Google AI Open Images:

Object detection & Visual Relationship

Konstantin Gavrilchik

Data: Open Images Dataset v4

Total images: 9M

Labeled images: 1.9M

Total size of resized to 512x512 images: ~600gb

Labels: 600 classes

Provided additional information: **class hierarchy**, **relationships between objects**

Test data

Total images: 100K

Total size of images: ~10gb

Labels: 500 classes

Submission limites:

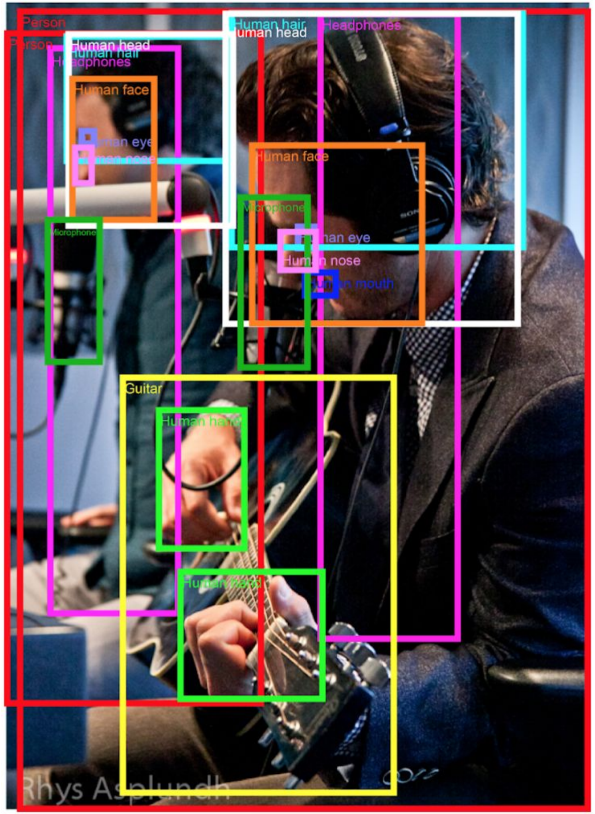
- 2gb for submission (unpacked)
- no more than 10 minutes for scoring

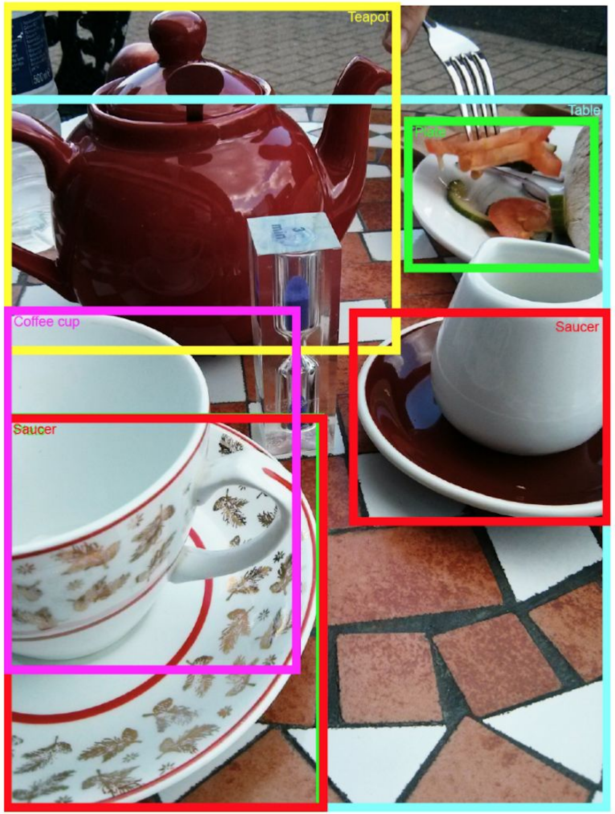
Hardware

CPU: Threadripper 1950X (32 threads)
GPU: 3x1080ti
RAM: 64gb

Thanks for n01z3 (Artur Kuzin)

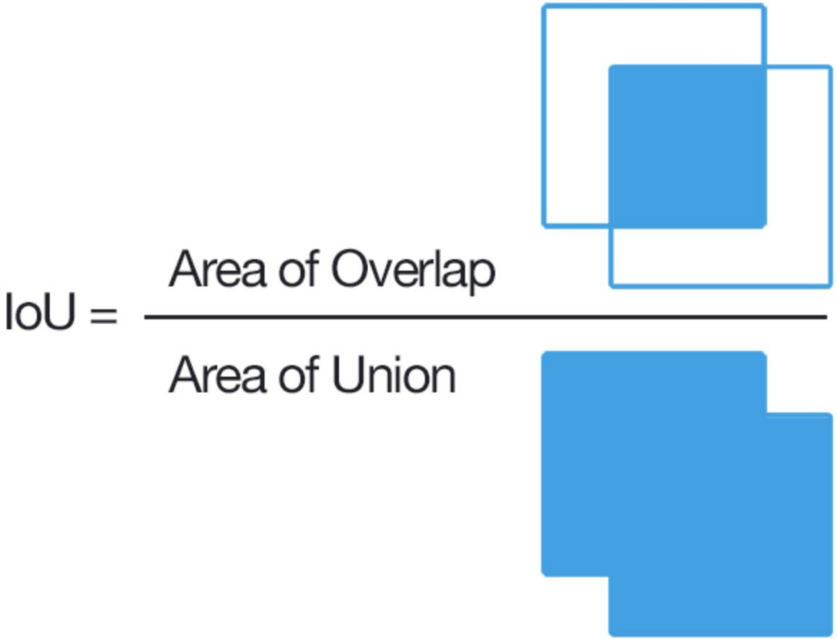
Data examples: Object detection



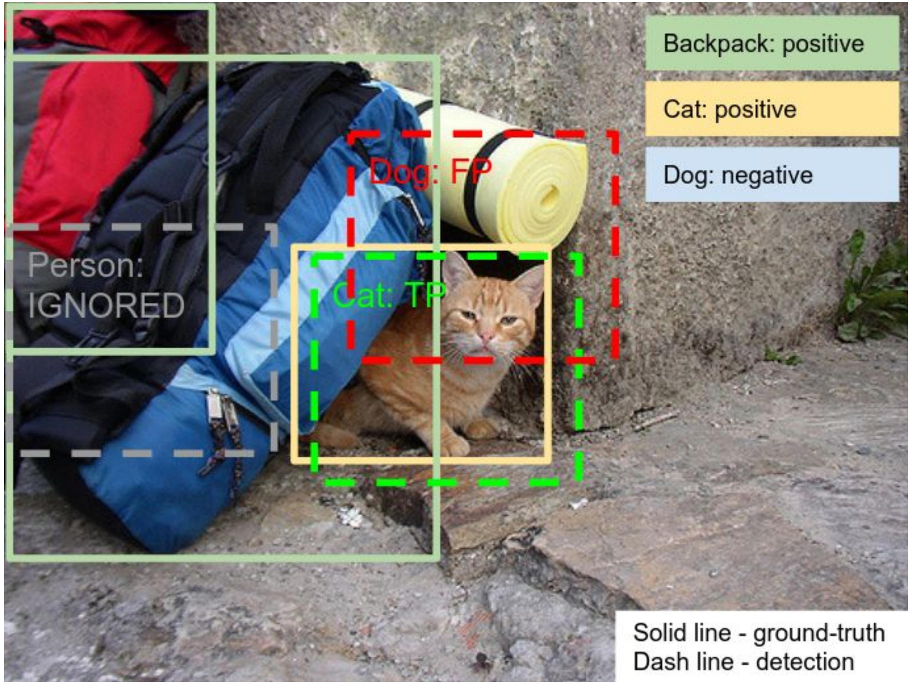


Metric: Object detection

mAP (mean average precision) at IoU > 0.5



$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Solution: pretrained models

Model 1: keras RetinaNet pretrained COCO (~0.07)

Model 2: tensorflow API (~0.23)

Model 3: keras RetinaNet trained on the given dataset (~0.27)

First solution

Model: pytorch RetinaNet (https://github.com/amirassov/fpnssd)

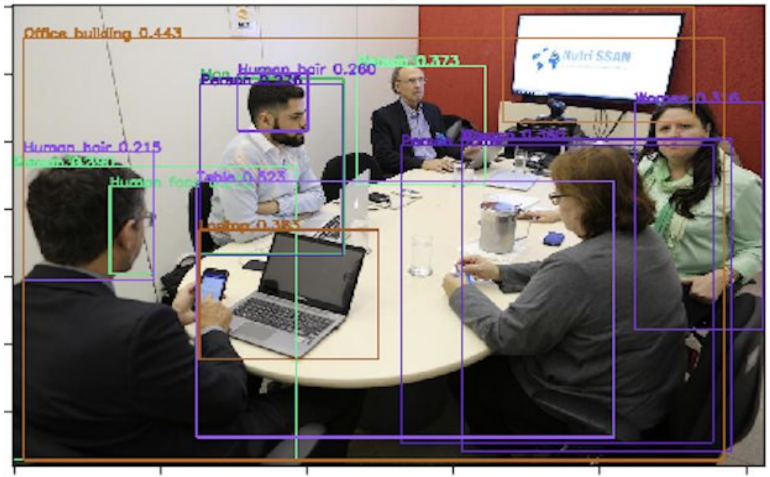
Training: ~20 epoch (1 weeks)

Augmentations: RandomGaussianNoise, Flips, Brightness, etc

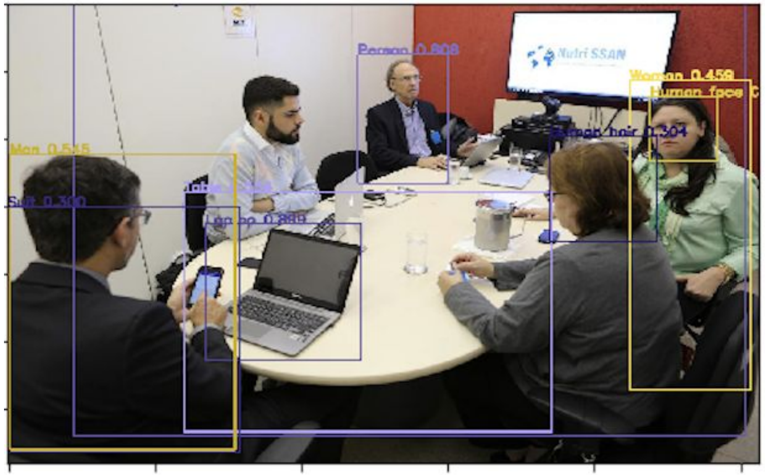
Inference: ~5 hours

+ non-maximum suppression

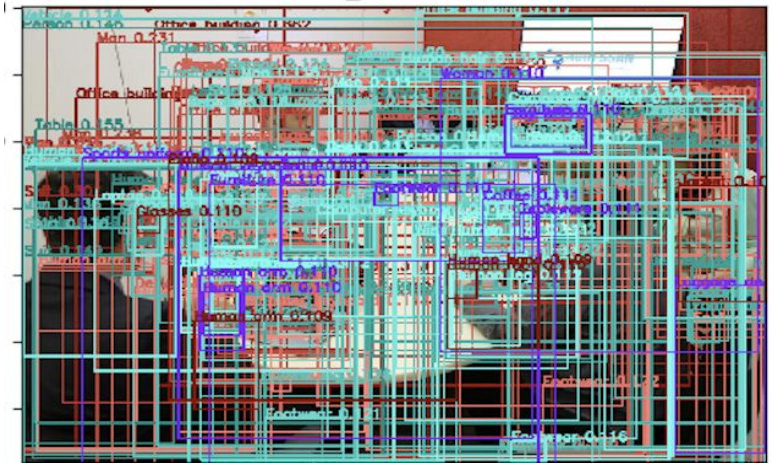
pytorch_0.03LB

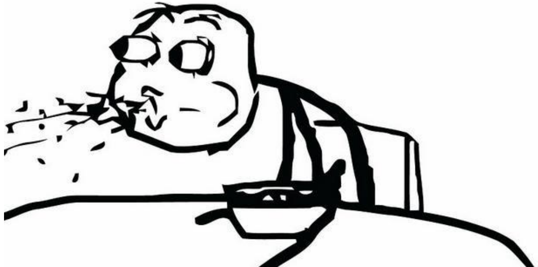


tf_0.23102LB

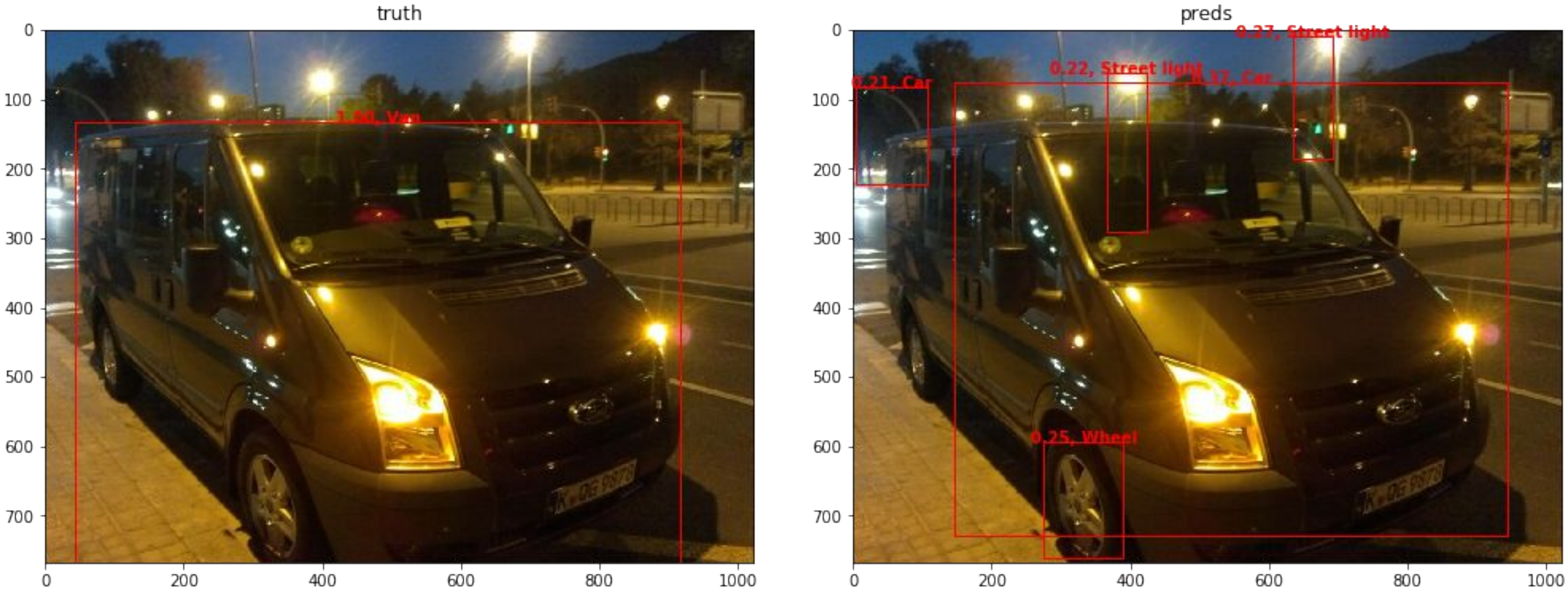


keras_0.27567LB





One more...


truth
1.00, Van
preds
0.27, Street light
0.22, Street light
0.37, Car
0.21, Car
0.25, Wheel
0
100
200
300
400
500
600
700
0
200
400
600
800
1000

Conclusion

- Train dataset labeled very bad
- Due to last point we have a lot of FP
- Metric does not penalized them

Best solution

- Train all models a little bit more
- Merge all predictions into one
- Apply NMS

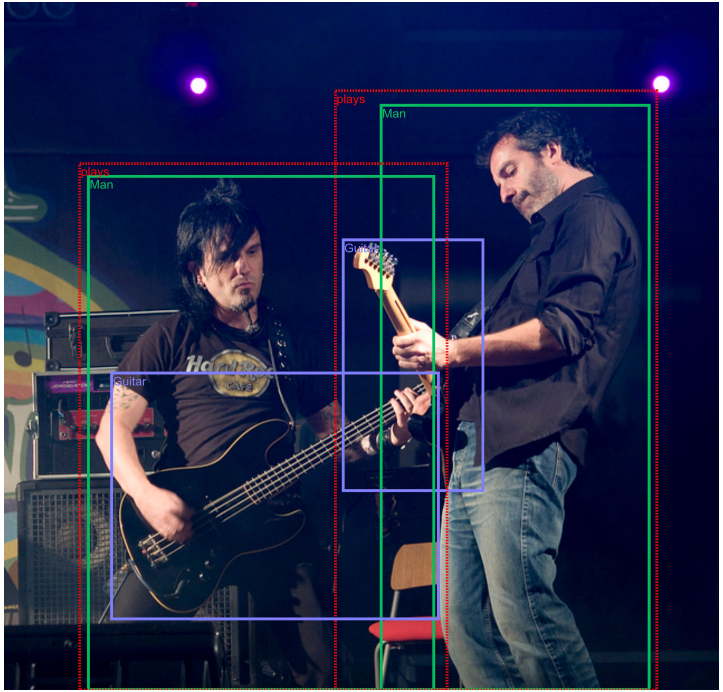
Final score: ~0.35 (45 place)

Other solutions

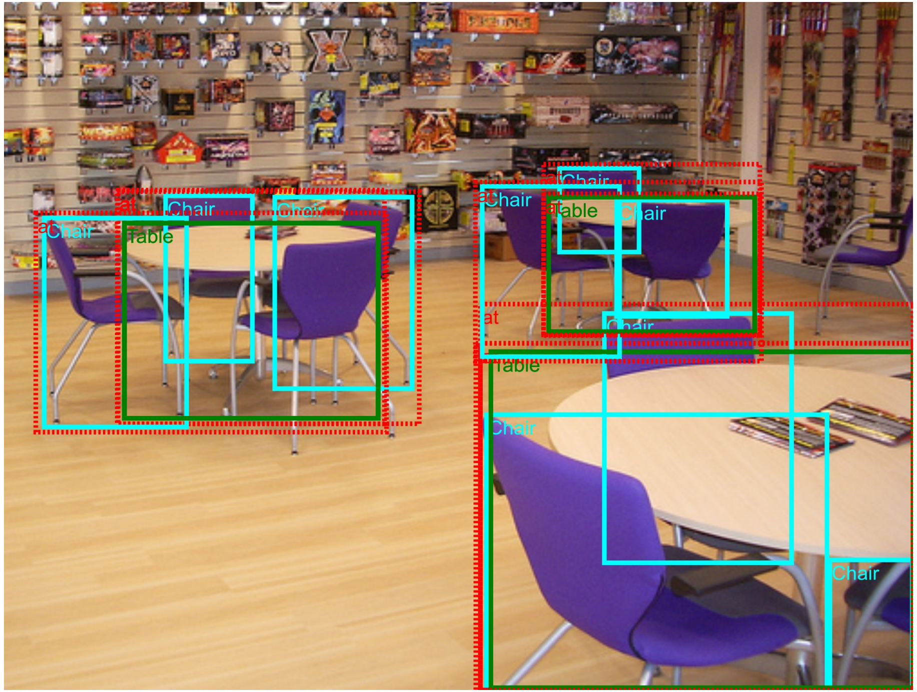
- ZFTurbo
  - Accurate data cleaning: score 0.45 (15 place)
  - Hierarchy of labels

- Seven asian guys
  - custom loss (published in paper after end of competition) + 512 GPU

Data examples: Visual Relationship



man playing guitar



chair at table

Metric: Visual Relationship

Weighed [0.4, 0.2, 0.4] of 3 metrics:

- mAP
- Recall@N with N=50
- mAP (taken over per-relationship APs)

Relationship types

329 unique triples: chair at table, man plays guitar, etc

10 unique relationship types:

is, at, on, holds, plays, inside_of, interact_with, wear, under, hits

Naive solution

Submit the most frequent triplets:

“chair at table” with median bounding boxes

Score: 0.00006

Heuristics

- We need accurate predictions (keras predictions is not good, let’s take pytorch RetinaNet with 0.03 score in the first competition)
- Run next algorithm:

```
for bbox1, bbox2 in product(preds, preds):
    if IoU(bbox1, bbox2) > 0.5:
        submit.append(bbox1 <most_common_relationship_type> bbox2)
```

- for each relationship type add simple heuristics:
  - <inside of> mean that the center of first bbox contains in the second

Score: 0.063 (28 place)

Thank you for your attention