



## درس شبکه های عصبی و یادگیری عمیق

### تمرین سوم

علی مرجبی داکتره	نام و نام خانوادگی	پرسش ۱
610300104	شماره دانشجویی	
رضا دلیر	نام و نام خانوادگی	پرسش ۲
610300050	شماره دانشجویی	
<b>1404/2/21</b>	مهلت ارسال پاسخ	

# فهرست

1	قوانين
1	پرسش 1- سگمنتیشن تصاویر شهری
1	1- مقدمه
1	2- توصیف مدل ارائه شده
1	Learning to Downsample .1
1	Global Feature Extractor .2
2	Feature Fusion Module .3
2	Classifier .4
4	3- مجموعه داده ها
6	4- معماری مدل
6	(بادگیری برای کاهش ابعاد) Learning to Downsample •
6	(استخراج ویژگی های سراسری) Global Feature Extractor •
7	Pyramid Pooling Module •
7	(ترکیب ویژگی ها) Feature Fusion Module •
8	(بخش نهایی، برای تولید سگمنتیشن) Classifier Head •
9	5- فرآیند آموزش مدل
9	1- آماده سازی داده ها
9	2- تقسیم مجموعه داده
9	3-تابع هزینه
9	4- بهینه ساز
9	5- تنظیم گر نرخ یادگیری یا Scheduler
9	6- آموزش مدل با هایبری یارامتر های مذکور
11	7- ارزیابی مدل
15	پرسش 2 : Oriented R-CNN برای تشخیص اشیاء
15	بخش اول: سوالات نظری
15	درک مفهومی:
16	اجزای مدل:
17	تفاوت Oriented RPN با RPN سنتی
19	نحوه انجام عملیات Rotated RoIAlign
19	گام اول: دریافت ورودی (جعبه مایل)
19	گام دوم: تبدیل به مستطیل چرخیدشده
20	گام سوم: Feature Map
20	گام چهارم: تقسیم به شبکه هی $m \times m$
20	گام پنجم: نمونه داری و محاسبه میانگین
21	عملکرد و کارایی:
23	بخش دوم: بیانه سازی عملی
25	CNN-R Oriented: آموزش مدل
28	ارزیابی و تحلیل نتایج:

قبل از پاسخ دادن به پرسش ها، موارد زیر را با دقت مطالعه نمایید:

- از پاسخ های خود یک گزارش در قالبی که در صفحه‌ی درس در سامانه‌ی Elearn با نام **REPORTS\_TEMPLATE.docx** قرار داده شده تهیه نمایید.
- پیشنهاد می‌شود تمرین‌ها را در قالب گروه‌های دو نفره انجام دهید. (بیش از دو نفر مجاز نیست و تحويل تک نفره نیز نمره‌ی اضافی ندارد) توجه نمایید الزاماً در یکسان ماندن اعضای گروه تا انتهای ترم وجود ندارد. (یعنی، می‌توانید تمرین اول را با شخص A و تمرین دوم را با شخص B و ... انجام دهید)
- **کیفیت گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است؛ بنابراین، لطفاً تمامی نکات و فرضهایی را که در پیاده‌سازیها و محاسبات خود در نظر می‌گیرید در گزارش ذکر کنید.**
- در گزارش خود مطابق با آنچه در قالب نمونه قرار داده شده، برای شکل‌ها زیرنویس و برای جدول‌ها بالانویس در نظر بگیرید.
- الزاماً به ارائه توضیح جزئیات کد در گزارش نیست، اما باید نتایج بدست آمده از آن را گزارش و تحلیل کنید.
- **تحلیل نتایج الزاماً می‌باشد، حتی اگر در صورت پرسش اشاره ای به آن نشده باشد.**
- دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند؛ بنابراین، هرگونه نتیجه و یا تحلیلی که در صورت پرسش از شما خواسته شده را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.
- **کدها حتماً باید در قالب نوت بوک با پسوند ipynb. تهیه شوند، در پایان کار، تمامی کد اجرا شود و خروجی هر سلول حتماً در این فایل ارسالی شما ذخیره شده باشد.** بنابراین برای مثال اگر خروجی سلولی یک نمودار است که در گزارش آورده اید، این نمودار باید هم در گزارش هم در نوت بوک کدها وجود داشته باشد.
- **در صورت مشاهده‌ی تقلب امتیاز تمامی افراد شرکتکننده در آن، ۱۰۰- لحظ می‌شود.**
- تنها زبان برنامه نویسی مجاز Python است.
- استفاده از کدهای آماده برای تمرینها به هیچ وجه مجاز نیست. در صورتی که دو گروه از یک منبع مشترک استفاده کنند و کدهای مشابه تحويل دهند، تقلب محسوب می‌شود.
- نحوه محاسبه تاخیر به این شکل است: پس از پایان رسیدن مهلت ارسال گزارش، حداقل تا یک هفته امکان ارسال با تاخیر وجود دارد، پس از این یک هفته نمره آن تکلیف برای شما صفر خواهد شد.
- سه روز اول: بدون جریمه
  - روز چهارم: ۵ درصد
  - روز پنجم: ۱۰ درصد
  - روز ششم: ۱۵ درصد
  - روز هفتم: ۲۰ درصد

- حداقل نمره ای که برای هر سوال می توان اخذ کرد ۱۰۰ بوده و اگر مجموع بارم یک سوال بیشتر از ۱۰۰ باشد، در صورت اخذ نمره بیشتر از ۱۰۰، اعمال نخواهد شد.
- برای مثال: اگر نمره اخذ شده از سوال ۱ برابر ۱۰۵ و نمره سوال ۲ برابر ۹۵ باشد، نمره نهایی تمرین ۹۷.۵ خواهد بود و نه ۱۰۰.
- لطفا گزارش، کدها و سایر ضمایم را به در یک پوشه با نام زیر قرار داده و آن را فشرده سازید، سپس در سامانه [Elearn](#) بارگذاری نمایید:

HW[Number]\_[Lastname]\_[StudentNumber]\_[Lastname]\_[StudentNumber].zip

(HW1\_Ahmadi\_810199101\_Bagheri\_810199102.zip) (مثال:

- برای گروه های دو نفره، بارگذاری تمرین از جانب یکی از اعضا کافی است ولی پیشنهاد می شود هر دو نفر بارگذاری نمایند.

## پرسش 1- سگمنتیشن تصاویر شهری

### 1.1- مقدمه

یکی از مسائل مهم در حوزه‌ی بینایی ماشین است که در آن هدف، برچسب‌گذاری پیکسل به پیکسل تصویر ورودی بر اساس کلاس‌های معنایی مشخص است. این مسئله در کاربردهایی مانند رانندگی خودران، پژوهشی، نقشه‌برداری و پایش محیطی اهمیت دارد.

در این پژوهه، ما از معماری Fast-SCNN برای انجام semantic segmentation استفاده کردیم که یکی از معماری‌های سریع و سبک مناسب است. هدف اصلی این پژوهه، آموزش یک مدل Fast-SCNN برای انجام قطعه‌بندی معنایی بر روی یک دیتابس مشخص و ارزیابی عملکرد آن با معیارهای رایج از جمله loss، میانگین دقت IoU، ضریب Dice و دقت کلی (accuracy) است.

### 1.2- توصیف مدل ارائه شده

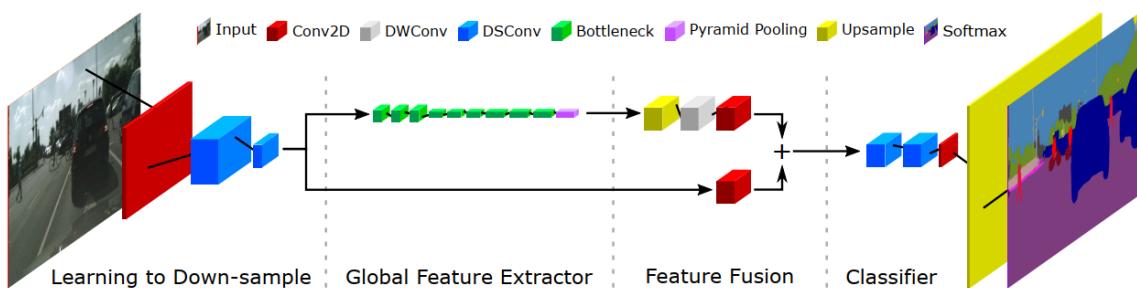


Figure 1. Fast-SCNN shares the computations between two branches (encoder) to build a above real-time semantic segmentation network.

معماری Fast-SCNN از سه بخش اصلی تشکیل شده است:

#### Learning to Downsample .1

- شامل چند لایه‌ی Convolution و Depthwise Separable Convolution است.
- قابلیت کاهش ابعاد تصویر ورودی با کمترین محاسبه ممکن و استخراج ویژگی‌های اولیه.
- قابلیت سریع کردن محاسبات و آمده‌سازی تصویر برای لایه‌های بعدی.

#### Global Feature Extractor .2

- شامل لایه‌های Pyramid Pooling Module مشابه Bottleneck و یک MobileNetV2 است.
- استخراج ویژگی‌های سراسری و عمیق از تصویر.
- نقش: درک اطلاعات زمینه‌ای و سطوح بالا، افزایش درک معنایی مدل.

### **Feature Fusion Module .3**

- ترکیب خروجی Global Feature Extractor (ابعاد پایین) با ویژگی‌های Learning to Downsample (ابعاد بالاتر).
- از تکنیک‌های upsampling و concatenation استفاده می‌شود.
- نقش: ترکیب اطلاعات دقیق مکانی (از لایه‌های ابتدایی) با اطلاعات معنایی (از لایه‌های عمیق).

### **Classifier .4**

- لایه‌های نهایی convolution برای پیش‌بینی کلاس هر پیکسل.
- معمولاً شامل یک لایه Conv 1x1 برای کاهش به تعداد کلاس‌ها و سپس upsample به ابعاد اصلی تصویر.

برای مقایسه آن با مدل U-net می‌توان ویژگی‌های آنها را در جدول زیر، رو بروی هم مشاهده کرد:

U-Net	Fast-SCNN	ویژگی
encoder-decoder	light-weight	نوع معماری
دقت بالا، مخصوصاً در تصاویر پزشکی	سرعت بالا	هدف اصلی
زیاد (وابسته به عمق شبکه)	بسیار کم	مقدار پارامترها
(encoder) چند مرحله	فقط یک بار انجام می‌شود	Downsampling
skip connection چند مرحله با	یکبار نهایی	Upsampling

skip connection در هر سطح	ترکیب از یک نقطه feature (fusion)	استفاده از skip
حفظشده از طریق skip ها	کمتر	اطلاعات مکانی دقیق
نه در حالت عادی، مگر با سادهسازی	بله	قابل استفاده برای real-time

در معماری Fast-SCNN، برای رسیدن به عملکرد سریع در کنار حفظ دقت قابل قبول، از مجموعه‌ای از ساختارهای مدرن و بهینه‌شده در طراحی بلاک‌های اصلی استفاده شده است. این ساختارها به طور ویژه با هدف کاهش پیچیدگی محاسباتی و استفاده مؤثر از منابع طراحی شده‌اند، و در عین حال اجازه می‌دهند که مدل بتواند ویژگی‌های مهم درون تصویر را با کیفیت مناسب استخراج کند. در ادامه، شرح تشریحی هر یک از این ساختارها آمده است.

یکی از اجزای اصلی بهکار رفته در این معماری Depthwise Separable Convolution است. این ساختار نوعی کانولوشن سبک‌شده محسوب می‌شود که به جای اعمال یک فیلتر دو بعدی به صورت همزمان روی تمامی کanal‌های تصویر (که در کانولوشن معمولی اتفاق می‌افتد)، ابتدا یک کانولوشن مجزا به ازای هر کanal (که به آن Depthwise Convolution گفته می‌شود) انجام می‌دهد. در این مرحله، فیلترها فقط اطلاعات فضایی را از یک کanal استخراج می‌کنند و هیچ ترکیبی میان کanal‌ها انجام نمی‌گیرد. سپس در مرحله‌ی دوم، یک کانولوشن  $1 \times 1$  با نام Pointwise Convolution بر خروجی‌ها اعمال می‌شود تا بتواند بین کanal‌های استخراج شده ارتباط برقرار کرده و اطلاعات ترکیبی مورد نظر را ایجاد کند. این روش باعث کاهش چشمگیر تعداد پارامترها و محاسبات نسبت به کانولوشن استاندارد می‌شود، در حالی که بخش عمده‌ای از توانایی استخراج ویژگی حفظ می‌شود.

در ادامه، Inverted Residual Block که نخستین بار در MobileNetV2 معرفی شد، یکی از هسته‌های ساختاری مهم در Fast-SCNN بهشمار می‌رود. برخلاف بلاک‌های سنتی residual که ابتدا ویژگی‌ها را فشرده می‌کرند و سپس توسعه می‌دادند، در اینجا ابتدا تعداد کanal‌ها توسط یک کانولوشن  $1 \times 1$  افزایش می‌یابد (مرحله‌ی expansion)، سپس یک Depthwise Convolution بر روی این فضای گسترش‌یافته انجام می‌گیرد تا ویژگی‌های فضایی استخراج شود، و در نهایت، با یک کانولوشن  $1 \times 1$  دیگر، ابعاد به سطح اولیه بازگردانده می‌شود (مرحله‌ی projection). اگر ابعاد ورودی و خروجی این بلاک برابر باشند، از یک مسیر میان‌بر (skip connection) نیز استفاده می‌شود تا اطلاعات اولیه حفظ شده و گرادیان در طول آموزش بهتر جریان یابد. این ساختار هم از لحاظ محاسباتی بهینه است و هم اجازه می‌دهد مدل به شکل عمیق‌تری بدون انفجار یا ناپدید شدن گرادیان‌ها آموزش بییند.

از سوی دیگر، برای اینکه مدل بتواند اطلاعات زمینه‌ای (context) را از بخش‌های بزرگ‌تر تصویر برداشت کند، از Pyramid Pooling Module بهره گرفته می‌شود. این مازول در مرحله‌ای از شبکه که ویژگی‌ها در سطح پایین‌تر اما با اطلاعات معنایی غنی‌تر هستند، فعال می‌شود. در اینجا، تصویر یا feature map به بخش‌هایی با اندازه‌های مختلف (مثل  $1 \times 1$ ،  $2 \times 2$ ،  $3 \times 3$  و  $6 \times 6$ ) تقسیم می‌شود و از هر بخش میانگین‌گیری صورت می‌گیرد (با استفاده از Average Pooling). سپس خروجی‌های این pooled feature maps از طریق کانولوشن  $1 \times 1$  به ابعاد موردنیاز رسانده شده و به اندازه تصویر اصلی upsample می‌شوند. در نهایت همه‌ی این لایه‌ها به همراه feature map اولیه در راستای کanal‌ها به هم متصل می‌شوند. هدف از این مازول آن است که مدل بتواند هم

اطلاعات محلی (local) و هم اطلاعات گسترده (global) را درک کند، بهخصوص در نواحی‌ای که اشیای مختلف ممکن است از نظر بافت شباهت زیادی به یکدیگر داشته باشند ولی در زمینه‌های متفاوتی قرار گرفته‌اند.

در این بین، نقش Depthwise Convolution به عنوان یک عنصر پایه‌ای در بسیاری از مراحل شبکه بسیار پرنگ است. همان‌طور که اشاره شد، این نوع کانولوشن به تنها یک بر کانال اعمال می‌شود و برای استخراج ویژگی‌های فضایی درون‌کانالی استفاده می‌گردد. چون هیچ ارتباطی بین کانال‌ها در این مرحله برقرار نمی‌شود، عملیات بسیار سریع‌تر است. البته به تنها یک ناقص است و باید با Pointwise ترکیب شود تا شبکه قدرت استخراج اطلاعات ترکیبی را داشته باشد.

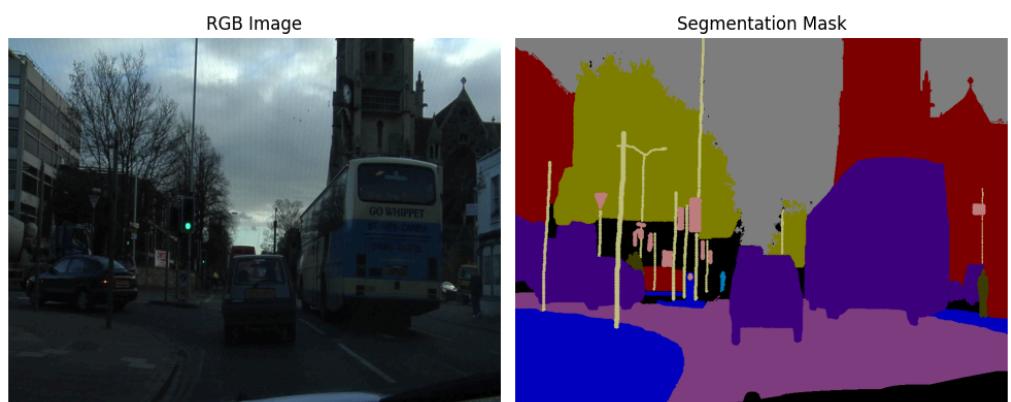
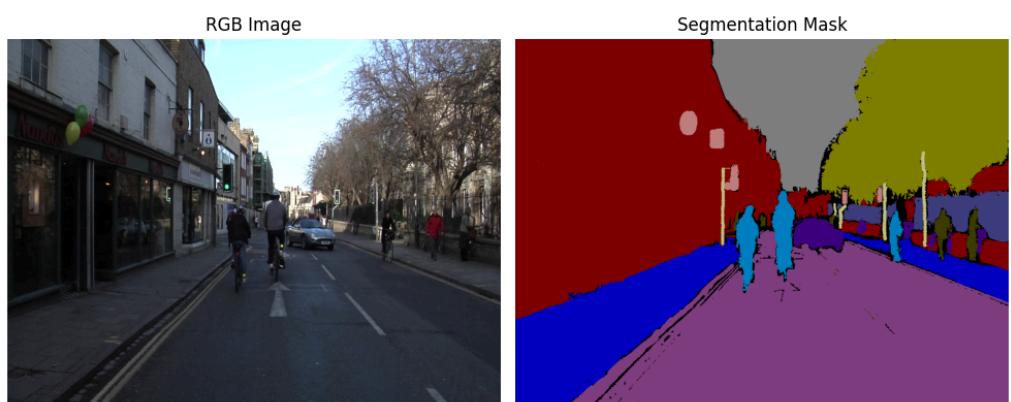
در مجموع، Fast-SCNN با بهکارگیری هوشمندانه این ساختارها موفق شده است تعادلی بین دقیق و سرعت ایجاد کند و بتواند برای کاربردهای real-time مانند پردازش تصویر روی گوشی‌های همراه یا ربات‌ها بسیار مناسب باشد.

### ۱.۳- مجموعه داده‌ها

دیتابست مورد استفاده شامل تصاویر RGB و ماسک‌های متاظر است که هر ماسک دارای برچسب‌هایی برای کلاس‌های مختلف می‌باشد. این کلاس‌ها به صورت عددی مشخص شده‌اند، که در نهایت هم متریک IoU برای هر کلاس قابل مشاهده است.

Class	Semanticity
0	Bicyclist
1	Building
2	Car
3	Fence
4	Pedestrian
5	Pole
6	Road
7	Sidewalk
8	SignSymbol
9	Sky
10	Tree
11	Void

با نگاه انداختن به عکس‌های زیر، می‌توان درکی شهودی از مجموعه داده‌های موجود در مجموعه داده داشت:



## ۱.۴- معماری مدل

### (یادگیری برای کاهش ابعاد) Learning to Downsample •

این بخش برای کاهش اندازه‌ی تصویر و استخراج ویژگی‌های پایه‌ای استفاده می‌شود و شامل سه لایه‌ی پیاپی است:

خروجی	Kernel / Stride / Padding	Channel	نوع لایه	لایه
(H/2, W/2, 32)	3x3 / stride=2 / padding=1	3 → 32	Conv2D (معمولی)	1
(H/4, W/4, 48)	3x3 / stride=2 / padding=1	32 → 48	Depthwise Separable Conv2D	2
(H/8, W/8, 64)	3x3 / stride=2 / padding=1	48 → 64	Depthwise Separable Conv2D	3

### (استخراج ویژگی‌های سراسری) Global Feature Extractor •

این بخش با استفاده از ساختار مشابه Pyramid Pooling Module و MobileNetV2 Bottlenecks ویژگی‌های عمیق و معنایی تصویر را استخراج می‌کند.

تکرار	Stride	Expansion	Channels	Block
بار 3	2	×6	64 → 64	1
بار 3	2	×6	64 → 96	2
بار 3	1	×6	96 → 128	3

● bottleneck هر شامل:

○ Expand  $1 \times 1$  conv

○ Depthwise conv  $3 \times 3$

○ Projection  $1 \times 1$  conv

○ residual connection

### ● Pyramid Pooling Module

PPM از چند مسیر موازی با مقیاس‌های مختلف استفاده می‌کند:

Upsample	بعد از	Output Size	Pooling Size	مسیر
H/8 × W/8		1×1	1×1	1
H/8 × W/8		2×2	2×2	2
H/8 × W/8		3×3	3×3	3
H/8 × W/8		6×6	6×6	4

سپس همهی خروجی‌ها concatenate می‌شوند و از یک convolution  $1 \times 1$  کاهش بعد استفاده می‌شود.

### ● Feature Fusion Module (ترکیب ویژگی‌ها)

در این بخش، ویژگی‌های سطح بالا از Global Feature Extractor با ویژگی‌های سطح پایین از Downsample ترکیب می‌شوند:

توضیح	عملیات	مرحله
با bilinear به اندازه‌ی $(H/8)$	Upsample ویژگی‌های عمیق	1
بمنظور هماهنگی در ابعاد	سطح پایین channel برای کاهش $1 \times 1$ Conv	2
ترکیب معنایی و مکانی	ویژگی‌ها concatenate جمع یا	3
پردازش ویژگی ترکیبی	Conv $3 \times 3$ (Separable)	4

هدف، بهره‌گیری از دقت موقعیتی ویژگی‌های اولیه و اطلاعات معنایی ویژگی‌های عمیق.

#### • (بخش نهایی برای تولید سگمنتیشن Classifier Head)

لایه	نوع لایه	Channel‌ها	خروجی
1	Depthwise Separable Conv2D	$128 \rightarrow 128$	$H/8, W/8$
2	Depthwise Separable Conv2D	$128 \rightarrow 128$	$H/8, W/8$
3	Conv $1 \times 1$	$n\_classes \rightarrow 128$	$H/8, W/8$
4	Upsample	$H/8 \rightarrow H, W$	خروجی نهایی

خروجی نهایی، یک Segmentation Map با اندازه برابر با تصویر اصلی، که در آن هر پیکسل به یک کلاس تخصیص داده می‌شود.

## 1.5- فرآیند آموزش مدل

در ادامه مراحل اصلی پیاده‌سازی، تنظیمات مدل، و فرآیند آموزش بهصورت کامل آورده شده است.

### 1.4.1- آماده سازی داده ها

#### ● بارگذاری تصاویر و ماسک ها

تصاویر RGB با اندازه اصلی (480×360) خوانده شدند.

#### ● تبدیل رنگی به شناسه های کلاس

هر پیکسل رنگی ماسک به یک عدد کلاس (NUM\_CLASSES-1...0) نگاشت شد.

#### ● و نرمال سازی Resize

```
self.img_transform = transforms.Compose([
    transforms.Resize(img_size, interpolation=Image.BILINEAR),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
```

ماسک ها با `Image.NEAREST` تغییر اندازه یافتد تا ایندکس های کلاس صحیح باقی بمانند.

### 1.4.2- تقسیم مجموعه داده

90 درصد داده ها به سمت آموزش و 10 درصد به سمت ارزیابی مدل اختصاص یافتند.

### 1.4.3- تابع هزینه

از `CrossEntropyLoss` بدون `ignore_index` (چون همه پیکسل ها دارای کلاس معنبر بودند) استفاده شد

### 1.4.4- بهینه ساز

از Adam با نرخ یادگیری اولیه 0.001 بهره برده شد.

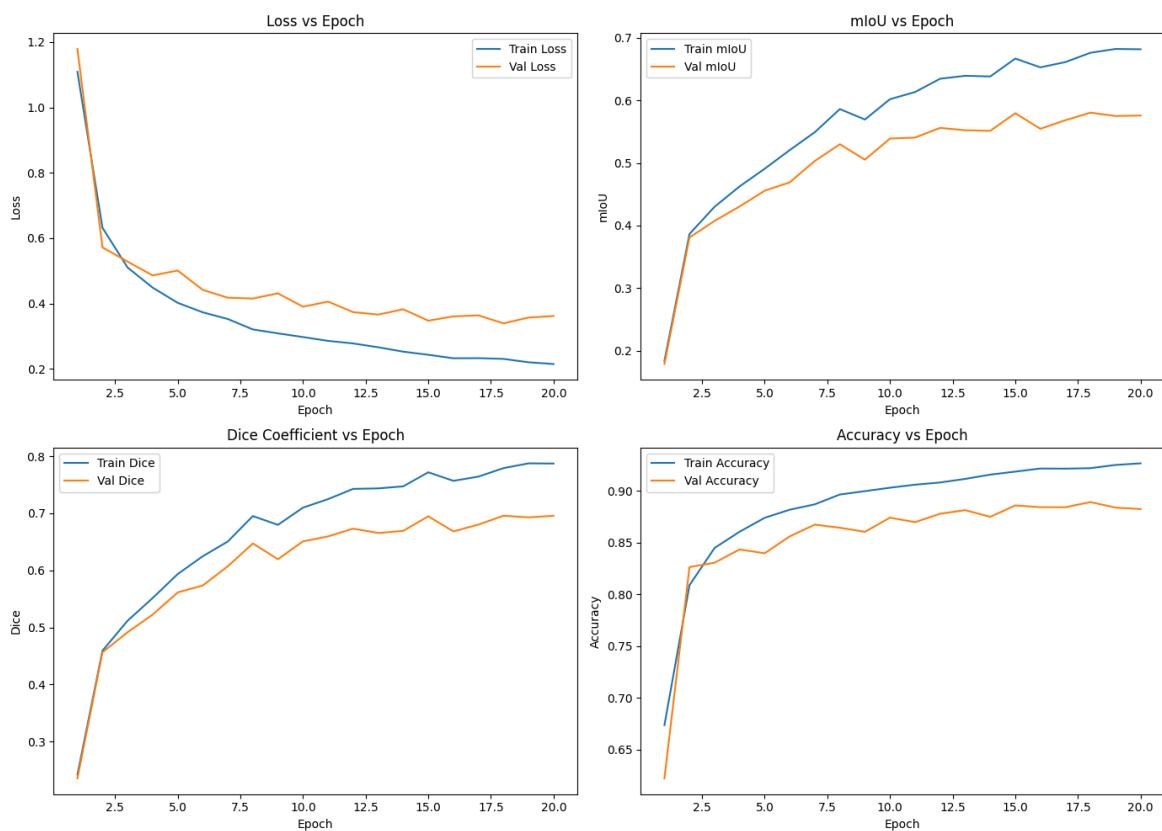
### 1.4.5- تنظیم گر نرخ یادگیری یا Scheduler

برای کاهش تدریجی نرخ یادگیری هر 20 epoch، از StepLR با فاکتور  $\gamma=0.5$  استفاده شد

### 1.4.6- آموزش مدل با هایپر پارامتر های مذکور

هایپر پارامتر	مقدار
epochs	70
batch size	8
learning rate	$1e-4$
scheduler step size	20
$\gamma$ (gamma)	0.5

پس از آموزش، نتایج متریک ها در طول epoch ها به صورت زیر قابل مشاهده هستند.

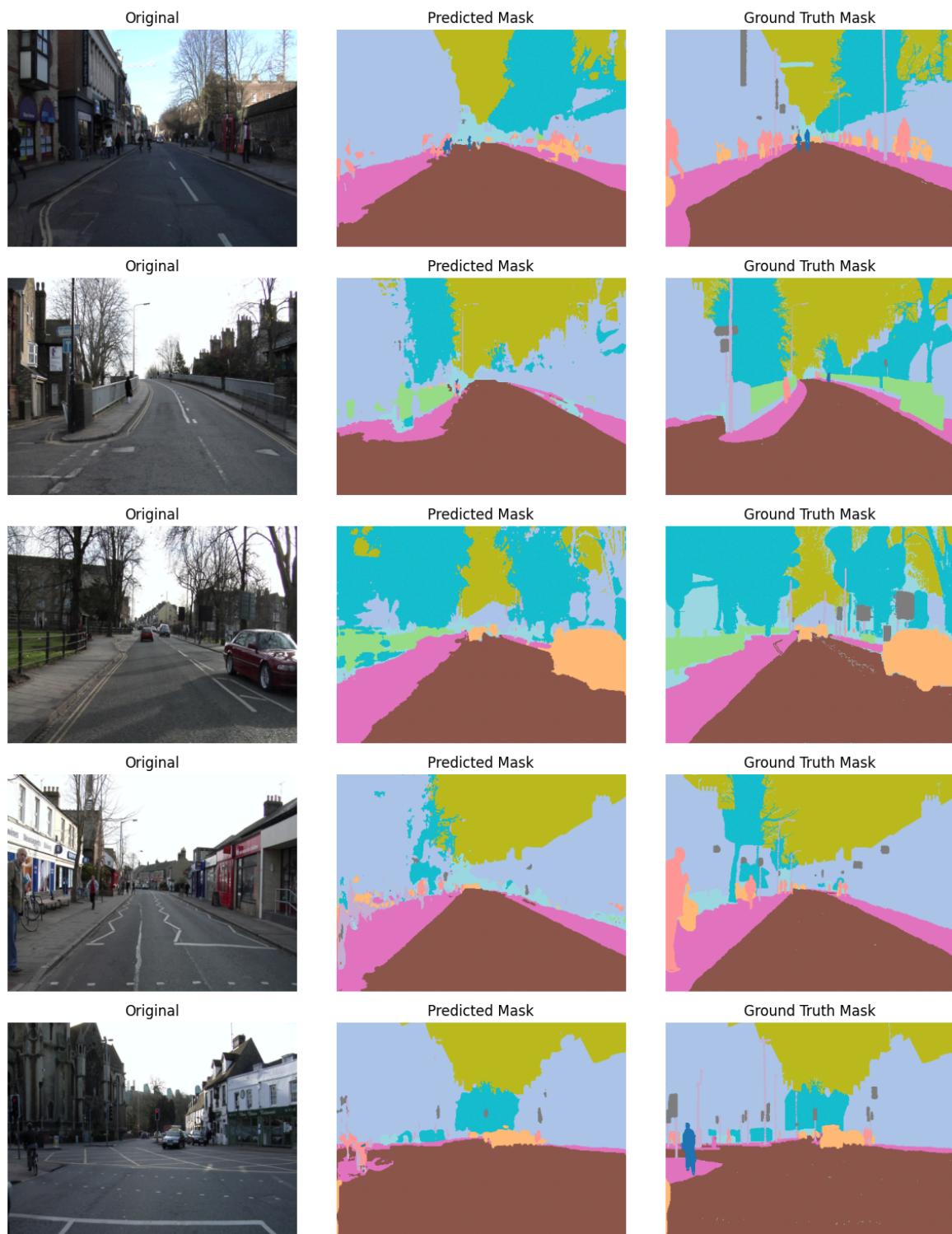


روندهای هزینه و صعودی متریک های ارزیابی مدل (هر دو با شبیه زیاد شروع و در ادامه با کاهش شبیه رو برو می شوند که کاملا طبیعی و درست است)، نشان از مناسب بودن پارامتر ها و عدم وجود مشکل در روند آموزش دارد.

## ۱.۶- ارزیابی مدل

پس از اتمام فرآیند آموزش و ثبت معیارهای مختلف، مدل را روی مجموعه ارزیابی (۱۰٪ داده‌ها) سنجیدیم و نتایج زیر به دست آمد:

Some Test Samples





مقدار validation loss به میزان 0.3618 است.

مقدار Mean IoU هم نیز، 0.5760 است.

تفسیر	IoU	نام کلاس	اندیس کلاس
عملکرد متوسط؛ دوچرخه‌سواران کم حجم و متحرک هستند.	0.5456	Bicyclist	0
عملکرد خوب؛ ساختمان‌ها اجسام بزرگ و ثابت‌اند.	0.7784	Building	1
عملکرد مناسب؛ خودروها روی جاده مشخص‌اند.	0.7218	Car	2
ضعیفتر؛ حصارها باریک و پس‌زمینه پیچیده دارند.	0.4395	Fence	3
ضعیف؛ آدم‌ها کوچک و دستخورده سایه و نور هستند.	0.2988	Pedestrian	4
پایین‌ترین عملکرد؛ تیرک‌ها نازک و اغلب محو شده.	0.1449	Pole	5
بهترین؛ جاده بخش غالب تصویر است و رنگ یکنواخت دارد.	0.9383	Road	6
خوب؛ پیاده‌روها معمولاً مجزا و ثابت‌اند.	0.7501	Sidewalk	7
متوسط رو به پایین؛ علائم کوچک و رنگ‌های مشابه پس‌زمینه.	0.3102	SignSymbol	8
بسیار خوب؛ آسمان بخش گسترده و یکنواخت تصویر است.	0.9196	Sky	9
مناسب؛ درخت‌ها حجم و بافت غنی دارند.	0.7226	Tree	10
بد؛ بخش‌های نامشخص یا مناطق بدون کلاس واضح.	0.3415	Void	11

اجسام بزرگ و با رنگ یکنواخت (جاده، آسمان، ساختمان) بالاترین IoU را دارند ( $> 0.75$ ) زیرا فضای زیادی از تصویر را پوشش می‌دهند و الگوهای رنگی ثابتی دارند.

اجسام کوچک یا باریک (تیرک‌ها، انسان‌ها، علائم) کمترین IoU را دارند. دلیل اصلی آن‌ها حجم کم، پوشش سایه/نور و آمیختگی با پس‌زمینه است.

کلاس Fence (حصار) هم به دلیل خطوط باریک و پس‌زمینه پیچیده ضعیف عمل کرده است.

از بین چند نمونه‌ی تصادفی نشان داده شده:

- **جاده (قهوه‌ای تیره)**: در همه تصاویر به خوبی جدا شده است.
  - **آسمان (آبی روشن)**: ناحیه‌ی آسمان به درستی و بدون نویز مشخص است.
  - **پیاده‌رو و ساختمان**: اغلب بخش‌ها دقیقاً با مرز واضح جدا می‌شوند.
  - **عابران پیاده و تیرک‌ها**: برخی موارد حذف یا با پس زمینه ترکیب شده‌اند.
- مثلًا در نمونه‌ی اول، تیرک و چند عابر کوچک در پیش‌بینی حذف شدند اما بخش‌های جاده و آسمان با وحدت رنگی قوی شناسایی شده‌اند. در نمونه‌ی چهارم نیز برخی علائم ترافیکی کوچک از دست رفته‌اند.
- در کل، میانگین  $IoU$  حدود **0.576** و Dice حدود **0.67** نشان‌دهنده‌ی یادگیری مطلوب مدل است و با اعمال نمونه هایی از متد‌های بهبود دهنده می‌توان به نتایج بهتری دست یافت.

## پرسش ۲. برای تشخیص اشیاء Oriented R-CNN :

### بخش اول: سوالات نظری

درک مفهومی:

الف. انگیزه اصلی توسعه CNN-R Oriented را توضیح دهد. این مدل چه محدودیتهایی از روش‌های قبلی را برطرف میکند؟ مثالهایی ارائه کنید.

انگیزه اصلی توسعه مدل Oriented R-CNN برطرف کردن محدودیت‌های محاسباتی و کارایی روش‌های پیشین در حوزه تشخیص اجسام چرخیده شده بود. در کاربردهایی مانند تصاویر ماهواره‌ای که در آن‌ها باید کشتی‌ها یا هواپیماهایی که با زاویه‌های مختلف در تصویر ظاهر می‌شوند تشخیص داده شوند، مدل‌های قبلی مانند Rotated RPN از صدها anchor با زوایا و اندازه‌های مختلف استفاده می‌کردند. این روش اگرچه دقت بالایی داشت، اما بهشدت زمانبر و سنگین از نظر حافظه بود. همچنین روش‌هایی مانند RoI Transformer با تبدیل باکس‌های افقی به باکس‌های چرخیده، تعداد anchor‌ها را کاهش می‌دادند اما همچنان مراحل تبدیل بسیار پیچیده‌ای دارند و هزینه‌ی محاسباتی آن به نسبت بالاست. مدل Oriented R-CNN با ارائه شبکه‌ای سبک به نام Oriented RPN تولید مستقیم نواحی چرخیده را فراهم می‌کند و با استفاده از روش نمایش "midpoint offset representation"، تنها با شش پارامتر قادر است جهت و بعد جسم را بدقت نمایش دهد. این روش تعداد پارامتر‌ها و پیچیدگی شبکه را بسیار کاهش میدهد. برای مثال، در تصویری از یک بندرگاه که کشتی‌ها در زوایای مختلفی پهلو گرفته‌اند، این مدل می‌تواند بدون نیاز به صدها anchor و با سرعت بالا آن‌ها را به درستی تشخیص دهد.

ب. مزایای استفاده از نمایش "offset midpoint" نسبت به نمایشهای سنتی جعبه‌های محدودکننده را توضیح دهد. مثالهایی برای روشن شدن توضیحات خود ارائه کنید.

نمایش "offset midpoint" که در مدل Oriented R-CNN معرفی شده، مزایای قابل توجهی نسبت به نمایشهای سنتی جعبه‌های محدودکننده (bounding boxes) دارد. در روش‌های سنتی، معمولاً از چهار پارامتر (مرکز، عرض، ارتفاع، و زاویه) یا مختصات گوشش‌های جعبه برای نمایش اشیاء استفاده می‌شود، اما این نمایش‌ها در مواجهه با اشیاء چرخیده شده پیچیدگی‌هایی در یادگیری و پیش‌بینی ایجاد می‌کنند. در مقابل، نمایش "offset midpoint" با استفاده از شش پارامتر، یک راه حل ساده و مؤثر برای نمایش دقیق اشیای با جهت‌گیری دلخواه ارائه می‌دهد. این روش نه تنها قابلیت استفاده مجدد از ساختارهای موجود در RPN سنتی را حفظ می‌کند، بلکه سازگاری بهتری با عماری‌های یادگیری عمیق دارد و ریسک overfitting را کاهش می‌دهد. از همه مهمتر، این نمایش امکان یادگیری پایدارتر و محدودسازی بهتر جعبه‌های پیشنهادی را فراهم می‌کند. ایده اصلی این است که به جای استفاده از هایی با زوایای مختلف، از "midpoint offset" استفاده شود که وظیفه یادگیری زاویه و چرخش را به شبکه می‌سپارد. یعنی مدل یاد می‌گیرد که چگونه این anchor‌های ساده‌ی افقی را با استفاده از 6 پارامتر به جعبه‌های چرخیده با زاویه دلخواه تبدیل کند. این 6 پارامتر شامل مختصات مرکز جعبه، عرض، ارتفاع، پارامترهایی برای تخمین زاویه چرخش هستند و به این صورت، مدل به جای جستجو در میان anchor‌های با anchor زوایای مختلف، خوش زاویه‌ی مناسب را یاد می‌گیرد و اعمال می‌کند. این باعث کاهش چشمگیر هزینه محاسباتی و حافظه مصرفی می‌شود. به عنوان مثال، در یک تصویر ماهواره‌ای، چندین کشتی در یک بندرگاه در زوایای مختلفی نسبت به دوربین قرار گرفته‌اند. در روش‌های قدیمی، برای پوشش همه زوایا باید دهها anchor در هر نقطه قرار داده شود (مثلًا 54 انکر در Rotated RPN)، که موجب مصرف بالای حافظه و کاهش سرعت مدل می‌شود. اما با استفاده از نمایش midpoint offset، مدل می‌تواند با تنها شش پارامتر، موقعیت و زاویه دقیق هر کشتی را بدون نیاز به anchor‌های زیاد و پیچیدگی‌های محاسباتی شناسایی کند.

### اجزای مدل:

الف. معماری RPN Oriented را شرح دهید و تفاوت آن را با RPN سنتی توضیح دهید. برای روشن تر شدن توضیحات خود از نمودار یا طرح استفاده کنید.

معماری Oriented RPN در سیستم Oriented R-CNN برای شناسایی اشیاء که به صورت متمایل در تصاویر حضور دارند، مناسب است.

### معماری Oriented RPN

#### 1. ورودی ویژگی‌ها:

- ورودی Oriented RPN مجموعه‌ای از ویژگی‌های استخراج شده از FPN (Feature Pyramid Network) است که شامل پنج سطح ویژگی {P2, P3, P4, P5, P6} است.
- هر یک از این سطوح ویژگی، به همراه یک head با طراحی مشابه که از یک لایه کانولوشنی  $3 \times 3$  و دو لایه کانولوشن  $1 \times 1$  تشکیل شده، به Oriented RPN متصل می‌شود.

#### 2. oriented proposals

- برای هر مکان در نقشه ویژگی، سه anchor با نسبت‌های مختلف {1:1, 1:2, 2:1} به هر سطح ویژگی اختصاص داده می‌شود.
- هر anchor دارای مختصات (ax, ay) برای مرکز آن و ابعاد aw و ah برای عرض و ارتفاع است. به عبارت دیگر، یک anchor به صورت یک وکتور ۴ بعدی تعریف می‌شود:  $a = (ax, ay, aw, ah)$ .

#### 3. رگرسیون برای هر proposal

- یک لایه کانولوشن  $1 \times 1$  به عنوان شاخه رگرسیون عمل می‌کند و offset ها را برای هر proposal نسبت به anchors پیش‌بینی می‌کند.
- جابجایی‌ها شامل  $\delta = (\delta x, \delta y, \delta w, \delta h, \delta \alpha, \delta \beta)$  هستند که به ترتیب جابجایی‌های افقی و عمودی برای مختصات و ابعاد proposal ها و همچنین جابجایی‌های زاویه‌ای برای گشتاورهای متمایل از anchors را مشخص می‌کنند.

#### 4. فرمول رگرسیون:

این فرمول جابجایی‌های مختصات و ابعاد proposal ها را محاسبه کرده و oriented proposals را ایجاد می‌کند.

## تفاوت RPN با Oriented RPN

در RPN سنتی، پیشنهادات به طور معمول به مسیله horizontal anchors ایجاد می‌شوند که شامل جعبه‌های مرزی مستطیلی هستند. این anchors تنها می‌توانند به طور دقیق اشیاء را در تصاویر با جعبه‌های مرزی افقی شبیه‌سازی کنند و برای شناسایی اشیاء متمایل مناسب نیستند.

Oriented RPN به طور خاص به شناسایی oriented bounding boxes طراحی شده است. این معماری به جای استفاده از anchors افقی، از anchors چرخانده شده با زاویه‌های مختلف برای شبیه‌سازی جعبه‌های مرزی متمایل استفاده می‌کند. علاوه بر این، با استفاده از جابجایی‌های زاویه‌ای  $\alpha$  و  $\beta$ ، Oriented RPN قادر به محاسبه جابجایی‌های متمایل برای هر پیشنهاد است که می‌تواند اشیاء چرخیده و متمایل را شبیه‌سازی کند.

در Oriented RPN به دلیل استفاده از rotated anchors، تعداد anchors مورد نیاز برای هر موقعیت به سه عدد کاهش یافته است (در مقایسه با تعداد زیاد anchors در RPN سنتی). این کاهش تعداد anchors باعث کاهش محاسبات و بهینه‌تر شدن مصرف حافظه می‌شود.

در RPN سنتی، رگرسیون برای پیش‌بینی مختصات جعبه‌های مرزی افقی و ابعاد آن‌ها انجام می‌شود. در Oriented RPN، رگرسیون به طور مشابه انجام می‌شود، اما علاوه بر ابعاد و مختصات، جابجایی‌های زاویه‌ای برای شبیه‌سازی جعبه‌های مرزی متمایل نیز در نظر گرفته می‌شود.

Input Image



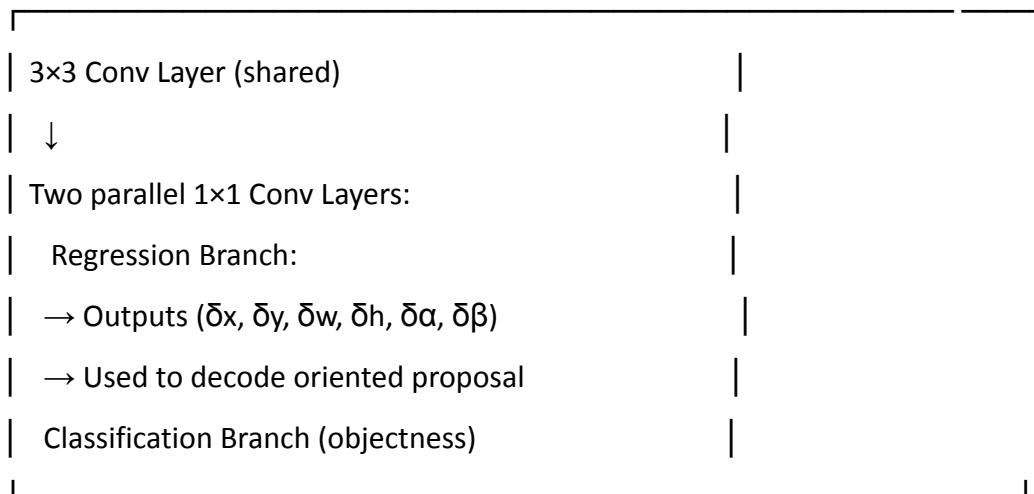
FPN (Feature Pyramid Network)



Multi-scale feature maps: {P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub>, P<sub>5</sub>, P<sub>6</sub>}



For each level (P<sub>i</sub>):



Generate oriented proposals:

$(x, y, w, h, \Delta\alpha, \Delta\beta)$  for each anchor

↓

Final Output: Sparse set of oriented region proposals

ب. نحوه فرمولبندی تابع هزینه (در RPN Oriented function loss) را توضیح دهید. هر یک از اجزای این تابع را بهطور واضح تعریف و هدف آنها را شرح دهید.

تابع هزینه بهصورت ترکیبی از دو بخش اصلی تعریف می‌شود:

**Classification Loss .1**

**Regression Loss .2**

فرمول کلی تابع هزینه به شکل زیر است:

$$L_1 = \frac{1}{N} \sum_{i=1}^N F_{cls}(p_i, p_i^*) + \frac{1}{N} p_i^* \sum_{i=1}^N F_{reg}(\delta_i, t_i^*)$$

$N$ : تعداد کل anchor ها در یک mini-batch که بهصورت پیشفرض، مقدار آن ۲۵۶ در نظر گرفته شده است.

$p_i$ : احتمال پیش‌بینی شده توسط شبکه برای اینکه anchor مربوط به شیء واقعی باشد.

$p_i^*$ : برچسب واقعی (Ground Truth) برای anchor شماره‌ی  $i$ . مقدار ۱: anchor  $i$  مثبت و مقدار ۰: anchor  $i$  منفی

ج: تابع هزینه طبقه‌بندی برای anchor شماره‌ی  $i$ . از Cross Entropy Loss استفاده می‌شود. هدف:  $F_{cls}(p_i, p_i^*)$  یادگیری درست اینکه anchor مربوط به شیء هست یا پس زمینه.

د: بردار ۶بعدی از پارامترهای پیش‌بینی شده توسط مدل برای anchor شماره‌ی  $i$ . این‌ها مقادیر نرمال‌شده‌ای هستند که موقعیت و شکل جعبه‌ی مایل را نسبت به anchor توصیف می‌کنند.

ه: بردار ۶بعدی مقدار هدف برای anchor شماره‌ی  $i$ . شامل پارامترهایی مشابه  $\delta$  است ولی بر اساس جعبه‌ی واقعی تعریف می‌شود.

$$\mathbf{t}_i^* = (t_x^*, t_y^*, t_w^*, t_h^*, t_\alpha^*, t_\beta^*)$$

$F_{reg}(\delta_i, t_i^*)$ : تابع هزینه‌ی مکانیابی که اختلاف بین بردار پیش‌بینی‌شده و مقدار واقعی را اندازه‌می‌گیردو فقط زمانی محاسبه می‌شود که anchor مثبت باشد.

هدف کلی این تابع بهبود تشخیص دقیق موقعیت جعبه‌های مایل (oriented bounding boxes) از طریق یادگیری دقیق مقادیر offset نسبت به anchor و یادگیری درست اینکه کدام anchorها واقعاً شیء هستند و کدام‌ها نیستند (positive) از طریق طبقه‌بندی دقیق است.

### :Rotated RoI Align

الف. هدف از **RoIAlign Rotated** چیست؟ گامبه‌گام توضیح دهد که چگونه این عملیات انجام می‌شود و یک مثال روشن ارائه کنید.

هدف اصلی Oriented RoIAlign، استخراج ویژگی‌های دقیق و rotation-invariant از نواحی مایل (Oriented) از نواحی مایل (Proposals) در تصویر است.

بر خلاف RoIAlign عادی که فقط برای جعبه‌های محور-محور (axis-aligned) کارایی دارد، Rotated RoIAlign می‌تواند با جعبه‌هایی که زاویه دارند (مثل اجسام چرخیده شده یا کج شده) کار کند. این قابلیت، دقت مدل را در تشخیص اشیاء مایل افزایش می‌دهد.

### نحوه انجام عملیات

#### گام اول: دریافت ورودی (جعبه‌ی مایل)

ورودی، یک جعبه مایل (Oriented Proposal) است که معمولاً یک متوازی‌الأضلاع است. این جعبه با چهار رأس مشخص می‌شود:

$$(v_1, v_2, v_3, v) = v$$

#### گام دوم: تبدیل به مستطیل چرخیده شده

برای آسان‌سازی پردازش، جعبه را به یک مستطیل oriented تبدیل می‌کنیم. قطر کوتاه را کش می‌دهیم تا طول آن برابر با قطر بلند شود.

پارامترهای مستطیل به صورت زیر تعریف می‌شوند:

$$(x, y, w, h, \theta)$$

x,y: مرکز مستطیل

w,h: عرض و ارتفاع

$\theta$ : زاویه چرخش نسبت به محور افقی

### گام سوم: Feature Map

چون Feature Map نسبت به تصویر ورودی مقیاس کوچکتری دارد، مختصات و ابعاد مستطیل را با نسبت  $s$  مقیاس می‌کنیم:

$$\begin{cases} w_r = w/s, & h_r = h/s \\ x_r = \lfloor x/s \rfloor, & y_r = \lfloor y/s \rfloor \end{cases}$$

### گام چهارم: تقسیم به شبکه‌ی $m \times m$

جعبه را به  $m \times m$  سلوول (مثلًا  $7 \times 7$ ) تقسیم می‌کنیم.  
هدف: استخراج ویژگی‌های با ابعاد ثابت برای همهٔ نواحی، بدون توجه به ابعاد یا زاویه.

### گام پنجم: نمونه‌برداری و محاسبه میانگین

در هر سلوول، با اعمال تبدیل چرخشی  $R(x, y, \theta)$  مختصات نقاط را به درستی روی نقشه ویژگی می‌اندازیم.

● برای هر سلوول ( $i, j$ ) و کanal  $c$ :

$$\mathbf{F}'_c(i, j) = \sum_{(x, y) \in \text{area}(i, j)} \mathbf{F}_c(R(x, y, \theta)) / n$$

مثال:

فرض کنید در تصویر یک ماشین پارک شده با زاویه  $30^\circ$  درجه وجود دارد.

در RoIAlign عادی یک جعبه محور-محور (مثلًا مستطیل صاف) روی ماشین می‌کشند اما چون ماشین زاویه دارد، بخش زیادی از آن بیرون از جعبه می‌ماند و ویژگی‌ها ناقص و نادرست می‌شوند.

در Rotated RoIAlign جعبه‌ای هم‌استتا با زاویه ماشین رسم می‌شود (مثلًا  $30^\circ = \theta$ ). این جعبه دقیقاً ماشین را می‌پوشاند و ویژگی‌ها به‌طور دقیق، چرخش‌ناپذیر و کامل استخراج می‌شوند.

ب. مشکالت احتمالی در صورت عدم استفاده از RoIAlign Rotated را توضیح دهد. برای استدلال خود مثالها یا توجیه‌های نظری ارائه کنید.

در صورت عدم استفاده از Rotated RoIAlign و بهکارگیری روش‌های معمول مانند RoIAlign عادی که فقط از جعبه‌های محور-محور (axis-aligned) استفاده می‌کنند، مشکلات متعددی در استخراج ویژگی‌ها برای اشیاء با زاویه ایجاد می‌شود. در این حالت، اگر شیء مورد نظر در تصویر دارای چرخش باشد (مثلًاً یک پنجره‌ای که کج فرار گرفته)، جعبه پیشنهادی نمی‌تواند بهدرستی آن را پوشش دهد و بخش زیادی از اطلاعات مفید یا درون جعبه قرار نمی‌گیرد یا نویز‌های اطراف وارد جعبه می‌شوند. این موضوع باعث می‌شود ویژگی‌های استخراج شده ناتمام یا غیر دقیق باشند و مدل در تشخیص و طبقبندی اشیاء دچار کاهش عملکرد شود. از نظر نظری، این مسئله باعث نقض spatial alignment (rotation-invariant) می‌گردد که برای کاربردهایی مانند تشخیص در تصاویر هوایی یا صنعتی چرخش‌ناپذیر (rotation-invariant) هستند. برای مثال، اگر یک پنجره کج در تصویر باشد ولی با RoIAlign عادی یک جعبه صاف روی آن کشیده شود، تنها بخشی از پنجره وارد ناحیه می‌شود یا حتی با بخشی از دیوار قاطی می‌شود، و این باعث استخراج ویژگی اشتباه و کاهش دقت نهایی می‌گردد.

#### عملکرد و کارایی:

الف. توضیح دهد که CNN-R Oriented چگونه به دقت و کارایی بالا دست پیدا می‌کند. به طور مشخص به آزمایشها و نتایج ارائه شده در مقاله ارجاع دهد.

مدل Oriented R-CNN با ترکیب یک ساختار دو مرحله‌ای و طراحی مناسب برای تشخیص اشیای چرخیده، به دقت و کارایی بالا دست یافته است. در مرحله اول، بخش Oriented RPN با انتخاب ۱۰۰۰ پیشنهاد از هر بخش تصویر و استفاده از معیار IoU برابر با ۰.۵، موفق به دستیابی به دقت recall تا ۹۲.۸٪ شده است که نشان‌دهنده توانایی بالا در تشخیص اشیای با زوایا و نسبت‌های مختلف است. در ادامه، این مدل در مقایسه با ۱۹ روش دیگر روی مجموعه DOTA و ۱۰ روش دیگر روی HRSC2016، توانسته است با استفاده از backbone backbones R-50-FPN و R-101-FPN به ترتیب دقت mAP برابر با ۷۵.۸۷٪ و ۷۶.۲۸٪ را کسب کند و حتی با R-50-FPN به دقت ۸۰.۸۷٪ با استفاده از آموزش و تست چندمقیاسی رسیده است. همچنین، این مدل با سرعت ۱۵.۱ فریم بر ثانیه روی GPU، کارایی‌ای نزدیک به مدل‌های تکمرحله‌ای دارد ولی با دقتی بسیار بالاتر، که این توازن بین دقت و سرعت یکی از مزیت‌های اصلی آن محسوب می‌شود.

ب. عوامل مؤثر در کارایی محاسباتی چارچوب CNN-R Oriented را توضیح دهد و نقش هر یک از این عوامل را مشخص کنید.

کارایی محاسباتی بالای Oriented R-CNN نتیجه ترکیب چندین طراحی در معماری و فرایند پردازش آن است. اول از همه، در مرحله تولید پیشنهادها (proposal generation)، این مدل فقط ۲۰۰۰ پیشنهاد را در هر سطح از FPN نگه می‌دارد و با استفاده از NMS افقی با آستانه IoU برابر ۰.۸، بسیاری از پیشنهادهای اضافی و تکراری را حذف می‌کند. این کار باعث کاهش حجم داده‌های ورودی به مراحل بعدی و در نتیجه صرفهجویی در زمان پردازش می‌شود.

در مرحله دوم، فقط ۱۰۰۰ پیشنهاد برتر (بر اساس امتیاز طبقبندی) انتخاب می‌شوند و این تعداد به عنوان ورودی به head مدل داده می‌شود، که باعث ایجاد توازن بین سرعت و دقت می‌شود. در نهایت، مدل با استفاده از NMS با آستانه ۰.۱، روی خروجی‌های نهایی اعمال فیلتر می‌کند تا نتایج نهایی بدون همپوشانی زیاد ارائه شوند.

همچنین استفاده از backbone سبکتر و بیینه مانند ResNet-50-FPN، که هم دقت بالایی دارد و هم از نظر محاسباتی مفرون به صرفه است، یکی از عوامل مهم در کاهش زمان اجرای مدل است. نتایج آزمایش‌ها نشان می‌دهند که این چارچوب با وجود دقت بالا (ta mAP %80.87)، با سرعت تقریباً 15.1 فریم بر ثانیه در یک کارت گرافیک RTX 2080Ti اجرا می‌شود، که نزدیک به سرعت روش‌های تکمرحله‌ای است، اما با دقتی به مراتب بیشتر. همه این عوامل در کنار هم باعث شده‌اند که Oriented R-CNN در عین حفظ دقت بالا، کارایی محاسباتی خوبی نیز داشته باشد.

ج. یک تحلیل انتقادی کوتاه از مقایسه CNN-R Oriented با سایر آشکارسازهای جهتدار دو مرحله‌ای ذکر شده در مقاله ارائه کنید و نقاط قوت و ضعف کلیدی را برجسته کنید.

در مقایسه با سایر آشکارسازهای جهتدار دو مرحله‌ای، Oriented R-CNN عملکرد چشمگیری دارد و در اکثر موارد نتایج بهتری ارائه می‌دهد. طبق نتایج ارائه شده در مقاله، این مدل با استفاده از backbone backbone DOTA و ResNet-101-FPN و ResNet-50-FPN کسب کند که از سایر روش‌های مقایسه شده برتر است. حتی با backbone ResNet-101-FPN (R-50-FPN)، عملکرد آن از روش‌هایی با backbone سنگین‌تر (مثل R-101-FPN) نیز بهتر بوده که نشان‌دهنده کارایی بالا و طراحی بیینه مدل است. همچنین، با بهکارگیری آموزش و تست چند مقیاسی، دقت مدل به mAP %80.87 نیز می‌رسد که بسیار رقابتی است.

از نظر نقاط قوت، این مدل تعادل مناسبی بین دقت و سرعت برقرار کرده است؛ بهطوری که با سرعت تقریبی 15.1 فریم بر ثانیه، نزدیک به روش‌های تکمرحله‌ای عمل می‌کند اما با دقت بسیار بالاتر. همچنین، استفاده از NMS چند مرحله‌ای (افقی و جهتدار)، انتخاب دقیق تعداد پیشنهادها، و طراحی کارآمد RPN باعث کاهش سربار محاسباتی شده‌اند.

در مقابل، از نقاط ضعف احتمالی می‌توان به پیچیدگی پیاده‌سازی و نیاز به تنظیمات خاص مانند thresholds برای NMS و اندازه‌های برش اشاره کرد که ممکن است در کاربردهای real time یا در محیط‌های منابع محدود چالش برانگیز باشند. همچنین، عملکرد مدل در صورت کاهش تعداد پیشنهادها (مثلاً از 1000 به 300) کاهش چشمگیری دارد که نشان می‌دهد مدل وابستگی بالایی به تعداد کافی proposal دارد.

## بخش دوم: پیاده سازی عملی

راه اندازی محیط و آماده سازی مجموعه داده:

مجموعه داده HRSC 2016 را با استفاده از کد ارائه شده دانلود و پیش پردازش کنید. یک کلاس سفارشی Dataset PyTorch midpoint offset تبدیل می‌کند. قطعاتی از کد و حداقل سه مثال از بارگذاری موفق داده ها ارائه دهید.

دانلود مجموعه دیتا با استفاده از کد ارائه شده:

کلاس ایجاد شده:

```
class HRSC2016Dataset(Dataset):
    def __init__(self, dataset_path):
        [7]   import kagglehub
        dataset_path = kagglehub.dataset_download('weiming97/hrsc2016-ms-dataset')

        self.image_files = [f for f in os.listdir(self.image_dir)]
        self.annotation_files = [f for f in os.listdir(self.annotation_dir)]
```

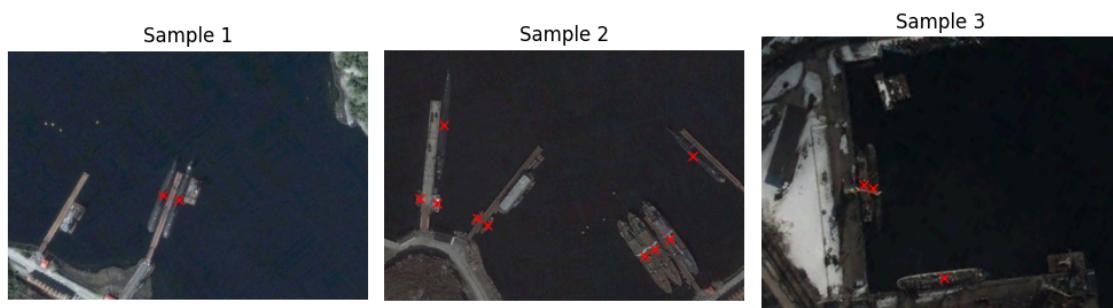
در این کلاس، دیتاست از مسیر گفته شده لود میشود و سپس تمامی عکس ها و annotation های موجود در دیتاست در فایل های annotation\_files و image\_files لود قرار میگیرند.

```
def __getitem__(self, idx):
    image_file = self.image_files[idx]
    image_path = os.path.join(self.image_dir, image_file)
    image = Image.open(image_path).convert("RGB")

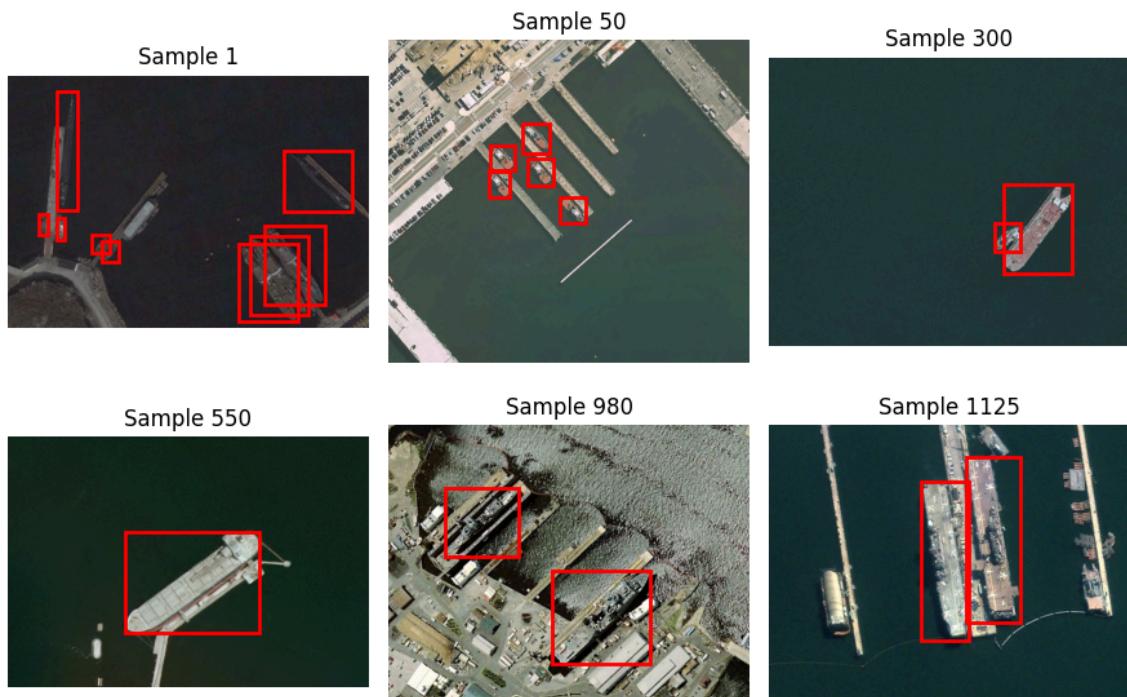
    annotation_file = image_file.replace('.bmp', '.xml')
    annotation_path = os.path.join(self.annotation_dir, annotation_file)
    boxes = self.parse_annotation(annotation_path)
    boxes = self.convert_to_midpoint_offset(boxes)

    return image, boxes
```

این تابع وظیفه لود کردن عکس ها و باکس های متناظر با آن عکس ها بر عهده دارد. برای این کار فایل عکس با اندیس مورد نظر را باز میکند، همچنین فایل annotation مرتبط با آن عکس را نیز باز کرده و آن هارا با استفاده از تابع دیگر تعریف شده و انجام چند عملیات ساده به midpoint-offset تبدیل میکند و سپس عکس و midpoint-offset ها را برمیگرداند.



6 نمونه از عکس های بارگذاری شده. در این عکس ها محل باکس ها با  $\times$  قرمز علامت گذاری شده اند تا به درستی محاسبه شدن midpoint-offset ها مشخص باشد.



در تصاویر بالا هم باکس های اصلی برای چند نمونه دیگر نمایش داده شده اند.

## آموزش مدل :CNN-R Oriented

الف. مدل CNN-R Oriented را (بدون استفاده از ResNet FPN-50) مطابق مقاله به مدت ۳۶ دوره epoch (آموزش دهد). تنظیمات آموزش خود شامل برنامه نرخ یادگیری، تنظیمات بهینهساز، اندازه دستهها و هرگونه افزایش دادهها augmentation (را بهطور واضح مستند کنید).

:augmentation

```
# ===== Augmentation =====
def get_transform(train=True):
    if train:
        return A.Compose([
            A.Resize(256, 256),
            A.HorizontalFlip(p=0.5),
            A.VerticalFlip(p=0.2),
            A.RandomBrightnessContrast(p=0.2),
            A.Rotate(limit=10, p=0.3, border_mode=0),
            ToTensorV2()
        ])
    else:
        return A.Compose([
            A.Resize(256, 256),
            ToTensorV2()
        ])
```

در این بخش از کد، برای افزایش میزان داده ها augmentation را به صورت تغییر رنگ ها، چرخش و flip کردن انجام داده ایم.

تنظیمات مدل بر اساس مقاله به این صورت انجام شده است:

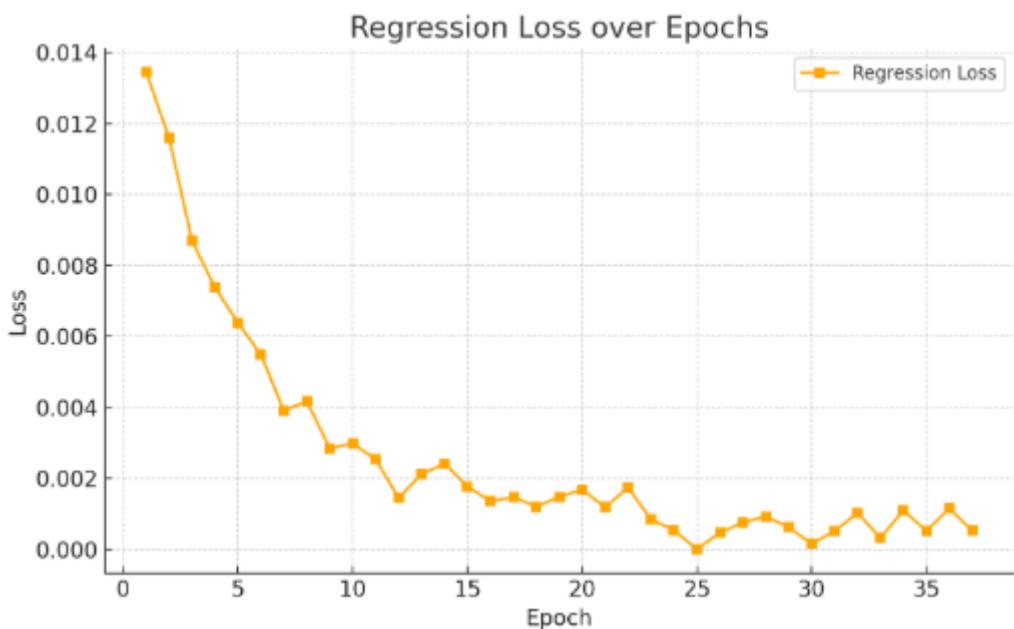
تنظیمات	بخش مدل
ResNet-50 pretrained	Backbone
Conv2d(2048 -> 256, kernel_size = 1)	FPN Layer
Conv2d from 256 to 256 with kernel size 3	classification head

and padding 1, followed by ReLU  Conv2d from 256 to 2 with kernel size 1	
Conv2d(256 → 256, kernel_size=3, padding=1) + ReLU  (Conv2d(256 → 40, kernel_size=1	regression head
حداکثر 10	تعداد باکس ها
2 تا (کشتی و بک گراند)	تعداد کلاس ها
CrossEntropyLoss	Loss Function for classification
SmoothL1Loss	Loss Function for regression
Adam and SGD	Optimizer
Scheduler Type: StepLR  Step Size: 5 epochs  Gamma (Decay Rate): 0.1	Scheduler
36	تعداد epoch ها
cuda	device

ب. نمودارهای مربوط به توابع هزینه در فرآیند آموزش را رسم و تحلیل کنید. تفسیر این نمودارها را ارائه دهید و الگوهای همگرایی، عالمند بیشبرازش و پایداری کلی آموزش را بررسی کنید.

Training Loss over Epochs





با توجه به نمودارهای نمایش داده شده برای Training Loss و Regression Loss در طول 36 دوره آموزش، می‌توان نتیجه گرفت که مدل به خوبی در حال یادگیری ویژگی‌های داده‌هاست. کاهش سریع اولیه در هر دو نمودار نشان‌دهنده‌ی آن است که مدل در ابتدای آموزش به سرعت در حال تنظیم وزن‌ها و کاهش خطأ بوده است و به علت انتخاب نرخ یادگیری با مقدار بالا این اتفاقه افتاده است. با کاهش نرخ یادگیری در 36 دوره آموزش میزان loss بیشتر ولی هموار تر خواهد بود. سپس کاهش تدریجی و هموار این ضررها نشان می‌دهد که فرآیند آموزش به سمت همگرایی پیش می‌رود. ثبات و کمنوسانی مقادیر Loss در انتهای نمودار نیز گویای آن است که مدل از overfitting رنج نمی‌برد و احتمالاً توانایی تعمیم نسبتاً خوبی دارد. در مجموع، الگوی نزولی و پایدار هر دو ضرر (کلاس‌بندی و رگرسیون) تأییدی بر اثربخشی معماری شبکه، انتخاب بهینه‌ساز، و تنظیمات یادگیری است.

## ارزیابی و تحلیل نتایج:

الف. مدل آموزش دیده خود را با نمایش پروپوزالها و جعبه های Truth Ground روی تصاویر مجموعه آزمون ارزیابی کنید. حداقل پنج تصویر با پیش‌بینی های مدل در مقایسه با Truth Ground ارائه دهید. تحلیل دقیق هر تصویر شامل دقت پیش‌بینی ها، خطاهای و دلایل احتمالی پیش‌بینیهای نادرست را شرح دهید.

در تصاویر زیر 6 مورد از خروجی های داده شده مدل رسم شده است. در این شکل ها باکس های قرمز نشانده‌ند بناکس های grand\_truth و باکس های آبی مربوط به proposal مدل اند. با توجه به این باکس ها در اکثر موارد مدل توانسته حدودی از باکس های اصلی را پیدا کند اما اینکه دقیقاً باکس را پیش‌بینی نکرده نشان میدهد که مدل چهار بیش برآراش نشده است. در مواردی نیز تعداد باکس ها به درستی پیش‌بینی نشده است. به نظر میرسد با تغییرات در لایه های مدل و تغییر پارامتر ها بتوان به نتیجه بهتری دست یافت اما با توجه به خواسته مسئله این مدل فقط بر روی تعداد کمی پارامتر تست شده است.



الف. عملکرد مدل خود را با معیارهای ارائه شده در مقاله اصلی مقایسه کنید. نتایج خود را بهطور واضح بیان کرده و تفاوت‌های احتمالی با مقاله را تحلیل کنید. به دلایل احتمالی این تفاوت‌ها اشاره و پیشنهادهای مشخص و عملی برای بهبود پیاده‌سازی فعلی خود ارائه دهید.

در پیاده‌سازی انجامشده، روند کاهش loss طی دوره‌های آموزش و نتیجه ارزیابی با معیار های دیگر نشان‌دهندهٔ یادگیری مؤثر مدل و همگرایی تدریجی آن است. بهویژه، کاهش یکنواخت و پیوستهٔ مقدار loss طبقه‌بندی و رگرسیون نشان‌دهندهٔ آن است که مدل توانسته به مرور ویژگی‌های کلیدی تصاویر ورودی را استخراج کرده و اطلاعات مکانی و معنایی کشنی‌ها را بدرستی یاد بگیرد. با این حال، نتایج نهایی می‌تواند تحت تأثیر چندین عامل به اندازهٔ نتایج مقاله خوب نباشد. برای مثال، استفاده از معماری ساده‌تر در مدل شبکه عصبی، کمبود تکنیک‌های افزایش داده برای بهبود تنوع مجموعه داده، و همچنین انتخاب تعداد محدود برای جعبه‌های پیش‌بینی شده دلیل کاهش دقت نهایی مدل نسبت به مدل‌های پیشرفته‌تر مقاله است. علاوه بر این، استفاده‌نکردن از anchorهای چرخشی با زوایای متفاوت - که برای تشخیص اجسامی با موقعیت‌های زاویه‌دار ضروری است - یکی دیگر از محدودیت‌های پیاده‌سازی فعلی محسوب می‌شود. در نهایت، با گسترش مجموعه داده، افزایش دوره‌های آموزش، افزودن لایه‌های پیچیده‌تر در معماری شبکه، بهره‌گیری از anchorهای چرخشی و تکنیک‌های افزایش داده، دقت مدل را ارتقاء می‌دهد.