

گزارش پروژه درس هوش مصنوعی

پروژه میوه فروشی

استاد درس : دکتر اکبری

علیرضا شفقانی ۹۸۱۳۰۲۱

امیرمهدی ابوطالبی ۹۸۱۳۰۳۴

1.ObjectDetection

پروژه اول برنامه است ، این برنامه با استفاده از کتابخانه `pixellib` اشیاء درون یک شی را شناسایی میکند و میوه های آنها را جدا میکند .

برنامه اصلی و مدل اصلی است که با استفاده از الگوریتم `cnn` پیاده سازی شده است ، با کمک کتابخانه `tensorflow` و `keras` مدل `Cnn` ساخته شده و تمامی جزئیات کد و توضیحات در فایل ژوپیتر قابل مشاهده است.

ورودی برنامه : یک آدرس تصویر ، تصویر ذخیره شده در پوشه برنامه و آدرس آن

خروجی : برنامه یک تصویر به عنوان خروجی تحویل میدهد که روی آن تمامی اشیاء مشخص شده اند.

الگوریتم و توضیحات: ، این برنامه بسیار کوتاه است زیرا کتابخانه کاملی این کار را به خوبی انجام میدهد ، این کتابخانه از کتابخانه `opencv` استفاده میکند ، همچنین این برنامه از الگوریتم `mask_rcnn` برای تشخیص اشیاء کمک میگیرد و به کمک آن میتواند ۸۰ نوع شی را شناسایی کند.

مدل `mask_cnn_coco` برای این برنامه استفاده شده است.

2.FruitDetection

برنامه اصلی و مدل اصلی است که با استفاده از الگوریتم `cnn` پیاده سازی شده است ، با کمک کتابخانه `tensorflow` و `keras` مدل `Cnn` ساخته شده و تمامی جزئیات کد و توضیحات در فایل ژوپیتر قابل مشاهده است.

داده ها : از یک دیتاست حاوی ۱۳۱ نوع میوه ۳۶۰ درجه مختلف برای ساخت مدل استفاده شده است.

ورودی برنامه : برنامه برای تست داده ورودی میپذیرد و میتوان با آنرا با استفاده از مدل train شده تست کرد

File loading:

برای بارگذاری فایل ها از متود های tensorflow استفاده شده است ، فایل ها به صورت بخش بخش وارد برنامه میشوند و داده

نام گذاری لیبل داده ها براساس نام پوشه های آنهاست ، تا استفاده از متود glob مسیر تمامی تصاویر ذخیره میشود و سپس با توجه به shape داده ها و batch_size آنها در حافظه برنامه بارگذاری میشوند. توضیحات بیشتر در مورد لود کردن داده ها و بهینه سازی سرعت برنامه در منابع آورده شده است.

Image preprocessing :

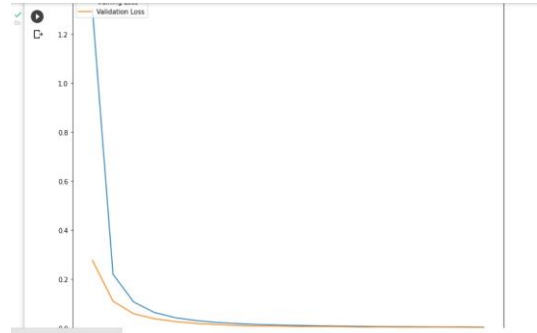
تمامی تصاویر در ابتدا به ماتریس هایی تبدیل میشوند که در هر درایه ۳ عدد در بازه ۰ تا ۲۵۵ وجود دارد ، اندازه ماتریس دقیقا اندازه تصویر است .

سپس تصاویر استاندارد میشوند و تمامی اعداد ذخیره شده بین ۰ تا ۱ قرار میگیرند ، این کار با یک تقسیم و چرخش روی کل داده صورت میگیرد .

Overfitting and accuracy :

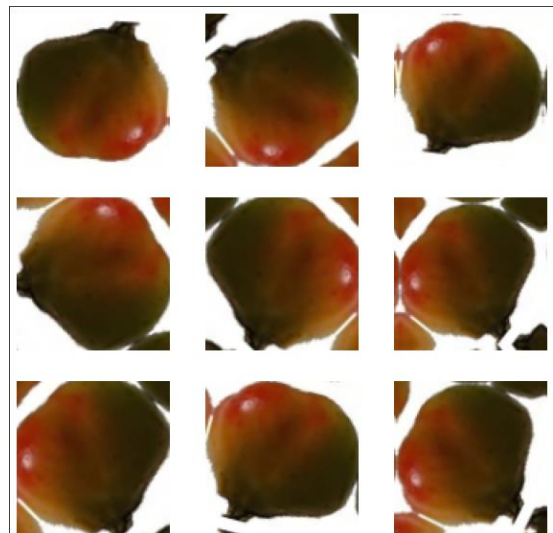
اگر با داده های خام ساختن مدل و train آن شروع شود ، ممکن است دقت کافی وجود نداشته باشد ، با نتایج تست شده قبلی و نمودار ها فهمیدیم که این دقت پایین است.

همچنین اگر feature های زیادی استخراج شود و بیش از اندازه از داده های train پی روی کنیم طبیعتاً overfit رخ میدهد: نتایج تست های قبلی :



برنامه برای جلوگیری از این مشکلات و افزایش دقت برای ساخت مدل ، از data agumentation و dropout استفاده کرده است.

از داده های موجود در دیتاست train دوباره داده های دیگری را تولید کرده ایم که هم به الگوریتم cnn کمک میکند و هم به بهتر شدن دقت مدل ما و بیشتر شدن دیتا ورودی :



که این عملیات شامل چرخش ، بزرگنمایی ، آینه کردن است.

همچنین برای بهتر کردن عملکرد مدل لایه های dropout و flatten اضافه شده اند ، توضیحات مربوط به این دولایه نیز در منابع به صورت کامل وجود دارد.

خروجی: برنامه یک مدل برای تشخیص نوع میوه (۱۳۱ نوع میوه مختلف) میسازد ، این مدل از الگوریتم cnn استفاده میکند و با ورود داده تست میتواند نوع label آنرا مشخص کند .

الگوریتم و توضیحات: توضیحات الگوریتم به صورت کامل در ویدیو و پاور گفته شده است ، پارامتر ها و تمامی ویژگی های دیگر توابع ، کتاب خانه های استفاده شده ، توضیحات کد و نتایج خروجی به صورت کامل در فایل موجود است.

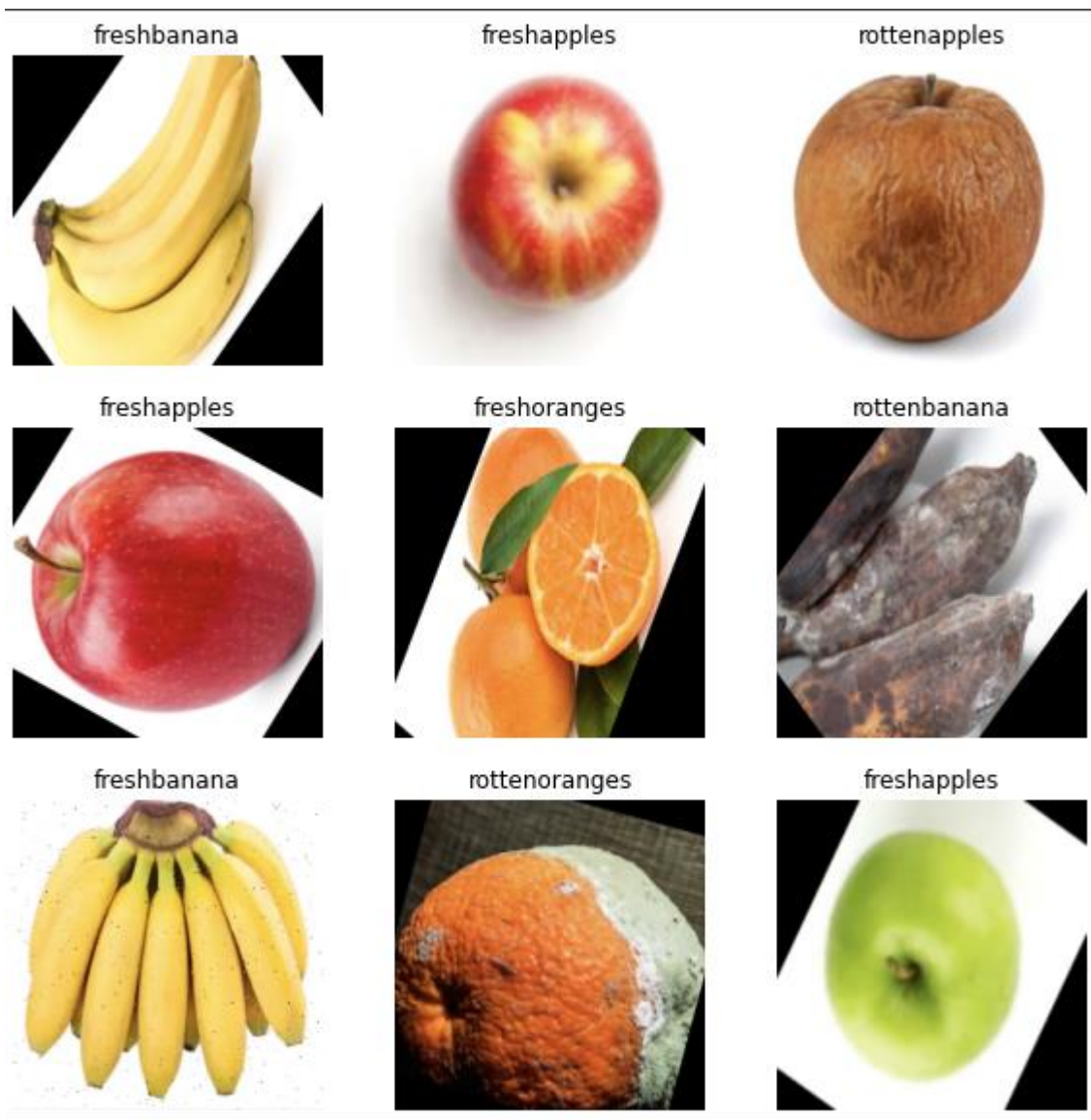
3. RottenOrFreshFruitDetection

برنامه سوم برای یکی از چالش های اصلی مسئله است که با استفاده از الگوریتم cnn پیاده سازی شده است ، با کمک کتابخانه tensorflow و keras مدل Cnn ساخته شده و تمامی جزئیات کد و توضیحات در فایل ژوپیتتر قابل مشاهده است.

داده ها: از یک دیتا ست [fresh and rotten fruit](#) برای ساختن مدل استفاده شده است .

ورودی برنامه: اندازه تصاویر ، برنامه برای تست داده ورودی میپذیرد و میتوان با آنرا با استفاده از مدل train شده تست کرد .

خروجی: لیبل داده های مشخص شده پس از طبقه بندی و بررسی داده تست ، به عنوان خروجی نمایش داده میشود .



خروجی برنامه ۶ حالت مختلف است ، آیا موز سالم یا فاسد است ، آیا سیب سالم یا فاسد است و آیا پرتقال سالم یا فاسد است.

الگوریتم و توضیحات: تمامی بخش های این برنامه مانند بخش های برنامه قبلی است ، با این تفاوت که برای ساخت مدل این برنامه ، خودمان از ابتدا مدل را نساختیم و از transfer learning استفاده کرده ایم ، با استفاده از ResNet50 که یک مدل برای پردازش تصویر است از کتابخانه keras توانسته ایم به دقت بسیار

خوبی برای داده هایمان دست یابیم ، توضیحات کامل transfer learning و متود های آن در منابع آورده شده است .

4.Client

یک برنامه کلاینت نهایی باید برای پروژه ساخته شود ، تا به عنوان ورودی آدرس تصویر را دریافت کند ، سپس پس از تشخیص اشیاء با اجرای برنامه اول روی آن تصاویر میوه ها را استخراج کند ، آنها را در مسیر مشخصی ذخیره میکند و سپس برنامه دوم برای تشخیص نوع میوه ها اجرا میشود ، مدل ساخته شده است و آماده test است ، با خواندن تصاویر در مسیر ذخیره شده آنها توسط مدل test میشوند و label آنها مشخص میشود ، سپس این اطلاعات به کاربر نمایش داده میشود ، که از هر میوه چه تعدادی وجود دارد و هر تصویر مربوط به چه میوه ای است ، البته برنامه با اجرای برنامه سوم ، میتواند سالم بودن و یا فاسد بودن میوه را به کاربر اطلاع دهد . مشکل اصلی دیتاست برنامه آخر است ، که تنها برای ۳ میوه وجود دارد ، برای تشخیص سالم بودن و یا خرابی میوه های دیگر ، به دیتاست دیگر با داده های بیشتر نیاز است .

۵.منابع

[pixellib](#)

[keras.transferclasificationmodules](#)

[imagingloading.keras](#)

[imageclassificationtensorflow](#)

[transfer learning](#)

[cnntensorflow](#)

[cnn](#)

[data augmentation](#)

با تشکر از استاد اکبری

