Ex No:08	Pose Estimation
Date:	

Aim:

To write python code to perform Pose Estimation

Algorithm:

Step 1 : Start the program

Step 2 : Import cv2

Step 3 : Initialize Pose estimator

Step 4 : Read an image

Step 5 : Convert the image to RGB format

Step 6 : Process the RGB image to get the result

Step 7 : Draw detected skeleton on the image

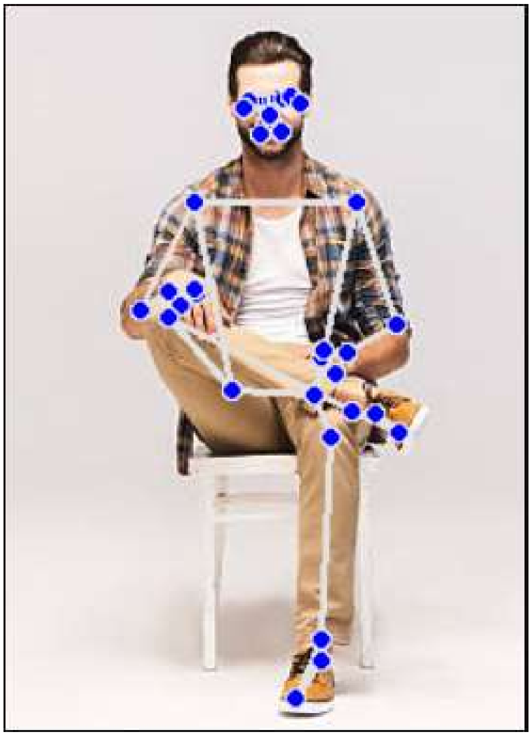
Step 8 : Show the final output image

Step 9 : Stop the program

Program:

```
import cv2
import mediapipe as mp
from google.colab.patches import cv2_imshow
mp_drawing = mp.solutions.drawing_utils
mp_pose = mp.solutions.pose
pose = mp_pose.Pose(
   min_detection_confidence=0.5,
   min_tracking_confidence=0.5)
image = cv2.imread('sitting.jpg')
RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
results = pose.process(RGB)
image_with_landmarks = RGB.copy() # Create a copy to avoid modifying the original image
mp_drawing.draw_landmarks(
   image_with_landmarks, results.pose_landmarks, mp_pose.POSE_CONNECTIONS)
cv2_imshow( cv2.cvtColor(image_with_landmarks, cv2.COLOR_RGB2BGR))
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:



Result:

Thus the code for pose estimation has been executed successfully and the output is verified.

Ex No:09	**3D Reconstruction-creating depth map from stereo images**
Date:	

Aim:

To write python code to perform 3D Reconstruction-creating depth map from stereo images

Algorithm:

Step 1 : Start the program

Step 2 : Import cv2

Step 3 : Initialize Pose estimator

Step 4 : Read an image

Step 5 : Use StereoBM

Step 6 : Compute disparities

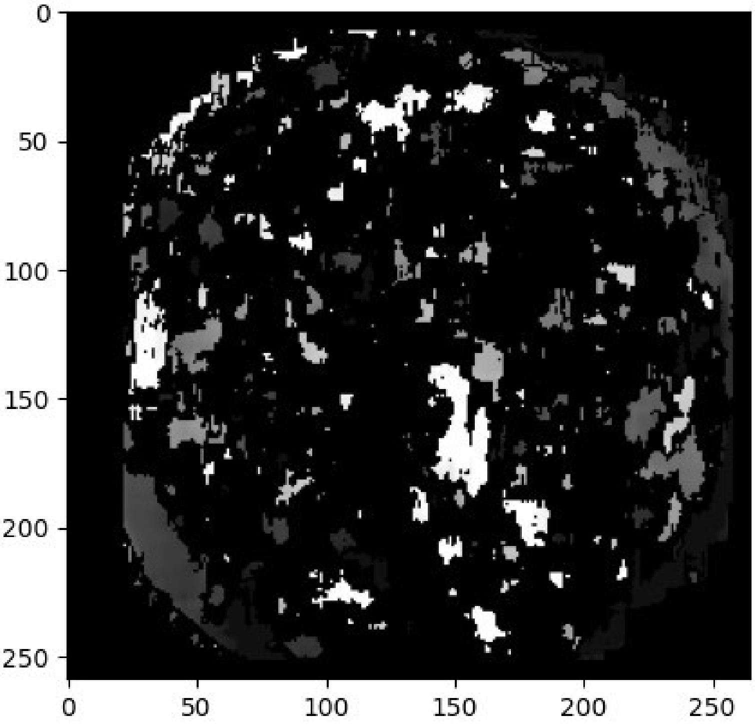
Step 7 : Show the final output image

Step 8 : Stop the program

Program:

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt
left_image = cv2.imread('left.jpg', cv2.IMREAD_GRAYSCALE)
height, width = left_image.shape
right_image = cv2.imread('right.jpg', cv2.IMREAD_GRAYSCALE)
right_image = cv2.resize(right_image, (width, height))
stereo = cv.StereoBM.create(numDisparities=16, blockSize=15)
disparity = stereo.compute(left_image,right_image)
plt.imshow(disparity,'gray')
plt.show()
```

Output:



Result:

Thus the code for 3D Reconstruction-creating depth map from stereo imageshas been executedsuccessfully and the output is verified.