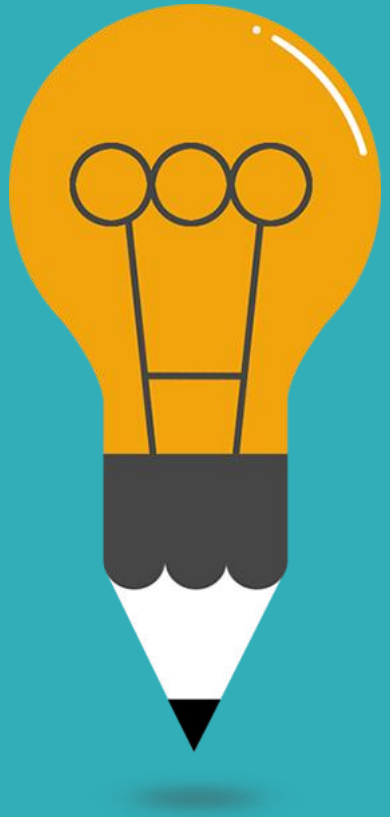
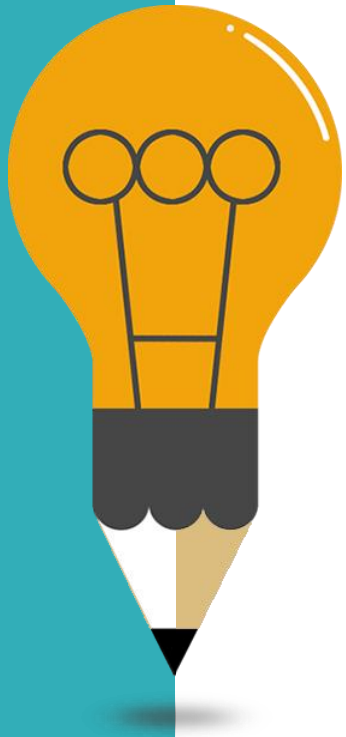


مدل کپچا شکن

محسن مدنی
سهیل اشعاری
یاسین اسفندیاری

بهار ۹۷

مراحل کار



۱ جمع آوری داده

کپچا و لیبل های آن

۲ تمیز کردن داده

تقسیم کپچا به رقم های آن

۳ ساخت و آموزش مدل

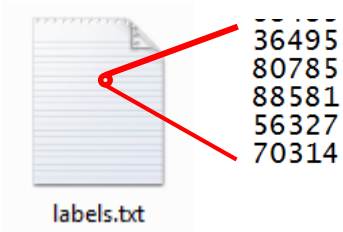
ساخت مدل به کمک داده ها

۴ ارزیابی مدل

پیش بینی کپچا ها به کمک مدل آموزش دیده

جمع آوری داده

در این مرحله ما به کمک زبان php توانستیم کپچای مورد نظر خود را شبیه سازی و یک دیتاست با حجم تقریبی ۲۰ هزار کپچا به همراه لیبل را جمع آوری کنیم.



تمیز کردن داده

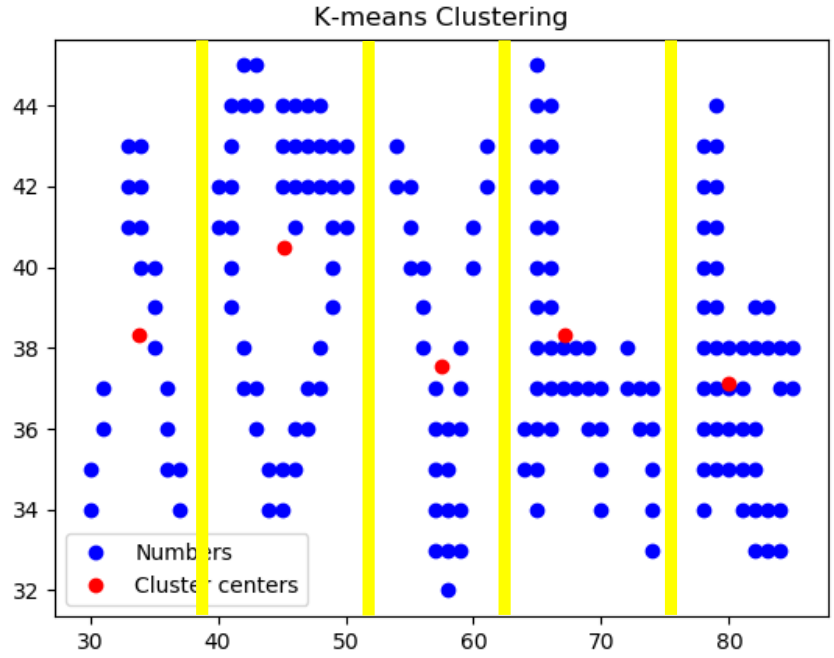
به طور خلاصه در این مرحله تلاش بر از بین بردن نویز ها (مانند خط های اضافه تصویر) می باشد. در ادامه برای جدا سازی ارقام ما از دو روش خوشه بندی و لبه یابی استفاده کردیم، در ادامه به نحوه کار این روش ها و نتایج آن بر روی داده ها می پردازیم.

حذف نویز

به کمک فیلتر های محو کننده، erode و dilate و آستانه گذاری و چندین و چند ابزار دیگر موفق شدیم تا داده ها را عاری از هرگونه نویز اضافی کنیم و آن ها را به فضای تک کانال ببریم.

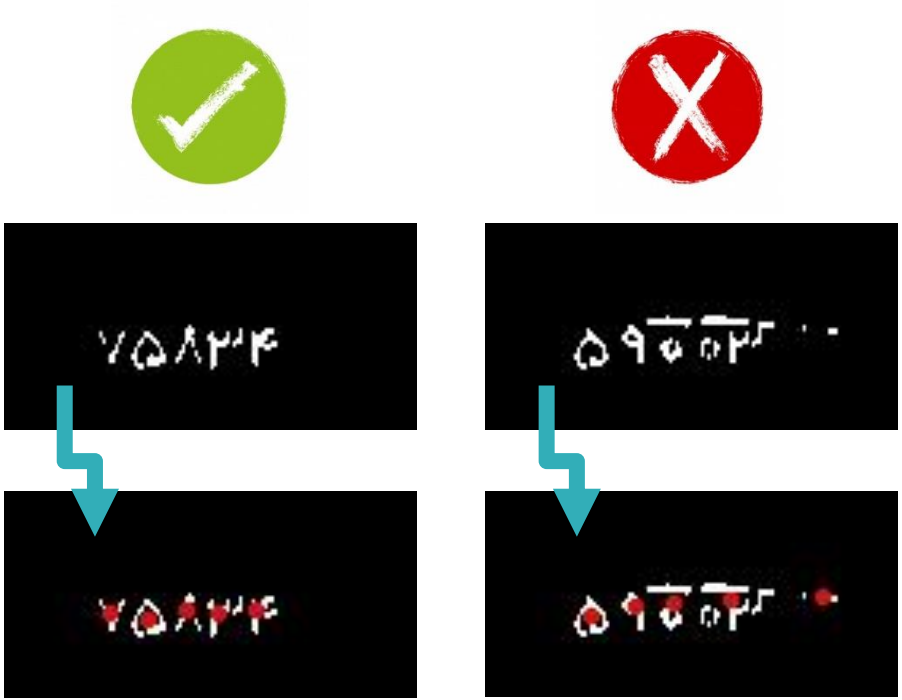
خوشه بندی

در این روش هدف تقسیم داده ها با مقدار رنگی ۲۵۵ به ۵ بخش مختلف و سپس جداسازی ارقام است. به کمک این روش ما بر روی بیشتر داده موفق به خوشه بندی شدیم ولی به علت فاصله کم میان ارقام این روش از دقت کمی برخوردار بود.



خوشه بندی

همانطور که پیش تر گفتیم نرخ دقت این روش چندان مورد قبول نبود.

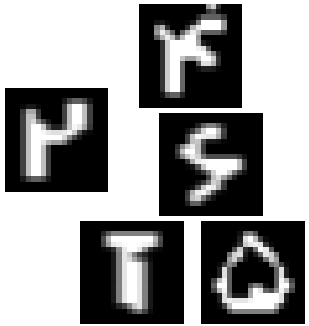


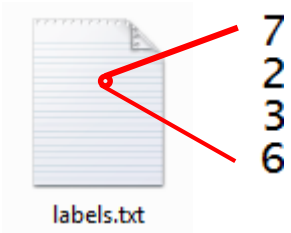
لبه یابی

به کمک لبه یابی توانستیم دقت را بالا برده و داده های کمتری را از دست بدهیم، این روش به کمک کتابخانه openCV پیاده سازی شده است. هرچند همچنان این روش هم دارای مشکلاتی مانند یافتن بخش های بیش تر از تعداد موارد مورد نیاز، دوتا یکی گرفتن اعداد و .. می باشد.

جداسازی ارقام

در نهایت بعد از یافتن لبه ها نوبت به جداسازی ارقام و ساخت دیتاست جدید می رسد. درصد بالایی از دیتاست جدید خوانایی بالا برای چشم عادی دارند.

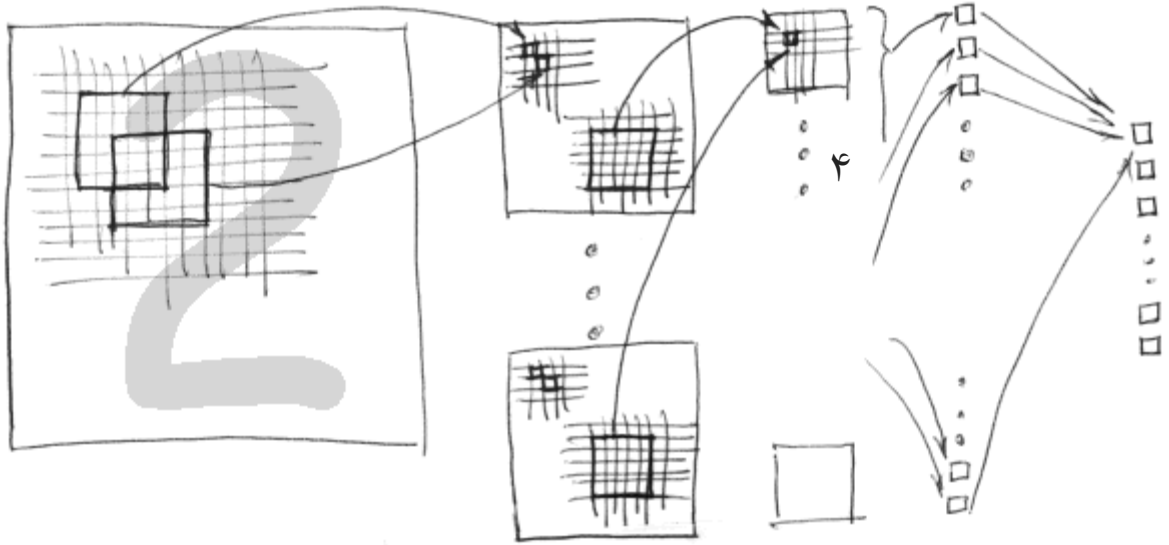





THE
END
IS
NEAR

ساخت و آموزش مدل

مدل CNN ما تحت فریمورک Keras به کمک دیتاست ذکر شده در اسلاید های قبلی ساخته و آموزش داده شد.



معماری مدل

در این بخش شما با معماری مدل و همچنین برخی از Hyperparameter های مدل کپچا شکن آشنا می شوید.

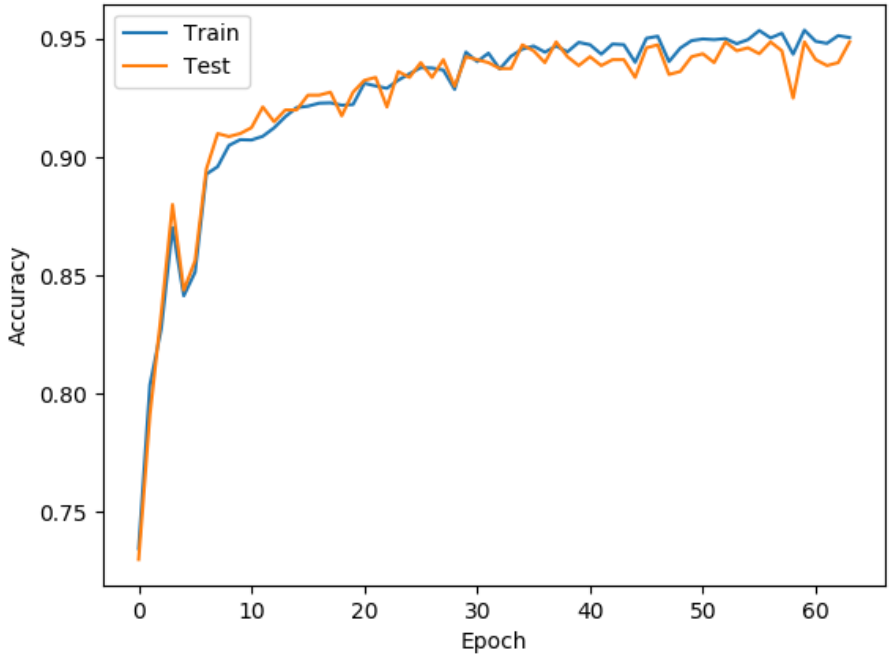
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 18, 18, 32)	320
max_pooling2d_1 (MaxPooling2	(None, 9, 9, 32)	0
conv2d_2 (Conv2D)	(None, 7, 7, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 3, 3, 64)	0
conv2d_3 (Conv2D)	(None, 3, 3, 64)	4160
flatten_1 (Flatten)	(None, 576)	0
dropout_1 (Dropout)	(None, 576)	0
dense_1 (Dense)	(None, 128)	73856
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 10)	1290

Total params: 98,122
Trainable params: 98,122
Non-trainable params: 0

```
# Settings :
EPOCH_NO = 64
BATCH_SIZE = 64
OPTIMIZER = 'rmsprop'
METRICS = 'accuracy'
```

ارزیابی مدل

با آزمون و خطا بر روی پارامتر ها، مشکلات Performance و Overfitting و .. تا درصد قابل قبولی برطرف شد و دقت پیش بینی مدل به ۹۵ درصد بر روی داده های validation رسید.



پایان