

مقدمه

رشد روزافزون داده‌های تصویری و متنی در فضای دیجیتال، نیاز به روش‌های خودکار برای استخراج و پردازش اطلاعات را بیش از پیش آشکار کرده است. یکی از حوزه‌های مهم در این زمینه، **تشخیص متن در تصاویر (Optical Character Recognition - OCR)** است که در کاربردهایی مانند دیجیتال‌سازی اسناد، خواندن تابلوها، و پردازش محتوای بصری به کار می‌رود.

یکی از چالش‌های خاص این حوزه، استخراج متن از **کتاب‌های کمیک** است. در کمیک‌ها، متن معمولاً درون حباب‌های گفتاری یا فکری قرار می‌گیرد و سبک طراحی، نوع فونت و حتی کیفیت چاپ می‌تواند کار شناسایی متن را دشوار کند. روش‌های سنتی OCR مانند Tesseract به تنهایی در این شرایط دقت کافی ندارند و نیاز به ترکیب آن‌ها با تکنیک‌های نوین پردازش تصویر و یادگیری عمیق وجود دارد.

انگیزه‌ی شخصی من برای این پروژه از تجربه‌ی کاری‌ام در زمینه‌ی ترجمه‌ی کمیک‌ها شکل گرفت. در آن دوران، بخش زیادی از زمان صرف پیدا کردن و بازنویسی متن داخل تصاویر می‌شد. داشتن یک ابزار خودکار برای شناسایی حباب‌های گفتاری و استخراج متن، می‌توانست فرایند ترجمه را بسیار سریع‌تر و دقیق‌تر کند. همین موضوع ایده‌ی پروژه‌ی حاضر را شکل داد.

در این پروژه، با استفاده از **شبکه‌ی YOLO** برای شناسایی حباب‌های گفتاری و **EasyOCR** برای استخراج متن، سیستمی طراحی شده است که قادر است متن موجود در کمیک‌ها را شناسایی، استخراج و اصلاح کند. این کار علاوه بر تسهیل فرایند ترجمه و ویرایش کمیک‌ها، نمونه‌ای از به‌کارگیری ترکیب روش‌های یادگیری عمیق و پردازش زبان طبیعی برای حل مسائل واقعی است.

استفاده از Flask به عنوان رابط کاربری

برای این پروژه یک رابط وب ساده با استفاده از فریم‌ورک **Flask** طراحی شد تا کاربر بتواند به راحتی تصاویر کمیک را آپلود کرده و نتایج استخراج متن را مشاهده کند.

مهم‌ترین قابلیت‌ها:

- **آپلود تصویر:** کاربر می‌تواند یک یا چند تصویر با فرمت‌های jpg, jpeg, png, webp آپلود کند.

- پردازش خودکار: پس از آپلود، تصویر به ماژول پردازش فرستاده می‌شود تا متن حباب‌های دیالوگ استخراج گردد.
- اصلاح متن: متن‌های استخراج‌شده توسط ماژول **SpellChecker** و **LanguageTool** اصلاح می‌شوند.
- خروجی نهایی: نتیجه شامل تعداد حباب‌های متنی، تعداد کل کلمات، و میانگین کلمات به ازای هر حباب است که به کاربر نمایش داده می‌شود.

ساختار کد:

1. **تنظیمات آپلود:** محل ذخیره‌ی فایل‌ها مشخص می‌شود و تنها پسوندهای مجاز پذیرفته می‌شوند.
2. **صفحه اصلی (/):** فرم آپلود به کاربر نمایش داده می‌شود.
3. **آپلود و پردازش (/upload):** فایل‌ها پردازش شده و نتایج در قالب یک صفحه HTML نمایش داده می‌شوند.
4. **نمایش نتایج:** از قالب‌های HTML (مانند `index.html` و `results.html`) برای نمایش نتایج استفاده می‌شود.

تشخیص و استخراج متن (Detection & OCR)

برای استخراج متن از حباب‌های دیالوگ در کمیک، از ترکیب **YOLO** برای تشخیص حباب‌ها و **EasyOCR** برای شناسایی متن داخل آن‌ها استفاده شد.

مراحل به این صورت است:

1. **تشخیص حباب‌ها:**
 - با استفاده از مدل YOLO، تمامی حباب‌های متنی در تصویر کمیک شناسایی می‌شوند.
 - مختصات هر حباب استخراج شده و در صورت نیاز، تصاویر بریده‌شده‌ی آن‌ها ذخیره می‌شوند.
2. **پیش‌پردازش تصویر (Preprocessing):**

- تصاویر بریده شده از حباب‌ها با روش‌هایی مثل بزرگ‌نمایی (Upscaling) و فیلتر Gaussian Blur پیش‌پردازش می‌شوند تا کیفیت متن بهبود یابد.
- 3. **تشخیص متن (OCR):**
 - از کتابخانه EasyOCR برای شناسایی متن داخل هر حباب استفاده شده است.
 - متون استخراج شده در صورت نیاز برای تصحیح به ماژول Spellcheck ارسال می‌شوند.
- 4. **مرتب‌سازی به ترتیب خواندن (Reading Order):**
 - چون موقعیت حباب‌ها روی صفحه تصادفی است، متن‌ها بر اساس مختصات (بالا به پایین، چپ به راست) مرتب‌سازی می‌شوند تا ترتیب طبیعی خواندن حفظ شود.

پیش‌پردازش تصویر (Preprocessing)

برای بهبود دقت **تشخیص متن (OCR)**، تصاویر حباب‌های دیالوگ قبل از ارسال به EasyOCR پیش‌پردازش می‌شوند. این مرحله شامل چند تکنیک اصلی است:

1. **بزرگ‌نمایی پویا (Dynamic Upscaling)**
 - اگر اندازه‌ی حباب کوچک باشد، تصویر ۳ برابر بزرگ‌تر می‌شود.
 - اگر اندازه‌ی حباب بزرگ باشد، تنها ۱.۵ برابر بزرگ‌تر می‌شود.
 - این کار باعث می‌شود متن‌های کوچک خوانا تر شوند.
2. **تبدیل به سطح خاکستری (Grayscale Conversion)**
 - تصاویر رنگی به خاکستری تبدیل می‌شوند تا پردازش سریع‌تر و دقیق‌تر انجام شود.
3. **بهبود کنتراست با CLAHE**
 - از تکنیک CLAHE برای افزایش وضوح متن در پس‌زمینه‌های نامنظم استفاده می‌شود.
4. **حذف نویز (Noise Reduction)**

◦ فیلتر Bilateral برای حذف نویز استفاده می‌شود بدون اینکه لبه‌های متن آسیب ببیند.

5. عملیات مورفولوژیکی (Morphological Operations)

◦ با عملیات Close، سوراخ‌های کوچک در متن بسته می‌شوند تا OCR راحت‌تر عمل کند.

6. تغییر اندازه برای YOLO (YOLO Resize)

◦ در صورت نیاز، تصویر به ابعاد سازگار با YOLO (۶۴۰×۶۴۰) تغییر اندازه داده می‌شود.

ماژول (Utilities (utils.py

در این ماژول، توابع کمکی برای ذخیره و مرتب‌سازی حباب‌های گفتاری پیاده‌سازی شده است. این توابع بخش اصلی پردازش را ساده‌تر کرده و امکان دیباگ و کنترل بهتر را فراهم می‌کنند.

save_bubble_images

این تابع، حباب‌های شناسایی‌شده توسط مدل YOLO را از تصویر اصلی جدا کرده و به صورت فایل‌های تصویری مستقل ذخیره می‌کند.

- کاربرد اصلی: دیباگ و بررسی دقت تشخیص مدل.
- جزئیات: هر ناحیه‌ی شناسایی‌شده (bounding box) بریده شده و در پوشه‌ی جداگانه ذخیره می‌شود. این کار کمک می‌کند بتوانیم کیفیت بخش OCR (تشخیص متن) را روی حباب‌ها ارزیابی کنیم.

sort_bubbles_by_reading_order_

این تابع وظیفه‌ی مرتب‌سازی حباب‌ها را بر اساس ترتیب طبیعی خواندن متن بر عهده دارد.


- از مختصات بالا-چپ هر حباب (y, x) استفاده می‌کند.
- یک فرمول وزنی برای ترکیب مختصات به کار می‌برد:

$$\text{score} = y + \alpha * x$$

که در آن:

- y اولویت بالاتری دارد (خواندن از بالا به پایین).
- x با ضریب α در نظر گرفته می‌شود تا ترتیب چپ به راست هم لحاظ شود.

اهمیت این مازول

- ذخیره‌سازی حباب‌ها امکان مقایسه‌ی دیداری خروجی مدل و متن استخراج‌شده را فراهم می‌کند.
-  مرتب‌سازی خواندن، اطمینان می‌دهد که متن نهایی همان ترتیبی را دارد که خواننده در کمیک تجربه می‌کند.

پردازش متن و اصلاح خطاها (Text Processing & Spellchecking)

بعد از اینکه متن خام توسط EasyOCR استخراج شد، معمولاً شامل نویزهایی مثل غلط‌های املائی، حروف جاافتاده یا تشخیص اشتباه کاراکترهاست. برای رفع این مشکلات مراحل زیر در سیستم پیاده‌سازی شده‌اند:

1. استفاده از SpaCy برای شناسایی موجودیت‌ها (NER)

- اسامی شخصیت‌ها و سازمان‌ها با مدل زبان `en_core_web_sm` تشخیص داده می‌شوند.
- این موجودیت‌ها موقتاً با **placeholder** جایگزین می‌شوند تا در فرآیند اصلاح تغییر نکنند.

2. اصلاح گرامر و دستور زبان (Grammar Correction)

- ابزار `language_tool_python` روی هر جمله اعمال می‌شود تا ساختار گرامری و نگارشی اصلاح شود.

3. اصلاح غلط‌های املائی (Spellchecking)

- از `pyspellchecker` برای اصلاح کلمات نادرست استفاده می‌شود.
- اگر کلمه‌ای ناشناخته باشد نزدیک‌ترین کلمه معتبر جایگزین می‌شود.

4. بازگرداندن موجودیت‌ها (Entity Restoration)

- بعد از اصلاح متن، موجودیت‌های محافظت‌شده دوباره به جای اصلی خود برگردانده می‌شوند.

5. مرتب‌سازی متن‌ها بر اساس ترتیب خواندن

- موقعیت حباب‌ها (مختصات x و y) ذخیره شده و بر اساس الگوریتم مرتب‌سازی مخصوص

sort_bubbles_by_reading_order_ به ترتیب منطقی
(چپ به راست و بالا به پایین) مرتب می‌شوند.

تکنولوژی‌ها و کتابخانه‌های استفاده‌شده

1. **YOLO (Ultralytics)**
 - یک مدل یادگیری عمیق برای تشخیص اشیا (Object Detection).
 - در این پروژه برای پیدا کردن موقعیت دقیق *حباب‌های دی‌الوگ* در صفحات کمیک استفاده شده.
2. **EasyOCR**
 - یک کتابخانه OCR متن‌باز برای استخراج متن از تصویر.
 - اینجا متن داخل حباب‌ها رو به صورت رشته استخراج می‌کنه.
3. **OpenCV (cv2)**
 - کتابخانه پردازش تصویر.
 - برای پیش‌پردازش تصویرها (مثل تغییر اندازه، بلور، بهبود کنتراست) و آماده‌سازی برای OCR استفاده شده.
4. **spaCy**
 - یک NLP library قدرتمند.
 - در پروژه برای شناسایی اسامی خاص (مانند شخصیت‌ها یا سازمان‌ها) استفاده می‌شه تا توی اصلاح متن اشتباه تصحیح نشن.
5. **LanguageTool**
 - یک ابزار بررسی دستور زبان و نگارش.
 - در این پروژه متن خروجی OCR رو بررسی و اصلاح می‌کنه تا روان‌تر و درست‌تر بشه.
6. **pySpellChecker**
 - یک ابزار ساده برای اصلاح غلط‌های املایی.
 - کلماتی که EasyOCR اشتباه تشخیص می‌ده، با این کتابخانه تصحیح می‌شن.
7. **Flask**
 - یک وب‌فریم‌ورک سبک پایتون.
 - برای ساختن رابط کاربری (UI) ساده جهت تست و نمایش خروجی OCR استفاده شده.
8. **NumPy**
 - کتابخانه محاسبات عددی در پایتون.
 - در این پروژه بیشتر برای عملیات روی ماتریس‌ها و تصاویر (که به صورت آرایه هستند) استفاده می‌شه.