REAL-TIME PPE DETECTION

Using the YOLOX Framework

Product owner
Orange: Jaroslaw.Legierski@orange.com
Authors: Eljasiak B., Jowik E., Kierznowski M., Wójcik M.

Project brief

Goal:

create and validate a model for objects **detection and classification** based on image analysis using the YOLOX framework.

Solution pillars:

1. Training data
2. Model providing probability predictions on multi-class labels

Detailed scope

Requirements for the model

- Object detection using YOLOX framework
- Supported object classes:
  - Person
  - Helmet
  - Head
  - Vest
- Emphasize on minimizing false positive detections, such as bald people or people wearing hats.
- Compatibility with YOLOX framework (final tests to be conducted using the Orange framework)
- Support for YOLOX "S" model

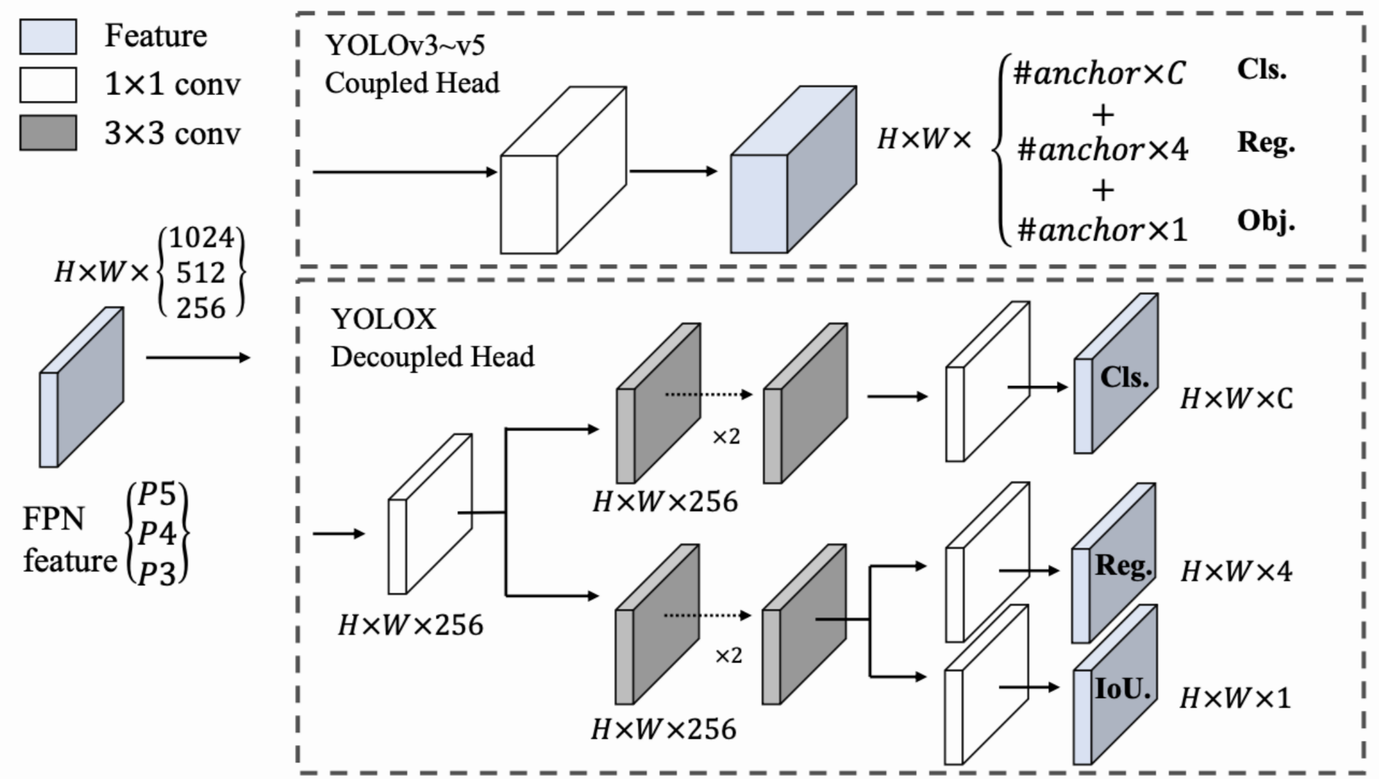
Deliverables

- Model formats:
  - ONNX
  - Native format for YOLOX (BIN and XML)
- Training and validation datasets
- Test results, including:
  - Recall (R), Precision (P),
    mean Average Precision (mAP)
    for each object class
  - Evaluation of algorithm performance

YOLOX

Concept

1. solution for real-time object detection tasks
2. builds upon the popular You Only Look Once (YOLO) family of models
3. extension of YOLOv3



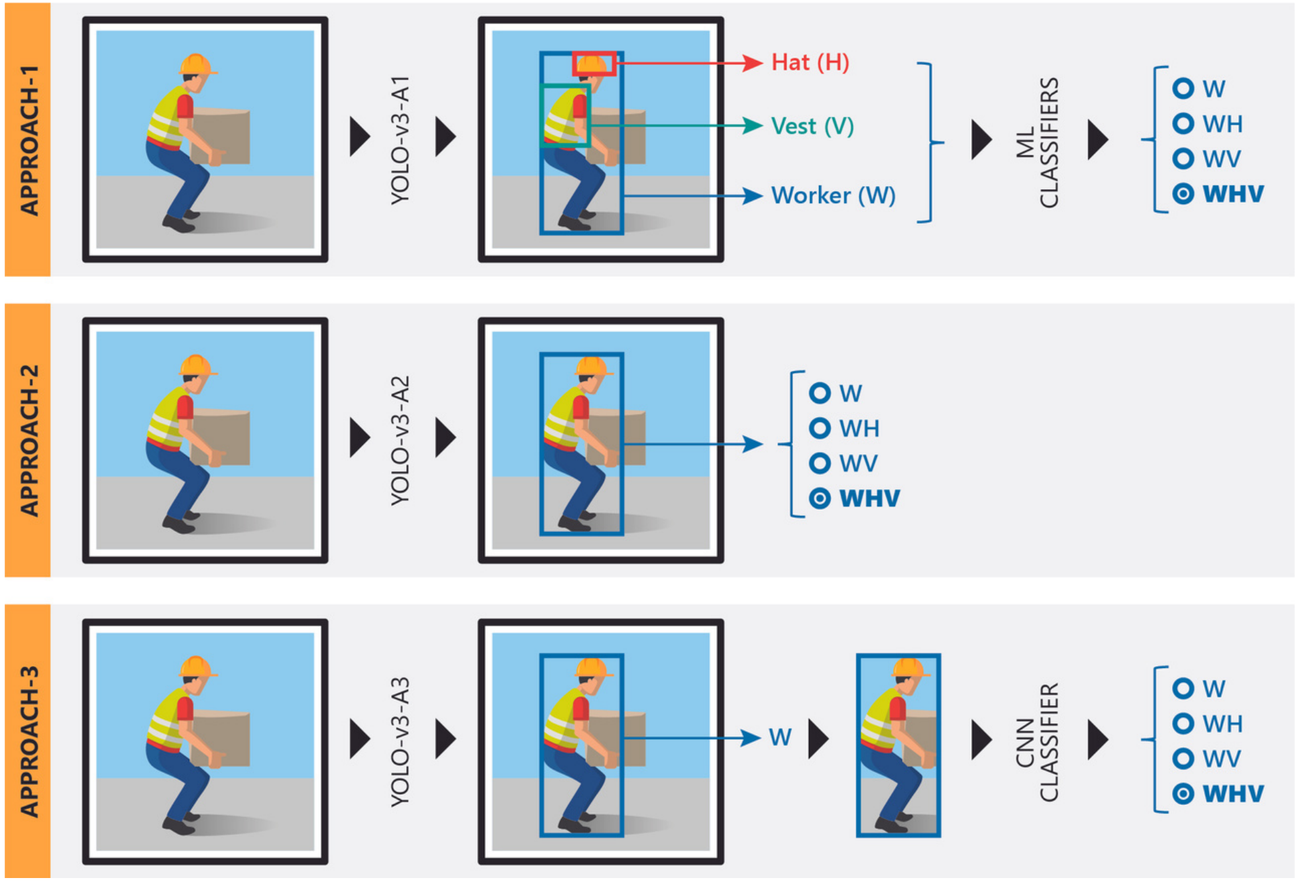
Implementation details

YOLOX is a single-stage object detector that modifies YOLOv3 with a DarkNet53 backbone. Specifically, YOLO's head is replaced with a decoupled one. For each level of FPN feature, it adopts a 1 × 1 conv layer to reduce the feature channel to 256 and then add two parallel branches with two 3 × 3 conv layers each for classification and regression tasks respectively.

Existing methods

Approach-1: YOLO-v3-A1 model detects object classes individually (here: worker, hat, and vest. Next, ML classifiers (Neural Network, Decision Tree) classify each worker as W (wearing no hat or vest), WH (wearing only hat), WV (wearing only vest), or WHV (wearing both hat and vest).

Approach-2: YOLO-v3-A2 model localizes workers in the input image and directly classifies each detected worker as:
W, WH, WV, or WHV.



Approach-3: YOLO-v3-A3 model first detects all workers in the input image and then, a CNN-based classifier model is applied to the cropped worker images to classify the detected worker.

Hard Hat Workers Dataset

- Object detection dataset of workers in workplace settings that require a hard hat.
- Annotations also include examples of just "person" and "head," for when an individual may be present without a hard hart.

Hard Hat Workers Object Detection Dataset

Download 7035 free images labeled with bounding boxes for object detection.

Roboflow

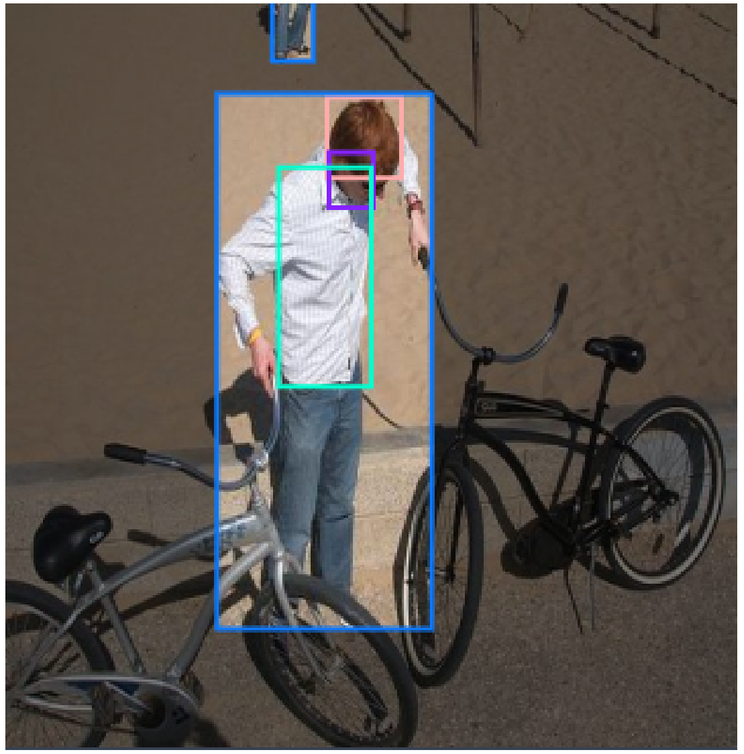
Supplementary Datasets

PPE v3 Computer Vision Project

Available labels

face nomask, face_wmask, goggles, hand noglove, hand_wglove, head_nohelmet, head_whelmet, person, vest

Construction PPEs Computer Vision Project



Available labels

Glove, Goggles, Hard Hat, Hardhat, Helmet, Mask, Person, Safety-belt, Safety-hat, Shoes, Vest, belt, board, chest, fire, glasses, gloves, head, headset, helmet, mask, no belt, person, safety-shoes, scaffold, smoke, vest, wood

Supplementary Datasets

ppe8 Computer Vision Project

Available labels

boots, face_mask, face_nomask, glasses, goggles, hand_glove, hand_noglove, head_helmet, head_nohelmet, person, shoes, vest

PPE - Combined Model Computer Vision Project

Available labels

Fall-Detected, Gloves, Goggles, Hardhat, Ladder, Mask, NO-Gloves, NO-Goggles, NO-Hardhat, NO-Mask, NO-Safety Vest, Person, Safety Cone, Safety Vest

Encountered issues

Some certain datasets lacked the comprehensive selection of labels, thereby yielding incomplete coverage across the entirety of the set. Consequently, the dataset transfers erroneous information to the model, leading to potential inaccuracies in its predictions.

Solution proposal

We used the available pre-trained architectures
to add the missing bounding boxes
of *person* and *head* class.

Concatenated dataset

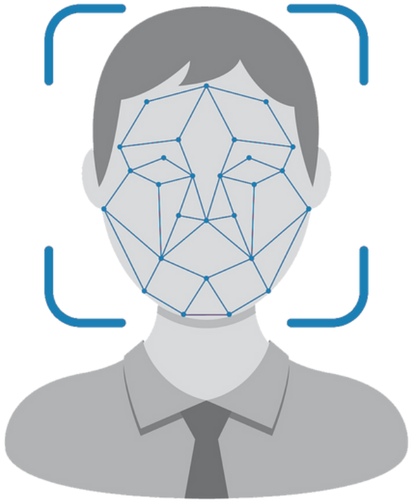
The whole dataset is depicted in 39800 images. It is divided into train and validation subsets in an 80:20 ratio.

Selected class	Number of occurences
Person	132548
Head	80377
Helmet	57754
Vest	29421

Models Used to Add Missing Annotations

YOLOX-l Base Model
Person class

YOLOX-s Model
Trained on Wider Face Dataset
No helmet class



YOLOX
Exceeding YOLO series in 2021

Technologies

YOLOX

We forked the official YOLOX repository to adjust it to our needs and prepare experiments for training.

PyTorch

The whole framework is compatible with PyTorch. It also provides some C++-compiled evaluators for better performance

Weights & Biases

We used it to log experiments and save artifacts in order to share results with other members in real-time.

pycocotools, CVAT, Python

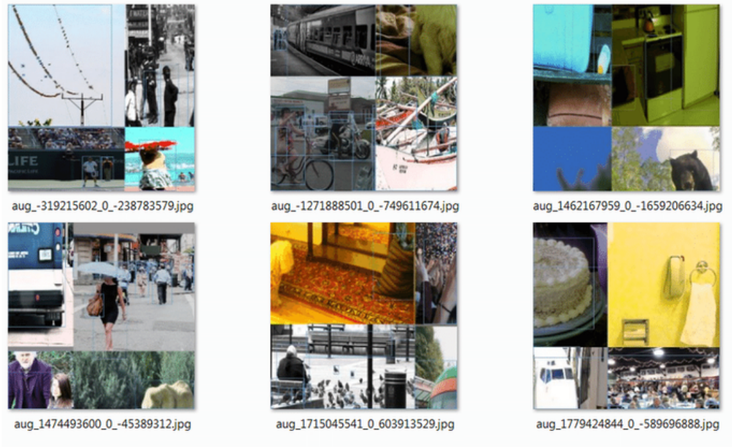
Tools necessary to work with MS-COCO data format. They allowed us to check, edit, and add annotations.

Data Augmentation

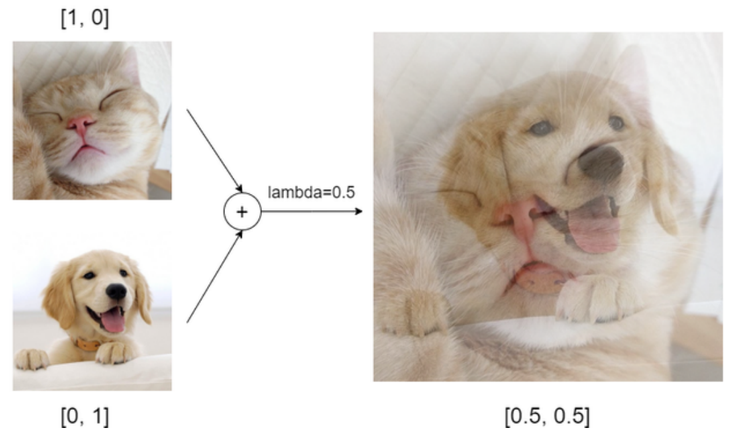
We tested wide variety of augmentation techniques

- Mosaic
- Mixup
- HSV change
- Fliping image
- Rotating image in 3 dimensions

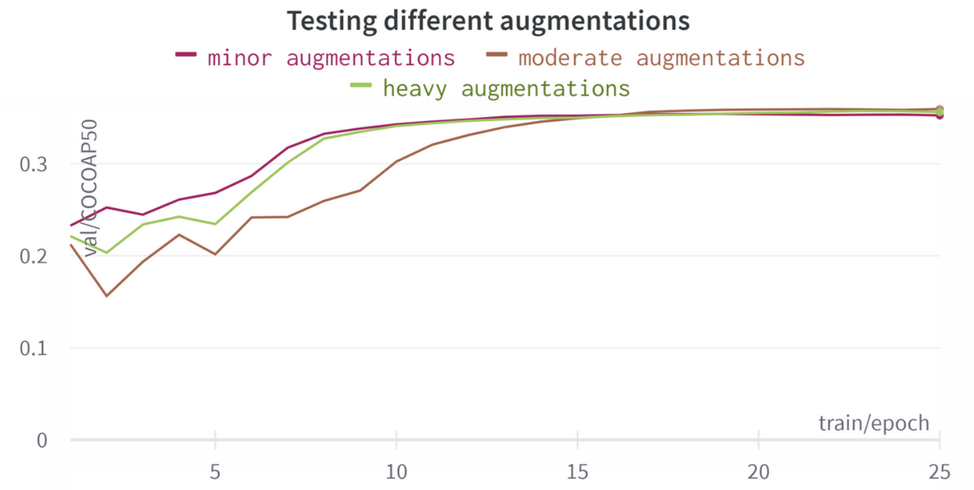
Mosaic



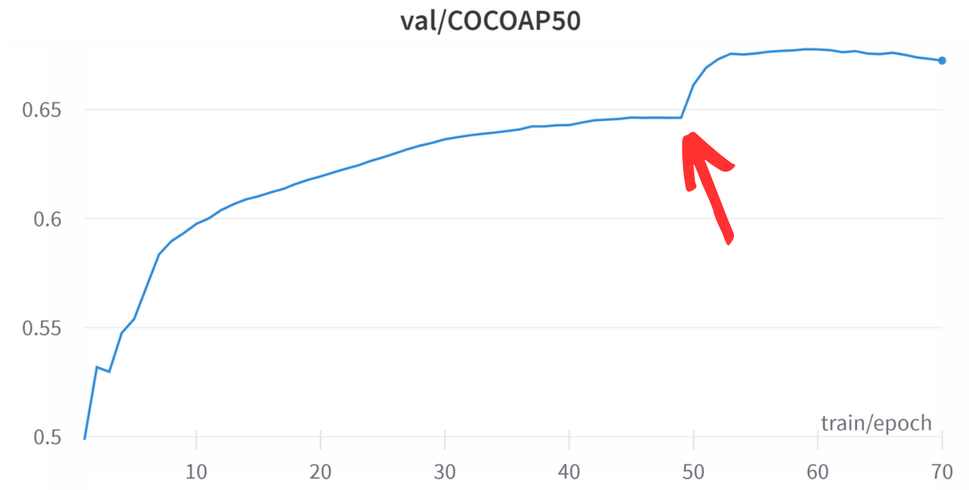
Mixup



Data Augmentation



testing different augmentation settings



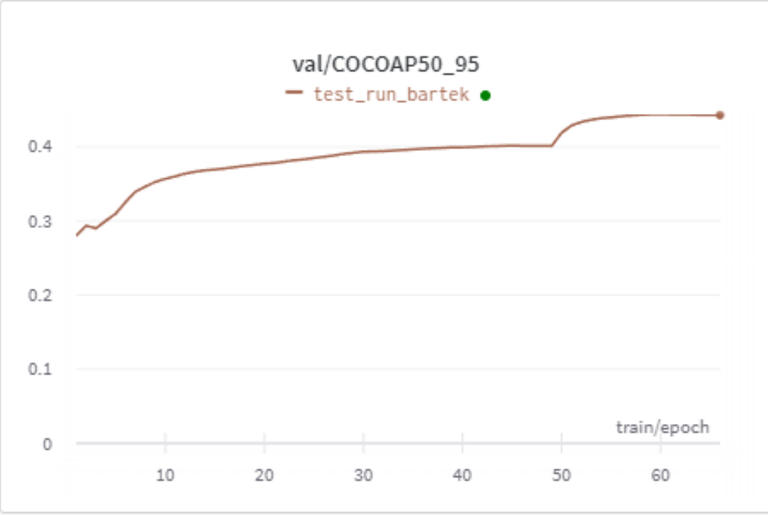
impact of augmentations visible during training

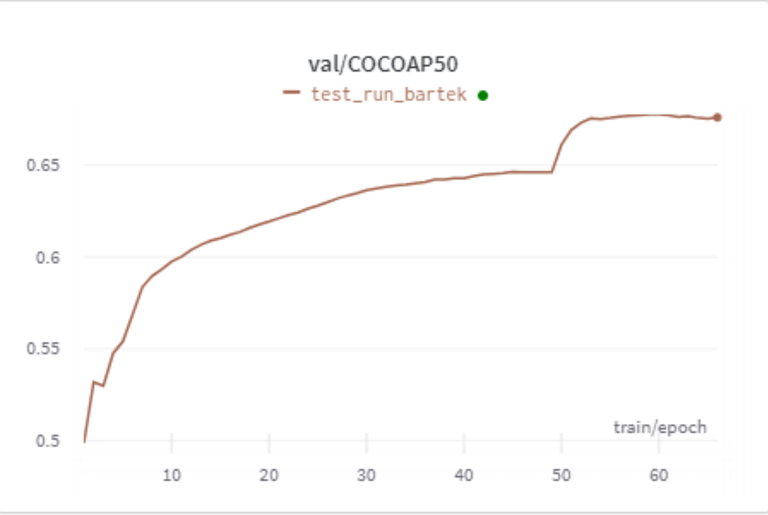
Evaluation

Scenario	COCO AP50	COCO AP50:95:0.5
1	0.3669	0.2295
2	0.5356	0.3456
3	**0.6777**	**0.4437**

We evaluated the models on a validation dataset sequentially in three different scenarios:

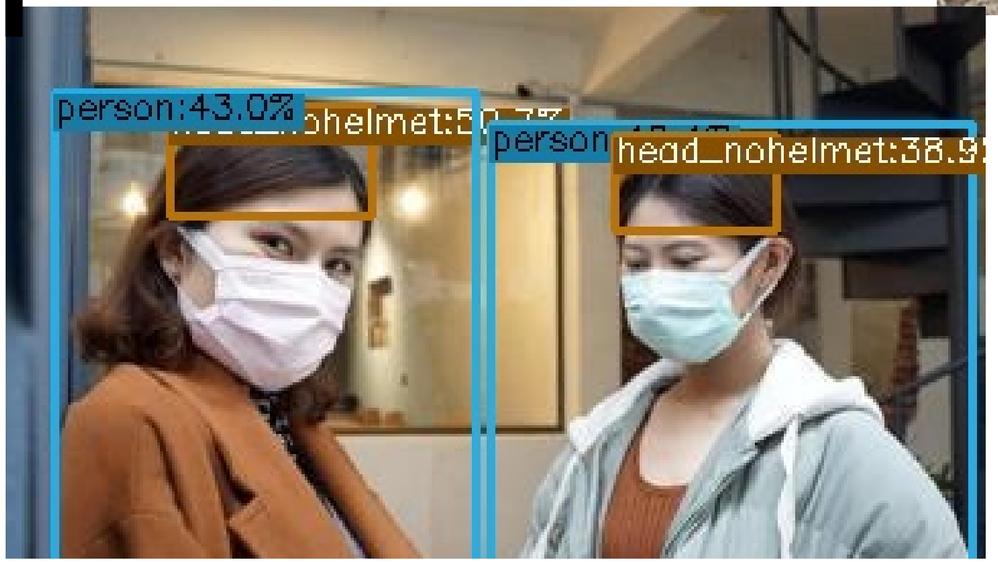
- **Scenario 1:** without adding any annotations
- **Scenario 2:** with *people* annotations added
- **Scenario 3:** with *people* and *no_helmet* annotations added

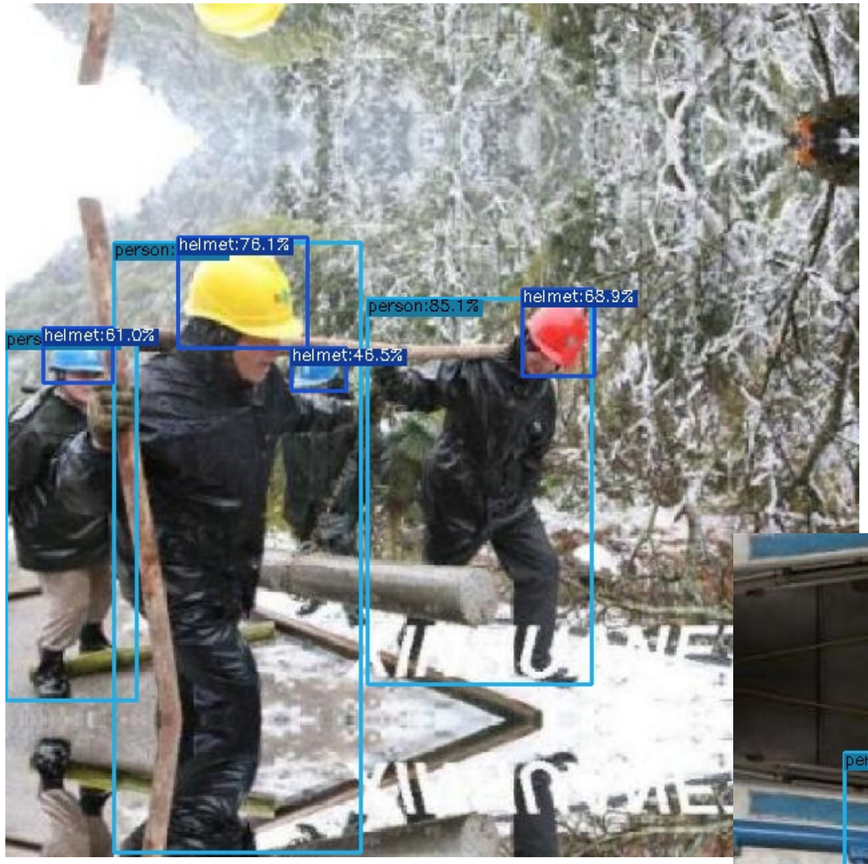




Scenario 1

Example inference



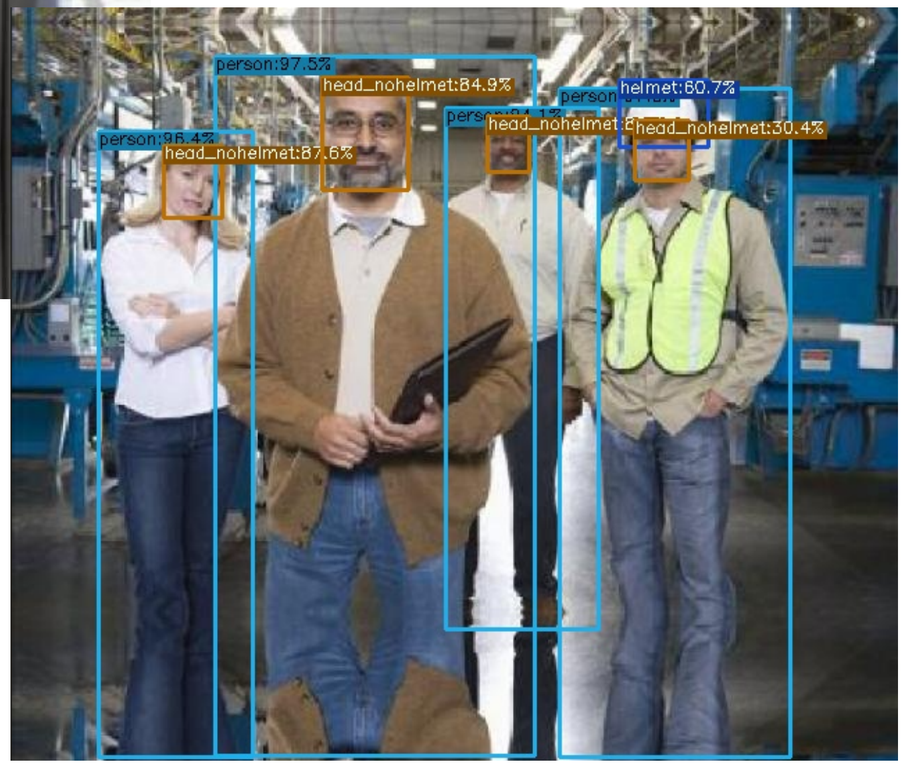


Scenario 2

Example inference

Scenario 3

Example inference



Summary

Final Conclusions

- Fine-tuning YOLOX is possible for large enough datasets
- Generating new annotations using pre-trained models may significantly improve the results due to the lack of negative cases
- The final model still underperforms in vest class
- YOLOX provides really fast inference (~ 40 FPS on GPU)
- There are a lot of hyperparameters and augmentation features provided

Bibliography

1. **Orange research work**: Towards Edge-Cloud Architectures for Personal Protective Equipment Detection
2. https://www.forecr.io/blogs/ai-algorithms/how-to-train-a-custom-object-detection-model-with-yolox
3. https://javamana.com/2022/03/20220317064015468l.html
4. https://yolox.readthedocs.io/en/latest/demo/onnx_readme.html
5. https://public.roboflow.com/object-detection/hard-hat-workers
6. https://github.com/ciber-lab/pictor-ppe#dataset
7. related works provided by the product owner: https://github.com/ejowik/2023L-data-science-workshop/blob/main/docs/related_works.md
8. https://github.com/VarCode-ai/yolox-face-detection