DEEPFAKE DETECTION

USING CNN and LSTM

Group Members :-

Pawan Sharma

Ved Prakash Tripathi

Mayank Chauhan

Naman Tripathi

Under the guidance of : -

Dr. Divya Sharma

Outline of This Presentation

- Problem Statement
- Motivation
- Introduction
- Can we detect deepfake with naked eyes?
- How to create Deepfakes
- CNN
- System Architecture
- Data set Exploration
- Pre-processing
- ResNext- CNN
- LSTM
- Training Workflow
- Prediction Workflow
- Results and Confidence of Predictions
- Code Exploration
- Tools and Technologies Used
- References

Problem Statement

To Design and Develop a Deep Learning algorithm to classify the video as Deepfake or real.

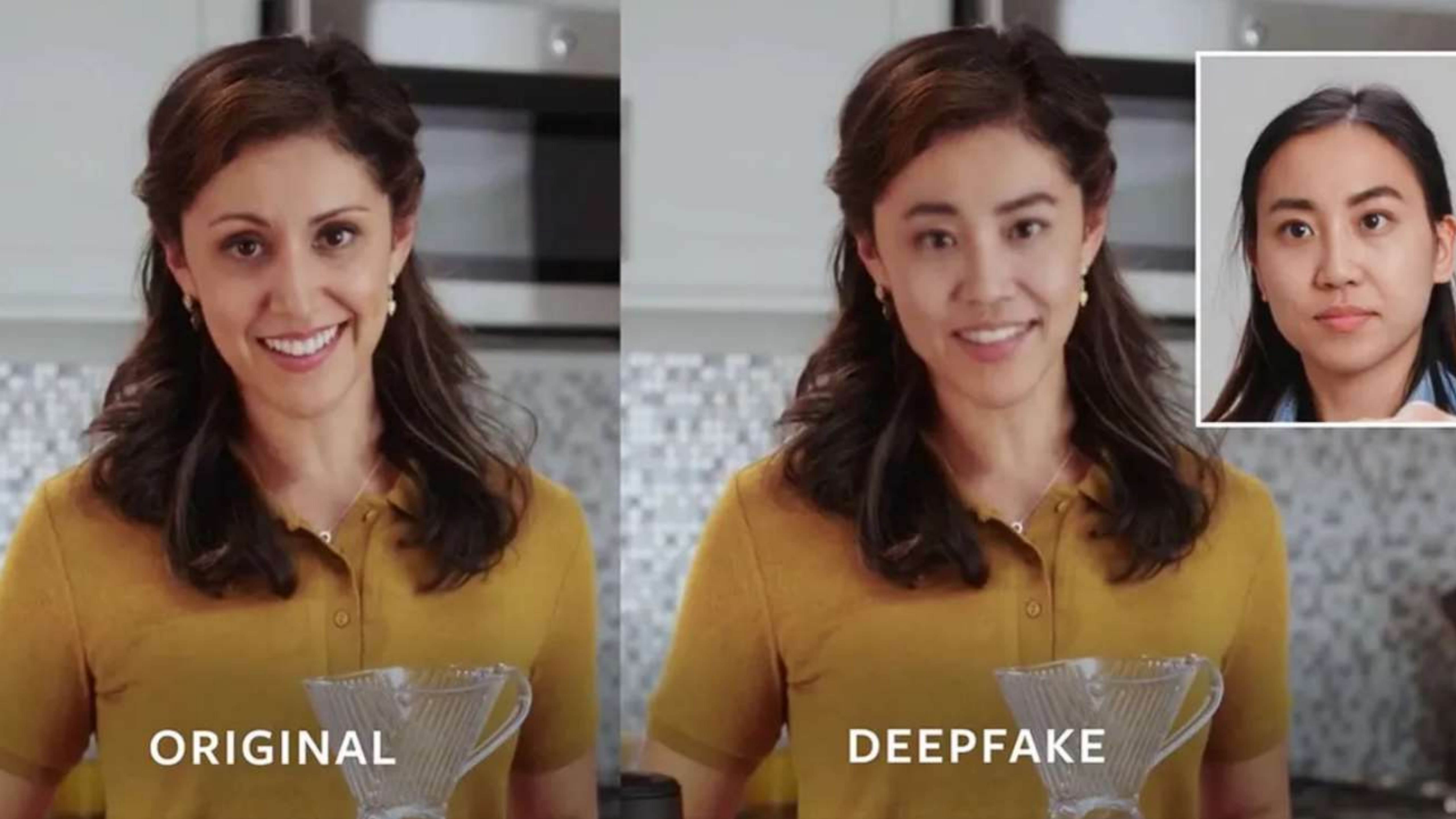
MOTIVATION

- Deepfake technology has the potential to undermine trust in visual media. By developing effective detection methods, we can help ensure that people can rely on the authenticity of images and videos, preserving trust in our digital age.
- Deepfakes can be used to create and spread false information, leading to misinformation campaigns, Fake News, Revenge Porn etc. By detecting and flagging deepfakes, we can contribute to the fight against misinformation, protecting individuals and society from manipulation.
- Deepfake detection research contributes to the advancement of digital forensics capabilities. By developing robust techniques for identifying manipulated media, forensic investigators can more effectively analyze and authenticate digital evidence, supporting legal proceedings and investigations.

❑Introduction

- Deepfake refers to a form of synthetic media created using deep learning techniques, particularly deep neural networks. It involves using artificial intelligence algorithms to manipulate or generate realistic-looking images, videos, or audio that appear to be authentic but are actually fabricated or altered.

- 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.



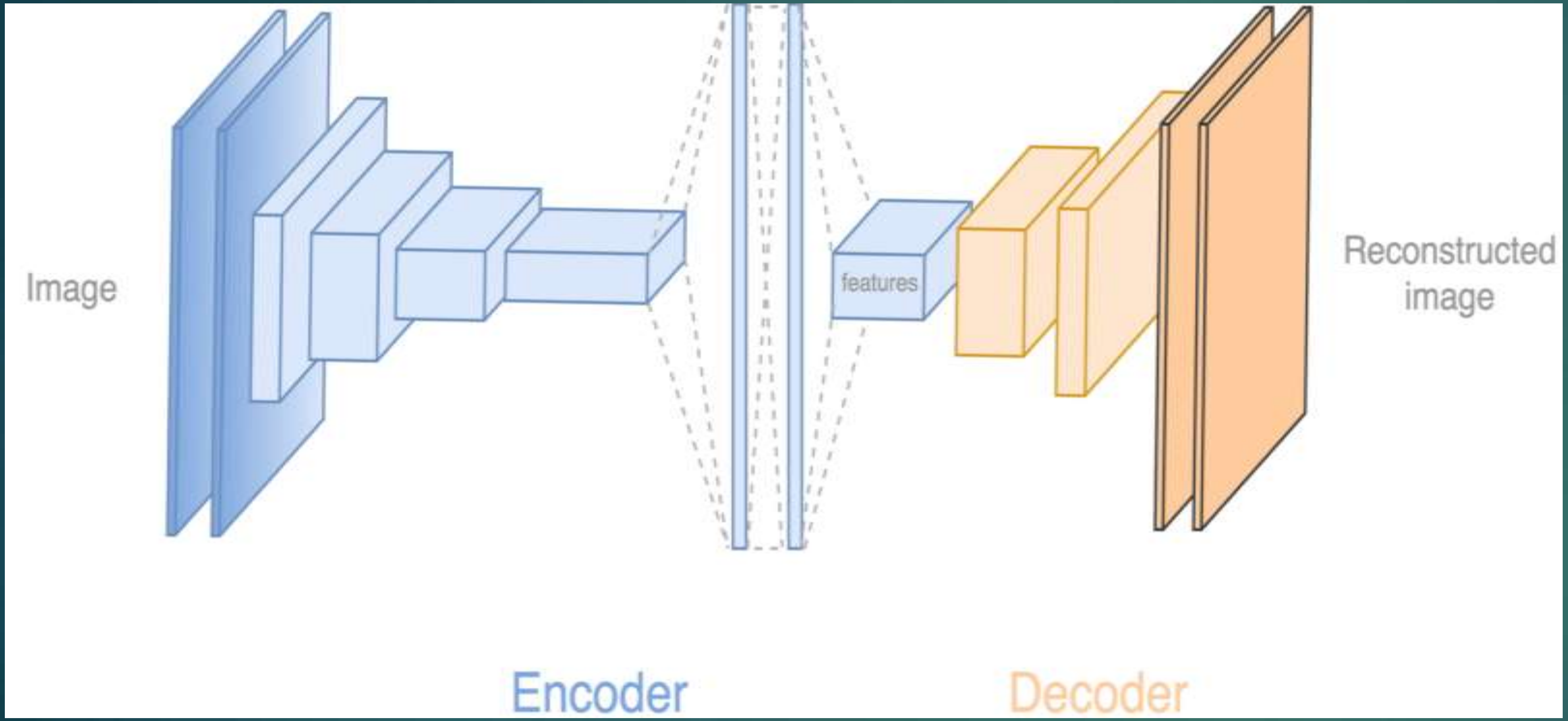
ORIGINAL
DEEPFAKE

❑Can we detect Deep fakes with naked eyes?

- Here is a Video of **Barack Obama** saying inappropriate things about Ex- President **Donald Trump** and many other stuffs.

Click here

❑ How Deep Fakes Are Created ?



Tools for deep fake creation.

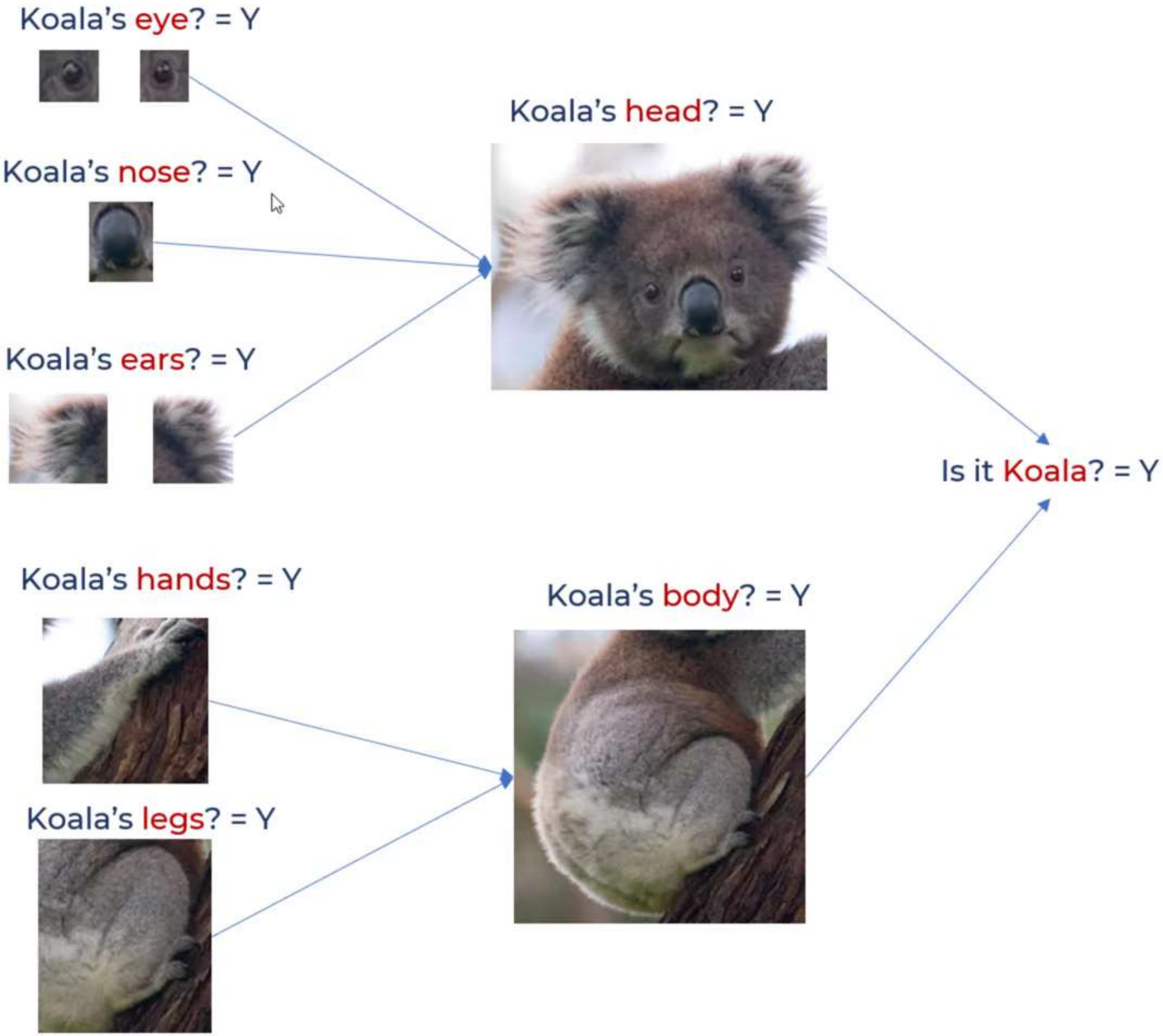
- Faceswap
- Faceit
- DeepFaceLab
- DeepfakeCapsuleGAN
- Large resolution facemasked

❑ Proposed Solution

- Our system uses a Res-Next Convolution Neural Networks to extract frame-level features. These features are then used to train a Long Short Term Memory(LSTM) based Recurrent Neural Network(RNN) to classify whether the video is subject to any kind of manipulation or not, i.e whether the video is deep fake or real video.
- To emulate the real time scenarios and make the model perform better on real time data, we trained our method with large amount of balanced and combination of various available dataset like FaceForensic++ , Deepfake detection challenge , and Celeb-DF .

❑ What is CNN (Convolutional Neural network) ?

- A CNN is a type of computer program that is designed to recognize patterns and objects in images, similar to how the human brain recognizes things.
- It’s a multi-layered neural network that processes the image by passing it through various filters, looking for specific features such as edges, textures, and shapes. These filters allow the CNN to identify the objects or patterns within the image.
- In other words, A CNN is an image-processing tool that can be used to identify and classify objects in images or videos, and extract useful information from them.



Koala's eye? = Y

Koala's nose? = Y

Koala's ears? = Y

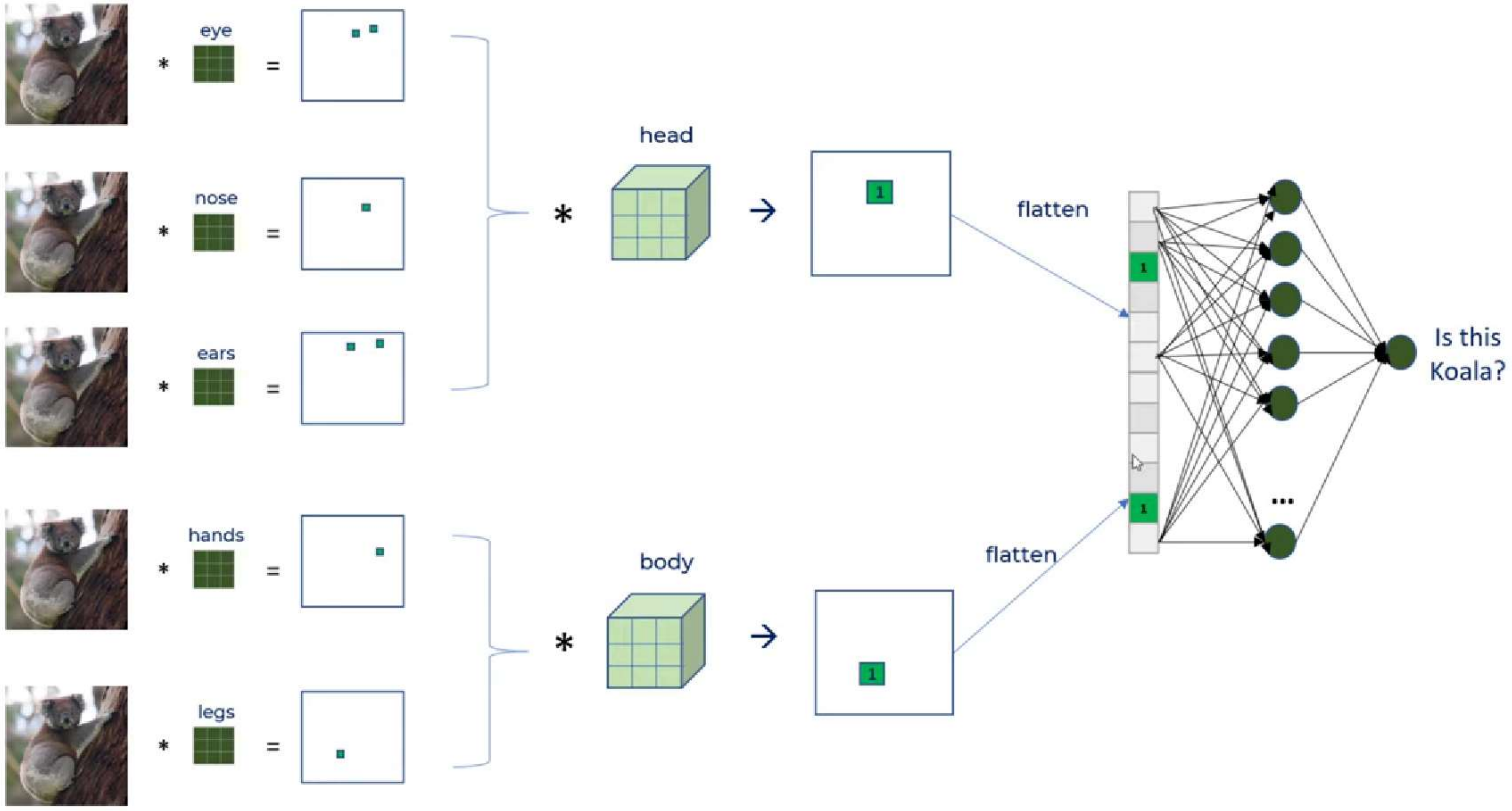
Koala's head? = Y

Koala's hands? = Y

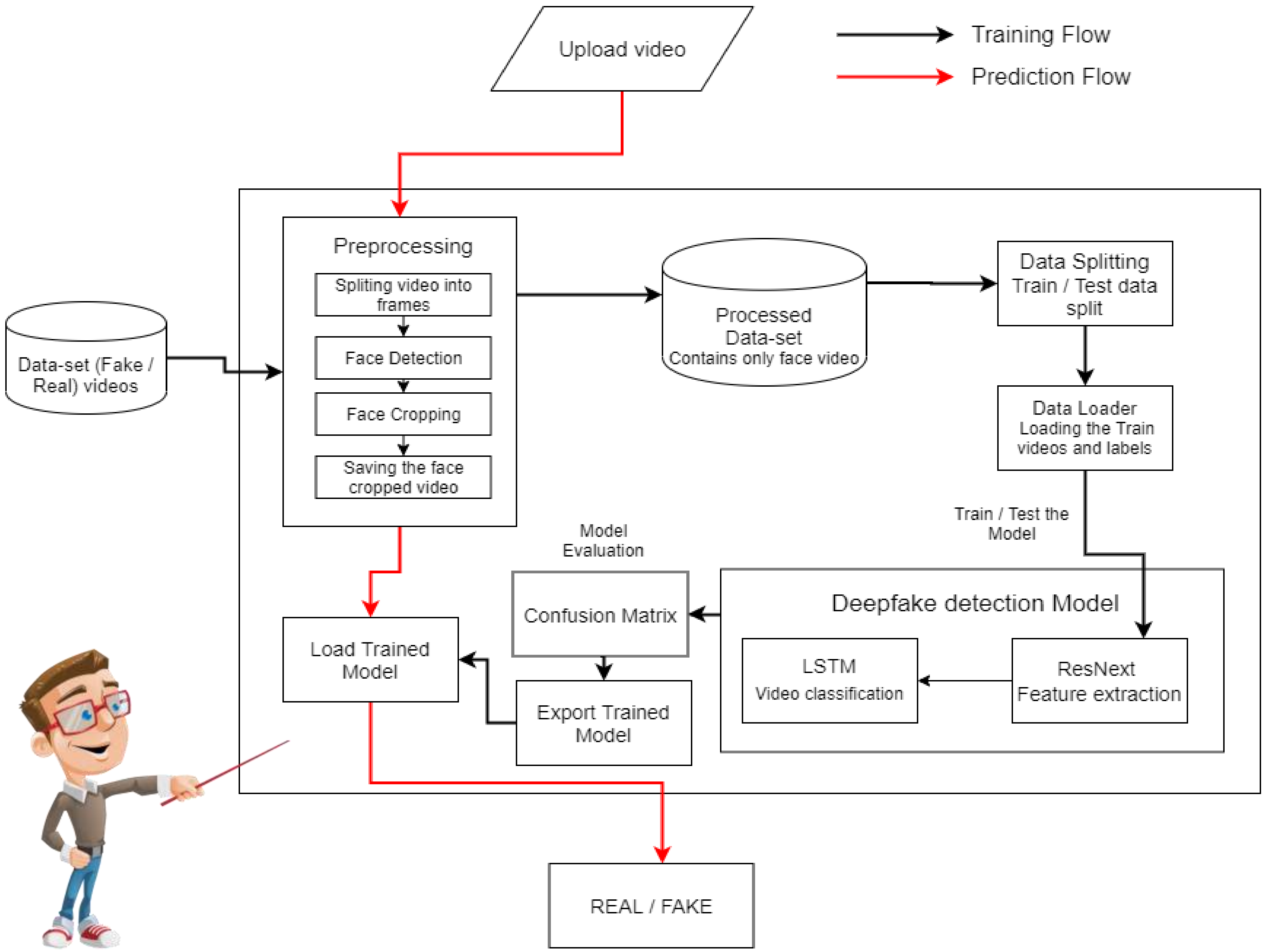
Koala's legs? = Y

Koala's body? = Y

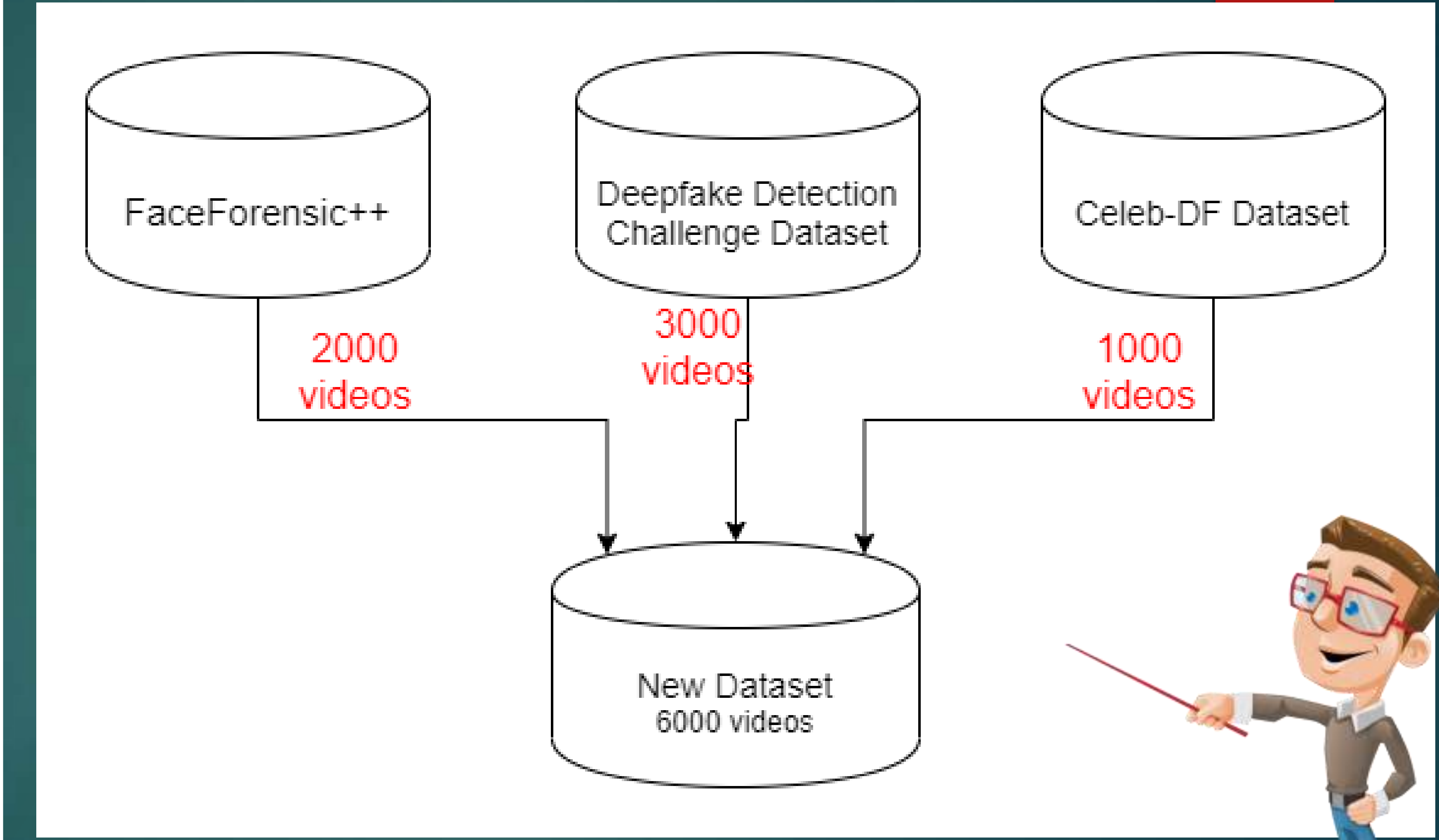
Is it Koala? = Y


eye
*
=
nose
*
=
ears
*
=
hands
*
=
legs
*
=
*
head
1
flatten
*
body
1
flatten
1
1
...
Is this Koala?

System Architecture



Data-set Exploration



Pre-processing

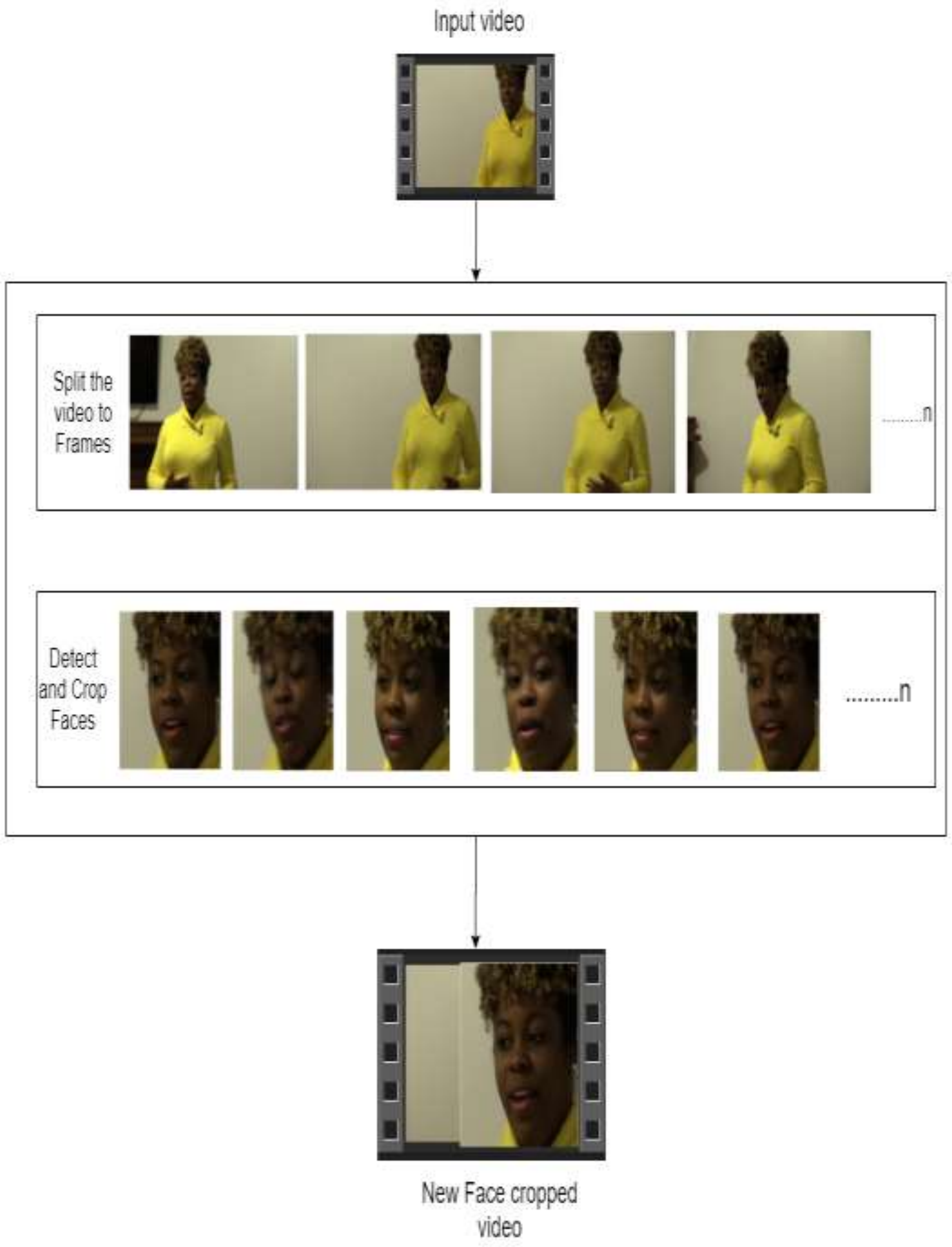
1 SPLIT VIDEO INTO FRAMES

2 FACE DETECTION

3 CROPING FACE

4 CREATING NEW FACE CROPPED VIDEO

5 SAVING THE FACE CROPPED VIDEO



❑ ResNext - CNN

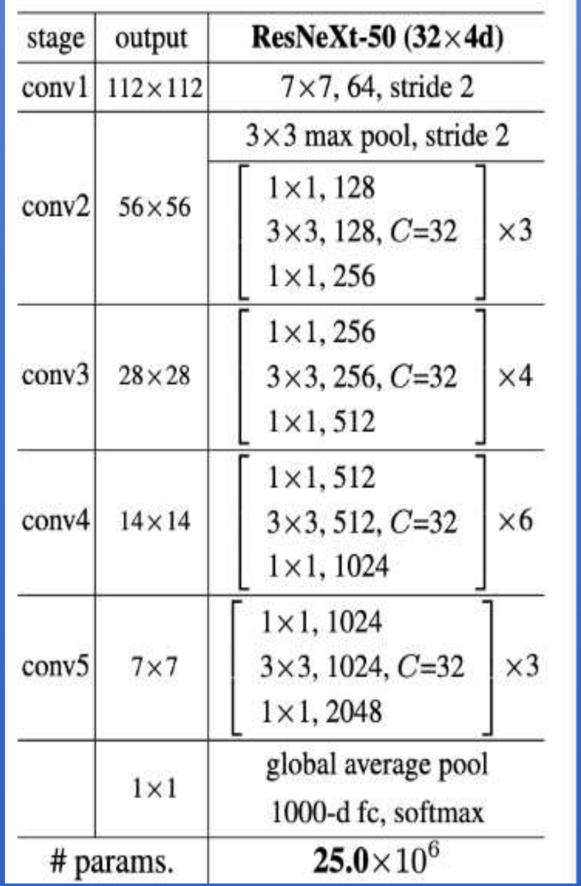
- The pre-trained model ResNext of CNN is used for feature extraction.
- ResNext is residual CNN network optimized for high performance on deeper neural networks.
- For the experimental purpose we have used resnext50_32x4d model. We have used a ResNext of 50 layers and 32 x 4 dimensions.
- Following, 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

- LSTM stands for long short-term memory, and it is a type of recurrent neural network (RNN) architecture. LSTM networks are specifically designed to model and process sequential data, such as time series, text, or speech.
- LSTMs can be used to process sequential feature vectors or sequences of feature vectors. By feeding the feature vectors into the LSTM layer, the model can learn the patterns and dependencies present in the sequence of features.

Model Architecture

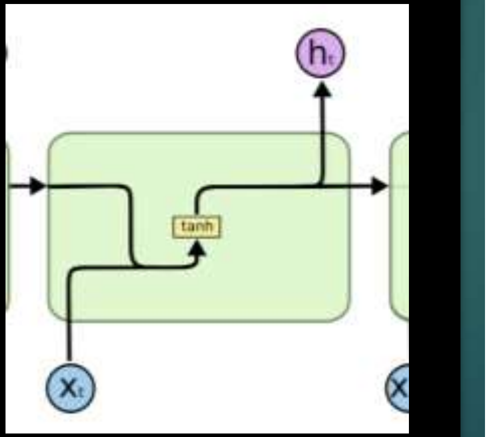
ResNext-50

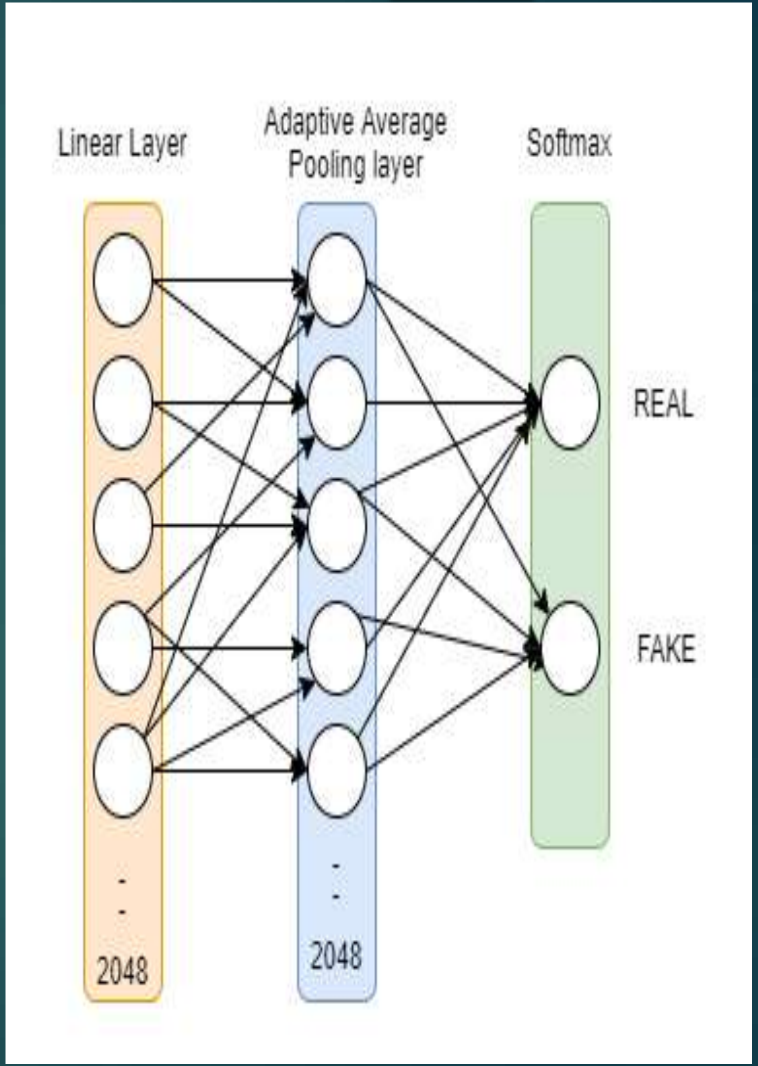


stage	output	ResNeXt-50 (32×4d)
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2 1×1, 128 3×3, 128, C=32 1×1, 256 ×3
conv3	28×28	1×1, 256 3×3, 256, C=32 1×1, 512 ×4
conv4	14×14	1×1, 512 3×3, 512, C=32 1×1, 1024 ×6
conv5	7×7	1×1, 1024 3×3, 1024, C=32 1×1, 2048 ×3
	1×1	global average pool 1000-d fc, softmax
# params.		**25.0×10^6**

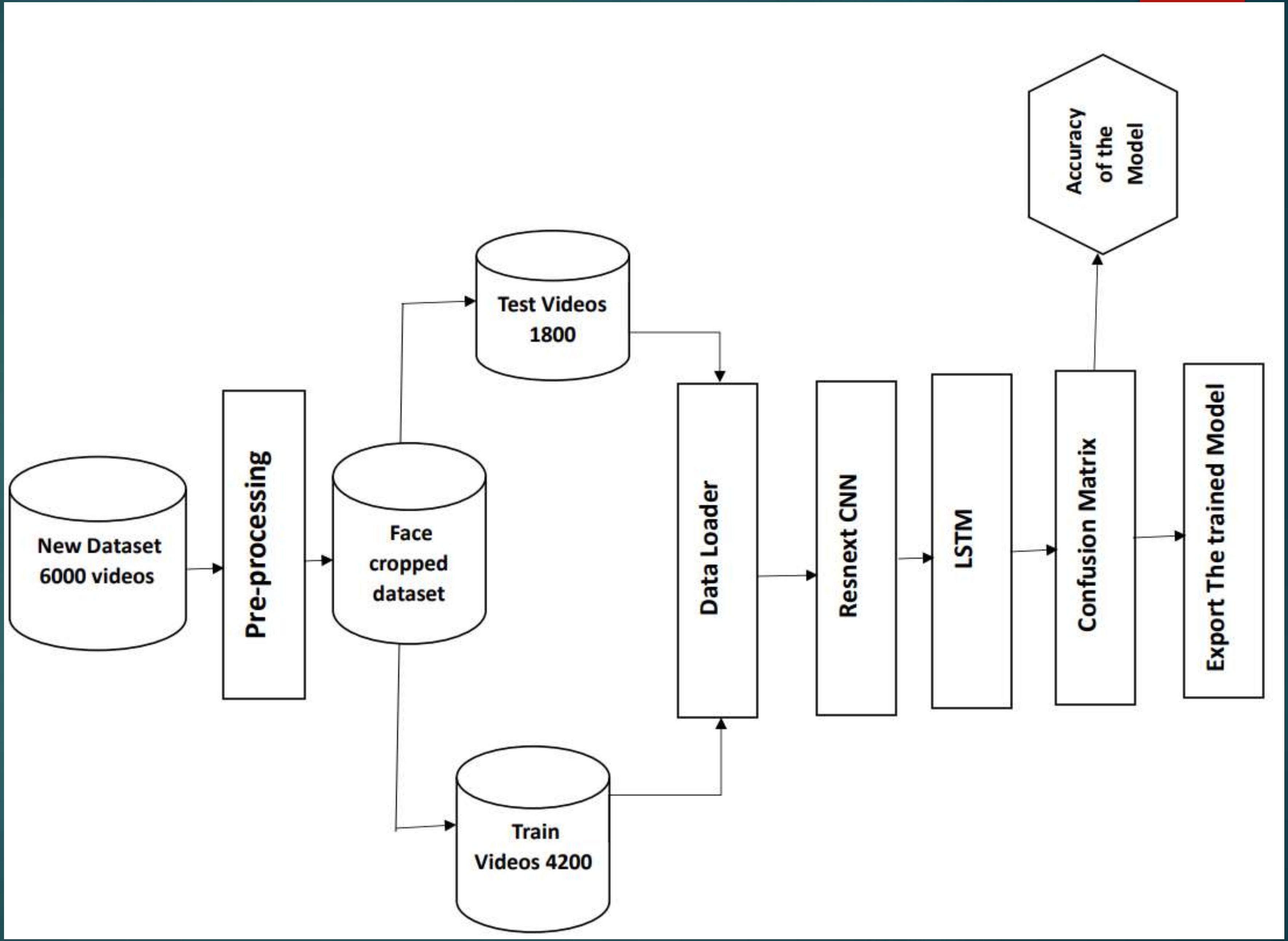
Sequential Layer

1 LSTM layer with 2048 shape input vector and 2048 latent features along with 0.4 chance of dropout and ReLU Activation function





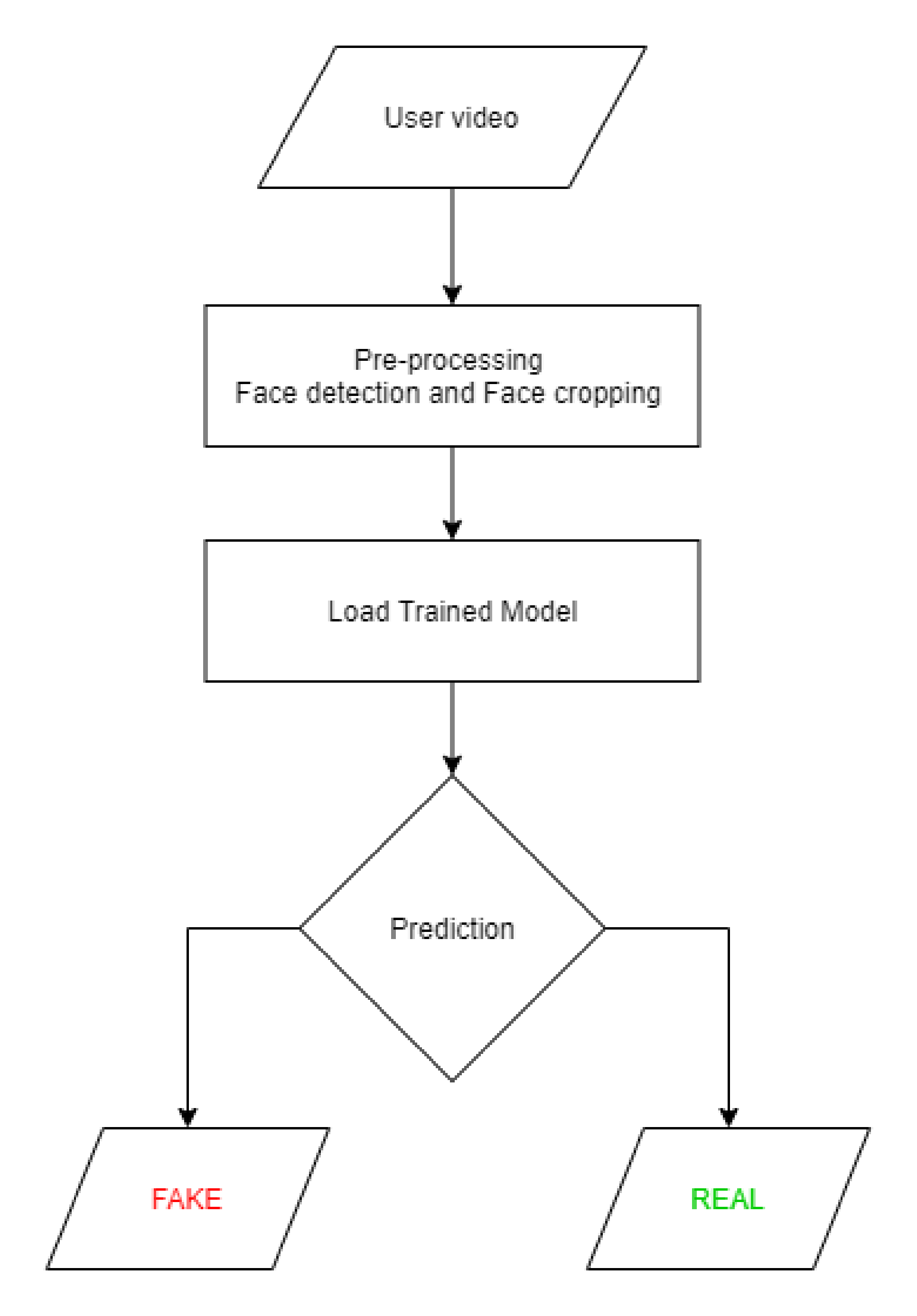
Training Workflow



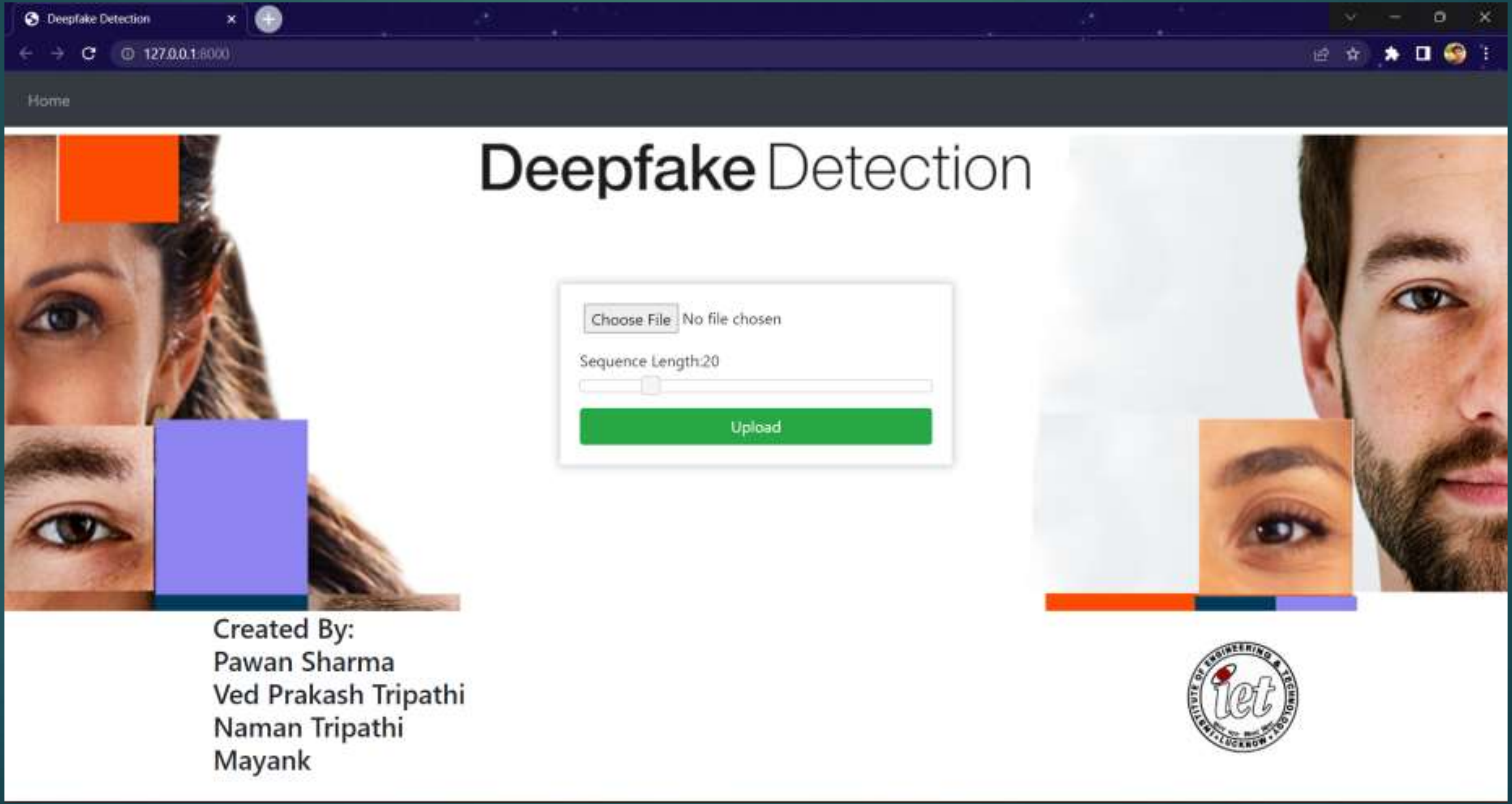
❑ CONFUSION MATRIX

- Confusion matrix is used to evaluate our model and calculate the accuracy.
- A confusion matrix is a table that is often used to evaluate the performance of a classification model. It provides a summary of the predicted and actual class labels of a dataset, allowing for the calculation of various metrics to assess the model's accuracy, precision, recall, and other performance measures.

Prediction Workflow



❑ RESULT AND CONFIDENCE OF PREDICTION



Deepfake Detection

Choose File No file chosen

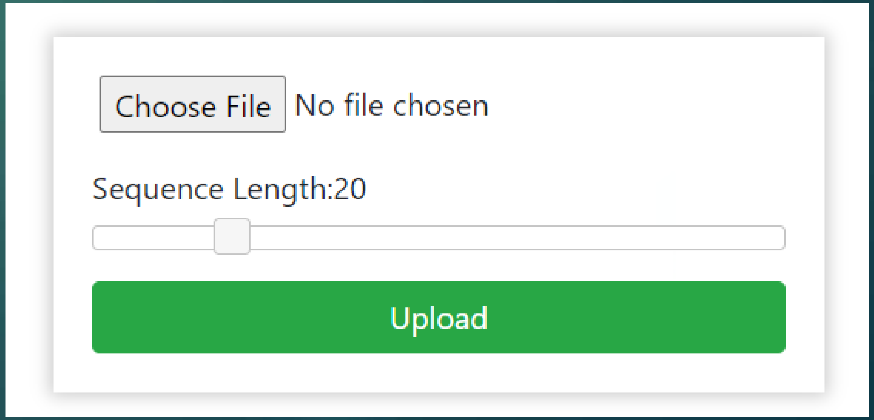
Sequence Length:20

Upload

Created By:
Pawan Sharma
Ved Prakash Tripathi
Naman Tripathi
Mayank

What is Sequence Length?

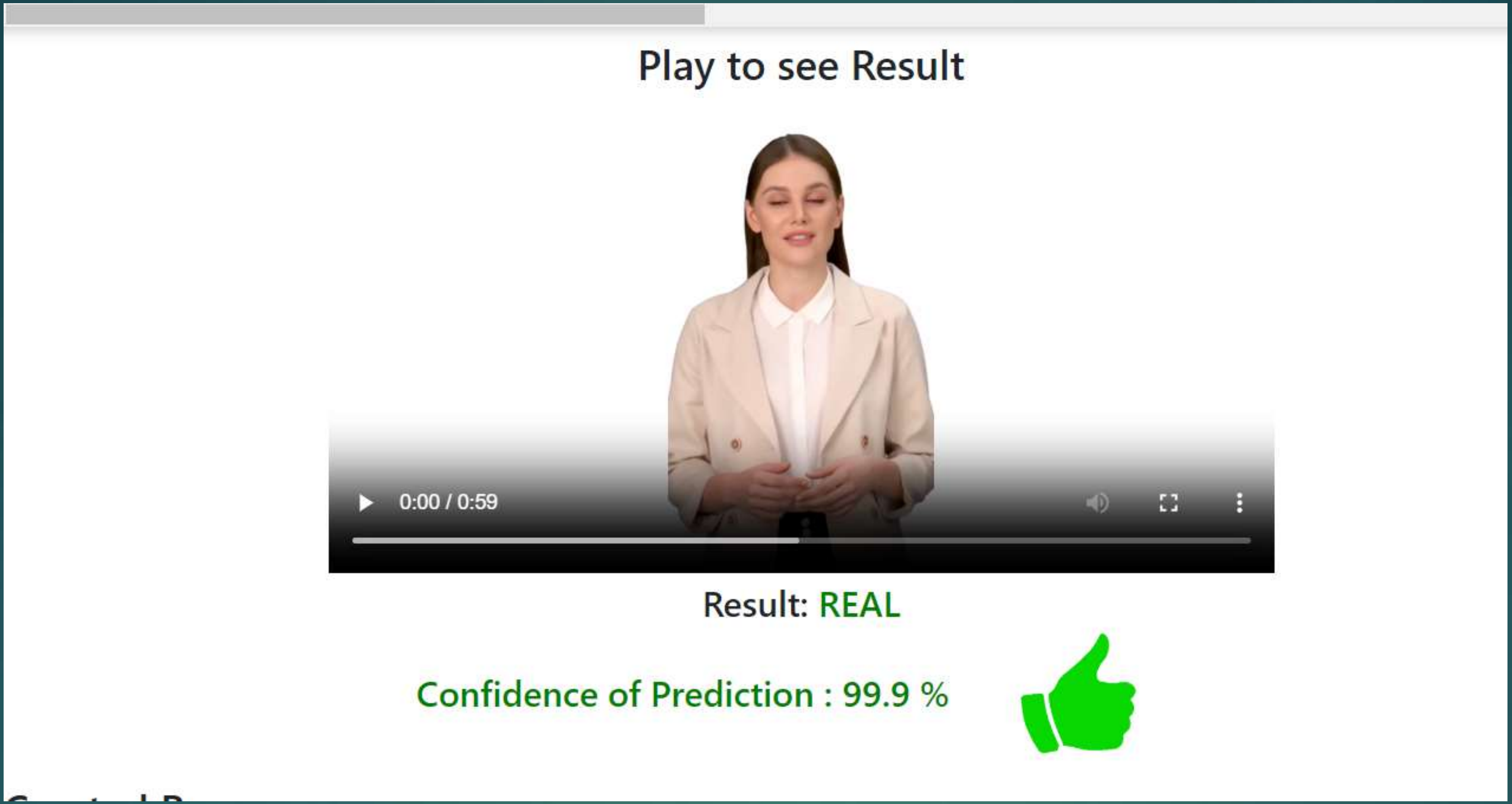
- The Videos are splitted in frames at 30 frames per second and due to this we have a large number of frames which id very difficult to process.
- Sequence Length is used to set the number of frames we are going to process.
- If it is set to 20 then , it will process only first 20 frames but if it is set to 100 then it will process the first 100 frames.



Sequence Length:20

- ❑ Here is the Result and Confidence of Prediction of an AI generated Video at different Sequence Lengths

Click to watch this video



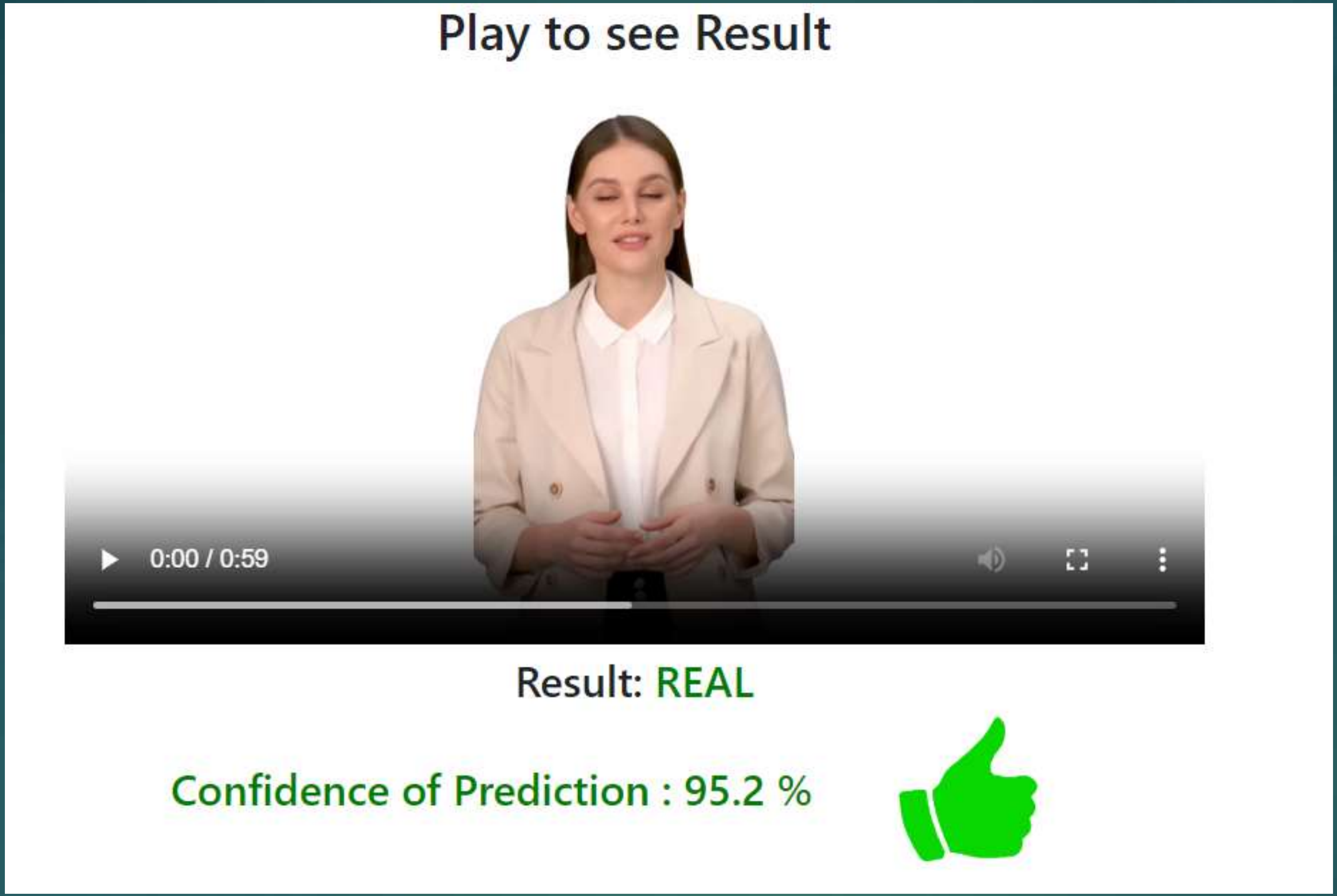
Play to see Result

Result: REAL

Confidence of Prediction : 99.9 %

Sequence length = 20

Processing first 20 frames



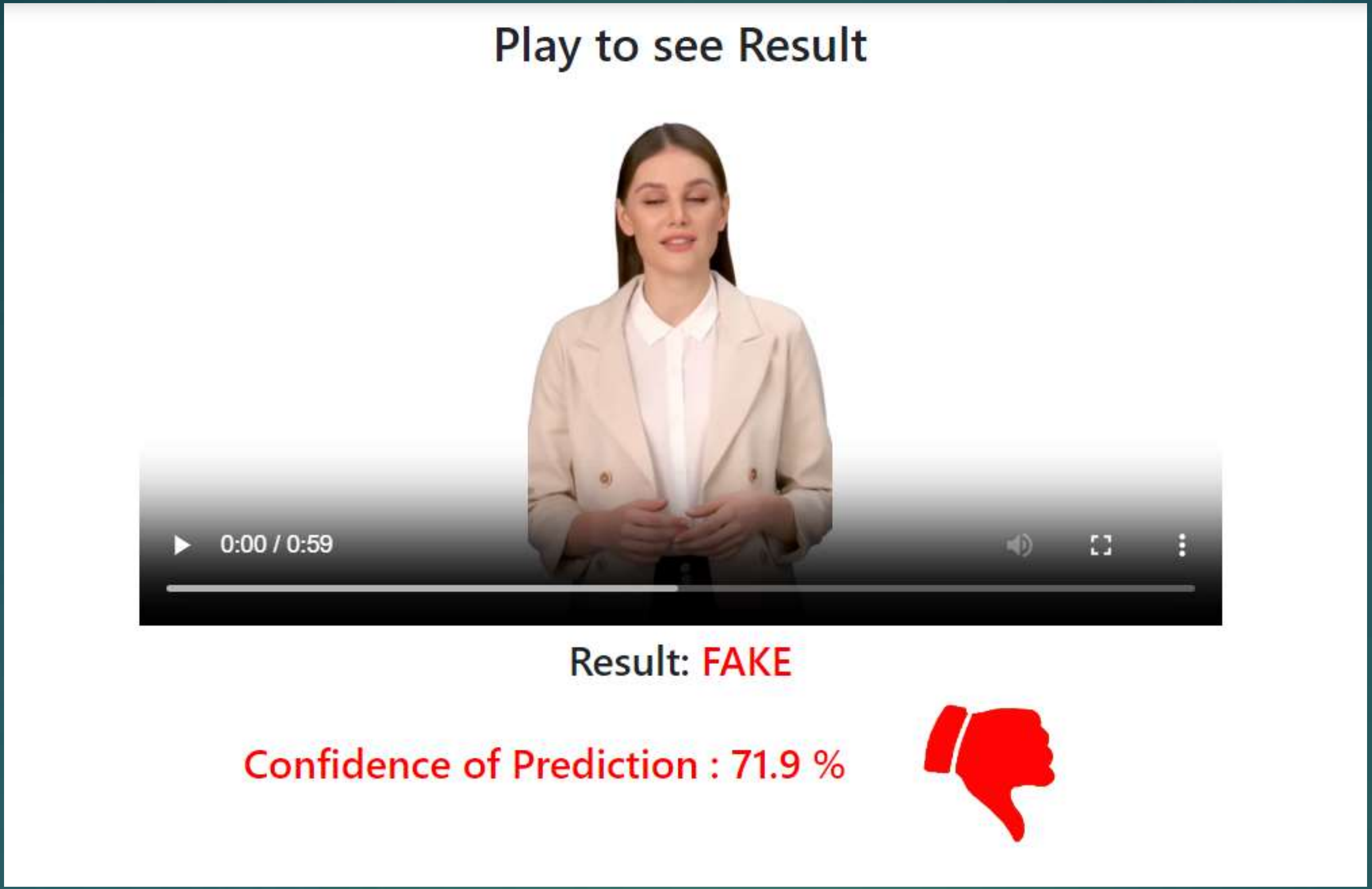
Play to see Result

Result: REAL

Confidence of Prediction : 95.2 %

<u>Sequence Length = 60</u>

Processing first 60 frames



Play to see Result

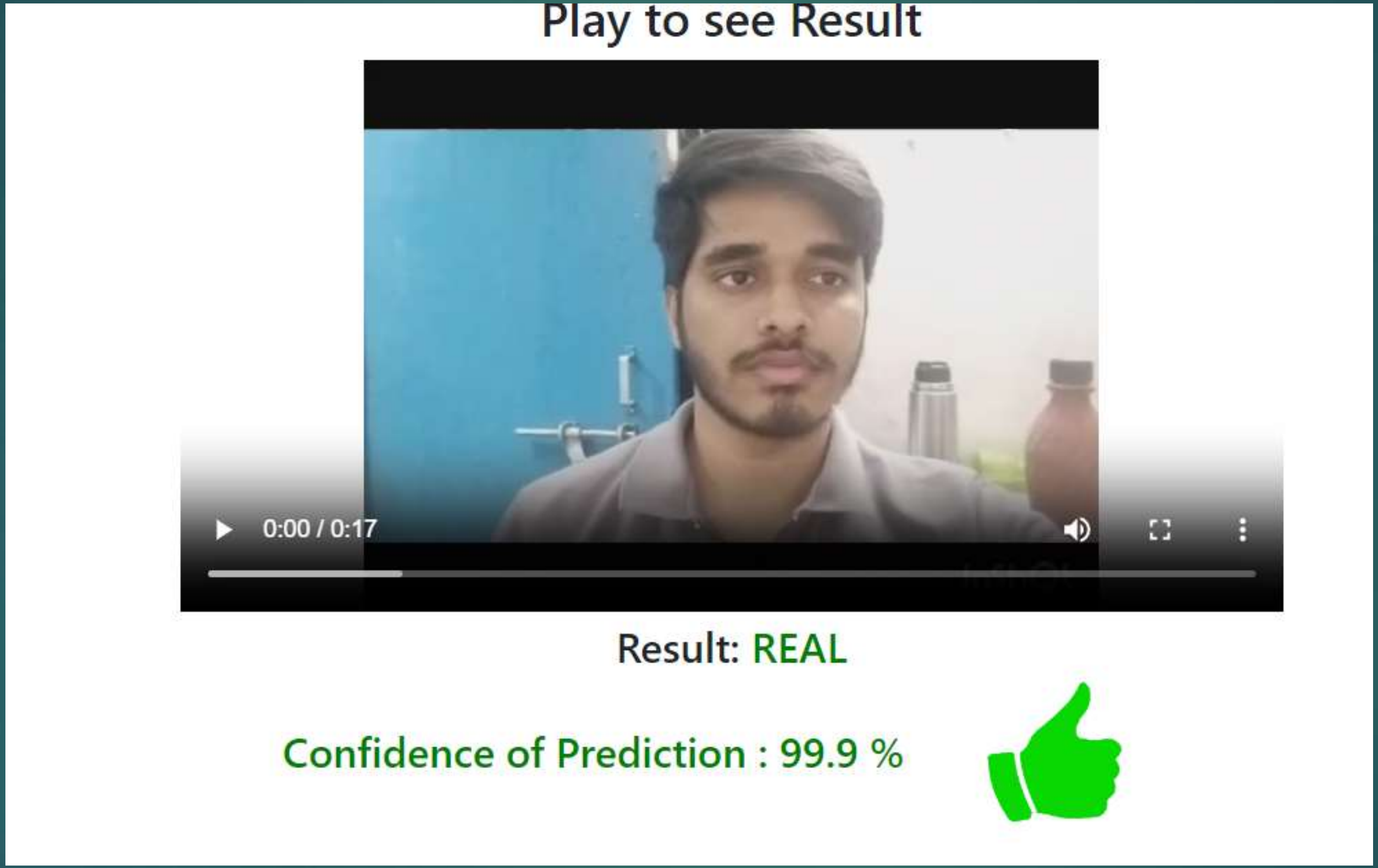
Result: FAKE

Confidence of Prediction : 71.9 %

Sequence Length = 100

Processing first 100 frames

❑ Result for a Real Time Video

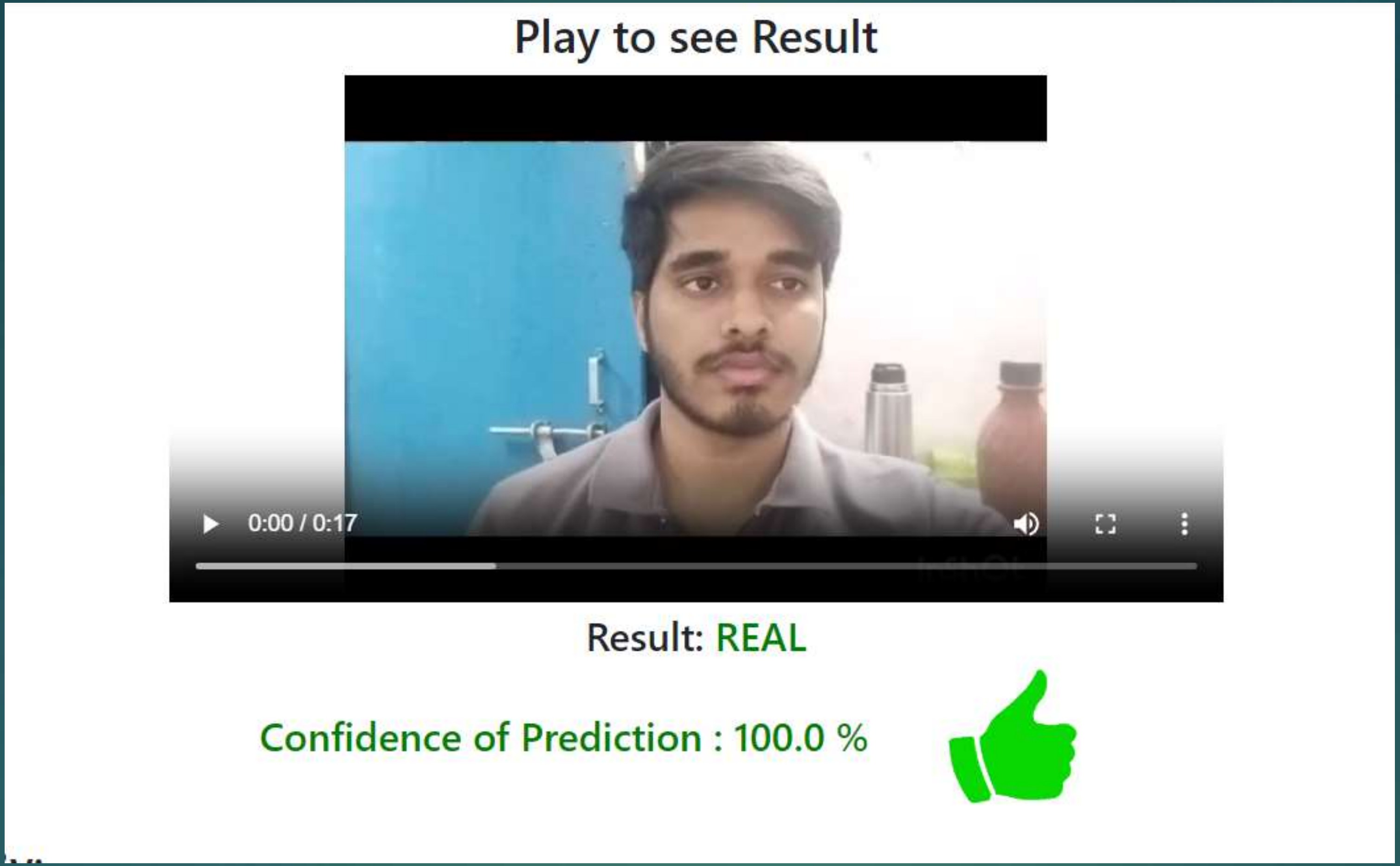


Play to see Result

Result: REAL

Confidence of Prediction : 99.9 %

<u>At Sequence Length = 20, or 40</u>



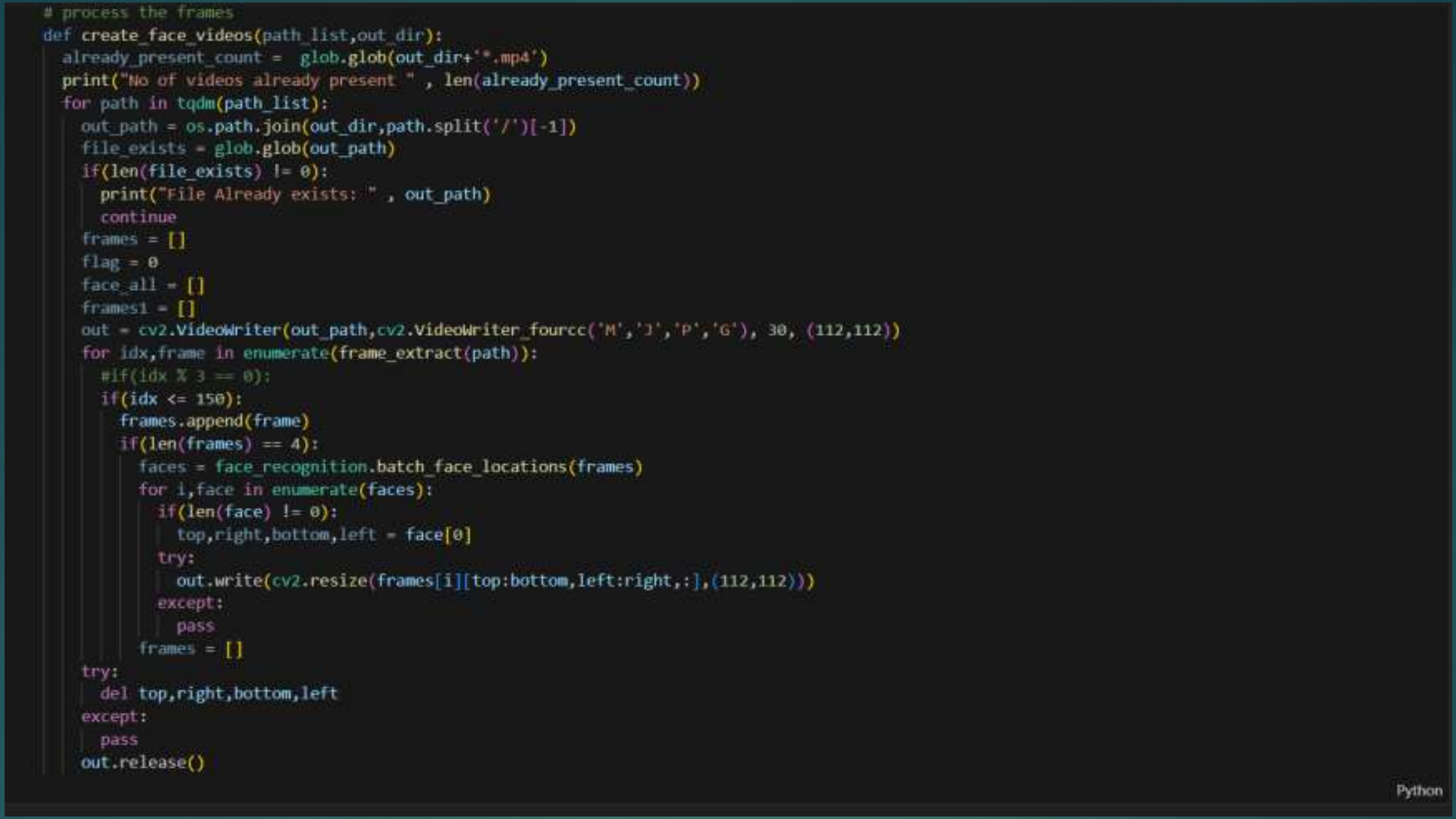
Play to see Result

Result: REAL

Confidence of Prediction : 100.0 %

<u>At Sequence Length = 60 or 80 or 100</u>

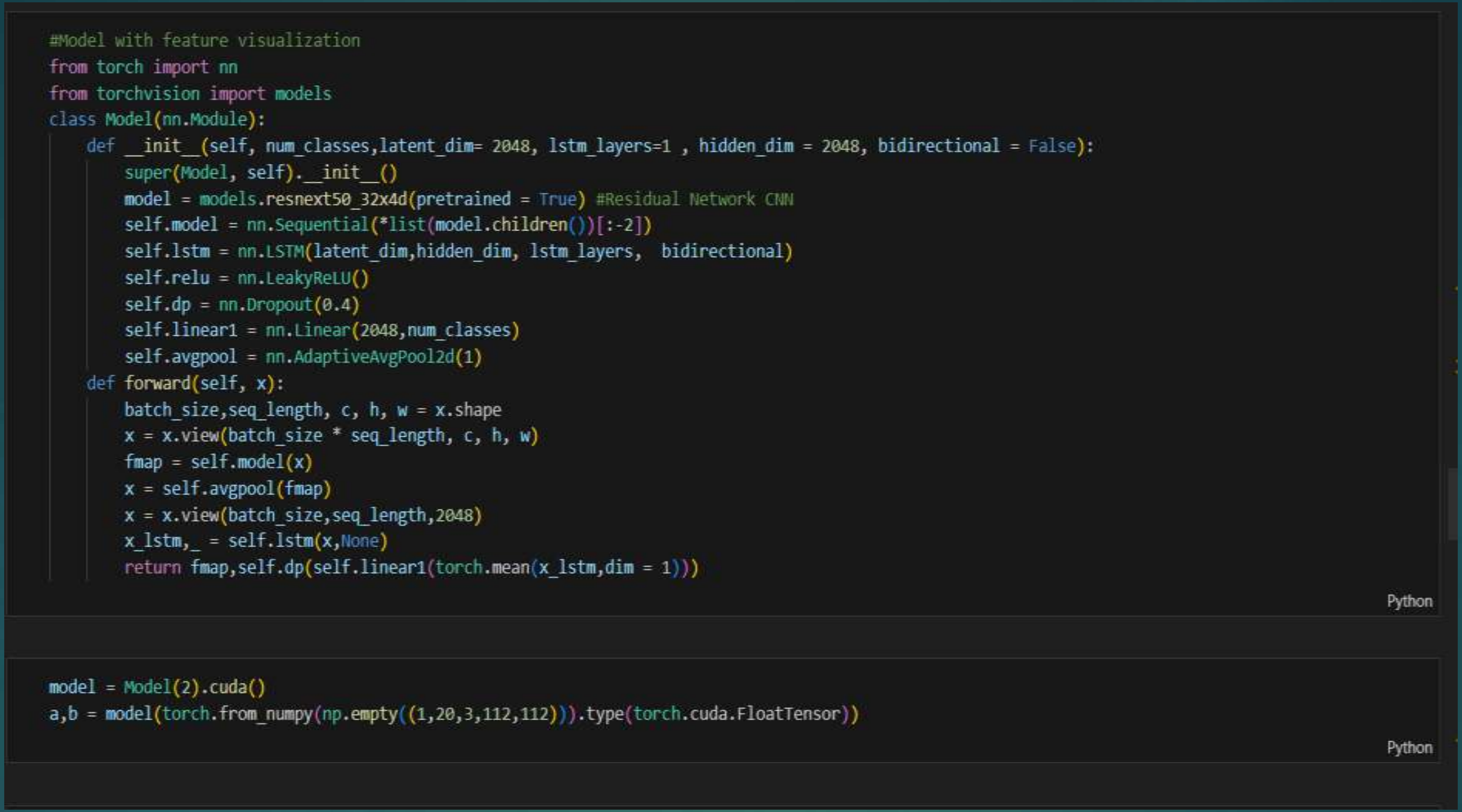
❑ Code for Pre- Processing



```python
# process the frames
def create_face_videos(path_list,out_dir):
  already_present_count =  glob.glob(out_dir+'*.mp4')
  print("No of videos already present " , len(already_present_count))
  for path in tqdm(path_list):
    out_path = os.path.join(out_dir,path.split('/')[-1])
    file_exists = glob.glob(out_path)
    if(len(file_exists) != 0):
      print("File Already exists: " , out_path)
      continue
    frames = []
    flag = 0
    face_all = []
    frames1 = []
    out = cv2.VideoWriter(out_path,cv2.VideoWriter_fourcc('M','J','P','G'), 30, (112,112))
    for idx,frame in enumerate(frame_extract(path)):
      #if(idx % 3 == 0):
      if(idx <= 150):
        frames.append(frame)
        if(len(frames) == 4):
          faces = face_recognition.batch_face_locations(frames)
          for i,face in enumerate(faces):
            if(len(face) != 0):
              top,right,bottom,left = face[0]
            try:
              out.write(cv2.resize(frames[i][top:bottom,left:right,:],(112,112)))
            except:
              pass
          frames = []
    try:
      del top,right,bottom,left
    except:
      pass
    out.release()
```

Click here to see Full Code

❑Code for Model Creation

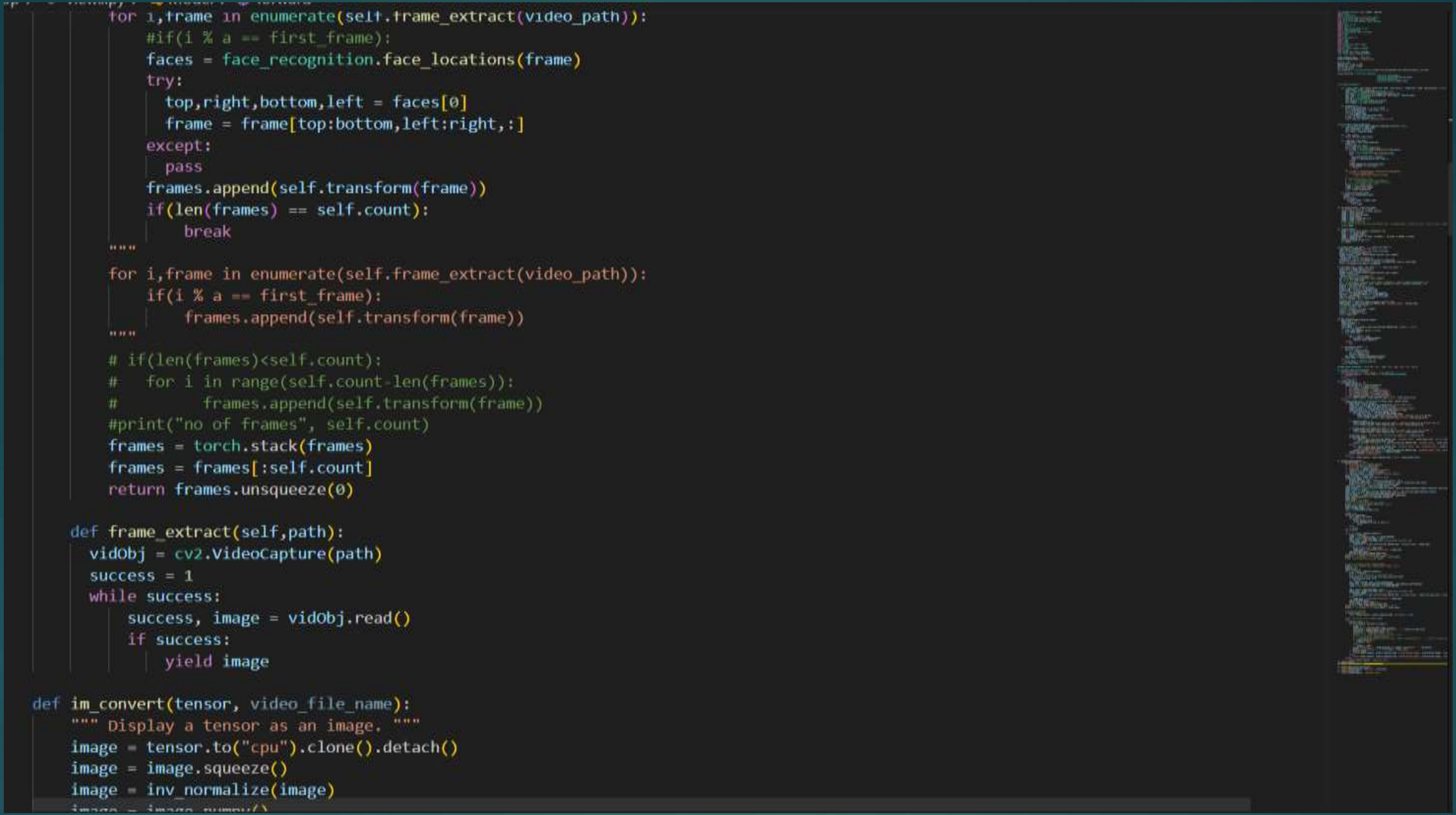


```python
#Model with feature visualization
from torch import nn
from torchvision import models
class Model(nn.Module):
    def __init__(self, num_classes,latent_dim= 2048, lstm_layers=1 , hidden_dim = 2048, bidirectional = False):
        super(Model, self).__init__()
        model = models.resnext50_32x4d(pretrained = True) #Residual Network CNN
        self.model = nn.Sequential(*list(model.children())[:-2])
        self.lstm = nn.LSTM(latent_dim,hidden_dim, lstm_layers,  bidirectional)
        self.relu = nn.LeakyReLU()
        self.dp = nn.Dropout(0.4)
        self.linear1 = nn.Linear(2048,num_classes)
        self.avgpool = nn.AdaptiveAvgPool2d(1)
    def forward(self, x):
        batch_size,seq_length, c, h, w = x.shape
        x = x.view(batch_size * seq_length, c, h, w)
        fmap = self.model(x)
        x = self.avgpool(fmap)
        x = x.view(batch_size,seq_length,2048)
        x_lstm,_ = self.lstm(x,None)
        return fmap,self.dp(self.linear1(torch.mean(x_lstm,dim = 1)))
```

```python
model = Model(2).cuda()
a,b = model(torch.from_numpy(np.empty((1,20,3,112,112))).type(torch.cuda.FloatTensor))
```

Click here to see Full Code

❑ Code for Prediction using Trained Model



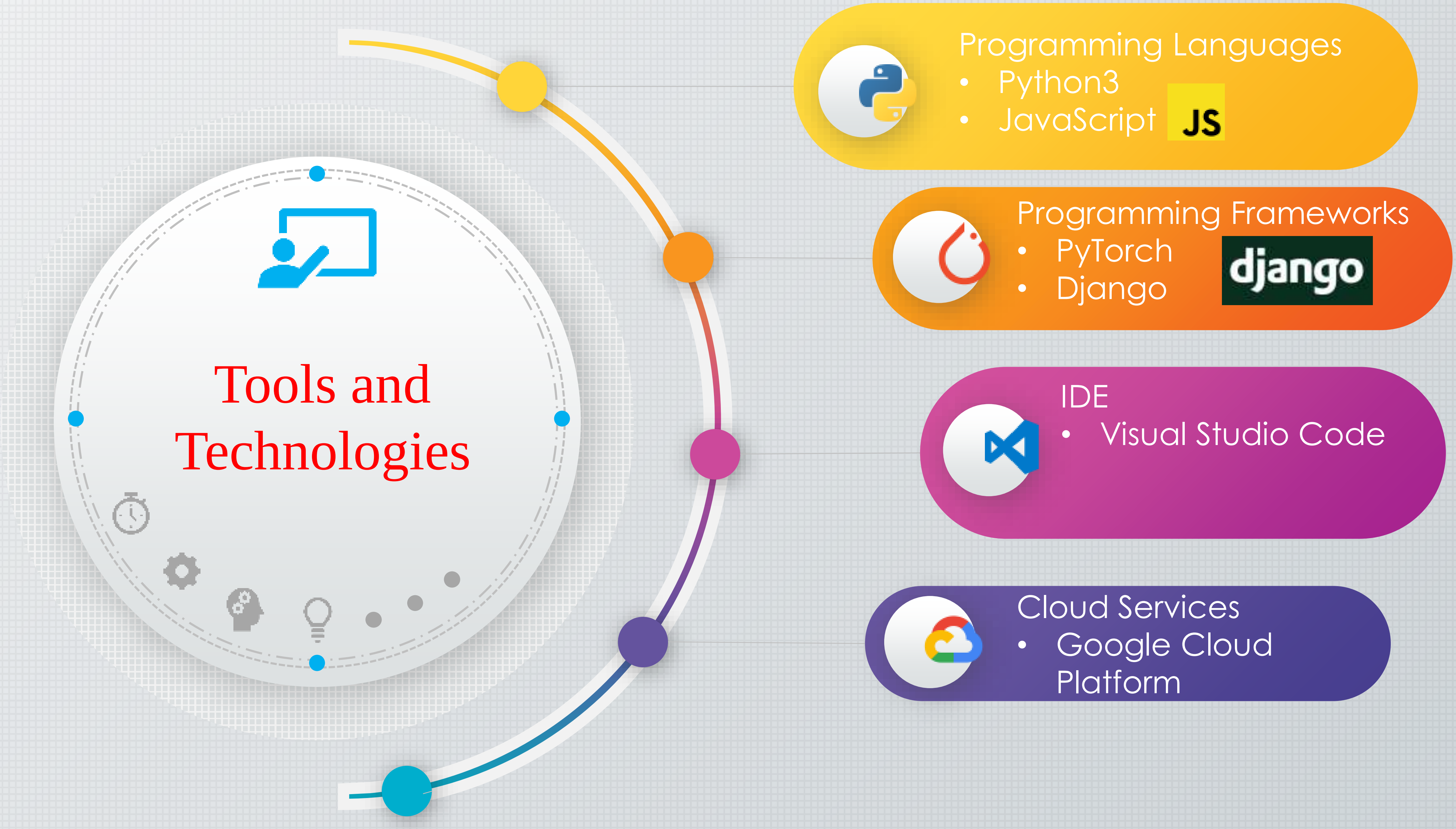
```
        for i,frame in enumerate(self.frame_extract(video_path)):
            #if(i % a == first_frame):
            faces = face_recognition.face_locations(frame)
            try:
              top,right,bottom,left = faces[0]
              frame = frame[top:bottom,left:right,:]
            except:
              pass
            frames.append(self.transform(frame))
            if(len(frames) == self.count):
                break
        """
        for i,frame in enumerate(self.frame_extract(video_path)):
            if(i % a == first_frame):
                frames.append(self.transform(frame))
        """
        # if(len(frames)<self.count):
        #   for i in range(self.count-len(frames)):
        #         frames.append(self.transform(frame))
        #print("no of frames", self.count)
        frames = torch.stack(frames)
        frames = frames[:self.count]
        return frames.unsqueeze(0)

    def frame_extract(self,path):
      vidObj = cv2.VideoCapture(path)
      success = 1
      while success:
          success, image = vidObj.read()
          if success:
              yield image

def im_convert(tensor, video_file_name):
    """ Display a tensor as an image. """
    image = tensor.to("cpu").clone().detach()
    image = image.squeeze()
    image = inv_normalize(image)
    image = image.numpy()
```

Click here to see Full Code

Tools and Technologies



Programming Languages

- Python3
- JavaScript

Programming Frameworks

- PyTorch
- Django

IDE

- Visual Studio Code

Cloud Services

- Google Cloud Platform

❑ References

- Deepfake detection challenge dataset : https://www.kaggle.com/c/deepfake-detection-challenge/data
- Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies,Matthias Nießner, “FaceForensics++: Learning to Detect Manipulated Facial Images” in arXiv:1901.08971.
- Yuezun Li , Xin Yang , Pu Sun , Honggang Qi and Siwei Lyu “Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics” in arXiv:1909.12962
- https://www.geeksforgeeks.org/software-engineering-cocomo-model/
- 10 deepfake examples that terrified and amused the internet : https://www.creativebloq.com/features/deepfake-examples
- Deepfakes, Revenge Porn, And The Impact On Women : https://www.forbes.com/sites/chenxiwang/2019/11/01/deepfakes-revenge-porn-and-the-impact-on-women/

Click here

Thank
you!!