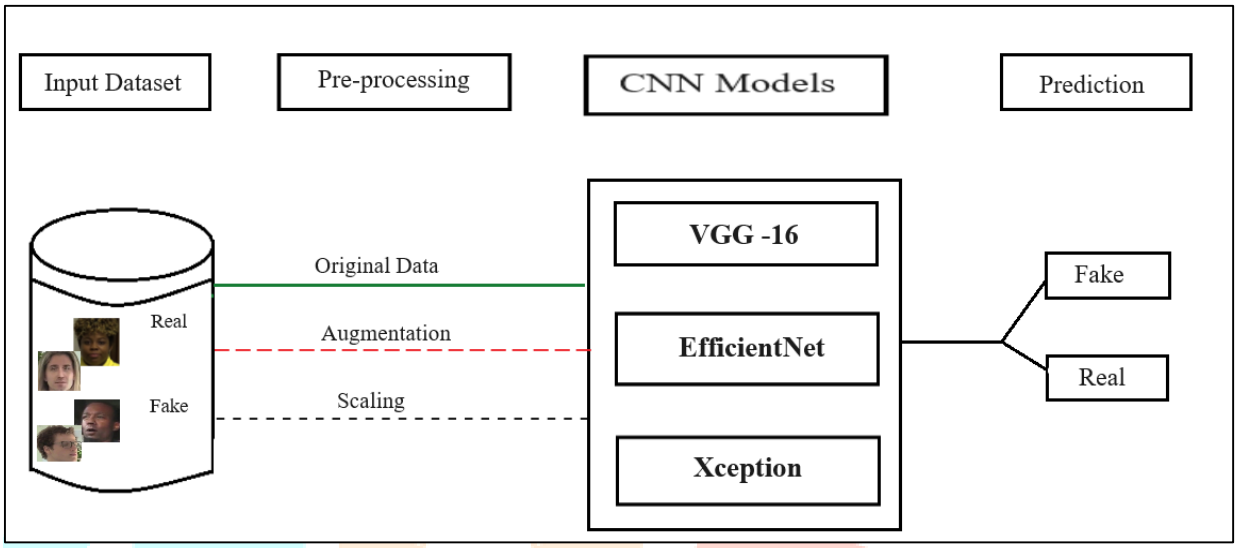

Figure 4: General overview of the model to detect deepfake images

RESULT

The three models were evaluated using a set of four performance metrics: Accuracy, Precision, Recall, and F1 Score.

EfficientNetB0: With an accuracy of 0.83, EfficientNet achieves a commendable precision of 0.85, indicating its ability to accurately classify positive instances. Moreover, it maintains a relatively high recall of 0.8, demonstrating its effectiveness in capturing positive instances within the dataset. The F1 Score of 0.82 underscores EfficientNet's balanced performance across precision and recall, making it a reliable choice for classification tasks.

VGG16: While VGG16 achieves the highest accuracy among the models with 0.88, its precision and recall values are comparatively lower at 0.62 and 0.5, respectively. This indicates that while VGG16 performs well in overall classification accuracy, it may struggle with accurately identifying positive instances, as reflected in its lower precision and recall values. The F1 Score of 0.55 further confirms the model's limitations in achieving a balance between precision and recall.

Xception: Xception exhibits an accuracy of 0.87, similar to VGG16. However, its precision, recall, and F1 Score values are notably lower at 0.46, 0.38, and 0.41, respectively. This suggests that while Xception may accurately classify instances to some extent, it lacks precision in identifying positive instances and has difficulty in capturing all positive instances within the dataset.

Overall EfficientNetB0 emerges as the most balanced and reliable model among the three, showcasing consistent performance across accuracy, precision, recall, and F1 Score metrics. VGG-16, while achieving high accuracy, demonstrates shortcomings in precision and recall. Xception, despite its comparable accuracy, exhibits lower precision, recall, and F1 Score values, indicating areas for improvement in classification performance. These findings can guide future research and inform decision-making regarding model selection and optimization strategies in deep learning applications.

CNN Models	Accuracy	Precision	Recall	F1 Score
EfficientNetB0	0.83	0.85	0.8	0.82
VGG16	**0.88**	0.62	0.5	0.55
Xception	0.87	0.46	0.38	0.41

Table 2: Performance metrics of EfficientNetB0, VGG-16, and Xception models

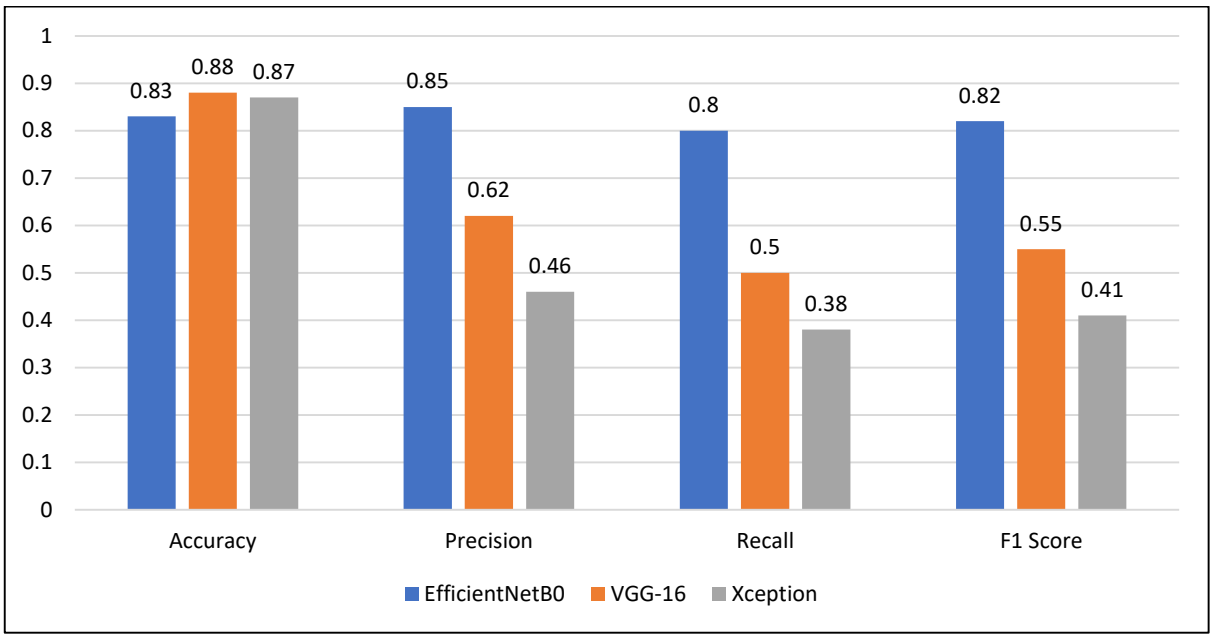

Figure 5: Analysis of performance metrics of EfficientNetB0, VGG-16, and Xception models

Confusion matrix was also used to assess the three models' performance. It provides a detailed breakdown of the model's predictions, allowing for a deeper analysis of classification accuracy, misclassifications, true positives, true negatives, false positives, and false negatives.

Figure 6,7,8 are the Confusion matrix of EfficientNetB0, VGG-16, and Xception models. In summary, EfficientNetB0 appears to be the most balanced model among the three, demonstrating a good balance between true positives, false positives, false negatives, and true negatives. VGG-16 shows a higher false positive rate, while Xception exhibits higher false positive and false negative rates, indicating areas where these models may require improvement in classification performance.

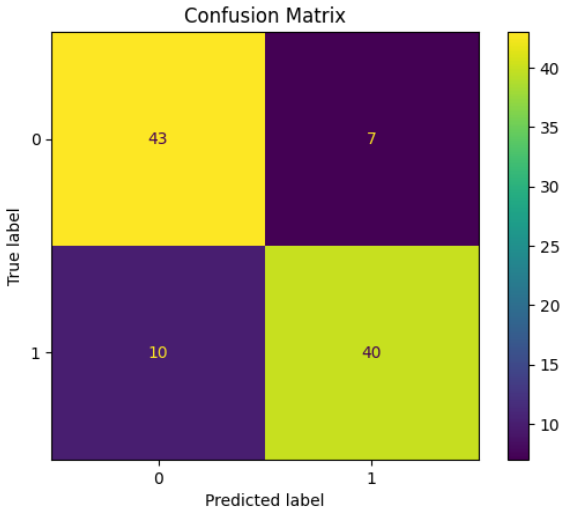

Figure 6: Confusion matrix of EfficientNetB0

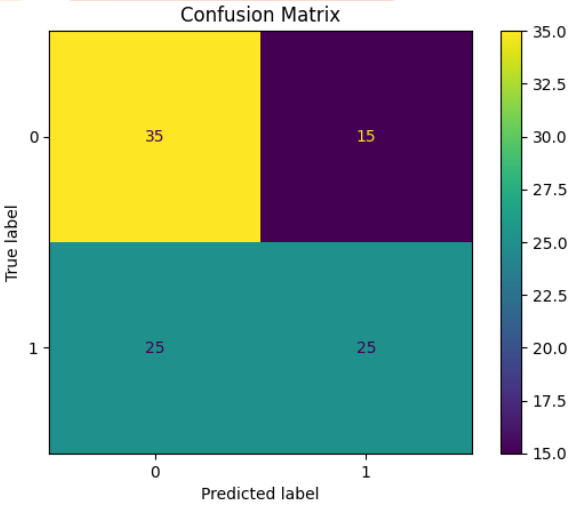

Figure 7: Confusion matrix of VGG-16

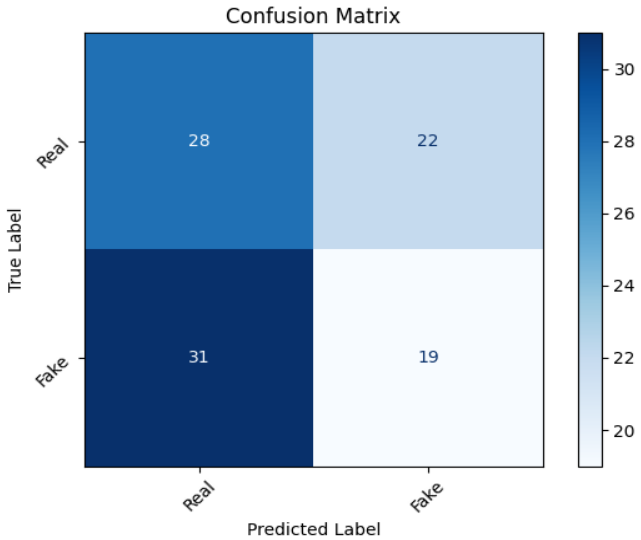

Figure 8: Confusion matrix of Xception

Figure 9 compares the prediction of correctly classified and misclassified images using the three CNN models. EfficientNetB0's balanced performance is further emphasized, with accurate classification and a relatively low number of misclassifications. VGG-16's higher false positive rate indicates a tendency to classify some images incorrectly as positive when they are negative, while Xception's higher false positive and false negative rates suggest challenges in both positive and negative classifications.

Figure 9: CNN Model Comparison: Correct vs. Misclassified Predictions

CONCLUSION

Our research demonstrates the effectiveness of EfficientNetB0, VGG-16, and Xception models for detecting deepfake images with minimal parameter tunning of these pre-trained models. Even though we have used only small set of images due to computational constraints, we achieved commendable performance. A diverse set of images from different datasets can be used which were created using different techniques for better generalization of the deepfake detection model. With the ability to accurately predict the authenticity of unseen images, our model presents a promising solution to detect deepfake images. Future research may use ensemble and fusion methods and apply advanced data augmentation techniques to further enhance model's performance. Likewise, it's essential to think about the ethical implications of deepfake technology and the impact on humanity of creating trustworthy detection techniques to prevent against any kind of misuse.

REFERENCES

[1] Yogesh & Tanwar, Sudeep & Bhattacharya, Pronaya & Gupta, Rajesh & Alsuwian, Turki & Davidson, Inno & Mazibuko, ThokoZile. (2023). An Improved Dense CNN Architecture for Deepfake Image Detection. IEEE Access. PP. 1109/ACCESS.2023.3251417.

[2] Hanqing, Zhao & Wei, Tianyi & Zhou, Wenbo & Zhang, Weiming & Chen, Dongdong & Yu, Nenghai. (2021). Multi-attentional Deepfake Detection. 2185-2194. 10.1109/CVPR46437.2021.00222.

[3] Nirkin, Yuval & Wolf, Lior & Keller, Yosi & Hassner, Tal. (2020). DeepFake Detection Based on the Discrepancy Between the Face and its Context.

[4] Sohail Ahmed Khan and Duc-Tien Dang-Nguyen. (2022). Hybrid Transformer Network for Deepfake Detection. https://doi.org/10.1145/3549555.3549588

[5] Munir, Kashif & Raza, Ali & Almutairi, Mubarak. (2022). A Novel Deep Learning Approach for Deepfake Image Detection. Applied Sciences. 12. 10.3390/app12199820.

[6] A. Malik, M. Kuribayashi, S. M. Abdullahi and A. N. Khan, "DeepFake Detection for Human Face Images and Videos: A Survey," in IEEE Access, vol. 10, pp. 18757-18775, 2022, doi: 10.1109/ACCESS.2022.3151186.

[7] Dolhansky, Brian & Bitton, Joanna & Pflaum, Ben & Lu, Jikuo & Howes, Russ & Wang, Menglin & Ferrer, Cristian. (2020). The DeepFake Detection Challenge Dataset.

[8] Tan, Mingxing & Le, Quoc. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.

[9] Tammina, Srikanth. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. International Journal of Scientific and Research Publications (IJSRP). 9. p9420. 10.29322/IJSRP.9.10.2019.p9420.

[10] Chollet, Francois. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. 1800-1807. 10.1109/CVPR.2017.195.