

۱. برای حل این سوال به دیتای صورت و غیرصورت نیاز داریم که برای دیتای غیر صورت را از داده‌های سوال قبل استفاده می‌کنیم (داده‌های منظره).

در ابتدا عکس‌های صورت و غیرصورت را در تابع `read_images` به صورت سیاه و سفید می‌خوانیم و `integral` آن‌ها را محاسبه می‌کنیم (خروجی این تابع عکس‌های اینتگرال شده می‌باشد). چون تعداد داده‌ها زیاد بود و ران کردن زمان زیادی نیاز داشت تنها ۱۵۰۰ تا از داده‌های صورت و همچنین ۱۰۰۰ تا از داده‌های غیرصورت را برای `train` کردن استفاده می‌کنیم.

سپس برای هر `haar_fiture` در تابع `hash_haars` یک کد می‌سازیم که با استفاده از آن کد بتوانیم برای اینتگرال عکس‌ها مقدار آن `haar_fiture` را محاسبه کنیم. در این تابع از تابع‌های `hash_haar_square` و `hash_haar_triple` و `hash_haar_domino` استفاده شده است که هر یک از این تابع‌ها یک تایپ از هار را محاسبه می‌کند، کد ساخته شده برای هر هار فیچر شامل مختصات بالا-چپ، طول در راستای x ، طول در راستای y ، تایپ هار و همچنین افقی یا عمودی بودن آن می‌باشد.

سپس در تابع `calculate_haar` با کمک این کدها مقدار هر هار را روی عکس‌های صورت و غیرصورت محاسبه می‌کنیم و دیتای X و y را برای آدابوست مهیا می‌کنیم.

سپس ۱۰ استیج داریم که در هر استیج ۴ فیچرها وجود دارد که این‌ها را با کمک تابع `train` (که در آن `adaboost` پیاده سازی شده است) بدست می‌آوریم و سپس برای عکس داده شده فیس‌ها را در تابع `extract_faces` بدست می‌آوریم.

(آ) `train`: در این تابع الگوریتم `adaboost` پیاده سازی شده است به این صورت که در ابتدا وزن هر کلاس برابر و مجموع آن‌ها برابر با ۱ می‌باشد، سپس به تعداد `iter_num` که در اینجا برابر با ۴ است (تعداد کلاسیفایرها در هر استیج) بهترین فیچرها را با کمک تابع `get_best_classifier` بدست می‌آوریم و α و وزن‌های جدید را با توجه به فرمول گفته شده در کلاس محاسبه می‌کند، خروجی این تابع ۴ بهترین هار فیچر می‌باشد که برای هر کدام شماره‌ی آن فیچر و آلفا و ترشهلد و جهت ترشهلد را برمی‌گرداند.

(ب) `get_best_classifier`: در این تابع دیتای X (مقدار هار فیچرها برای همه‌ی عکس‌ها)، y (لیبل هر

عکس) و همچنین وزن‌های هر عکس ورودی داده می‌شود و این تابع از روی فیچرهای باقیمانده بهترین فیچر(فیچری که ارور آن کمترین باشد) را انتخاب می‌کند.

(ج) `extract_faces`: در این تابع هارفیچرهای خوب استخراج می‌شوند و سپس به ازای سایزهای مختلف عکس داده شده `window`های 24×24 از این عکس استخراج می‌شود و برای هر کدام با کمک فیچرهای هر استیج α و `streshhold` که داریم در تابع `calculate_adaboost` محاسبه می‌کنیم که آیا این عکس صورت می‌باشد یا نه، در اولین استیجی که این عکس را غیرصورت شناخت دیگر ادامه‌نمیدهد و اگر در همه‌ی استیج‌ها این عکس صورت شناخته شده بود، این عکس را به عنوان صورت انتخاب می‌کند و در فلدر `faces` ذخیره می‌کند.

برای ران کردن این کد `BASE_DIR` را برابر با پوشه ای که در آن هستید قرار دهید و سپس از `python prob1.py` استفاده کنید. این کار به `ram` بسیار زیادی نیاز دارد به همین دلیل روی سرور ران کردم اما بسیار طول می‌کشد به‌همین دلیل تا مهلت ارسال قادر به مشاهده‌ی نتیجه نشدم.

۲. در این سوال ابتدا دیتای هر فرد را می‌خوانیم و آن دیتا را به `tran` و `validation` و `test` تقسیم می‌کنیم.

سپس در تابع `get_cov_matrix` ماتریس‌های کواریانس و میانگین صورت‌ها محاسبه می‌شوند به این صورت که همه‌ی عکس‌ها به‌صورت ستونی در ماتریس `data_matrix` ذخیره می‌شوند و `cov_mat` برابر با حاصل ضرب `data_matrix` در ترانپوز خودش می‌باشد. میانگین عکس‌ها در عکس `mean_faces.jpg` نمایش داده شده است.

سپس در تابع `pca` بردارویژه‌های ماتریس کواریانس به صورت نرمال محاسبه می‌شوند و به ترتیب مقادیر ویژه‌ی آن‌ها مرتب می‌شوند. در این مرحله چون بردار ویژه‌ها نرمال هستند عکس آن‌ها واضح نیست (عکس‌های با فرمت `eig_faces*.jpg`) به همین دلیل بردارها را در فایل `eig_vectors` ذخیره کرده‌ام.

سپس در تابع `train` بهترین تعداد بردار ویژه‌ها برای کلاسیفای کردن انتخاب می‌شود به این صورت که برای هر d که تعداد بردارویژه‌های انتخاب شده می‌باشد، به ازای هر عکس `strain` و هر عکس `validation` ضرایب بردار ویژه‌ها برای ساختن آن عکس را محاسبه می‌کند و به کمک `knn` که در آن $k = 1$ می‌باشد، کلاس هر داده‌ی `validation` را انتخاب می‌کند و چک می‌کند که درست انتخاب کرده یا نه، و در نهایت بهترین d را انتخاب می‌کند که ۱۲۸ می‌باشد.

سپس `score`، d منتخب را برای داده‌های `test` محاسبه می‌کند که برابر با ۸۰٪ می‌باشد. (که این `score` برابر با تعداد عکس‌هایی است که درست پیشبینی شده‌اند).

برای ران کردن کد این سوال `BASE_DIR` را برابر با آدرس پوشه‌ای که در آن هستید قرار دهید و سپس `python prob۲.py` را اجرا کنید.