

مدل CRNN:

مدل CRNN (Convolutional Recurrent Neural Network) برای OCR (تشخیص کاراکتر نوری) زبان فارسی یک معماری ترکیبی از CNN و LSTM است که از مقاله‌ای الهام گرفته شده است. این مدل جانشین مدل قبلی با نام `hezarai/crnn-base-fa-64x256` است و شامل بهبودهای زیر می‌باشد:

1. استفاده از دیتاست ۵ برابر بزرگتر: این مدل از یک دیتاست بسیار بزرگتر نسبت به مدل قبلی استفاده می‌کند که دقت و کارایی آن را بهبود می‌بخشد.
2. تغییر اندازه تصویر ورودی از ۲۵۶x۶۴ به ۳۸۴x۳۲: این تغییر باعث می‌شود که مدل بتواند بهتر با تصاویر ورودی کار کند و دقت بیشتری داشته باشد.
3. افزایش حداکثر طول خروجی از ۶۴ به ۹۶: طول نمونه‌ها در دیتاست حداکثر ۴۸ بود که برای جلوگیری از مشکلات CTC (Connectionist Temporal Classification) در طول آموزش این افزایش طول انجام شده است.
4. پشتیبانی از اعداد و کاراکترهای ویژه: مدل جدید کاراکترهای بیشتری را پشتیبانی می‌کند که شامل اعداد و کاراکترهای ویژه می‌شود. این تغییرات در فایل `model_config.yaml` تعریف شده‌اند.
5. مدیریت خودکار کاراکترهای LTR (چپ به راست): مانند اعداد در متن‌های فارسی: این مدل توانایی مدیریت کاراکترهایی که جهت نوشتن آن‌ها از چپ به راست است را به صورت خودکار دارد.

موارد استفاده و محدودیت‌ها:

- بهینه‌سازی برای اسناد چاپی/اسکن‌شده: این مدل به طور خاص برای تشخیص کاراکتر در اسناد چاپی و اسکن‌شده بهینه‌سازی شده است.
- مناسب برای متن‌های با طول حداکثر ۵۰ کاراکتر: بهترین عملکرد این مدل برای متن‌هایی با طول حداکثر ۵۰ کاراکتر است. برای استفاده در یک خط لوله کامل OCR، ابتدا باید از یک مدل تشخیص متن برای استخراج باکس‌های متن (ترجیحاً در سطح کلمه) استفاده شود و سپس این مدل برای تشخیص کاراکترها به کار گرفته شود.
- قابلیت فاین‌تیون برای حوزه‌های دیگر: این مدل می‌تواند برای حوزه‌های دیگر مانند تشخیص شماره پلاک خودرو یا متن‌های دست‌نویس نیز فاین‌تیون شود.

این مدل با ترکیب ویژگی‌های CNN و LSTM، توانایی استخراج ویژگی‌های مکانی و زمانی از تصاویر متنی را داراست و می‌تواند به طور موثری کاراکترهای موجود در تصاویر فارسی را تشخیص دهد.

نمونه خروجی مدل :

اطمینان حاصل کردم	
crnn-base-fa-64x256	اطمینان حاصل کردم
trocr-base-fa-v1	اصفهان حاصل کرد عم
trocr-base-fa-v2	لحیثان حاصل کردم
crnn-fa-printed-96-long	اطمینان حاصل کردم

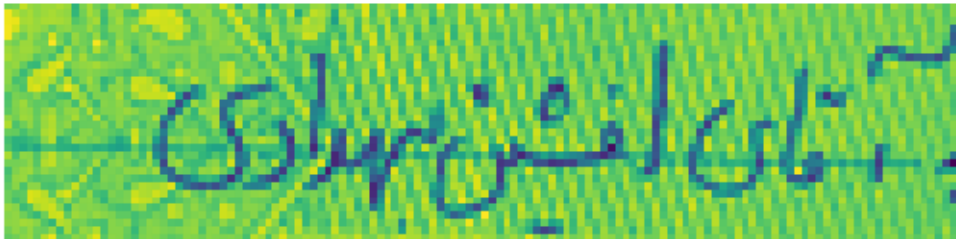

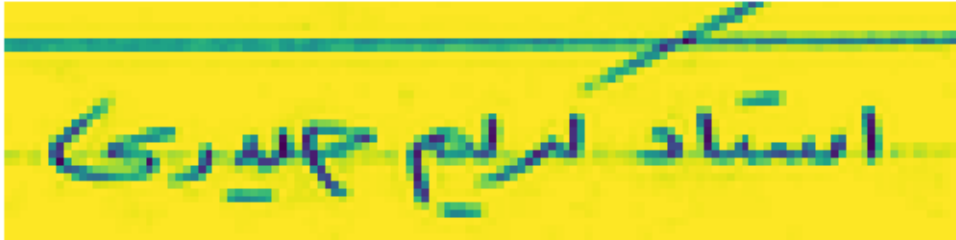
تعداد لایه‌ها و غیره، عملکرد مدل را بهبود داد.	
crnn-base-fa-64x256	تعاه دیما و غیره، صلعرد معد را بهوه دام
trocr-base-fa-v1	عنا مد مدوه من حیر عنه
trocr-base-fa-v2	
crnn-fa-printed-96-long	تعداد لایه‌ها و غیره، علکرد مدل ر، بهبود دادم

ورژن crnn-fa-printed-96-long روی دیتا یکسان از همه بهتر عمل کرده و بهترین مدل در بین ورژن های موجود هست

نوع این دیتا ها به صورت جمله ای یا کلمات هست ولی با دادن دیتا های با پاراگراف بیشتر مدل [hezarai/crnn-fa-printed-96-long](https://github.com/hezarai/crnn-fa-printed-96-long) اصلا دقت خوبی ندارد

ورودی و خروجی مدل:

آموزش و تست مدل :

Input	Output
Prediction: [{'text': 'یادہ بن شاہ اقا'}] 	اقای اشنبہدای
Prediction: [{'text': 'مہدی روگلا یحارط'}] 	طراحی الگوریتم
Prediction: [{'text': 'یاردیج مہرک داتسا'}] 	آئموہ فاینال قر

1. آمادہ سازی دادہ ہا

جمع آوری دادہ ہا:

تصاویر مختلفی از چک‌ها که جمع‌آوری شده است. این تصاویر شامل انواع چک‌های مختلف با قالب‌ها و طرح‌های متنوع بودند. این دیتا‌ها در قالب pdf بودند که آقای حامی آنها را پیش‌پردازش کرد و به صورت clean شده و تمیز شده درآوردند

تقسیم قسمت‌های مختلف چک :

تقسیم قسمت‌های مختلف چک با yolov8 انجام شد و چک به قسمت‌های مختلف شکسته شد

برچسب‌گذاری داده‌ها:

برای برچسب‌گذاری داده‌ها در یک فایل اکسل انجام شد که این اکسل رو تا ستون دارد که اولین ستون مسیر عکس‌های برش خورده چک و دیتا‌های جمع‌آوری شده توسط خودم انجام شد و دومین ستون متن اون عکس هست نمونه :

file_name	text
10 nomre final.png	10 نمر فینال
10.png	انواع روش‌های مقایسه الگوریتم‌ها
11.png	به سخت افزار بستگی دارد
12.png	چه کوچک تاثیر ندارد

تقسیم داده‌ها:

داده‌ها را به مجموعه‌های آموزشی، اعتبارسنجی و تست تقسیم کردم (به عنوان مثال، 70% آموزش، 20% اعتبارسنجی، 10% تست).

2. آماده‌سازی محیط توسعه

نصب کتابخانه‌ها و ابزارهای مورد نیاز:

hezar و pandas را نصب کردم.

تنظیم محیط کاری:

یک ساختار پوشه مناسب برای پروژه خود ایجاد کردم که نتایج مدل در این پوشه ذخیره شود

3. آموزش مدل و fine_tune

دیتا ورودی اکسل را به مدلی که خودم برای hezar توسعه دادم دادم تا با داده دست نویس آموزش داده شود

4. بهبود و بهینه سازی

تنظیم هایپرپارامترها:

با تغییر هایپرپارامترها مانند نرخ یادگیری، تعداد لایه ها و غیره، عملکرد مدل را بهبود دادم.

افزایش داده ها:

از تکنیک های افزایش داده ها (data augmentation) برای افزایش تنوع مجموعه داده استفاده کردم و مدل را مجدد آموزش دادم.

تنظیم دقیق مدل: (Fine-tuning)

اگر دسترسی به یک مدل پیش آموزش یافته داشتم، از آن برای تنظیم دقیق مدل خود استفاده کردم.



