Farshad Borjalizade

Object detection with custom version of yolov2

University of Tehran

Dr.Sajedi

Mordad 1400

فهرست

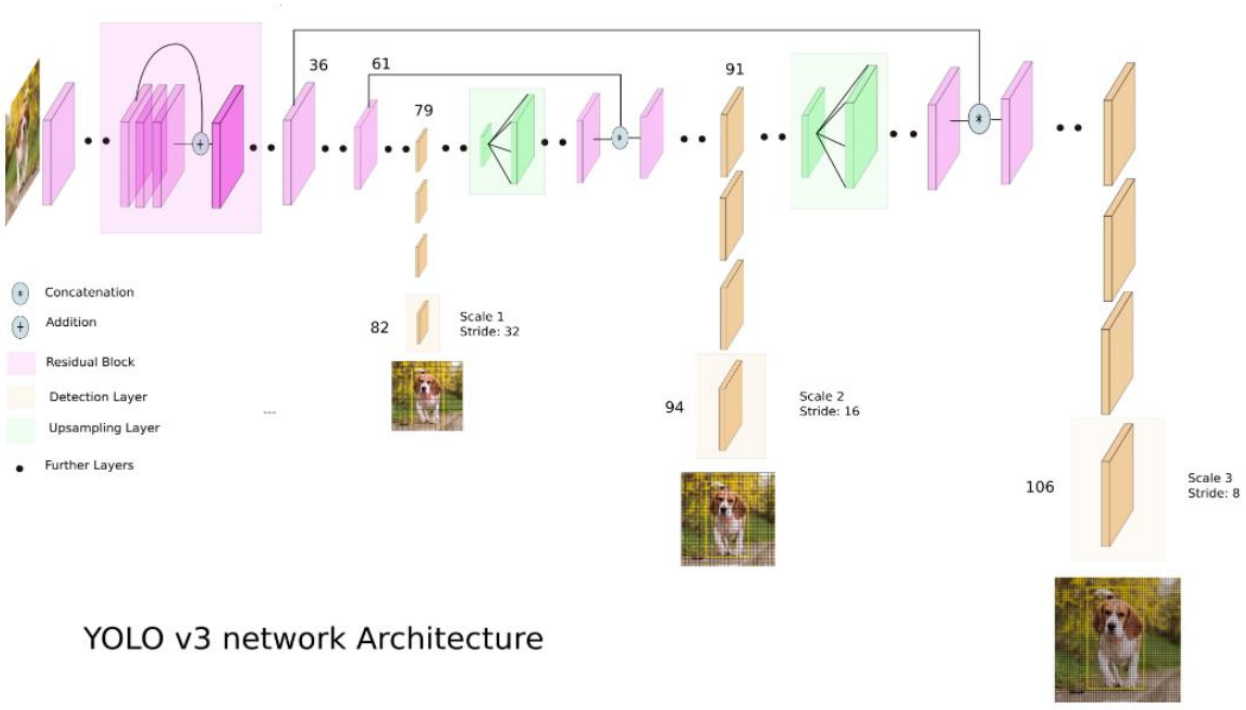
چکیده

در بینایی ماشین تشخیص اشیا (object detection) همواره یکی از چالش برانگیزترین و محبوب ترین موضوعات به دلیل کاربردهای فراوانی که دارد محسوب می شود. روش های متعددی در این زمینه بیان شده است که یکی از بهترین های آن ها استفاده از شبکه های عصبی پیچیده عمیق (DCNNs) است این روش بهترین نتایج را در مقایسه با روش های دیگر گرفته است، این روش برای بهینه سازی پارامترهای مدل و یادگیری از روش های مبتی بر مشتق (gradient based) استفاده می کند. در این پروژه قصد داریم که با الهام گرفتن از مدل های YOLOv2 و YOLOv3 مدلی را طراحی کنیم که میزان خطای آن کمتر از مدل های گفته شده باشد. شبکه خود را با مجموعه داده Pascal VOC2012 آموزش داده ایم که در مورد آن در بخش مربوطه توضیحات کامل بیان شده است.

مقدمه

در چند سال اخیر یادگیری عمیق (deep learning) به طور گسترده ای در زمینه های مختلف بینایی ماشین (computer vision) از جمله دسته بندی تصاویر (image classification) و تشخیص اشیا و تقسیم بندی تصویر (image segmentation) مورد استفاده قرار گرفته است، یکی از موضوعاتی که بسیار مورد توجه قرارگرفته، تشخیص اشیاست. چالش هایی که در این حوزه مطرح است را می توان به دو قسمت تقسیم کرد اول اینکه مکان شی مورد نظر را در تصویر شناسایی کنیم (location problem) و دوم اینکه که در مکان مشخص شده چه شی ای قرار دارد (category problem). در روش های سنتی تشخیص اشیا از غلتاندن پنجره (sliding windows) با اندازه های متفاوت و کاندید قرار دادن مناطق متفاوت تصویر این کار انجام می شد با ظهور شبکه های عمیق و خصوصا شبکه های پیچشی(convolutional neural network) ها و استفاده آن در مدل های مختلف R-CNN و Fast R-CNN و Faster R-CNN جهش قابل توجهی در این زمینه رخداد اما یک ایراد بزرگی که به این مدل ها می شد گرفت بار محاسباتی و زمان بسیار آن ها برای یادگیری شبکه بود که منشأ این ایراد را می توان به دو عامل تقسیم کرد اول اینکه این شبکه ها end-to-end نبودند به این معنا که شبکه بدین صورت نیست که تمام کار های تشخیص تصویر به صورت یکپارچه انجام شود و دوم اینکه مدل زمان بسیار زیادی را صرف یادگیری و کاندید قرار دادن نواحی مختلف تصویر برای اینکه مشخص کند آیا در منطقه مورد نظر شی ای وجود دارد یا خیر می کرد.

با معرفی مدل YOLO (you only look once) مشـــکل کاندید قرار دادن نواحی مختلف تصـویر را به عنوان یک مسـئله رگرسـیون در نظر می گیرد و نواحی محتمل وجود شـی را شناسایی می کند، این روش به طور قابل توجهی سرعت یادگیری مدل را افزایش می دهد اما نسـبت به مدل Faster R-CNN از میزان خطای بیشـتری برای تشـخیص مکان اشـیا برخوردار است، برای حل این مشکل نسخه های متفاوتی از YOLO منتشر شده است که با استفاده از تکنیک های متنوع سعی بر این داشتند که علاوه بر حفظ سرعت مدل ایراد دقت پایین مدل برای تشـخیص مکان را هم رفع کنند. YOLOv3 توانسـت با اسـتفاده از سـاختار شبکه Darknet-53 توازن خوبی بین سرعت و دقت مدل، کار تشخیص اشیا را انجام دهد و به اصطلاح جز مدل های برتر (state-of-the-art) این حوزه قرار گرفت.

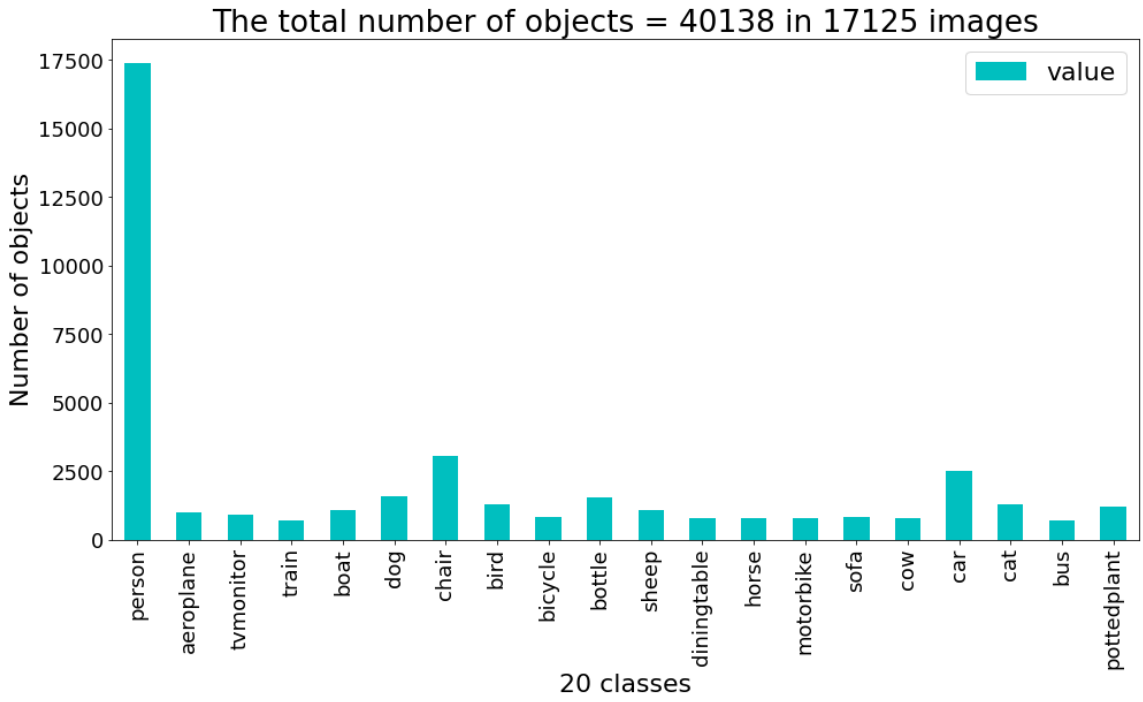


تصویر ۱– نمونه ای از ساختار YOLOv3

ایده مدلی که پیاده سازی کرده ایم ترکیبی از مدل های YOLOv3 و YOLOv2 است که با بسیاری از پارامترها و ساختار مدل های YOLO اشتراک دارد.

مجموعه داده

در این پروژه از مجموعه داده معروف Pascal VOC2012 استفاده می کنیم این مجموعه داده به طور کلی دارای ۲۰ کلاس از ۴ دسته متفاوت که انسان (Person) ۱ کلاس، حیوان (Animal) ۶ کلاس، وسایل نقلیه (Vehicle) ۷ کلاس و وسایل خانه (Indoor) ۶ کلاس می باشد که نتایج را بر روی آن بیان می کنیم در تصویر ۲ تعداد کلاس ها و اشیا موجود در هر کلاس به همراه تعداد تمامی اشیا موجود در مجموعه داده نشان داده شده است.



تصویر ۲– توزیع تعداد اشیا در کلاس ها

```
{'name': 'aeroplane', 'xmin': 86, 'ymin': 115, 'xmax': 312, 'ymax': 270},
{'name': 'aeroplane', 'xmin': 110, 'ymin': 130, 'xmax': 163, 'ymax': 182},
{'name': 'person', 'xmin': 162, 'ymin': 266, 'xmax': 177, 'ymax': 339}
{'name': 'person', 'xmin': 21, 'ymin': 279, 'xmax': 36, 'ymax': 352}
```

تصویر۳- نمونه ای از داده های برچسب گذاری شده

پیاده سازی مدل

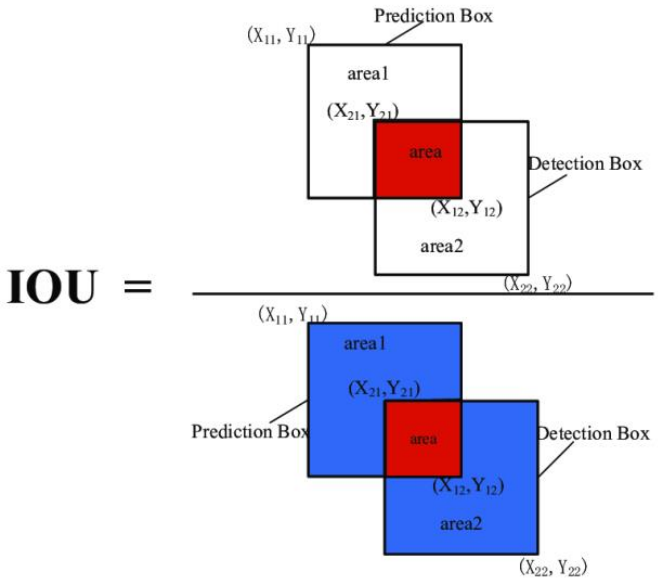
برای پیاده سازی مدل تشخیص شی ابتدا نیاز به یک سری از تعریف جزئیات و مفروضات داریم.

Intersection Over Union – IOU

در بسیاری از مدل های تشخیص اشیا IOU به عنوان معیاری برای ارزیابی در نظر گرفته می شود، در واقع معیاری است برای بررسی میزان احتمال وجود یک شی در یک مکان از تصویر.

فرض کنید که یک تصویر برچسب خورده داریم یعنی تمام اشیا موجود در تصویر را به همراه محل رخدا آن ها (bounding box coordinates) را داریم سپس مدل ما هم یک سری از مکان ها را به عنوان رخداد شی ای در نظر میگیرد معیار IOU کاری انجام می دهد به این صورت است که بررسی می کند چقدر

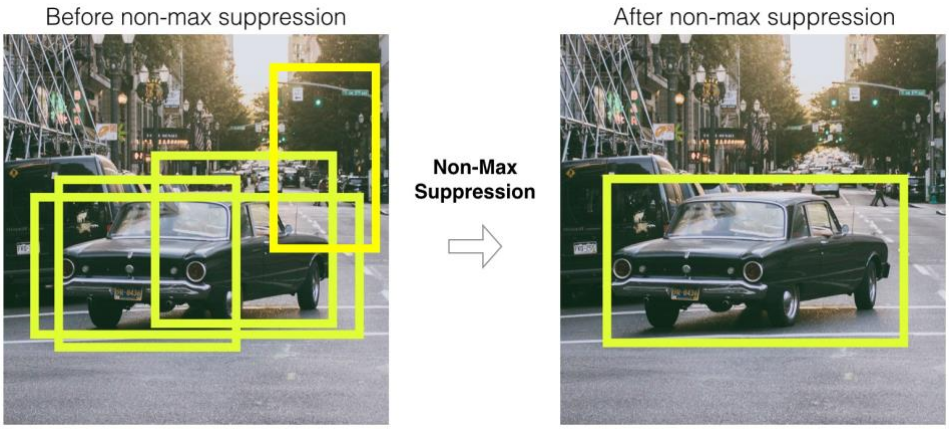
محـدوده ای کـه مـدل مـا انتخـاب کـرده اسـت بـا چیـزی کـه در واقعیـت وجـود دارد اشتراک و هم پوشانی دارد.



تصویر۴- نحوه عملکرد الگوریتم IOU

Non Maximum Suppression - NMS

در فراینـــد آمـــوزش مـــدل و در بخـــش تعیـــین محـــل رخـــداد شـــی در تصـــویر (bounding box coordinates) تعـدادی مکـان بـرای تشـخیص شـی پـیش بینـی کرده ایـم امـا نکتـه ای کـه وجـود دارد ایـن اسـت کـه ممکـن اسـت ایـن پنجـره هـا شـی های یکسانی را پـیش بینـی کـرده باشـند بـرای مثـال اگـر در تصـویر یـک ماشـین داشـته باشـیم قصـد داریـم کـه بهتـرین پنجـره ای کـه مکـان رخـداد شـی را پـیش بینـی کـرده اسـت را داشـته باشـیم و بقیـه پـیش بینـی هـا را حـذف کنـیم، در واقـع هـدف حـذف کردن پنجره هایی است که یک شی یکسان را نشان می دهند.

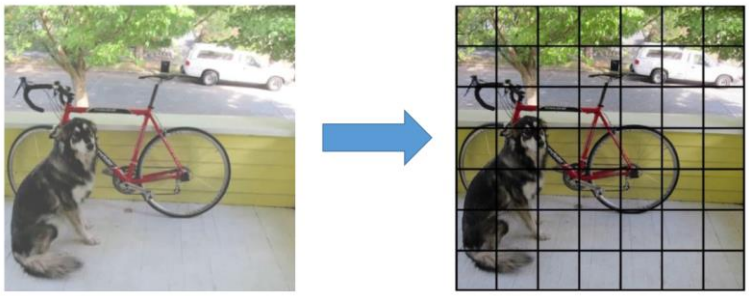


تصویر۵- مثالی از اعمال الگوریتم NMS

تعریف مدل

ساختار کلی مدل به چهار قسمت تقسیم می شود که عبارت اند از:

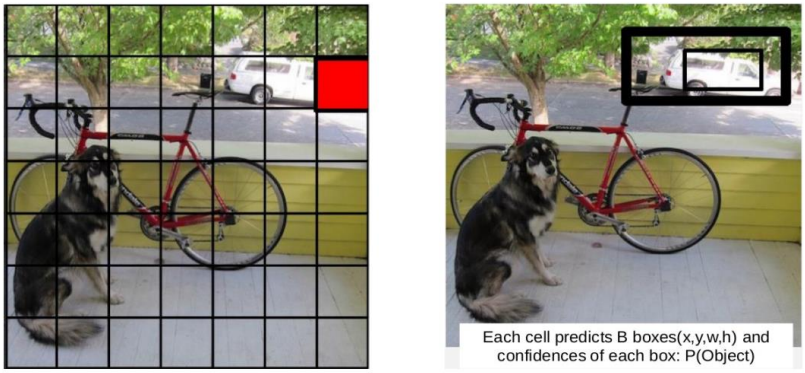
۱. **اندازه تصاویر ورودی : در مجموعه داده** Pascal VOC2012 **تصاویر با اندازه های متفاوتی وجود دارند اما از آنجایی که اندازه ورودی شبکه نمی تواند متغیر باشد پس اندازه تصاویر را به ۴۱۶ ×۴۱۶ تبدیل می کنیم و هرکدام از تصاویر را به گرید سل های S × S تقسیم بندی می کنیم و معمولا هم این اعداد فرد هستند تا بتوان مرکز آن را به طور دقیق تشخیص داد و هر کدام از این گریدها مسئول شناسایی یک شی هستند.**



تصویر۶- اعمال گرید بندی S × S تصویر

۲. **شبکه استخراج ویژگی : استخراج ویژگی را با استفاده از شبکه های عمیق پیچشی (deep convolutional neural networks)و الگوریتم Kmeans برای دسته بندی و انتخاب بهترین IOU و Batch Normalization و استفاده از تکنیک concatenate به لایه های مختلف انجام داده ایم.**

۳. **پیش بینی محدوده شی (Bounding box prediction) : هر گرید سلی که دارای B جعبه پیش بینی است و برای هر جعبه یک احتمالی داریم که نشان دهنده این است که چقدر محتمل است در جعبه پیش بینی شده یک شی وجود داشته باشد (confidence scores) علاوه بر این برای نشان دادن مختصات هر جعبه هم به چهار متغیر نیازمندیم که به اضافه احتمال وجود شی در جعبه (x , y , w , h , conf) سر جمع پنج متغیر و به علاوه تعداد کلاس هایی که در مجموعه داده داریم (بیست کلاس) و در مجموع یک بردار سه بعدی داریم (S × S × B × (5 + C)) و با استفاده از آن پیش بینی را انجام می دهیم.**



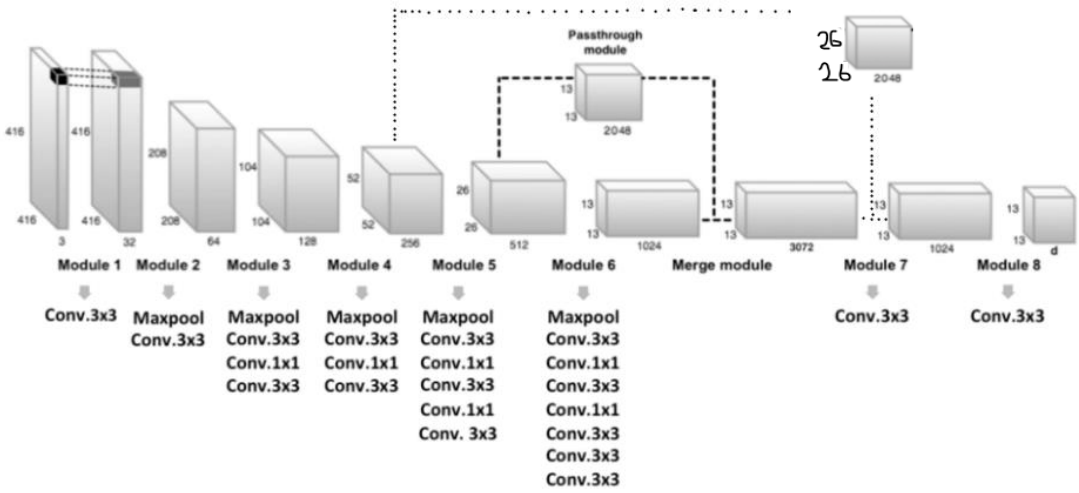
تصویر۷- در این تصویر برای هر سلول دو Bounding box در نظر گرفته شده است.

۴. **تشخیص نهایی : در نهایت می بایستی ما بهترین و محتمل ترین جعبه را برای تصویر و تشخیص شی انجام دهیم که این کار را با اعمال الگوریتم NMS انجام خواهیم داد.**

تصویر۸- نمونه ای تشخیص نهایی

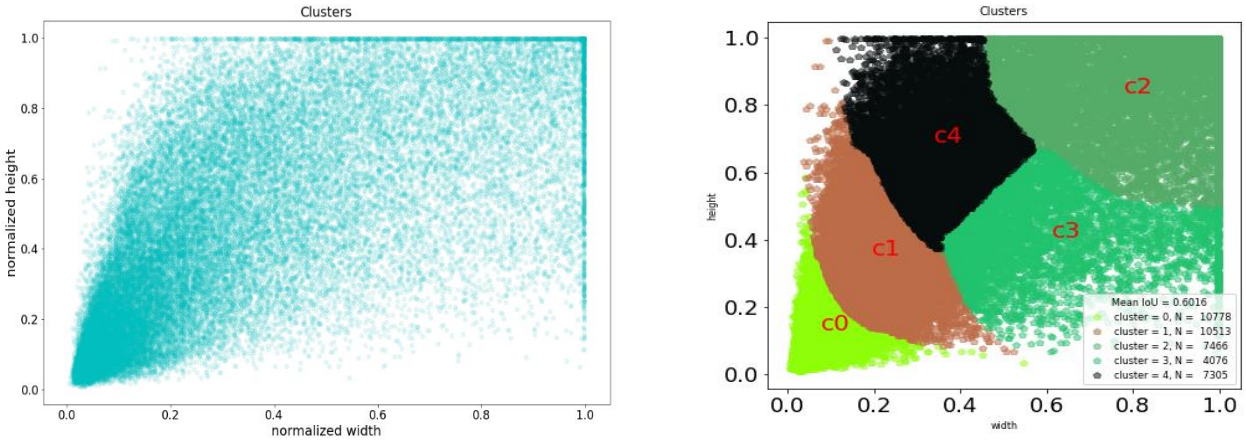
ساختار شبکه عمیق استخراج ویژگی ها

مدل طراحی شده در واقع ترکیبی از مدل های YOLOv3 و YOLOv2 است که با استفاده از تکنیک ترکیب کردن ویژگی های استخراج شده در لایه های پایین تر به بالاتر به تشخیص بهتر و دقیق تر اشیا کمک می کند.

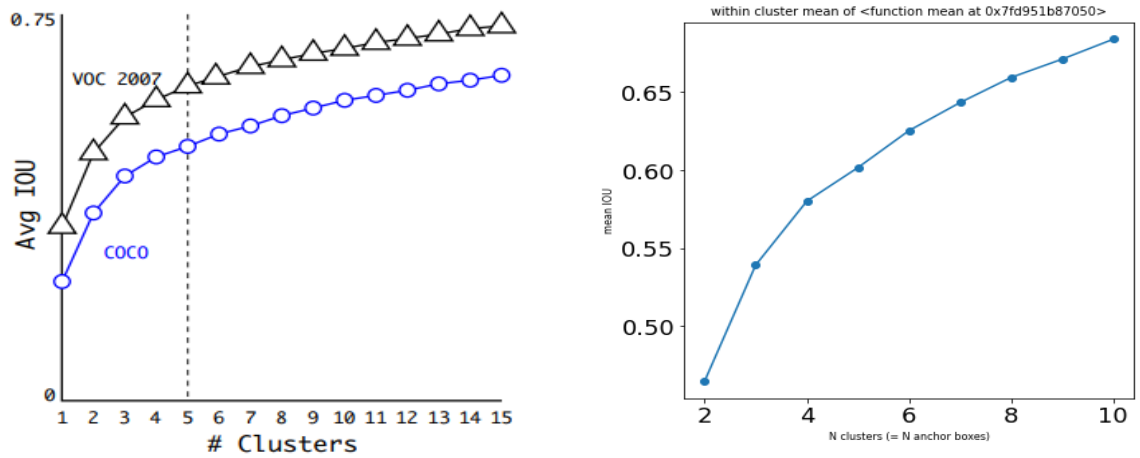


تصویر ۹- مدل پیاده سازی شده برای استخراج ویژگی ها

برای اینکه تشخیص اشیا دقیق تر صورت گیرد به جای اینکه به صورت تصادفی Bounding box تعریف کنیم Bounding box های از پیش تعریف شده ای به عنوان Anchor box را می توان در نظر گرفت و این کار را می توان به دو صورت انجام داد یکی اینکه به صورت دستی یک سری Anchor box بدست آورد و دوم اینکه به آن به عنوان یک مسئله خوشه بندی (Clustering) نگاه کرد زیرا هریک از اشیا دارای یک (w, h) است که میتوان آنها را با استفاده از الگوریتم kmeans خوشه بندی کرد در این مدل k=5 و معیار اندازه گیری را $d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid})$ در نظر گرفته ایم.



تصویر ۱۰– خوشه بندی پنج کلاسه بر اساس width و height تصاویر،



تصویر ۱۱– نتیجه میانگین IoU در تعداد خوشه بندی های متفاوت

پس پنج Anchor box که در واقع مرکز خوشه هاست را در نظر گرفتیم.

مدل دارای بیست و چهار لایه می باشد که به دلیل طولانی بودن از نمایش دادن آن در اینجا صرف نظر می کنیم (می توان آن را در قسمت کد مشاهده کرد)، تقریبا دارای پنجاه و سه میلیون پارامتر می باشد.

تابع خطا (Loss function)

همانند همه مدل های یادگیری می بایستی یک تابع خطایی را برای شبکه انتخاب کنیم تابع خطایی که در نظر گرفته ایم را می توان به سه بخش Localization و Confidence و Classification تقسیم کرد که در تصویر ۱۲ جزئیات بیشتری را می توان دید.

$$
\begin{aligned}
\text{loss}_{i,j} &= \text{loss}_{i,j}^{xywh} + \text{loss}_{i,j}^{p} + \text{loss}_{i,j}^{c} \\
\text{loss}_{i,j}^{xywh} &= \frac{\lambda_{\text{coord}}}{N_{L^{obj}}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} L_{i,j}^{\text{obj}} \Big[\left(x_{i,j} - \hat{x}_{i,j}\right)^2 + \left(y_{i,j} - \hat{y}_{i,j}\right)^2 + \\
&\qquad \left(\sqrt{w}_{i,j} - \sqrt{\hat{w}}_{i,j}\right)^2 + \left(\sqrt{h}_{i,j} - \sqrt{\hat{h}}_{i,j}\right)^2 \Big] \\
\text{loss}_{i,j}^{p} &= -\frac{\lambda_{\text{class}}}{N_{L^{obj}}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} L_{i,j}^{\text{obj}} \sum_{c \in \text{class}} p_{i,j}^{c} \log(\hat{p}_{i,j}^{c}) \\
\text{loss}_{i,j}^{c} &= \frac{\lambda_{\text{obj}}}{N^{conf}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} L_{i,j}^{\text{obj}} \left(IOU_{\text{prediuction}_{i,j}}^{\text{ground truth}_{i,j}} - \widehat{C}_{i,j} \right)^2 \\
&\qquad + \frac{\lambda_{\text{noobj}}}{N^{conf}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} L_{i,j}^{\text{noobj}} \left(0 - \widehat{C}_{i,j}\right)
\end{aligned}
$$

where:

- $N_{L^{obj}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} L_{i,j}^{\text{obj}}$
- $N^{conf} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} L_{i,j}^{\text{obj}} + L_{i,j}^{\text{noobj}}(1 - L_{i,j}^{\text{obj}})$
- $\text{prediuction}_{i,j} = (\hat{x}_{i,j}, \hat{y}_{i,j}, \hat{w}_{i,j}, \hat{h}_{i,j})$
- $\text{ground truth}_{i,j} = (x_{i,j}, y_{i,j}, w_{i,j}, h_{i,j})$
- λ_{coord}, λ_{class} and λ_{noobj} are scalars to weight each loss funciton

Here, $L_{i,j}^{\text{noobj}}$ and $L_{i,j}^{\text{obj}}$ are 0/1 indicator function such that:

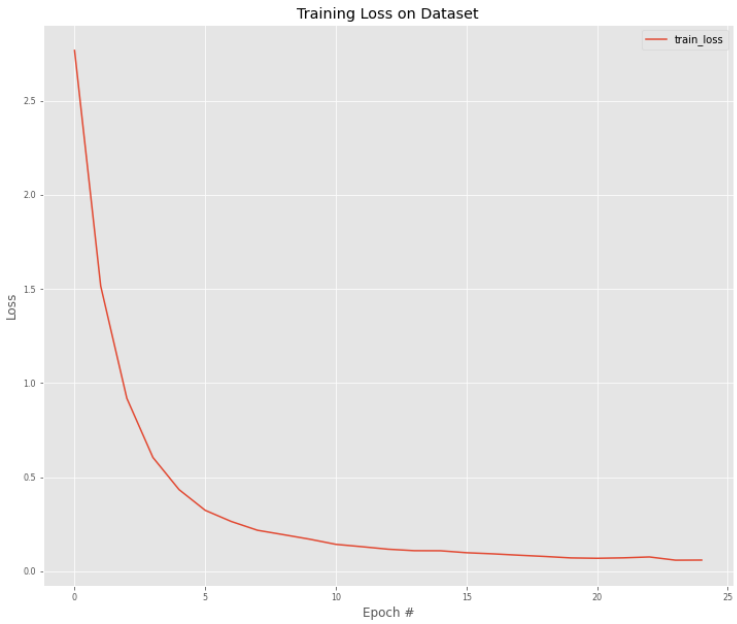
$$
L_{i,j}^{\text{obj}} = \begin{cases} 1 & \text{if } C_{i,j} = 1 \\ 0 & \text{else} \end{cases}
$$

$$
L_{i,j}^{\text{noobj}} = \begin{cases} 1 & \text{if } \max_{i',j'} IOU_{\text{prediuction}_{i,j}}^{\text{ground truth}_{i',j'}} < 0.6 \text{ and } C_{i,j} = 0 \\ 0 & \text{else} \end{cases}
$$

تصویر ۱۲- تابع خطا

آموزش مدل

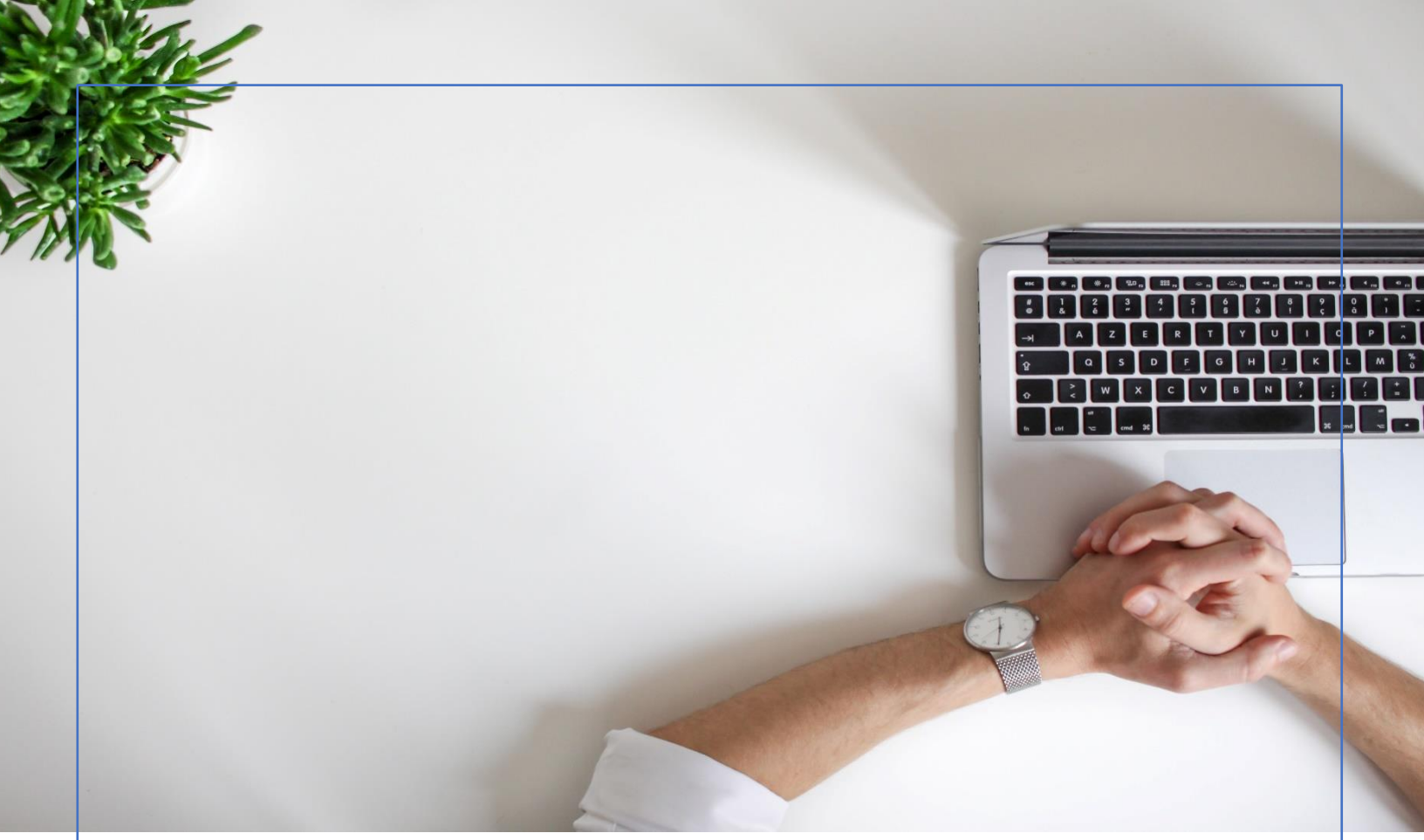
برای آموزش مدل از وزن های pretrain شـــده yolov2 اســـتفاده می کنیم و شـــبکه را با Learning rate=0.5e-4 و Epochs=25 و Batch size=16 به همراه پنجاه و سه میلیون پارامتر قابل یادگیری آموزش می دهیم زمان تقریبی برای آموزش مدل پنج ســـاعت به طول انجامید و مقدار خطای مدل تا حد قابل قبولی کاهش پیدا کرد و به ۰.۰۵ رســـیدیم که برای شبکه با این تعداد پارامتر بسیار مناسب است.



تصویر۱۳- نحوه کم شدن تابع خطا

تفاوت ها و شباهت ها با مقاله اصلی

مقاله اصلی	کار انجام شده
ورودی شبکه ۴۱۶*۴۱۶	ورودی شبکه ۴۱۶*۴۱۶
استفاده از RKCELM و AE برای استخراج ویژگی ها	استفاده از DCNN و Concatenate برای استخراج ویژگی ها
استفاده از K-means++ و K=5 برای خوشه بندی Anchor box	استفاده از K-means و K=5 برای خوشه بندی Anchor box
استفاده از معیار IOU برای انتخاب فاصله خوشه ها	استفاده از معیار IOU برای انتخاب فاصله خوشه ها
استفاده از تابع خطایی (Loss function) منحصر به فرد	استفاده از تابع خطایی (Loss function) متفاوت
استفاده از Bounding box prediction متفاوت	استفاده از Bounding box prediction مدل yolov2
سرعت بیشتر در آموزش شبکه	سرعت کمتر در آموزش شبکه
تشخیص بهتر اشیایی که هم پوشانی دارند	ضعف در تشخیص اشیا دارای هم پوشانی
مقدار خطای بیشتری دارد	بهبود ۰.۰۱ مقدار خطای مدل
تعداد پارامترهای کمتری دارد	تعداد پارامترهای بیشتری دارد



منابع

1. https://www.sciencedirect.com/science/article/abs/pii/S1051200420301019
2. https://arxiv.org/abs/1506.02640
3. https://ieeexplore.ieee.org/document/8100173
4. https://arxiv.org/abs/1804.02767
5. https://machinelearningmastery.com/how-to-perform-object-detection-with-yolov3-in-keras/
6. https://pjreddie.com/darknet/yolo/
7. https://arxiv.org/abs/1808.02350v1
8. https://pylessons.com/YOLOv3-explained/
9. http://host.robots.ox.ac.uk/pascal/VOC/voc2012/