

Image Processing Project

Armaghan Sarvar 9531807

برای انجام این پروژه، دیتاست مربوطه در محیط کولب دانلود و ذخیره‌سازی شد و کتابخانه‌های مورد نیاز نصب شدند:

```
wget http://www.vision.caltech.edu/Image_Datasets/Caltech256/256_ObjectCategories.tar

--2020-01-28 13:15:45-- http://www.vision.caltech.edu/Image_Datasets/Caltech256/256_ObjectCategories.tar
Resolving www.vision.caltech.edu (www.vision.caltech.edu)... 34.208.54.77
Connecting to www.vision.caltech.edu (www.vision.caltech.edu)|34.208.54.77|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1183006720 (1.1G) [application/x-tar]
Saving to: '256_ObjectCategories.tar'

256_ObjectCategories 100%[=====] 1.10G 8.35MB/s in 2m 20s

2020-01-28 13:18:06 (8.05 MB/s) - '256_ObjectCategories.tar' saved [1183006720/1183006720]

[ ] !tar -xvf 256_ObjectCategories.tar

[ ] import glob
import cv2 as cv
import keras
```

سوال ۱


از آنجا که هر کدام از پوشه‌های تصاویر شامل عکس‌های مربوط به یک دسته خاص بودند، من داده‌های test را ۲۰ درصد اول هر کدام از پوشه‌ها و داده‌های train را ۸۰ درصد بقیه‌ی هر کدام از پوشه‌ها در نظر گرفتم تا از همگی دسته‌ها هم داده test و هم داده train داشته باشیم.

اولین داده آموزشی:

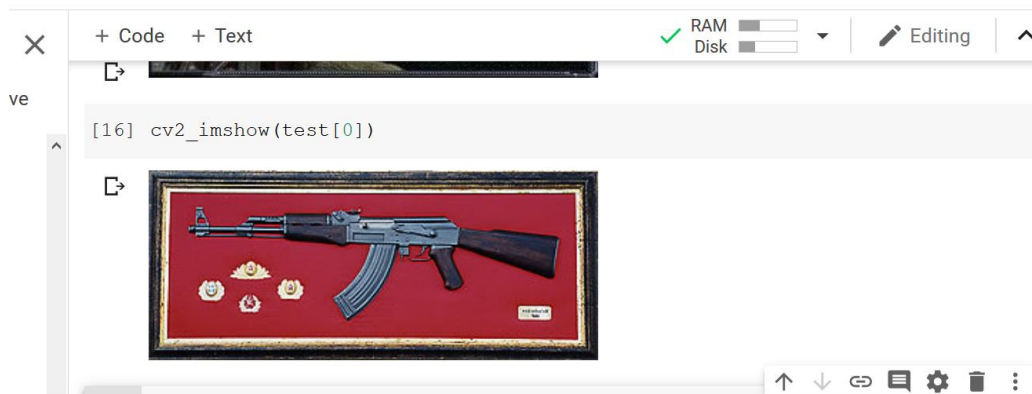
```
[5] print(len(train))
print(len(test))

24581
6026

[7] from google.colab.patches import cv2_imshow
cv2_imshow(train[0])
```



اولین داده آزمایشی:



سوال ۲

انتخاب ویژگی: (Feature Selection)

برای انتخاب ویژگی از تصاویر، از آن جا که هر کدام از تصاویر یک سایز متفاوت بودند، ابتدا تصمیم گرفتم همه را ۳۰ در ۳۰ بکنم. برای این کار از یک کرنل با سایز برابر با نسبت طول و عرض هر کدام از تصاویر آموزشی به تصویر هدف (۳۰ در ۳۰) در نظر گرفته شد و با استفاده از آن، بین این تعداد (قرار گرفته شده در این کرنل) از پیکسل‌های تصویر آموزش **میانگین‌گیری** انجام شد. برای شبکه عصبی اول یعنی MLP نیاز به داده آموزشی یک بعدی بود.

پس داده‌های آموزشی جدید (که همگی ۳۰ در ۳۰ بودند) هر کدامشان در یک آرایه با طول ۳۰ * ۳۰ (تعداد کل پیکسل‌ها) ریخته شدند.

پر کردن هر کدام از تصاویر آموزشی با سایز جدید:

چون آرایه ما (تصویر جدید ما) الان ۱ بعدی است، باید سطر و ستون را بر حسب متغیر i که بر یک بعد `iterate` می‌کند پیدا شوند:

```
row = int(i/ new_size[0])
col = int(i% new_size[1])
```

هر کدام از پیکسل‌های تصویر جدید از کدام پیکسل تصویر اصلی بدست آمده اند؟(نقطه‌ی اول کرنل در تصویر اصلی)

```
point_in_orig = [row * kernel_size[0], col * kernel_size[1]]
```

حالا باید به ازای هر کدام از مولفه های R و G و B میانگین گیری انجام شود. (به اندازه همان سائز کرنل که در بالا بدست آمد)

البته لازم به ذکر است ابتدا برای سادگی کار تصاویر را به صورت grayscale خواندم اما دقت خوبی بدست نیامد پس همه ی مولفه ها نگه داشته شدند.

برای ورودی شبکه عصبی داده ماتریسی مورد نیاز بود.

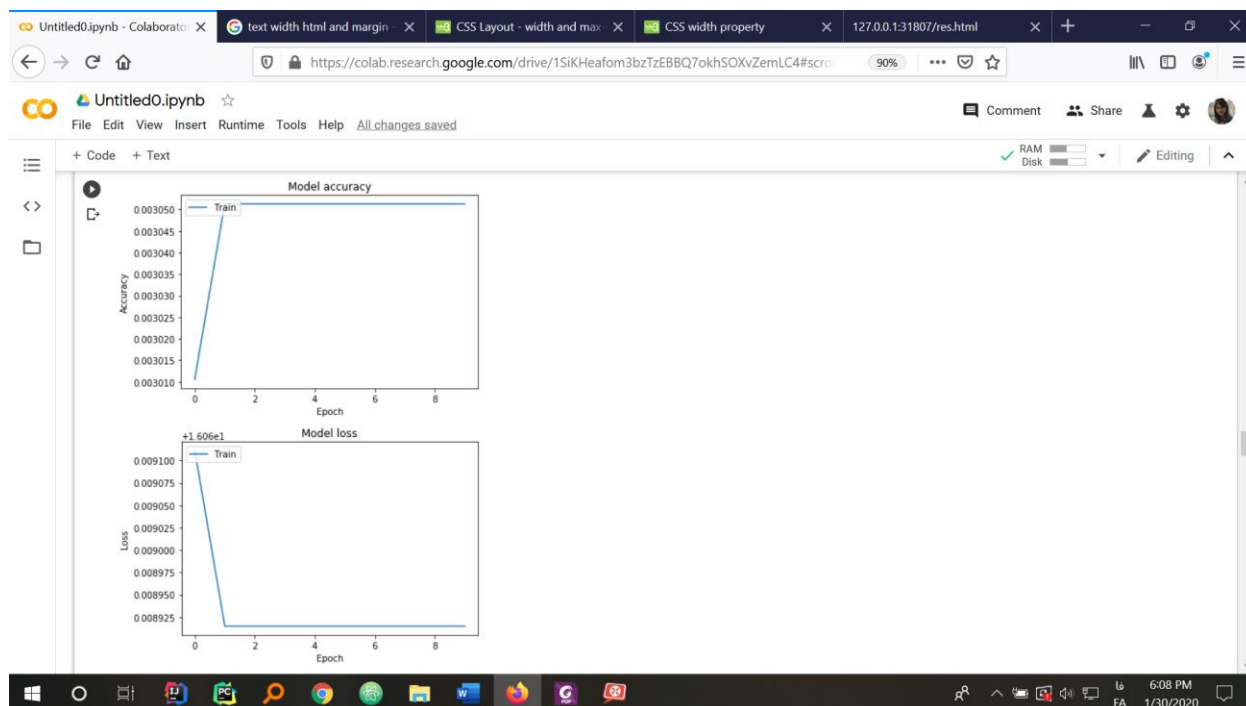
پس labelها که همان شماره پوشه تصاویر بودند و همچنین داده های آموزشی جدید به np array تبدیل شدند. اندازه ماتریس:

سطرها: تعداد داده ها (تصاویر)

ستون ها: تعداد ویژگی ها = $3 * \text{طول جدید} * \text{عرض جدید} (30 * 30 * 3)$

تعریف شبکه عصبی با پارامترهای گفته شده انجام شد و برای داده تست دقت شبکه عصبی کمتر بود.

تعداد featureها را به ۷۵۰۰ تغییر دادم یعنی به جای اینکه تصاویر را ۳۰ در ۳۰ کنم، آن ها را ۵۰ در ۵۰ کردم و در نتیجه آن چند درصد تغییر در دقت کلی مشاهده شد.



سوال ۳

برای شبکه عصبی CNN به وضوح نتیجه بهتری حاصل شد.

به طور کلی می‌توان گفت از آنجا که MLP به عنوان ورودی vector دریافت می‌کند و CNN برخلاف آن تنسور دریافت می‌کند، CNN قادر است روابط مکانی را بهتر تشخیص دهد. این مساله در پردازش تصویر حائز اهمیت خواهد بود چون ما علاقمند هستیم روابط بین پیکسل‌های نزدیک به هم را برای یادگیری مدل در نظر بگیریم.

خروجی برای شبکه عصبی CNN به صورت زیر است: (خروجی به تعداد نوروں ها وابسته است و این خروجی به ازای نوروں‌های تعریف شده بدست آمد).

