5/3/2021

Arjun Janamatti

ScalingWeb

AI Work Using Python

By Arjun Janamatti & Kushal Mandala

Table of Contents

[1.0 Script to identify unsafe frame in videos, unsafe speech in video 2](#_Toc71040688)

[1.1 Run the Script 2](#_Toc71040689)

[1.2 Methodology 2](#_Toc71040690)

[1.2.1 Transcript result 2](#_Toc71040691)

[1.2.2 Video classification result 3](#_Toc71040692)

[1.2.3 Evidence of unsafe frames 5](#_Toc71040693)

[2.0 Trending/Popular/Near-Location 6](#_Toc71040694)

[2.1 Run the script 6](#_Toc71040695)

[2.2 Methodology 6](#_Toc71040696)

[2.2.1 Pre-processing steps 6](#_Toc71040697)

[2.2.2 Top Trending Results 7](#_Toc71040698)

[2.2.3 Top Popular Results 7](#_Toc71040699)

[2.2.4 Near Distance 7](#_Toc71040700)

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

## 1.1 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.

## 1.2 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:

### 1.2.1 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.



### 1.2.2 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.

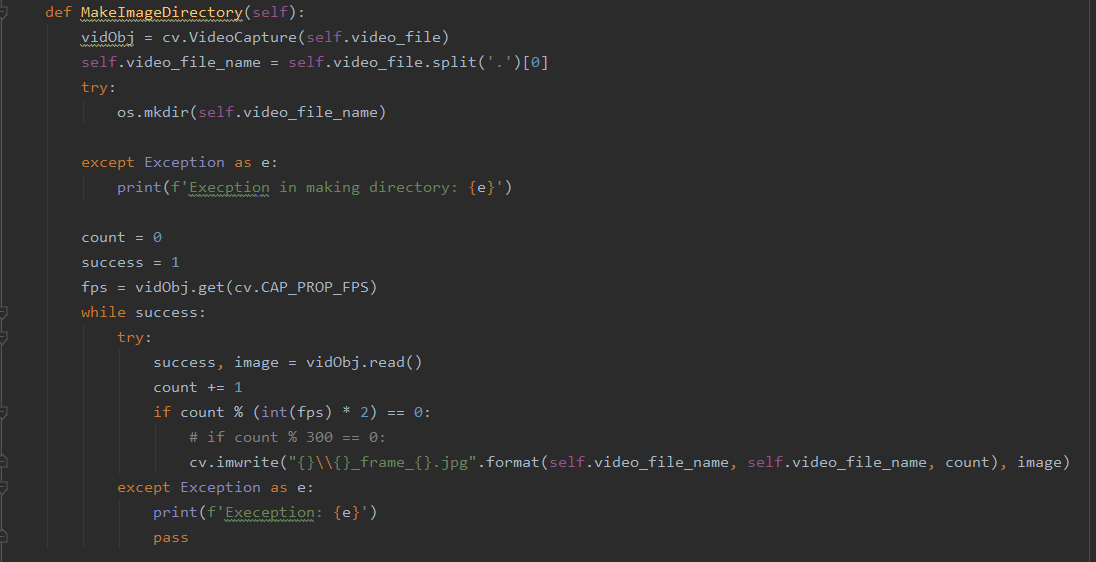
* + “Video is categorized as “ADMIN HAS TO VERIFY”, since percentage of unsafe images : percent\_unsafe”, here percent\_unsafe is percentage value of unsafe images in entire images folder.

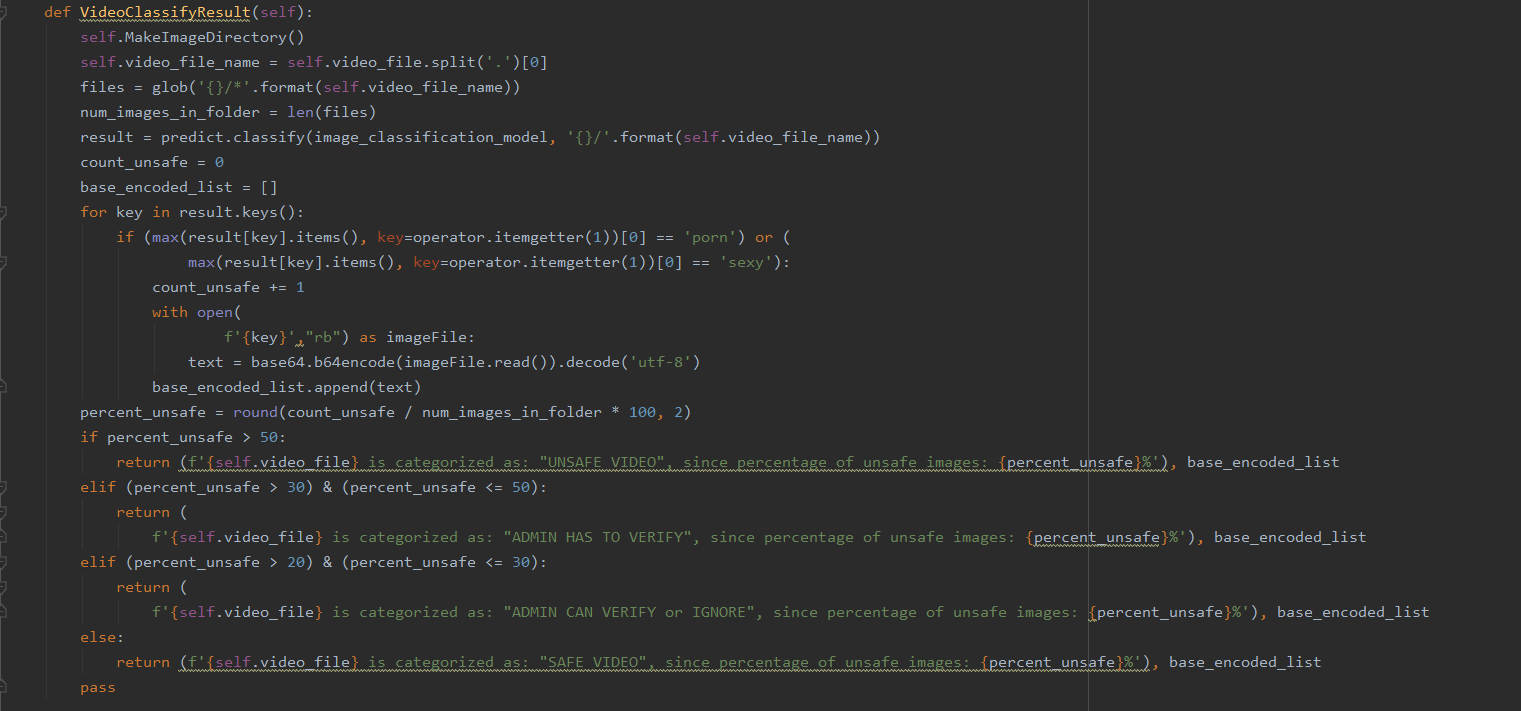
If the number of unsafe images is greater than 20 but less than 30%, then the following message will be delivered.

* + “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.





### 1.2.3 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.

# 2.0 Trending/Popular/Near-Location

## 2.1 Run the script

In the below link, one has to change the yellow highlight to trending, if trending is required, and for reviews, it has to be changed to reviews.

* <https://py1.realreviews.org/popular-user?categoryid=&userid=5fdf6bbcfe08e8c0191a7805>

For near location, one has to change the longitude and latitude

* <https://py1.realreviews.org/near-location?longlat=-90.1985,38.6364&userid=5fdf6bbcfe08e8c0191a7805>

In both the above links, the userid parameter will check if this user has been blocked by some other user of if this user has blocked some other users and filter the results accordingly.

## 2.2 Methodology

Flask RestAPI is created for the script, where one needs to mention the category\_id which is optional and user\_id for blockusers, will give the following results:

* “trending\_review”
* “trending-user”
* “popular-review”
* “popular-review”
* “near-location”
* “top-products”
* “top-services”

In the above list all the links will work except for top-products, and top-services, since there is no data in the backend for these, however once we have more data and old data without “reviews” keys is removed, will uncomment the section, so that this part of the script can also be consumed in the API.

### 2.2.1 Pre-processing steps

There are four main steps involved in pre-processing namely

* GetBlockUsersData
* GetTableDictionary
* MergeDataframe

#### 2.2.1.1 GetBlockUsersData

Here each entry to blockusers table is stored in the form of a list, where each list will have two user\_id, since requirement is to block “A” and “B” both, have used this method.

#### 2.2.1.2 GetTableDictionary

In this method will push the reviews table and likes table into a dictionary, which can be used in the future to dataframe. Currently am only using likes, once views and comments are used, I will use all those tables.

#### 2.2.1.3 MergeDataframe

Here will merge both the reviews and likes based on the review\_id common column and get the latitude, longitude separately, get only the dates from the updated dates, also here only will filter the data based on the “blockusersdata”

### 2.2.2 Top Trending Results

In the top trending results, if user or review is selected it will initially scan for past 7 days data and give the results, however if there are less than 10 results in past 7 days, then the algorithm will look for results for last 14 days, till the last day in the data.

### 2.2.3 Top Popular Results

In the top popular results, if user or review is selected it will initially scan for past 30 days data and give the results, however if there are less than 10 results in past 30 days, then the algorithm will look for results for last 60 days, till the last day in the data.

### 2.2.4 Near Distance

Here the algorithm will calculate the distance from the input latitude and longitude to all the reviews in the current database, and will sort in ascending order, the top 10 nearest location reviews will be output.