محمدحسین میرزایی ۹۹۵۲۲۱۵۸

امیرمحمد کمیجانی ۹۹۵۲۲۰۳۲

دیتاست:

برای این پروژه از دیتا ست های زیر استفاده شده است:

۱- https://www.kaggle.com/datasets/attentionlayer241/celeba-spoof-for-face-antispoofing/code
https://www.kaggle.com/code/shijunjie07/face-anti-spoofing-with-mobilenetv2

۲-

https://www.kaggle.com/datasets/minhnh2107/casiafasd

۳-

https://www.kaggle.com/datasets/faber24/lcc-fasd

https://www.kaggle.com/datasets/trainingdatapro/anti-spoofing-live

مورد اول مربوط به دیتاست بزرگ celeba می باشد که حدود ۸۰ گیگ بوده و دانلود و استفاده مستقیم از آن کار سختی بود برای همین از کد ضمیمه شده در ادامه و تغییراتی در آن استفاده کردم و بخشی از این دیتاست را دانلود کردم.

مورد دوم بخشی از دیتاست casiafasd هست که در دیتا ست های پیشنهاد شده نیز آمده بود و بیشتر تصویر های آن crop شده و از افرادی با صورت هایی شبیه به افراد آسیای شرقی می باشد.

مورد سوم نیز تصاویری از افرادی با چهره های غربی بود و به صورت crop شده قرار گرفته بود که از آن ها نیز در دیتاست استفاده کردم.

دقت شود که لیبل های مورد استفاده برای ما دو کلاس بندی live , spoof می باشد و هر کدام از این بخش ها را در قسمت مربوط به خود قرار دادم و همچنین با استفاده از کد هایی که در فایل dataCollection.ipynb و dataCol.ipynb زده شده است، صورت هر یک از این افراد را در شکل شناسایی کرده و به آن ها لیبلی با توجه به وضعیت liveness و همچنین باکسی با توجه به چهره داده می شود که در مدل train و در مرحله بعد قابل شناسایی باشد.

مورد چهارم نیز مربوط دسته ای از ویدیو های تشخیص liveness بوده که در بخش train استفاده نشده و صرفا، این ویدیو ها در بخش test برای میزان سنجش چهره استفاده شده است.

همچنین استفاده از دیتاست های متفاوت و لیبل زن صحیح آن ها که وقت بسیار زیادی از این پروژه را نیز به خود اختصاص داده برای دور شدن از بایاس داده ها می باشد که تصاویر صرفا بر اساس شناسایی داده های train از یک منطقه خاص جغرافیایی بایاس پیدا نکند.

در نهایت با استفاده از دیتاست های مطرح شده، یک فایل نهایی جمع آوری کرده و در Google drive خود ذخیره و برای استفاده در colab , Kaggle از آن استفاده می کنیم(فایل مربوط به این بخش نیز ضمیمه شده است.)

در بخش بعدی نیز دیتای کلی را به ۳ بخش train , eval , test تقسیم بندی می کنیم و مسیر آن ها و همچنین تعداد کلاس های مد نظر و هر کدام از کلاس های مد نظر را در فایل data.yaml برای یادگیری ثبت می کنیم.

یادگیری با شبکه عمیق:

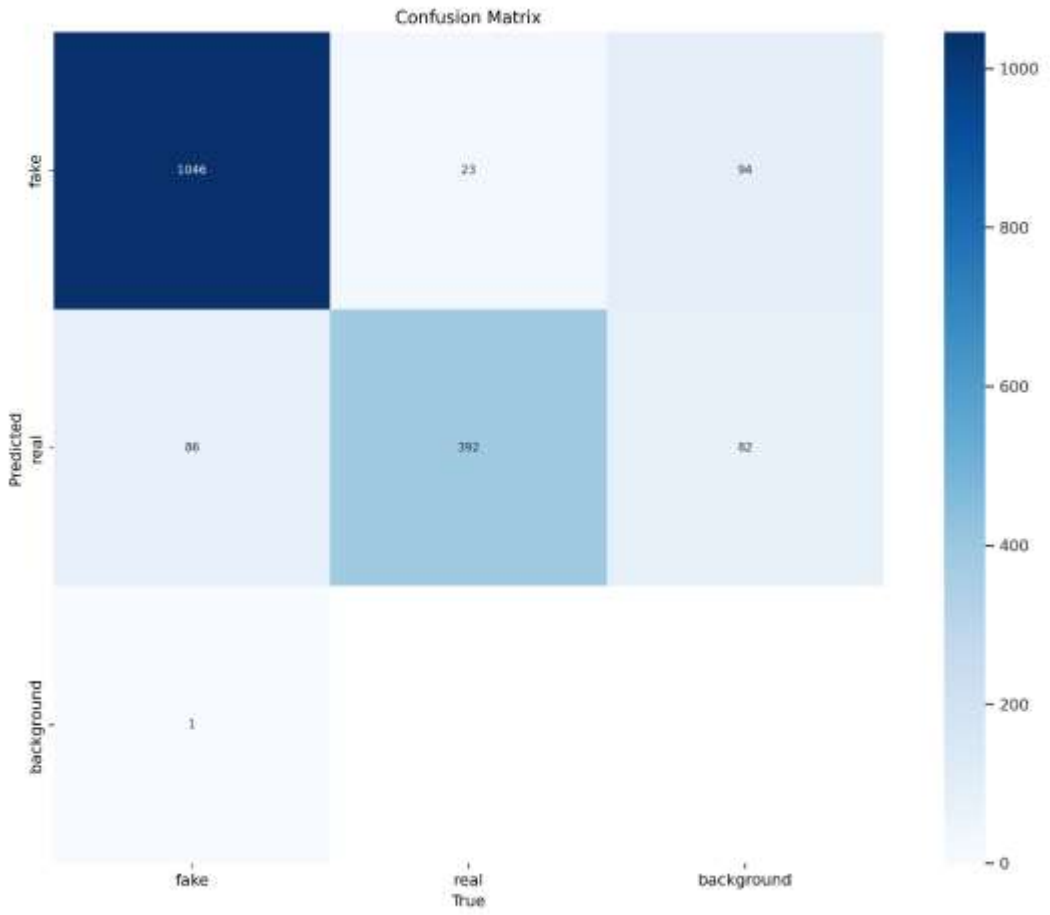
برای بخش train تنها کاری که انجام شده استفاده از YOLO می باشد و تنها تغییر ایجاد شده در آن استفاده از ۲ کلاس real , fake بوده و همچنین با توجه به داده های train ۲ بخش واقعی یا ساختگی بودن تصاویر که label خورده اند، مشخص شده و برای آن ها Loss حساب می کنیم و برای این کار epoch های با تعداد متفاوت در نظر می گیریم. حال در نهایت بهترین وزن های ممکن را که از بخش train بدست آمده استخراج می کنیم(این مرحله نیز در زمان اجرا زمان زیادی می برد و در سیستم محلی به محدودیت حافظه می انجامد). در بخش train نیز یک پارامتر patience وجود داشت که اگر به اندازه که برای آن گزارش شده داده بررسی می کردیم و بهبودی در epoch نبود به epoch بعدی برای اجرا می رفتیم.

در نهایت نیز در فایل main.ipynb وزن های بهینه بدست آمده از بخش train را به yolo می دهیم و با توجه به کلاس بندی جدید دو کلاسه از آن می خواهیم که تصاویر را دسته بندی کند.(این وزن های بهینه در فولدری به نام model ذخیره شده و برای هر چه بهتر شدن این محاسبات، دیتاست های مختلفی train شده و همچنین تعدادepoch ها نیز متفاوت گذاشته شده است که در یکی از آن ها دچار overfit شده ایم.)

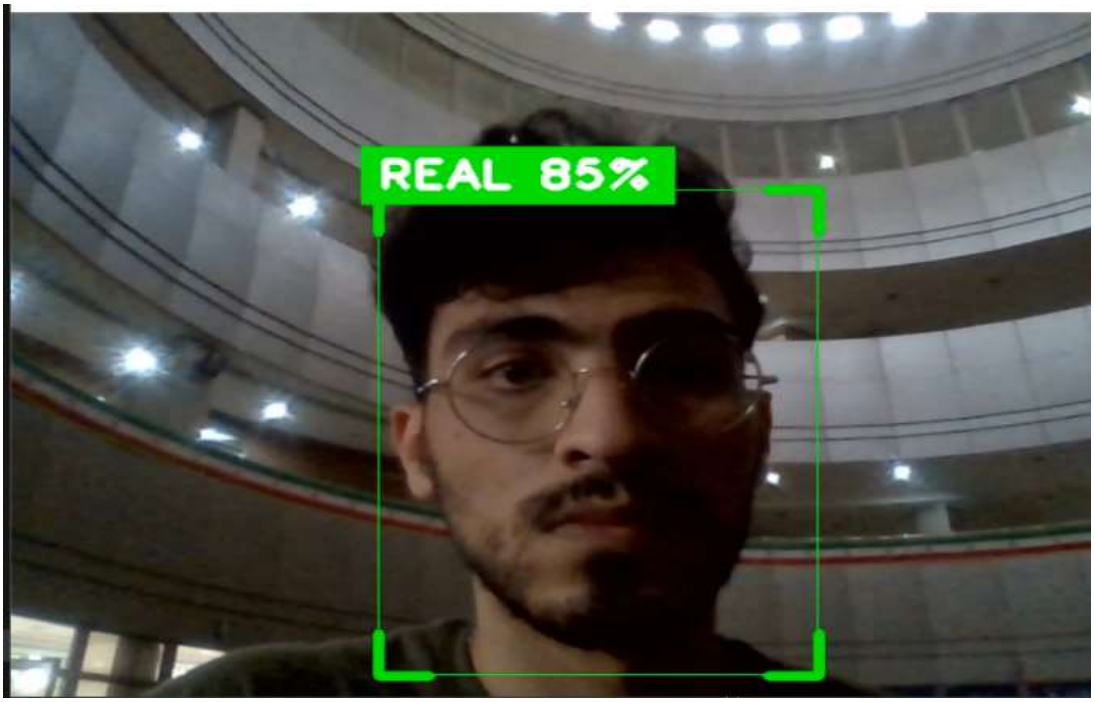
نمونه هایی از خروجی های بخش train:

که با توجه به یادگیری، در ابتدا box برای آن که در آن بخش از تصویر، چهره وجود دارد، در نظر گرفته شده و به آن احتمالی داده شده و همچنین برای آن box آن که تصویر live, fake هست مشخص شده است.

برای مدلی که با دیتای بزرگ تر نیز در نظر گرفته شده است، confusion matrix زیر در بهترین حالت گزارش شده است:



در نهایت نیز برای خروجی نهایی ویدیو ای از دانشجویان تهیه شد و چند تصویر از آن ها در گزارش نهایی به اشتراک گذاشته می شود:(چندین تصویر در فولدر نهایی ضمیمه شده است.)


REAL 85%


FAKE 79%