

به نام خدا

محمد امین جمشیدی گوه‌ریزی

PROJECT 4:

این دیتاست شامل یک سری عکس بافت شناسی (بافت سلولی بدن) است که در هر کدام از این عکس ها 7 نوع هسته (7 نوع از بافت بدن) میتواند موجود باشد (در کل 7 نوع وجود دارد الزاما هر 7 نوع از بافت بدن در یک عکس وجود ندارد)

که این 7 نوع به علاوه back ground تصویر میشود 8 کلاس که عبارت اند از:

back ground = 0

other = 1

inflammatory = 2

healthy epithelial = 3

dysplastic/malignant epithelial = 4

fibroblast = 5

muscle = 6

endothelial = 7

برای خروجی این دیتاست هم لیبل های خروجی دیتاست به ازای semantic segmentaion و هم لیبل خروجی object detection و سایر مختصات لازم برای object detection موجود است.

از آنجایی که قصد ما semantic segmentaion ما فقط به یک لیبل نیاز داریم که آن هم type_map است

Type_map هر تصویری یک ماتریس 1000x1000 است(در ضمن ابعاد عکس های ورودی دیتاست نیز 1000x1000 است)که با مقادیر 0 تا 7 مقدار دهی شده است که بیانگر این است که هر پیکسل مربوط به کدام کلاس است.

به علت کم بود تعداد داده ار 2 تکنیک agmmentation و transfer learning استفاده می کنیم

:Agmmentation

چون که باید تغییرات اعمال شده در هر تصویر کاملاً متناسب با تغییرات mask (همان لیبل خروجی تصویر) باشد باید مطمئن شویم که همان تغییری که برای تصویر اتفاق می افتد برای لیبل تصویر هم رخ می دهد از همین رو یک تابع برای این کار می نویسم

```
import cv2 as cv
def data_agmenting(x_train, y_train):
    new_x_train = list()
    new_y_train = list()
    for i in range(27):
        new_x_train.append(cv2.rotate(x_train[i],cv2.ROTATE_90_CLOCKWISE))
        new_y_train.append(cv2.rotate(y_train[i],cv2.ROTATE_90_CLOCKWISE))
        new_x_train.append(cv2.flip(x_train[i],0))
        new_y_train.append(cv2.flip(y_train[i],0))
        new_x_train.append(cv2.flip(x_train[i],1))
        new_y_train.append(cv2.flip(y_train[i],1))
    return new_x_train, new_y_train
```

این تابع با گرفتن list تصاویر و لیبل ها آنها تک تک تصاویر و لیبل ها را به صورت همزمان augment می کند که این agument عبارت است از:

90 درجه چرخاندن تصویر به صورت ساعت گرد

Flip کردن تصویر از چپ به راست

Flip کردن تصویر از بالا به پایین

که در این حالت به ازای هر تصویر 3 تصویر دیگر داریم

در این حالت مطمئن هستیم که لیبل متناسب با تصویر augment شده را داریم

Transfer learning:

در اینجا از کتابخانه

https://github.com/qubvel/segmentation_models

برای پیاده سازی و fine tuning مدل استفاده می کنیم

مدل را به این شکل می سازیم که:

```
model = sm.Unet('resnet152',  
               ,input_shape=(320,320,3),classes=8,activation='softmax',encoder_weights='imagenet',encoder_freeze=True)
```

همانطور که از کد معلوم است برای قسمت encoder از معماری

resnet152 استفاده شده است و بعد از آن برای وزن دهی اولیه

encoder از مقادیر وزن بر روی دیتاست iamgenet استفاده شده است

اول داده های ولیدیشن (15 درصد داده های train) را جدا می کنیم و با استفاده از این داده ها هاپر پارامتر ها را تنظیم کرده

که در این حالت به $iou=0.6$ برای داده های ولیدیشن می رسیم

حال داده تست را اعمال می کنیم

ابتدا به این صورت که وزن های لایه encode را freeze می کنیم و بعد از آموزش لایه decoder تمام مدل را آموزش می دهیم

که نتایج به شکل زیر است:

lou کل روی داده تست 0.41 و f1_score کل نیز برابر با 0.43 است.

lou برای تمام کلاس ها: (8 کلاس)

```
8.19606682e-01, 0.00000000e+00, 6.08695652e-02,  
3.38764863e-05,  
5.32683006e-02, 1.32112143e-01, 1.51041667e-03,  
0.00000000e+00
```

dic برای تمام کلاس ها: (8 کلاس):

```
0.89541033, 0. , 0.10348327, 0. ,  
0.04976073,  
0.20936416, 0. , 0.
```

Pixel acc=0.76

Confusion matrix:S

