**به نام خدا**

داکیومنت پروژه داده کاوی

استاد: حبیب رستمی

دانشجو: علیرضا خرم شکوه

980292619

در ابتدا برای دسترسی به دیتاست که در گوگل درایو قرار دارد باید به گوگل درایو متصل شویم.

سپس پیش پردازش انجام میدهیم. ابتدا در فایل اکسل، داده هایی که DM هستند را پاک میکنیم.

سپس داده ها را به داده های train, test و validation تقسیم میکنیم. برای این کار از لیست ها کمک میگیریم. 3 تا لیست داریم که مربوط به نام داده های test,train و validation است. سپس عکس های مربوط به آن نام ها را به آرایه numpy تبدیل میکنیم. در حقیقت عکس ها به صورت ماتریسی از اعداد هستند که در آرایه ذخیره شده اند. در همین حال resize انجام میدهیم.

حالا باید ببینیم هر عکس، چه لیبلی دارد. برای این کار، روی هر آرایه (که شامل عکس های test,train و validation است) یک حلقه for میزنیم و هر کدام از آیتم ها را در فایل اکسل بررسی میکنیم و مقدار ویژگی pathology classification آن را در یک لیست به نام label\_train ذخیره میکنیم. این کار را برای دو آرایه دیگر test و validation هم انجام میدهیم.

در نهایت داده های test , train و validation را داریم.

حالا باید آنها را نرمالایز کنیم. برای این کار لیست ها را بر 255 تقسیم میکنیم که داده ها بین 0 و 1 قرار بگیرند. همچنین تایپ آن ها را به float32 تبدیل میکنیم.

سپس one-hot-encoding انجام میدهیم. یعنی اینکه به جای لیبل ها، از اعداد استفاده میکنیم.

حالا مدل pre train را که در این پروژه EfficientNet V2 است را دانلود میکنیم.

سپس این مدل را در google drive ذخیره میکنیم تا برای دفعات بعد لازم نباشد دانلود کنیم.

حالا باید مدل خود را بسازیم. برای لایه اول از همان مدل pre train استفاده میکنیم و ویژگی trainable آن را False قرار میدهیم تا دوباره train نشود.

سپس از یک لایه flatten استفاده میکنیم و بعد از یک لایه dense و بعد از آن هم از یک لایه dropout استفاده میکنیم. به همین صورت چند لایه دیگر از این لایه ها را هم در ادامه قرار میدهیم.

این کار برای این است که از overfit شدن مدل جلوگیری کنیم. مثلا لایه dropout یک سری از node ها را غیرفعال میکند تا مدل نتواند آن ها را حفظ کند.

یک تابع earlyStopping هم تعریف میکنیم. این کار برای این است که روند train شدن را در نظر بگیریم و هر جا که تغییر مثبتی اتفاق نیوفتاد، مثلا val\_accuracy بهتر نشد، همان جا train را متوقف میکند.

حالا مدل را کامپایل میکنیم.

در مرحله بعد با دادن داده های train و validation به مدل، آ را train میکنیم.(با دستور model.fit)

در نهایت نمودار های val\_accuaracy, accuracy,loss و val\_loss را پلات (ترسیم) میکنیم.

در انتها با استفاده از دستور model.evaluate وبا دادن داده های test به آن، میزان دقت مدل روی دیتای تست را محاسبه میکنیم.