



توضیحات تکمیلی هر قسمت در کد کامل وجود دارد و داکيومنت اصلی ما داخل نوت بوک قرار دارد.

قسمت اول: پیش پردازش

۱- پیش پردازش همیشگی ما که نرمال کردن داده ها بین ۰ و ۱ می باشد را انجام دادیم.
(بدون این پیش پردازش هم خروجی تقریباً مناسبی گرفتیم که کد آن موجود است برای بررسی دقیق تر باید متریک ها را بررسی کنید)

۲- در ابتدا داده ها را نمایش دادیم و متوجه غیر بالانس بودن داده ها در هر label شدیم و برای رفع آن تلاش های زیادی انجام دادیم

از **resample** کردن کلاس با داده ی کمتر و **downsample** کردن کلاس با داده ی بیشتر و **augmentation** دستی و استفاده از **rotation** های ۹۰ درجه، ۱۸۰ درجه، ۲۷۰ درجه و خود تصویر و در کل سعی کردم که تعداد داده های کلاس ها را برابر کنم و در نهایت حقیقت ماجرا اصلاً خروجی بهتری نسبت به مدلی که روی داده های **unbalance** کار می کرد دریافت نکردم.

قسمت دوم: معماری مدل

در این قسمت تعداد خیلی زیادی کد اجرا و تست نموده ام و در هر اجرایی کامل متریک های ارزیابی را پرینت نموده ام و به علت ساده بودن توضیحات زیادی نداده ام.
و همچنین از مدل غیر خطی هم کمک گرفتم

نکته عجیب که در کل در این قسمت دیدم

اجرای یک مدل یکسان بود که به دو روش پیاده کردم اول با **functional model** و دوم با **sequential model** و در یک خروجی عجیب، مدل **sequential** در نهایت **performance** بهتری داشت.

در صورت ارزیابی مدل با استفاده از **precision** , **recall** بهترین عملکرد مدل در ساده ترین حالت خود و با استفاده از

قسمت سوم: توابع فعالسازی به جز **relu**

کد و توضیحات کامل در نوت بوک موجود است.

قسمت چهارم: **stride + max pooling + average pooling**

کد و توضیحات کامل در نوت بوک موجود است.