

**Electrical Engineering Department**

Multimedia – Dr.Sharifian

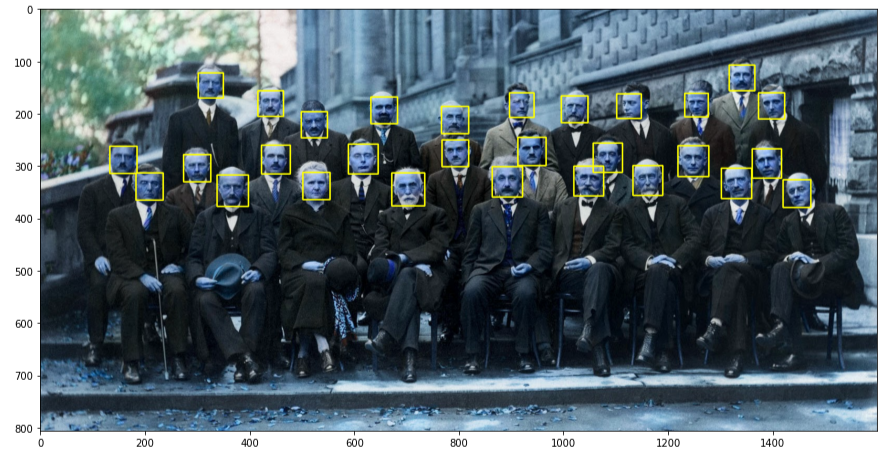
**Report HW 2**

نام دانشجو: علی بابالو – ۹۸۲۳۰۱۲

ایمیل :‌ [alibabaloo@aut.ac.ir](mailto:alibabaloo@aut.ac.ir)

**سوال ۱:**

برای قسمت اول برای تشخیص تعداد چهره های عکس از مدل HAAR frontface استفاده کردیم که شامل ۳ مدل default, alt, alt2 می­باشد که ۲ مدل دیفالت و alt2 را با یکدیگر مقایسه کردم که در مدل alt2 نتایج به درستی و تعداد چهره ها ۲۹ عدد تشخیص داده شد. برای رسیدن به این نتیجه مقدار پارامتر minNeighour را تغییر میدادم تا به نتیجه مطلوب برسم.

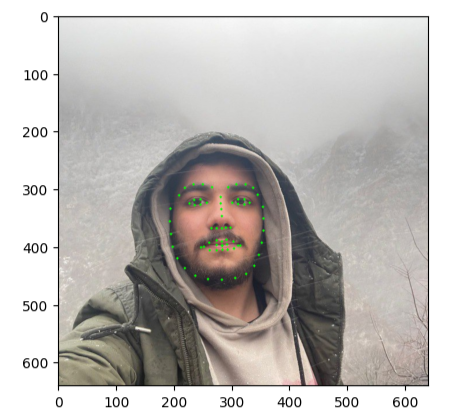


برای قسمت دوم این سوال بعلت اینکه لپتاپ بنده وب­کم ندارد از یک ویدیو از پیش ظبط شده استفاده کردم. ابتدا ویدیو را با استفاده از videocapture خوانده و با تابع read هر فریم آن را در حلقه می­خوانیم. سپس برای هر فریم از مدلی که در قسمت قبل استفاده کردیم بهره می­بریم تا چهره را تشخیص دهد و آن را با مستطیل نشان دهد. برای اتمام ویدیو نیز ۲ روش قرار دادیم یکی بعد از اینکه فریم آخر خوانده شد بصورت خود به خود خارج می­شود حالت دوم بدین صورت است که با فشردن کلید esc می­توان خارج شد.

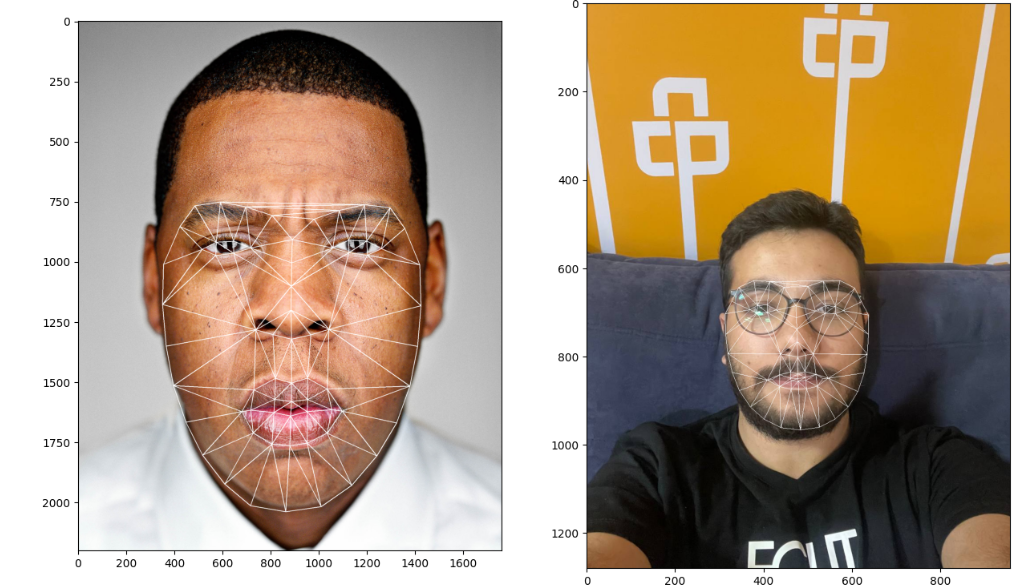
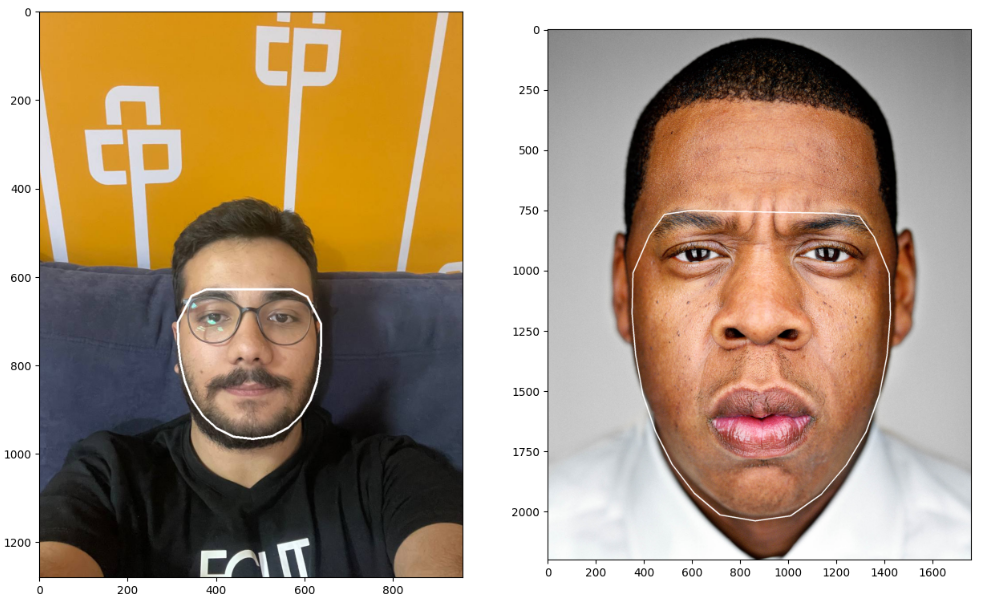
براي سنجش عملكرد كد نيز زمان مورد نياز براي پردازش 200 فريم محاسبه شده و از روي آن ميانيگن زمان پردازش هر فريم و نيز FPS به دست مي آيد. مقدار FPS طبيعتاً به ميزان پردازش مورد نياز براي هر فريم بستگي دارد. يكي از عوامل اصلي در اين مورد اندازه ي تصوير است، هر چه اندازه ي تصوير بزرگ تر باشد، ميزان پردازش مورد نياز بيش تر و FPS كمتر است و برای این ویدیو از سایز ۵۰۰\*۷۲۰ استفاده کردم که بنسبت سایز بزرگی است و به همین دلیل مقدار FPS کم شده است.

**سوال ۲:**

برای قسمت اول این سوال بعلت پولی بودن نتوانستم تصویر خوبی از fotor جنریت کنم بهمین علت از تصویر آقای Jay-Z استفاده کردم تا لندمارک ها را بر روی آن تشخیص دهم. برای پیدا کردن نقاط اساسی از کتابخانه dlib و مدل arks\_face\_68\_predictor\_s استفاده کردم که 68 نقطه اساسی بر روی صورت را مشخص می­کند و دایره های آن را بر روی تصویر سلبریتی و همچنین تصویر خودم کشیدم.



برای قسمت دوم و تعویض 2 چهره، ابتدا نقاط کلیدی را مانند قسمت قبل بدست می­آوریم سپس بدور چهره یک covex hull میکشیم و این نقاط را با توجه به درس استاد بصورت مثلثی به یکدیگر متصل می­کنیم.

برای قسمت بعد تنها کافی است آن مثلث هارا از تصویر یک به تصویر دوم منتقل کنیم(البته چون اسکیل تصاویر با هم یکی نبودند تصویر نهایی خیلی خوب نشد) براي اين كار از متدهاي getAffineTransform و warpAffine استفاده می کنیم؛ متد اول با در اختيار داشتن دو مثلث تبديلي را به ما مي دهد كه با اعمال آن بر يك تصوير، مثلث اول به مثلث دوم تبديل مي شود، متد دوم نيز اين تبديل به دست آمده را بر روي تصوير اعمال مي كند. در نهایت نتیجه بدست آمده را روی تصویر دیگر می­اندازیم.

**پایان**