https://medium.com/@vivek.yadav/part-1-generating-anchor-boxes-for-yolo-like-network-for-vehicle-detection-using-kitti-dataset-b2fe033e5807

Vivek Yadav

(Part 1) Generating Anchor boxes for Yolo-like network for vehicle detection using KITTI dataset.

Vivek Yadav
Follow

Jul 10, 2017 · 7 min read

Vivek Yadav, PhD

In this post, I will present steps for computing anchor boxes for YOLO9000 (or YOLOv2). YOLOv2 is a combined classification-bounding box prediction framework where we directly predict the objects in each cell and the corrections on anchor boxes. More specifically, YOLOv2 divides the entire image into 13X13 grid cells, next places 5 anchor boxes at each location and finally predicts corrections on these anchor boxes. YOLOv2 makes 5 predictions corresponding to corrections on location of center (x and y), height and width, and finally the intersection over union (IOU) between predicted bounding boxes and ground truth boxes. A unique feature of YOLOv2 is that all the predictions are have magnitude less than 1, as a result the chance of one type of cost dominating the optimization is less likely. A unique feature of YOLOv2 is that the anchor boxes are designed specifically for the given dataset using K-means clustering. Unlike other anchor boxes (or prior) based methods, like Single Shot Detection, YOLOv2 does not assume the aspect ratios or shapes of the boxes. As a result, the

YOLOv2 in general has lower localization loss and has higher intersection over union (IOU) between the target and network prediction. The rest of the post is organized as follows,

1. Data preparation
2. Exploratory data analysis
3. Generating anchor boxes using K-means clustering
4. Assigning anchor boxes to ground truth targets
5. **Data preparation**

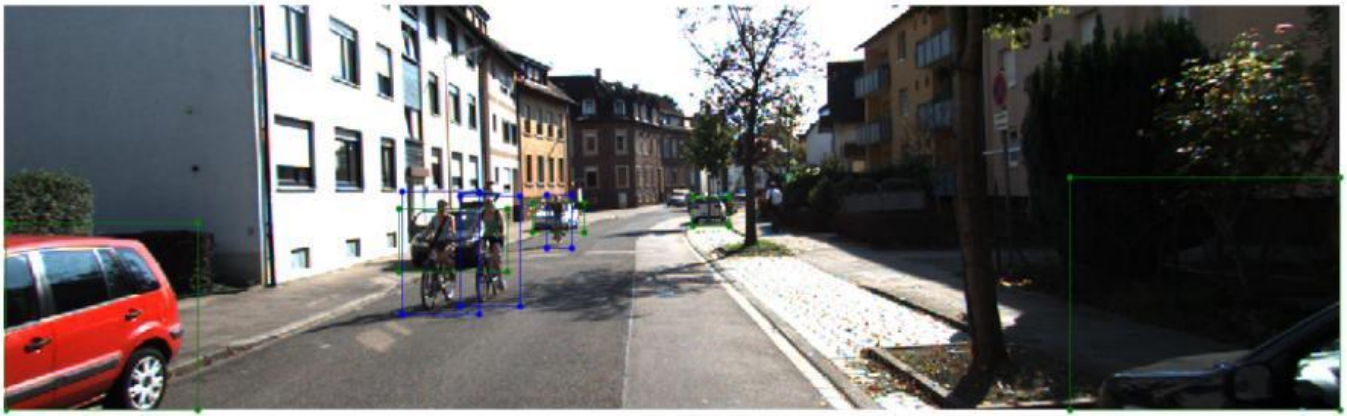
I first downloaded images and label from the kitti object-detection data set. I downloaded detection labels and images for the car's left-camera. After downloading, I put the images and labels in separate folders titled `kitti_l`and `kitti_labels` respectively. I then combined cars, trucks and van into one group called vehicles, and cyclists and pedestrian into a group called Person. In addition, I computed the center location, width and height of each bounding box, and normalized it by image dimensions. This allowed to resize image as needed and compute the same bounding boxes without any additional calculations. This also makes the process of data augmentation easy, that I will discuss in a more appropriate section later.

2. Exploratory data analysis (EDA)

EDA is perhaps the most important step of building any machine learning algorithm. Here, I will try and explain what YOLOv2 tries to do. Figure below presents a representative image from kitti dataset, in kitti-dataset each image is of size 1242 X 375, and there are about 7400 images with approximately 25000 annotations.

Please note, the labels are normalized by height and width of the image.

	0	label	xmin	xmax	ymin	ymax	x_c	y_c	h	w
0	Car	Vehicle	0.000000	0.144569	0.536203	1.000000	0.072284	0.768102	0.463797	0.144569
1	Car	Vehicle	0.294424	0.375907	0.500027	0.655829	0.335165	0.577928	0.155802	0.081483
2	Car	Vehicle	0.394778	0.433787	0.487112	0.560963	0.414283	0.524037	0.073850	0.039009
3	Car	Vehicle	0.514843	0.541845	0.469947	0.543529	0.528344	0.506738	0.073583	0.027002
4	Car	Vehicle	0.796575	1.000000	0.425053	1.000000	0.898288	0.712527	0.574947	0.203425
5	Cyclist	Person	0.296632	0.354714	0.457914	0.760775	0.325673	0.609345	0.302861	0.058082
6	Cyclist	Person	0.340588	0.385181	0.469278	0.743155	0.362885	0.606217	0.273877	0.044593
7	Cyclist	Person	0.405060	0.424271	0.467193	0.599011	0.414666	0.533102	0.131818	0.019210



Sample image and bounding box labels

Kitti is especially interesting data set, and more real-life type of data set. In kitti, images are organized by varying level of difficulty, and images have different types of issues that one may encounter in a more realistic data set. These issues can broadly be grouped into 3 types,

Occlusion: Occlusion happens when one object is either partially or completely occluded by

	0	label	xmin	xmax	ymin	ymax	x_c	y_c	h	w
0	Truck	Vehicle	0.200798	0.480927	0.327727	0.624947	0.340862	0.476337	0.297219	0.280129
1	Car	Vehicle	0.922280	1.000000	0.438930	0.534599	0.961140	0.486765	0.095668	0.077720
2	Car	Vehicle	0.450298	0.473263	0.475989	0.528021	0.461781	0.502005	0.052032	0.022965
3	Car	Vehicle	0.412361	0.439388	0.478209	0.531016	0.425874	0.504612	0.052807	0.027027
4	Car	Vehicle	0.465842	0.484384	0.472487	0.520401	0.475113	0.496444	0.047914	0.018541
5	Car	Vehicle	0.430306	0.452869	0.480214	0.524893	0.441587	0.502553	0.044679	0.022562
6	Pedestrian	Person	0.242039	0.271587	0.460561	0.728476	0.256813	0.594519	0.267914	0.029549

Example of occlusion

Overcrowded image: Most bounding box prediction methods predict a fixed number of bounding boxes. However, if the images is overcrowded, the algorithm will try to lump

	0	label	xmin	xmax	ymin	ymax	x_c	y_c	h	w
0	Car	Vehicle	0.592160	0.806873	0.415722	0.923155	0.699517	0.669439	0.507433	0.214714
1	Car	Vehicle	0.000000	0.150483	0.504439	1.000000	0.075242	0.752219	0.495561	0.150483
2	Car	Vehicle	0.000000	0.218912	0.506738	0.890481	0.109456	0.698610	0.383743	0.218912
3	Car	Vehicle	0.029331	0.260935	0.477727	0.765134	0.145133	0.621430	0.287406	0.231604
4	Car	Vehicle	0.150459	0.347542	0.421578	0.683422	0.249001	0.552500	0.261845	0.197083
5	Car	Vehicle	0.124005	0.302933	0.462487	0.725214	0.213469	0.593850	0.262727	0.178928
6	Car	Vehicle	0.564134	0.665657	0.479840	0.702086	0.614895	0.590963	0.222246	0.101523
7	Car	Vehicle	0.204778	0.340677	0.432326	0.649759	0.272728	0.541043	0.217433	0.135898
8	Car	Vehicle	0.250548	0.368638	0.468957	0.631658	0.309593	0.550307	0.162701	0.118090
9	Car	Vehicle	0.254915	0.379807	0.457193	0.613503	0.317361	0.535348	0.156310	0.124891
10	Car	Vehicle	0.322087	0.408413	0.457968	0.563155	0.365250	0.510561	0.105187	0.086326
11	Car	Vehicle	0.349444	0.418832	0.439037	0.547674	0.384138	0.493356	0.108636	0.069388
12	Car	Vehicle	0.523400	0.559871	0.467353	0.561952	0.541636	0.514652	0.094599	0.036471
13	Car	Vehicle	0.546253	0.612216	0.469037	0.636658	0.579234	0.552848	0.167620	0.065963
14	Car	Vehicle	0.514142	0.541128	0.453770	0.533289	0.527635	0.493529	0.079519	0.026986

Crowded bounding boxes

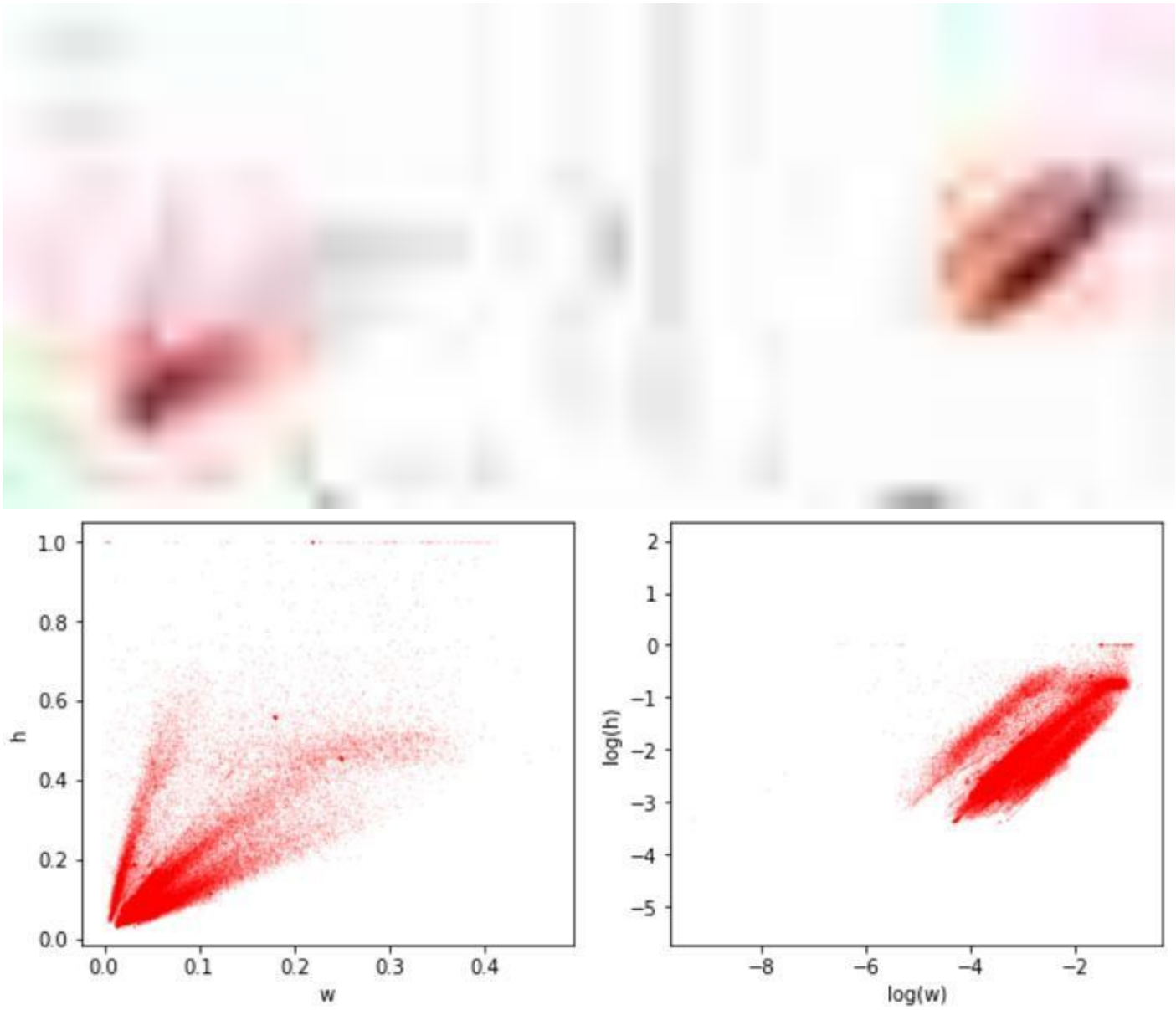
A nightmare for any human detection system

Incorrect annotations: The last issue is one where images are not completely annotated. For example, in the image below there are about 16 cars that are not tagged.

Mislabeled cars

3. Generating anchor boxes using K-means clustering

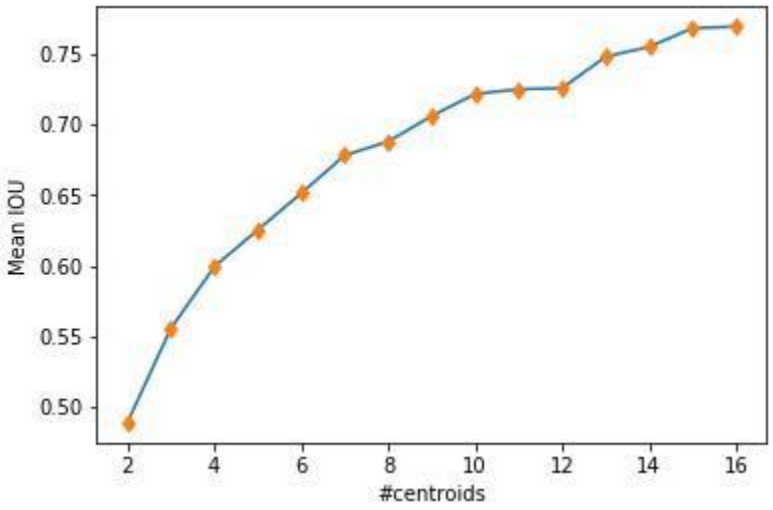
There are many ways to compute bounding boxes for detection tasks. One approach is to directly predict the bounding box values, however this approach is susceptible to errors as it tends to favor bounding boxes with large dimensions. Further, the training process is unstable because the range of values to predict can vary significantly. An alternate approach is to use template bounding boxes called anchor boxes or priors, and then use corrections on top of these anchor boxes to match the ground truth bounding box dimensions. In other models, like single shot detection (SSD), corrections are made on top of bounding box of fixed hand-selected sizes and aspect ratios. For example in SSD, there are 9 anchor/prior boxes are predicted per cell, based on hand selected aspect ratio and sizes. This however, does not guarantee that the anchor boxes are good candidates for bounding boxes. In YOLO, no anchor boxes are used and bounding box locations and dimensions are predicted directly. In YOLOv2, the first step is to compute good candidate anchor boxes. This is achieved using K-means clustering. However, using direct Euler distance metric for K-means minizers error for larger bounding boxes, but not for smaller boxes. Therefore, in YOLOv2, intersection over union (IOU) is used as a distance metric. The IOU calculations are made assuming all the bounding boxes are located at one point, i.e. only width and height are used as features. Figure below shows the height and width plotted against each other. Fixed slopes indicate that most bounding boxes have specific predefined aspect ratios, and size. This is not surprising given the fact that a person and vehicle are expected to have certain fixed dimensions.



I next used K-means clustering to compute cluster centers (centroids). As it is not clear how many centroids to use, I computed different number of clusters and computed the mean of maximum IOU between the bounding box and individual anchors. Figure below presents IOU vs number centroids data for kitti dataset.

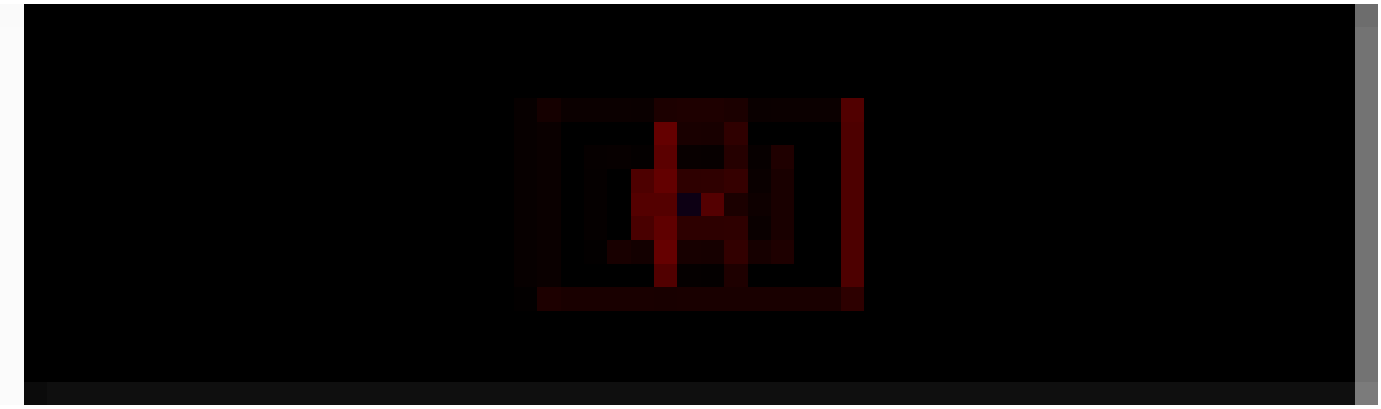
```
N# : 2 , IOU :  0.488644484481 , error : 9.29357749759e-05 , Loop : 11
N# : 3 , IOU :  0.55532066708 , error : 0.00011809146607 , Loop : 22
N# : 4 , IOU :  0.599185129049 , error : 0.000244664123957 , Loop : 22
N# : 5 , IOU :  0.62527360444 , error : 0.00359188677503 , Loop : 22
N# : 6 , IOU :  0.651239145788 , error : 0.000533576011715 , Loop : 22
N# : 7 , IOU :  0.678165404301 , error : 0.000801931966525 , Loop : 22
N# : 8 , IOU :  0.687932581035 , error : 0.00043891109532 , Loop : 22
N# : 9 , IOU :  0.706034723961 , error : 0.00131586814734 , Loop : 22
N# : 10 , IOU :  0.721710654919 , error : 0.00127953215042 , Loop : 22
N# : 11 , IOU :  0.724996568579 , error : 0.00181883498696 , Loop : 22
N# : 12 , IOU :  0.725789318078 , error : 0.000545839581194 , Loop : 22
N# : 13 , IOU :  0.748057540328 , error : 0.000740205429945 , Loop : 22
N# : 14 , IOU :  0.755013937165 , error : 0.000665192859546 , Loop : 22
N# : 15 , IOU :  0.76822900017 , error : 0.00185493962013 , Loop : 22
N# : 16 , IOU :  0.76944218528 , error : 0.000717979850448 , Loop : 22
```

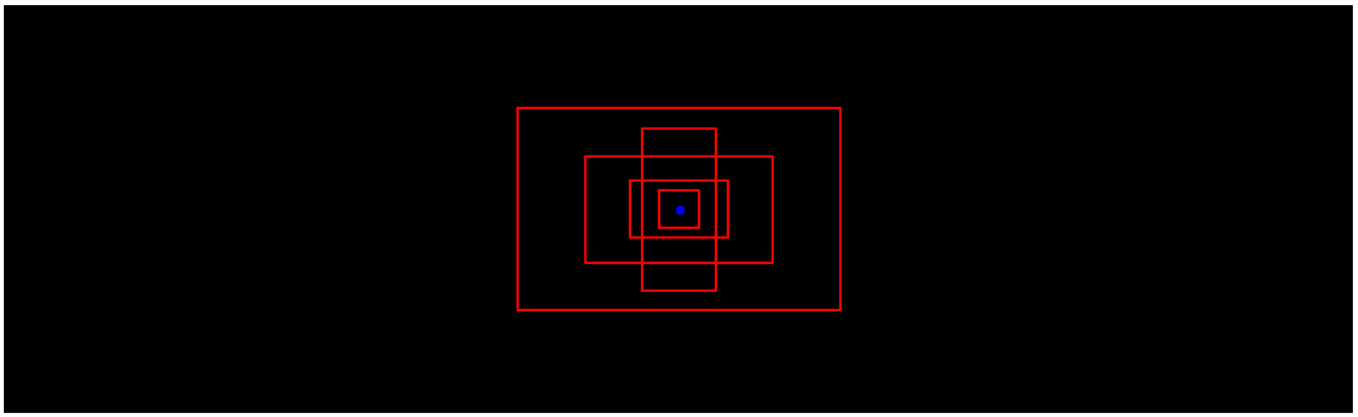
mean IOU vs number of centroids



mean IOU vs number of centroids

After plotting number of centroids vs mean IOU, it is clear that as number of centroids increase, the mean IOU between anchor boxes and bounding boxes plateaus. Choosing large number of prior boxes will allow for greater overlap between anchor boxes and bounding boxes, however, as the number of anchor boxes increase, the number of convolution filters in prediction filters increase linearly. Having 5 vs 10 bounding boxes will result in 13X13X35 vs 13X13X70 predictions, this will result in large network size and increased training time. So I stuck with 5 anchor boxes, to stay true with the original YOLOv2 implementation. At 5 anchor boxes, mean IOU was above 65%. The anchor boxes from K-means are plotted below,



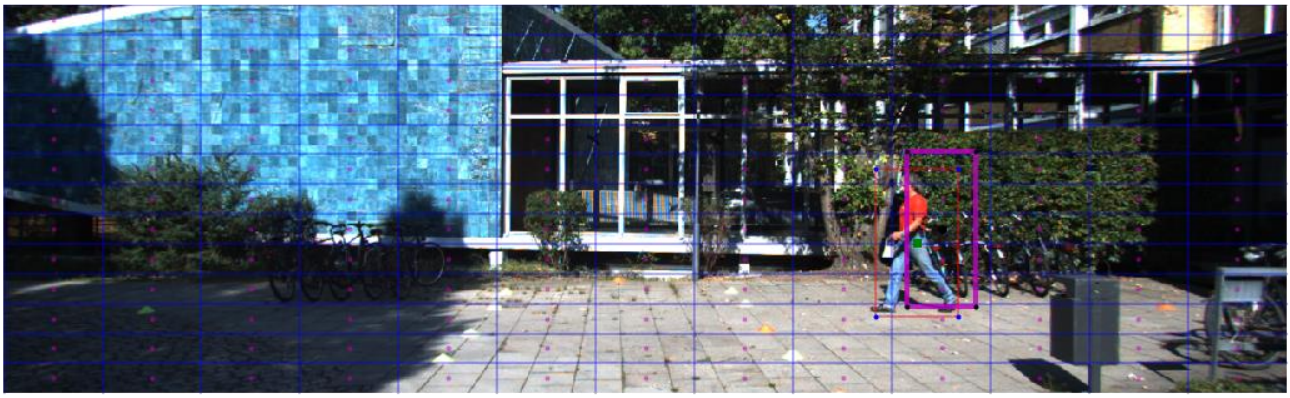


Predicted anchor boxes

As can be seen above, each anchor box is specialized for particular aspect ratio and size. A clearer picture is obtained by plotting anchor boxes on top of the image. In YOLOv2, an image is divided into 13X13 grid, and bounding box and class predictions are made for each anchor box located at those locations. The appropriate bounding box is selected as the bounding box with highest IOU between the ground truth box and anchor box. Note that this trick of assignment ensures that an anchor box predicts ground truth for an object centered at its own grid center, and not a grid cell far away (like YOLO may). Figure below presents anchor boxes and ground truth labels plotted on top of one another. Note that the anchor box that is responsible to predict a ground truth label is chosen as the box that gives maximum IOU when placed at the center of the ground truth box, i.e. only size is considered while assigning ground truth boxes. The location of anchor boxes is the center of the cell in which the center of the ground truth box falls. Figures below show the ground truth label (in green) and the assigned anchor box (in magenta). As we computed anchor boxes directly from the data set, we see a high overlap between the shapes of ground truth boxes and anchors.

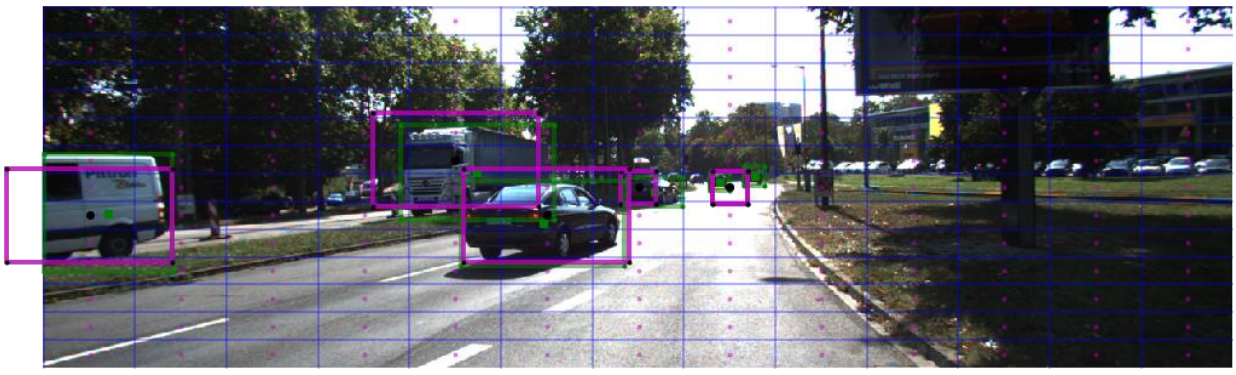
Anchors and ground truth boxes for easy case

Figure below presents bounding box for a person (pedestrian) note that the shape of the bounding box is very close to the true shape of the person, and a minor correction in location of bounding box will result in them overlapping well.



Anchor box and ground truth box for a pedestrian

Below are some more examples of ground truth bounding boxes and anchor boxes.



Conclusion

In this post, I covered the concept of generating candidate anchor boxes from bounding box data, and then assigning them to the ground truth boxes. The anchor boxes or templates are computed using K-means clustering with intersection over union (IOU) as the distance measure. The anchors thus computed do not ignore smaller boxes, and ensure that the resulting anchors ensure high IOU between ground truth boxes. In generating the target for training, these anchor boxes are assigned or are responsible for predicting one ground truth bounding box. The anchor box that gives highest IOU with the ground truth data when located at its center is responsible for predicting that ground truth label. The location of the anchor box is the center of the grid cell within which the ground truth box falls.

894

- Machine Learning
- Deep Learning
- Self Driving Cars
- Artificial Intelligence

894 claps

WRITTEN BY

Vivek Yadav

Follow

Staff Software Engineer at Lockheed Martin-Autonomous System with research interest in control, machine learning/AI. Lifelong learner with glassblowing problem.