Week 11 | Under Construction

What has been done this week

Mainly been writing report. Messed around with some triplet loss code at some point for the dataset that Anders and Shihav suggested. Need some more modification before it can correctly load the data.

- [ ] Report
    - [x] MobileNet
    - [x] ResNet
    - [x] SSD
    - [x] IOU
    - [x] Golf Club Terminology
    - [x] FaceNet / Triplet Loss
    - [ ] Method
        - [x] Detection
        - [ ] Classification
    - [ ] Conclusion and outlook
    - [ ] Baseball
    - [x] Tesseract
    - [x] Docker

- [x] OpenCV object tracker speed

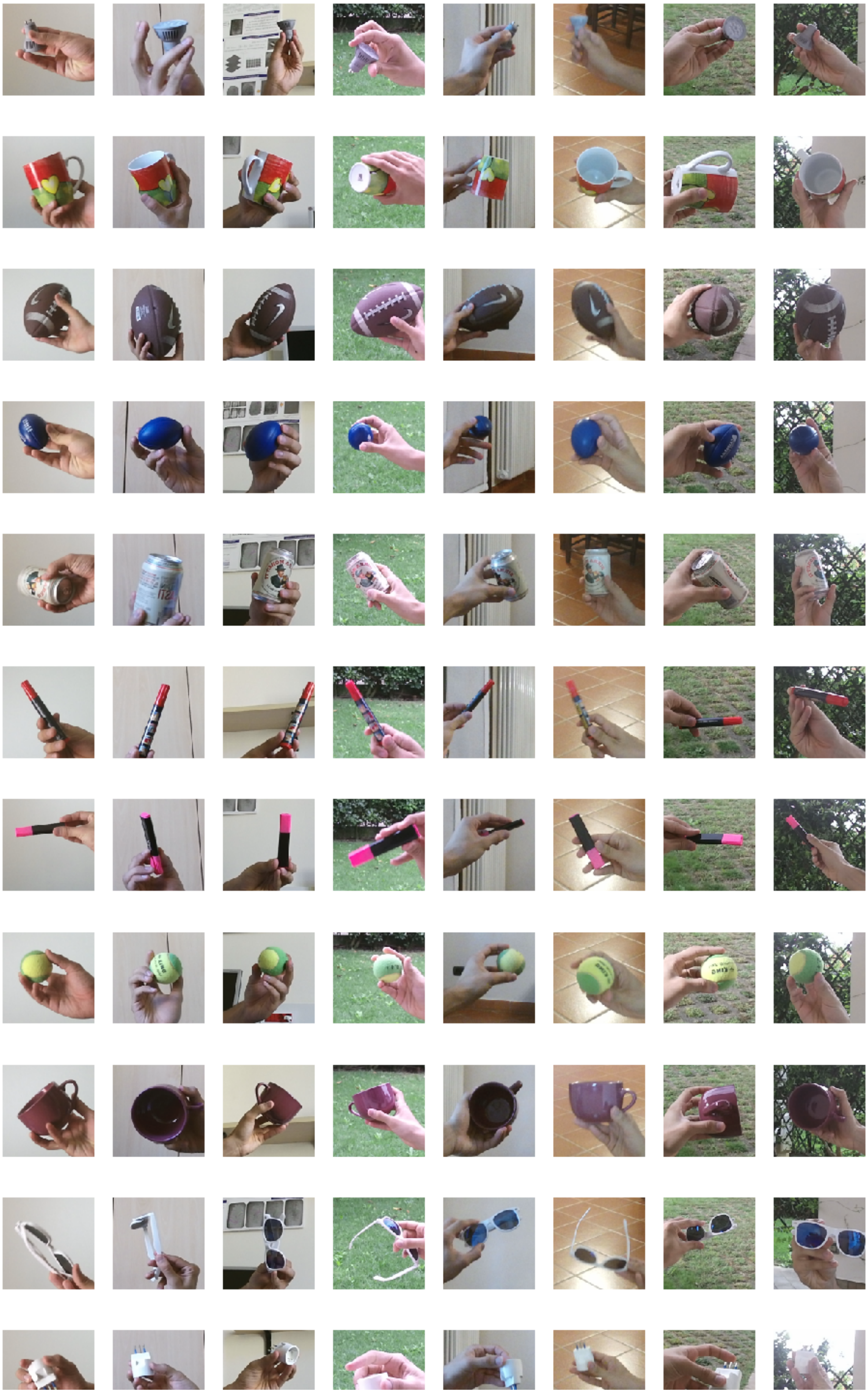
  1. **BOOSTING Tracker:** Based on the same algorithm used to power the machine learning behind Haar cascades (AdaBoost), but like Haar cascades, is over a decade old. This tracker is slow and doesn't work very well. Interesting only for legacy reasons and comparing other algorithms. *(minimum OpenCV 3.0.0)*
  2. **MIL Tracker:** Better accuracy than BOOSTING tracker but does a poor job of reporting failure. *(minimum OpenCV 3.0.0)*
  3. **KCF Tracker:** Kernelized Correlation Filters. Faster than BOOSTING and MIL. Similar to MIL and KCF, does not handle full occlusion well. *(minimum OpenCV 3.1.0)*
  4. **CSRT Tracker:** Discriminative Correlation Filter (with Channel and Spatial Reliability). Tends to be more accurate than KCF but slightly slower. *(minimum OpenCV 3.4.2)*
  5. **MedianFlow Tracker:** Does a nice job reporting failures; however, if there is too large of a jump in motion, such as fast moving objects, or objects that change quickly in their appearance, the model will fail. *(minimum OpenCV 3.0.0)*
  6. **TLD Tracker:** I'm not sure if there is a problem with the OpenCV implementation of the TLD tracker or the actual algorithm itself, but the TLD tracker was incredibly prone to false-positives. I do not recommend using this OpenCV object tracker. *(minimum OpenCV 3.0.0)*
  7. **MOSSE Tracker:** Very, *very* fast. Not as accurate as CSRT or KCF but a good choice if you need pure speed. *(minimum OpenCV 3.4.1)*
  8. **GOTURN Tracker:** The only deep learning-based object detector included in OpenCV. It requires additional model files to run (will not be covered in this post). My initial experiments showed it was a bit of a pain to use even though it reportedly handles viewing changes well (my initial experiments didn't confirm this though). I'll try to cover it in a future post, but in the meantime, take a look at Satya's writeup. *(minimum OpenCV 3.2.0)*

```
BOOSTING FPS: 30.609743172087462
MIL FPS: 16.56634784404572
KCF FPS: 204.1915752496908
TLD FPS: 24.685895385358393
MEDIANFLOW FPS: 153.2571068724692
GOTURN FPS: 21.2797809638136
MOSSE FPS: 1084.584846721536
CSRT FPS: 41.854202636453884
```

Tracker performance (by visual performance):

Algorithm	Performance (Visual inspection)	FPS
Boosting	*Bad*	30.6
MIL	OK	16.5
KCF	Bad-OK	204
TLD	Bad	24.6
MedianFlow	Bad-Ok	153
GOTURN	Bad	21.3
MOSSE	Ok	1084
CSRT	Good	41.8

- [x] Read SSD Paper
- [x] Look at the data from Anders and Shihav



Golf Terminology

resources: https://www.golf-drives.com/blog/golf-slang/ , https://golfsupport.com/blog/what-are-the-different-types-of-golf-clubs-available/, https://www.livestrong.com/article/139541-list-different-types-golf-clubs-uses/

Fairway: The centre, short-mown portion of a golf hole in between the teeing ground and the green.

Tee: The area of a golf hole where the ball is first struck.

Putting Green: The area of very low cut grass where the hole is located and putting is done.

Golf Club Types

Wood: Long-distance clubs with a large head and a long shaft used to drive the ball a great distance. A golf set usually has two to three fairway woods and a driver. The fairway woods are, as their names suggest typically used in the fairway, where the irons are out of reach. The driver usually has a loft between 7-11 degrees and the woods has loft between 12-20 degrees to produce higher shots.

Iron: Club with a solid metal head, with a flat angled face and shorter shaft than a wood. Irons are very versatile, but are generally used from the fairway or tee on short holes. Typically numbered from 1 to 9 where the shafts are getting shorter, loft angles higher and club heads heavier as the number increases.

Wedge: A subclass of irons with a greater loft. They usually start at a loft of 47-48 degrees, where the 9 iron usually end at 44-45. They are used for short-distance, high-altitude and high-accuracy shots, usually getting the ball onto the green from tricky position. In golf terms they are typically broken into 4 categories: pitching wedge, gap wedge, sand wedge and lob wedge. For classification, they are usually marked with either their loft, e.g "60" for a 60 degree loft or a letter like "P" for pithing wedge

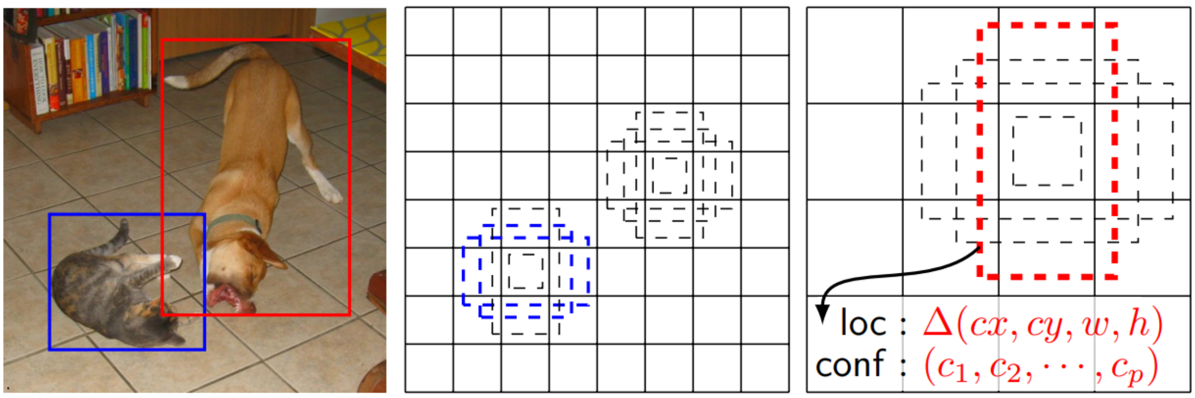
Hybrid: Newest category of club and is a cross between a wood and an iron. Numbered like irons, but doesn't go that high because the number corresponds to the iron that they replace. Most typical hybrids is a 2-hybrid and a 3-hybrid. Used to hit semi-long distance shots, and the design makes it easier to hit shots (like irons), while having the more long-range features of the wood.

Putter: A flat faced club used on or around the green as the last club to get the ball into the hole. They are designed to roll the ball along the green and thus have very little loft (typically 5 degrees). Putters have a large variance in shapes and sizes, since choosing a putter is a quite personal process.

Literature

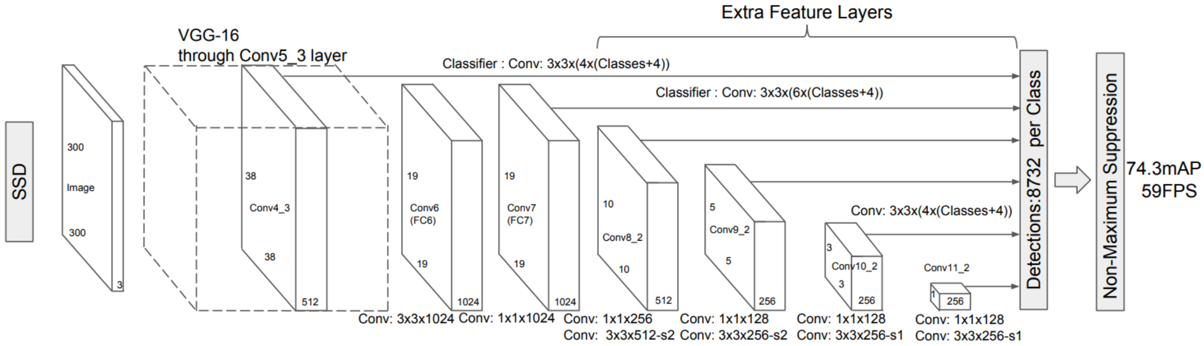
SSD: Single Shot MultiBox Detector https://arxiv.org/pdf/1512.02325.pdf

- Similair to Faster-RCNN in the sense that it also used anchors.
- To detect objects at multiple scales, it uses feature maps of different sizes. Small feature maps focuses on objects at larger scale.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

-



•

Training objective The SSD training objective is derived from the MultiBox objective [7,8] but is extended to handle multiple object categories. Let $x_{ij}^p = \{1, 0\}$ be an indicator for matching the i-th default box to the j-th ground truth box of category p. In the matching strategy above, we can have $\sum_i x_{ij}^p \geq 1$. The overall objective loss function is a weighted sum of the localization loss (loc) and the confidence loss (conf):

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{1}$$

where N is the number of matched default boxes. If $N = 0$, wet set the loss to 0. The localization loss is a Smooth L1 loss [6] between the predicted box (l) and the ground truth box (g) parameters. Similar to Faster R-CNN [2], we regress to offsets for the center (cx, cy) of the default bounding box (d) and for its width (w) and height (h).

$$\begin{aligned} L_{loc}(x, l, g) &= \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \mathrm{smooth_{L1}}(l_i^m - \hat{g}_j^m) \\ \hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w & \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h \\ \hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) & \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right) \end{aligned} \tag{2}$$

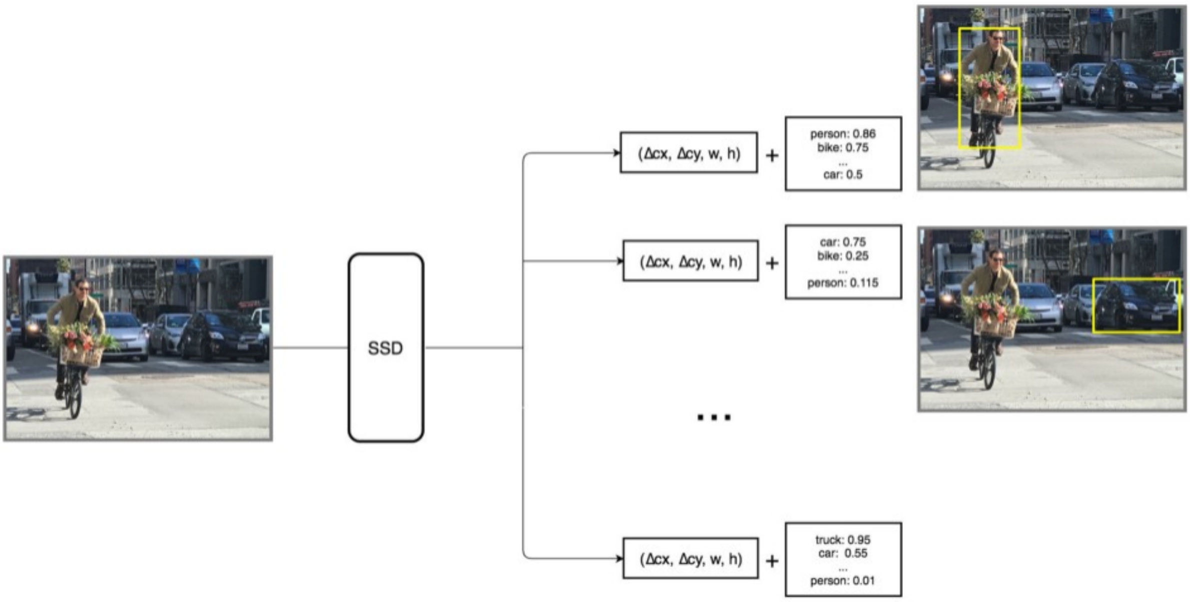
The confidence loss is the softmax loss over multiple classes confidences (c).

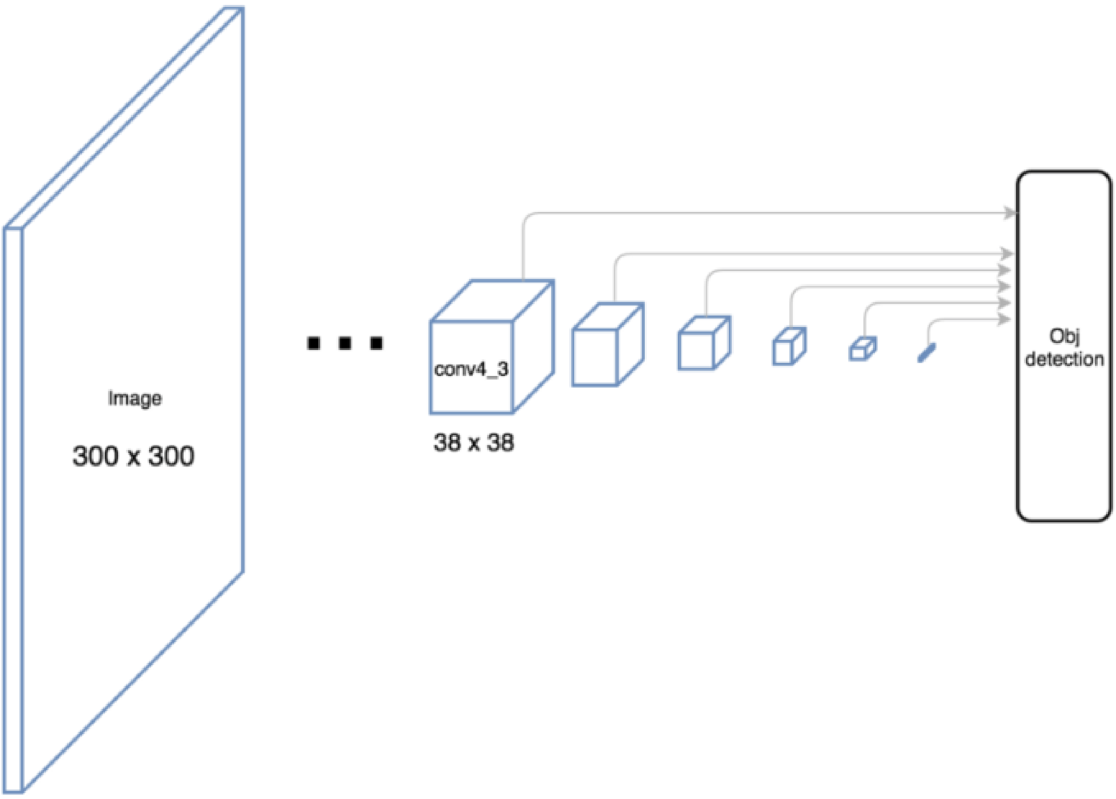
$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^p log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \tag{3}$$

and the weight term α is set to 1 by cross validation.

•


1
2
3
4
(4, 5)


SSD
(Δcx, Δcy, w, h)
person: 0.86
bike: 0.75
...
car: 0.5
(Δcx, Δcy, w, h)
car: 0.75
bike: 0.25
...
person: 0.115
...
(Δcx, Δcy, w, h)
truck: 0.95
car: 0.55
...
person: 0.01



- Matching:
  - For each groundtruth box:
    - Match with prior with highest IOU
  - Foe each prior:
    - IOU_list = IOU(prior, ground_truth_boxes)
    - If max(IOU_list) > threshold
      - Match prior with ground_truth_box argmax(IOU_List)
- Hard negative mining:
  - After the matching step, we will have unmatches_priors >>> matches_priors. --> Very unbalanced dataset --> Sort the unmatched_priors by their confidence loss and mix the hardest unmatches priors with matches priors in a ratio of 3:1

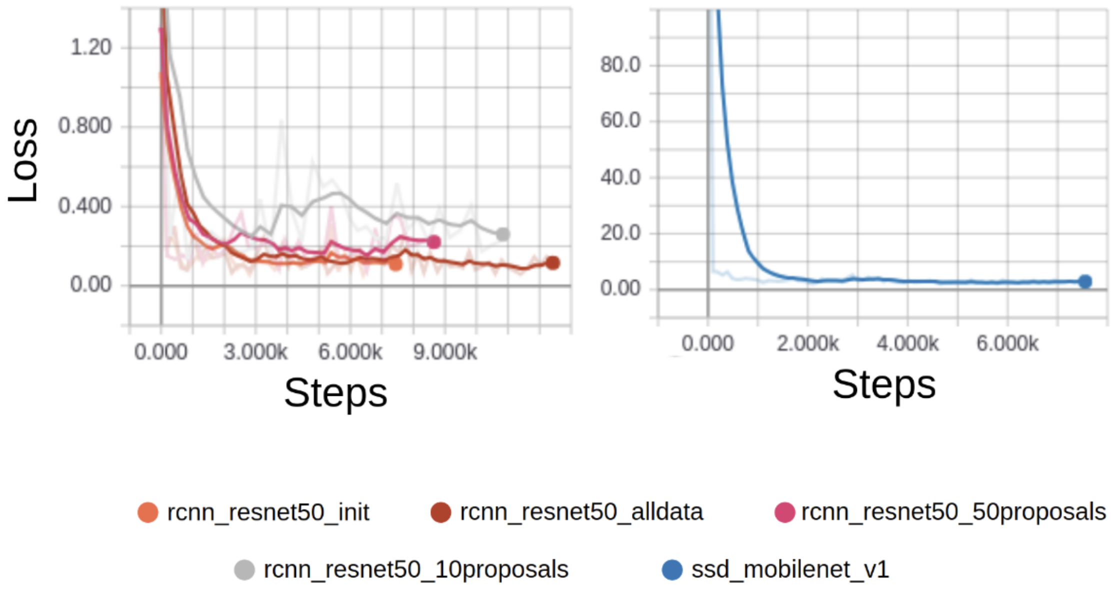


Figure 4.2: Loss during training of the 5 different detection models.

Model	Test IOU		Train IOU		FPS	Complete Miss
	Top1	Top5	Top1	Top5		(IOU = 0)
rcnn_resnet50_init	0.666	0.738	0.726	0.764	2.56	18.9%
rcnn_resnet50	**0.786**	**0.794**	**0.814**	**0.822**	2.55	**4.3%**
rcnn_resnet50_50proposals	0.756	0.769	0.774	0.785	5.61	7.1%
rcnn_resnet50_10proposals	0.672	0.582	0.724	0.726	6.99	18.9%
ssd_mobilenetv1	0.401	0.489	0.407	0.77	**25.4**	47%

Table 4.1: Performance of trained detection models.

What to do next week

High burnout at this point -> Will take an extended weekend to recharge.