

گزارشکار مدل های متن باز OCR فارسی :

ای گزارش بر اساس خروجی از 12 تصویر داری متن دست نویس و غیر دست نویس انگلیسی ایجاد شده است

: CRNN

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

1. **استفاده از دیتاست ۵ برابر بزرگتر** : این مدل از یک دیتاست بسیار بزرگتر نسبت به مدل قبلی استفاده می کند که دقت و کارایی آن را بهبود می بخشد.

2. **تغییر اندازه تصویر ورودی از ۲۵۶x۶۴ به ۳۸۴x۳۲** : این تغییر باعث می شود که مدل بتواند بهتر با تصاویر ورودی کار کند و دقت بیشتری داشته باشد.

3. **افزایش حداکثر طول خروجی از ۶۴ به ۹۶** : طول نمونه ها 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 می کند که شامل اعداد و کاراکترهای ویژه می شود. این تغییرات در فایل `model_config.yaml` تعریف شده اند.

5. **مدیریت خودکار کاراکترهای LTR (چپ به راست) مانند اعداد در متن های فارسی** : این مدل توانایی مدیریت کاراکترهایی که جهت نوشتن آنها از چپ به راست است را به صورت خودکار دارد.

موارد استفاده و محدودیت ها

- ****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

hezarai/crnn-base-fa-64x256:

`{'text': 'اطمینان حاصل کردم'}`

hezarai/trocr-base-fa-v1

اصفهان حاصل کرد عم

hezarai/trocr-base-fa-v2

لحیتان حاصل کردم

hezarai/crnn-fa-printed-96-long

اطمینان حاصل کردم

تعداد لایه‌ها و غیره، عملکرد مدل را بهبود دادم.

hezarai/crnn-base-fa-64x256:

{'text': 'تغاه دیما و غیره، صلعد مد را بهوه دام.'}
hezarai/trocr-base-fa-v1

{'text': 'ا عنا مد مدوه من حیر عنه'}
hezarai/trocr-base-fa-v2

{'text': 'لحیتان حاصل کردم'} عدد چنده در غیره
hezarai/crnn-fa-printed-96-long

{'text': 'تعداد لایه‌ها و غیره، عملکرد 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 اصلا دقت خوبی ندارد



{'text': ''}



{'text': ''} - }

از هر دو نگاه ریاضیات و سیاست، مادانسکی با مسئله بسیار سختی مواجه شده بود. به عنوان یک غیر نظامی جوان که یکی از باور های اساسی ارتش در مورد جنگ سرد را به چالش می کشید، باید تصمیم گیرندگان را متقاعد می کرد با وجود اینکه هنوز قاجعه ای رخ نداده، ممکن است یکی در آینده رخ دهد. او باید قضیه را از دید بیزی به افرادی غیر متخصص توضیح می داد و به دلیل بد گمانی ارتش نسبت به توصیه های بدون حساب و کتاب افراد غیر نظامی، ناچار بود تا حد امکان از کمترین تعداد فرضیات اولیه استفاده کند. همچنین با در نظر گرفتن تعداد کم حوادث و خوشبختانه آخرالزمان هایی که تا آن موقع رخ داده بود یک مشکل آماری هم وجود داشت

'text': 'تستست'}

نقاط قوت مدل

1. **داده ست بزرگتر:** استفاده از یک مجموعه داده 5 برابر بزرگتر، قابلیت های مدل را در شناسایی و تشخیص متون مختلف بهبود می بخشد.
2. **اندازه تصویر ورودی:** تغییر اندازه تصویر ورودی به 384x32 باعث افزایش دقت در تشخیص جزئیات متون شده است.
3. **طول خروجی بیشتر:** افزایش حداکثر طول خروجی از 64 به 96، امکان پردازش متون طولانی تر را فراهم می کند.
4. **پشتیبانی از اعداد و کاراکترهای ویژه:** افزودن پشتیبانی از کاراکترهای خاص و اعداد، کاربرد مدل را در متون مختلف افزایش می دهد.
5. **مدیریت خودکار کاراکترهای LTR:** قابلیت مدیریت خودکار کاراکترهای چپ به راست مانند اعداد در متن، دقت مدل را در تشخیص متون ترکیبی بهبود می بخشد.
6. **بهینه سازی برای متون چاپی/اسکن شده:** مدل به طور خاص برای متون چاپی و اسکن شده بهینه سازی شده است که باعث افزایش دقت در تشخیص این نوع متون می شود.
7. **قابلیت تنظیم برای حوزه های دیگر:** امکان تنظیم و آموزش مجدد مدل برای کاربردهای دیگر مانند پلاک خودرو یا متون دستنویس.

نقاط ضعف مدل

1. **محدودیت در طول متن:** مدل بهترین عملکرد را در متونی با طول حداکثر 50 کاراکتر دارد و ممکن است در متون بسیار طولانی تر دقت کاهش یابد.
2. **نیاز به مدل تشخیص متن:** برای استفاده بهینه از این مدل، نیاز به یک مدل تشخیص متن اولیه است تا بخش های متنی را به صورت کلمات جداگانه استخراج کند.
3. **محدودیت در کاربردهای غیرچاپی:** اگرچه مدل قابل تنظیم برای کاربردهای دیگر است، اما بهینه سازی اولیه آن برای متون چاپی و اسکن شده ممکن است باعث کاهش دقت در متون دستنویس یا دیگر کاربردها شود.
4. **حساسیت به کیفیت تصویر:** مدل ممکن است در تصاویر با کیفیت پایین یا نویز زیاد دچار مشکل شود.

5. پیچیدگی محاسباتی: افزایش اندازه ورودی و طول خروجی ممکن است باعث افزایش پیچیدگی محاسباتی و زمان پردازش شود.

نتیجه‌گیری

این مدل با بهبودهای قابل توجه در دقت و قابلیت‌های تشخیص، یک ابزار قدرتمند برای OCR متون چاپی و اسکن شده است. با این حال، محدودیت‌هایی در طول متون و نیاز به مدل تشخیص متن اولیه وجود دارد که باید در نظر گرفته شود.

: Persian OCR using LeNet5

این مخزن شامل کدهایی برای ساخت یک سیستم تشخیص کاراکتر نوری (OCR) است که به طور خاص برای حروف الفبای فارسی طراحی شده است. سیستم از معماری LeNet5 بهره می‌برد که یک شبکه عصبی کانولوشنی کلاسیک (CNN) است و به خاطر کارایی بالا در وظایف شناسایی تصویر شناخته شده است.

دیتاست:

سیستم OCR بر روی دیتاستی شامل تقریباً 570,000 تصویر برچسب‌خورده از حروف الفبای فارسی آموزش داده شده است. این دیتاست به صورت عمومی در سایت Kaggle موجود است و می‌توانید از [اینجا] (<https://www.kaggle.com>) به آن دسترسی داشته باشید.

دقت

پس از آموزش بر روی دیتاست ارائه شده، سیستم OCR به دقت حدود 91٪ در وظایف طبقه‌بندی دست یافته است.

معماری:

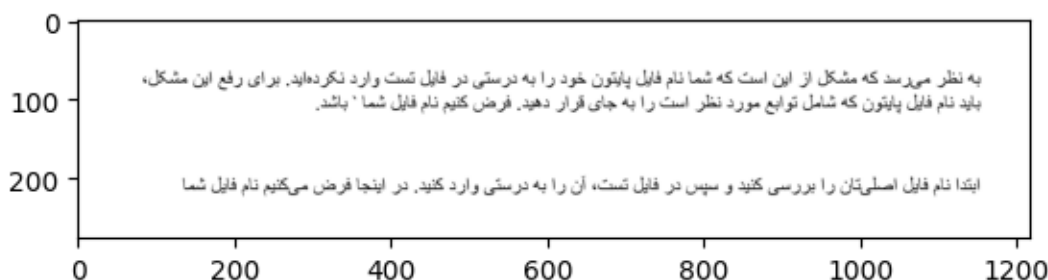
معماری LeNet5 به خاطر سادگی و کارایی در انجام وظایف طبقه‌بندی تصویر انتخاب شده است. این معماری شامل چندین لایه از جمله لایه‌های کانولوشن، پولینگ و کاملاً متصل است که یک چارچوب قوی برای شناسایی الگوها در تصاویر فراهم می‌کند.

تجربه من با مدل

شما مدل‌ها را با ساختار و معماری خودشان آموزش داده‌اید و روی داده‌های خودتان تست کرده‌اید. داده‌های این مدل بیشتر برای داده‌های تایپی در نظر گرفته شده است و نتایج نسبتاً خوبی روی داده‌های خودش به دست آمده است. اما وقتی که مدل را روی داده‌های خودتان تست کرده‌اید، نتایج کاملاً بد شده‌اند.

دلایل احتمالی برای نتایج ضعیف

1. **تفاوت در نوع داده‌ها** : اگر داده‌های شما تفاوت زیادی با داده‌های تایپی داشته باشند (مثل داده‌های دست‌نویس یا اسناد با کیفیت پایین)، مدل ممکن است نتواند به خوبی عمل کند.
 2. **پیش‌پردازش داده‌ها** : ممکن است نیاز به پیش‌پردازش متفاوتی برای داده‌های خودتان داشته باشید تا مدل بتواند الگوهای مشابهی را شناسایی کند.
 3. **نیاز به فاین‌تیون** : ممکن است نیاز باشد که مدل را روی داده‌های خودتان فاین‌تیون کنید تا عملکرد بهتری داشته باشد.
- نمونه خروجی مدل :



In [20]:

به نز مرشث اه منئدکن از ایش اتن اه نفما لام فایل بایقوزی خود را به درتت در فلیزی کیده و ارل لکرلهاین ر زای رع ایش مسئاکیگ بایل لام فایل بایقوزی اه سئامن قواع مورن فل اشد را به جایش رار دهیب ر رثرش کلیم لام فایی نفما 7 بانفد ب ابئد ا نام فلیزش اصقکازش را برشی کنیب و سبمن در فایل نثته ازش را به درتی وارد کفید ر در اینجا وطن مکنیم نام فایل تشما



In [23]:

ث ژ ا پ ا 7 ل ق ل ش 3

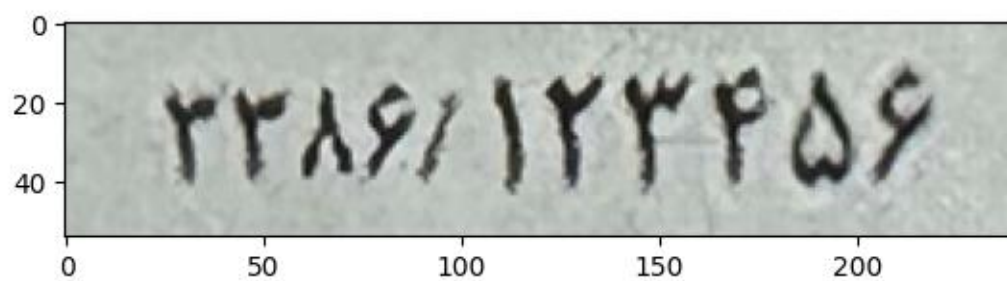
In [28]:



In [29]:

لالفیعن نلا فه ظا لف 3۱۱۱33 م ۱322 ق هطافاک

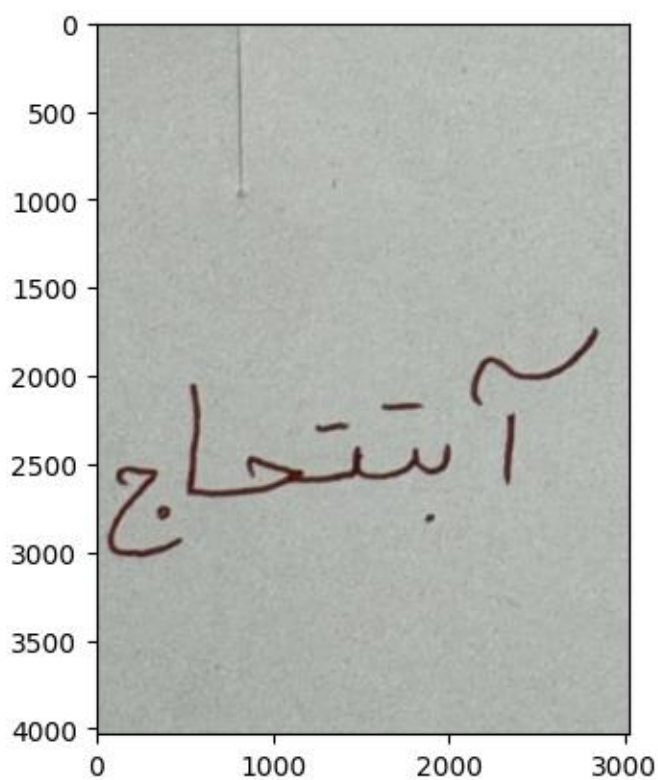
In [32]:



In [33]:

گ ظ 72 ک 72 ا 2 ف 7 ع 32 لا ۱

In [36]:



In [37]:

۱۴ گ تمود ی

In [40]:

نقاط قوت سیستم OCR

1. ****معماری LeNet5****: استفاده از معماری LeNet5، که یک شبکه عصبی پیچشی کلاسیک و موثر در وظایف شناسایی تصویر است، باعث افزایش دقت و کارایی سیستم OCR می‌شود.
2. ****داده‌ست بزرگ****: استفاده از یک مجموعه داده بزرگ با حدود 570,000 تصویر برچسبگذاری شده از حروف الفبای فارسی، به مدل کمک می‌کند تا الگوهای پیچیده را به خوبی شناسایی و یاد بگیرد.
3. ****دقت بالا****: دستیابی به دقت حدود 91% در وظایف دسته‌بندی، نشان‌دهنده عملکرد قدرتمند سیستم در تشخیص حروف الفبای فارسی است.
4. ****دسترسی به داده‌ست عمومی****: استفاده از یک مجموعه داده عمومی و در دسترس در پلتفرم Kaggle، امکان بازتولید و ارزیابی سیستم توسط محققان و توسعه‌دهندگان دیگر را فراهم می‌کند.
5. ****سهولت استفاده و پیاده‌سازی****: کد ارائه شده در مخزن به کاربران اجازه می‌دهد تا سیستم OCR را بر روی داده‌های جدید آموزش داده و ارزیابی کنند. همچنین مدل‌های پیش‌آموزش‌دیده و اسکریپت‌های مربوط به استنتاج برای استفاده آسان‌تر فراهم شده‌اند.

نقاط ضعف سیستم OCR

1. ****محدودیت به حروف الفبای فارسی****: این سیستم به طور خاص برای تشخیص حروف الفبای فارسی طراحی شده است و ممکن است در تشخیص حروف و کاراکترهای دیگر زبان‌ها عملکرد مناسبی نداشته باشد.
2. ****معماری قدیمی****: اگرچه LeNet5 یک معماری موثر و ساده است، اما معماری‌های جدیدتر مانند ResNet یا EfficientNet ممکن است دقت و کارایی بهتری ارائه دهند.
3. ****نیاز به داده‌های متنوع‌تر****: مجموعه داده استفاده شده ممکن است شامل تمامی حالت‌ها و تنوع‌های حروف الفبای فارسی نباشد. افزودن تصاویر بیشتر با تنوع‌های مختلف می‌تواند به بهبود دقت مدل کمک کند.
4. ****حساسیت به کیفیت تصویر****: مدل ممکن است در تصاویر با کیفیت پایین یا نویز زیاد دچار مشکل شود، که می‌تواند عملکرد سیستم را در شرایط واقعی کاهش دهد.

5. **پیچیدگی محاسباتی** با توجه به حجم داده ها و معماری استفاده شده، آموزش و استنتاج مدل ممکن است زمان بر و نیازمند منابع محاسباتی زیاد باشد.

نتیجه گیری

سیستم OCR ارائه شده با استفاده از معماری LeNet5 و یک مجموعه داده بزرگ، دقت بالایی در تشخیص حروف الفبای فارسی دارد. با این حال، محدودیت هایی مانند وابستگی به کیفیت تصویر و معماری قدیمی تر وجود دارد که می تواند در شرایط خاص مشکل ساز باشد. برای بهبود بیشتر، استفاده از معماری های به روزتر و داده های متنوع تر توصیه می شود.

<matplotlib.image.AxesImage at 0x16d01064a00>



'بیبیب م بهه عقیویم عوتهس ب سدیبیب هی زقو مسسس عتوتکتاگهمکو
وبشع ب بت بم'

RNN:

References

This paper helped me a lot, however my architecture is not same

• <https://arxiv.org/abs/1805.09441>

• [Pytorch Tutorial on RNNs](#)

For word segmentation using dilation see this:

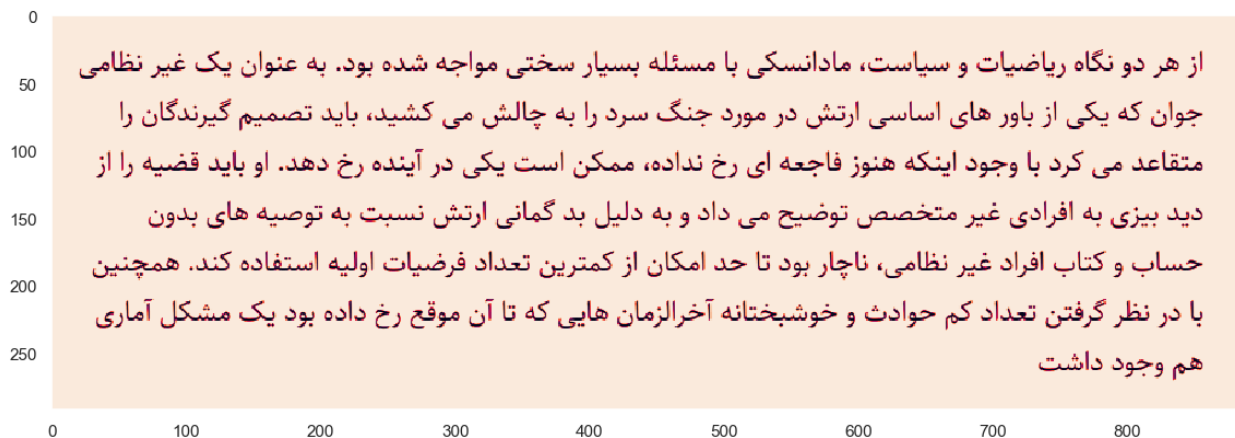
• <https://stackoverflow.com/a/10970473/4334320>

The text of the image which I used to show the final result is a translation of this book:

• The Theory That Would Not Die, Sharon McGrayne

- پیش پردازش های زیادی دارد که باید روی عکس ها انجام بشه روی عکس خودش مشکلی نداشت با تنظیم پیش پردازش ها میشه نتیجه رو بهتر کرد فقط نیاز به کار دارد یا با epoch بیشتر و دیتای بیشتر باید تست کنیم

<matplotlib.image.AxesImage at 0x16d7bb5dcd0>



In [41]:

```
''.join([image_to_word(image) for image in test_images])
```

Out[41]:

'از هر دو نگاه ریاضیات و سیاست مادانسکی با مسزلد بسیار سحننتی موا
جه شده بود به عنوان یک غیر نامی جوان کی بکی از باور های اساسب ار
تش دن موند جنگ سرد را به الش عمی کشیده باید تصمیم گیرندگان با م
نتظاعد عمی کرد با وجود اینکه هنوز فاجعه ای رنش ننادهی ممکن است ب
کی در آیندمه رش دهد او باید غضیه را از دید بیزگ به افرادی غیر مت
حه توضبت می داد و به دلیل بد گمانی ارتش نسبت به توحبه های بدون
حساب و کتاب افراد غیر نضلامی ناچار بود تا حد امکان از کمترین تعید
اد فرضیات اولیه استاده کندد ههجنبن بار در نر گرفتن تمداد کع حواد
ت و جوشبختانه آخرالزمان هرایی کی تا آن موقع رش داده بود یک مشکل
آماری هم وجود کداشت'



```
In [92]:
''.join([image_to_word(image) for image in test_images])
```

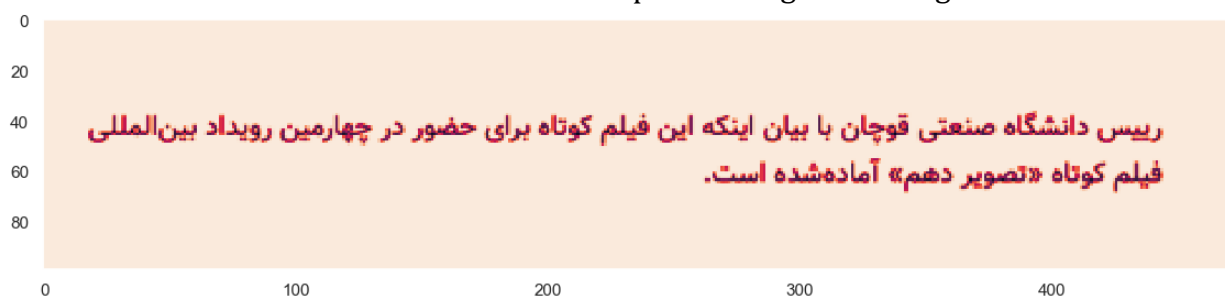
```
Out[92]:
'هرعا هو مه هققوقمس هقو س هو هو'
```

```
In [103]:
import cv2
%matplotlib inline
image_path = "C:/Users/RGS/Downloads/download.png"
```

```
img = cv2.imread(image_path, cv2.IMREAD_COLOR)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

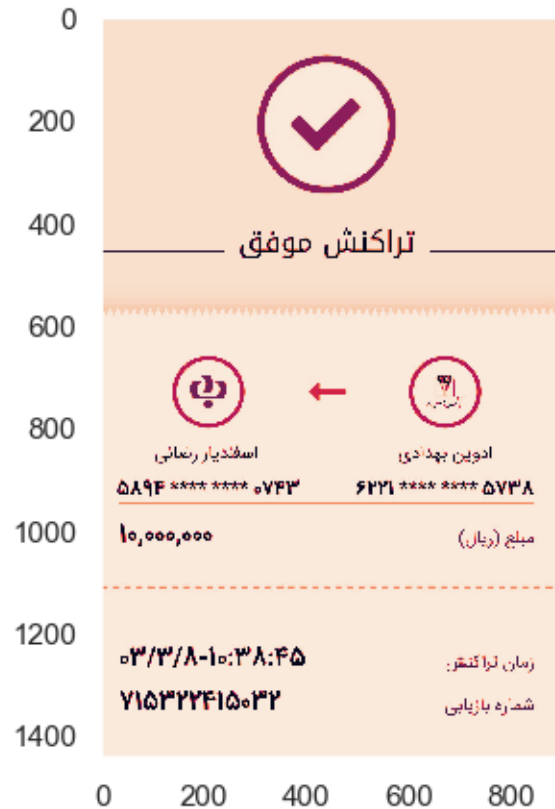
```
fig, ax = plt.subplots(figsize=(14, 5))
ax.imshow(gray, interpolation='nearest')
```

```
Out[103]:
<matplotlib.image.AxesImage at 0x16d1bc1e490>
```



```
In [104]:
''.join([image_to_word(image) for image in test_images])
```

```
Out[104]:
'مم س ایگاه کوده ولمهیر منن ن هعفغماسه سار له ار هنب پواهری مر س  
م ست روھسنریاه بالم'
```



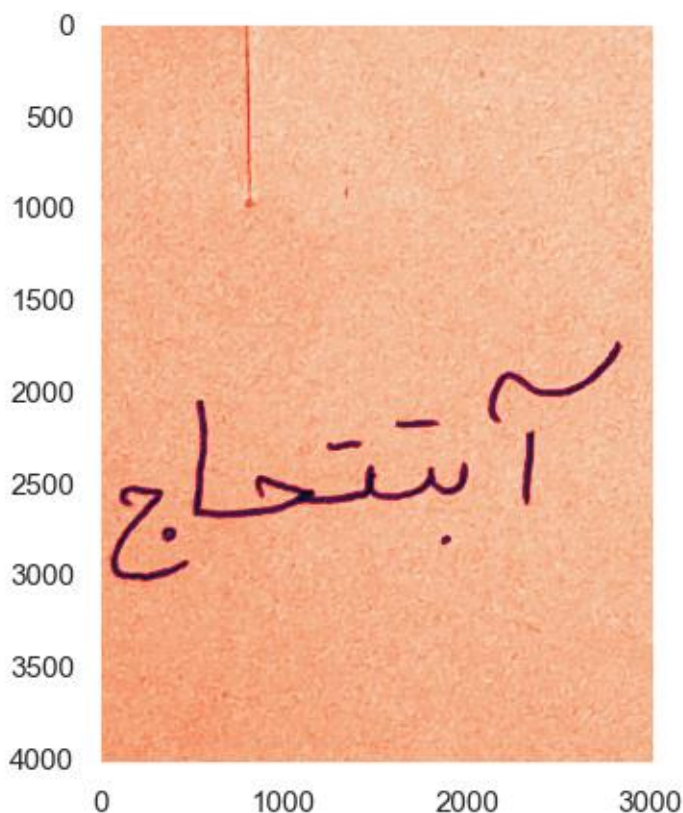
In [116]:

' '.join([image_to_word(image) for image in test_images])

Out[116]:

'عهقهقههم لآس ا ط ی م مسسم گرآآقگ العیمبیچم سس سس مهم وویم مهم قوم
 قه ا د م تاگتریل لژگذه آ ارعه ل قغکعقلرند ژل ق یر تغا قیکا گ گاه
 یگ کتقو چلگاوجه نق گقوکهوطقورگق قوویو گق قاگقعقا یا ص طئسگولطارهو
 گقگاوطزهق گا آ آگق آققهوه کمزک رزابل نچپ قار گو قع قل یقگ قار هه
 ر ق ق وگع س سر س سر س سر س سر س سر س سر س سر س سر س سر س سر س
 ر سر س سر س سر س سر س سر س سر س سر س سر س سر س سر س سر س سر س
 عه ه قق ققگق چ قوگق ب قگی گق رالققت آقا ه نهعتا غرض گا لرگتا
 گواعقا غیقیه آ گقاقلقها لقا آقهوه آ گهق تعح'

ه ه ه عم ه ه ح ه ه م و و ق قظ ه ه ه س س سه س ه مه ه مه ه ه
ه ه صو ه ه سه ه ه ع ه ه و و ف هق و و و ع و ه ه س ه ه بهعه
ص و ه و و قو غ و ه ب هق ه سب ه و هق و بب و ه و ه و ه و ه و ه و
و ق و
ه ه و
و و هق و
و
ققو س فق و و ق و قق و و ه و و و و و و و و و و و و و و و و و
و
ه بم و
ه قظ و
ب ف و ا ق و
و و هق ق و
ق ه هق و
و
ظ ف ق هق و
و و و ا ق ه و
ق و و قظ و ا ه و
و فق و ه ق و



'هق ل ييهم س غيهق آ ا سسس بس س فيس'
In [148]:



Out[149]:

'بیبیب م بهه عقیویم عوتهسس ب سدییب هی زقو مسسس عتوتکتتاگهمکو
 و بشع ب بت بم'
 نقاط قوت سیستم OCR مبتنی بر شبکه‌های عصبی بازگشتی (RNN) برای
 زبان فارسی

1. ****تشخیص در سطح کلمه****: سیستم طراحی شده برای تشخیص کلمات به جای حروف منفرد، که می‌تواند دقت و کارایی را در پردازش متون واقعی و پیوسته بهبود بخشد.
2. ****استفاده از RNN****: شبکه‌های عصبی بازگشتی (RNN) به ویژه برای پردازش داده‌های ترتیبی مانند متن مناسب هستند. این معماری قادر است وابستگی‌های طولانی‌مدت را در داده‌ها شناسایی کند.
3. ****کاربرد گسترده‌تر****: برخلاف سیستم‌های حروف‌محور، این مدل می‌تواند کلمات کامل را تشخیص داده و به همین دلیل برای کاربردهای واقعی‌تر مانند خواندن اسناد و متون چاپی یا دست‌نویس مناسب‌تر است.
4. ****پشتیبانی از زبان فارسی****: تمرکز بر زبان فارسی به این معنی است که مدل می‌تواند ویژگی‌های خاص و منحصر به فرد این زبان را بهتر شناسایی کند.
5. ****پتانسیل برای بهبود دقت با داده‌های بیشتر****: استفاده از داده‌های بیشتر و متنوع‌تر می‌تواند بهبود قابل توجهی در دقت و کارایی مدل ایجاد کند.

نقاط ضعف سیستم OCR مبتنی بر RNN برای زبان فارسی

1. ****پیچیدگی محاسباتی بالا****: شبکه‌های عصبی بازگشتی معمولاً نیاز به منابع محاسباتی زیادی دارند و ممکن است آموزش و استنتاج آنها زمان‌بر باشد.
2. ****چالش در مدیریت طول‌های متفاوت کلمات****: کلمات در زبان فارسی می‌توانند طول‌های متفاوتی داشته باشند که ممکن است چالش‌برانگیز باشد. این مشکل در شبکه‌های RNN قابل مدیریت است اما ممکن است بهینه‌سازی‌های خاصی نیاز داشته باشد.
3. ****حساسیت به کیفیت تصویر****: مدل‌های OCR معمولاً به کیفیت تصویر ورودی حساس هستند و نویز یا کیفیت پایین تصویر می‌تواند دقت تشخیص را کاهش دهد.
4. ****محدودیت در داده‌های آموزشی****: اگر مجموعه داده‌های آموزشی شامل تنوع کافی در کلمات و شرایط مختلف نباشد، مدل ممکن است در شرایط واقعی عملکرد ضعیفی داشته باشد.
5. ****پیچیدگی در تنظیمات و پیاده‌سازی****: استفاده از RNN به تنظیمات دقیق و پیچیده‌تری نیاز دارد که ممکن است برای برخی کاربران چالش‌برانگیز باشد.

نتیجه‌گیری

استفاده از RNN برای تشخیص کلمات در زبان فارسی می‌تواند مزایای قابل توجهی در دقت و کارایی داشته باشد، به ویژه در کاربردهای واقعی و متون پیوسته. با این حال، پیچیدگی محاسباتی و حساسیت به کیفیت تصویر نیز از چالش‌های اصلی این سیستم هستند. افزایش داده‌های آموزشی و بهینه‌سازی معماری می‌تواند به بهبود عملکرد سیستم کمک کند.

نتیجه گیری کلی :

در کل مدل CRNN بهتر از مدل‌های دیگر عمل کرده و نتایج خیلی بهتر از بقیه مدل‌هاست شاخص‌ترین نقطه ضعفی که میشه گفت این هست که همیشه برای متن‌های طولانی از آن استفاده کرد و باید با یک اسکریپت کلمه‌ها و جمله‌ها رو به هم وصل کرد تا بشه یک خروجی مطمئن و دقیق داشته باشیم

تشخیص لوگوهای بانک با استفاده از YOLOv8

****1. آماده سازی داده ها****

****جمع آوری داده ها**:**

من تصاویر مختلفی از لوگوهای بانکهای مختلف جمع آوری کردم. این تصاویر را از منابع آنلاین مانند وبسایتها و رسانه های اجتماعی به دست آوردم.

توجه کردم که مجموعه داده شامل تنوع بالایی از شرایط نوری، زوایا و پس زمینه ها باشد.

****برچسب گذاری داده ها**:**

برای برچسب گذاری داده ها از ابزارهایی مانند roboflow استفاده کردم. هر لوگو را با استفاده از جعبه های محدودکننده (bounding boxes) برچسب گذاری کردم و کلاس مربوطه (نام بانک) را به آن اختصاص دادم.

****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**:**

YOLOv8 را نصب کردم. از `ultralytics` استفاده کردم که شامل نسخه های مختلف YOLO است.

```
```bash
```

```
pip install ultralytics
```

```
```
```

****تنظیم محیط کاری**:**

یک ساختار پوشه مناسب برای پروژه خود ایجاد کردم. نمونه ای از ساختار پوشه:

```

...
└── data
    ├── train
    ├── val
    └── test
└── labels
    ├── train
    ├── val
    └── test
└── models
└── scripts
└── yolov8_config.yaml
...
```

train مدل YOLOv8

****تنظیم فایل پیکربندی**:**

یک فایل پیکربندی برای YOLOv8 ایجاد کردم که مسیر داده ها و تنظیمات مدل را مشخص کند. نمونه ای از: `yolov8_config.yaml`

```

```yaml
train: data/train
val: data/val
test: data/test
```



**nc: 5** تعداد کلاسها (تعداد بانکها)

**names: ['bank1', 'bank2', 'bank3', 'bank4', 'bank5']** نام کلاسها

...

**\*\*شروع train**

از اسکریپت‌های `ultralytics` برای آموزش مدل استفاده کردم:

**from ultralytics import YOLO**

ایجاد مدل

**model = YOLO('yolov8n.pt')** یا مدل دیگر مانند **yolov8s.pt**

آموزش مدل

**model.train(data='yolov8\_config.yaml', epochs=50, imgsz=640)**

**\*\*4. ارزیابی و تست مدل\*\***

**\*\*ارزیابی مدل\*\*:**

پس از آموزش، مدل را روی مجموعه داده اعتبارسنجی و تست ارزیابی کردم تا دقت، بازخوانی و سایر معیارها را بررسی کنم.

**results = model.val()**

**print(results)**

...

**\*\*تست مدل\*\*:**

مدل را روی تصاویر جدید تست کردم و عملکرد آن را بررسی کردم.

```
results = model.predict('path_to_test_image.jpg')
results.show()
'''
```

**\*\*5. بهبود و بهینه‌سازی\*\***

**\*\*تنظیم هایپرپارامترها\*\*:**

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

**\*\*افزایش داده‌ها\*\*:**

از تکنیک‌های افزایش داده‌ها (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 پیش‌آموزش یافته داشتم، از آن برای تنظیم دقیق مدل خود استفاده کردم.

**\*\*6. استفاده در دنیای واقعی\*\***

**\*\*پیاده‌سازی مدل در یک برنامه\*\*:**

مدل آموزش دیده را در یک برنامه وب یا موبایل پیاده‌سازی کردم تا بتوانم لوگوهای بانک‌ها را در تصاویر واقعی تشخیص دهم.

**\*\*به‌روزرسانی دوره‌ای\*\*:**

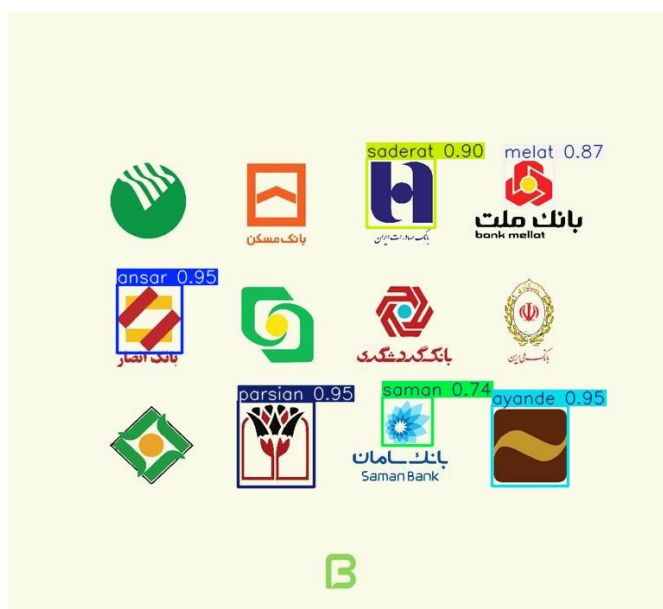
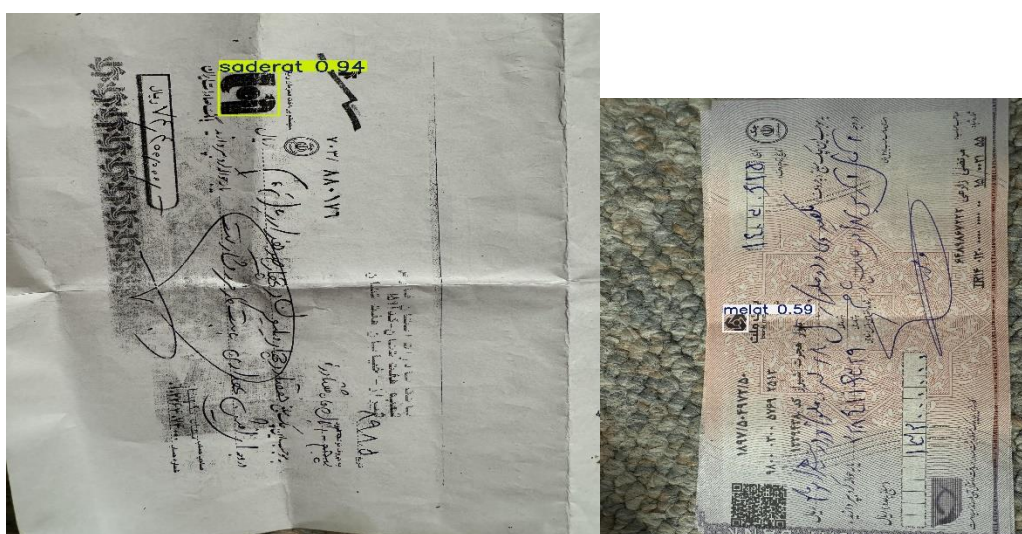
مجموعه داده خود را به‌روزرسانی کردم و مدل را به صورت دوره‌ای مجدداً آموزش دادم تا عملکرد آن بهبود یابد.

**\*\*گزارش‌دهی نتایج\*\*:**

نتایج نهایی را گزارش کردم و نمودارها و جداول مربوط به عملکرد مدل را تهیه کردم.

با پیروی از این نقشه راه، توانستم مدل YOLOv8 را برای تشخیص لوگوهای بانکهای مختلف آموزش داده و استفاده کنم.

نتایج مدل :



## تشخیص قسمت‌های مختلف چک با YOLOv8

### **\*\*1. آماده‌سازی داده‌ها\*\***

#### **\*\*جمع‌آوری داده‌ها\*\*:**

من تصاویر مختلفی از چک‌ها جمع‌آوری کردم. این تصاویر شامل انواع چک‌های مختلف با قالب‌ها و طرح‌های متنوع بودند. اطمینان حاصل کردم که مجموعه داده شامل تنوع بالایی از شرایط نوری، زوایا و پس‌زمینه‌ها باشد.

#### **\*\*برچسب‌گذاری داده‌ها\*\*:**

برای برچسب‌گذاری داده‌ها از ابزارهایی مانند LabelImg یا VGG Image Annotator (VIA) استفاده کردم. هر قسمت از چک را با استفاده از جعبه‌های محدودکننده (bounding boxes) برچسب‌گذاری کردم و کلاس مربوطه (مبلغ چک، کد ملی شخص، QR\_code، وجه چک، تاریخ چک، id\_check) را به آن اختصاص دادم.

#### **\*\*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\*\*:**

YOLOv8 را نصب کردم. از `ultralytics` استفاده کردم که شامل نسخه‌های مختلف YOLO است.

```bash

pip install ultralytics

...

****تنظیم محیط کاری**:**

یک ساختار پوشه مناسب برای پروژه خود ایجاد کردم. نمونه‌ای از ساختار پوشه:

...

└─ data

 └─ train

 └─ val

 └─ test

└─ labels

 └─ train

 └─ val

 └─ test

└─ models

└─ scripts

└─ yolov8_config.yaml

...

****3. آموزش مدل YOLOv8****

****تنظیم فایل پیکربندی**:**

یک فایل پیکربندی برای YOLOv8 ایجاد کردم که مسیر داده‌ها و تنظیمات مدل را مشخص کند. نمونه‌ای از: `yolov8_config.yaml`

```yaml

**train: data/train**

**val: data/val**

**test: data/test**

**nc: 6** تعداد کلاس‌ها (6 قسمت مختلف چک)

**names: ['amount', 'national\_id', 'QR\_code', 'check\_amount', 'check\_date', 'check\_id']**  
نام کلاس‌ها

...

**\*\*5. بهبود و بهینه‌سازی\*\***

**\*\*تنظیم هایپرپارامترها\*\*:**

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

**\*\*افزایش داده‌ها\*\*:**

از تکنیک‌های افزایش داده‌ها (**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 پیش‌آموزش یافته داشتم، از آن برای تنظیم دقیق مدل خود استفاده کردم.

**\*\*6. استفاده در دنیای واقعی\*\***

**\*\*پیاده‌سازی مدل در یک برنامه\*\*:**

مدل آموزش دیده را در یک برنامه وب یا موبایل پیاده‌سازی کردم تا بتوانم قسمت‌های مختلف چک را در تصاویر واقعی تشخیص دهم.

**\*\*به روزرسانی دوره ای\*\*:**

مجموعه داده خود را به روزرسانی کردم و مدل را به صورت دوره ای مجدداً آموزش دادم تا عملکرد آن بهبود یابد.

**\*\*گزارش دهی نتایج\*\*:**

نتایج نهایی را گزارش کردم و نمودارها و جداول مربوط به عملکرد مدل را تهیه کردم.

**: finding open source canscner**

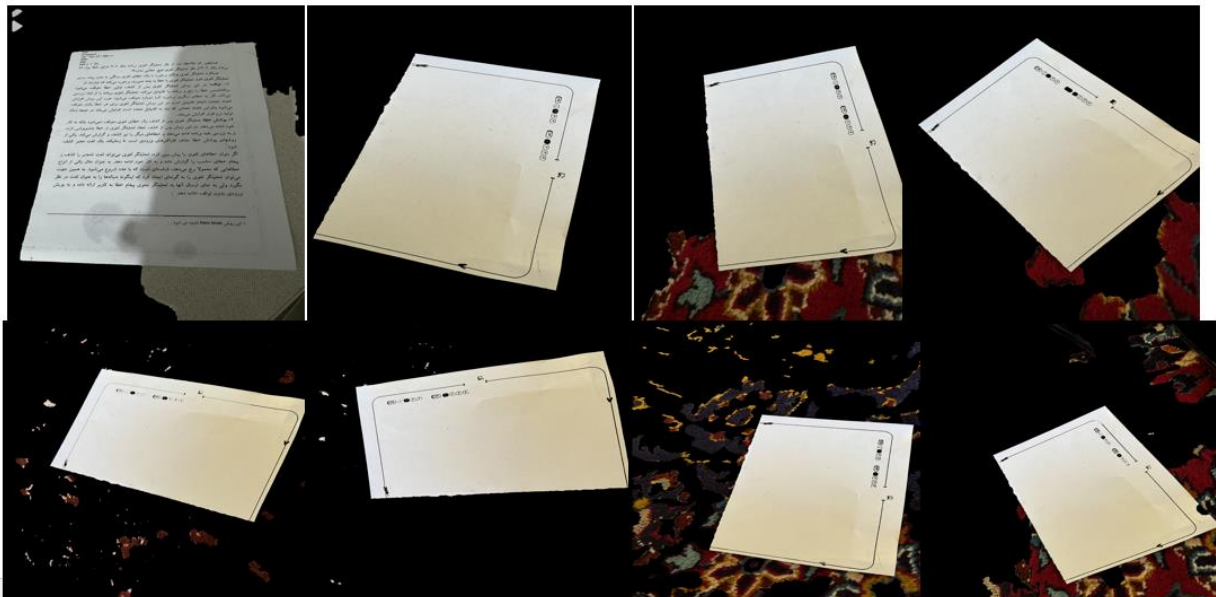
این مدل ها برای حذف بک گراند هستند که آقای مهندس حامی انتخاب کردند من با دادن حدود 10 تا عکس های مختلف این مدل هاروتست کردم

Easy OCR Result:





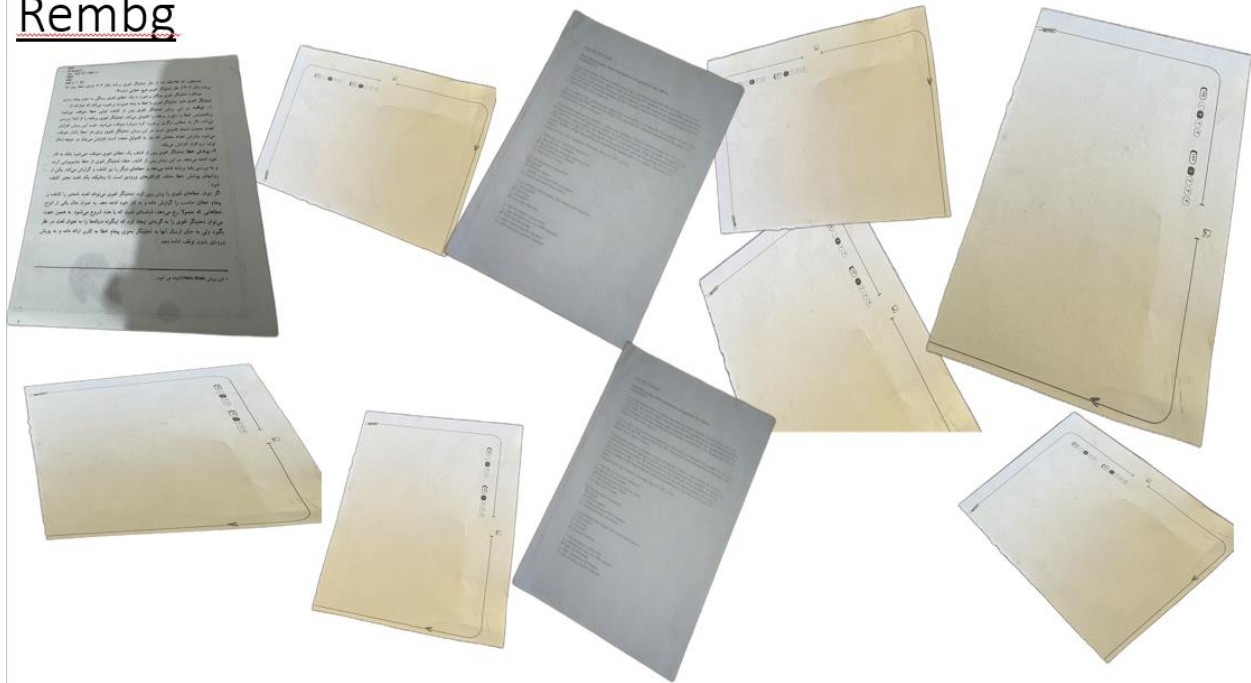
## grabcut-opencv Result:



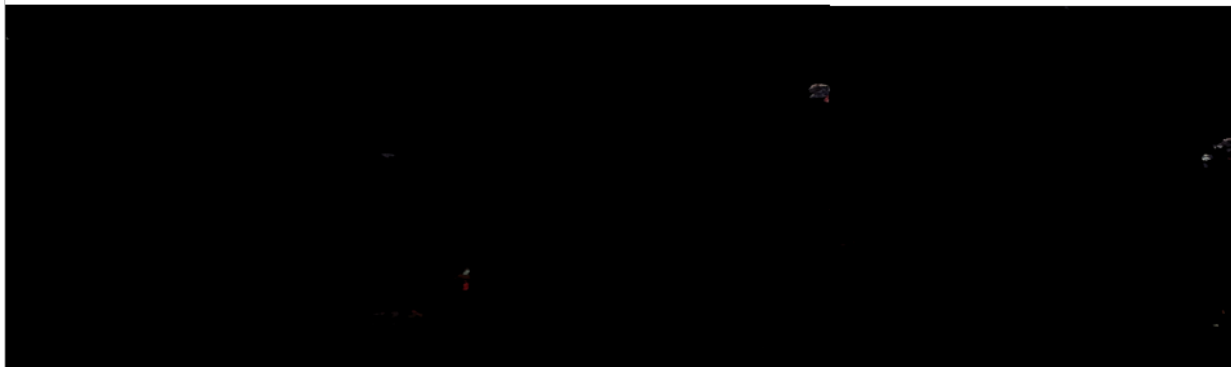
## Mean-shift



Rembg



BgRem-DeeplabV3



نتیجه گیری :

بهترین مدل برای حذف بک گراند **Rembg** هست که بهترین نتیجه رو در بین مدل های موجود داشته فقط باید یک پیش پردازش دیگه روی عکس خروجی انجام بشه که عکس های که به صورت کج گرفته یا زاویه بد گرفته شده صاف و بالانس بشه

مطالعه در مورد مدل های فارسی انگلیسی فعلی برای ocr در سطح word :

### معرفی مدل TrOCR

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

### توضیحات مدل TrOCR

- \*\*Encoder\*\* تصویر: \*\*Encoder\*\* تصویر از وزن های پیش آموزش داده شده مدل BEiT بهره می برد.

- \*\*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 استخراج‌شده 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 پیش‌آموزش‌داده 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

### درباره مجموعه داده

مجموعه داده های فرم های دستنویس IAM شامل فرم هایی از متن انگلیسی دستنویس است که می تواند برای آموزش و آزمایش تشخیص دهنده های متن دستنویس و انجام آزمایش های شناسایی و تأیید نویسنده استفاده شود.

### محتوا

مجموعه داده شامل فرم های کاملی از متن دستنویس بدون محدودیت است که با وضوح dpi300 اسکن شده و به عنوان تصاویر PNG با 256 سطح خاکستری ذخیره شده اند. فرم ها به دایرکتوری های جداگانه تقسیم می شوند به طوری که همه فرم ها در هر دایرکتوری توسط یک شخص نوشته می شوند.

### نتیجه گیری

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

### چکیده مقاله

تشخیص متن یک مسئله پژوهشی طولانی مدت برای دیجیتالی سازی اسناد است. روش های 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 پس پردازش برای بهبود دقت کلی نیاز است. در این مقاله، ما یک روش تشخیص متن انتها به انتها با استفاده از مدل های ترنسفورمر پیش آموزش داده شده تصویری و متنی، به نام TrOCR، پیشنهاد می دهیم که از معماری ترنسفورمر برای هر دو درک تصویر و تولید متن کلمه ای بهره می برد. مدل TrOCR ساده اما موثر است و می تواند با داده های مصنوعی بزرگ مقیاس پیش آموزش داده شده و با مجموعه داده های

برچسبگذاری شده انسانی تنظیم دقیق شود. آزمایشها نشان می‌دهند که مدل TrOCR از مدل‌های فعلی در تشخیص متون چاپی، دستنویس و صحنه پیشی می‌گیرد.

گزارش پروژه شناسایی متن دستنویس با استفاده از OCR و TensorFlow (CNN-RNN) :

#### 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 برچسب‌ها 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.

مقایسه مدل‌ها: TR OCR ، hezarai/crnn-fa-printed-96-long و مدل 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 | hezarai/crnn-fa-printed-96-long | مدل CNN-BILSTM |
|-------|--------|---------------------------------|----------------|
|       |        |                                 |                |

|                          |                                                      |                                                        |                                                      |
|--------------------------|------------------------------------------------------|--------------------------------------------------------|------------------------------------------------------|
| نوع مدل                  | Encoder-Decoder                                      | CRNN<br>(Convolutional<br>Recurrent Neural<br>Network) | Encoder-Decoder                                      |
| کاربرد اصلی              | شناسایی متن<br>دستنویس                               | شناسایی متن<br>چاپی فارسی                              | شناسایی متن<br>دستنویس                               |
| دقت                      | بالا برای متون<br>دستنویس                            | بالا برای متون<br>چاپی فارسی                           | بالا برای متون<br>دستنویس                            |
| پیچیدگی<br>محاسباتی      | بالا (نیاز به<br>منابع محاسباتی<br>زیاد)             | متوسط (کمتر<br>از Encoder-<br>Decoder)                 | بالا (نیاز به<br>منابع محاسباتی<br>زیاد)             |
| نیاز به<br>داده های زیاد | بله (برای آموزش<br>دقیق)                             | بله، اما کمتر<br>از TR OCR                             | بله (برای آموزش<br>دقیق)                             |
| خطر بیش‌برازش            | بالا (نیاز به<br>تکنیک‌هایی مانند<br>Early Stopping) | متوسط                                                  | بالا (نیاز به<br>تکنیک‌هایی مانند<br>Early Stopping) |
| پیش‌پردازش<br>داده ها    | نیازمند<br>پیش‌پردازش دقیق<br>و پیچیده               | نیازمند<br>پیش‌پردازش،<br>اما کمتر<br>پیچیده           | نیازمند<br>پیش‌پردازش دقیق<br>و پیچیده               |



|                            |                                               |                                  |                                               |
|----------------------------|-----------------------------------------------|----------------------------------|-----------------------------------------------|
| انعطاف‌پذیری               | بالا (تنظیمات و تغییرات زیادی امکان‌پذیر است) | متوسط (تنظیمات محدودتر)          | بالا (تنظیمات و تغییرات زیادی امکان‌پذیر است) |
| سرعت آموزش                 | کند (به دلیل پیچیدگی مدل)                     | سریع‌تر از TR OCR                | کند (به دلیل پیچیدگی مدل)                     |
| پشتیبانی از زبان‌ها        | چند زبانه                                     | فارسی                            | چند زبانه                                     |
| منابع و مستندات            | خوب مستند شده                                 | مستندات کمتر                     | موجود در GitHub                               |
| پشتیبانی از کاراکترهای خاص | بله (پیشرفته)                                 | بله (فارسی)                      | بله (پیشرفته)                                 |
| کاربرد در محیط‌های مختلف   | مناسب برای محیط‌های تحقیقاتی و صنعتی          | مناسب برای کاربردهای مخصوص فارسی | مناسب برای محیط‌های تحقیقاتی و صنعتی          |
| پایداری مدل                | پایدار در شرایط مختلف                         | پایدار برای متن‌های چاپی فارسی   | پایدار در شرایط مختلف                         |
| توسعه و بهبود              | نیاز به تخصص بالا                             | نیاز به تخصص متوسط               | نیاز به تخصص بالا                             |
| CTC Loss                   | بله                                           | بله                              | بله                                           |

| معماری مدل             | Transformer | با CRNN و Convolutional RNN | با Encoder-Decoder Convolutional و BiLSTM |
|------------------------|-------------|-----------------------------|-------------------------------------------|
| لایه های Convolutional | بله         | بله                         | بله                                       |
| لایه های LSTM          | بله         | بله                         | بله                                       |

### 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 پرازش.

#### 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. **\*\*مدل\*\***:

- **\*\*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.

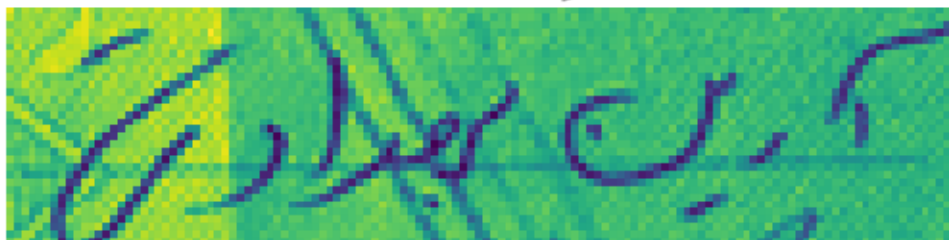
با توجه به نیازهای خاص پروژه، میتوان یکی از این سه مدل را انتخاب کرد. اگر هدف شناسایی متون دستنویس با دقت بالا و انعطافپذیری زیاد باشد، **\*\*TR OCR\*\*** یا **\*\*مدل CNN-BILSTM\*\*** مناسبتر هستند. اما اگر هدف شناسایی متون چاپی فارسی با سرعت و دقت بالا باشد، **\*\*hezarai/crnn-fa-printed-96-long\*\***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

نمونه های هر مدل :

Hezar:

['text': 'اری هدادی']

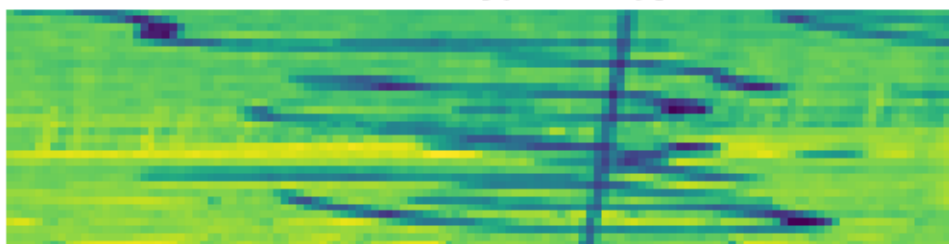
Prediction: [{ 'text': 'یاداده یرا' }]



<class 'list'>

[{ 'text': '' }]

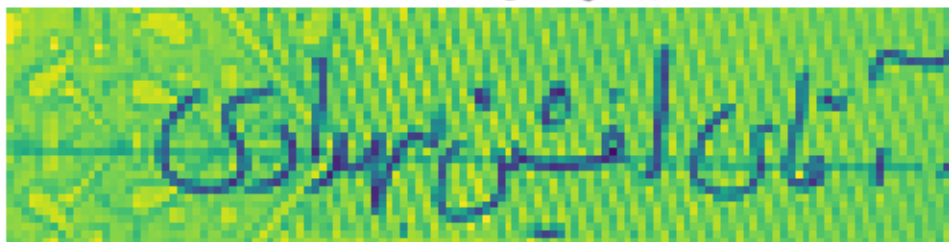
Prediction: [{ 'text': '' }]



<class 'list'>

[{ 'text': 'ای اشنبهدای' }]

Prediction: [{ 'text': 'یاده بنش ایا' }]



<class 'list'>

[{ 'text': 'طراحی الگوریتم' }]

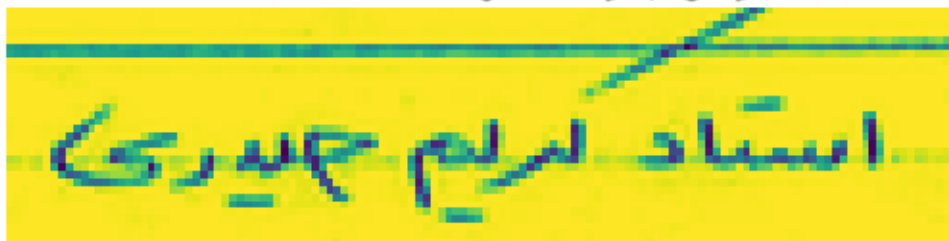
Prediction: [{'text': 'مٲی روگلا یحارط'}]

A photograph of a handwritten note on yellow paper. The text is written in Urdu in black ink. It reads 'ملا یحیٰ الہوریہ' (Mulla Yahiya Ali). The handwriting is cursive and slightly slanted.

<class 'list'>

['text': 'استاد کریم حیدری']

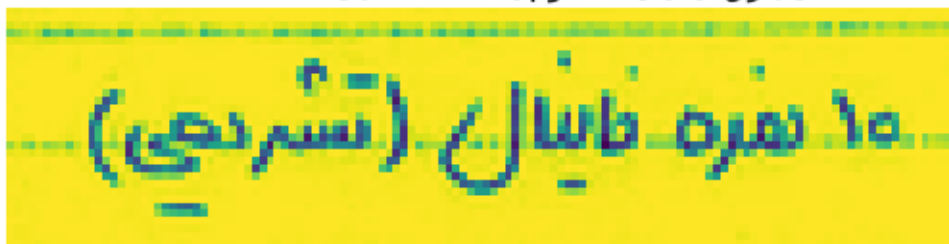
Prediction: [{'text': 'ی ردی ح م ی ر ک د ا ت س ا'}]

A photograph of a handwritten note on yellow paper. The text is written in Urdu in black ink. It reads 'استاد کریم حیدری' (Ustad Karim Haidri). The handwriting is cursive and slightly slanted.

<class 'list'>

['text': 'آ نموه فاینال قر']

Prediction: [{'text': 'رق لان یاف هوم ن آء'}]

A photograph of a handwritten note on yellow paper. The text is written in Urdu in black ink. It reads '۱۵ نمبرہ فاینال (تشریحی)' (15 Number Final (Descriptive)). The handwriting is cursive and slightly slanted.

<class 'list'>

['text': 'انواع ساختمان']

Prediction: [{'text': 'نامتخاس غاونا'}]

انواع ساختار

<class 'list'>

{'text': 'جدول درهم ساز'}

Prediction: [{'text': 'زاس مہرد لودج'}]

جدول درهم ساز

<class 'list'>

{'text': 'بہترین الگوریتم'}

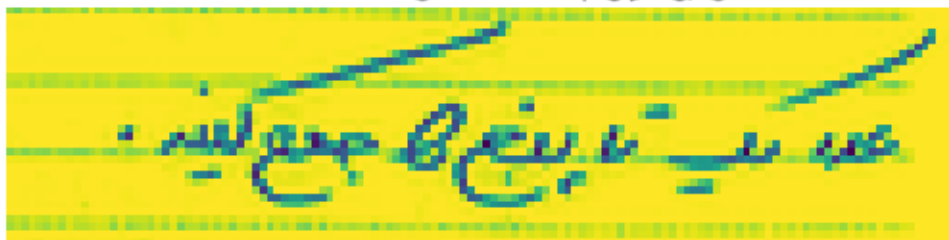
Prediction: [{'text': 'متریوگلا نیرتہب'}]

بہترین الگوریتم

<class 'list'>

{'text': 'عدد یک تاپج را جمع کنید'}

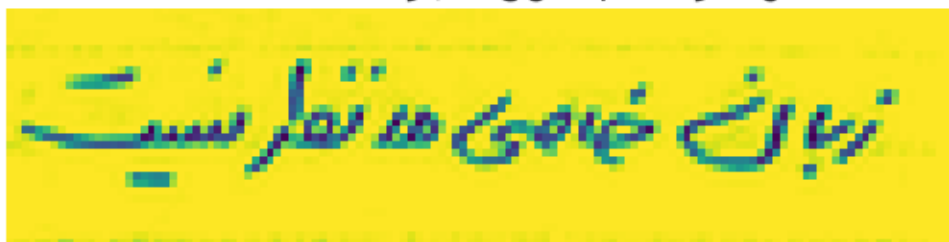
Prediction: [{'text': 'دینک عمج ارج پات کی ددع'}]



<class 'list'>

[{'text': 'زبان خاصی مد نظر نیست'}]

Prediction: [{'text': 'تسیرنرطن دم یصاخ نابز'}]



<class 'list'>

[{'text': 'ارم آن را ر می گردانند'}]

Prediction: [{'text': 'دنادرگ یم رارن آمرا'}]



<class 'list'>

[{'text': 'تاع بازگشتی بنوسید که'}]

Prediction: [{'text': 'هک دیسونب یتشگزاب عات'}]

A sample of handwritten Persian text in blue ink on a yellow background. The text is written in a cursive style and matches the prediction above it.

<class 'list'>

[{'text': 'اقواع رون غابرفقایه گویتم ها'}]

Prediction: [{'text': 'اه متیوگ هیاق فرباغ نورع اوقا'}]

A sample of handwritten Persian text in blue ink on a yellow background. The text is written in a cursive style and matches the prediction above it.

<class 'list'>

[{'text': 'به سخت افزار بتی درد'}]

Prediction: [{'text': 'درد یتب رازفا تخس هب'}]

A sample of handwritten Persian text in blue ink on a yellow background. The text is written in a cursive style and matches the prediction above it.

<class 'list'>

[{'text': 'چه کوچک تاثیر ندارد'}]



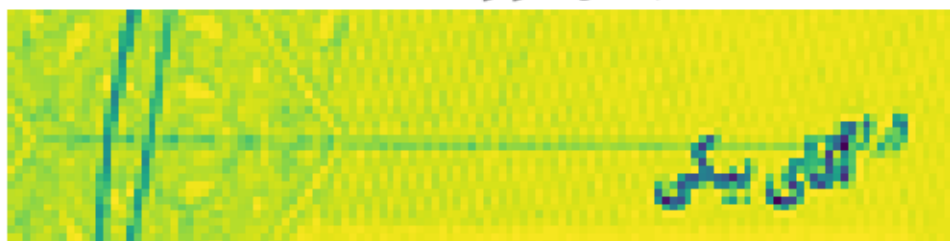
Prediction: [{'text': 'درادن ریثات کچوک هچ'}]



<class 'list'>

['text': 'ردرهگتای بیگی']

Prediction: [{'text': 'یگیب یاقگهردر'}]



<class 'list'>

['text': 'در و جنم ابواهی ابت تسوه']

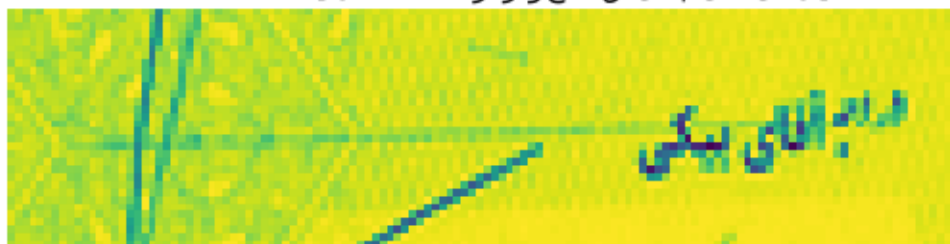
Prediction: [{'text': 'هوست تبا یح او با من جوردا'}]



<class 'list'>

['text': 'دردر وجه آقای بیگی']

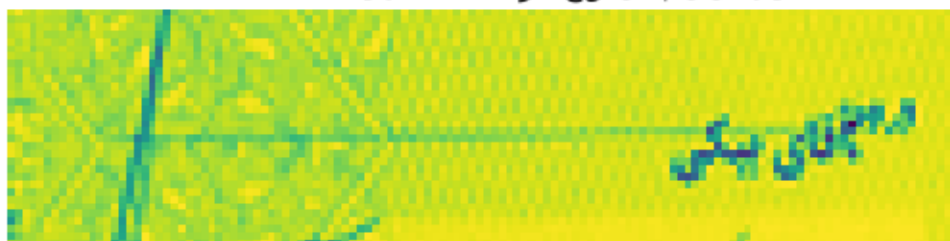
Prediction: [{'text': 'یگیب یاقا هج و ردرد'}]



<class 'list'>

{'text': 'در هجای بیگی'}

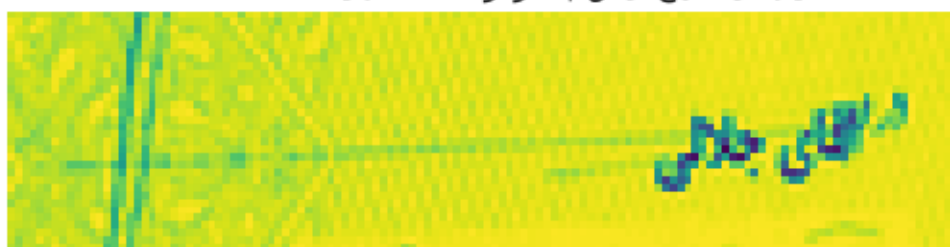
Prediction: [{'text': 'ییب یاقچه رد'}]



<class 'list'>

{'text': 'ردر هجای جلگی'}

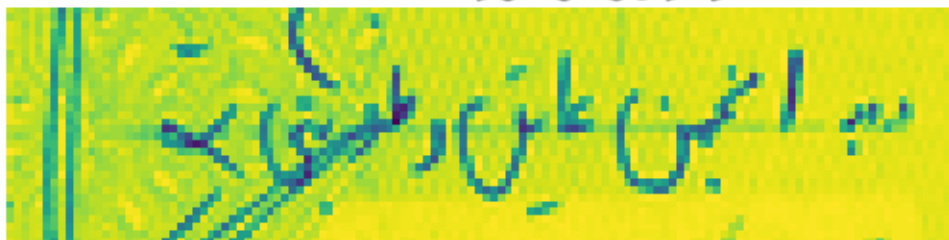
Prediction: [{'text': 'یگلج یاقبه ردرد'}]



<class 'list'>

{'text': 'درجه اجن عاق رطوتی'}

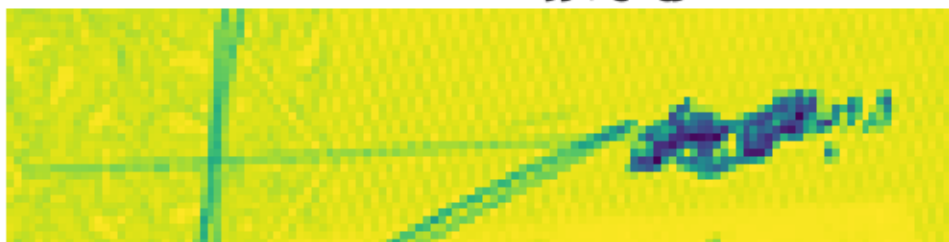
Prediction: [{'text': 'یت و طر قاع نجا هج رد'}]



<class 'list'>

[{'text': 'رو ب حیچو'}]

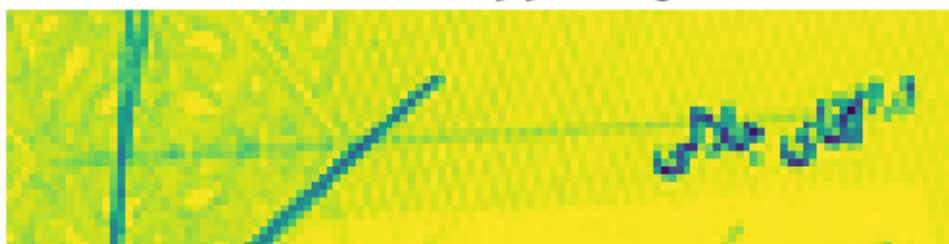
Prediction: [{'text': 'وچ یح ب وړ'}]



<class 'list'>

[{'text': 'در دره هگای جلاگی'}]

Prediction: [{'text': 'یگال ج ی اگه هردرد'}]



<class 'list'>

[{'text': 'در وجه ص'}]

Prediction: [{ 'text': 'ص هج و رد' }]



<class 'list'>

{ 'text': 'در وجه حاج حسین صفری' }

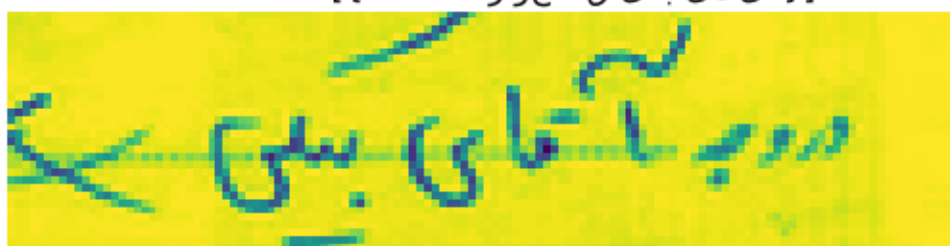
Prediction: [{ 'text': 'ص هج و رد' }]



<class 'list'>

{ 'text': 'در وجه آقای بیگی' }

Prediction: [{ 'text': 'ص هج و رد' }]



<class 'list'>

{ 'text': 'در وجه حاج ادی یونی' }

Prediction: [{'text': 'ی‌نوی یدا جاح هج و رد'}]



<class 'list'>

['text': 'در وجه آقای بیگی']

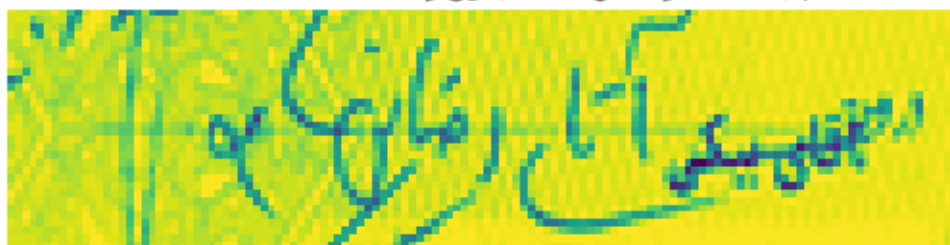
Prediction: [{'text': 'ی‌گیب یاق آ هج و رد'}]



<class 'list'>

['text': 'رجقبگی اسی رفاییم']

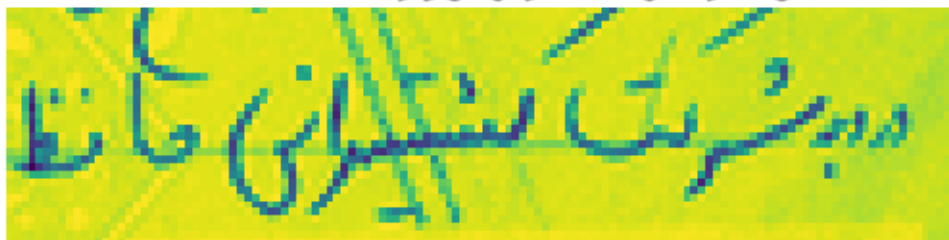
Prediction: [{'text': 'م‌بی‌ی‌افر ی‌سا ی‌گ‌ب‌ق‌ج‌ر'}]



<class 'list'>

['text': 'در وجه شرکت کشترا ی حافظ']

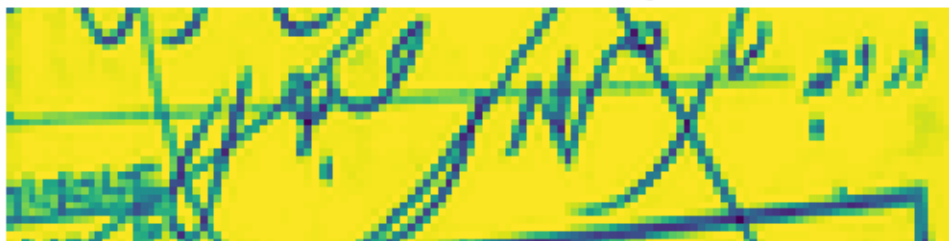
Prediction: [{'text': 'طافاح یارتشک تکرش هجو رد'}]



<class 'list'>

[{'text': 'هلا اگمهسیی'}]

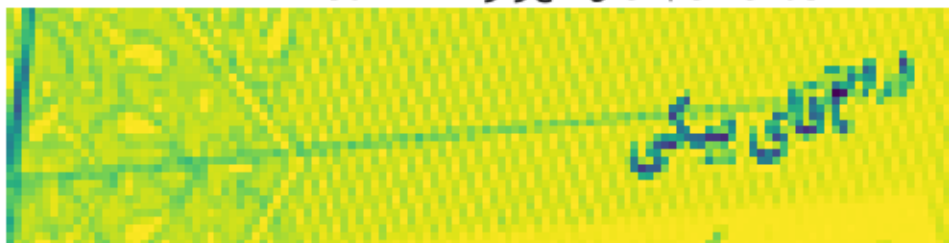
Prediction: [{'text': 'ی ی س هم گ ال ه'}]



<class 'list'>

[{'text': 'در درجه آقای بیگی'}]

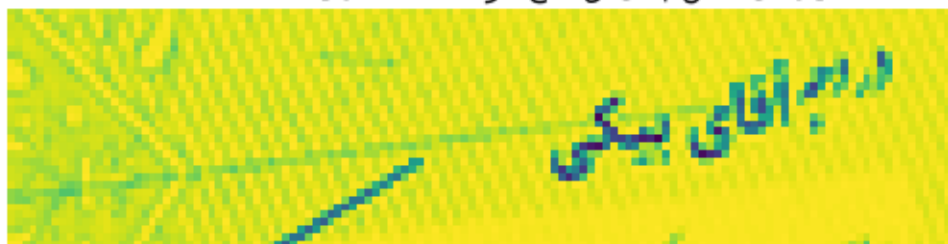
Prediction: [{'text': 'ی گی ب ی اق آ ه ج رد رد'}]



<class 'list'>

[{'text': 'در درجه آقای بسگی'}]

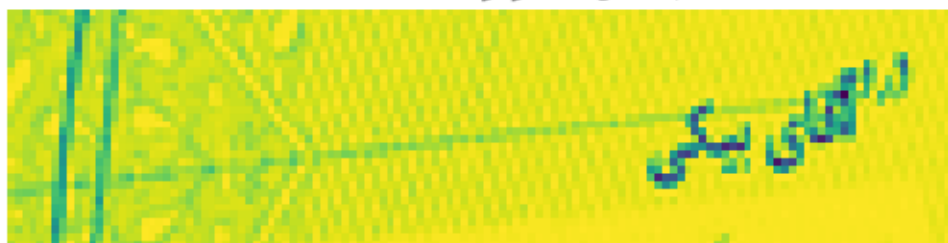
Prediction: [{'text': 'یگ سب یاقا هج رد'}]



<class 'list'>

['text': 'در دهر ه گقای بیگی']

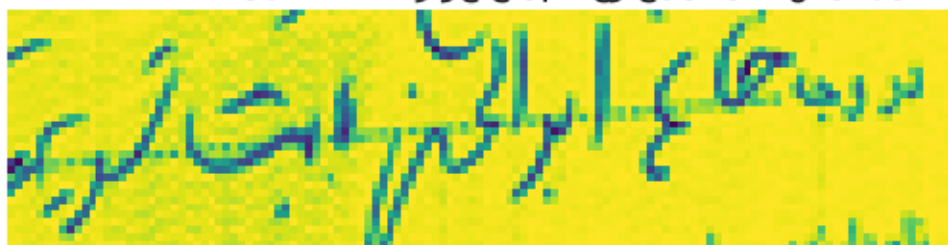
Prediction: [{'text': 'یگی ب یاقا ه رد'}]



<class 'list'>

['text': 'در دهر وچ خانم اجواحنی ایت سی']

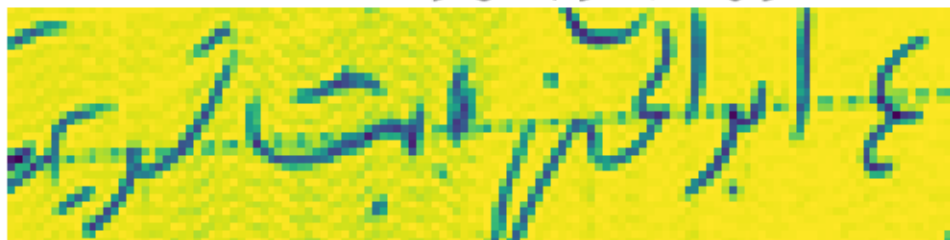
Prediction: [{'text': 'یس تیای نجاوچا مناخ چوردد'}]



<class 'list'>

['text': 'ر جه نم و نابته سوی']

Prediction: [{'text': 'یوس تهبان و من هج ر'}]



<class 'list'>

['text': 'در وجه حج اب یونی']

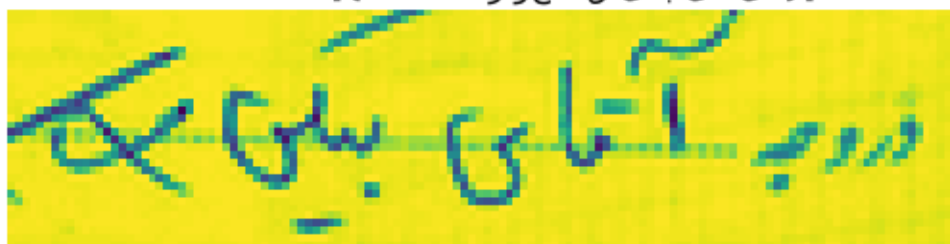
Prediction: [{'text': 'ینوی با حج هج ورد'}]



<class 'list'>

['text': 'در وجه آقای بیگی']

Prediction: [{'text': 'یگیب یاقا هج ورد'}]

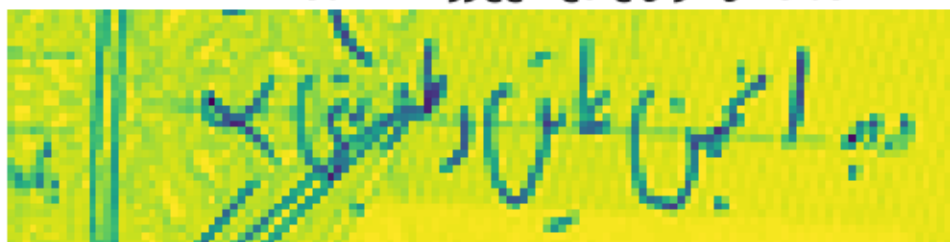


<class 'list'>

['text': 'ررجه اجن عاق رطقتی']



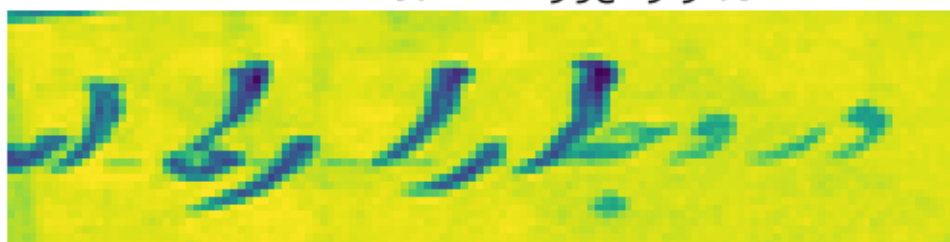
Prediction: [{'text': 'ایت‌ق‌ط‌ر‌ق‌اع‌ن‌ج‌ا‌ه‌ج‌ر‌ر'}]



<class 'list'>

{'text': 'در وجه ارا را'}

Prediction: [{'text': 'ار ارا ه‌ج‌و‌رد'}]



<class 'list'>

{'text': 'در وجه سنصص'}

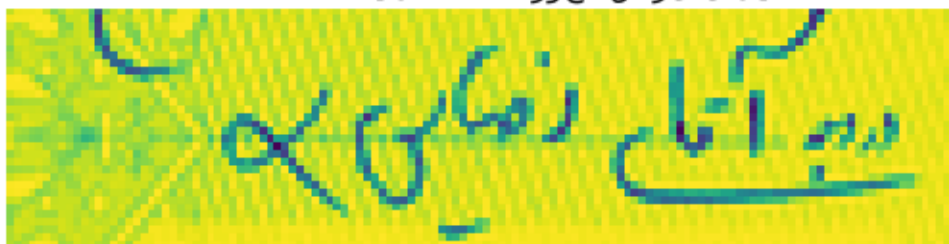
Prediction: [{'text': 'ص‌ص‌ن‌س‌ح‌ه‌ج‌و‌رد'}]



<class 'list'>

{'text': 'در وجه آقا ریای'}

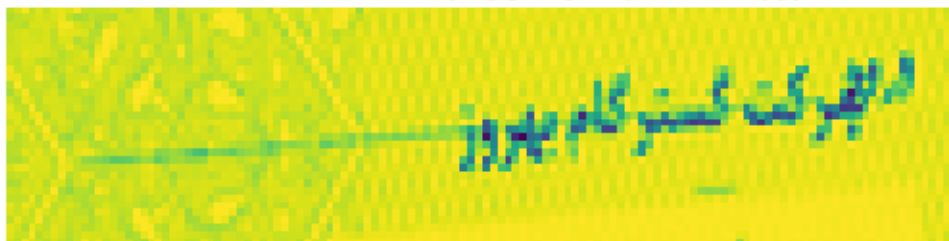
Prediction: [{ 'text': 'ایراق آه ورد' }]



<class 'list'>

{ 'text': 'درد جی بسترگاه بهروز' }

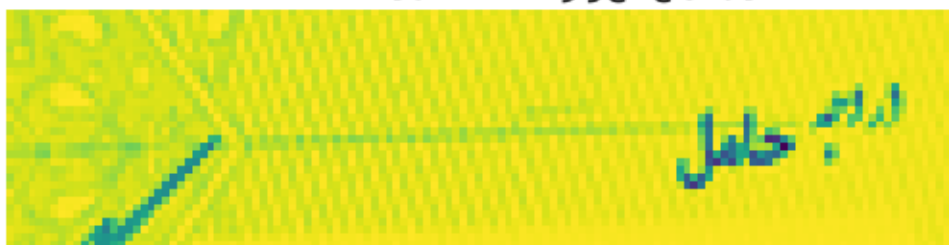
Prediction: [{ 'text': 'زورب هاگ رت سب ج رد' }]



<class 'list'>

{ 'text': 'درد وجه عال' }

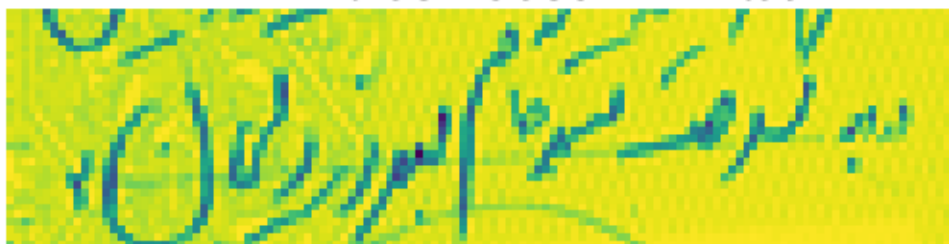
Prediction: [{ 'text': 'لاع هج ورد' }]



<class 'list'>

{ 'text': 'رد وجه گشوج ستام لبهرو ویلب' }

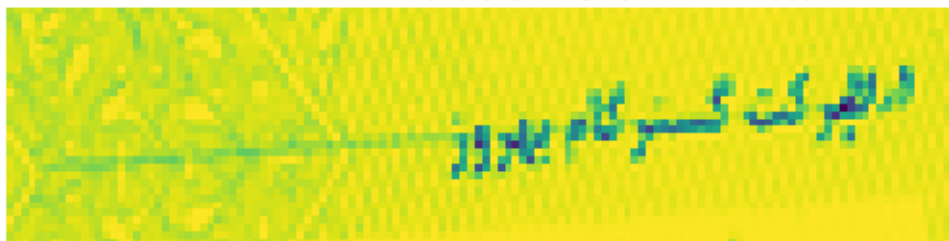
Prediction: [{'text': 'بلو وره بل ماتس جوشگ هجو در'}]



<class 'list'>

['text': 'دردر وجه گسس گاه بهروز']

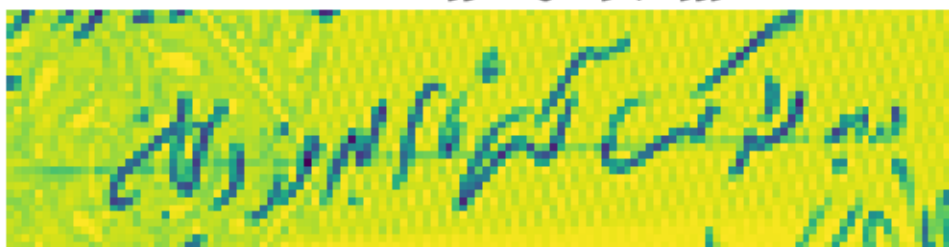
Prediction: [{'text': 'زوره ب هاگ سسگ هجو درد'}]



<class 'list'>

['text': 'دروه گسکرهم هروز دیان']

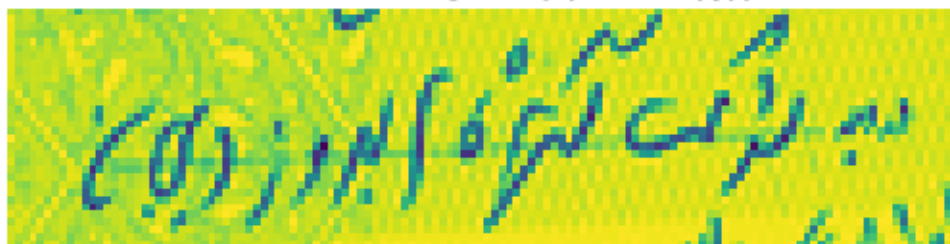
Prediction: [{'text': 'ناید زوره مهرکسگ هورد'}]



<class 'list'>

['text': 'د جه گتر رتهمبهروززدیان']

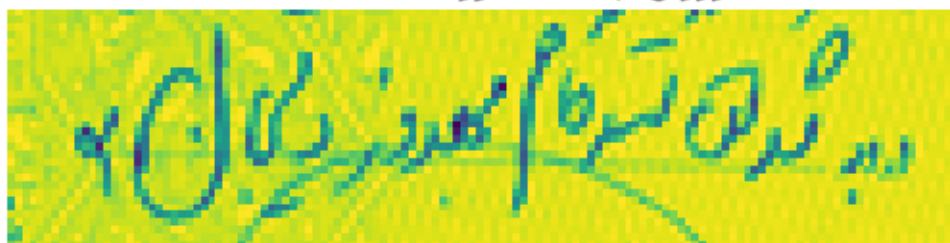
Prediction: [{'text': 'د' 'نای دز زوره ب م ه تر ر ت گ ه ج د'}]



<class 'list'>

['text': 'دروه گت گم جهرزلی ه']

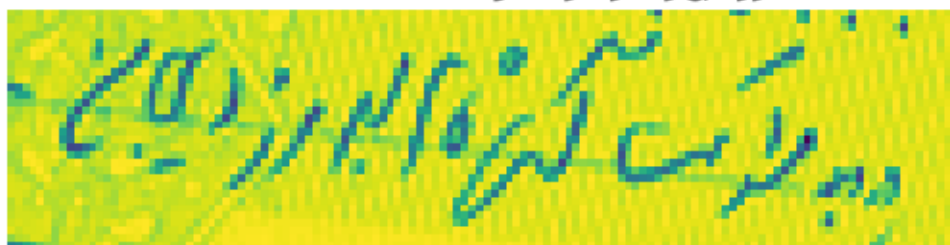
Prediction: [{'text': 'ه' 'یلرز ه ج م گ ت گ ه ورد'}]



<class 'list'>

['text': 'د وه گرتر گرجمهرز']

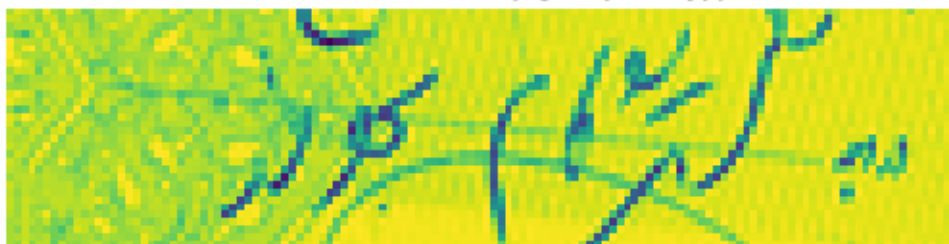
Prediction: [{'text': 'د' 'ز ره م ج ر گ ر ت ر گ ه و د'}]



<class 'list'>

['text': 'درجه گرام کوزز']

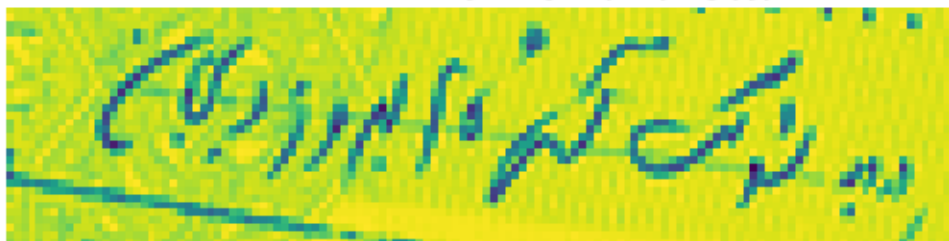
Prediction: [{'text': 'زرک مارگ هج رد'}]



<class 'list'>

['text': 'دوه گستر گرمجهروز']

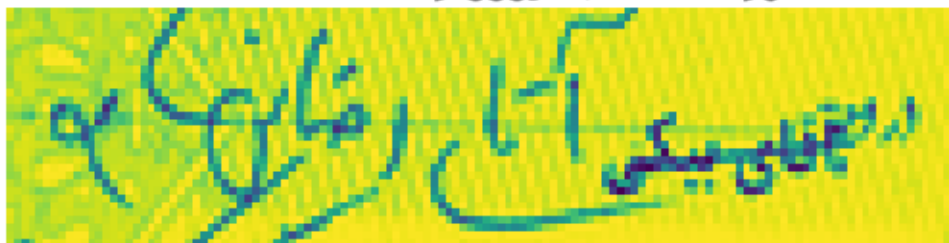
Prediction: [{'text': 'زورهجمرگ رتسگ هود'}]



<class 'list'>

['text': 'در ججچقیبیکى اای رضی']

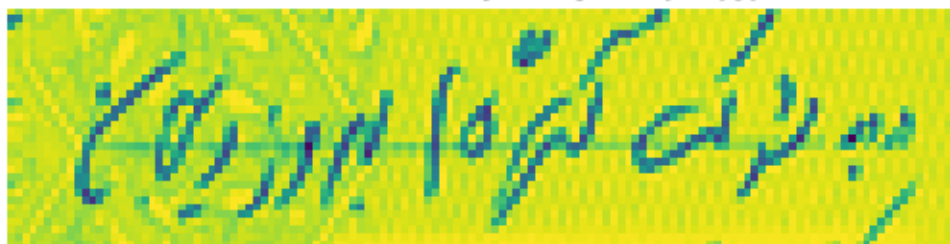
Prediction: [{'text': 'یضری اای کیبی قچج رد'}]



<class 'list'>

['text': 'دوده گسکترگ روز']

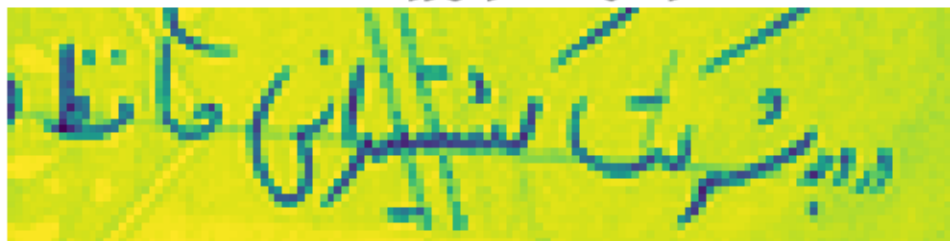
Prediction: [{'text': 'زور گرت کسگ هودا'}]



<class 'list'>

[{'text': 'دروجه رکت کشترا نی تافظ'}]

Prediction: [{'text': 'ظافات ینارت شک تکر هجورد'}]



<class 'list'>

[{'text': 'نویسنده : هادیایوسفی'}]

Prediction: [{'text': 'یوسفوی ای داه: هدنسی ون'}]



<class 'list'>

[{'text': 'برنامه سای یشفته'}]

Prediction: [{'text': 'هت فشی یاس همان رب'}]



<class 'list'>

2}] [{'text': 'جلسه

Prediction: [{'text': '2 هسلج'}]



<class 'list'>

:'}] [{'text': 'حل ریاضی':

Prediction: [{'text': 'ریاضی حل'}]



<class 'list'>

:'}] [{'text': 'الگوریتم بازگشتی'}]

Prediction: [{{'text': 'ایتش گزاب متیروگل'}}]



<class 'list'>

{{'text': 'روی خود مثال'}}

Prediction: [{{'text': 'لاثم دوخ یور'}}]



<class 'list'>

{{'text': 'مینویسم'}}

Prediction: [{{'text': 'مسی و نی م'}}]



<class 'list'>

{{'text': 'پارامتر های ورودی'}}



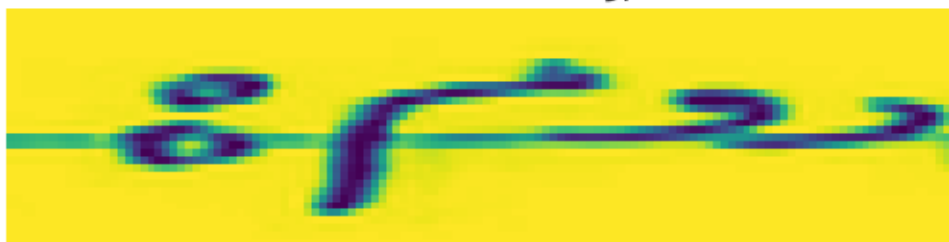
Prediction: [{ 'text': 'ی دورو یاه رت م ا راپ' }]



<class 'list'>

{ 'text': 'دوم' }

Prediction: [{ 'text': 'مود' }]



<class 'list'>

{ 'text': 'مقایسه' }

Prediction: [{ 'text': 'هسی اقم' }]



<class 'list'>

{ 'text': 'هم رشد' }

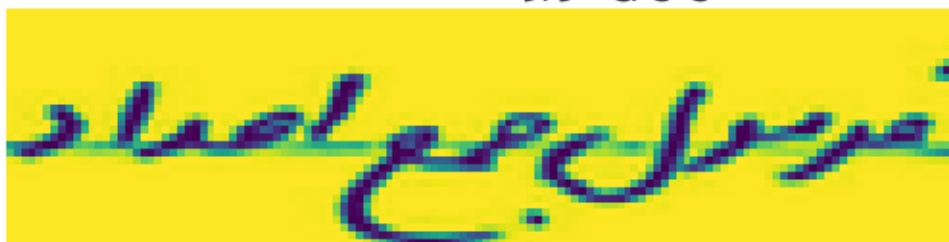
Prediction: [{ 'text': 'دشر مه' }]



<class 'list'>

{ 'text': 'فرمول جمع اعداد' }

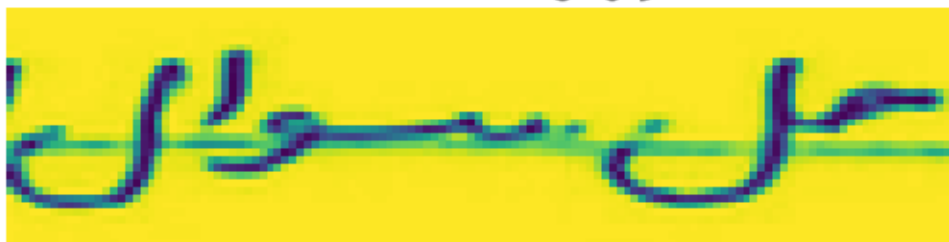
Prediction: [{ 'text': 'دادعا عمج لومرف' }]



<class 'list'>

{ 'text': 'حل سوال' }

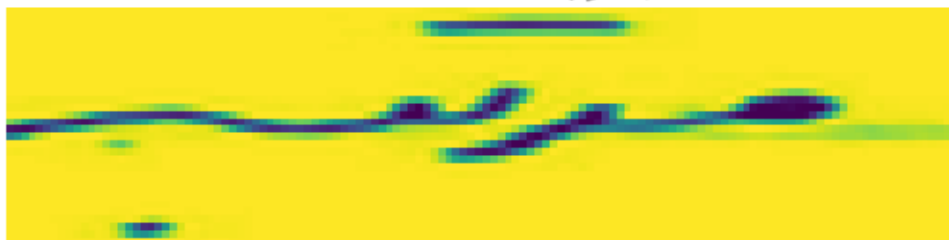
Prediction: [{ 'text': 'لاوس لح' }]



<class 'list'>

{ 'text': 'مرتبہ' }

Prediction: [{'text': 'هېت ټرم'}]



<class 'list'>

['text': 'میشه']

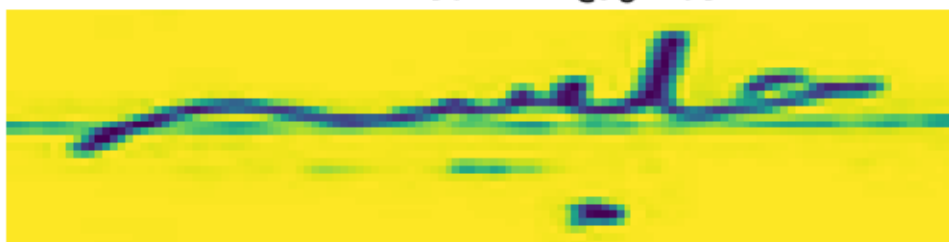
Prediction: [{'text': 'هشی م'}]



<class 'list'>

['text': 'جلسه']

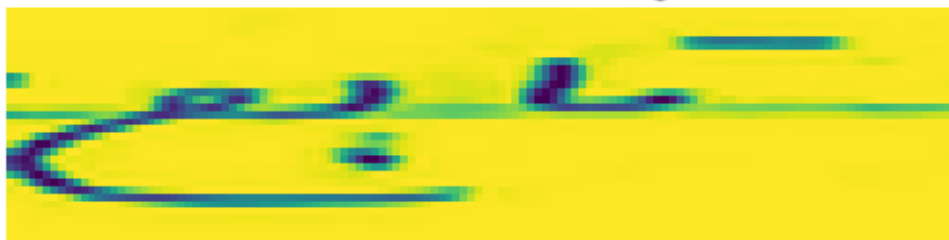
Prediction: [{'text': 'هسلج'}]



<class 'list'>

['text': 'تابع']

Prediction: [{'text': 'عبات'}]



<class 'list'>

['text': 'است']

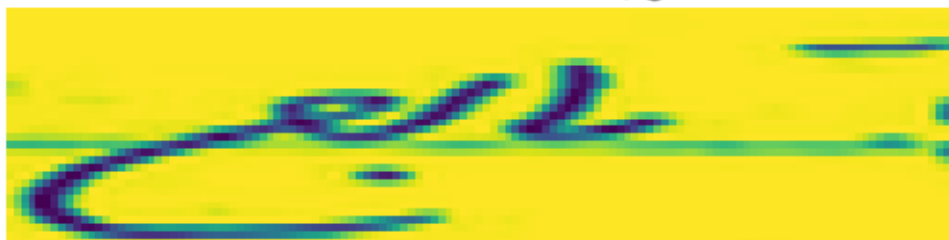
Prediction: [{'text': 'تسا'}]



<class 'list'>

['text': 'تابع']

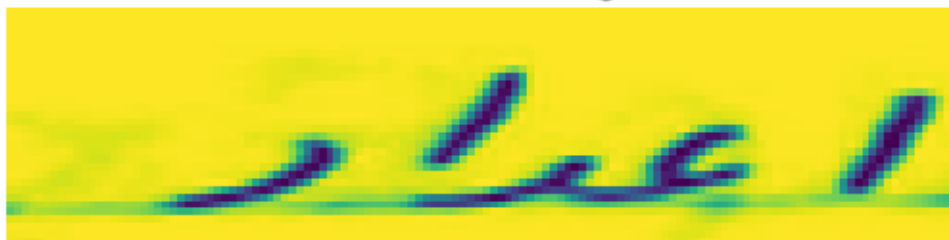
Prediction: [{'text': 'عبات'}]



<class 'list'>

['text': 'اعداد']

Prediction: [{'text': 'دادعا'}]



<class 'list'>

[{'text': 'هرگاه'}]

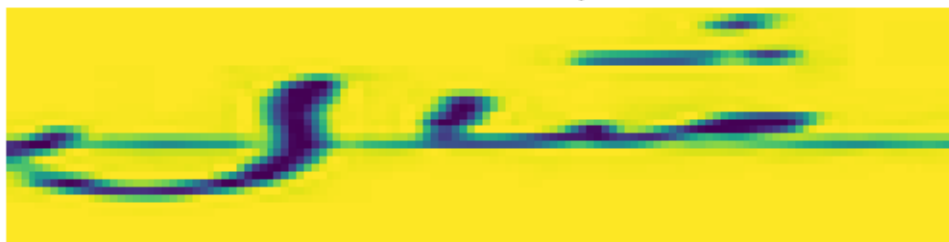
Prediction: [{'text': 'هاگره'}]



<class 'list'>

[{'text': 'مثال'}]

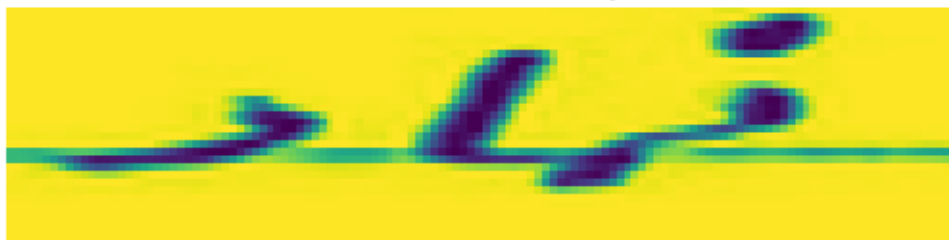
Prediction: [{'text': 'لاثم'}]



<class 'list'>

[{'text': 'نماد'}]

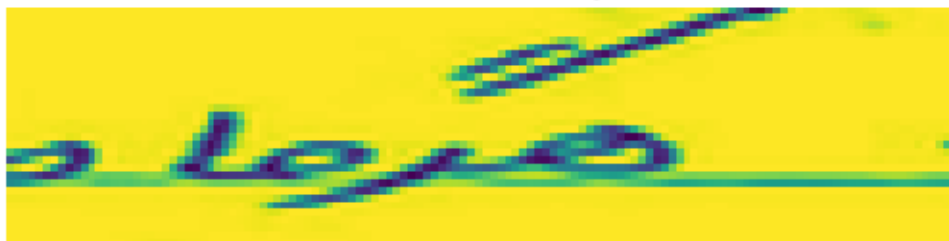
Prediction: [{'text': 'دامن'}]



<class 'list'>

[{'text': 'هرگاه'}]

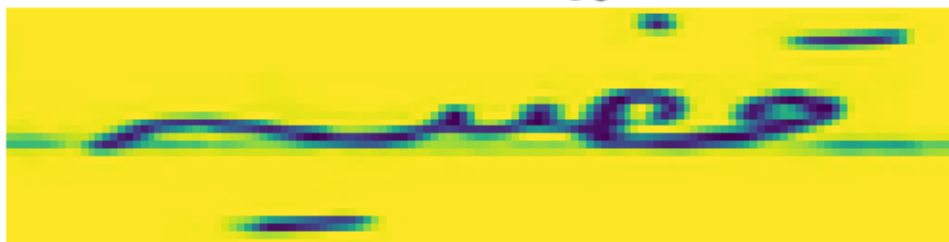
Prediction: [{'text': 'هاگره'}]



<class 'list'>

[{'text': 'قضیه'}]

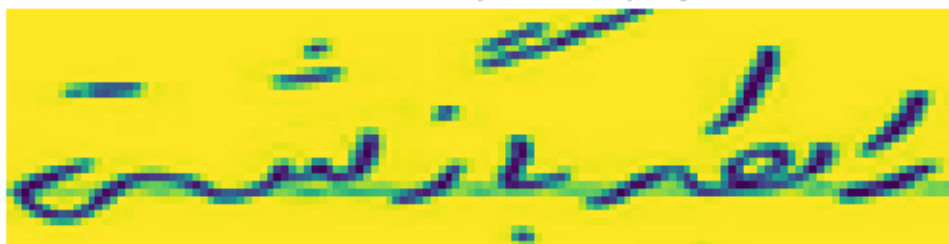
Prediction: [{'text': 'هیضوق'}]



<class 'list'>

[{'text': 'رابطه بازگشتی'}]

Prediction: [{'text': 'یتش گزاب هطبار'}]



<class 'list'>

{'text': 'به صورت'}

Prediction: [{'text': 'تروص هب'}]



<class 'list'>

{'text': 'انواع روش حل مسئله'}


Prediction: [{'text': 'هلئسم لح شورع اون'}]



<class 'list'>

{'text': 'مسائل'}

Prediction: [{'text': 'لئاسم'}]



<class 'list'>

[{'text': 'ضرب ماتريس'}]

Prediction: [{'text': 'سیرت ام برض'}]



<class 'list'>

[{'text': 'بنابراین این'}]

Prediction: [{'text': 'نیای اربانپ'}]

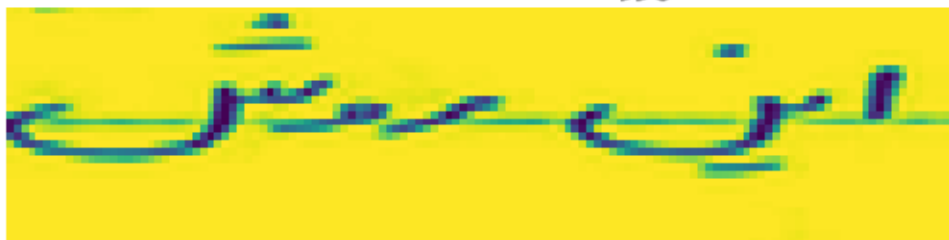


<class 'list'>

[{'text': 'این روش'}]



Prediction: [{'text': 'شور نی'}]



<class 'list'>

['text': 'به روش']

Prediction: [{'text': 'شور هب'}]



<class 'list'>

['text': 'دو ماتریس']

Prediction: [{'text': 'سیرت ام ود'}]



<class 'list'>

['text': 'ضرب دو چندجمله ای']

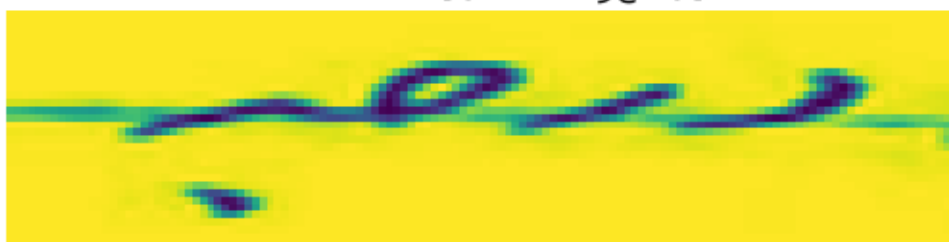
Prediction: [{ 'text': 'یا ہل مچ دن چ ود برض' }]



<class 'list'>

{ 'text': 'د رجہ' }

Prediction: [{ 'text': 'ہج رد' }]



<class 'list'>

{ 'text': 'د رجہ' }

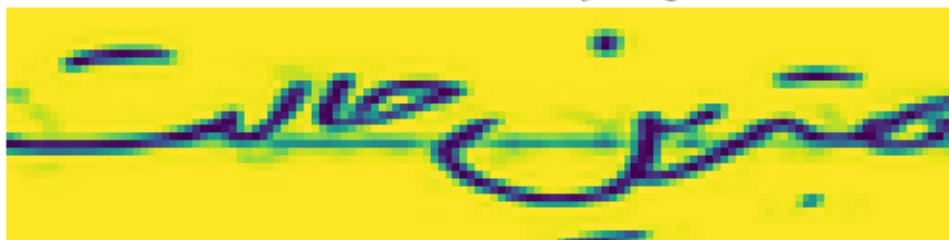
Prediction: [{ 'text': 'ہج رد' }]



<class 'list'>

{ 'text': 'بہترین حالت' }

Prediction: [{'text': 'تلاچ نیرت هب'}]



<class 'list'>

['text': 'بهترین حالت']

Prediction: [{'text': 'تلاچ نیرت هب'}]



<class 'list'>

['text': 'جست و جو دودویی']

Prediction: [{'text': 'جست و جو دودویی'}]



<class 'list'>

['text': 'جست و جو دودویی']

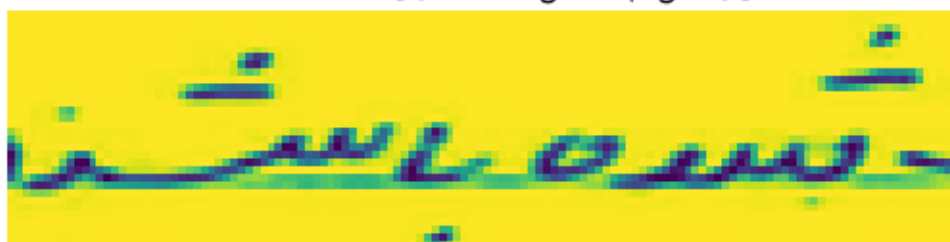
Prediction: [{'text': 'ی و دود و ج و ت س ج'}]



<class 'list'>

{'text': ' شده باشد '}

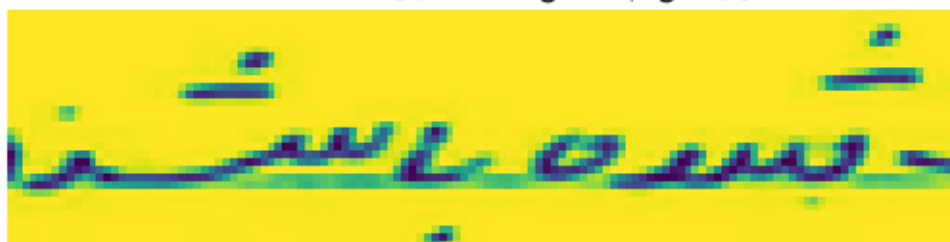
Prediction: [{'text': 'دش اب ه دش'}]



<class 'list'>

{'text': ' شده باشد '}

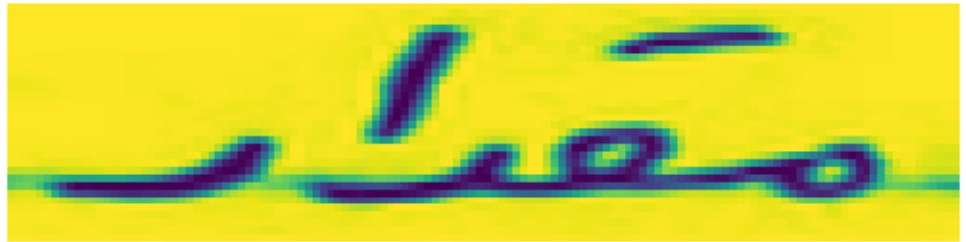
Prediction: [{'text': 'دش اب ه دش'}]



<class 'list'>

{'text': ' مقدار '}

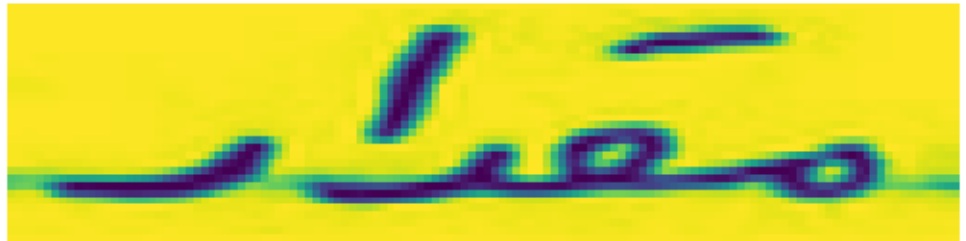
Prediction: [{'text': 'رادقم'}]



<class 'list'>

['text': 'مقدار']

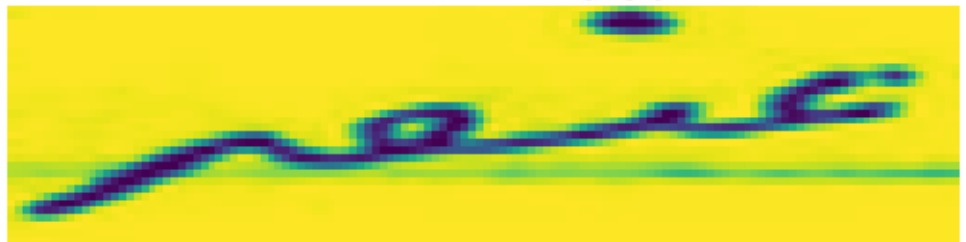
Prediction: [{'text': 'رادقم'}]



<class 'list'>

['text': 'عنصر']

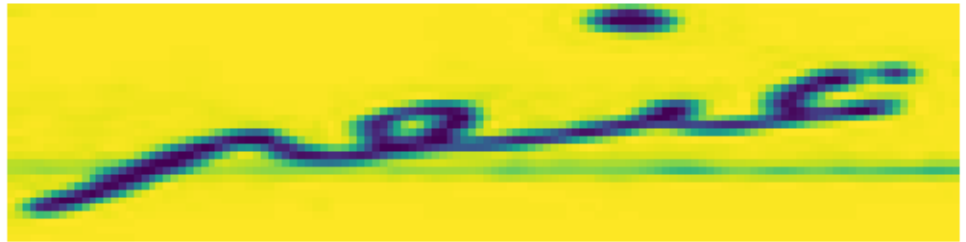
Prediction: [{'text': 'رضنع'}]



<class 'list'>

['text': 'عنصر']

Prediction: [{'text': 'رضنع'}]



<class 'list'>

['text': 'با تابع درهم ساز']

Prediction: [{'text': 'زاس مهرد عبات اب'}]



<class 'list'>

['text': 'با تابع درهم ساز']

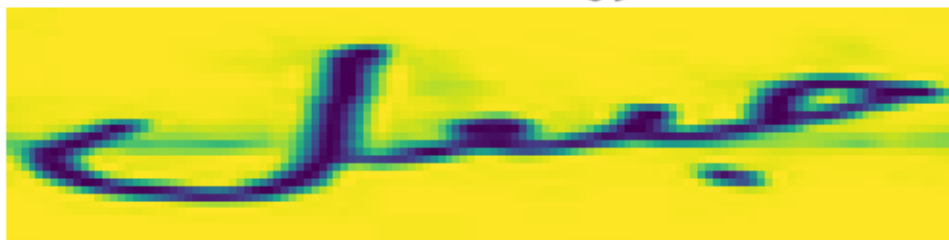
Prediction: [{'text': 'زاس مهرد عبات اب'}]



<class 'list'>

['text': 'جدول']

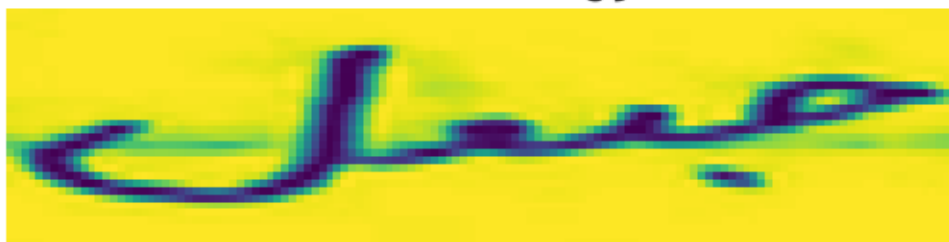
Prediction: [{'text': 'لودج'}]



<class 'list'>

['text': 'جدول']

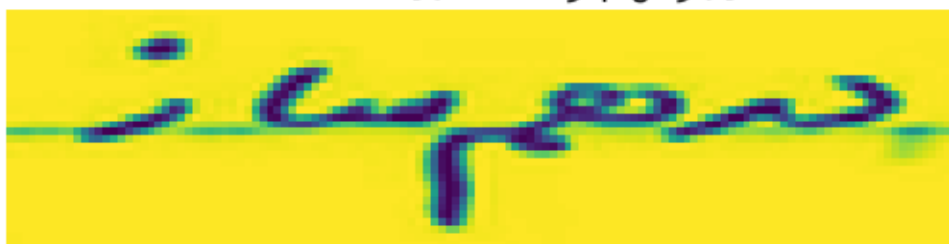
Prediction: [{'text': 'لودج'}]



<class 'list'>

['text': 'درهم ساز']

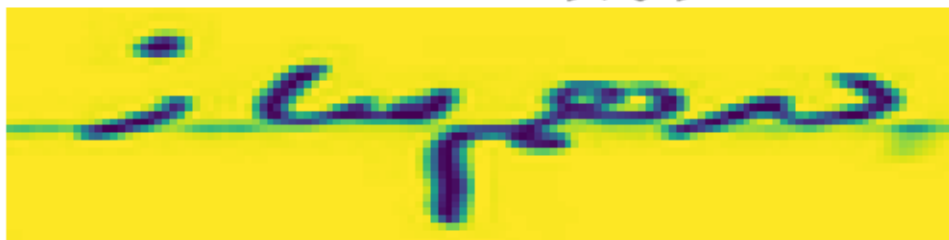
Prediction: [{'text': 'زاس مهرد'}]



<class 'list'>

['text': 'درهم ساز']

Prediction: [{'text': 'زاس مهرد'}]



<class 'list'>

{'text': 'با تابع'}

Prediction: [{'text': 'عبات اب'}



<class 'list'>

{'text': 'با تابع'}

Prediction: [{'text': 'عبات اب'}

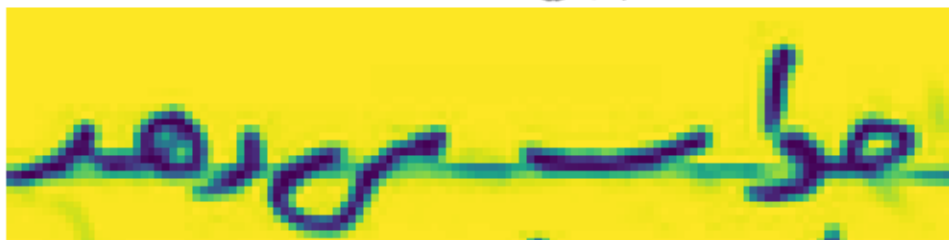


<class 'list'>

{'text': 'جواب میدهد'}



Prediction: [{'text': 'دهدیم باوج'}]



<class 'list'>

[{'text': 'دیتا خوب'}]

Prediction: [{'text': 'بوخ اتی د'}]



<class 'list'>

[{'text': 'واحد است'}]

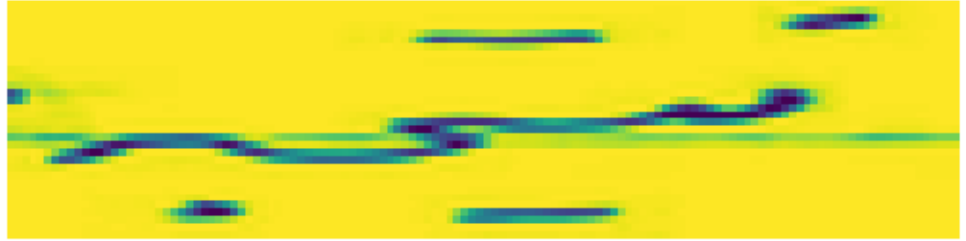
Prediction: [{'text': 'تسا دجاو'}]



<class 'list'>

[{'text': 'نتیجه'}]

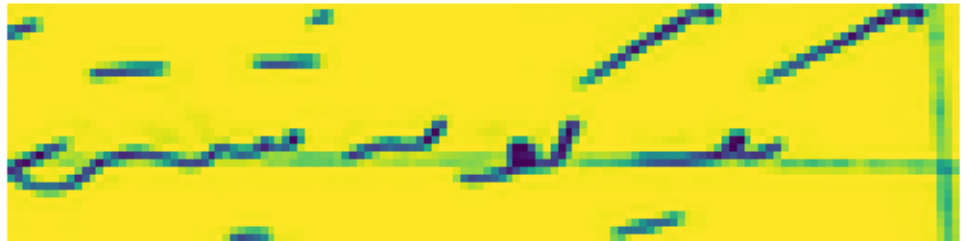
Prediction: [{'text': 'هچ یتن'}]



<class 'list'>

[{'text': 'یک کوله پشته'}]

Prediction: [{'text': 'یت شپ هلوک کی'}]



<class 'list'>

[{'text': 'حل این مسئله'}]

Prediction: [{'text': 'هلئسم نیا لح'}]



<class 'list'>

[{'text': 'کوله پشته'}]

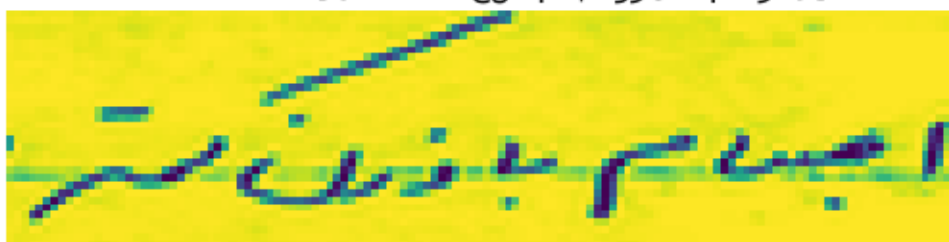
Prediction: [{'text': 'ی ت ش پ ه ل و ک'}]



<class 'list'>

['text': 'اجسام با وزن کمتر']

Prediction: [{'text': 'ر ت م ک ن ز و اب م اس ج ا'}]



<class 'list'>

['text': 'ابتدا انتخاب شوند']

Prediction: [{'text': 'د ن و ش ا خ ت ن ا د ت ب ا'}]



<class 'list'>

['text': 'الگوریتم پیچدگی زمانی']

Prediction: [{'text': 'ی نامز ی گدچی پ تیروگل'}]



<class 'list'>

{'text': 'مرتب سازی اولیه'}

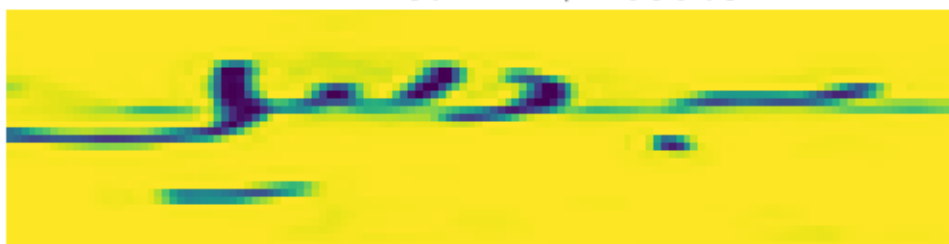
Prediction: [{'text': 'هیلوا یزاس بترم'}]



<class 'list'>

{'text': 'به دلیل'}

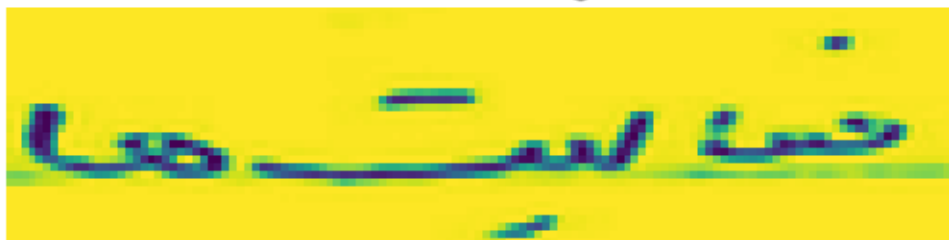
Prediction: [{'text': 'لیلد هب'}]



<class 'list'>

{'text': 'فعالیت ها'}

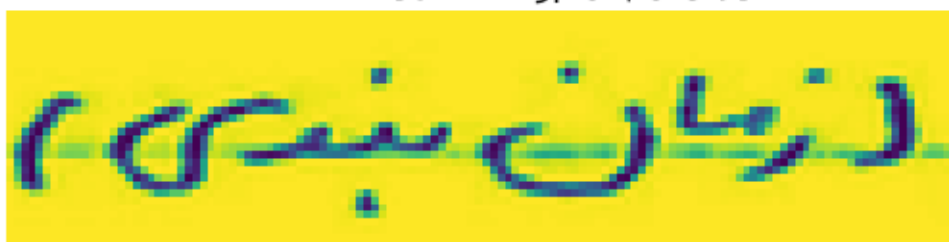
Prediction: [{'text': 'اه تیلاعف'}]



<class 'list'>

[{'text': 'زمان بندی'}]

Prediction: [{'text': 'ی‌دنب نامز'}]



<class 'list'>

[{'text': 'در این صورت'}]

Prediction: [{'text': 'تروص نی‌ا رد'}]



<class 'list'>

[{'text': 'بیشترین'}]

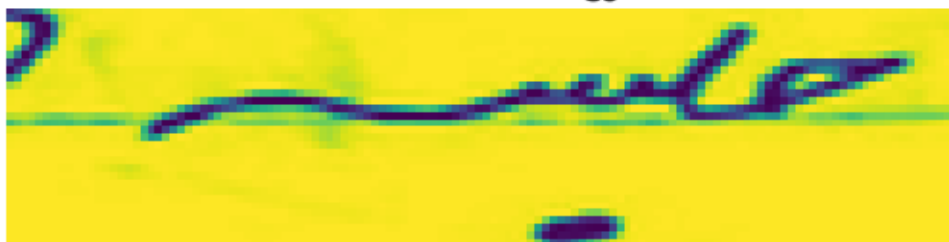
Prediction: [{ 'text': 'نیرتشیب' }]



<class 'list'>

{ 'text': 'حسه' }

Prediction: [{ 'text': 'هسح' }]



<class 'list'>

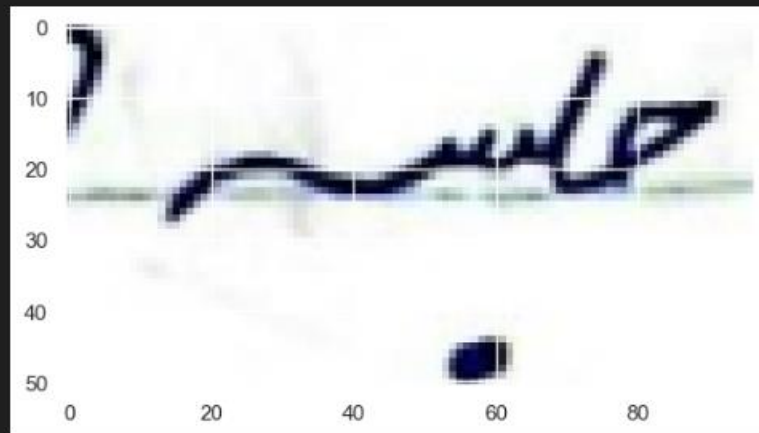
{ 'text': 'متاسیه' }

Prediction: [{ 'text': 'هساتم' }]

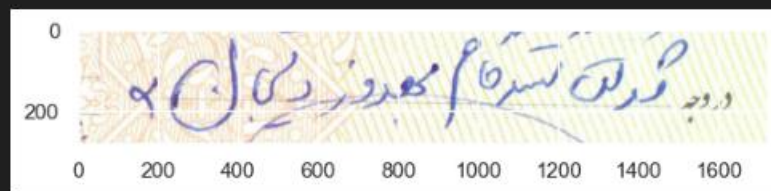


CNN\_Bilstm:

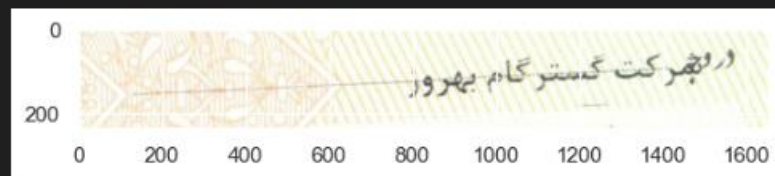
['ε']



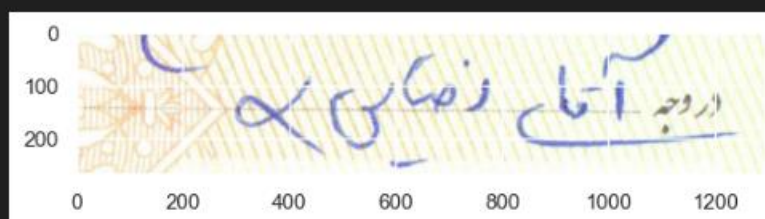
['در وجه گستر بهروز']



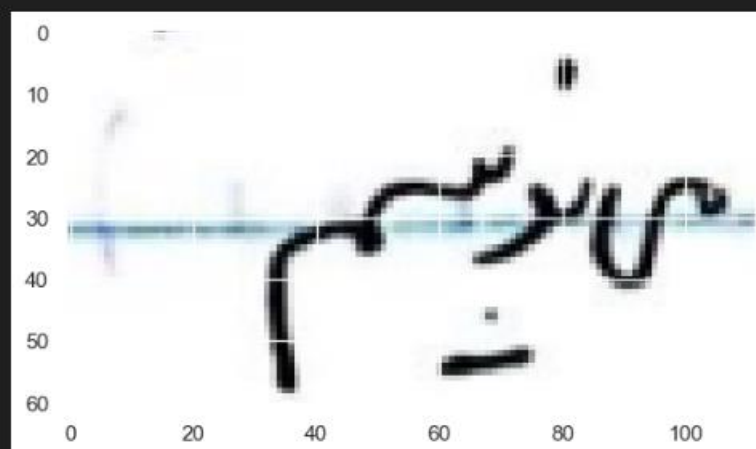
['در وجه گستر بهروز']



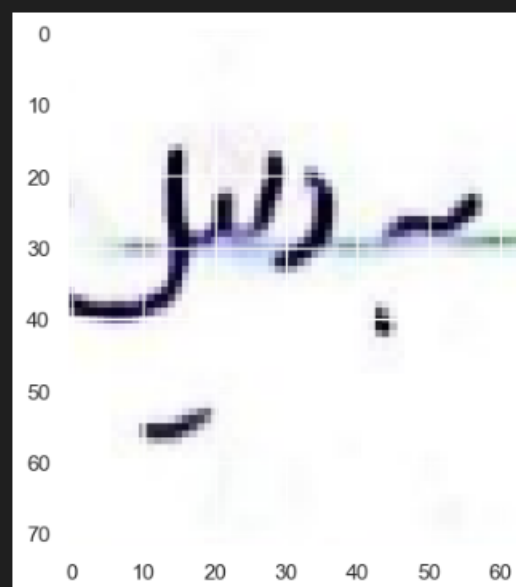
['در وجه آقی']



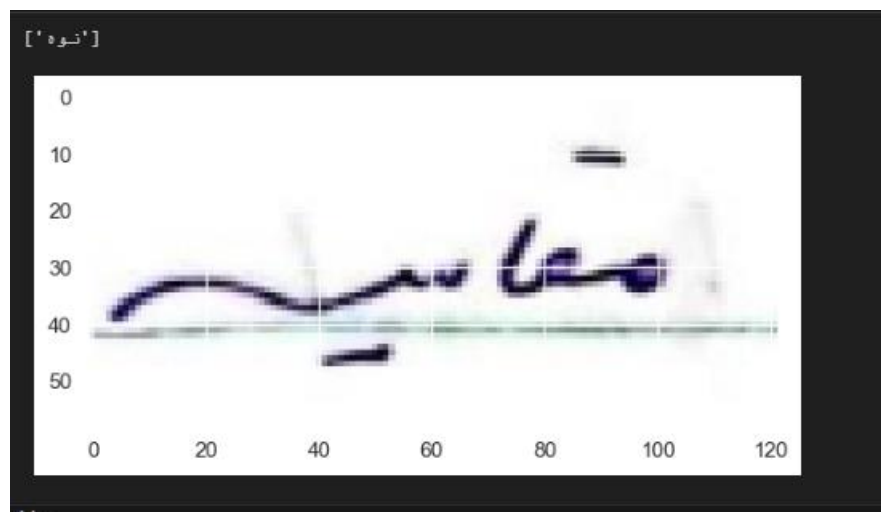
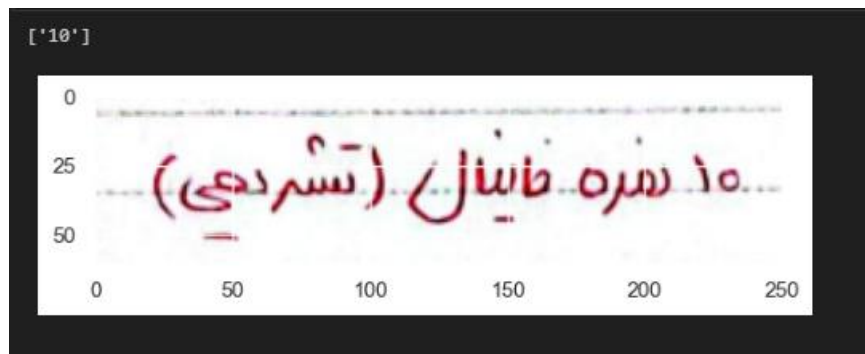
['مینو']



['بیل']







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





