NSFW VIDEO DETECTOR

Taking references from : https://www.pyimagesearch.com/2019/07/15/video-classification-with-keras-and-deep-learning/ (https://www.pyimagesearch.com/2019/07/15/video-classification-with-keras-and-deep-learning/)

Used my image classifier over a condom As Government band Condom ads on tv between 6 am to 10pm as they are indecent see here (https://www.thehindu.com/news/national/govt-bans-condom-ads-from-6-am-to-10-pm-because-they-are-indecent/article21461765.ece)

So are they really indecent? I ran my model over a condom ad and found yes they really are indecent as my classifier identify it as porn and sexy on many frames. Check the video in Readme

In [92]:
```
import cv2
from keras.models import load_model
import numpy as np
from collections import deque
import warnings
warnings.filterwarnings("ignore")
```

In [90]:
```
model = load_model("Final_weights.h5")
```

In [4]:
```
labels = {0 : "Neutral", 1 : "Porn", 2 : "Sexy"}
```

In [82]:
```
size = 128
input_vid = "2.mp4"
output_vid = "Output/1.avi"
```

In [83]:
```
# Mean Subtraction
# mean = np.array([123.68, 116.779, 103.939][::1], dtype="float32")
Q = deque(maxlen=size)
```

In [96]:
```
vs = cv2.VideoCapture(input_vid)
writer = None
(W, H) = (None, None)

# loop over frames from the video file stream
while True:
    # read the next frame from the file
    (grabbed, frame) = vs.read()

    # if the frame was not grabbed, then we have reached the end
    # of the stream
    if not grabbed:
        break

    # if the frame dimensions are empty, grab them
    if W is None or H is None:
        (H, W) = frame.shape[:2]

    output = frame.copy()
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame = frame/255.0
    frame = cv2.resize(frame, (224, 224)).astype("float32")

#     frame -= mean

    # make predictions on the frame and then update the predictions
    # queue
    preds = model.predict(np.expand_dims(frame, axis=0))[0]
    print(preds)
    Q.append(preds)

    # perform prediction averaging over the current history of
    # previous predictions

    results = np.array(Q).mean(axis=0)
    i = np.argmax(preds)
    label = labels[i]
    # draw the activity on the output frame
    text = "activity: {}:".format(label)
    cv2.putText(output, text, (35, 50), cv2.FONT_HERSHEY_SIMPLEX, 1.25, (0, 255, 0), 5)

```

```python
    # check if the video writer is None
    if writer is None:
        # initialize our video writer
        fourcc = cv2.VideoWriter_fourcc(*"MJPG")
        writer = cv2.VideoWriter(output_vid, fourcc, 30, (W, H), True)

    # write the output frame to disk
    writer.write(output)

    # show the output image
    cv2.imshow("Output", output)
    key = cv2.waitKey(1) & 0xFF

    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
        break

# release the file pointers
print("[INFO] cleaning up...")
# writer.release()
vs.release()
```

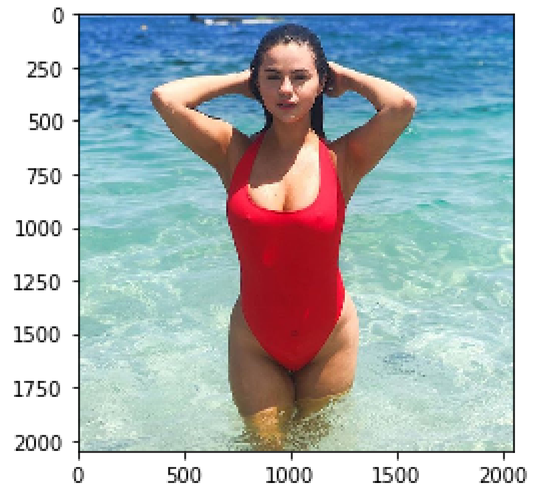
```
[0.8370673  0.0855138  0.07741895]
[0.8351137  0.09766516 0.06722112]
[0.8617761  0.06741591 0.07080795]
[0.85944206 0.06907373 0.0714843 ]
[0.7173338  0.13086972 0.15179652]
[0.7792561  0.10571906 0.11502485]
[0.63839614 0.19626305 0.16534078]
[0.7667552  0.12438072 0.10886402]
[0.75802594 0.13886988 0.10310415]
[0.6523344  0.19880359 0.1488621 ]
[0.76185405 0.12753384 0.11061214]
[0.796366   0.10788083 0.09575319]
[0.7735909  0.12518159 0.10122744]
[0.79267615 0.11130308 0.09602076]
[0.8038922  0.10882613 0.08728163]
[0.8331898  0.08105817 0.085752  ]
[0.97178507 0.01810611 0.01010886]
[0.9750852  0.0159148  0.00900009]
[0.9749875  0.01606384 0.00894867]
[0.968542   0.02118417 0.01027384]
```

In [94]:
```python
# https://stackoverflow.com/a/53403805/7437264
from PIL import Image
import numpy as np
from skimage import transform
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
def load(filename):
    np_image = Image.open(filename)
    np_image = np.array(np_image).astype('float32')/255
    np_image = transform.resize(np_image, (224, 224, 3))
    np_image = np.expand_dims(np_image, axis=0)
    img=mpimg.imread(filename)
    plt.imshow(img)
    return np_image

image = load("2.jpg")
ans = model.predict(image)
maping = {0 : "Neutral", 1 : "Porn", 2 : "Sexy"}
new_ans = np.argmax(ans[0])

print(maping[new_ans], np.round(ans,2))
print("With {} probability".format(ans[0][new_ans]))
```

```
Sexy [[0.01 0.   0.99]]
With 0.9895815849304199 probability
```



Summary

Classification of videos are very similar to classify images and we have to properly process video frames before sending them to classifier as they do effect the output but here we didn't take account of temporal nature,
As purpose of our classifier is to identify the type and if type comes out to be porn or sexy upto a certain threshold then we can block the video content for children.