computer vision project

By: rahmoun oussama

Technical Keywords

- Deep learning
- Computer vision
- Res-Next Convolution Neural Network
- Long short-term memory (LSTM)
- OpenCV
- Face Recognition
- GAN (Generative Adversarial Network)
- PyTorch.

1-Problem introduction

1-Problem Statement

Convincing manipulations of digital images and videos have been demonstrated for several decades through the use of visual effects, recent advances in deep learn- ing have led to a dramatic increase in the realism of fake content and the accessibility in which it can be created. These so-called AI-synthesized media (popularly referred to as deep fakes).Creating the Deep Fakes using the Artificially intelligent tools are simple task. But, when it comes to detection of these Deep Fakes, it is major chal- lenge. Already in the history there are many examples where the deepfakes are used as powerful way to create political tension[14], fake terrorism events, revenge porn, blackmail peoples etc.So it becomes very important to detect these deepfake and avoid the percolation of deepfake through social media platforms. We have taken a step forward in detecting the deep fakes using LSTM based artificial Neural network.

2-Goals and objectives

Goal and Objectives:

- Our project aims at discovering the distorted truth of the deep fakes.
- Our project will distinguish and classify the video as deepfake or pristine.

2-Methodologies of Problem solving

1-Analysis

- Solution Requirement

  We analysed the problem statement and found the feasibility of the solution of the problem.. After checking the feasibility of the problem statement. The next step is the data- set gathering and analysis. We analysed the data set in different approach of training like negatively or positively trained i.e training the model with only fake or real video's but found that it may lead to addition of extra bias in the model leading to inaccurate predictions. So after doing lot of research we found that the balanced training of the algorithm is the best way to avoid the bias and variance in the algorithm and get a good accuracy.

- Solution Constraints

  We analysed the solution in terms of cost,speed of processing,requirements,level of expertise, availability of equipment's.

- Parameter Identified
  1. Blinking of eyes
  2. Teeth enchantment
  3. Bigger distance for eyes
  4. Moustaches
  5. Double edges, eyes, ears, nose
  6. Iris segmentation
  7. Wrinkles on face
  8. Inconsistent head pose
  9. Face angle
  10. Skin tone
  11. Facial Expressions
  12. Lighting
  13. Different Pose
  14. Double chins
  15. Hairstyle

16. Higher cheek bones

2-Development

After analysis we decided to use the PyTorch framework along with python3 lan- guage for programming. PyTorch is chosen as it has good support to CUDA i.e Graphic Processing Unit (GPU) and it is customize-able. Google Cloud Platform for training the final model on large number of data-set.

3-Evaluation

We evaluated our model with a large number of real time dataset which include YouTube videos dataset. Confusion Matrix approach is used to evaluate the accuracy of the trained model.

4-Outcome

The outcome of the solution is trained deepfake detection models that will help the users to check if the new video is deepfake or real.

5-Software Resources Required

Platform :

1. Operating System: Windows 7+
2. Programming Language : Python 3.0
3. Framework: PyTorch 1.4 ,
4. Cloud platform: Google Cloud Platform
5. Libraries : OpenCV, Face-recognition

6-Project task set

Major Tasks in the Project stages are

- Task 1: Data-set gathering and analysis
  This task consists of downloading the dataset. Analysing the dataset and mak- ing the dataset ready for the preprocessing.

- Task 2 : Module 1 implementation
  Module 1 implementation consists of splitting the video to frames and cropping each frame consisting of face.

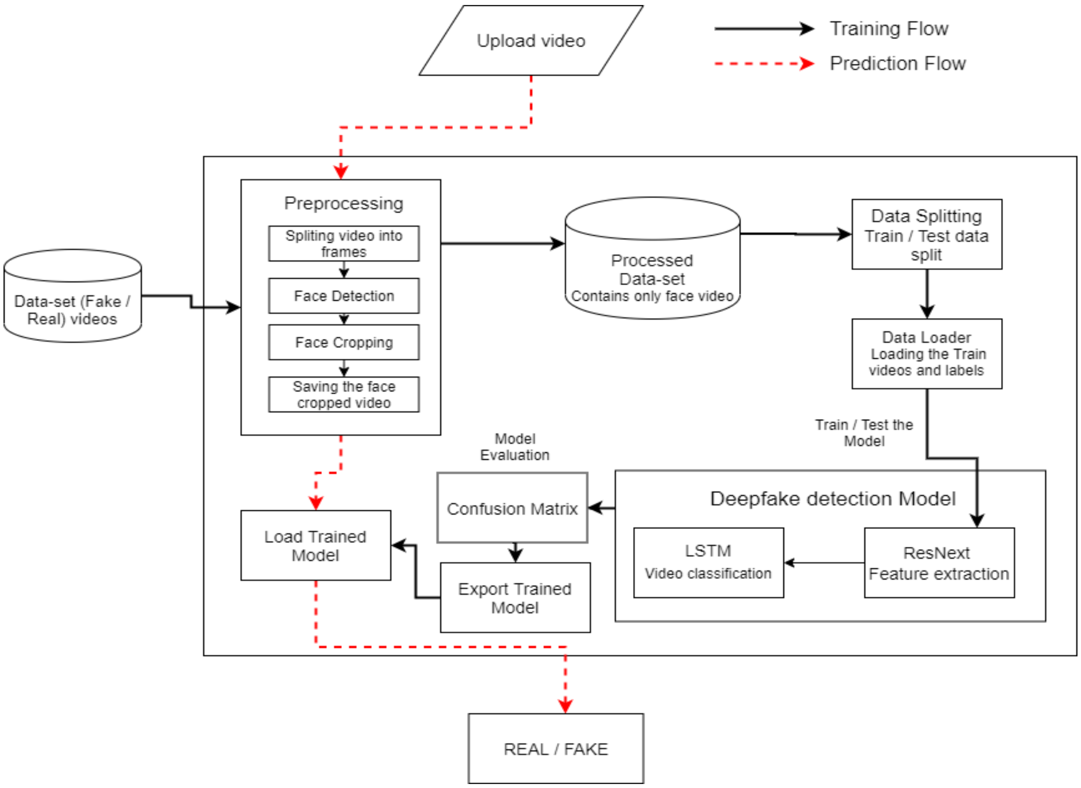
- Task 3: Pre-processing
  Pre-processing includes the creation of the new dataset which includes only face cropped videos.

- Task 4: Module 2 implementation
  Module 2 implementation consists of implementation of DataLoader for load- ing the video and labels. Training a base line model on small amount of data.

- Task 5 : Hyper parameter tuning
  task includes the changing of the Learning rate, batch size, weight decay and model architecture until the maximum accuracy is achieved.

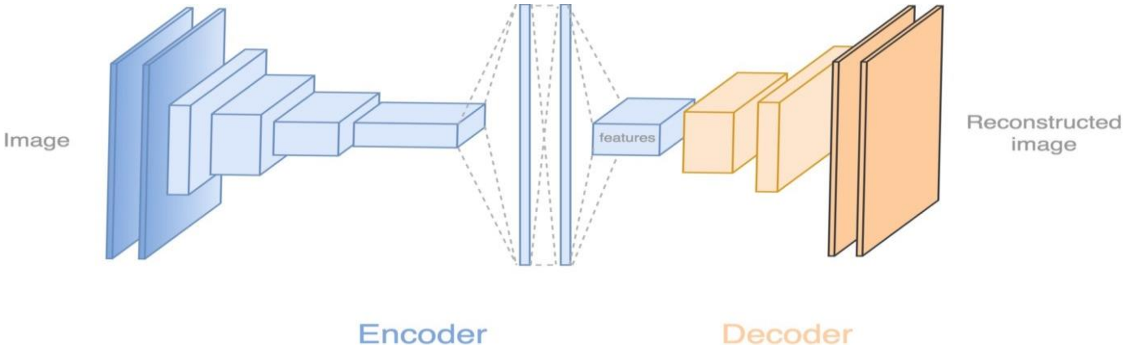
- Task 6 : Training the final model
  The final model on large dataset is trained based on the best hyper parameter identified in the Task 5.

7-System Architecture


Upload video
Training Flow
Prediction Flow
Data-set (Fake / Real) videos
Preprocessing
Spliting video into frames
Face Detection
Face Cropping
Saving the face cropped video
Processed Data-set Contains only face video
Data Splitting Train / Test data split
Data Loader Loading the Train videos and labels
Train / Test the Model
Model Evaluation
Confusion Matrix
Deepfake detection Model
LSTM Video classification
ResNext Feature extraction
Load Trained Model
Export Trained Model
REAL / FAKE

8-How deepfake videos created architecture

To detect the deepfake videos it is very important to understand the creation process of the deepfake. Majority of the tools including the GAN and autoencoders takes a source image and target video as input. These tools split the video into frames , detect the face in the video and replace the source face with target face on each frame. Then the replaced frames are then combined using different pre-trained models. These models also enhance the quality of video my removing the left-over traces by the deepfake creation model. Which result in creation of a deepfake looks realistic in nature. We have also used the same approach to detect the deepfakes. Deepfakes created using the pretrained neural networks models are very realistic that it is almost impossible to spot the difference by the naked eyes. But in reality, the deepfakes creation tools leaves some of the traces or artifacts in the video which may not be noticeable by the naked eyes. The motive of this paper to identify these unnoticeable traces and distinguishable artifacts of these videos and classified it as deepfake or real video.



.

: Deepfake generation

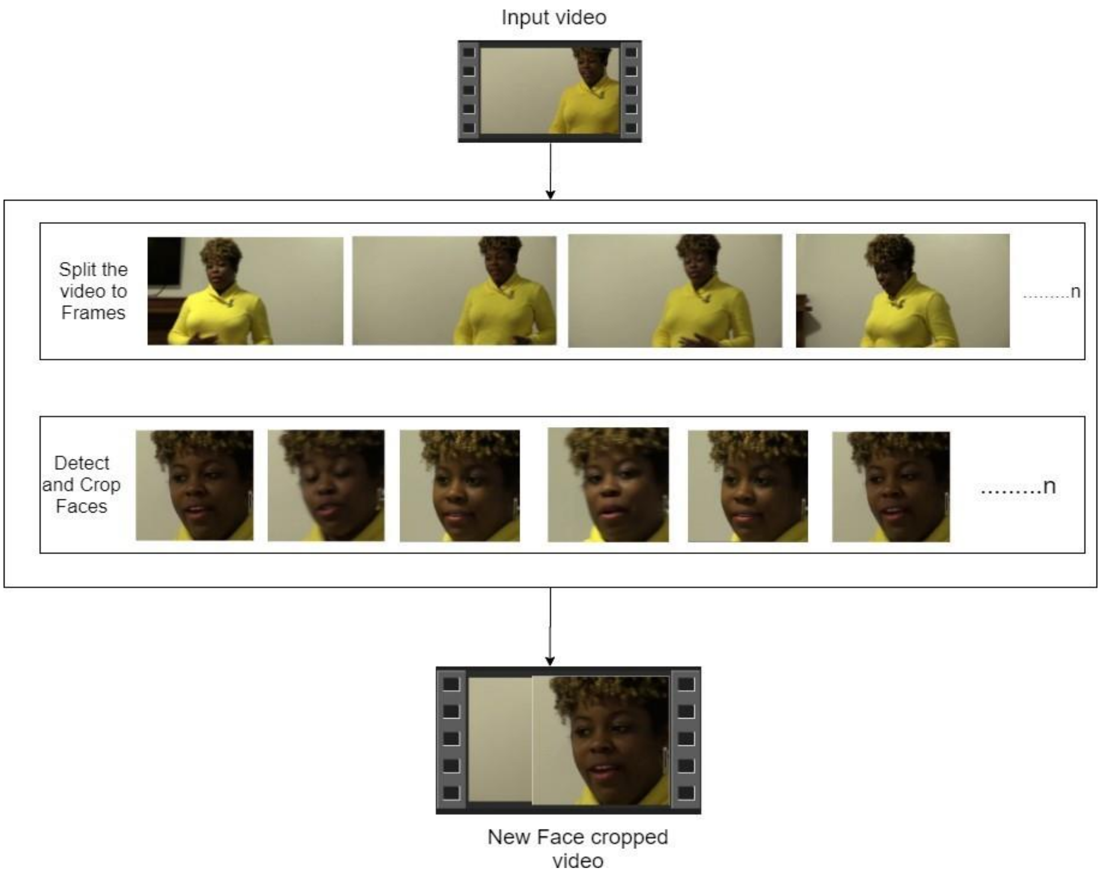
Face Swapped deepfake generation

3-Architectural Design

1-Pre-processing

In this step, the videos are preprocessed and all the unrequired and noise is removed from videos. Only the required portion of the video i.e face is detected and cropped. The first steps in the preprocessing of the video is to split the video into frames After splitting the video into frames the face is detected in each of the frame and the frame is cropped along the face. Later the cropped frame is again converted to a new video by combining each frame of the video. The process is followed for each video which leads to creation of processed dataset containing face only videos. The frame that does not contain the face is ignored while preprocessing.

To maintain the uniformity of number of frames, we have selected a threshold value based on the mean of total frames count of each video. Another reason for selecting a threshold value is limited computation power. As a video of 10 second at 30 frames per second(fps) will have total 300 frames and it is computationally very difficult to process the 300 frames at a single time in the experimental envi- ronment. So, based on our Graphic Processing Unit (GPU) computational power in experimental environment we have selected 150 frames as the threshold value. While saving the frames to the new dataset we have only saved the first 150 frames of the video to the new video. To demonstrate the proper use of Long Short-Term Memory (LSTM) we have considered the frames in the sequential manner i.e. first 150 frames and not randomly. The newly created video is saved at frame rate of 30 fps and resolution of 112 x 112.



2- Data-set split

The dataset is split into train and test dataset with a ratio of 70% train videos and 30% test videos. The train and test split is a balanced split i.e 50% of the real and 50% of fake videos in each split.

3- Model Architecture

Our model is a combination of CNN and RNN. We have used the Pre- trained ResNext CNN model to extract the features at frame level and based on the extracted features a LSTM network is trained to classify the video as deepfake or pristine. Us- ing the Data Loader on training split of videos the labels of the videos are loaded and fitted into the model for training.

ResNext :

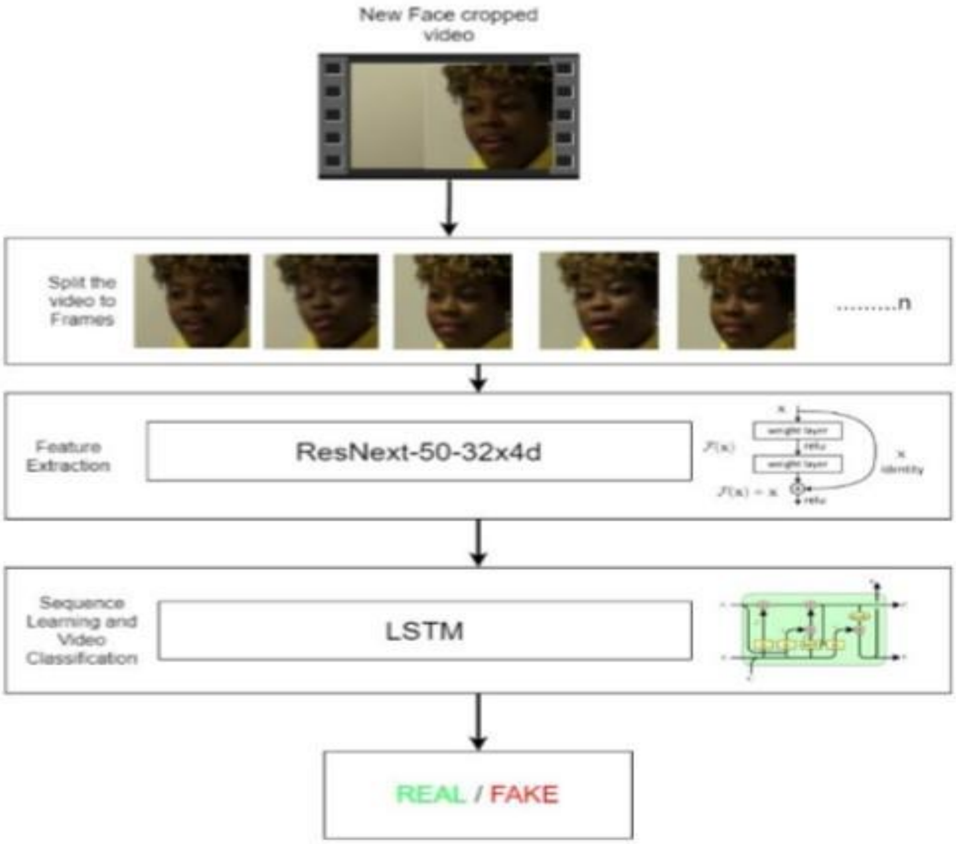
Instead of writing the code from scratch, we used the pre-trained model of ResNext for feature extraction. ResNext is Residual CNN network optimized for high per- formance on deeper neural networks. For the experimental purpose we have used resnext50_32x4d model. We have used a ResNext of 50 layers and 32 x 4 dimen- sions.

Following, we will be fine-tuning the network by adding extra required layers and selecting a proper learning rate to properly converge the gradient descent of the model. The 2048-dimensional feature vectors after the last pooling layers of ResNext is used as the sequential LSTM input.

LSTM for Sequence Processing:

2048-dimensional feature vectors is fitted as the input to the LSTM. We are using 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable to do achieve our objective. LSTM is used to process the frames in a sequential manner so that the temporal analysis of the video can be made, by comparing the frame at ‘t’ second with the frame of ‘t-n’ seconds. Where n can be any number of frames before t.

The model also consists of Leaky Relu activation function. A linear layer of 2048 input features and 2 output features are used to make the model capable of learning the average rate of correlation between eh input and output. An adaptive average polling layer with the output parameter 1 is used in the model. Which gives the the target output size of the image of the form H x W. For sequential processing of the frames a Sequential Layer is used. The batch size of 4 is used to perform the batch training. A SoftMax layer is used to get the confidence of the model during predication.



4- Hyper-parameter tuning

It is the process of choosing the perfect hyper-parameters for achieving the maximum accuracy. After reiterating many times on the model. The best hyper-parameters for our dataset are chosen. To enable the adaptive learning rate Adam[21] optimizer with the model parameters is used. The learning rate is tuned to 1e-5 (0.00001) to achieve a better global minimum of gradient descent. The weight decay used is 1e-3.

As this is a classification problem so to calculate the loss cross entropy approach is used.To use the available computation power properly the batch training is used. The batch size is taken of 4. Batch size of 4 is tested to be ideal size for training in our development environment.

4-Application

1-Introduction

There are many examples where deepfake creation technology is used to mis- lead the people on social media platform by sharing the false deepfake videos of the famous personalities like Mark Zuckerberg Eve of House A.I. Hearing, Don- ald Trump's Breaking Bad series where he was introduces as James McGill, Barack Obama's public service announcement and many more [5]. These types of deepfakes creates a huge panic among the normal people, which arises the need to spot these deepfakes accurately so that they can be distinguished from the real videos.

Latest advances in the technology have changed the field of video manipulation. The advances in the modern open source deep learning frameworks like TensorFlow, Keras, PyTorch along with cheap access to the high computation power has driven the paradigm shift. The Conventional autoencoders[10] and Generative Adversarial Network (GAN) pretrained models have made the tampering of the realistic videos and images very easy. Moreover, access to these pretrained models through the smartphones and desktop applications like FaceApp and Face Swap has made the deepfake creation a childish thing. These applications generate a highly realistic synthesized transformation of faces in real videos. These apps also provide the user with more functionalities like changing the face hair style, gender, age and other attributes. These apps also allow the user to create a very high quality and indistin- guishable deepfakes. Although some malignant deepfake videos exist, but till now they remain a minority. So far, the released tools [11,12] that generate deepfake videos are being extensively used to create fake celebrity pornographic videos or revenge porn [13]. Some of the examples are Brad Pitt, Angelina Jolie nude videos. The real looking nature of the deepfake videos makes the celebraties and other fa- mous personalities the target of pornographic material, fake surveillance videos, fake news and malicious hoaxes. The Deepfakes are very much popular in creating the political tension [14]. Due to which it becomes very important to detect the deepfake videos and avoid the percolation of the deepfakes on the social media platforms.

2-Tools and Technologies Used

Programming Languages

Python3

Programming Frameworks

PyTorch

IDE

1. Google Colab
2. Jupyter Notebook
3. Visual Studio Code

Libraries

1. torch
2. torchvision
3. os
4. numpy
5. cv2
6. matplotlib
7. face_recognition

8. json
9. pandas
10. copy
11. glob
12. random
13. sklearn

3-Model Details

The model consists of following layers:

ResNext CNN : The pre-trained model of Residual Convolution Neural Net- work is used. The model name is resnext50_32x4d()[22]. This model consists of 50 layers and 32 x 4 dimensions. Figure shows the detailed implementation of model.

Sequential Layer : Sequential is a container of Modules that can be stacked together and run at the same time. Sequential layer is used to store feature vector returned by the ResNext model in a ordered way. So that it can be passed to the LSTM sequentially.

LSTM Layer : LSTM is used for sequence processing and spot the temporal change between the frames.2048-dimensional feature vectors is fitted as the input to the LSTM. We are using 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable to do achieve our objective. LSTM is used to process the frames in a sequential manner so that the temporal analysis of the video can be made, by comparing the frame at ‘t’ second with the frame of ‘t-n’ seconds. Where n can be any number of frames before t.

ReLU:A Rectified Linear Unit is activation function that has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. The operation of ReLU is closer to the way our biological neurons work. ReLU is nonlinear and has the advantage of not having any backpropagation errors unlike the sigmoid function, also for larger Neural Networks, the speed of building models based off on ReLU is very fast.

Dropout Layer : Dropout layer with the value of 0.4 is used to avoid over- fitting in the model and it can help a model generalize by randomly setting the output for a given neuron to 0. In setting the output to 0, the cost function becomes more sensitive to neighbouring neurons changing the way the weights will be updated during the process of backpropagation.

Adaptive Average Pooling Layer : It is used To reduce variance, reduce com- putation complexity and extract low level features from neighbourhood.2 di- mensional Adaptive Average Pooling Layer is used in the model.

4-Application:

You will find a example with details in the Jupiter file

Conclusion:

We presented a neural network-based approach to classify the video as deep fake or real, along with the confidence of proposed model. Our method is capable of predicting the output by processing 1 second of video (10 frames per second) with a good accuracy. We implemented the model by using pre-trained ResNext CNN model to extract the frame level features and LSTM for temporal sequence process- ing to spot the changes between the t and t-1 frame. Our model can process the video in the frame sequence of 10,20,40,60,80,100.