5/3/2021

Arjun Janamatti

ScalingWeb

AI Work Using Python

By Arjun Janamatti & Kushal Mandala

Table of Contents

[Script to identify unsafe frame in videos, unsafe speech in video 2](#_Toc70978436)

[Run the Script 2](#_Toc70978437)

[Methodology 2](#_Toc70978438)

[Transcript result 2](#_Toc70978439)

[Video classification result 3](#_Toc70978440)

[Evidence of unsafe frames 5](#_Toc70978441)

# Script to identify unsafe frame in videos, unsafe speech in video

## Run the Script

Here we can use PostMan to use API, since this is currently in local system, I am using Postman to upload the file, here the only thing to be noted is the url link is [http://127.0.0.1:5000:/video/upload] and in the postman upload file option upload the file, this will give the results of speech, video and evidence of unsafe speech and unsafe images in base64 format.

## Methodology

Below are the key packages/ models used for the prediction algorithm.

* Deep speech model = “deepspeech-0.9.3-models.pbmm”, this is the main deep speech model used for converting the speech to text
* Deep speech scorer = “deepspeech-0.9.3-models.scorer” here the scorer improves the probability of decoding of speech into text
* Image\_classification\_model = “nfsw.299x299.h5”, this is the deep learning model which is generated after training the algorithm with many images of which are safe and unsafe.

Flask RestAPI is created for the script, where one needs to upload a video, will give the following results:

* “Transcript result”
* “Video classification result”
* “Evidence of unsafe frames”

Lets discuss in detail for each of the above results on how it is obtained and type of outputs expected:

### Transcript result

The script for getting the transcript results has 3 methods namely VideoToText, ProfaneWordList, TextResult

* VideoToText: Initially in the script we will initialize the video file and name of the video file, followed by the VideoToText method. Here using ffmpeg, the video is convert to audio with a bit rate as per the requirement of Deepspeech. Followed by using the deepspeech model and scorer to convert the audio to text, and this text is the output for this method. Also each time this method is called a temporary audio file is created and deleted at the end of this method.
* ProfaneWordList: Initially after searching in the internet for the profane words list, this list was later verified and edited, this list is sent as output for the method.
* TextResult: This is the main part of the script, here initially the VideoToText is called, followed by decoding to “utf-8”, and then it records and counts the number of matching words in both the profane words and text form the video. This will give the number of profane words and profane words list in the output.



### Video classification result

There are two methods involved to get the result of unsafe content in the videos namely MakeImageDirectory and VideoClassifyResult.

* MakeImageDirectory: Here each video is devided into frames, followed by the video directory creation and images are saved for each second of the video.
* VideoClassifyResult: Here initially the MakeImageDirectory is called and with this we have images from each second of the video, next will label each image as safe or unsafe, if the image is unsafe then that image is converted to base64 and saved in a list and number of unsafe images are counted for a video. Finally if the number of unsafe images is more than 50% of total number of images of video then it is termed as unsafe and below message will be delivered.
* (f'{self.video\_file} is categorized as: "UNSAFE VIDEO", since percentage of unsafe images: {percent\_unsafe}%')
  + “Video is categorized as “UNSAFE VIDEO”, since percentage of unsafe images : percent\_unsafe”, here percent\_unsafe is percentage value of unsafe images in entire images folder.

If the number of unsafe images is greater than 30 but less than 50%, then the following message will be delivered.

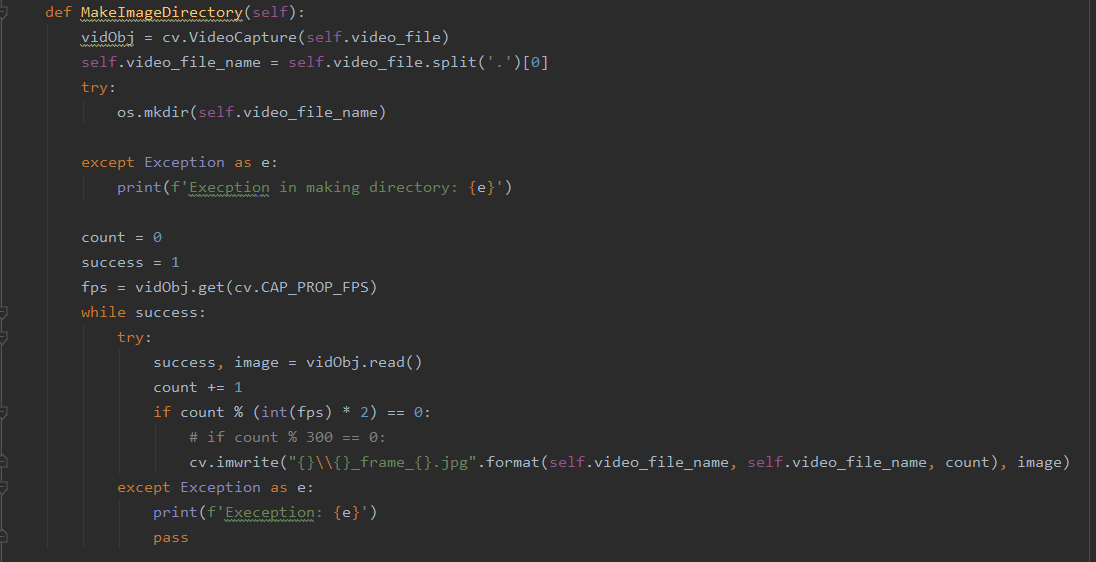
* + “Video is categorized as “ADMIN HAS TO VERIFY”, since percentage of unsafe images : percent\_unsafe”, here percent\_unsafe is percentage value of unsafe images in entire images folder.

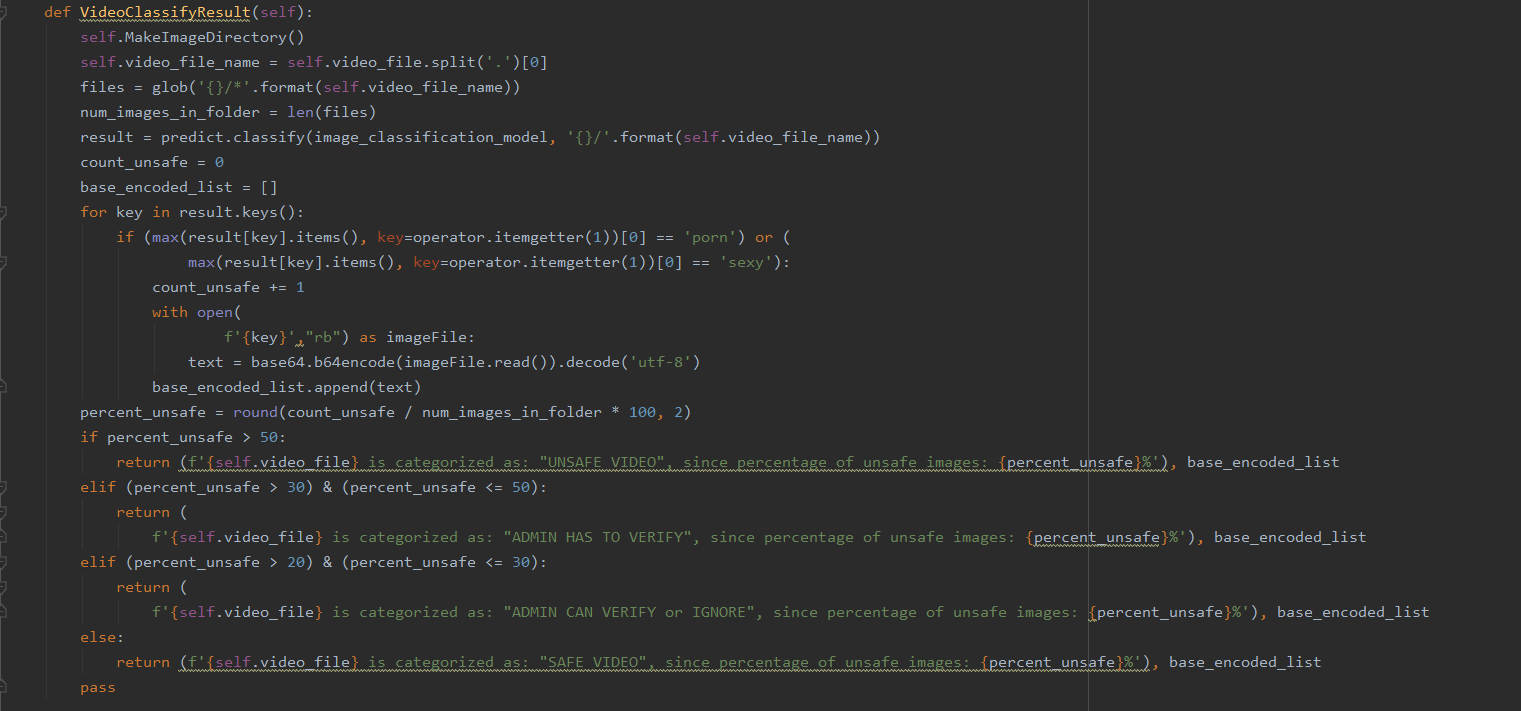
If the number of unsafe images is greater than 20 but less than 30%, then the following message will be delivered.

* + “Video is categorized as “ADMIN CAN VERIFY OR IGNORE”, since percentage of unsafe images : percent\_unsafe”, here percent\_unsafe is percentage value of unsafe images in entire images folder.

If the number of unsafe images is less than 20%, then the following message will be delivered.

* + “Video is categorized as “SAFE VIDEO”, since percentage of unsafe images : percent\_unsafe”, here percent\_unsafe is percentage value of unsafe images in entire images folder.





### Evidence of unsafe frames

This part of the result uses the MakeImageDirectory and VideoClassifyResult, here the list of unsafe images are given as output in the form of base64.