

در این سوال برای داده های منفی negative از مجموعه داده Caltech-256 استفاده شده است که برخی پوشه های آن که شامل چهره میشد حذف شده اند هر چند تعدادی چهره در بین دیتاست به این بزرگی، مدل من را دچار خطای بزرگی نمیکرد و قبل از حذف نیز نتایجی به همین اندازه مطلوب بدست آوردم.

از آنجایی fit کردن مدل svm برای من با 20000 داده ی train بسیار زمان بر بود، آزمایش کردن مقادیر مختلف هایپر پارامتر ها بسیار طولانی و زمانبر میشد. برای همین از تعداد 1000 داده منفی (به صورت رندوم) و 400 داده مثبت (بصورت رندوم) برای داده ی آموزش train استفاده کردم. دلیل اینکه تعداد داده های منفی را بیشتر در نظر گرفتم این بود که تعدادی ناحیه اضافه اشتباه عنوان چهره تشخیص داده میشد (مثلا زانوی بازیکن ها) که با زیاد کردن داده های منفی، ماشین آموزش دید که گزینه های متنوع تری را بعنوان تصویر منفی تشخیص دهد. هرچند با توجه به دقت مدل یا false positive ها نیز میشد پی برد که تعداد داده های منفی بیشتر باشد، دقت بهبود میابد. همچنین به تعداد 100 داده مثبت و 100 داده منفی برای test و validation استفاده کردم.

تصاویر چهره (داده های مثبت) را از چهار طرف به اندازه ی مساوی crop کردم تا ابعاد آن 140×100 شود (ارتفاع 140 و عرض 100). ولی داده های منفی را با resize کردن به این ابعاد رساندم. این ابعاد به طور تجربی بدست آمدند. سپس بردار ویژگی های داده ها را با استفاده از تابع hog از پکیج skimage.feature با پارامتر های اندازه هر سلول 8×8 پیکسل و اندازه هر بلوک 4×4 سلول

`pixels_per_cell=(8, 8), cells_per_block=(4, 4)` بدست آورده ام. سپس یک مدل svm با کرنل rbf و gammaی auto و سایر پارامتر ها که همان پیش فرض هست را به بردار های ویژگی داده های آموزش fit کردم. دقت میانگین (AP) مدل در آخرین اجرا برابر شد با 98.5٪ که از تابع آماده average_precision_score بدست آمده. (تصاویر ضمیمه شده (نتایج) مربوط به همین آخرین اجرا میباشد) البته از آنجایی که داده های آموزش و تست و اعتبارسنجی اولیه به طور رندوم از دیتاست انتخاب میشوند، با هر بار اجرا، اندکی تغییر در میزان دقت میانگین Ap رخ میدهد ولی به طور معمول این مقدار در حدود 99٪ بوده (البته اکثر اوقات 100 درصد هم میشده).

حال در گام بعدی برای یافتن چهره های عکس های داده شده از هرم استفاده شده. هرمی که سه level یا سطح دارد و در هر سطح خود به دنبال چهره هایی با سایز مشخص میگردد. برای آنکه برنامه ی ما جامع تر شود میتوان تعداد level های هرم را زیاد کرد تا چهره با انواع اندازه ها را یافت کند ولی از آنجایی که اجرای هر level هرم با sliding window زمانبر هست این سه سطح هرم طوری تعیین

شده اند که تمام چهره های سه تصویر فوتبالیست ها را با اینکه ابعاد چهره های مختلفی دارند (از نظر پیکسلی) به خوبی پیدا کنند. نکته مهمی که در هرم ما وجود دارد این است سائز تصویر ثابت بوده و در هر level هرم، ابعاد sliding window تغییر میکند (زیرا بزرگ کردن ابعاد تصویر اصلی زمان اجرای sliding window را بسیار طولانی کرد و ترجیه دادم که ابعاد خود sliding window را بجای تصویر اصلی بزرگ کنم). در هر level هرم یک sliding window زده شده است که چهره هایی با scale خاصی را پیدا کند. این کار با دو for تو در تو با پرش گام 10 پیکسل 10 پیکسل انجام شده که تمام مستطیل های ممکن را بررسی میکند و اگه مستطیلی پیدا کرد که predict کردن آن جواب مثبت دهد، آن را بعنوان چهره تشخیص داده و location مستطیل را در آرایه ای ذخیره میکند. در آخر تابعی پیاده سازی شده که تقریبا کار nonmaximal suppression را انجام میدهد. به این صورت که از بین مستطیل هایی که فاصله ای کمتر از یک threshold ای داشتند، بزرگترین مستطیل را نگه دارد و بقیه مستطیل های نزدیکش را حذف کند. سپس مستطیل های باقی مانده را رسم کند. این threshold تابعی از ابعاد چهره هایی است که detect میشود.

نکته: برای آنکه سمت راست ترین چهره در تیم پرسپولیس (و کلا هر چهره ای که در کناره ی تصویر قرار دارد) باید ابتدا تصویر داده شاده را به اندازه ای pad کنیم تا sliding window بتواند آزادانه در لبه های تصویر حرکت کند و چهره های کناره زمین را نیز تشخیص دهد.

نکته: از تصاویر رنگی در این داده ها استفاده شده.

نکته: تابعی با نام findFace پیاده سازی شده که یک تصویر و یک عدد اعشاری k میگیرد و تمام چهره هایی که scale ای برابر k دارند را یافت میکند. برای مثال اگر $k=0.6$ باشد تمام چهره هایی که ابعاد پیکسلی آنها 0.6 ابعاد چهره های داخل دیتاستمان (که قرار دادیم 140×100) هست را یافت میکند.

مثلا: $84=0.6 \times 140$ و $60=0.6 \times 100$ پس یعنی تمام چهره هایی که ارتفاع آنها 84 و عرض آنها 60 پیکسل هست به ازای $k=0.6$ هست، یافت میشود. پس کافیت در هر level هرم یک scale تعیین کنیم و هر level هرم را اجرا کنیم تا مستطیل هایی پیدا کند. این سه scale برای سه level هرم ما برابر هستند با 0.6 و 0.9 و 1.5 که میتوانیم برای پیدا کردن چهره های بزرگتر و یا کوچکتر تنها scale های جدیدی به لیست scales در تابع FaceDetector اضافه کنیم.