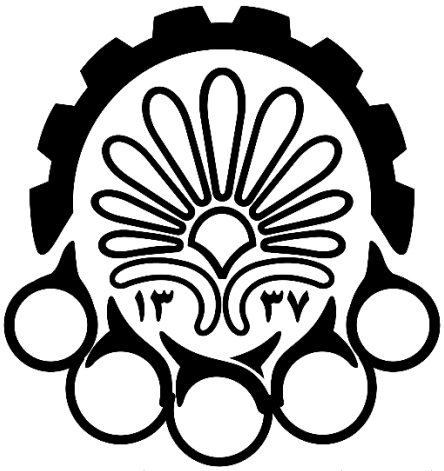


دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

تمر ین ششم درس بینایی ماشین

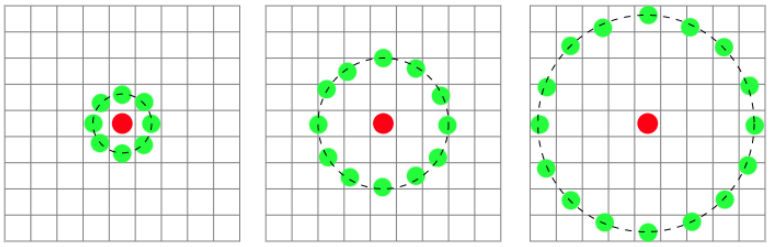
دکتر صفابخش

غلامرضا دار ۴۰۰۱۳۱۰۱۸

زمستان ۱۴۰۱

فهرست مطالب

الف)



برای استخراج ویژگی از نواحی تصویر به کمک LBP به این گونه عمل میکنیم که به ازای هر پیکسل از تصویر، یک عدد باینری مطابق با الگوریتم LBP استخراج میکنیم. این عدد برای هر پیکسل، از مقایسه مقدار آن پیکسل با پیکسل‌های اطرافش که بر روی محیط یک دایره به شعاع مشخص هستند به دست می‌آید. اگر پیکسل مرکزی از یک پیکسل اطراف بزرگتر باشد یک بیت ۱ و اگر کوچک تر باشد یک بیت ۰ به کد باینری اضافه میکنیم. حاصل یک عدد باینری برای هر پیکسل است.

برای استخراج ویژگی از یک ناحیه تصویر، میتوانیم در آن ناحیه، LBP را برای همه پیکسل‌ها محاسبه کنیم. یک سری عدد بدست می‌آید که بر روی آنها هیستوگرام میگیریم. این هیستوگرام میزان فراوانی هر مقدار منحصر به فرد را در آن ناحیه بیان میکند.این هیستوگرام، ویژگی استخراج شده از ناحیه است. همچنین می‌توانیم با مشخص کردن یک اندازه مانند ۱۶ برای bin_count هیستوگرام، سایز بردار ویژگی را برای همه تصاویر، یکسان کنیم. تصویر زیر نتیجه استخراج ویژگی LBP از ده تصویر مربوط به کلاس Weapon است. همانطور که دیده می‌شود اکثر تکه تصاویر، بیان مشابهی دارند.

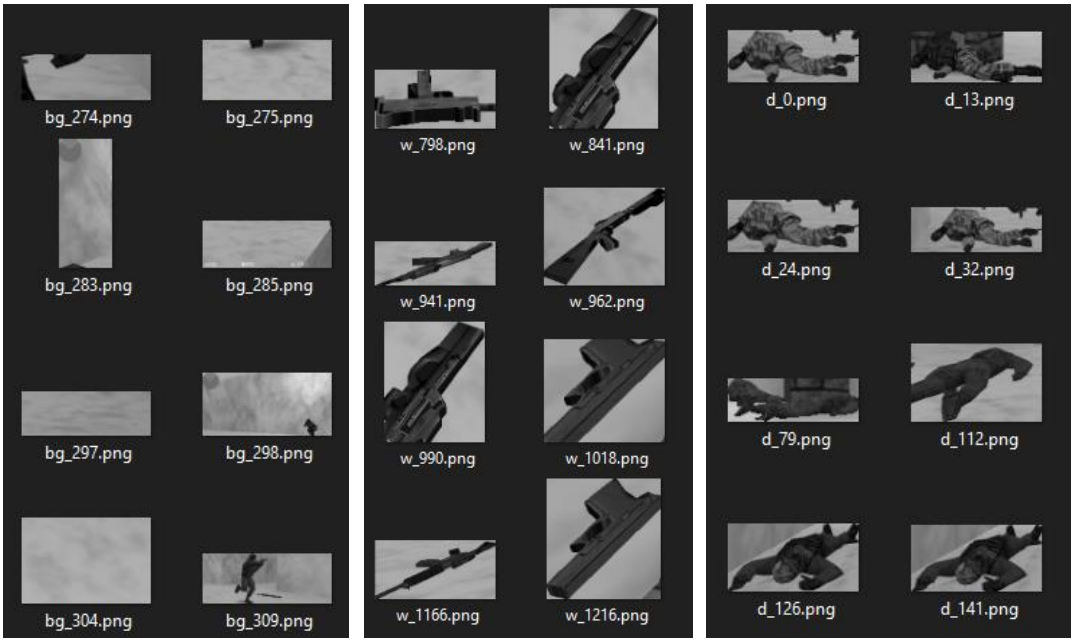
ب)

مشکل اصلی‌ای که روش‌های بر پایه keypoint مانند SIFT دارند این است که به استخراج Keypoint وابسته اند و در واقع این روش‌ها به ازای هر Keypoint یک بردار ویژگی خروجی می‌دهند. این امر باعث می‌شود در تصاویری با اندازه‌های مختلف که به احتمال زیاد تعداد Keypoint های متفاوتی دارند، اندازه بردار ویژگی کلی تصویر یکسان نباشد و برای هر تصویر ابعاد متفاوتی داشته باشد.

ج)

در این بخش قصد داریم با تکمیل کدهای فراهم‌شده، در نهایت یک تصویر ورودی را بگیریم، با استفاده از الگوریتم Selective_Search تعدادی ناحیه بر روی آن پیدا کنیم، با کمک روش LBP برای این ناحیه ها ویژگی استخراج کنیم، این ویژگی‌ها را به یک دسته‌بند بدهیم و پس از تعیین کلاس شئ موجود در آن ناحیه، در تصویر اصلی یک مستطیل با رنگ مربوط به آن ناحیه رسم کنیم. در نهایت Object Detection را انجام داده ایم.

مرحله اول، استخراج نواحی‌ای از تصویر اصلی و نسبت دادن یک برچسب به آن است. این بخش توسط کد فراهم شده انجام شده است. این نواحی مشابه شکل زیر هستند.



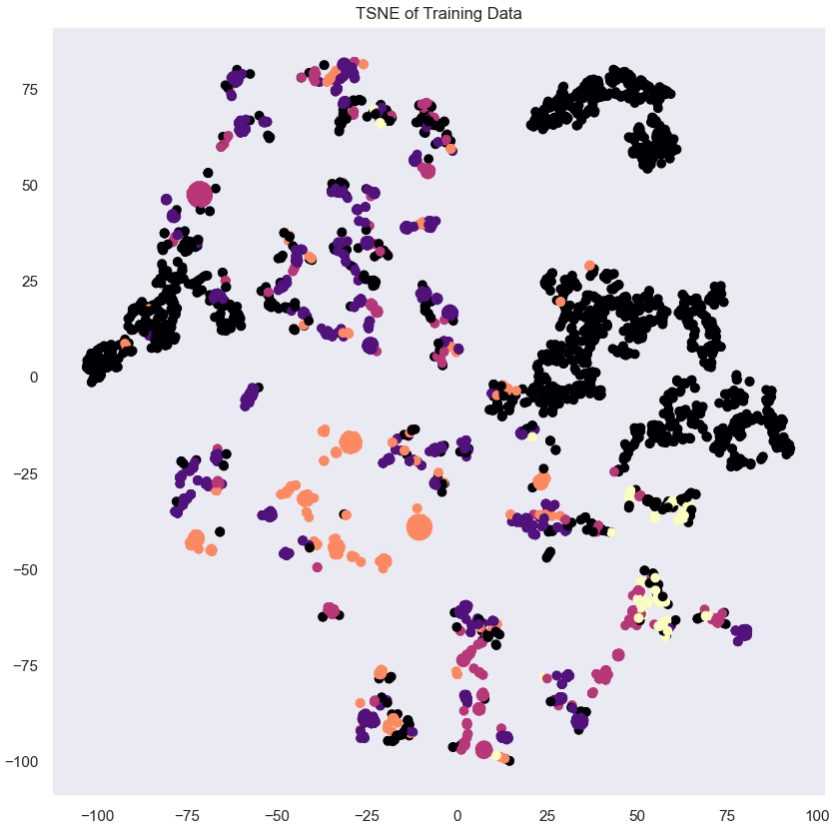
در **مرحله دوم** از این نواحی یک هیستوگرام به کمک روش LBP استخراج میکنیم. این هیستوگرام در واقع یک بردار ۱۶ بعدی است(تعداد bin های هیستوگرام) که به عنوان ویژگی برای دسته‌بندی هر ناحیه استفاده می‌شود. همانطور که در تصاویر زیر میبینید، تصاویر مربوط به هر دسته، تقریبا LBP یکسانی دارند که این امر باعث تمیز این تصاویر می‌شود.

در ادامه با استفاده از دسته‌بند مانند KNN یا SVM، سعی می‌کنیم بیاموزیم که هر دسته از تصاویر، چه شکلی دارند. این باعث می‌شود که در مرحله تست اگر یک تصویر ناحیه از دسته Weapon به مدل دادیم، مدل آنرا Weapon تشخیص دهد.

در این بخش ۴ روش مختلف آزمایش شد که نتایج خلاصه آن‌ها در جدول زیر قابل مشاهده است.

ویژگی	دسته‌بند	صحت(ACCURACY)
صرفا میانگین رنگ ناحیه!	KNN(k=1)	90%
۵ ویژگی پایه مانند ... ,MIN, MAX, MEAN	KNN(k=1)	95%
شبکه عصبی EfficientNet-b0	شبکه عصبی	96%
LBP با ۱۶ بعد	KNN(k=1)	97%

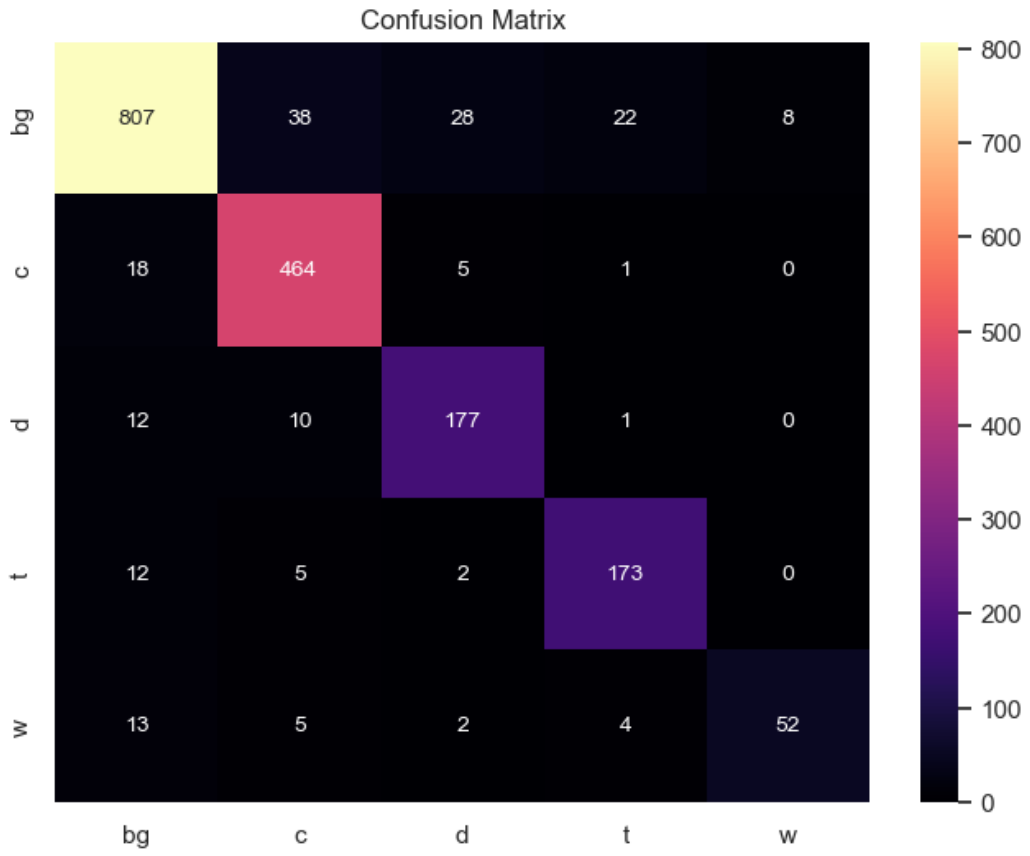
نکته قابل توجه این است که صرفا استفاده از میانگین مقدار پیکسل‌های ناحیه به عنوان ویژگی نیز می‌تواند تا حد زیادی این کلاس‌ها را از یکدیگر جدا کند. و ویژگی‌های (min, max, mean, std, median) در کنار هم با دقت ۹۵ درصد تقریبا با شبکه‌های عصبی و LBP هم تراز بودند. تصویر زیر نتیجه اجرای الگوریتم کاهش بعد T-SNE بر روی این ۵ ویژگی است. مشاهده می‌شود که همین ۵ ویژگی به تنهایی می‌توانند به خوبی کلاس‌های مسئله را از یکدیگر جدا کنند.



در جدول صفحه قبل ویژگی Accuracy گزارش شد که برای یک مسئله دسته‌بندی چندکلاسه با توزیع نامتوازن معیار مناسبی نیست. به همین دلیل در چند صفحه آتی، جدولی شامل معیارهای سنجش بیشتر مانند f1, recall, precision و همچنین ماتریس درهم ریختگی را گزارش میکنیم.

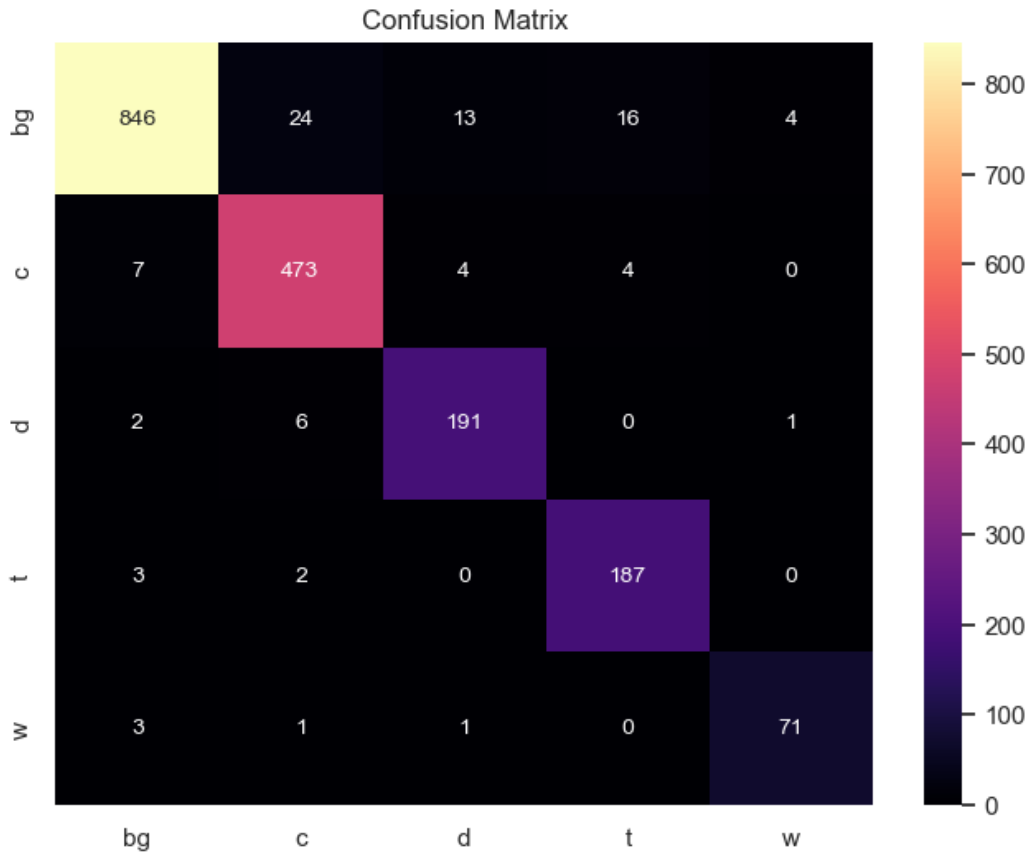
روش اول (صرفا میانگین)

	precision	recall	f1-score	support
0	0.94	0.89	0.91	903
1	0.89	0.95	0.92	488
2	0.83	0.89	0.86	200
3	0.86	0.90	0.88	192
4	0.87	0.68	0.76	76
accuracy			0.90	1859
macro avg	0.88	0.86	0.87	1859
weighted avg	0.90	0.90	0.90	1859



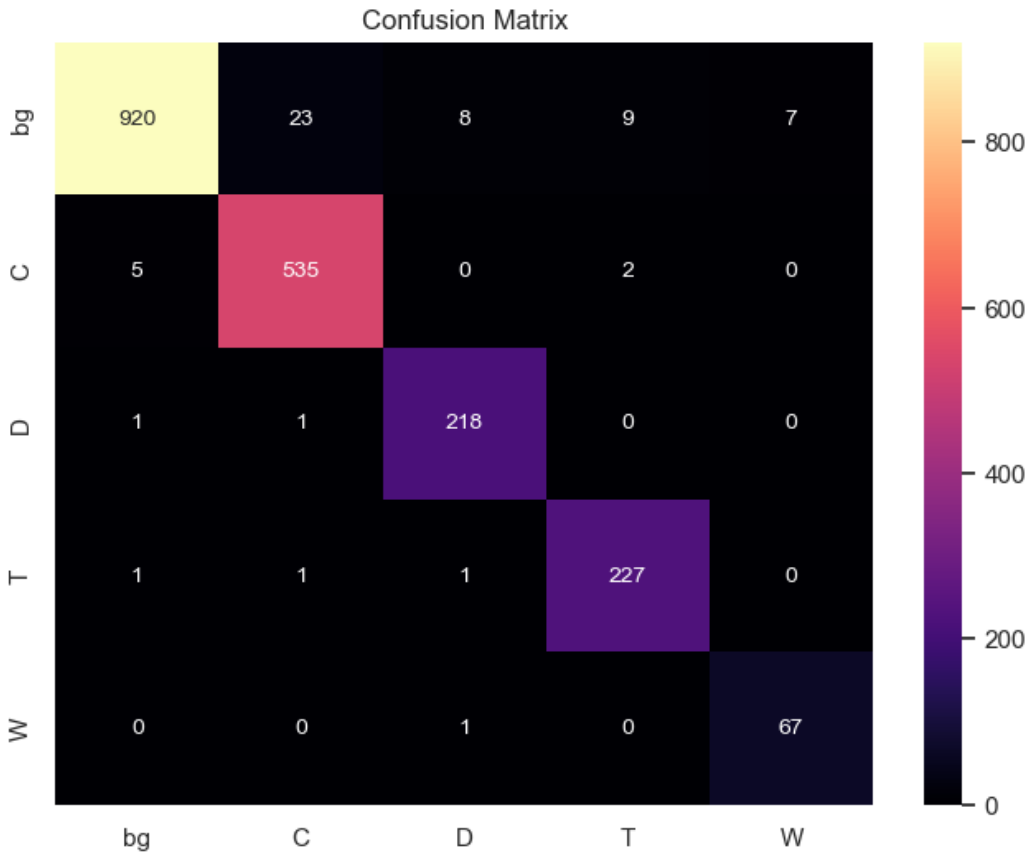
روش دوم (min, max, mean, median, std)

	precision	recall	f1-score	support
0	0.98	0.94	0.96	903
1	0.93	0.97	0.95	488
2	0.91	0.95	0.93	200
3	0.90	0.97	0.94	192
4	0.93	0.93	0.93	76
accuracy			0.95	1859
macro avg	0.93	0.95	0.94	1859
weighted avg	0.95	0.95	0.95	1859

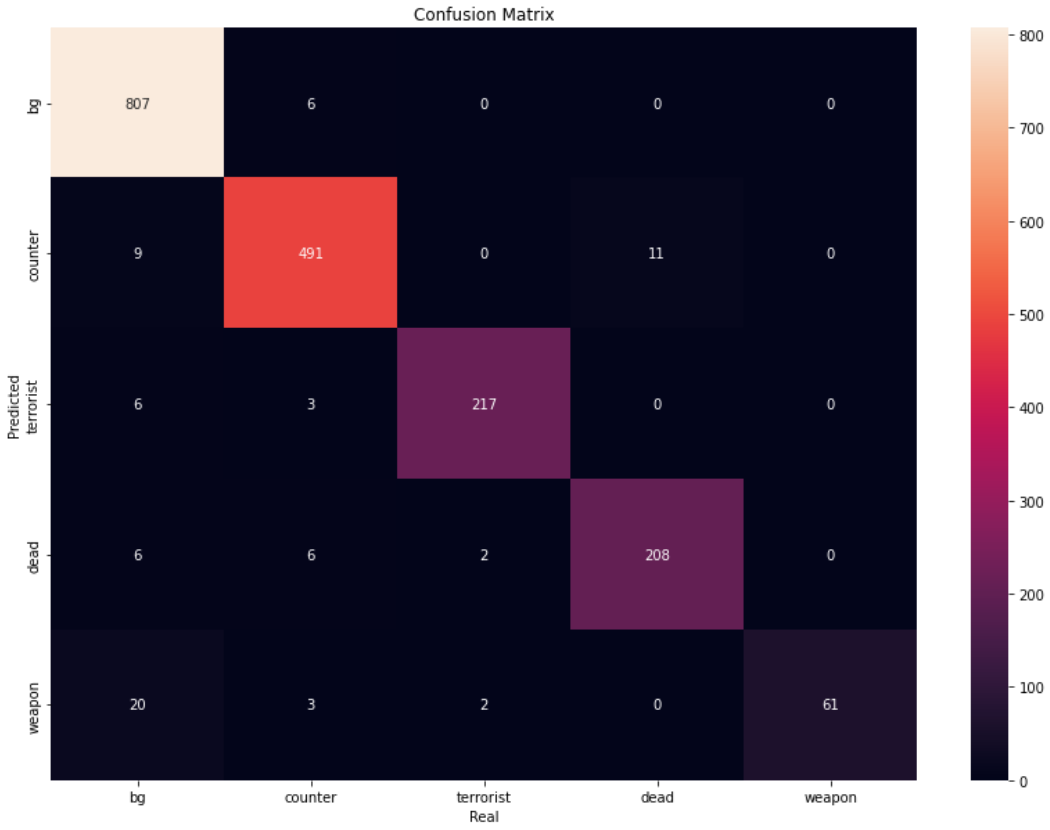


روش سوم (LBP)

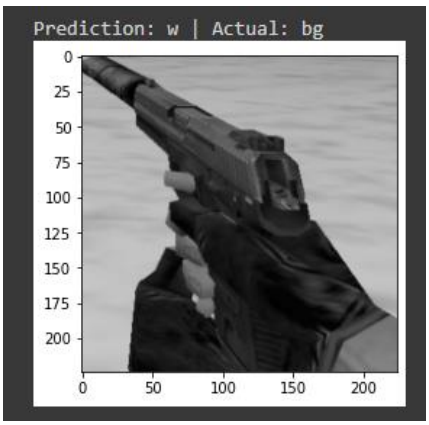
	precision	recall	f1-score	support
0	0.99	0.95	0.97	967
1	0.96	0.99	0.97	542
2	0.96	0.99	0.97	220
3	0.95	0.99	0.97	230
4	0.91	0.99	0.94	68
accuracy			0.97	2027
macro avg	0.95	0.98	0.97	2027
weighted avg	0.97	0.97	0.97	2027

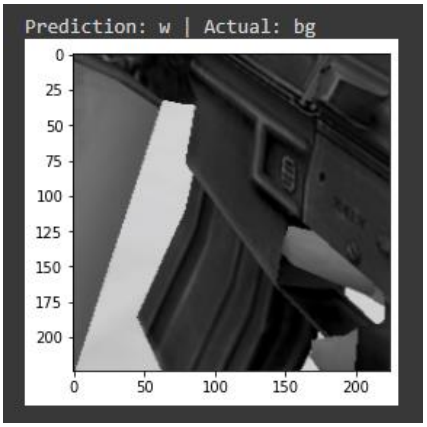


روش چهارم (شبکه‌عصبی)



توضیحاتی در مورد ماتریس درهم‌ریختگی: یکی از مشکلاتی که به چشم آمد، اشتباه بودن برچسب تعدادی از تصاویر کاندید بود. این برچسب‌ها بدون نظارت انسان و به صورت اتوماتیک بر اساس معیار IOU محاسبه شده اند و به همین دلیل دارای خطاهایی هستند. به عنوان مثال تصاویر زیر تصاویری از اسلحه هستند که در دیتاست برچسب بکگراند خورده اند. اگر به ماتریس بالا نگاه کنید میبینید که تعداد ۲۰ تصویر توسط مدل اسلحه تشخیص داده شده اند اما برچسب بکگراند داشته اند. بخش زیادی از این اشتباهات به دلیل برچسب‌های اشتباه بوجود آمده اند و دقت واقعی بالاتر است.

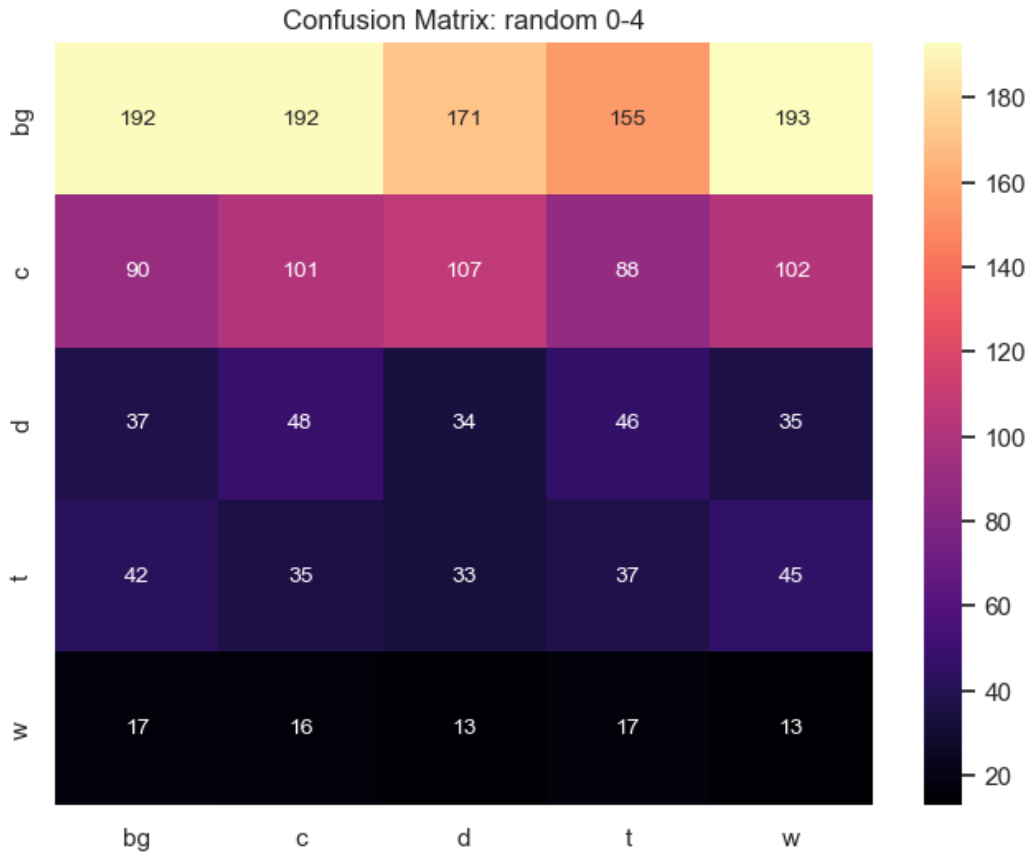




روش رندوم (baseline)

به عنوان یک مدل پایه، پیش‌بینی هایی به صورت رندوم انجام دادیم که دقت ۲۰ درصد را طبق انتظار(۵ کلاسه) به همراه دارد.

	precision	recall	f1-score	support
0	0.51	0.21	0.30	903
1	0.26	0.21	0.23	488
2	0.09	0.17	0.12	200
3	0.11	0.19	0.14	192
4	0.03	0.17	0.06	76
accuracy			0.20	1859
macro avg	0.20	0.19	0.17	1859
weighted avg	0.34	0.20	0.24	1859



شاید از نتایج چند صفحه قبل انتظار برود که این روش با دقت بسیار بالایی موقعیت تمامی اشیاء صحنه را تشخیص دهد و کلاس درست را به هر کدام نسبت دهد. اما این‌طور نیست ما در این سوال صرفا بخش Classification کار را انجام دادیم و سایر بخش‌های مهم کار از قبل انجام شده بود. به عنوان مثال بخش تشخیص ROI های تصویر و نسبت دادن برچسب به هر کدام از قبل در فایل dataset.npy تهیه شده بود و ما در تعیین این نواحی دستی نداشتیم. کار ما صرفا گرفتن یک تصویر ناحیه، استخراج ویژگی از آن و دسته‌بندی شئ موجود در آن ناحیه بود که در آن به دقت ۹۶ درصد رسیدیم.

در ادامه تعدادی از داده‌ها و برچسب‌های موجود در دیتاست این سوال را مشاهده میکنیم.

اسلحه در درست بازیکن برچسب نخورده. اما قاعدتا به دلیل شباهت زیاد به اسلحه، توسط Classifier اسلحه تشخیص داده میشود. دو کاراکتر در انتهای تصویر وجود دارند که برچسب نخورده اند اما به دلیل شباهت به کاراکترها توسط دسته‌بند، در یکی از دو دسته C یا T تشخیص داده می‌شوند.

تعدادی از نتایج بر روی داده تست

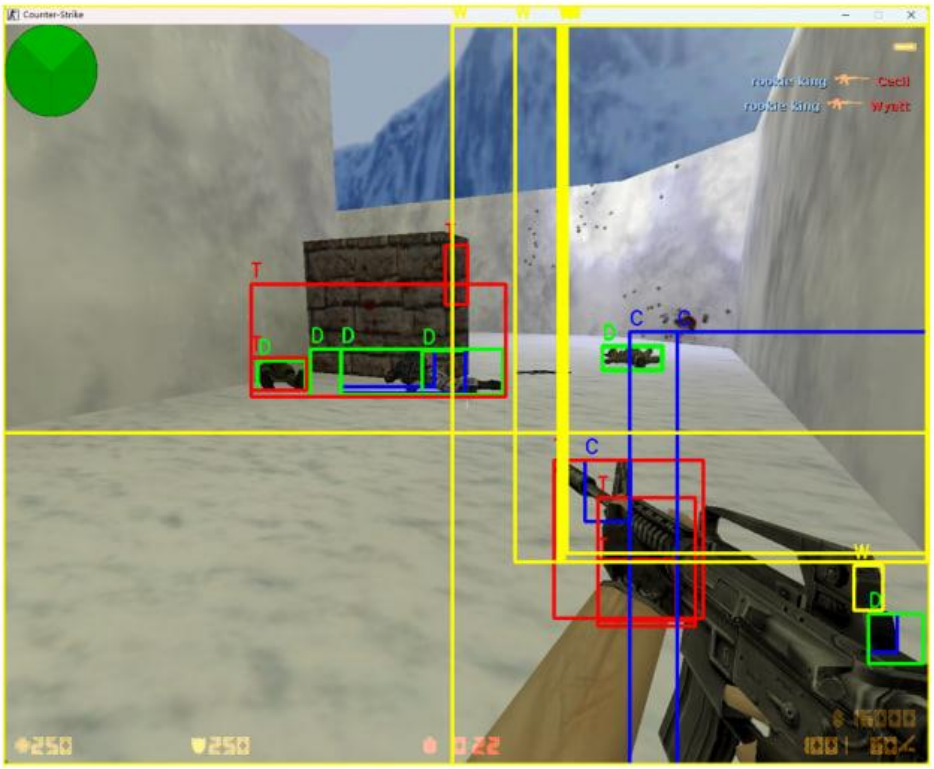

Counter-Strike
Calvin
Ferris
T
250
250
36

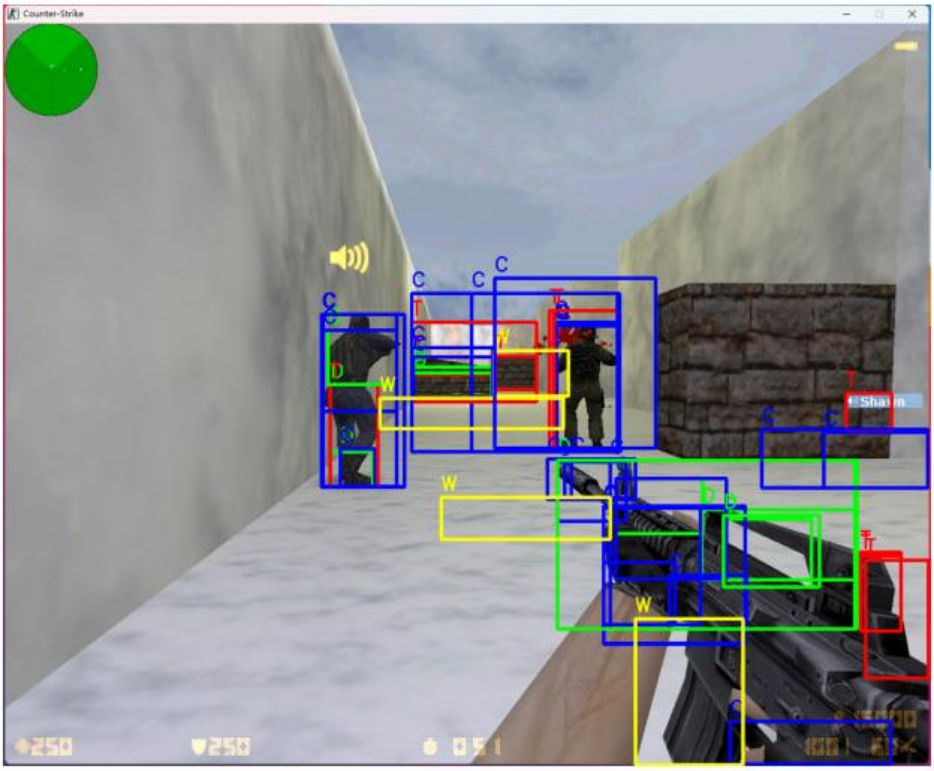

Counter-Strike
rookie king
Cecil
rookie king
Wyatt
T
C
D
D
250
250
22


Counter-Strike
T
T
C
250
250


Counter-Strike
Eugene
Clarence
Ferris
Shawn
rookie king
Ferris
Ron
Calvin
C
250
250

نکته در مورد نتایج: نتیجه دسته‌بندی بر روی خروجی Selective_Search مشابه تصاویر زیر است. برای بهبود نتیجه ناحیه های تولید شده بسیار باریک یا بسیار بزرگ یا بسیار کوچک را حذف میکنیم.





د)

از آنجایی که ممکن است ناحیه‌های بدست‌آمده توسط الگوریتم Selective_Search دقت کافی نداشته باشند و عمل Localization را به خوبی انجام ندهند، می‌توان با استفاده از Regression، ویژگی‌های استخراج شده ناحیه را گرفت و ۴ عدد به عنوان Offset برای ۴ x,y,w,h مستطیل بدست آمده توسط Selective_Search تخمین زد. با این ۴ عدد میتوان مستطیل ذکرشده را بهبود داد.

در این مقاله برای انجام این امر، صرفا از موقعیت مستطیل‌هایی استفاده می‌کنند که در کلاس مربوطه، حداقل یک مستطیل ground truth نزدیک آنها باشد. به این ترتیب، تنظیم کردن موقعیت مستطیل‌ها با تشخیص‌های اشتباه دچار گمراهی نمی‌شود.

which training pairs (P, G) to use. Intuitively, if P is far from all ground-truth boxes, then the task of transforming P to a ground-truth box G does not make sense. Using examples like P would lead to a hopeless learning problem. Therefore, we only learn from a proposal P if it is *nearby* at least one ground-truth box. We implement "nearness" by assigning P to the ground-truth box G with which it has maximum IoU overlap (in case it overlaps more than one) if and only if the overlap is greater than a threshold (which we set to 0.6 using a validation set). All unassigned proposals are discarded. We do this once for each object class in order to learn a set of class-specific bounding-box regressors.

این بخش در **کد پیاده‌سازی شد** اما دقت Regression بسیار پایین بود(شاید ویژگی‌های LBP برای این منظور اطلاعات کافی نداشته باشند زیرا انتظار میرود نوعی اطلاعات مکانی از پیکسل‌های تصویر در این ویژگی‌ها نهفته باشد. به عنوان مثال ویژگی‌های استخراج شده توسط شبکه عصبی می‌توانند در این امر موفق تر باشند.

```python
reg = LinearRegression()
# reg = SVR(kernel='rbf', C=1e3, gamma=0.1)
# reg = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=1, random_state=0, loss='squared_error')
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)
print(reg.score(X_test, y_test))
```

Python

```
0.11759292081264594
```

مقدار R2-Score=0.11 که بسیار پایین است.

هـ)

این الگوریتم در زمان تست، از بین تمام نواحی، آنهایی که با یک ناحیه دیگر که امتیاز بالاتری گرفته است، اشتراک زیادی داشته باشند را حذف می‌کند. به این شکل فقط مهم‌ترین bounding box ها باقی می‌مانند. در این مقاله این کار را براس هر کلاس شئ به طور مستقل انجام می‌دهند.

class. Given all scored regions in an image, we apply a greedy non-maximum suppression (for each class independently) that rejects a region if it has an intersection-over-union (IoU) overlap with a higher scoring selected region larger than a learned threshold.

پایان