



هدف پروژه

هدف این پروژه، بهبود شناسایی اعداد دست‌نویس با استفاده از الگوریتم‌های طبقه‌بندی مانند درخت تصمیم (Decision Tree) و SVM است. برای دستیابی به این هدف، ابتدا ویژگی‌های تصاویر با به‌کارگیری فیلترهای مختلف استخراج می‌شوند و سپس به کمک کاهش ابعاد و تنظیم بهینه‌ی هایپرپارامترها، مدل‌ها آموزش داده می‌شوند. ارزیابی و مقایسه دقت مدل‌ها در مراحل مختلف پردازش داده انجام می‌گیرد تا بتوان تاثیر هر فاز را بر عملکرد نهایی بررسی کرد و با استفاده از تکنیک‌هایی مانند pruning، از overfitting جلوگیری نمود.

دیتاست MNIST (Modified National Institute of Standards and Technology)

دیتاست استفاده‌شده در این پروژه، دیتاست MNIST است که شامل ۷۰۰۰۰ تصویر سیاه و سفید از ارقام ۰ تا ۹ است که به‌صورت متوازن در ۱۰ کلاس مختلف توزیع شده‌اند، به‌طوری که هر کلاس نماینده‌ی یکی از ارقام است. هر تصویر دارای ابعاد ۲۸ در ۲۸ پیکسل است و هر پیکسل مقداری بین ۰ تا ۲۵۵ دارد که شدت رنگ آن را نشان می‌دهد. این داده‌ها می‌توانند به دو صورت مورد استفاده قرار گیرند: به‌صورت ماتریسی ۲۸ در ۲۸ که مقادیر پیکسل‌ها را حفظ می‌کند، یا به‌صورت آرایه‌ای یک‌بعدی با ۷۸۴ مقدار که تمام مقادیر پیکسل‌ها به‌ترتیب در آن ذخیره شده است.

فاز اول: Image Filtering

در این فاز از شما خواسته شده تا با توجه به توضیحات داده شده در کلاس حل تمرین، فرآیند کانولوشن یک فیلتر (کرنل) بر روی تصویر را پیاده‌سازی کنید. برای این منظور، تابعی طراحی کنید که دو پارامتر تصویر و کرنل را دریافت کند و عملیات کانولوشن را اجرا کند. سپس، با استفاده از این تابع، فیلتر سوبل (Sobel) را بر روی تصاویر مجموعه داده اعمال کنید. توجه داشته باشید فیلتر سوبل شامل دو کرنل است و باید هر دو این فیلترها را به‌طور جداگانه بر روی تصویر اعمال کنید. [\(لینک کمکی\)](#)

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$

پس از اعمال این فیلتر، با اعمال تابع آماده ای به نام **hog** بر روی تصاویر اصلی (تصاویر اولیه که فیلتر سوبل بر روی آن‌ها اعمال نشده است)، فیچرهای جدیدی از تصویر استخراج کنید و در کنار فیچرهایی که با کمک سوبل استخراج کردید؛ قرار دهید و برای هر تصویر یک بردار ویژگی بسازید.

- خروجی تصاویر بعد از اعمال Sobel و همچنین بعد از اعمال Hog را visualize کنید و تاثیر هرکدام را بررسی کنید. (برای hog فقط visualize کردن کافی است و نیازی به تحلیل نیست و یک بار hog اعمال شود و یکبار نشود).

- پس از انجام تمام فازهای پروژه، یک بار به جای فیلتر سوبل، فیلتر دیگری به دلخواه پیدا کنید و بر روی تصاویر اعمال کنید و مجدداً تمام مراحل این فاز را انجام دهید (یک بار hog اعمال شود و یک بار نشود).

- همچنین یک بار این فاز را کلاً انجام ندهید تا تاثیر تصاویر خام را متوجه شوید.

- بررسی کنید که هر کدام از فیلترها چه کاری انجام می‌دهند و چه تاثیری بر روی طبقه بندی نهایی شما دارند و چگونه می‌توان آن‌ها را بهبود داد.

فاز دوم: Image Centering and PCA

در این مرحله ابتدا میانگین تمام تصاویر دیتاست را محاسبه و آن‌ها را visualize کنید. سپس تصویر میانگین بدست آمده را از تمامی تصاویر دیتاست کسر کنید تا تصاویر حول میانگین کل متمرکز شوند. تعدادی از این تصاویر centered (تصاویری که حول میانگین کل متمرکز شده‌اند) را نیز نمایش دهید تا اثر centering را مشاهده کنید.

سپس با اعمال **PCA** بر روی این دیتای جدید centered، ابعاد دیتا را کاهش دهید. از شما می‌خواهیم از **plot scree** استفاده کرده و مقدار مناسبی برای **n_components** را بدست آورید. الگوریتم‌های طبقه بندی را در مرحله بعدی با دیتای بدست آمده (پس از اعمال pca) آموزش دهید و دقت مدل را بررسی کنید.

همچنین دیتاست را به دو بخش آموزشی و آزمایشی (train و test) تقسیم کنید، به طوری که ۸۰ درصد داده‌ها برای train و ۲۰ درصد برای test باشد.

فاز سوم: Decision Tree Hyperparameter Tuning

هدف این فاز بهینه‌سازی درخت تصمیم است، به طوری که با تنظیم مقادیر مناسب برای هایپرپارامترهای مختلف از مشکلات **overfitting** و **underfitting** جلوگیری شود و بهترین مقادیر برای پارامترها بدست بیاید. هایپرپارامترهای **حداکثر عمق درخت**، **حداقل تعداد نمونه‌ها برای هر تقسیم**، **حداقل تعداد نمونه‌ها برای هر برگ و تابع تقسیم** را به ازای مقادیر مختلف تست کنید تا بهترین مقدار آن‌ها را بیابید تا از ایجاد گره‌های بیش از حد کوچک یا تقسیمات غیرضروری جلوگیری شود. برای این منظور از روش **grid search** استفاده کنید. (لینک کمکی)

برای اینکه مدل به داده‌های جدید بهتر تعمیم دهد و از خطر **overfitting** جلوگیری شود، لازم است که مدل به طور مؤثر ارزیابی شود. یکی از روش‌های مؤثر برای ارزیابی تعمیم‌پذیری (**generalization**)، **K-fold Cross-Validation** است (در لینک کمکی قبلی این روش نیز توضیح داده شده است).

- برای این فاز نیاز است که علاوه بر grid search، روش K-fold Cross-Validation با $K=5$ را نیز انجام دهید. می‌توانید در هنگام فراخوانی grid search، پارامتر مناسبی را ست کنید تا cross validation در هنگام جستجو برای پارامترهای بهینه توسط تابع grid search انجام شود و نیاز پیاده سازی آن نیست.

- در نهایت، مدل انتخاب شده به عنوان بهترین مدل را برای استفاده در مراحل بعدی پروژه مشخص کنید.

یک مدل طبقه‌بندی SVM نیز با دو نوع دیتای زیر آموزش دهید و دقت آنرا با دقت مدل درخت نهایی این بخش مقایسه کنید. توجه داشته باشید که مدل‌ها باید بر روی دیتای train فیت شوند و بر روی دیتای test ارزیابی شوند.

- HOG + sobel filter

- raw data

فاز چهارم: تحلیل دقت مدل و معیارهای سنجش

این فاز به تحلیل دقیق عملکرد مدل اختصاص دارد. برای هر کلاس، معیارهای Precision، Recall و F1-Score را محاسبه کنید و در فایل مستندات خود ثبت کنید تا عملکرد مدل در شناسایی نمونه‌های هر کلاس ارزیابی شود. با ایجاد confusion matrix و visualize کردن آن به صورت Heatmap، می‌توان مشاهده کرد کدام کلاس‌ها بیشتر با یکدیگر اشتباه گرفته شده‌اند. ([لینک کمکی](#))

فاز پنجم: Overfitting and Pruning

برای کاهش overfitting، تکنیک pre-pruning روی درخت تصمیم اعمال می‌شود. در این فاز، ابتدا مدل را با تنظیمات خاصی به گونه‌ای طراحی کنید که روی دیتا overfit شود (با تنظیم مقدار هایپرپارامترها). سپس با استفاده از تکنیک pre-pruning (محدود کردن هایپرپارامترهای عمق درخت و تعداد نمونه‌ها برای هر تقسیم) مدل را بسازید و نشان دهید که با گذاشتن این محدودیت overfitting مدل کاهش می‌یابد.

نکات و توضیحات تکمیلی

- فاز دوم را به ازای این دو حالت دیتاست انجام دهید: `sobel + HOG` و `raw data`
- فاز سوم را با تمام حالات دیتا، با فیلتر `sobel`، با فیلتر `sobel + HOG`، با فیلتر `HOG + HOG`، و دیتای خام، انجام دهید و دقت مدل را به ازای آن‌ها به دست آورید و مقایسه کنید.
- فاز چهارم و پنجم را با مدلی انجام دهید که با دیتایی که فیلتر `sobel` و `HOG` روی آن اعمال شده آموزش دیده است. (مدل حاصل از `grid search`)
- انجام پروژه می‌تواند در قالب گروه‌های دو نفره و یا به صورت انفرادی صورت گیرد.
- علاوه بر سورس کد پروژه، فایل **مستندات** نیز باید آپلود شود.
- نام اعضای گروه در فایل مستندات ذکر شود و فقط یکی از اعضا پروژه را آپلود کند.
- هر گونه شباهت نامتعارف بین کد شما و کد سایر گروه‌ها تقلب محسوب می‌شود و نمره‌ای برای این پروژه دریافت نخواهید کرد.
- در صورت نوشتن داکيومنت تمیز (برای مثال با `LATEX`) نمره اضافه برای شما در نظر گرفته خواهد شد.
- فایل شامل سورس کد پروژه و مستندات را در قالب فایل `zip` و با نام شماره دانشجویی خود ذخیره و ارسال نمایید.
- در صورت داشتن هرگونه سوال می‌توانید با [SoroushPasandideh](#) و [Aalireza_s](#) و [Kourosh_Hsz](#) در ارتباط باشید و یا در گروه درسی مطرح نمایید.

موفق باشید؛ تیم حل تمرین