*Detection of Image and Video Manipulation*

Ankit Basrur   
Information Technology  
Dwarkadas J. Sanghvi College of EngineeringMumbai,India  
ankit.basrur@djsce.edu.in

Dr. Vinaya Sawant   
HOD of Information Technology  
Dwarkadas J. Sanghvi College of EngineeringMumbai,India  
vinaya.sawant@djsce.ac.in

Devansh Mehta   
Information Technology  
Dwarkadas J. Sanghvi College of EngineeringMumbai,India  
devansh.mehta1@djsce.edu.in

Dhrumil Mehta  
Information Technology  
Dwarkadas J. Sanghvi College of EngineeringMumbai,India  
dhrumil.mehta@djsce.edu.in

*Abstract*— Deepfakes are increasingly exponentially which can be a threat to privacy of an individual. Deepfake creation is easily done by layman due to availability in various tools available, which can be used for the defamation of an individual/organization/community. Therefore, different methods have been introduced to address this issue. Early methods were based on the inconsistencies of fake video obtained by using handcrafted features. Recent methods, on the other hand, applies deep learning to automatically extract prominent and discriminatory features to detect deepfakes. This paper proposes detecting Deepfake using XceptionNet and DenseNet for images and videos respectively.

Keywords— Deepfake Detection, Image Processing, Neural Networks, Machine learning, Deep learning.

# Introduction

The increasing growth of mobile camera technology and social media have creation and sharing of images and videos more convenient. The number of fake images as well as videos are increasing due to high level of editing tools available on Internet. The time required in manipulation of the data available to the person has decreased significantly due to Deepfake creation tools which are easily available. The knowledge of ML and computer vision decreases the manual steps required for image or video editing. This has led to an increase in spreading replicated images and videos (Deepfake). Deepfakes are synthetic media in which a person in an existing image or video is replaced with someone else's likeness.

Convolutional Neural Networks (CNNs) have gained much popularity in recent years for vision-related applications and have the potential to achieve high accuracy. Different type of Generative Adversarial Network (GAN) algorithm is used for creation of fake images, which makes it difficult to differentiate between real and fake images. In this paper, we have used separate architecture for detection of Deepfake in images and for videos. We have used MTCNN for face detection followed by face alignment using dlib. We trained models using different backbone networks, XceptionNet and DenseNet, which can be used for images and videos.

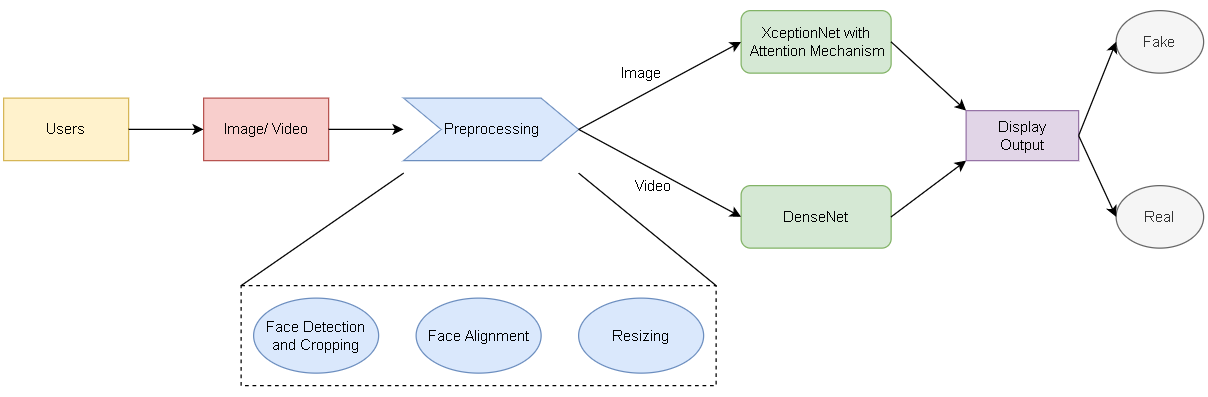
# SCOPE

This system is generally user friendly as the user only has to upload image/video files and this will also be feasible to use on laptop as well as your smartphones. The solution is basically based on the detection of the images and videos and is based on two Algorithms that will give the user results of detection at the end of the process based on the fake probability. The image will first be checked that if the file will be able to upload, if yes then the process will go forward else the message will be shown that the file is not detected. Each step of this process have a significance in itself as for pre-processing the images will be first aligned by aligning the eyes of the person and then going through some parameters the system will go through the algorithm .At the end of process the system will give the result and it will give fake probability and if the result says that the image is real it does not go through frame by frame distribution. If the image is fake then the user can go through the next step where the probability of the image being fake is given for a person .

# Proposed system design

Since deepfakes can be easily generated, it becomes essential to stop inappropriate news from spreading which can cause a huge damage. Hence, a novel and helpful way to deal with detection of low- and high-quality pictures utilizing profound learning methods is recommended. An image or video is taken as input from user in real time. The proposed arrangement would be a web application/online interface. The utilization of neural networks and image processing is illustrated. Processed media is then sent to our networks depending upon its type and the results are shows on each frame as output.

Fig. 1. below is the system architecture of our proposed system. It includes abstract relations with proposed features.



1. System Architecture

In the first stage, we convert video into sequence of frames. Our aim was to find all faces in a given frame. Post that, we subjected the images to some preprocessing algorithms such as MTCNN and dlib.

After initial preprocessing of all images, feature extraction and deep feature learning is performed. At the end of our network, we have used sigmoid function which get a number between 0 and 1 for each output.

# DATASET

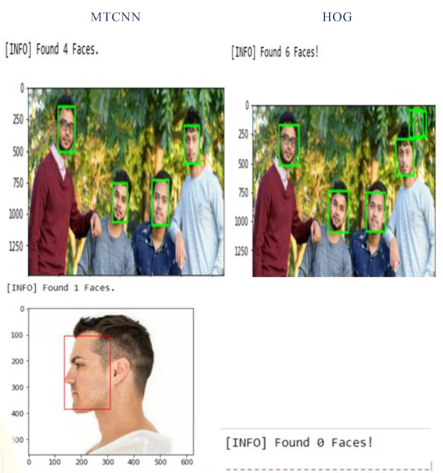
Our dataset is a subset of DFFD dataset. It is mainly classified into fake and real. The dataset consists of over 12,000 images in the train set and 6000 images in the test set. Each image is subject to preprocessing, deep feature learning and extraction, and image classification. These images are of high quality and pristine. Making our model train on such dataset can be helpful when input cannot be easily distinguished. Each image goes through the above-mentioned stages, and at each stage, the output after the process is displayed.

# IMPLEMENTATION

## Image Preprocessing

The preprocessing step includes Face Detection, alignment and rescaling it. Various algorithms were implemented to find the best fit for our system. For Face Detection the following algorithms were implemented, Histogram of Oriented Gradients (HOG) and Multitask Cascade Convolutional Neural Network (MTCNN).

* **Histogram of Oriented Gradients (HOG)**: HOG uses Haar Cascade Classifier for detecting and then cropping the face region out of the given input. This face detection is based on Sliding Window Classifier. A geometric face region is formed with the detection of eyes performed using the Haar Cascade Classifier, while nose detection has been used as a confirmation tool with the detection of eyes. It does not work well for non-frontal faces as described in [1].
* **Multitask Cascade Convolutional Neural Network (MTCNN)**: MTCNN is a neural network which detects faces and facial landmarks on images. It is relatively newer method of face detection. This technique consists of 3 neural networks which are connected in cascade, first detects the bounding boxes of faces in an image along with 5 features of facial landmarks. Each NN passes its inputs to a CNN. Bounding boxes with scores are returned. In the final stage, it computes 5 features of facial landmarks along with the confidence value [2]. MTCNN is highly accurate for detecting non-frontal faces making it ideal for our use case.



1. *Image subjected to various processing methods*

Face alignment is an important process to reduce irregularities so that our model is consistent and trains faster. We have used OpenCV and dLib for getting the coordinates of eyes and nose. It consists of following steps [3]: Initially we have to detect the face and eyes after which centre of eyes is found. Connecting those two points and forming a right-angled triangle will help us in finding the angle. The last step is rotating the image by that angle using OpenCV.



Fig. 3. *Face Alignment using OpenCV*

Table 1 *Face detection models*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model name** | **Method** | **Bounding Box Shape** | **Time** | **Downsides** |
| HOG | Sliding window classifier | Square | Least | Bounding box always a square which leads to over high lightening of region.  Cannot detect non-frontal faces |
| MTCNN | Cascade of Neural Networks | Arbitrary rectangle | Highest | Consist of 3 neural networks thus takes more time. |

## Classification

* **XceptionNet with Attention Mechanism:** XceptionNet is a neural network which uses residual connections and is pretrained on ImageNet, fine-tuned on our dataset. It moves the weights for learning features that are more beneficial for detection i.e., objects, skin color alteration, distortion, etc [4][5]. XceptionNet with replacement of the final fully connected layer to 2 outputs and proved to be the best network for detection. We chose XceptionNet because it has two advantages: It makes the model to look for more useful features. Second, using it along with Attention Map to detect patches within image and the attention-based layer used sigmoid function. The dimension of input image was changed so that it fits the input of our network i.e. 299 x 299. We generate face mask using the features obtained in the earlier layer which helps model to learn in a better manner. XceptionNet with replacement of the final fully connected layer to 2 outputs and proved to be the best network for detection [4] and then adding Attention Model between block 4 and 5 in XceptionNet for better results.
* **DenseNet:** DenseNet became popular because of efficient back-propagation and concept of concatenating to reduce the size of feature map for deeper layers [6][7]. It consists of Batch Normalization, DenseBlock and Transition Layer. It extracts feature from selected regions of the face and collectively it adds the feature to the backbone network which acts as learner [7]. It reduces the number of parameters to be passed to the next layer and each layer has access to gradients from loss function which helps in better back propagation. We have changed the input size to 224x224. After which we freeze the initial layers of network during training. Training of network was divided into two parts viz., freezing the initial layer for 10 epochs and unfreezing and training the final layers for 5 epochs. It was observed that the training speed for DenseNet was much faster than that of XceptionNet.

Both the networks were trained for 15 epochs using Adam optimizer with learning rate as 1e-4 with an early stopping condition to avoid overfitting.

When a new image is fed into the model for the prediction it gives the results in the probabilistic results. The study was based on a dataset that included 18000 images.

# experimental results

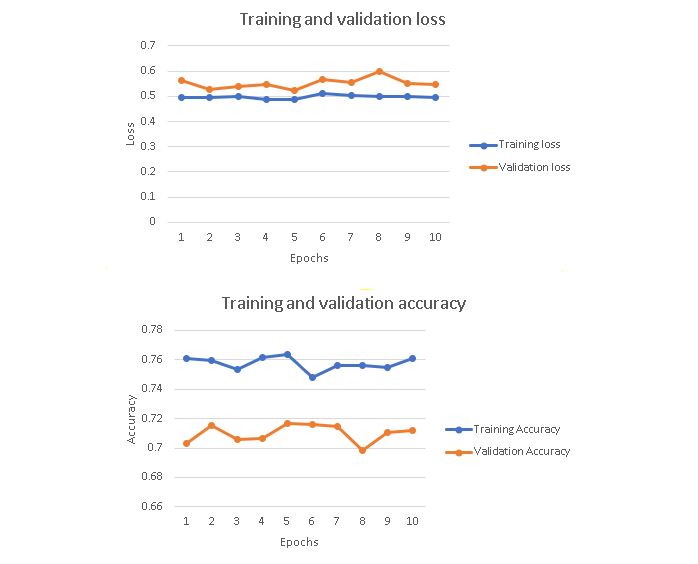
* **XceptionNet Analysis:** On analysis we found out that the accuracy of XceptionNet and DenseNet is almost same. The reason being the size of our dataset which was comparatively less for neural network to learn new features. In the last few epochs the training and validation accuracy did not improve which means model lost it's ability to distinguish between real and fake image after being trained again and again on the same dataset. The validation loss decreased in the final epoch.



1. Training Loss and Accuracy for XceptionNet

* **DenseNet Analysis:** As we can see there is huge difference between training and validation accuracy which means that the model is in overfitting state. By increasing our training dataset size and by using batch normalisation it is possible to prevent our model from going to this condition.

Also, the training accuracy kept on increasing in the final layers whereas validation accuracy stopped improving after 10th epoch. As compared to XceptionNet, the loss in DenseNet is more. Training and Validation Loss did not improve as the training continued. So we feel that XceptionNet, in turn produced better performance.



1. Training Loss and Accuracy for DenseNet

# conclusion

The proposed system is significantly useful for detecting deepfake. The presented system is accurate, efficient, low cost, portable and user friendly so that any user can access it. Based on our study we have used MTCNN for pre-processing as it is capable to detect non-frontal faces. For the detection purpose DenseNet and XceptionNet is used for video and image detection respectively as DenseNet can identify differences between two frames. In future, the system can be improved by training on larger dataset and to reduce false positive ratios.

After studying and analyzing different algorithms similar to our project we conclude that it is possible to implement this project. And from technical, operational and economic feasibilities we can conclude that the system is possible to implement.

# FUTURE SCOPE

We are currently trying to run this model on a bigger dataset which may help in improving efficiency of our model. The detection is based on the discrepancies found on single frame but we did not detect the alignment of audio with input video. Detecting on the basis of audio and training our model on a larger dataset can be helpful in understanding more about ways in which DeepFake can be detected.

##### References

1. sukeshroy, “Face Detection and Recognition Using OpenCV,” [Online]. Available: https://www.datasciencelearner.com/face-detection-recognition-python-hog-tutorial/. [Accessed 7 August 2020].
2. K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks," *IEEE,* 2016.
3. V. Kovenko, "How to precisely align face in Python with OpenCv and Dlib," Towards Data Science, 11 August 2019. [Online]. Available: https://towardsdatascience.com/precise-face-alignment-with-opencv-dlib-e6c8acead262. [Accessed 26 August 2020].
4. A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images," *2019 IEEE/CVF International Conference on Computer Vision (ICCV),* 2019.
5. M. Koopman, A. M. Rodriguez and Z. Geradts, "Detection of Deepfake Video Manipulation," Proceedings of the 20th Irish Machine Vision and Image Processing conference, 20 8 2018.
6. G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *arXiv CVPR 2017,* vol. 5, 2018.
7. E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi and P. Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos," arXiv, 16 May 2019.