

# مطالعه در مورد مدل های فارسی انگلیسی فعلی برای ocr در سطح word: 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 فاین تیون برای حوزه های دیگر: این مدل می تواند برای حوزه های دیگر مانند تشخیص شماره پلاک خودرو یا متن های دستنویس نیز فاین تیون شود.

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

نمونه خروجی مدل :

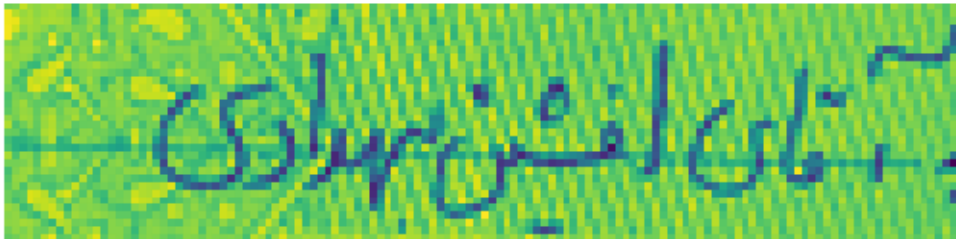

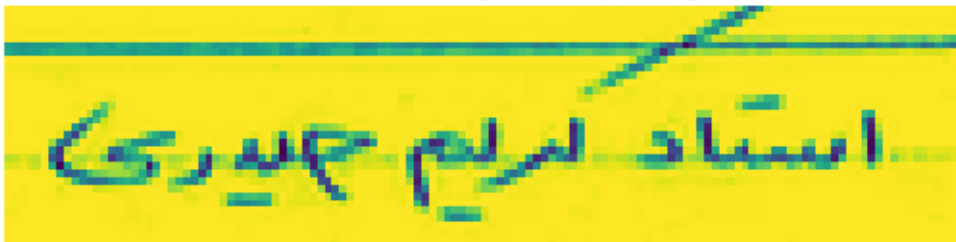
اطمینان حاصل کردم	
crnn-base-fa-64x256	'اطمینان حاصل کردم
trocr-base-fa-v1	اصفهان حاصل کرد عم
trocr-base-fa-v2	لحیثان حاصل کردم
crnn-fa-printed-96-long	اطمینان حاصل کردم

تعداد لایه ها و غیره، عملکرد مدل را بهبود داد.	
crnn-base-fa-64x256	تجاه دیما و غیره، صلکرد معد را بهوه دام
trocr-base-fa-v1	عنا مد مدوه من حیر عنه
trocr-base-fa-v2	
crnn-fa-printed-96-long	تعداد لایه ها 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:

Input	Output
Prediction: [{"text": 'یادہ بن شاہ اقا'}] 	آقای اشنبہدای
Prediction: [{"text": 'متریوگلا یحارط'}] 	طراحی الگوریتم
Prediction: [{"text": 'یردیج میرک داتسا'}] 	آئموہ فاینال قر

آموزش و تست مدل :

1. آماده سازی داده ها

جمع آوری داده ها:

تصاویر مختلفی از چکها که جمع آوری شده است . این تصاویر شامل انواع چکهای مختلف با قالبها و طرحهای متنوع بودند. این دیتا ها در قالب pdf بودش که اقای حامی آنها را پیش پردازش کرد و به صورت clean شده و تمیز شده درآوردند

تقسیم قسمت های مختلف چک :

تقسیم قسمت های مختلف چک با yolov8 انجام شد و چک به قسمت های مختلف شکسته شد

برچسبگذاری داده ها:

برای برچسبگذاری داده ها در یک فایل اکسل انجام شد که این اکسل رو تا ستون دارد که اولین ستون مسیر عکس های برش خورده چک و دیتا های جمع اوری 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 روش های مقایسه الگوریتم ها
11.png	به سخت افزار بستگی دارد
12.png	چه کوچک تاثیر ندارد

تقسیم داده ها:

داده ها را به مجموعه های آموزشی، اعتبارسنجی و تست تقسیم کردم (به عنوان مثال، 70% آموزش، 20% اعتبارسنجی، 10% تست).

2. آماده سازی محیط توسعه

نصب کتابخانه ها و ابزارهای مورد نیاز:

hezar 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.



ده مرداد یک هزار و سیصد و خود و شش

ID 0.80

۱۹۸۱/۶۹۸۷۲۷

بانک تجارت

BANK TEJARAT

tarikh A 0.73

۱۳۴۶/۰۵/۱۰

تاریخ به نام و نوشته شود

معادل شصت و شش میلیون و دویست و هشتاد هزار تومان تمام

بموجب این چک مبلغ شصت و سی و دو میلیون و هشتصد هزار ریال تمام

ریال

یا به حواله کرد

mablagh A 0.56

۶۳۲,۸۰۰,۰۰۰Rls

۶۳۲,۸۰۰,۰۰۰/۱

شماره حساب ۰۰۲۳ ۰۸ ۳۹۰۸

مصطفی و علی اصغر قلم خوردگی و بستن بوجه

IR ۱۳۰۱ ۸۰۰۰ ۰۰۱۴ ۳۹۰۸ ۰۰۲۳

20 10 18 198 1 698 7 27 1 7

	۱۳۴۶/۰۵/۱۰	ماری چها ۱
	۶۳۲,۸۰۰,۰۰۰/۱	16016111
	۱۹۸۱/۶۹۸۷۲۷	1981898727

بیست و یکم مرداد ماه یک هزار و چهارصد و یک

تاریخ ۱۴۰۱-۱۰-۲۱

تاریخ به نام و نوشته شود

بموجب این چک مبلغ (پروت) پانصد میلیون ریال تمام

در شرکت تندیس پلاستیک بابت نایلکس بابت کد شناسه ۱۴۰۰۹۹۳۲۴۹۵

شماره و شناسی اتباع ملی

۵۴۱۰۳۹۵

حافظ شمالی کد

۲۷۶۳ ۰۰۰۱ ۵۷۳۶ ۲۵۲۰

۱۴۰۰۳/ ۴۶۰۵۲۵

بانک پارسیان

شعبه

ریال

یا به حواله کرد و پرداخت

مبلغ (ریال)

۵۰۰.۰۰۰.۰۰۰/۱

#500,000,000 RLS

شماره حساب ۰۰۲۳ ۰۸ ۳۹۰۸

مصطفی و علی اصغر قلم خوردگی و بستن بوجه

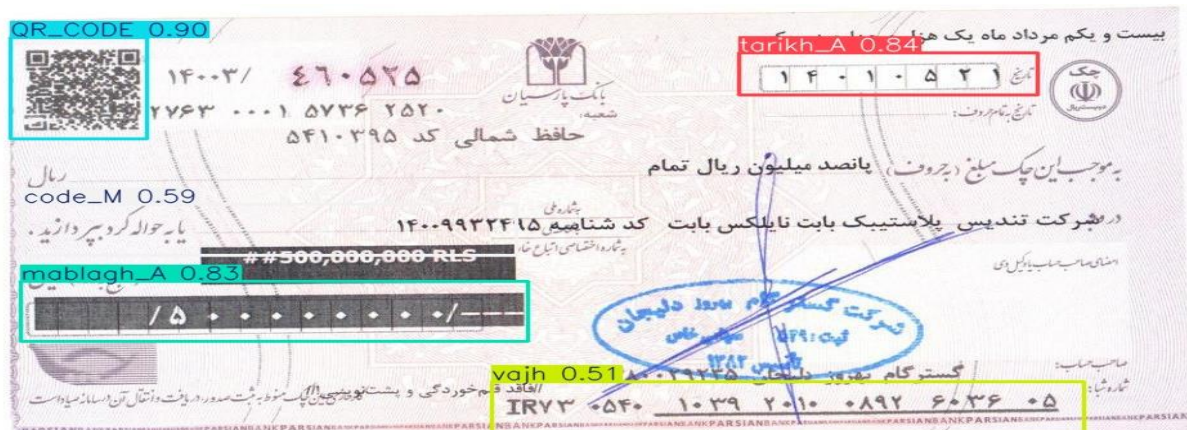
IRV ۳۰۵۴۰ ۱۰۳۹ ۲۰۱۰ ۰۸۹۲ ۶۰۳۶ ۰۵

۱۰۷۸۰۰۲۲۴۵

شماره شبا

PARSIAN





	500000000
	14010521
	nan
	460525/14003

## مدل TrOCR :

مدل TrOCR یک مدل encoder-decoder است که شامل یک ترنسفورمر تصویری به عنوان encoder و یک ترنسفورمر متنی به عنوان decoder می‌باشد. این مدل توسط Li و همکاران در مقاله‌ای با عنوان "TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models" در این مخزن [microsoft/trocr-base-handwritten](https://huggingface.co/microsoft/trocr-base-handwritten) منتشر شده است.

## توضیحات مدل TrOCR:

- Encoder تصویر: Encoder تصویر از وزن‌های پیش‌آموزش داده شده مدل BEiT بهره می‌برد.

- Decoder متن: Decoder متن نیز از وزن‌های مدل RoBERTa استفاده می‌کند.

## نحوه عملکرد:

تصاویر به مدل به صورت توالی‌ای از قطعات با اندازه ثابت (رزولوشن 16x16) ارائه می‌شوند که به طور خطی جاسازی شده‌اند و همچنین این قطعات با استفاده از encoder تصویری BEiT پردازش می‌شوند. سپس، ویژگی‌های استخراج شده از رمزگذار به decoder متنی RoBERTa ارسال می‌شوند تا توالی‌های متنی تولید شوند.

## نتایج و کاربردها:

مدل TrOCR توانسته است در تشخیص متن از تصاویر (OCR) به نتایج برجسته‌ای دست یابد و در کاربردهای مختلفی از جمله تشخیص متون دستنویس و چاپی مورد استفاده قرار گیرد. برخی از ویژگی‌ها و کاربردهای این مدل شامل موارد زیر است:

- سرعت بالا: مدل TrOCR بهینه‌سازی شده و می‌تواند با سرعت بالایی عملیات OCR را انجام دهد.

- دقت بالا: استفاده از وزن‌های پیش‌آموزش داده شده مدل‌های BEiT و RoBERTa باعث افزایش دقت در تشخیص متون شده است.

- انعطاف‌پذیری: این مدل قابلیت استفاده در کاربردهای مختلف از جمله تشخیص متون دستنویس، چاپی و حتی متون پیچیده‌تر را داراست.

## منابع و مستندات:



- مقاله مرجع مدل: [TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models](https://arxiv.org/abs/2109.10282)

- مخزن مدل: [microsoft/trocr-base-handwritten](https://huggingface.co/microsoft/trocr-base-handwritten)

## معرفی دیتاست IAM:

مجموعه داده‌های فرم‌های دست‌نویس IAM شامل فرم‌هایی از متن انگلیسی دست‌نویس است که می‌تواند برای آموزش و آزمایش تشخیص‌دهنده‌های متن دست‌نویس و انجام آزمایش‌های شناسایی و تأیید نویسنده استفاده شود.

## محتوا:

مجموعه داده شامل فرم‌های کاملی از متن دست‌نویس بدون محدودیت است که با وضوح dpi300 اسکن شده و به‌عنوان تصاویر PNG با 256 سطح خاکستری ذخیره شده‌اند. فرم‌ها به دایرکتوری‌های جداگانه تقسیم می‌شوند به طوری که همه فرم‌ها در هر دایرکتوری توسط یک شخص نوشته می‌شوند.

## نتیجه‌گیری:

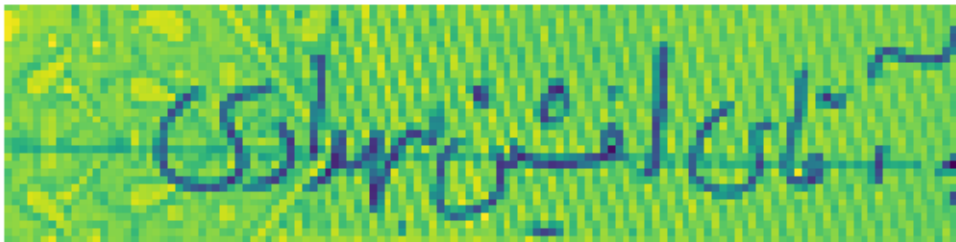

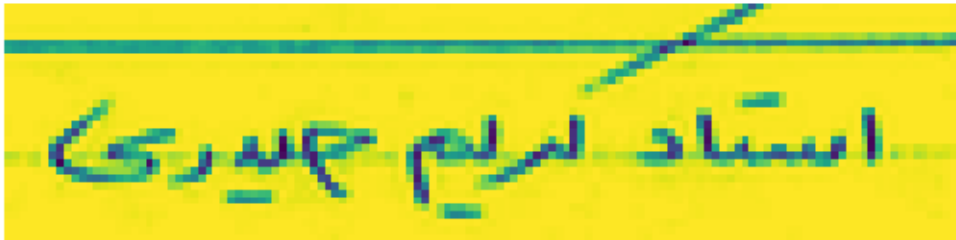
مدل TrOCR یک ابزار قدرتمند و دقیق برای تشخیص متن از تصاویر است که با استفاده از تکنیک‌های پیشرفته ترنسفورمر می‌تواند به دقت و سرعت بالایی دست یابد. این مدل در کاربردهای مختلفی مانند تشخیص متون دست‌نویس و چاپی بسیار موثر است و می‌تواند به عنوان یک راه‌حل جامع OCR مورد استفاده قرار گیرد.

## چکیده مقاله:

تشخیص متن یک مسئله پژوهشی طولانی‌مدت برای دیجیتالی‌سازی اسناد است. روش‌های موجود معمولاً بر اساس CNN برای درک تصویر و RNN برای تولید

کاراکترها ساخته شده اند. علاوه بر این، معمولاً یک مدل زبان دیگر به عنوان یک مرحله پس‌پردازش برای بهبود دقت کلی نیاز است. در این مقاله، ما یک روش تشخیص متن انتها به انتها با استفاده از مدل‌های ترنسفورمر پیش‌آموزش‌داده شده تصویری و متنی، به نام TrOCR، پیشنهاد می‌دهیم که از معماری ترنسفورمر برای هر دو درک تصویر و تولید متن کلمه‌ای بهره می‌برد. مدل TrOCR ساده اما موثر است و می‌تواند با داده‌های مصنوعی بزرگ‌مقیاس پیش‌آموزش داده شده و با مجموعه داده‌های برجسب‌گذاری‌شده انسانی تنظیم دقیق شود. آزمایش‌ها نشان می‌دهند که مدل TrOCR از مدل‌های فعلی در تشخیص متون چاپی، دست‌نویس و صحنه پیشی می‌گیرد.

ورودی و خروجی مدل:

Input	Output
Prediction: [{'text': 'یادہ بن شاہ اقا'}] 	آقای اشنہدای
Prediction: [{'text': 'متریوگلا یحارطا'}] 	طراحی الگوریتم
Prediction: [{'text': 'یردیج میرک داتسا'}] 	آئموه فاینال قر

آموزش و تست مدل :

1. آماده سازی داده ها

جمع آوری داده ها:

تصاویر مختلفی از چکها که جمع آوری شده است . این تصاویر شامل انواع چکهای مختلف با قالبها و طرحهای متنوع بودند. این دیتا ها در قالب pdf بودش که آقای حامی آنها را پیش پردازش کرد و به صورت clean شده و تمیز شده درآوردند

تقسیم قسمت های مختلف چک :

تقسیم قسمت های مختلف چک با yolov8 انجام شد و چک به قسمت های مختلف شکسته شد

برچسبگذاری داده ها:

برای برچسبگذاری داده ها در یک فایل اکسل انجام شد که این اکسل رو تا ستون دارد که اولین ستون مسیر عکس های برش خورده چک و دیتا های جمع آوری شده توسط خودم انجام شد و دومین ستون متن اون عکس هست نمونه :

file_name	text
10 nomre final.png	10 نفرینال
10.png	انواع روش های مقایسه الگوریتم ها
11.png	به سخت افزار بستگی دارد
12.png	چه کوچک تاثیر ندارد

تقسیم داده ها:

داده ها را به مجموعه های آموزشی، اعتبارسنجی و تست تقسیم کردم (به عنوان مثال، 70% آموزش، 20% اعتبارسنجی، 10% تست).

2. آماده سازی محیط توسعه

نصب کتابخانه ها و ابزارهای مورد نیاز:

transformers

datasets

jiwer

نصب کردم .

### 3- توسعه کد آموزش :

#### 1. ایجاد دیتاست:

- تعریف یک کلاس دیتاست برای پردازش و آماده‌سازی تصاویر و متون.
- آماده‌سازی تصاویر (تغییر اندازه و نرمال‌سازی) و تبدیل متون به توکن‌ها.

#### 2. بارگذاری پردازشگر: TrOCR

- بارگذاری پردازشگر مدل TrOCR برای آماده‌سازی داده‌ها.

#### 3. ایجاد مجموعه‌های آموزشی و آزمایشی:

- ایجاد دیتاست‌های آموزشی و آزمایشی با استفاده از پردازشگر و داده‌های تقسیم شده.

#### 4. نمایش آماری دیتاست‌ها:

- نمایش تعداد نمونه‌های موجود در مجموعه‌های آموزشی و آزمایشی.

#### 5. نمایش یک نمونه از دیتاست:

- نمایش یک نمونه از داده‌های پردازش شده برای اطمینان از درستی آماده‌سازی داده‌ها.

#### 6. بارگذاری مدل: TrOCR

- بارگذاری مدل از پیش آموزش دیده TrOCR.

#### 7. تنظیمات مدل:

- تنظیم توکن‌های ویژه برای مدل.
- تنظیم اندازه واژگان (vocab size).
- تنظیم پارامترهای beam search.

#### 8. تنظیمات آموزش:

- تعریف آرگومان‌های آموزش شامل تنظیمات مربوط به تعداد اپوک‌ها، اندازه بچ، استراتژی ارزیابی و غیره.

#### 9. تعریف معیار ارزیابی:

- بارگذاری معیار ارزیابی (cer (Character Error Rate).

- تعریف تابع محاسبه معیار ارزیابی برای استفاده در طول آموزش.

## 10. ایجاد و اجرای Trainer:

- ایجاد یک نمونه از Seq2SeqTrainer با استفاده از مدل، پردازشگر، آرگومان‌های آموزش، معیار ارزیابی و دیتاست‌ها.
- اجرای آموزش مدل با استفاده از `trainer.train()`.

**نتیجه مدل:** نتیجه مدل اصلا خوانا و قابل بررسی اصلا نیست

## مدل (CNN-bilstm):

مقدمه:

امروزه هوش مصنوعی و یادگیری عمیق توانایی انجام بسیاری از وظایف را دارند، از جمله چت کردن، خواندن تصاویر و شناسایی صداها. یکی از کاربردهای مهم هوش مصنوعی، تشخیص نوری کاراکتر (OCR) است که می‌تواند متون را از تصاویر یا ویدیوها استخراج کند. این گزارش به توضیح پروژه‌ای می‌پردازد که در آن از تکنیک‌های یادگیری عمیق برای شناسایی متن دست‌نویس با استفاده از TensorFlow استفاده شده است.

## تعریف OCR:

تشخیص نوری کاراکتر به زبان ساده به معنای شناسایی و استخراج متون از تصاویر است. این فناوری در بسیاری از دستگاه‌ها از جمله تلفن‌های

هوشمند استفاده می‌شود. روش‌های مختلفی برای پیاده‌سازی OCR وجود دارد که در این پروژه از معماری **Encoder-Decoder** استفاده شده است.

### نقاط قوت مدل:

- دقت بالا: مدل **Encoder-Decoder** با استفاده از ساختارهای پیچیده مانند **LSTM** و **BiLSTM** قادر به درک و تفسیر متون دستنویس با دقت بالا است.
- پیش‌پردازش موثر: پیش‌پردازش دقیق تصاویر و برجسبها به بهبود عملکرد مدل کمک می‌کند.
- تنظیمات قابل انعطاف: امکان تنظیم هایپرپارامترها و استفاده از لایه‌های مختلف مانند لایه‌های کانولوشن و لایه‌های تراکم برای بهینه‌سازی مدل وجود دارد.

### نقاط ضعف مدل:

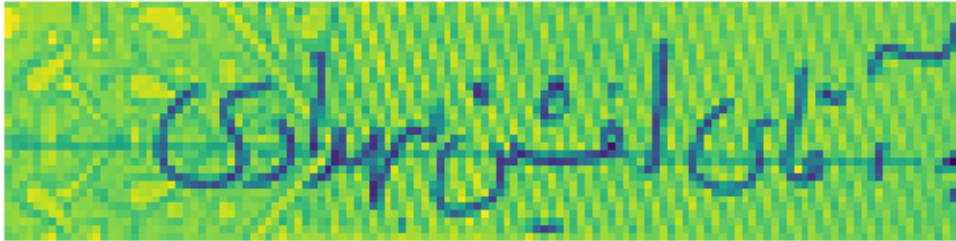

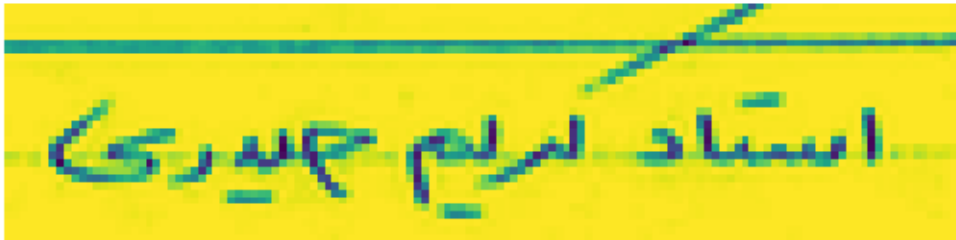
- پیچیدگی محاسباتی: مدل‌های **Encoder-Decoder** به منابع محاسباتی زیادی نیاز دارند و زمان آموزش طولانی‌تری دارند.
- نیاز به داده‌های زیاد: برای آموزش دقیق و بهبود عملکرد، مدل به مجموعه داده‌های بزرگ و متنوع نیاز دارد.
- خطر بیش‌برازش: به دلیل پیچیدگی مدل، خطر بیش‌برازش وجود دارد که با استفاده از تکنیک‌های مانند توقف زودهنگام (**Early Stopping**) کنترل می‌شود.

### نتیجه‌گیری:

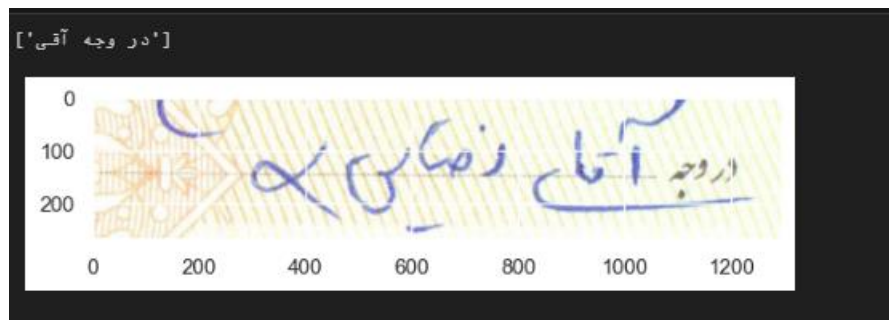
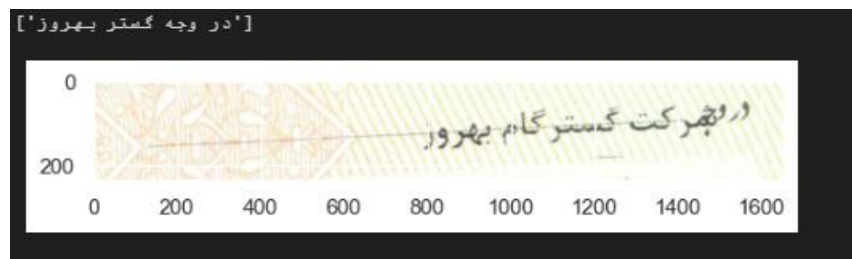
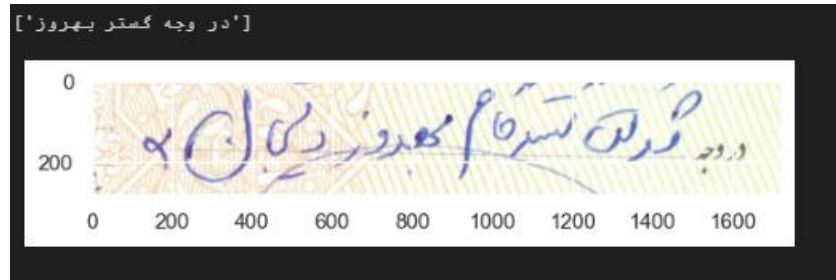
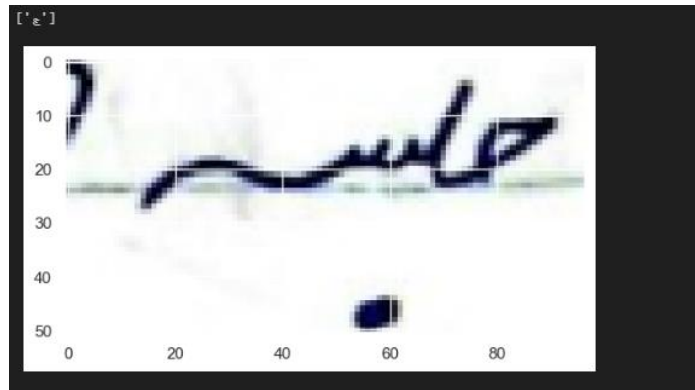
این پروژه نشان می‌دهد که با استفاده از تکنیک‌های یادگیری عمیق و معماری **Encoder-Decoder** می‌توان به دقت بالایی در شناسایی متون دستنویس دست یافت. هرچند که این مدل‌ها نیازمند منابع محاسباتی زیادی هستند و به داده‌های گسترده‌ای نیاز دارند، توانایی‌های پیشرفته آنها در پردازش و شناسایی دقیق متون بسیار قابل توجه است.



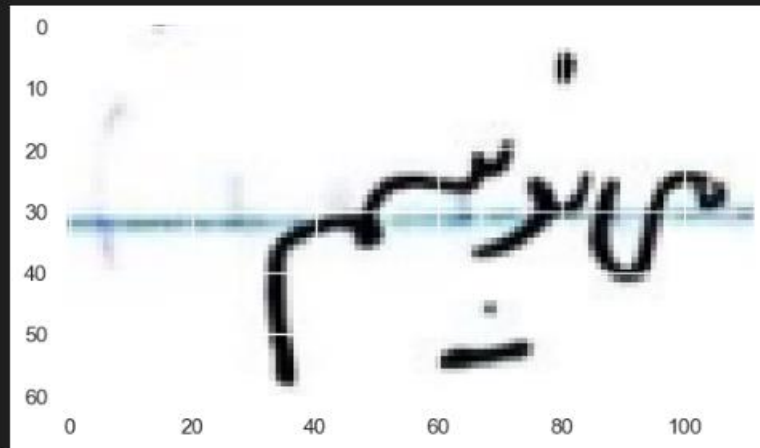
ورودی و خروجی مدل:

Input	Output
Prediction: [{'text': 'یادہ بن شاہ اقا'}] 	آقای اشنہدای
Prediction: [{'text': 'متریوگلا یحارطا'}] 	طراحی الگوریتم
Prediction: [{'text': 'یردیج میرک داتسا'}] 	آئموہ فاینال قر

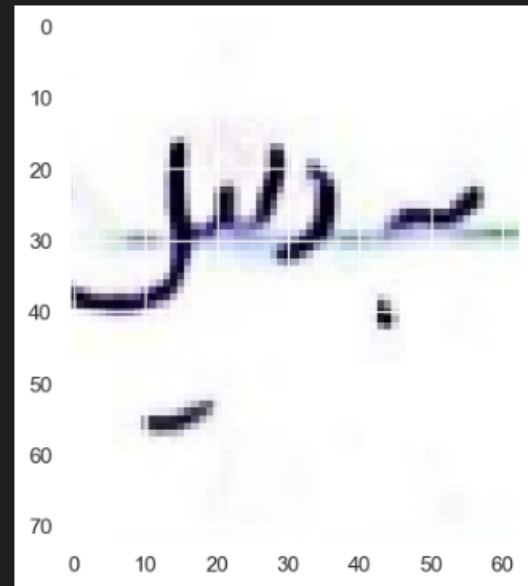
نتیجہ مدل:



['مینو']

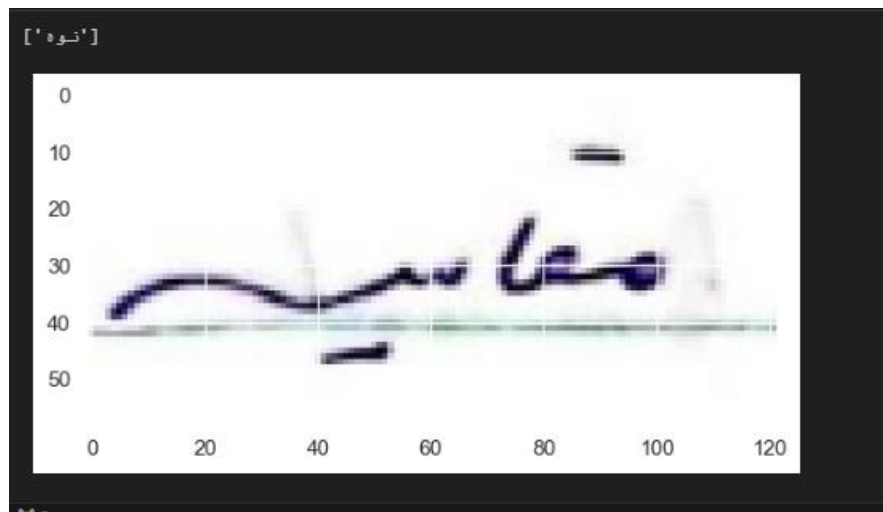


['بیل']



['10']





## مقایسه مدل‌ها : TR\_OCR - crnn\_fa\_printed\_96\_long- CNN\_BILSTM-

در این بخش، سه مدل مختلف برای شناسایی متن مورد بررسی قرار می‌گیرند: TR OCR، hezarai/crnn-fa-printed-96-long و مدل CNN-BILSTM. هر کدام از این مدل‌ها نقاط قوت و ضعف خاص خود را دارند که در جدول زیر به صورت دقیق مقایسه شده‌اند.

ویژگی	TR OCR	crnn-fa-printed-96-long	CNN-BILSTM
نوع مدل	Encoder-Decoder	CRNN (Convolutional Recurrent Neural Network)	Encoder-Decoder
کاربرد اصلی	شناسایی متن دستنویس	شناسایی متن چاپی فارسی	شناسایی متن دستنویس
دقت	بالا برای متون دستنویس	بالا برای متون چاپی فارسی	بالا برای متون دستنویس
پیچیدگی محاسباتی	بالا (نیاز به منابع محاسباتی زیاد)	متوسط (کمتر از Encoder-Decoder)	بالا (نیاز به منابع محاسباتی زیاد)
نیاز به داده‌های زیاد	بله (برای آموزش دقیق)	بله، اما کمتر از TR OCR	بله (برای آموزش دقیق)
خطر بیش‌برازش	بالا (نیاز به تکنیک‌هایی مانند Early Stopping)	متوسط	بالا (نیاز به تکنیک‌هایی مانند Early Stopping)

نیازمند پیش‌پردازش دقیق و پیچیده	نیازمند پیش‌پردازش دقیق و پیچیده	نیازمند پیش‌پردازش، اما کمتر پیچیده	پیش‌پردازش داده‌ها
بالا (تنظیمات و تغییرات زیادی امکان‌پذیر است)	متوسط (تنظیمات محدودتر)	بالا (تنظیمات و تغییرات زیادی امکان‌پذیر است)	انعطاف‌پذیری
کند (به دلیل پیچیدگی مدل)	سریع‌تر از TR OCR	کند (به دلیل پیچیدگی مدل)	سرعت آموزش
چند زبانه	فارسی	چند زبانه	پشتیبانی از زبان‌ها
موجود در GitHub	مستندات کمتر	خوب مستند شده	منابع و مستندات
بله (پیشرفته)	بله (فارسی)	بله (پیشرفته)	پشتیبانی از کاراکترهای خاص
مناسب برای محیط‌های تحقیقاتی و صنعتی	مناسب برای کاربردهای مخصوص فارسی	مناسب برای محیط‌های تحقیقاتی و صنعتی	کاربرد در محیط‌های مختلف
پایدار در شرایط مختلف	پایدار برای متن‌های چاپی فارسی	پایدار در شرایط مختلف	پایداری مدل



نیاز به تخصص بالا	نیاز به تخصص متوسط	نیاز به تخصص بالا	توسعه و بهبود
بله	بله	بله	CTC Loss
با Encoder-Decoder و Convolutional و BiLSTM	با CRNN و Convolutional و RNN	Transformer	معماری مدل
بله	بله	بله	لایه های Convolutional
بله	بله	بله	لایه های LSTM

## نتیجه گیری

### 1. TR OCR :

- نقاط قوت: دقت بالا در شناسایی متون دستنویس، انعطاف پذیری بالا، قابلیت تنظیم هایپرپارامترها.
- نقاط ضعف: پیچیدگی محاسباتی بالا، نیاز به داده های زیاد، خطر بیش برآزش.

### 2. hezarai/crnn-fa-printed-96-long :

- نقاط قوت: دقت بالا برای متون چاپی فارسی، پیچیدگی کمتر نسبت به TR OCR، سرعت آموزش بالاتر.

- نقاط ضعف: محدود به زبان فارسی، انعطافپذیری کمتر، نیاز به پیشپردازش متون چاپی.

### 3. CNN-BILSTM:

- نقاط قوت: دقت بالا در شناسایی متون دستنویس، انعطافپذیری بالا، مناسب برای محیطهای تحقیقاتی.

- نقاط ضعف: پیچیدگی محاسباتی بالا، نیاز به داده‌های زیاد، خطر بیش‌برازش.

با توجه به نیازهای خاص پروژه، میتوان یکی از این سه مدل را انتخاب کرد. اگر هدف شناسایی متون دستنویس با دقت بالا و انعطافپذیری زیاد باشد، TR OCR یا مدل CNN-BILSTM مناسبتر هستند. اما اگر هدف شناسایی متون چاپی فارسی با سرعت و دقت بالا باشد، `hezarai/crnn-fa-printed-96-long` گزینه بهتری خواهد بود.

### نتیجه گیری کلی:

شده `pre train` از وزن های `hezar` نتایج بدست آمده برای مدل CNN-BILSTM استفاده کرده و در کل بهتر عمل کرده ولی مدل چون خودش روی داده انگلیسی ترین شده و ما از اول روی داده نداشته و نیاز `hezar` کردیم نتیجه بهتری از مدل `train خودمون` به داده بیشتر دارد