Python les-materialen

Usage example of MediaPipe Face Mesh Solution API in Python (see also http://solutions.mediapipe.dev/face\_mesh).

!pip install mediapipe

Upload any image that contains face(s) to the Colab. We take two example images from the web: https://unsplash.com/photos/JyVcAIUAcPM and https://unsplash.com/photos/auTAb39ImXg

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)

garrett-jackson-auTAb39ImXg-unsplash.jpg

png

radu-florin-JyVcAIUAcPM-unsplash.jpg

png

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

For the MediaPipe Face Mesh solution, we can access this module as mp\_face\_mesh = mp.solutions.face\_mesh.

You may change the parameters, such as static\_image\_mode, max\_num\_faces, and min\_detection\_confidence, during the initialization. Run help(mp\_face\_mesh.FaceMesh) to get more informations about the parameters.

import mediapipe as mp  
mp\_face\_mesh = mp.solutions.face\_mesh  
  
help(mp\_face\_mesh.FaceMesh)

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

# Run MediaPipe Face Mesh.  
with mp\_face\_mesh.FaceMesh(  
 static\_image\_mode=True,  
 refine\_landmarks=True,  
 max\_num\_faces=2,  
 min\_detection\_confidence=0.5) as face\_mesh:  
 for name, image in images.items():  
 # Convert the BGR image to RGB and process it with MediaPipe Face Mesh.  
 results = face\_mesh.process(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))  
  
 # Draw face landmarks of each face.  
 print(f'Face landmarks of {name}:')  
 if not results.multi\_face\_landmarks:  
 continue  
 annotated\_image = image.copy()  
 for face\_landmarks in results.multi\_face\_landmarks:  
 mp\_drawing.draw\_landmarks(  
 image=annotated\_image,  
 landmark\_list=face\_landmarks,  
 connections=mp\_face\_mesh.FACEMESH\_TESSELATION,  
 landmark\_drawing\_spec=None,  
 connection\_drawing\_spec=mp\_drawing\_styles  
 .get\_default\_face\_mesh\_tesselation\_style())  
 mp\_drawing.draw\_landmarks(  
 image=annotated\_image,  
 landmark\_list=face\_landmarks,  
 connections=mp\_face\_mesh.FACEMESH\_CONTOURS,  
 landmark\_drawing\_spec=None,  
 connection\_drawing\_spec=mp\_drawing\_styles  
 .get\_default\_face\_mesh\_contours\_style())  
 mp\_drawing.draw\_landmarks(  
 image=annotated\_image,  
 landmark\_list=face\_landmarks,  
 connections=mp\_face\_mesh.FACEMESH\_IRISES,  
 landmark\_drawing\_spec=None,  
 connection\_drawing\_spec=mp\_drawing\_styles  
 .get\_default\_face\_mesh\_iris\_connections\_style())  
 resize\_and\_show(annotated\_image)

Face landmarks of garrett-jackson-auTAb39ImXg-unsplash.jpg:

png

Face landmarks of radu-florin-JyVcAIUAcPM-unsplash.jpg:

png