

Face detection

Name : Nguyễn Sỹ Hà

MSSV : 2151013018

Detect human face in image or video using Haar-like or SIFT descriptors.

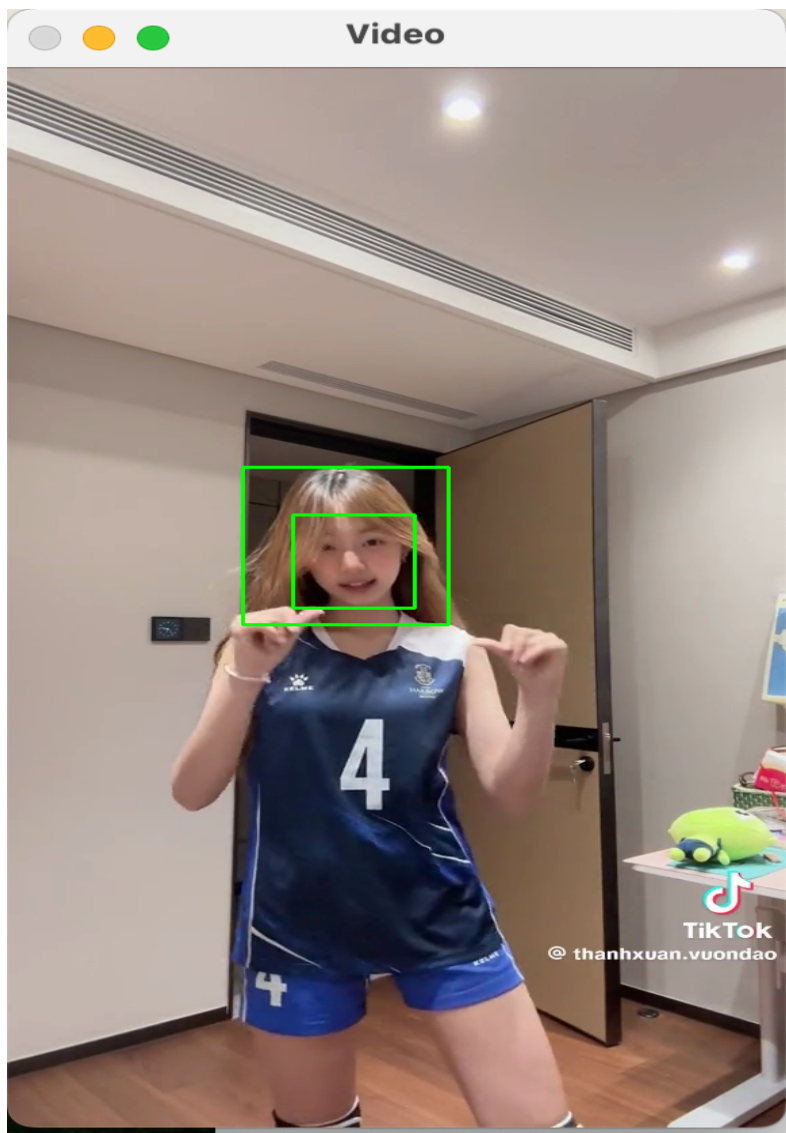
Input: image

Output: image with green rectangles grounding the faces appeared in the input image

Hint: using Haar-like or SIFT descriptors with cascade classifier.

Turn in: one unique PDF with source code + output image

Next assignment: panorama



Code python:

```
import cv2

# Load the pre-trained Haar Cascade face detector
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
'haarcascade_frontalface_default.xml')

# Open the video file
video_capture = cv2.VideoCapture('/Users/syha/Documents/Thị Giác Máy Tính/dance.mp4')

# Get the video properties
fps = video_capture.get(cv2.CAP_PROP_FPS)
width = int(video_capture.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(video_capture.get(cv2.CAP_PROP_FRAME_HEIGHT))

# Define the codec and create a VideoWriter object
fourcc = cv2.VideoWriter_fourcc(*'mp4v') # You can change this to other codecs like
'XVID', 'MJPG', etc.
output_video = cv2.VideoWriter('dance1.mp4', fourcc, fps, (width, height))

# Check if the video file opened successfully
if not video_capture.isOpened():
    print("Error: Could not open video file")
    exit()

# Loop through each frame of the video
while True:
    # Read the next frame
    ret, frame = video_capture.read()

    # If there are no more frames, break out of the loop
    if not ret:
        break

    # Convert the frame to grayscale
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Detect faces in the grayscale frame
    faces = face_cascade.detectMultiScale(gray_frame, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))

    # Draw rectangles around the detected faces
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    # Write the frame to the output video
    output_video.write(frame)
```

```
# Display the frame with detected faces
cv2.imshow('Video', frame)

# Break out of the loop if the 'q' key is pressed
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Release the video capture and writer objects, and close all windows
video_capture.release()
output_video.release()
cv2.destroyAllWindows()
```