



## ۱ استخراج ویژگی

برای این بخش، ابتدا ما به راه های معمول مثل LBP و استفاده از هیستوگرام فکر کردیم و در ابتدا یک classifier بر روی هیستوگرام LBP تصاویر پیاده سازی کردیم که دقتی در حدود ۶۶.۰ میداد.

در گام بعدی ما سعی کردیم کمی مراحل پیش پردازش را سرعت دهیم تا بتوانیم به راحتی یک مدل را در کانفیگ های مختلف تست بگیریم، برای این کار تابع extraction feature را با استفاده از multiprocessing به طور موازی پیاده کردیم.

سپس سعی کردیم که دقت مدل را بهبود دهیم، اولین قدم ما افزایش ویژگی هایی بود که classifier بر پایه آن ها تصمیم گیری میکرد، پس فوری و هیستوگرام رنگی تصویر را نیز اضافه کردیم ولی برای هیستوگرام رنگی به جای اینکه در فضای RGB هیستوگرام را حساب کنیم، ابتدا تصویر را به فضای HSV بردیم. زیرا به نظرمان به طور خاص saturation میتواند کمکمان کند. زیرا برخلاف تصویر زنده که درخشش نور روی جاهای خاصی همانند چشم، گونه و یا پیشانی است، در LCD درخشش نور به طور بازتاب یک منبع نوری خارجی مثل چراغ، خورشید و یا فلش دوربین است و saturation در یک ناحیه دایره ای پیک میزند. استفاده از saturation باعث شد که مدل ما دچار بیش برازش هم نشود و به دقت هشتاد درصد برسیم.

مرحله بعدی کار ما tune کردن classifier بود. ما برای classify کردن تصاویر از یک SVM با کرنل rbf بود که متغیر های آن را هیچ تغییری نداده بودیم. اول کمی سعی کردیم C را تنظیم بکنیم تا موارد اشتباه را کمتر کنیم، سپس سعی کردیم ایده ای مشابه forest random را روی svm پیاده سازی کنیم. به اینصورت که به جای آموزش یک SVM با هزار داده، ۵ svm با ۲۰۰ داده آموزش دهیم و بین آن ها رای بگیریم، ولی متأسفانه این موضوع دقت ما را افزایش نداد.

مراحلی که میخواستیم سراغ آن برویم ولی وقت نشد از قرار زیر است:

- استفاده از ابزاری هایی مثل detection line hough برای تشخیص مستطیل
- اضافه کردن SIFT به ویژگی های استخراج شده.
- استفاده از augmentaion data که بتوانیم نا همسانی داده را کنترل کنیم

### ۱.۱ فوری

برای قسمت فوری، ما به جای اینکه از تصویر خاکستری فوری بگیریم، از کانال s تصویر فوری گرفتیم، دلیل این امر این بود که همانطور که گفتیم تغییرات s برای ما به مراتب مهم تر از تغییرات سایر چنل ها (چه RGB و چه HSV) است. با این کار توانستیم به دقت ۸۴ درصد برسیم. کار دیگری که برای این موضوع کردیم، این بود که سعی کردیم عناصر فرکانس پایین را حذف کنیم تا تمرکز classifier بیشتر به سمت جزئیات برود ولی این موضوع خیلی تاثیری نداشت. یکی دیگر از کارهای ما این بود که به جای خود فوری، از log آن استفاده کردیم تا اختلاف بین اعضا زیاد نشود، زیرا در شبکه های عصبی، بهتر است که داده به صورت نرمال بیاید ولی ما چون مقدار بیشینه را نداشتیم نمیتوانستیم به صورت کامل داده ها را نرمالیزه کنیم پس به گرفتن لوگاریتم اکتفا کردیم.

## ۲ یادگیری عمیق

برای این بخش، ما پایه مدل را مدل از پیش آموخته mobilenet با وزن های imagenet گرفتیم، زیرا پارامتر های آن به نسبت شبکه های دیگر همچون VGG و YOLO ... کمتر بود (هرچند بعداً سعی کردیم VGG را هم امتحان کنیم) ما همه لایه های mobilenet را فریز کردیم و لایه نهایی آن را حذف کردیم و به جای آن یک لایه FC برای تصمیم گیری نهایی با تابع فعال ساز softmax در نظر گرفتیم، و برای شروع epoch ۲۰ برای آموزش در نظر گرفتیم ولی مشاهده کردیم که تقریباً پس از epoch ۷ مدل اشباع شد و به دقت ۱۰۰ رسید ولی وقتی رو داده های تست مدل را ران کردیم، دقت ۹۰ گرفتیم و دیدیم overfit رخ داده، برای این کار یکی از اولین کارهایی که کردیم اضافه کردن یک لایه dropout قبل از لایه FC بود. سپس سعی کردیم که با تغییر عکس ها مسئله را برای