Python les-materialen

Usage example of MediaPipe Selfie Segmentation Solution API in Python (see also http://solutions.mediapipe.dev/selfie\_segmentation).

!pip install mediapipe

Upload any person image to the Colab. We take two example images from the web: https://unsplash.com/photos/oB1mqkdDiU0 and https://unsplash.com/photos/fU3EJRO\_qGY.

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)

ilya-mirnyy-fU3EJRO\_qGY-unsplash.jpg

png

nikho-mageza-oB1mqkdDiU0-unsplash.jpg

png

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

For the MediaPipe Selfie Segmentation solution, we can access this module as mp\_selfie\_segmentation = mp.solutions.selfie\_segmentation.

You may change the model selection parameter during the initialization. Run help(mp\_selfie\_segmentation.selfie\_segmentation) to get more informations about the parameter.

import mediapipe as mp  
mp\_selfie\_segmentation = mp.solutions.selfie\_segmentation  
  
help(mp\_selfie\_segmentation.SelfieSegmentation)

# Show segmentation masks.  
BG\_COLOR = (192, 192, 192) # gray  
MASK\_COLOR = (255, 255, 255) # white  
  
with mp\_selfie\_segmentation.SelfieSegmentation() as selfie\_segmentation:  
 for name, image in images.items():  
 # Convert the BGR image to RGB and process it with MediaPipe Selfie Segmentation.  
 results = selfie\_segmentation.process(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))  
   
 # Generate solid color images for showing the output selfie segmentation mask.  
 fg\_image = np.zeros(image.shape, dtype=np.uint8)  
 fg\_image[:] = MASK\_COLOR  
 bg\_image = np.zeros(image.shape, dtype=np.uint8)  
 bg\_image[:] = BG\_COLOR  
 condition = np.stack((results.segmentation\_mask,) \* 3, axis=-1) > 0.2  
 output\_image = np.where(condition, fg\_image, bg\_image)  
  
 print(f'Segmentation mask of {name}:')  
 resize\_and\_show(output\_image)

Segmentation mask of ilya-mirnyy-fU3EJRO\_qGY-unsplash.jpg:

png

Segmentation mask of nikho-mageza-oB1mqkdDiU0-unsplash.jpg:

png

# Blur the image background based on the segementation mask.  
with mp\_selfie\_segmentation.SelfieSegmentation() as selfie\_segmentation:  
 for name, image in images.items():  
 # Convert the BGR image to RGB and process it with MediaPipe Selfie Segmentation.  
 results = selfie\_segmentation.process(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))  
  
 blurred\_image = cv2.GaussianBlur(image,(55,55),0)  
 condition = np.stack((results.segmentation\_mask,) \* 3, axis=-1) > 0.1  
 output\_image = np.where(condition, image, blurred\_image)  
   
 print(f'Blurred background of {name}:')  
 resize\_and\_show(output\_image)

Blurred background of ilya-mirnyy-fU3EJRO\_qGY-unsplash.jpg:

png

Blurred background of nikho-mageza-oB1mqkdDiU0-unsplash.jpg:

png