Deepfake Detection Using Deep learning

by

Kammari Santhosh

Roll. No.: 2018IMT-043

ABV–INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT GWALIOR (M.P.), INDIA

TABLE OF CONTENTS

- INTRODUCTION
- MOTIVATION
- LITERATURE SURVEY
- OBJECTIVES
- SYSTEM ARCHITECTURE
- EXPERIMENTAL ANALYSIS AND RESULTS
- TASKS TO BE COMPLETED
- REFERENCES

Introduction I

- Deep fakes are created by combing and superimposing existing images and videos onto source images or videos using a deep learning technique known as generative adversarial network.
- Deepfake systems typically need various photos and video data to develop models to create realistic videos and images.
- Higher deep networks and freely accessible data have made fake videos and images almost equal to humans and even advanced computer algorithms.
- However, when the concept of neural networks grew big, and understanding improved its computational abilities, people started using this technology in their daily lives.
- Researchers chose to find methods of deep face detection to help people from this danger.
- Video manipulation and Digital image technologies are developing quickly for many decades.

Motivation I

1. As deepfakes are fake videos generated using some digital software like faceswap,faceit,deeplab etc.in which computer creates an AI video which lead to new video that shows the actions that wasnât happened.but the results looks like acceptable.they are so difficult to identify that they are false.
2. Developing these techniques made a great impact on social and politics.The real danger is because of false information it creates mistrust or lack of interest in people about what they hear in online.

Motivation II

These problems may leads to worst conditions like

- Fake News
- Malicious hoaxes
- Financial fraud
- Politician videos.
- facial reanimation

are some of them this is why deepfake detection is needed.

Literature Survey

Author	Title	Year	Publish	Work
Karnouskos, Stamatis	Digital Media: The Era of Deepfakes	2020	IEEE	considering the how deepfakes effects the media
Nguyen, Thanh and Nguyen, Cuong M. and Nguyen	Deep Learning for Deepfakes Creation and Detection: A Survey	2019	IEEE	the objective is survey of how to detect and create a deep fake
Meng, Cunyan and Zhang, Xinghui	Video Encryption Based on OpenCV	2010	IEEE	the objective here is to generate a dataset from input video using opencv

Literature Survey

Author	Title	Year	Publish	Work
Morozov,Artem A.,Maksutov, and	Deepfake Detection Methods using Machine Learning	2020	IEEE	the objective here is to create fake video using gan and model experiments
Piotr Kawa and P. Syga	Deepfake Detection with Low-Resources	2020	IEEE	deep fake detection experimentst using mesonet-4 and mesonet-inception -4
Pokroy, Artem A. and Egorov, Alexey D	DeepFake Detection: Comparing Pretrained Models	2021	IEEE	the study conducted is to compare the accuracies of various deep network models
Mahamkali, Naveenku-	OpenCV for CV Applica-	2015	IEEE	feature detection&object tracking using opencv

OBJECTIVES

1. In this system,a pretrained model is used on training set and model with best accuracy is used to determine wheather the video is fradulent or nonfraudulent
2. To train the model,dataset must be retrieved and data must be preprocessed in following steps:
   - extract frames from video
   - filtering the extracted frames
   - splitting of data into train and test

System Architecture

The final purpose of this work is to recognize whether a video obtained real or knowing whether it was created using some other deepfake methods. As the input to the model or the system is a video

But, deep learning models do take pictures as input, so there needs to be a change in the input to model from the system. This can be achieved through pre-processing.

System Architecture Cont..

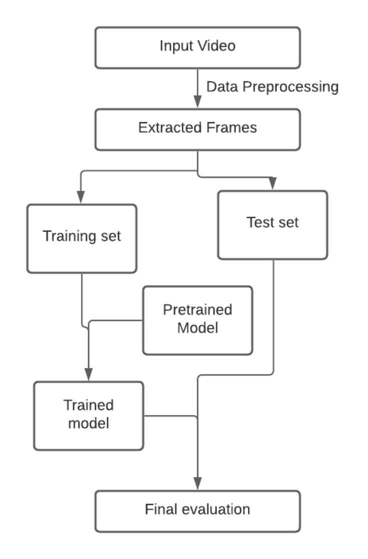


Figure: Basic Architecture

System Architecture Cont..

Data Preprocessing:
module need to check how other factors impact on the model training.Most of the video frames not only contains faces but also person's body parts and background area of the image.Thus unrelated data can highly impact our model training.
So we need to focus on the face area and our preprocessing module need to capture the face as input.preprocessing consists of three steps:

- collect frames from the video
- detecting faces from the individual images.
- resizing and saving locally.

System Architecture Cont..

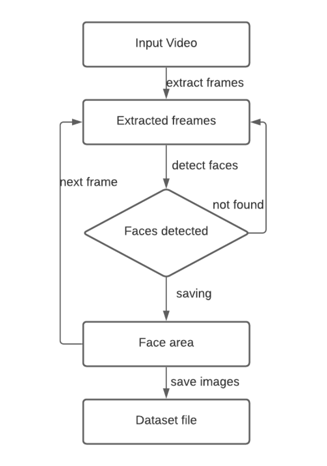


Figure: preprocessing

EXPERIMENTAL ANALYSIS AND RESULTS

Importing the dataset and required libraries,the data is fetched using a opencv.the data is analyzed and preprocess in steps:
1)Extraction of frames: Using opencv tool(imread,imwrite) we can extract the frames from the input video with certain frame rate depending how no of images we want to store.

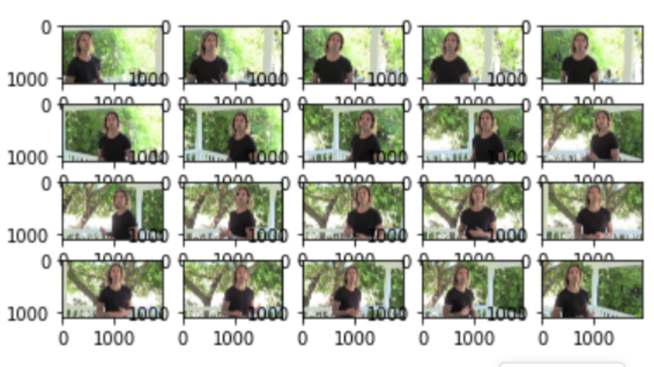
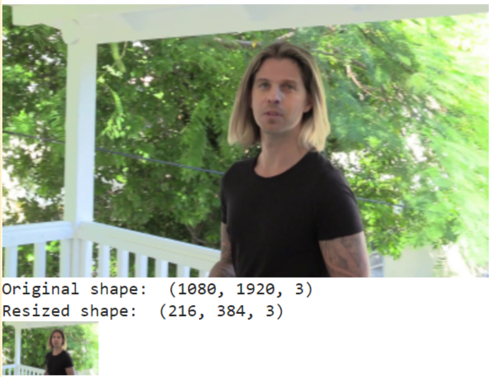


Figure: datase frames

EXPERIMENTAL ANALYSIS AND RESULTS

2)size reduction: the dimensions of each of the image that we have extracted from video are much large, so for better output and to take less time we have scale down each image.

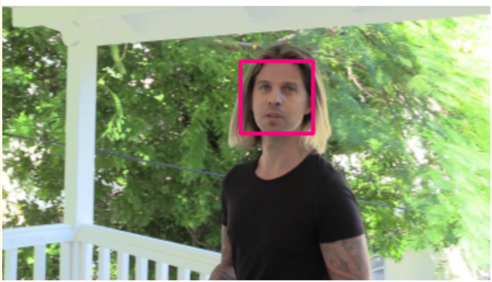


```
Original shape:  (1080, 1920, 3)
Resized shape:  (216, 384, 3)
```

Figure: size reduction

EXPERIMENTAL ANALYSIS AND RESULTS

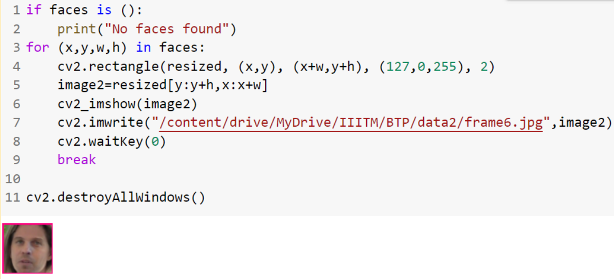
3)face identification: Now that we have resized our image, Apply the CascadeClassifier function, an OpenCV function, to an XML file location for frontal face detection, i.e., haarcascade_frontalface_default.XML, in our project. I have used path in the machine that I have downloaded.
Generally the images are of RGB channel that we see. But OpenCv reads the image in BGR format, so we have to convert into gray channel because it is simple to process also takes less time for computing as it holds just one channel of white-black.



EXPERIMENTAL ANALYSIS AND RESULTS

With the help of function detectMultiScale,we get the coordinates of face,as x-coordinate,y-coordinate,width and height.With the help of x and y coordinated a sqaure is drawn.

4)modifying the data: for an image if we found a face then we will replace it with updated image else remove the image from dataset which is initially created.



```python
if faces is ():
    print("No faces found")
for (x,y,w,h) in faces:
    cv2.rectangle(resized, (x,y), (x+w,y+h), (127,0,255), 2)
    image2=resized[y:y+h,x:x+w]
    cv2_imshow(image2)
    cv2.imwrite("/content/drive/MyDrive/IIITM/BTP/data2/frame6.jpg",image2)
    cv2.waitKey(0)
    break

cv2.destroyAllWindows()
```

Figure: data modification

Tasks to be completede

1. To study the deep learning model used to detect the deepfake video
2. To implement the code and visualize the output based on input video

References I

(1),(2),(3),(4),(5),(6),(7),(8)

[1] S. Karnouskos, “Artificial intelligence in digital media: The era of deepfakes,” *IEEE Transactions on Technology and Society*, vol. 1, no. 3, pp. 138–147, 2020.

[2] T. Nguyen, C. M. Nguyen, T. Nguyen, D. Nguyen, and S. Nahavandi, “Deep learning for deepfakes creation and detection: A survey,” 09 2019.

[3] C. Meng and X. Zhang, “Video encryption based on opencv,” in *2010 2nd International Workshop on Database Technology and Applications*, 2010, pp. 1–4.

[4] A. A. Maksutov, V. O. Morozov, A. A. Lavrenov, and A. S. Smirnov, “Methods of deepfake detection based on machine learning,” in *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2020, pp. 408–411.

References II

[5] P. Kawa and P. Syga, “A note on deepfake detection with low-resources,” *ArXiv*, vol. abs/2006.05183, 2020.

[6] A. A. Pokroy and A. D. Egorov, “Efficientnets for deepfake detection: Comparison of pretrained models,” in *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, 2021, pp. 598–600.

[7] N. Mahamkali and V. Ayyasamy, “Opencv for computer vision applications,” 03 2015.

[8] D. GÃ¼era and E. J. Delp, “Deepfake video detection using recurrent neural networks,” in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1–6.