



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
مکاترونیک

گزارش مینی پروژه ۲

نام و نام خانوادگی	کامیار رحمانی
شماره دانشجویی	۸۱۰۱۹۹۴۲۲
تاریخ ارسال گزارش	۱۴۰۲/۱/۱۸

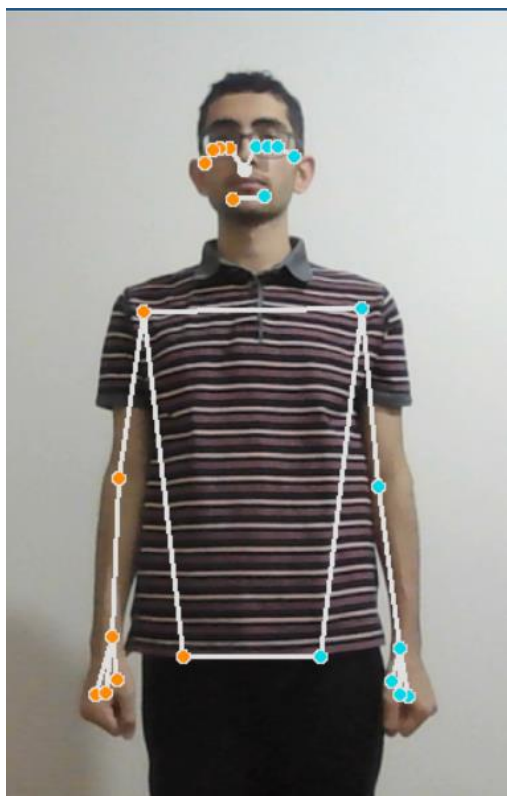
فهرست گزارش سوالات

بخش اول : کار با Mediapipe و pose estimation ۳

بخش دوم : Face detection ۶

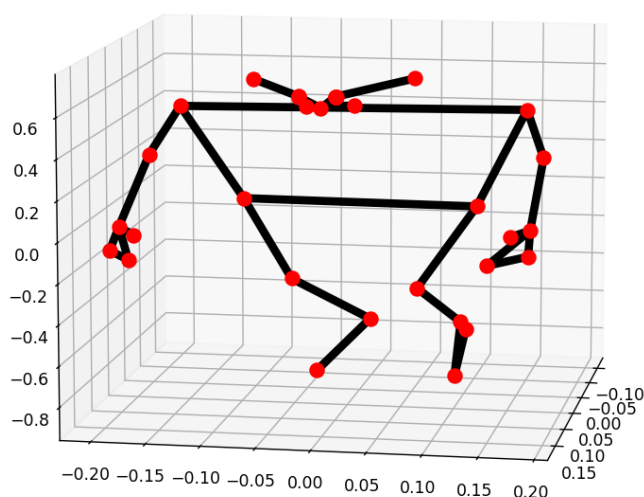
بخش اول : کار با Mediapipe و pose estimation

در ابتدا با ران کردن کدی که در اختیار ما قرار گرفته بود نحوه کارکرد pose estimation را مشاهده کردیم :

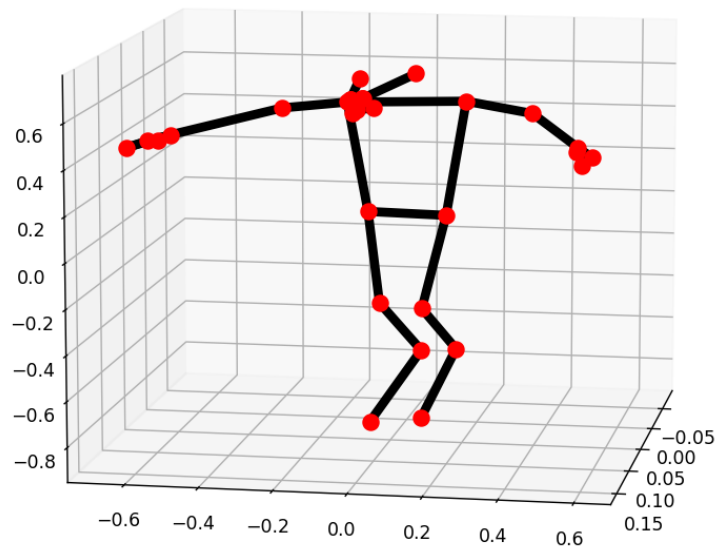


می بینیم که نقاط کلیدی نمایش داده شده (به دلیل محدودیت frame و بکم هر ۳۳ نقطه در تصویر قرار نگرفته است)

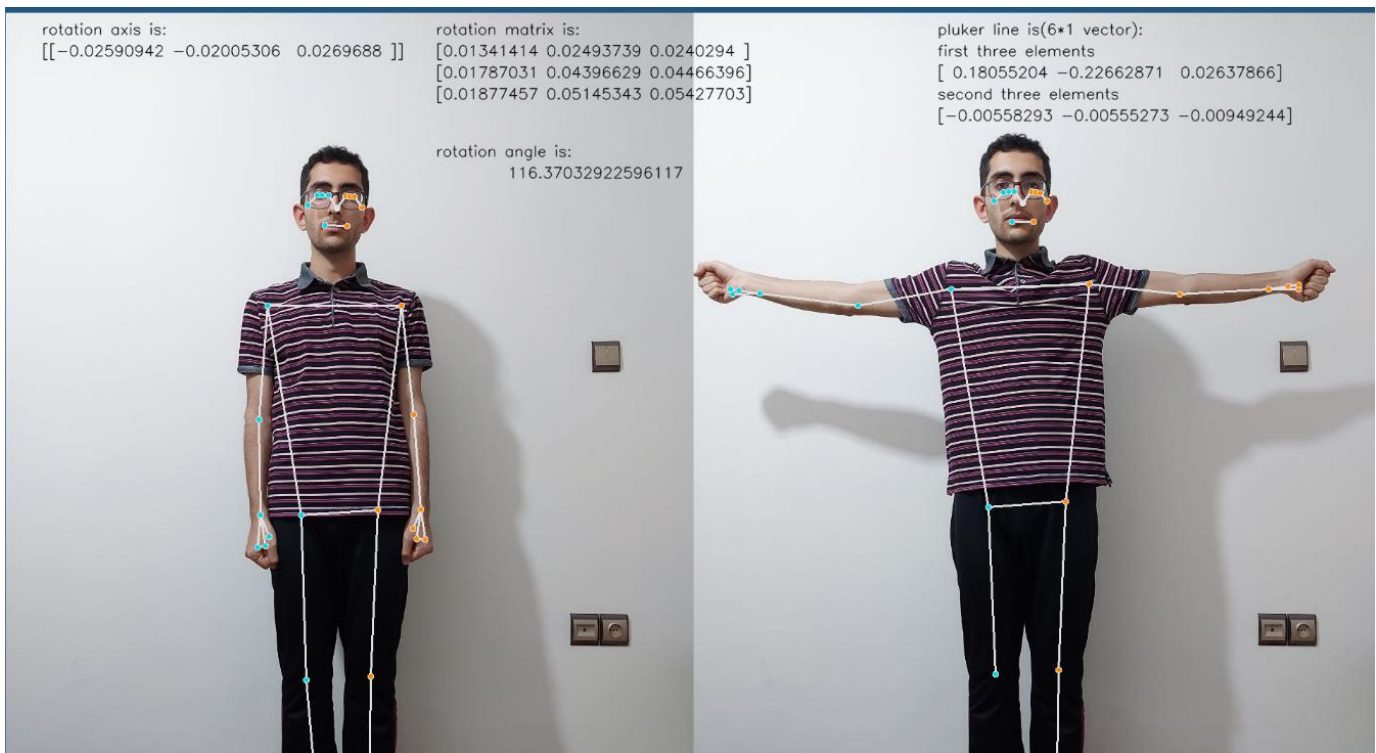
در دو حالت مختلف از خود تصویر برداری کردم و با استفاده از کد پایتون خروجی مد نظر را بدست آوردم. (دترمینان ماتریس دوران نیز به یک تغییر داده شده است)
مختصات سه بعدی بدن در حالت ۱ :



مختصات سه بعدی بدن در حالت ۱ :



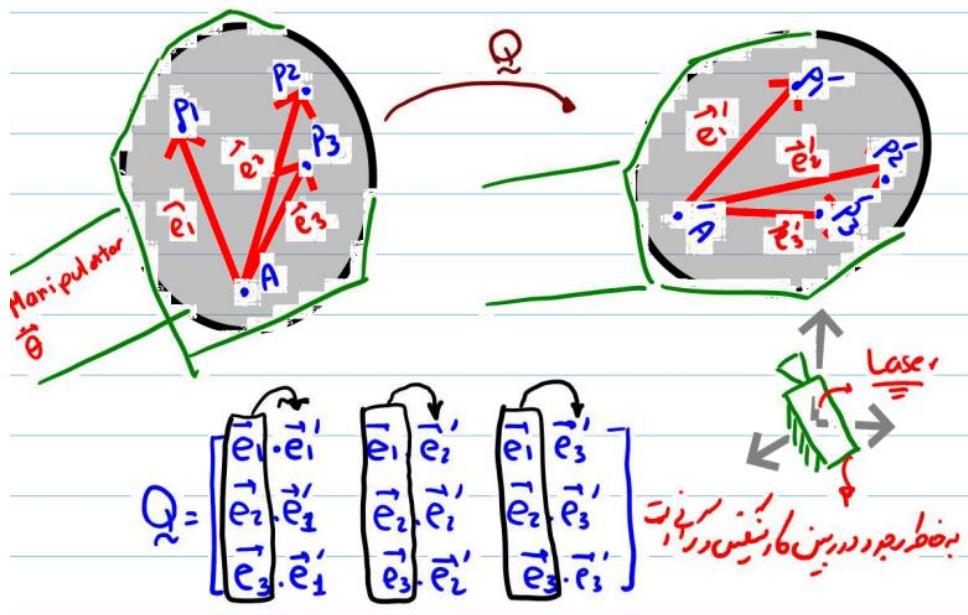
خروجی نهایی و بدست آوردن ماتریس دوران، محور دوران، زاویه دوران و خط پلوکر :



ماتریس دوران : (بعد از یک کردن دترمینان)

$$Q = \begin{pmatrix} 2.43739711 & 4.53121349 & 4.36622988 \\ 3.24709944 & 7.98883433 & 8.11560274 \\ 3.41140754 & 9.34927455 & 9.8623333 \end{pmatrix}$$

برای بدست آوردن ماتریس دوران از رابطه زیر استفاده کردیم :



طبق شکل فوق یک نقطه را به عنوان مبدا در نظر گرفتیم که همان شانه یا LEFT_SHOULDER است و سه نقطه دیگر شامل LEFT_WRIST و LEFT_INDEX و LEFT_ELBOW را نیز در نظر گرفتیم. با بدست آوردن بردارهای وصل کننده این سه نقطه به نقطه مبدا (شانه) بردارهای e_1, e_2, e_3 و همچنین بردارهای e_1_{prime} و e_2_{prime} و e_3_{prime} نیز بدست خواهند آمد. و در نهایت با نوشتن تابعی طبق رابطه بالا ماتریس دوران را بدست می آوریم.

برای بدست آوردن مختصات پلوکر، همانطور که می دانیم نمایش پلوکری به صورت زیر است :

$$\kappa = \begin{bmatrix} \mathbf{e} \\ \mathbf{n} \end{bmatrix}$$

که در رابطه بیان شده e برابر راستای خط و n همان مماس خط می باشد.
 می دانیم که n همان ضرب داخلی بردار p_0 در راستا یا همان e است.
 بردار p_0 از رابطه زیر محاسبه می شود :

$$p_0 = \frac{(Q - 1)^T (Qa - a')}{2(1 - \cos \phi)},$$

که در رابطه بالا a یک نقطه از جسم صلب و a_{prime} تغییر یافته همان بردار است. در کد من این نقطه را آرنج در نظر گرفتیم و p_0 را محاسبه کردم و در نهایت با استفاده از آن نمایش پلوکری را پیدا کردم.

بخش دوم : Face detection

(الف)

در این بخش وقتی کد را اجرا کردیم عکس ها شناسایی شدند. کد را به گونه ای تغییر دادم که تعداد صورت های شناسایی شده یا همان TP ها را بشمارد که تعداد آنها 42 تا بود.

Number of detected faces 42

سایر عناصر confusion matrix شامل FN , FP را به صورت دستی با شمردن حساب کردم که :

تعداد عکس هایی که در آنها چهره وجود داشته ولی شناسایی نشده است :

$$FN = 13$$

تعداد عکس هایی که در آنها چهره وجود نداشته ولی مدل چهره شناسایی کرده است :

$$FP = 0$$

تعداد TN را نیز بینهایت گرفتیم چون عملاً بینهایت object به جز face در تصویر است که به درستی شناسایی نشده است :

$$confusion_matrix = \begin{pmatrix} 42 & 13 \\ 0 & \infty \end{pmatrix}$$

محاسبه precision و recall :

$$precision = \frac{TP}{TP + FP} = \frac{42}{42} = 1$$

$$recall = \frac{TP}{TP + FN} = \frac{42}{42 + 13} = \frac{42}{55} = 0.764$$

هم چنین میدانیم که شکل کلی ماتریس آشفته به صورت زیر است :

		Predicted Class		
		Positive	Negative	
Actual Class	Positive ω_2	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative ω_1	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

(ب)

خروجی کد ویرایش شده :



در این بخش برای کشیدن مستطیل سیاه از مختصات چشم چپ و راست استفاده کردم ولی چون مبدا در تصویر بالا گوشه سمت چپ تصویر در نظر گرفته می شود یک scaling به اندازه طول و عرض تصویر انجام دادم و در نهای با اضافه کردن مقداری آفست پهنای مستطیل را هم در راستای x و هم در راستای z تغییر دادم.