

به نام خدا

امیر محمد کمیجانی ۹۹۵۲۲۰۳۲

محمد حسین میرزایی ۹۹۵۲۲۱۵۸

توضیحات کد :

ابتدا با توجه به نیاز های برنامه کتابخانه های مدنظر را ایمپورت کردیم.

سپس فایل دیتاست را از گوگل درایو دریافت کرده و فولد ری تحت عنوان Dataset را ساختیم.

سپس برای هر یک از دیتاهای train, validation, test تابعی برای کار کردن بر روی دیتاست نوشتیم.

برای دیتاست train به این صورت میباشد که ابتدا تصویر را خوانده و سپس آنرا به فضای RGB میبرد و بر اساس input\_shape ی که به آن میدهیم تصویر را به آن اندازه ریسایز میکند.

برای دیتا ارزیابی و تست، به این صورت میباشد که در دیتاست اصلی به دلیل اینکه از مدل yolo استفاده کرده ایم و فایل های تکست لیبل شامل ۵ مقدار بود که مقدار اول شامل لیبل و ۴ مقدار باقی مانده شامل مختصات مرکز تصویر میباشد. لیبل 0 برابر زنده نبودن تصویر و لیبل ۱ برابر زنده بودن.

بعد از جمع آوری دیتا به سراغ به دست آوردن استخراج ویژگی از تصویر رفتیم. بدین صورت که برای هر تصویر ویژگی فرکانس را در دامنه فرکانس و ویژگی های مربوط به بافت را با استفاده از LBP بدست آوردیم.

بعد از بدست آوردن و مرتب کردن ویژگی ها، آنها را با هم ترکیب میکنیم.

برای ترکیب کردن این فیچرها با هم از شافل کردن دیتا استفاده کردیم. این کار بدین منظور انجام شد که ، مدل ابتدا فقط بر روی ویژگی های فرکانس تمرکز نکند و وزن هایی را بدست بیاورد سپس با همان وزن ها به سراغ ویژگی بافت برود و وزن ها تفاوت زیادی کنند. مشخصا این کار نتایج خوبی نداشته است.

در مرحله بعد به دلیل شافل کردن ، باید لیبیل ها را نیز duplicate کنیم و برای هر لیبیل بلافاصله یک کپی از همان مقدار داشته باشیم.

کلیات کد بدین صورت بود :

Cnn,Bigger CNN : برای آموزش و ارزیابی بر اساس استخراج فیچر از این دو مدل استفاده کردیم که مدل Bigger Cnn دارای معماری و تعداد لایه های بیشتری میباشد.

نتایج این دو نسبتا شبیه به هم میباشد و خروجی یکسانی تولید میکنند تنها تفاوت در این است که Bigger Cnn هنگامی که تصاویر اصلی و کراپ شده را دریافت میکند دارای عملکرد ضعیفی است و دقت ۳۸ درصد میدهد در صورتی که CNN با لایه های کمتر دارای دقت بهتری میباشد.

علت این مورد را اینگونه ارزیابی میکنیم که ، که شبکه کانولوشنی بزرگتر بیشتر به دیتا فیت شده است و انعطاف کمتری نسبت به شبکه کانولوشنی کوچکتر دارد در نتیجه دقت آن پایین تر میباشد. یعنی در اصل وزن های بیشتری نسبت به شبکه کوچکتر مقدار گرفتند و با تصویر کلی یا کراپ شده وزن ها دیرتر آپدیت میشوند و میتوانند مقادیر بهتری را پیشبینی کنند.

InceptionV3 : بعد از کلیات عملیات های بر روی کد چون دریافتی این شبکه باید دارای ۳ کانال باشد ورودی ها را به ۳ شبکه منتقل میکنیم و مقادیر را در ۳ کانال کپی میکنیم. علت اینکه از اول با ۳ کانال کار نکردیم این بود که در LBP, Fourier باید تصاویر را gray کرده و به محیط باینری ببریم. (به خصوص در LBP با توجه به اینکه local binary pattern میباشد. در فوریه اما میتوانیم برای هر کانال فوریه بدست آوریم اما با توجه به اینکه این دو دیتا باید ترکیب شوند ابعاد را یکسان گرفتیم).

نتایجی که در inception v3 گرفتیم نتایج نسبتا خوبی هم برای تصویر اصلی و کراپ شده بودند و هم تصویر با ویژگی ها خوب عمل کرده است. علت این مورد را اینطور تحلیل میکنیم که از وزن های imagenet استفاده کرده ایم و چون وزن ها از تصاویر اصلی بدست آمده اند در نتیجه وقتی از وزن های مدل جدید که بر اساس بافت است استفاده میکنیم خیلی دچار تغییر نشده است و تصاویر اصلی را نیز میتواند درک و پیش بینی خوب کند.

نکته مهم : در ارزیابی مدل های فوق فقط به ویدیو های گرفته شده اکتفا نکردیم بلکه از یک دیتاست بزرگتر برای تست استفاده کردیم که فریم به فریم ویدیو ها به عکس تبدیل شده و به مدل ها داده شده برای بدست آوردن مقادیر.

نکته مهم : با توجه به اینکه در فایل های متفاوتی کد مدل ها را زدیم برای هر کدام یک کد برای تولید یک فایل csv برای بررسی مدل و دقت آن قرار داده ایم.