

راهنمای فوری

برای اجرای کدهای Mine Problem

به منظور سادگی در گرفتن اجراها، تمام فایل‌های وابسته در فولدرهای ارسالی قرار گرفته‌اند. فقط باید از ورژن‌های جدید نرم افزار MATLAB (ترجیحاً ۲۰۲۱) استفاده شود و اطمینان حاصل کنید که کتابخانه‌های Deep Learning در آن نصب شده باشد. در فایل ارسالی دو پوشه موجود است:

- **Final Codes and Models**: در این پوشه هر آنچه برای به دست آوردن خروجی نیاز است وجود دارد. سه شبکه عصبی پس از آموزش به صورت فایل‌های `mat`. در این فولدر کپی شده‌اند تا کدها بتوانند به صورت مستقیم از آنها استفاده کنند. کد اصلی که خروجی نهایی را تولید میکند فایل `finalMain.m` است. در این کد، تصاویر موجود در فولدر `test-rqd` یکی یکی خوانده شده و به ازای هر `run` در تصاویر مقدار `rqd` محاسبه شده و نمایش داده می‌شود. سپس یک فایل به اسم `output.xlsx` ایجاد می‌شود که خروجی‌های نهایی در آن وجود دارند. همچنین یک کد دیگر برای تست تصاویر به صورت تکی نوشته شده است که برای آن اجرای آن می‌توانید از فایل `singleTest.m` استفاده نمایید. کافی است به متغیر `str` آدرس تصویر دلخواه را بدهید. این کد `run`‌های موجود در تصویر را جداسازی کرده و به ازای هر `run` مقدار `rqd` را محاسبه کرده و نمایش می‌دهد. مابقی فایل‌ها توابع وابسته‌ای هستند که در کدهای اصلی مورد استفاده قرار گرفته‌اند و به صورت تکی قابل اجرا نمی‌باشند.
- **Supporting Codes**: کدهای اولیه که برای آموزش دادن مدل‌های هوش مصنوعی استفاده شده‌اند در این فولدر قرار گرفته‌اند. در این فولدر سه زیر فولدر به شرح زیر وجود دارد:

- **Training Codes for Wood Detection**: همانطور که از نامش پیداست، کدهایی است که برای یافتن تکه چوب‌ها استفاده می‌شود. تعدادی تصویر در فولدر `dataset` وجود دارد که ورودی‌ها و خروجی‌های مدل شبکه عصبی عمیق را فراهم می‌کنند. دقت شود که این دیتاها به صورت دستی ساخته شده‌اند و از دیتاهای آموزشی که در مسابقه وجود داشته، به شکل دیگری استفاده شده است (توضیحات بیشتر در فیلم آموزشی قرار می‌گیرد). یک کد به نام `aug.m` وجود دارد که با استفاده از تکنیک `data augmentation` دیتاهای آموزش را افزایش داده و در فولدر `Data` قرار می‌دهد. این کار قبلاً انجام شده و فولدر `Data` پر است. از این رو اجرای `aug` ضروری نیست. برای آموزش دادن مدل باید از کد `semantic.m` استفاده شود. با اجرای این کد، یک شبکه عصبی به کمک دیتاهای موجود در فولدر `Data` آموزش داده می‌شود و پس از اجرای کامل، یک فایل به نام `WoodNet.mat` ایجاد می‌شود. این فایل همان فایلی است که در کدهای نهایی برای جداسازی تکه چوب‌های بنفش استفاده می‌شود.
- **Training Codes for Rock Detection**: ای فولدر نیز دقیقاً مشابه با فولدر قبلی است. با این تفاوت که به جای تکه چوب‌های بنفش، سنگ‌های بزرگتر از ۱۰ سانتیمتر را جداسازی می‌کند. دقت شود که برای هر دو مدل از معماری `DeepLab v3+` استفاده شده که یک شبکه عصبی `Encoder-Decoder` برای استفاده در مسائل `Semantic Segmentation` است. کدهای این فولدر در نهایت یک مدل به نام `RockNet.mat` ایجاد می‌کنند که در الگوریتم نهایی برای جداسازی سنگ‌های بزرگتر از ۱۰ سانتیمتر استفاده می‌شود.
- **Training Codes for Empty Detection**: این فولدر نیز مشابه با دو فولدر قبلی است. اما اینجا به جای `segmentation` با یک مسئله‌ی `classification` طرف هستیم. در این فولدر مدلی آموزش داده می‌شود که بتواند

تفاوت سطرهای خالی و پر را از هم تشخیص دهد. چرا که در برخی جعبه‌ها، وقتی که گمانه به پایان می‌رسد بخشی از جعبه خالی است و الگوریتم باید بتواند این را تشخیص دهد به طوری که آن قسمت خالی را معادل با یک run در نظر نگیرد. برای این منظور از یک شبکه googlenet استفاده شده است که با استفاده از تکنیک Transfer Learning و دیتاهایی که به صورت دستی جمع آوری و مبسوط شده آموزش داده می‌شود. این کدها نیز پس از اجرا یک شبکه در قالب یک فایل به نام BoxNet.mat ایجاد می‌کند.

با تشکر

مجید فرزانه

Majid.farzaneh91@gmail.com