LOK 1 - Object Localisation

Micha Birklbauer, Nicole Hölzl

Overview

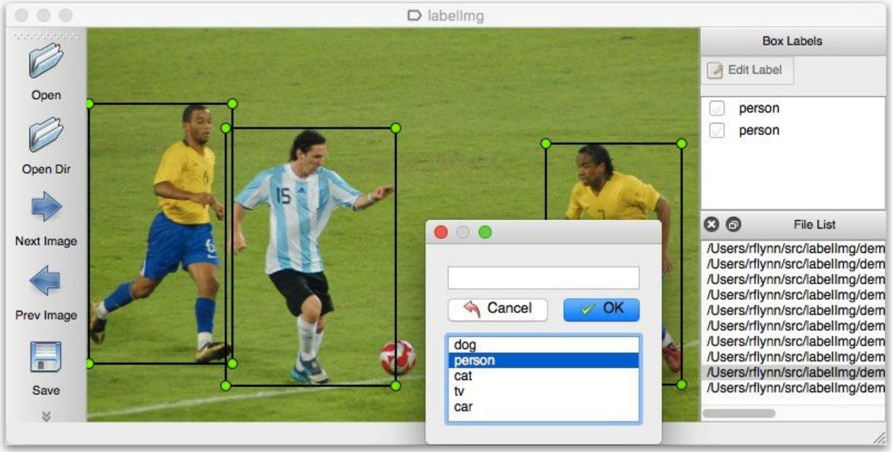
- Data
- Data Annotation
- Data Augmentation

MPII Human Pose Dataset

- 25K images extracted from YouTube videos
- 40K people with annotated body joints
- 410 human activities

Data Annotation

- 267 randomly selected images with single persons manually annotated
  - 217 for training
  - 50 for testing
- Tool: labelImg
  - Open-Source
  - Python
  - GUI
- https://github.com/tzutalin/labelImg

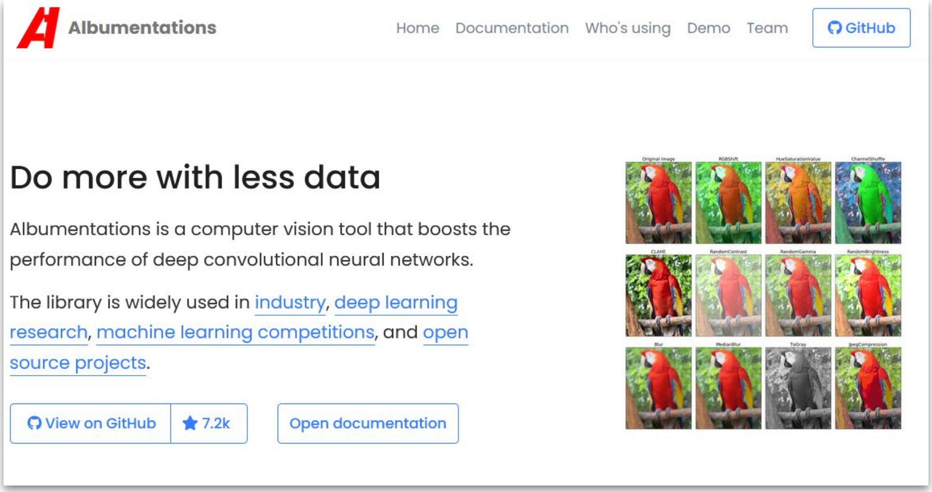


Data Augmentation

- Generation of new training images
- Transformations:
    - Horizontal Flip
    - Vertical Flip
    - Rotation
    - Gray colouring
    - Sepia colouring
    - Inverting
    - Posterizing
    - Normalizing

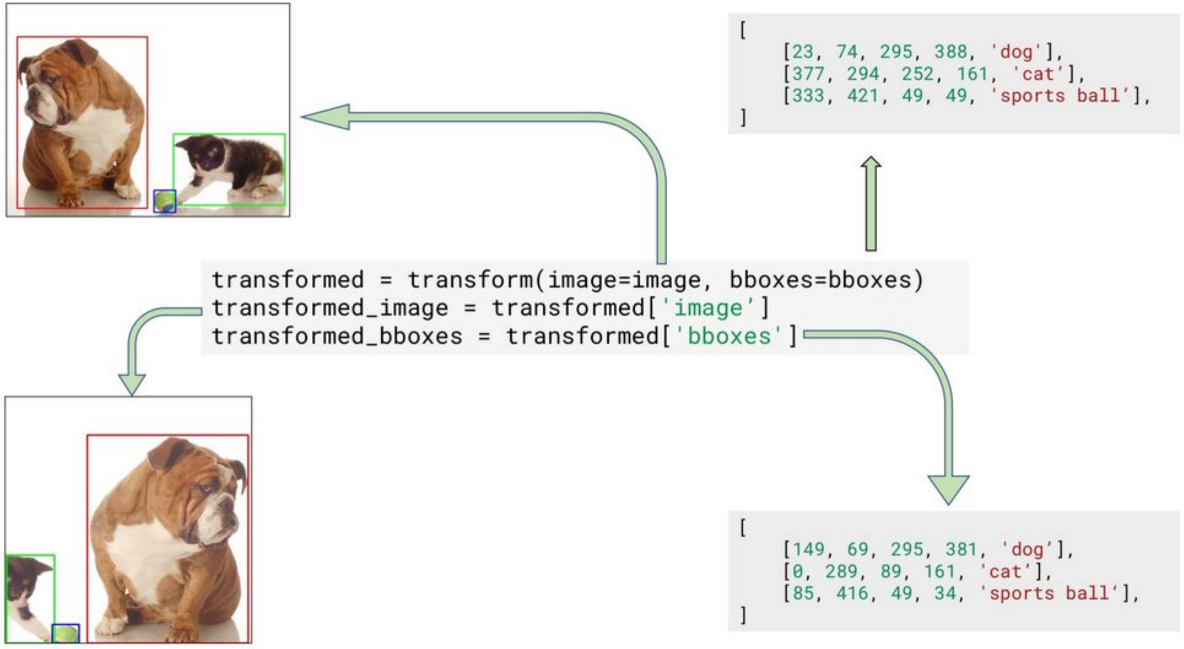
Data Augmentation - Albumentations

- Convenient tool to auto-transform bounding boxes
- Offers to define augmentation pipelines
- Good online documentation
- Easy to use
- Python package
- https://albumentations.ai/



Data Augmentation - Albuminations Example

```
transformed = transform(image=image, bboxes=bboxes)
transformed_image = transformed['image']
transformed_bboxes = transformed['bboxes']
```



Example input and output data for bounding boxes augmentation

Methods

- Haar Cascade
- YOLOv3
- EfficientDet

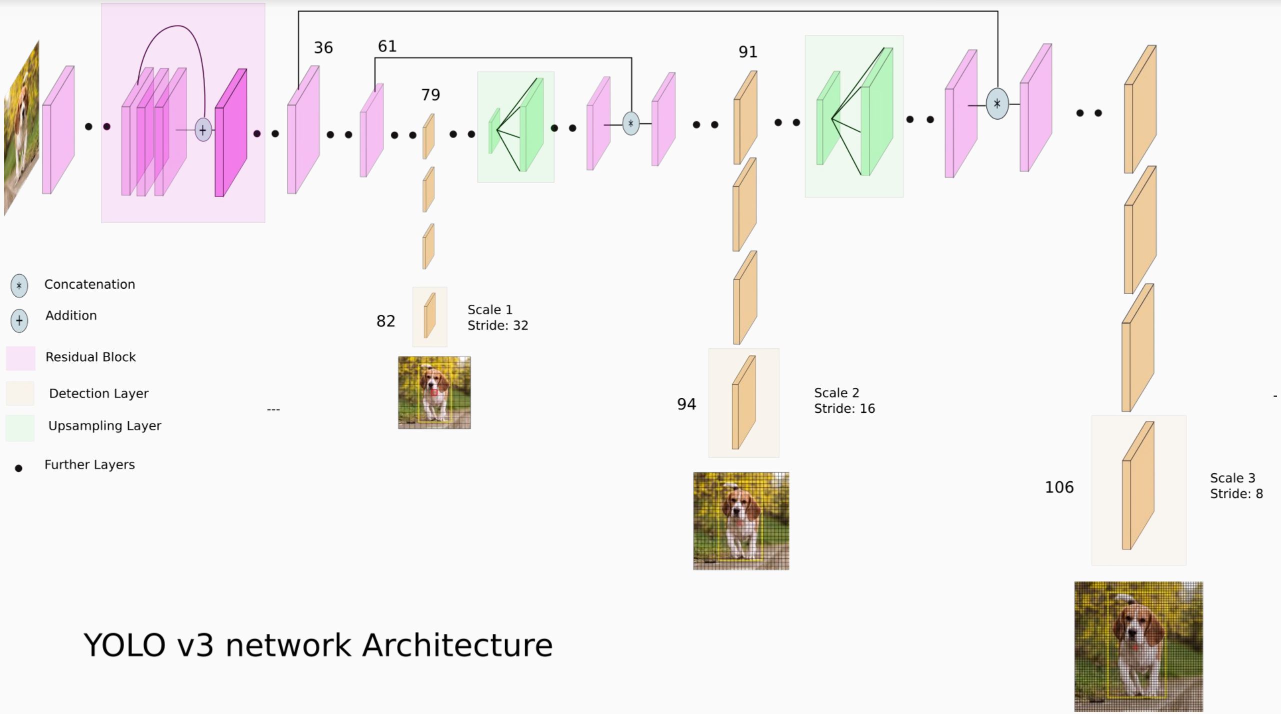
Haar Cascade

- Object localization using Haar Cascade Classifiers
- Training with normal or augmented images
- Training with normal images:
  - 217 positive samples
  - 300 negative samples
- Training with augmented images:
  - 1953 positive samples
  - 300 negative samples
- Negative samples mostly images of nature, landscapes etc.

YOLOv3

- The model is 3x faster then RetinaNet on COCO
- YOLO v3 uses a variant of Darknet which has 53 layers trained on Imagenet
- YOLO looks at the whole image context to predict the bounding boxes
- Input image size for the model is 416 x 416
- Trained on COCO 2017 (mAP 55.3)
- YOLOv3 implementation can be found here:
  - https://pjreddie.com/media/files/papers/YOLOv3.pdf

YOLOv3 - Architecture



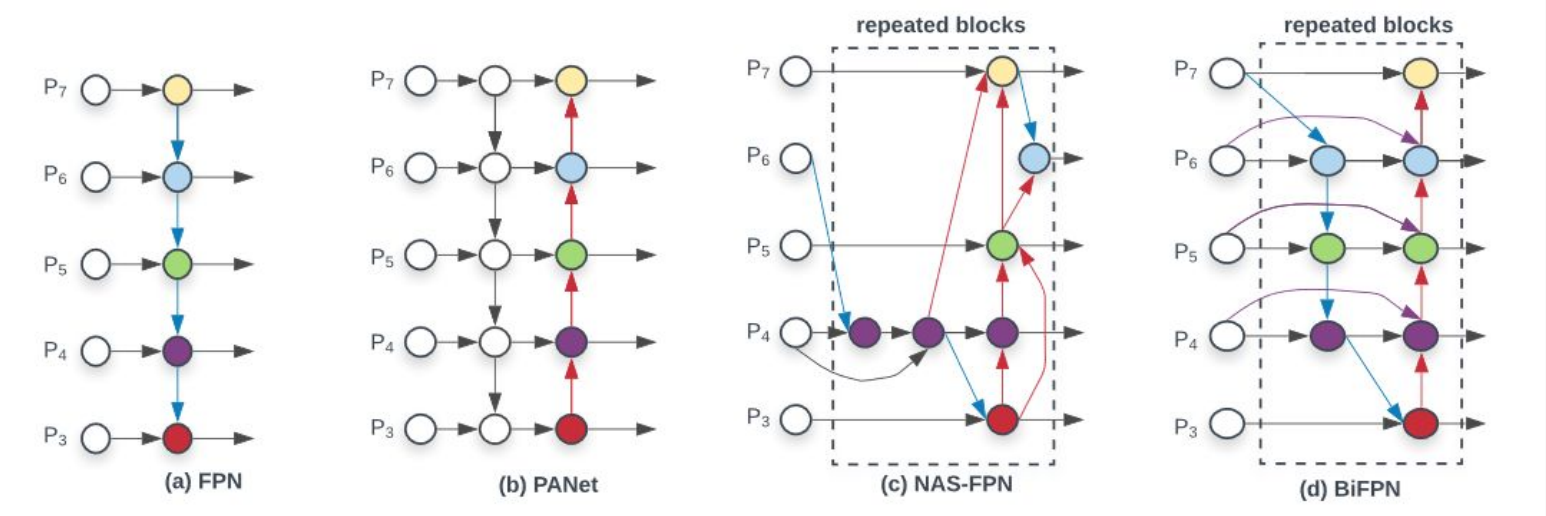
YOLO v3 network Architecture

EfficientDet - Tan et al., July 2020

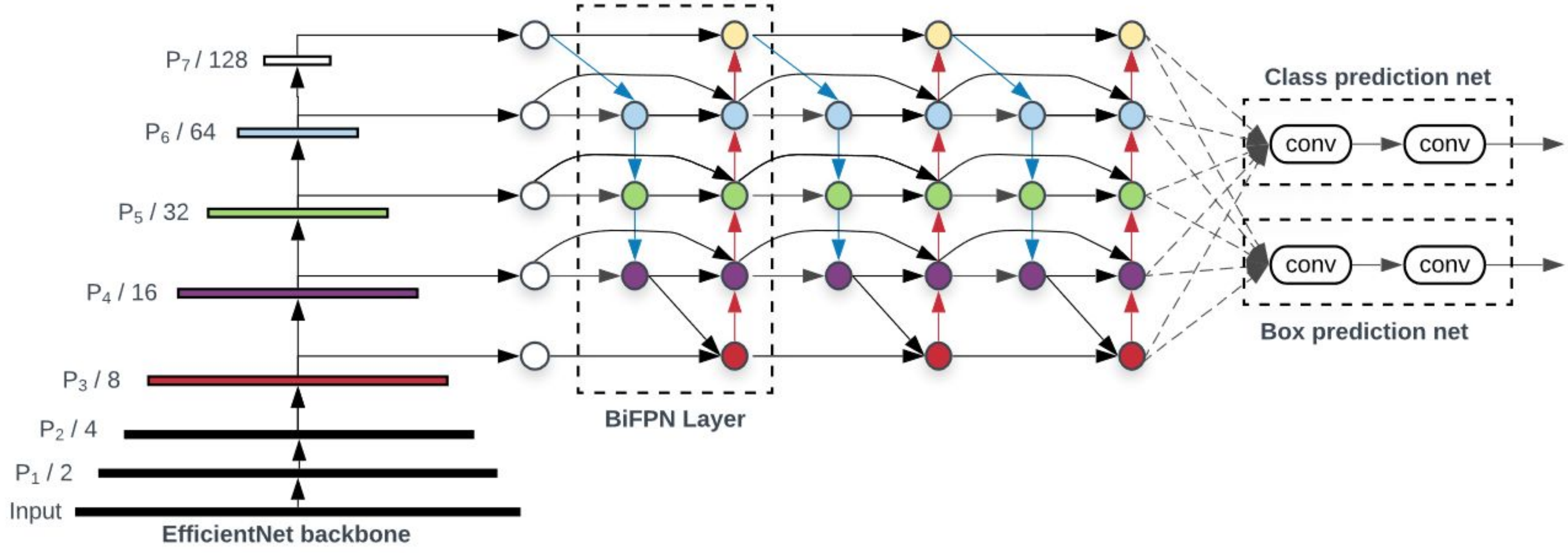
- Single Shot / One-Stage Detection
  - Take single shot to detect objects in image instead of two like in region proposal approaches.
- EfficientNet: Uniform scaling of depth, width and resolution of CNN by a compound coefficient.
  - 84.3% top-1 accuracy on ImageNet, while being 8.4x smaller and 6.1x faster on inference than the best existing CNN.
- BiFPN feature extraction
- Box and class predictors share the same features.
- Trained on COCO 2017 (mAP 45.4)
- Available via TensorFlow Hub
  - https://tfhub.dev/tensorflow/efficientdet/d3/1

EfficientDet - BiFPN

- Bi-directional Feature Pyramid Network



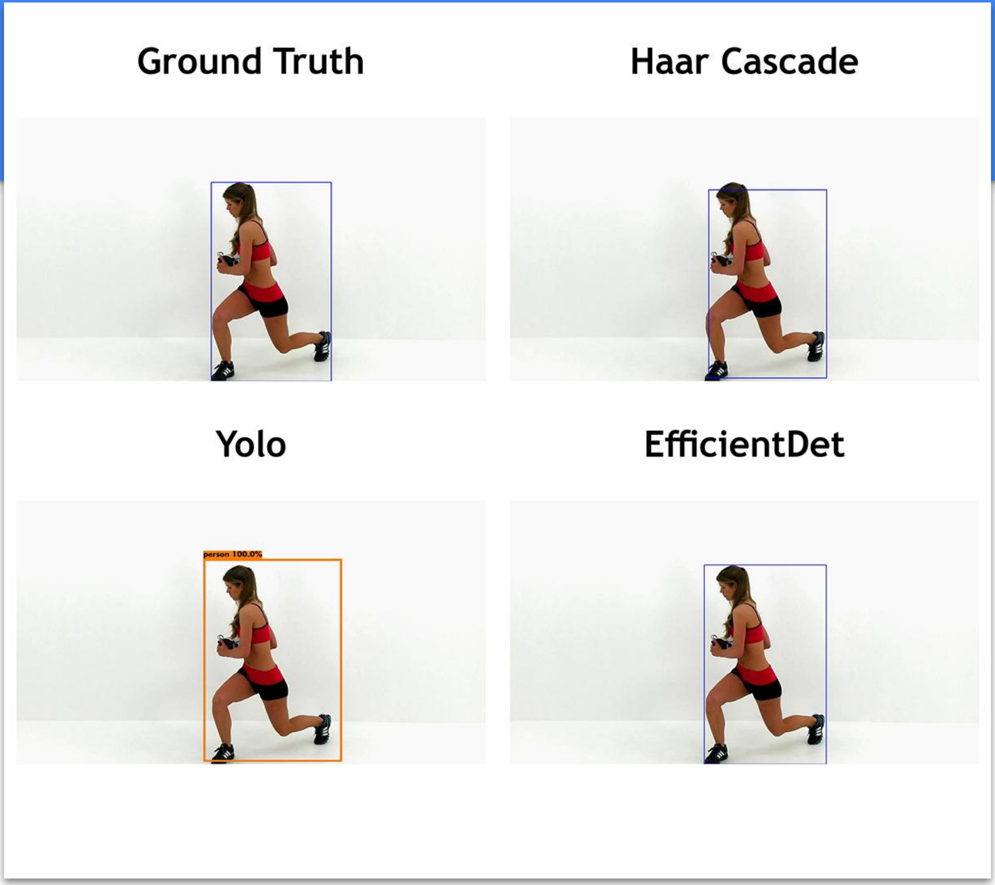
EfficientDet - Architecture



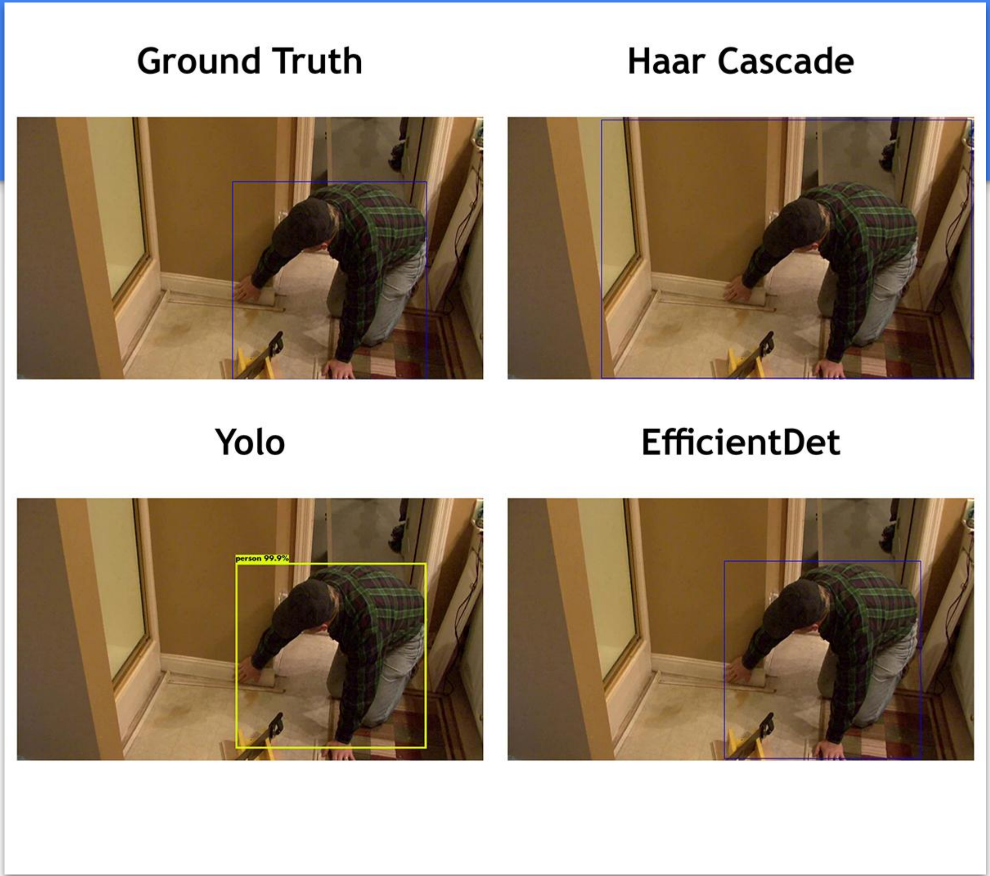
Results

- Haar Cascade
- YOLOv3
- EfficientDet

Results - Visual Comparison (The Good)

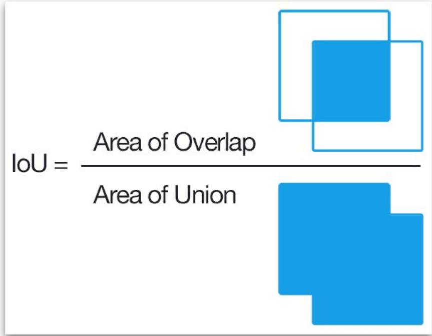


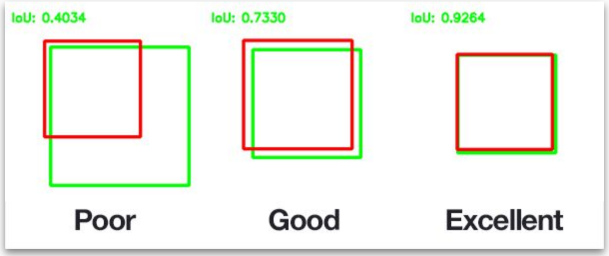
Results - Visual Comparison (The Bad & Ugly)



Results - mean Average Precision

- Jaccard Index (Intersection over Union)
- Box is considered True Positive if Jaccard Index > 0.5
- Average Precision AP = TP / (TP + FP)
- mAP is the mean of the AP of all classes
- In our case mAP = AP since we only consider one class (person)





Results - Comparison

mAP for our models:

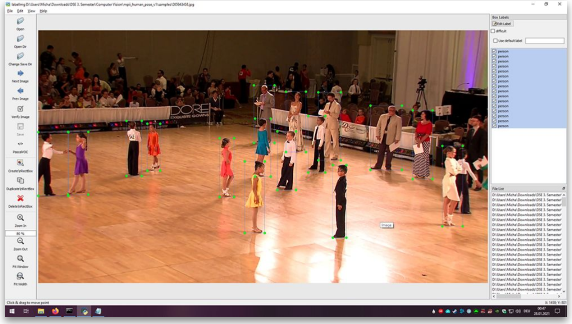
● EfficientDet:	54.55
● YOLO:	54.55
● HC:	03.26
● HC with data augmentation:	01.82

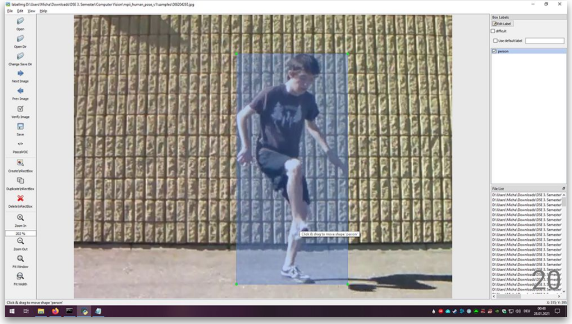
Discussion

- **Annotation Challenges**
- Bad Haar Cascade Results
- Comparing COCO mAP to MPII mAP

Some of the annotation challenges were:

- Do you classify mirror images as persons?
- Pictures are too blurred to properly draw bounding box.
- There are too many people.
- Are hair part of the body or not (e.g. in cases of special haircuts)?
- Does sport equipment belong to the body or not? E.g. a bicycle probably not but a baseball glove is debatable.





Discussion

- Annotation Challenges
- **Bad Haar Cascade Results**
- Comparing COCO mAP to MPII mAP

Possible explanations for bad Haar Cascade results:

- Too little negative images
- Negative images are not diverse enough
- Too little positive images

Possible explanations for even worse Haar Cascade results when using data augmentation:

- Augmentations influence the training negatively
- More data ≠ better data

Discussion

- Annotation Challenges
- Bad Haar Cascade Results
- **Comparing COCO mAP to MPII mAP**

Differences in mAP for Yolo and EfficientDet between datasets:

- YOLO COCO vs. YOLO MPII:
  55.3 54.5
- EfficientDet COCO vs. EfficientDet MPII:
  45.4 54.5

Possible reason why EfficientDet is significantly better on our data:

- Less complexity than in COCO e.g. we only predict persons.

Conclusion:

Existing models perform very well on new data and there is no necessity to re-train them from zero to get good predictions.

Furthermore computer vision is very dependant on the training data and manual annotation is a tedious and error prone task which elevates the usage of existing models even further.

Thanks!

Contact us:

Micha Birklbauer
Nicole Hölzl

https://github.com/t0xic-m
https://github.com/nhoelzl