

گزارش پروژه پایانی مبانی بینایی کامپیوتر

یگانه مرشدزاده، غزاله محمودی و نرگس سادات حسینی مونس

۱۵ بهمن ماه ۱۳۹۹

این گزارش شامل توضیحی کلی درمورد کد و مفاهیم و خروجی ها است.

قسمتی از کد ها در google colab نوشته و اجرا شدهاند که به صورت آنلاین با لینک زیر در دسترس است:

FCV Project Final Version

شایان ذکر است که در کد تنها کامنت ها و print های ضروری لحاظ شدهاند تا نتیجه نهایی بهتر مشاهده شود.

تقسیم داده آموزش و تست

برای آمادهسازی داده در اولین قدم داده را به دو قسمت تست و آموزش تقسیم می کنیم. برای این کار با توجه به نسبت داده ها اولیه که در اختیار ما قرار گرفته بود آنها را تقسیم کردیم (که به نسبت ۸۰ به ۲۰ تقسیم انجام شد).

دقت نهایی به دست آمده

نتایج خروجی و همچنین دقت کد برای هر کلاس و به طور کلی در فایل prediction.txt قرار داده شده است.

پلاک (چه سالم، چه مخدوش) است یا غیر پلاک؟

استخراج ویژگی

در مرحله اول سعی کردیم با انجام پردازش اولیه بر روی تصاویر داده شده،اطلاعات و ویژگیهای موثر در روند استخراج پلاک را به دست بیاوریم. از جمله پردازشهایی که در این پروژه بررسی شده است:

● پردازشهای مورفولوژی

- Opening
- Closing
- Erosion
- Dilation
- BlackHat

O TopHat / WhiteHat

- آستانهگذاری وفقی Adaptive thresholding
 - فیلتر sobel افقی و عمودی
- GaussianBlur برای کمکردن نویز در تصویر و همچنین هموار کردن تصویر

روند کد

برای دسته بندی این سه دسته داده،در ابتدا داده های پلاک را از غیر پلاک تشخیص دادیم.

- برای این کار ابتدا با استفاده از contour ها و تصویر آستانه گذاری شده و خاکستری شده، نواحی
 از کل تصویر که امکان دارد کاراکتر باشند را پیدا میکنیم.
- سیس برای هر کدام از این کاندیدها بررسی میشود که آیا این کاندید می تواند کاراکتر باشد یا نه؟
 - در صورتی که تشخیص داده شود که این کاراکتر است، یک لیستی از کاراکتریهایی که در همسایگی آن کاراکتر وجود دارد را ایجاد میکنیم.
- هدف این لیست این است که می توان با پیدا کردن همسایههای یک لیست به بدست آوردن
 پلاک های محتمل نزدیک شد. زیرا چند کاراکتر در کنار یکدیگر میتوانند تشکیل یک پلاک بدهند.
- برای سنجش همسایگی از مواردی همچون فاصله، زاویه، میزان تفاوت در مساحت و میزان
 تفاوت در طول و عرض میان دو کاراکتر کاندید استفاده شدهاست. به عبارتی در این قسمت
 مناطق با تراکم بالای کاراکترها به عنوان نقاط برگزیده هستند.
 - حال با داشتن این مناطق که کاندید های پلاک هستند باید سعی کنیم که پلاک را استخراج کنیم:
- پس از به دست آوردن دستههای همسایگی شامل کاراکترها، که نشان دهنده پلاکهای کاندید است، برای هر دسته، کاراکترهای آن دسته را به صورتی مرتب میکنیم که چپترین کاراکتر در اولین ایندکس لیست و راستترین کاراکتر در آخرین مکان این لیست کاندید پلاک قرار بگیرد (مرتب کردن بر اساس مولفه ایکس به صورت صعودی)
- سپس ویژگیهای این لیست، عرض پلاک (تفاوت x اولین کاراکتر و آخرین کاراکتر)،
 میانگین ارتفاع پلاک بر اساس ارتفاع کاراکترها و همچنین اختلاف ارتفاع اولین کاراکتر و

آخرین کاراکتر (برای به دست آوردن پارامترها برای تبدیل affine تصویر کاندید پلاک) را بدست می آوریم.

- از ویژگی های بدست آمده استفاده میکنیم تا برای هر کاندید پلاک یک مجموعه اطلاعات کامل داشته باشیم و در صورت انتخاب این کاندیدا به عنوان پلاک میتوان تبدیلات و تغییرات لازم برای crop کردن تصویر اولیه و ایجاد یک تصویر دقیقتر از پلاک را به طور دقیق پیادهسازی کرد.
- پس از به دستآوردن پلاک های محتمل، اگر این لیست خالی بود به معنای عدم تشخیص پلاکی در عکس است (کلاس ۲ => اصلا پلاک نیست!) و در غیر اینصورت در تصویر پلاک تشخیص داده شده است. (کلاس ۰ یا ۱ => پلاکی که چه مخدوش چه سالم است!)

حالا که پلاک است، سالم است یا مخدوش؟

چالش اصلی در تشخیص پلاک مخدوش از سالم/غیر مخدوش بود.

دادهها

به این منظور در تصاویر، قسمت پلاک های تشخیص داده شده را (اعم از مخدوش و غیر مخدوش) crop کردیم تا به عنوان ورودی به شبکه عصبی بدهیم.

با بررسی های انجام شده متوجه شدیم علاوه بر اینکه داده ها unbalanced هستند، تعداد دیتا های پلاک مخدوش بسیار کم است. به همین علت از روی پلاک های سالم، پلاک مخدوش تولید کردیم تا برای آموزش به شبکه عصبی بدهیم.

شبکه عصبی

در این میان شبکه های عصبی مختلفی مانند Resnet و mobile net را با در نظر گرفتن وزن های از پیش train و imagenet و trainable امتحان کردیم. در لایه های انتهایی این شبکه classifier خودمان را قرار می دادیم تا به یادگیری بپردازد.

روند پیشرفت کار

چالش اصلی عدم یادگیری صحیح شبکه و حفظ کردن داده های آموزشی بود.

- از جمله مشكلات اولیهای که با آن روبرو شدیم این بود که دادههای داده شده بالانس نبودند و این
 موضوع باعث ایجاد مشکل در آموزش شبکه شد.
- برای رفع این مشکل ابتدا سعی کردیم از دادن وزن به هر کلاس داده ها استفاده کنیم به این
 ترتیب که کلاس اقلیت دارای وزن بیشتر و کلاس اکثریت دارای وزن کمتر است.
 - اما تاثیر مثبتی در راستای یادگیری نداشت.
 - در مرحله بعد سعى كرديم با افزايش ديتا اين دادهها را بالانس كنيم.
- در مرحله اول برای افزایش داده از تصاویر پلاکهای سالم استفاده کرده و با استفاده از کتابخانههای موسمت مانند دایره و مربع با رنگهای و اندازههای متفاوت (به صورت تصادفی) تصاویر مخدوش را تولید کردیم.
 - در مرحله دوم از data generator برای تولید دادهاضافه استفاده کردیم که تنها تغییراتی که برای این نوع داده ها (پلاک) قابل اعمال بودند،
 - 1. تغییر رنگ
 - 2. تغییر زاویه حداکثر تا ۲۰ درجه بودند.
 - با توجه به اینکه اضافه کردن داده کمکی به بهبود آموزش شبکه نکرد سعی کردیم از
 تکنیکهایی همچون:
 - dropout ■
 - تغییر ایتیمایز (Adam , SGD)
 - تغییر نرخ یادگیری
 - تغییر تابع فعال سازی لایه آخر به sigmoid , tanh , softmax
 - تغییر classifier به Dense با لایه های بیشتر

برای حل مشکل استفاده کنیم که متاسفانه بی فایده بود و در بهترین حالت شبکه تنها توانایی حفظ کردن داده و یعنی overfitting رخ میداد.

○ به علت وجود مشكلات گفته شده در اين مرحله زمان زيادي صرف كرديم و:

- شبکه های مختلف
- ورودی های مختلف

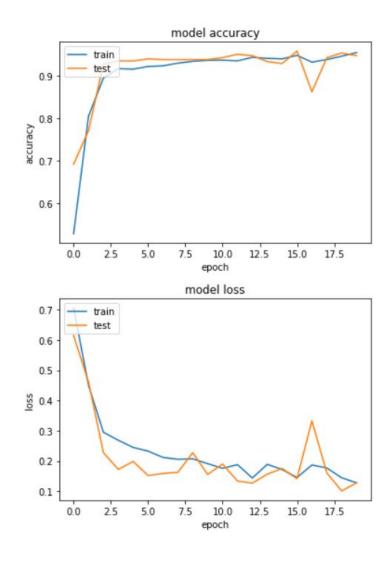
را تست کردیم اما یادگیری مناسبی در شبکه حاصل نشد.

- در classifier از شبکه های عصبی از CNN هم استفاده کردیم که روی mnist دقت 90
 می داد ولی مناسب این مسئله نبود.
 - یکی دیگر از روشهای دیگری که برای تشخیص پلاک مخدوش از سالم امتحان کردیم
 استفاده از OCR بود.
- برای این کار ابتدا سعی شد سورس کدهای OCR برای زبان فارسی پیدا شود که کد مناسب پیدا نشد.
- در مرحله بعد سعی کردیم از Tesseract برای تشخیص حروف در تصویر استفاده شود که آنهم نتیجه مطلوب نداشت.
 - روش دیگری که برای تشخیص پلاک مخدوش از غیر مخدوش تست کردیم این بود که:
 - پلاک کراپ شده را باینری کنیم.
 - سپس connected component ها را پیدا کردیم.
 - با بررسي تعداد آن ها، به مخدوش يا سالم بودن پلاک پي ببريم.
 - در این مرحله از open و close و .. برای پیدا کردن بهتر open در این مرحله از component استفاده کردیم.

كه با وجود صرف وقت زياد نتيجه خوبي حاصل نشد. :)!

- روش دیگری که برای تشخیص پلاک مخدوش استفاده کردیم این بود که اگر در تصویر crop شده پلاک، یک شکل هندسی مانند دایره، مستطیل و یا مثلث وجود داشت، این تصویر یک یلاک مخدوش می باشد.
- علت استفاده از این روش این بود که با یک نگاه کلی به دیتاست متوجه شدیم که در تعدادی از پلاکهای مخدوش تصویر به گونه ای میباشد که پس از پردازشهای

اولیه بر روی آن یک شکل هندسی در تصویر به وجود می آید و آن حالت پلاک بودن و وجود ۵ رقم کنارهم را به هم می ریزد. در واقع تعداد حروف از پلاک اصلی و سالم است.



روش های دیگری که برای حل پروژه،امتحان شده است

YOLO

در ابتدا سعى بر پيدا كردن پروپوزال توسط شبكه هايي همچون YOLO داشتيم.

- با توجه به اینکه bounding box به عنوان ورودی نبود و باید برای شبکه با توجه به نوع داده ها مشخص می شد (که در عمل برای این حجم از داده ممکن نبود)، منصرف شدیم.
- همچنین به چالش های همچون استفاده از فریم ورک darknet برخوردیم که بسیار اذیت کننده بود.

3-class classification problem with neural network

کار دیگری که در ابتدا مسیر امتحان کردیم کلاسه بندی کل سه دسته به کمک شبکه های pretrain کار دیگری که در ابتدا مسیر امتحان کردیم کلاسه بندی کل سه دسته به کمک شبکه های resnet همچون resnet بود که با توجه به نوع مسئله میسر نیست.

- Automatic License Plate Detection & Recognition using deep learning
- femioladeji/License-Plate-Recognition-Nigerian-vehicles: A python program that uses the concept of OCR using machine learning to identify the characters on a Nigerian license platefemioladeji/License-Plate-Recognition-Nigerian-vehicles: A python program that uses the concept of OCR using machine learning to identify the characters on a Nigerian license plate
 - تشخیص پلاک فارسی خودرو بینایی کامپیوتر
- https://github.com/fatemeakbari/machine-license-recognition
- amirmgh1375/iranian-license-plate-recognition: iranian car license plate recognition with yolo object detection and cnn lstm ctc loss
- pouryash/openCV-plate-recogenation-python
 - و توسعه سیستم تشخیص پلاک خودرو با استفاده از فناوری یادگیری ماشین در پایتون مجله شهاب
- CNN Application-Detecting Car Exterior Damage(full implementable code)
- Mask R-CNN | Building Mask R-CNN For Car Damage Detection
- PyImageSearch Detecting Natural Disasters with Keras and Deep Learning
- MicrocontrollersAndMore/OpenCV 3 KNN Character Recognition Python
- Segmentation in OCR !!. A basic explanation of different levels... | by Susmith Reddy
- Segmenting Words and Characters
- mammadjv/vehicle-registration-plate-recognition: Implementation of Vehicle Registration

 Plate Recognition using CNNs, my bachelor thesis
- MicrocontrollersAndMore/OpenCV 3 License Plate Recognition Python
- anuj200199/licenseplatedetection: License plate Object Detection through YOLOv3 and Recognition through pytesseract
- resnet keras