

هدف اصلی

همان طور که در مقاله آمده است هدف اصلی تحقیق محققان آن، ارائه یک سیستم خودکار و سریع است که COVID-19 را از عکس‌های رادیوگرافی اشعه ایکس قفسه سینه با استفاده از پردازش تصویر و الگوریتم‌های یادگیری ماشین شناسایی کند. در مقاله گفته شده است که از ۳۴۰ عکس رادیوگرافی که ۱۷۰ مورد سالم و ۱۷۰ مورد مبتلا هستند استفاده شده است. در دیتاست معرفی شده از سوی نویسندگان کل داده ها ۱۲۴۸ مورد است که ما نیز در مدل خود از کل دیتای معرفی شده استفاده کرده ایم.

خواندن دیتا

ابتدا تصاویر را از روی مسیر تعیین شده در فایل excel به صورت gray با استفاده از open cv فراخوانی می‌کنیم. در ادامه با تغییر سایز عکس ها، به ماتریسی که رنگ هر پیکسل از عکس ها به صورت مقداری بین ۰ و ۲۵۵ که نشانگر میزان رنگ بین سفید و مشکی است، می‌رسیم. در ادامه داده ی تست و ترین را از یکدیگر جدا کرده ایم. با توجه به مقادیر test size ها این روند صورت گرفته است.

گام اول: Extraction of feature descriptors

در این مرحله با توجه به این که استفاده از الگوریتم surf در نسخه های جدید open cv مقدور نبود به ناچار از الگوریتم sift استفاده کردیم. این الگوریتم تفاوت چندانی با surf نداشته. پس در نهایت از این الگوریتم به عنوان جایگزین استفاده کردیم.

The scale-invariant feature transform (SIFT) is a computer vision algorithm to detect, describe, and match local features in images.

همان طور که از مخفف این الگوریتم برمی‌آید، یک تکنیک قدرتمند برای تطبیق تصویر است که می‌تواند ویژگی‌هایی را در تصاویری که نسبت به مقیاس بندی، چرخش و اعوجاج پیوسته تغییر ندارند، شناسایی و مطابقت دهد.

ما از این الگوریتم استفاده کردیم و feature descriptor ها را استخراج کرده ایم. تعداد این فیچرها برای داده های ترین ۳۱۶,۵۵۵ مورد و برای داده های تست ۱۲۸,۷۷۰ مورد یافت شده است.

گام دوم: Clustering of feature descriptors

در این مرحله برای کاهش مقادیر فیچرها به مقدار قابل قبولی آن ها را در ۵۰۰ کلاستر دسته بندی می‌کنیم. FD_train و سپس FD_test را به K-means فیت کرده و سنترهای ۵۰۰ کلاستر را تحت عنوان visual-vocabulary (VV) در نظر می‌گیریم.

گام سوم: Classification of images

در این مرحله برای فیت کردن داده ها به الگوریتم SVM از فاصله اقلیدسی بین داده ها و سنتر کلاسترها استفاده کردیم و در مجموع برای ۵۰۰ سنتر آنها را با هم جمع کرده ایم.

برای predict کردن هم به طور مشابه از فاصله بین داده های تست و سنتر های کلاسترهای تست بهره برده ایم.

در ادامه به پیاده سازی مدلی برای CNN می پردازیم. هرچند هدف اصلی این تحقیق صرفا ساخت مدلی براساس image processing و الگوریتم های machine learning بوده است. در بخش پایانی برای مقایسه نتایج حاصل شده به ساخت مدلی بر پایه CNN پرداخته شده است، که ما نیز آن را در اینجا پیاده سازی کردیم. با توجه به این که به صورت جزئی در مقاله مراحل تشریح داده نشده در برخی از مدل پیاده سازی شده، ما از پیشفرض های خودمان استفاده کرده ایم.

همانطور که در مقاله نیز اشاره شده است در این قسمت داده ها به صورت ۳ کلاسه مورد بررسی قرار گرفته اند. و در انتها که مدل را تعریف کردیم، داده های ترین را به صورت نرمالیزه شده به مدل فیت میکنیم.

برای ساخت مدل CNN از کتابخانه keras استفاده شده است. در ابتدا که داده ترین به عنوان ورودی گرفته می شود تحت عنوان یک ماتریس با ابعاد مشخص بوده که با استفاده از لایه Convolution-2D فیچرها استخراج می شوند. در این لایه با استفاده ضرب داخلی کرنل تعریف شده در فیلتر و جمع درایه ها به ماتریس جدیدی میرسیم که به آن feature map گویند که شامل اطلاعات مفیدی درباره تصویر است. در لایه های استفاده شده Conv2D مدل، از activation: RELU استفاده کردیم. کار این تابع این است که خروجی لایه را از یک مقدار bias کم کند (۵۰۰) و در نهایت مقادیر منفی ماتریس حاصل را صفر کند.

در ادامه پس از هر لایه Conv، از یک لایه BatchNorm هم برای نرمالیزیشن، re-scaling و re-centering استفاده کرده ایم.

با استفاده از pooling سعی بر این است که ابعاد ماتریس بدست آمده از لایه قبلی را کاهش دهیم. به این شکل که بزرگترین مقدار بین درایه های ماتریسی که بین سطرها و ستون ها با stride مشخص حرکت می کند را به عنوان مقدار نهایی در ماتریس بعدی در نظر می گیرد.

Flatten ماتریس ورودی خود را تبدیل به یک vector میکند.

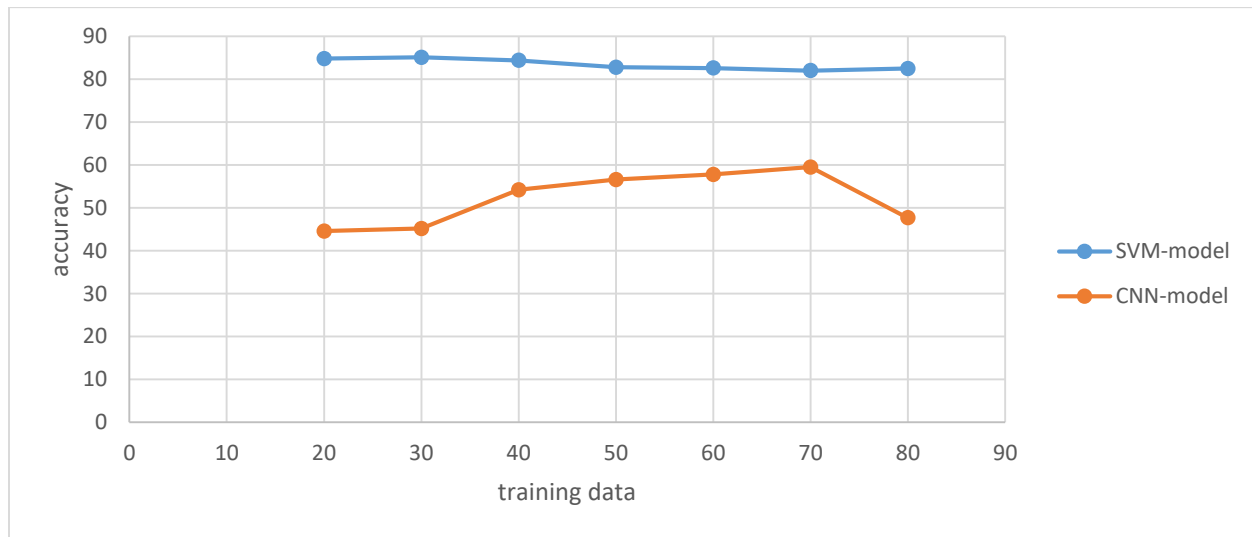
با استفاده از Dropout به میزان مشخص شده ای و رندم داده های ورودی هر لایه را صفر میکند. از مزیت های آن این است که از overfitting نخواهیم داشت.

در نهایت لایه Dense، خروجی را به صورت ۰-۱-۲ به می دهد.

و با استفاده از evaluate داده های تست را وارد کرده و در نهایت مقدار accuracy مدل را برای تست سائز های مختلف محاسبه می کنیم.

در ادامه به نتایج بدست آمده از دو مدل و مقایسه آن ها می پردازیم:

Training:Testing	SVM-model	CNN-model
20:80	84.8	44.6
30:70	85.1	45.2
40:60	84.4	54.2
50:50	82.8	56.6
60:40	82.6	57.8
70:30	82.0	59.5
80:20	82.5	47.7



نتیجه: همانطور که مشاهده میشود SVM-model به مراتب دقت بالاتری نسبت به CNN-model دارد.