Python les-materialen

Usage example of MediaPipe Holistic Solution API in Python (see also http://solutions.mediapipe.dev/holistic).

!pip install mediapipe

Upload any image that that has a person. We take two example images from the web: https://unsplash.com/photos/v4zceVZ5HK8 and https://unsplash.com/photos/e\_rhazQLaSs.

from google.colab import files  
uploaded = files.upload()

import cv2  
from google.colab.patches import cv2\_imshow  
import math  
import numpy as np  
  
DESIRED\_HEIGHT = 480  
DESIRED\_WIDTH = 480  
def resize\_and\_show(image):  
 h, w = image.shape[:2]  
 if h < w:  
 img = cv2.resize(image, (DESIRED\_WIDTH, math.floor(h/(w/DESIRED\_WIDTH))))  
 else:  
 img = cv2.resize(image, (math.floor(w/(h/DESIRED\_HEIGHT)), DESIRED\_HEIGHT))  
 cv2\_imshow(img)  
  
# Read images with OpenCV.  
images = {name: cv2.imread(name) for name in uploaded.keys()}  
# Preview the images.  
for name, image in images.items():  
 print(name)  
 resize\_and\_show(image)

david-hofmann-e\_rhazQLaSs-unsplash.jpg

png

thao-le-hoang-v4zceVZ5HK8-unsplash.jpg

png

All MediaPipe Solutions Python API examples are under mp.solutions.

For the MediaPipe Pose solution, we can access this module as mp\_holistic = mp.solutions.holistic.

You may change the parameters, such as static\_image\_mode and min\_detection\_confidence, during the initialization. Run help(mp\_holistic.Holistic) to get more informations about the parameters.

import mediapipe as mp  
mp\_holistic = mp.solutions.holistic  
  
help(mp\_holistic.Holistic)

# Import drawing\_utils and drawing\_styles.  
mp\_drawing = mp.solutions.drawing\_utils   
mp\_drawing\_styles = mp.solutions.drawing\_styles

# Run MediaPipe Holistic and draw pose landmarks.  
with mp\_holistic.Holistic(  
 static\_image\_mode=True, min\_detection\_confidence=0.5, model\_complexity=2) as holistic:  
 for name, image in images.items():  
 # Convert the BGR image to RGB and process it with MediaPipe Pose.  
 results = holistic.process(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))  
  
 # Print nose coordinates.  
 image\_hight, image\_width, \_ = image.shape  
 if results.pose\_landmarks:  
 print(  
 f'Nose coordinates: ('  
 f'{results.pose\_landmarks.landmark[mp\_holistic.PoseLandmark.NOSE].x \* image\_width}, '  
 f'{results.pose\_landmarks.landmark[mp\_holistic.PoseLandmark.NOSE].y \* image\_hight})'  
 )  
  
 # Draw pose landmarks.  
 print(f'Pose landmarks of {name}:')  
 annotated\_image = image.copy()  
 mp\_drawing.draw\_landmarks(annotated\_image, results.left\_hand\_landmarks, mp\_holistic.HAND\_CONNECTIONS)  
 mp\_drawing.draw\_landmarks(annotated\_image, results.right\_hand\_landmarks, mp\_holistic.HAND\_CONNECTIONS)  
 mp\_drawing.draw\_landmarks(  
 annotated\_image,  
 results.face\_landmarks,  
 mp\_holistic.FACEMESH\_TESSELATION,  
 landmark\_drawing\_spec=None,  
 connection\_drawing\_spec=mp\_drawing\_styles  
 .get\_default\_face\_mesh\_tesselation\_style())  
 mp\_drawing.draw\_landmarks(  
 annotated\_image,  
 results.pose\_landmarks,  
 mp\_holistic.POSE\_CONNECTIONS,  
 landmark\_drawing\_spec=mp\_drawing\_styles.  
 get\_default\_pose\_landmarks\_style())  
 resize\_and\_show(annotated\_image)

Downloading model to /usr/local/lib/python3.7/dist-packages/mediapipe/modules/pose\_landmark/pose\_landmark\_heavy.tflite  
Nose coordinates: (183.17447662353516, 254.24443006515503)  
Pose landmarks of david-hofmann-e\_rhazQLaSs-unsplash.jpg:

png

Nose coordinates: (206.55911922454834, 203.0031123161316)  
Pose landmarks of thao-le-hoang-v4zceVZ5HK8-unsplash.jpg:

png

# Run MediaPipe Holistic and plot 3d pose world landmarks.  
with mp\_holistic.Holistic(static\_image\_mode=True) as holistic:  
 for name, image in images.items():  
 results = holistic.process(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))  
  
 # Print the real-world 3D coordinates of nose in meters with the origin at  
 # the center between hips.  
 print('Nose world landmark:'),  
 print(results.pose\_world\_landmarks.landmark[mp\_holistic.PoseLandmark.NOSE])  
   
 # Plot pose world landmarks.  
 print(f'Pose world landmarks of {name}:')  
 mp\_drawing.plot\_landmarks(  
 results.pose\_world\_landmarks, mp\_holistic.POSE\_CONNECTIONS)

Nose world landmark:  
x: -0.6770456433296204  
y: 0.11645156145095825  
z: -0.20827025175094604  
visibility: 0.9999970197677612  
  
Pose world landmarks of david-hofmann-e\_rhazQLaSs-unsplash.jpg:

png

Nose world landmark:  
x: -0.053249772638082504  
y: -0.6610630750656128  
z: -0.21999822556972504  
visibility: 0.999910831451416  
  
Pose world landmarks of thao-le-hoang-v4zceVZ5HK8-unsplash.jpg:

png

# Run MediaPipe Holistic with `enable\_segmentation=True` to get pose segmentation.  
with mp\_holistic.Holistic(  
 static\_image\_mode=True, enable\_segmentation=True) as holistic:  
 for name, image in images.items():  
 results = holistic.process(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))  
  
 # Draw pose segmentation.  
 print(f'Pose segmentation of {name}:')  
 annotated\_image = image.copy()  
 red\_img = np.zeros\_like(annotated\_image, dtype=np.uint8)  
 red\_img[:, :] = (255,255,255)  
 segm\_2class = 0.2 + 0.8 \* results.segmentation\_mask  
 segm\_2class = np.repeat(segm\_2class[..., np.newaxis], 3, axis=2)  
 annotated\_image = annotated\_image \* segm\_2class + red\_img \* (1 - segm\_2class)  
 resize\_and\_show(annotated\_image)

Pose segmentation of david-hofmann-e\_rhazQLaSs-unsplash.jpg:

png

Pose segmentation of thao-le-hoang-v4zceVZ5HK8-unsplash.jpg:

png