Persolo: A Pedestrian Is a Person Through Thick and Thin

Marawan Elbatel [1 2] **Hussein Maher** [1] **Mohamed Abouzeid** [3] **AbdElMoniem Bayoumi** [1]

Abstract

Pedestrian detection is crucial for trending computer vision applications, including pedestrian tracking and autonomous driving. Despite achieving high results on pedestrian detection benchmarks, the existing detectors fail to generalize on different datasets and unseen distributions, proving their impracticality in real-life applications. In this study, we modify YOLO-V4 to focus only on pedestrian detection, and name it "Persolo." We show that pre-training on a dense dataset, Open Images, increases the performance and generalization capabilities of the model as opposed to tailoring it towards specific benchmarks. We also experiment with two fine-tuning techniques to evaluate them on the MOT20 detection benchmark. The classical fine-tuned (FT) model, surpassing the state-of-the-art pedestrian detectors evaluated on the benchmark, shows contextual bias as opposed to the mixed-batch (MB) fine-tuned model. On the MOT20 detection benchmark, a mixed-batch (MB) fine-tuned Persolo outperforms all state-of-the-art pedestrian detectors from a run-time and average precision perspective with reduced data bias.

1. Introduction

Pedestrian detection is a bottleneck for many vision-based and robotics applications. For example, pedestrian detection is crucial for autonomous vehicles for safe navigation, besides pedestrian tracking in video surveillance cameras and action recognition systems (Yang et al., 2018; Wu et al., 2021; Girish et al., 2020). However, such real-time industry applications need a computationally efficient model with robust detection performance while being generic to deal with various scenarios other than those used for training. Accordingly, existing models, ranging from simple classic approaches to complex learning-based models, face problems with such a dilemma of contradicting objectives.

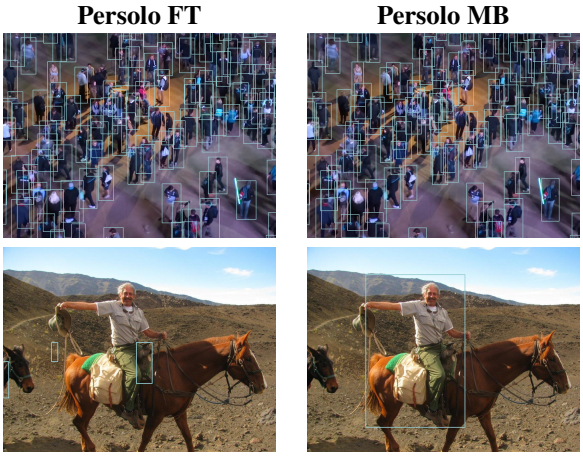


Figure 1. Examples of classical fine-tuned (FT) and mixed-batch (MB) fine-tuned Persolo on different images taken at a score threshold of 0.25.

Although the recent advancements in deep learning have led to significant accuracy improvements, such techniques suffer from high run-time complexity in crowded real-world scenarios. Furthermore, existing pedestrian detection models fail to generalize on other datasets generated from different distributions other than those used for training. Thus, other approaches consider using general-purpose object detectors to improve generalization despite trading off the accuracy. Hasan *et al.* (Hasan et al., 2021) show that training general object detection models on large and diverse datasets leads to better performance. Furthermore, they evaluate their hypothesis using the general-purpose detectors: Cascade R-CNN (Cai & Vasconcelos, 2021), Faster R-CNN (Ren et al., 2015), and embedded vision-based backbones such as MobileNet (Howard et al., 2017), and show that they outperform the state-of-the-art pedestrian-specific detection models. Unfortunately, such a paradigm does not fit real-world applications from the perspective of run-time complexity due to being computationally expensive.

This paper proposes modifying YOLO-V4 (Bochkovskiy et al., 2020) to focus on pedestrian detection instead of general-purpose detection and calls it "Persolo." YOLO-V4 is widely used in the industry since it is a computationally light general-purpose detection model that balances

[1]Department of Computer Engineering, Cairo University, Gamaa st, Giza, Egypt [2]Computer Vision and Robotics Institute, University of Girona, Girona, Spain [3]Dell Technologies, Cairo, Egypt. Correspondence to: Marawan Elbatel <marawan.mohamed98@eng-st.cu.edu.eg>.

detection accuracy and run-time speed while being generic. Thus, utilizing YOLO-V4 in pedestrian detection yields a model that operates in real-time while being robust and cable of dealing with scenarios from various distributions. Accordingly, as opposed to pre-training YOLO-V4 on MS COCO (Lin et al., 2014), we pre-train Persolo on the Open Images V6 dataset (Kuznetsova et al., 2020) which has five times the percentage of images containing people compared to MS COCO.

We evaluate our model on the MOT20Det (Dendorfer et al., 2020) benchmark as it contains a lot of diverse situations with dense crowds. As a result, our model significantly outperforms the state-of-the-art pedestrian detectors from a run-time and accuracy perspective. Furthermore, we evaluate the generalization ability of our model via cross-evaluating it on the Oxford TownCentre dataset (Benfold & Reid, 2011) compared to the state-of-the-art general object detectors that are widely used in the industry, showing our model's superior performance.

2. Related Works

Some of the works that use CNN-based models for pedestrian detection utilize R-CNN architecture for the detection task (Hosang et al., 2015; Girshick et al., 2014; Zhang et al., 2016). Zhang *et al.* (Zhang et al., 2020) introduce RPN+BF which uses Region Proposal Network (RPN) while using boosted forests (BF) as a way to improve performance. Even though RPN+BF achieves acceptable performance, it is not optimized end-to-end. Ren *et al.* (Ren et al., 2015) introduce Faster R-CNN as an improved model for object detection. It becomes one of the most used architectures for pedestrian detection in literature. Although Faster R-CNN is used in literature, its two-stage design makes it slower than the previous classical models, making its usage impractical in real-time applications.

On the MOT20 pedestrian detection benchmark, Wang *et al.* (Wang et al., 2021) propose a multipurpose model that achieves both joint object detection and multi-object tracking by using graph neural networks. Additionally, Ciampi *et al.* (Ciampi et al., 2020) introduce a model utilizing Faster R-CNN but with the additional usage of a synthetic dataset created from the photorealistic graphical engine of the video game GTA V (Grand Theft Auto V). Fabbri *et al.* (Fabbri et al., 2021) compare the results of Ciampi *et al.* with the same model, Faster R-CNN, but pre-trained on a large synthetic dataset, MOTSynth.

Advancing the state-of-the-art of pedestrian detection task, Hasan *et al.* (Hasan et al., 2021) evaluate existing state-of-the-art pedestrian detection models with cross-dataset evaluation. They prove that existing pedestrian detection models show data bias to specific benchmarks, failing to generalize when cross-evaluated. Furthermore, they show that training state-of-the-art general object detectors can significantly surpass pedestrian detection models. They utilized Cascade R-CNN (Cai & Vasconcelos, 2021) to support their hypothesis. Unfortunately, despite the improved detection accuracy, relying on a general-purpose classifier led to high computational complexity, making its usage impractical in real-time applications.

As opposed to the state-of-the-art, this paper introduces a real-time pedestrian detection model that surpasses the state-of-the-art detection accuracy and maintains a generic performance. We modify YOLO-V4 to amplify the person class bias and trained it with a dense training pipeline to be able to perform accurately on a pedestrian detection benchmark and cross-evaluated the model on another dataset to prove that it can be used in a real-time setting with cross-domain pedestrian datasets compared to the original YOLO-V4 that is widely used in the industry.

3. Persolo

We modify the general-purpose object detector YOLO-V4 to focus only on pedestrian detection in specific. Industrial applications tend to deploy YOLO-V4 due to its efficient inference time; thus, shifting its focus to pedestrian detection leads to a significant run-time and accuracy improvement compared to state-of-the-art pedestrian detection models. Accordingly, we modify YOLO-V4 to amplify the person class bias in the general-purpose object detector while generalizing on other datasets with cross-evaluation. The model architecture consists of a backbone, neck, and head as described in the original YOLO-V4 paper (Bochkovskiy et al., 2020). The model's detailed architecture can be found in Appendix A.2.

4. Experimental Settings

4.1. Datasets

We follow a dense and diverse training strategy for our pedestrian detection module; thus, we hydrate data that are diverse, large, and with a high rate of persons per image. A general object detector backbone is usually pre-trained on large scale datasets, ImageNet (Deng et al., 2009), MS COCO (Lin et al., 2014), or Open Images (Kuznetsova et al., 2020). Pedestrian detection benchmarks provide data that are either limited in size or with a low number of persons per image, making it hard to train the state-of-the-art general object detector to generalize on other datasets. Therefore, we find it more convenient to hydrate data from an open-source data resource that satisfies our needs. Compared to MS COCO dataset as shown in Fig. 2, a more complex dataset resource we take into consideration is Open Images V6, a dataset of 9 million varied images with rich annota-

tions (Kuznetsova et al., 2020).

The Open Images dataset contains approximately 16 million boxes for 600 classes annotated on 1.9M images, making it the largest existing dataset with object annotations. The dataset is divided into three splits, training, validation, and test. The person class contains 906,091 boxes in the Open Images training set with 861,892 confident bounding annotated on 212,668 images. A sample for a person in the Open Images and MS COCO dataset can be found in Appendix A.1. The Open Images test set consists of 14,433 images with 32,426 annotated bounding boxes representing pedestrians. We also use the MOT20Det (Dendorfer et al., 2020) dataset for experimenting with our model and the Town Center dataset (Benfold & Reid, 2011) for cross-dataset evaluation to have a fair comparison between our model and existing real-time models used in the industry.

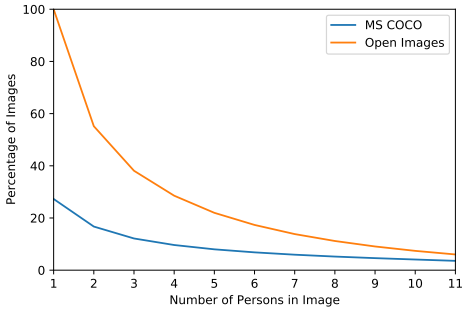


Figure 2. Percentage of images containing n persons in MS COCO and Open Images Dataset.

4.2. Implementation Details

4.2.1. Persolo: Training on Open Images

We train our YOLO-based model with input size (608×608) and only one output class on 212,668 images for 35,000 iterations, with a batch size of 64. Furthermore, we perform multiple data augmentation techniques on input images during training to reduce overfitting. We perform random cropping, horizontal flipping, and translation in both axes. The validation and test datasets consist of 4,774 and 14,433 images, respectively.

We load weights of the backbone pre-trained on MS COCO dataset (Lin et al., 2014). We do not freeze any layer to allow the gradient to propagate to the early layers. The learning rate is equal to 0.0013 with no learning rate scheduler. The optimization technique for backpropagation is Stochastic Gradient Descent (SGD) with a momentum of 0.949. We use the same loss functions described in the original YOLO-V4 paper (Bochkovskiy et al., 2020).

We use Google Colab for training and testing. The GPUs used for this experiment are Tesla K80 and Tesla T4. We simulate the testing and cross-evaluation with the YOLO Darknet learning repository by Alexey AB on GitHub (Bochkovskiy et al., 2020).

Finally, we serve the model and deploy it on a docker container with the following docker image specification, TensorFlow-GPU-2.3.0-rc0 and CUDA 10.1.2. The network can achieve 7.6 on Tesla K80, 16 fps on RTX 2060-Mobile, and 29.4 fps on a Tesla T4.

4.2.2. Fine-Tuning on MOT20 Detection Dataset

After training our model on a complex and dense pedestrian dataset, we utilize a training pipeline to fine-tune our model on another dense dataset, MOT20 (Dendorfer et al., 2020). MOT20 detection training dataset consists of 4 videos, totaling 8931 frames, in unconstrained environments with an average number of 149.7 pedestrians per frame. We use three video sequences in the training dataset, and one video for validation purposes. We fine-tune Persolo with two different techniques, Persolo FT and Persolo MB. Persolo fine-tuned (FT) is the classical fine-tuned model as shown in Fig. 3(a).

To increase the generalization capability of our model, we combine an equal number of images randomly chosen from the Open Images dataset and the MOT20 dataset in a mixed-batch fine-tuning as shown in Fig. 3(b). We name this mixed-batch fine-tuning experiment "Persolo MB."

We train each model for 7000 iterations with batch size equaling 64. The experiment hyperparameters are the same chosen in section 4.2.1.

5. Experimental Results

5.1. Testing

We describe in this section the testing and cross-dataset evaluation of our models, Persolo, Persolo FT, and Persolo MB. We use the average precision (AP) to evaluate our models, a Pascal VOC metric (Everingham et al., 2009).

Pre-training on the Open Images dataset provides a dense and diverse training pipeline for our model before participating in the MOT20 detection benchmark. After fine-tuning our model on MOT20Det, Persolo MB ranks as the 1st open-source model on the benchmark challenge according to the average precision metric. We show the recent results for the MOT20Det challenge in Table 1. Our model provides the optimal average precision and speed compared to other models. The precision-recall curve of our model downloaded from the challenge website can be found in Appendix. A.3.

In the industry, our model presents a robust real-time detection of pedestrians and succeeds in increasing the person

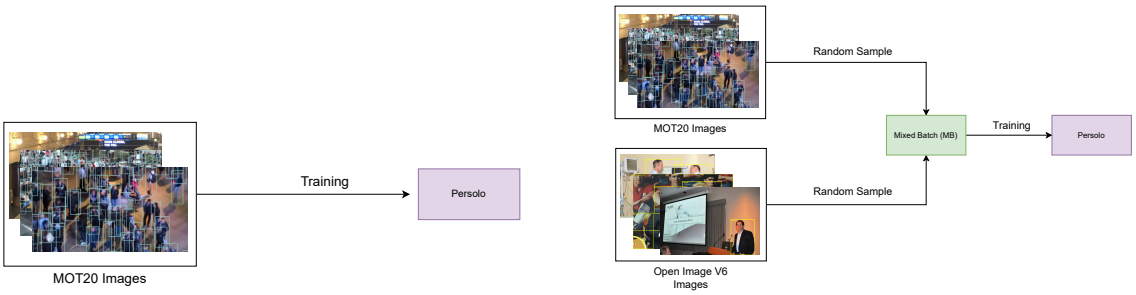


Figure 3. (a) Persolo fine-tuned (FT) model training. (b) Persolo mixed-batch (MB) model training.

Table 1. MOT20 Detection Benchmark

Model	AP↑	TP↑	FP↓	FN↓	Pr.↑	Rec.↑	FPS ↑
YOLO-V4 (Bochkovskiy et al., 2020)	0.56	210,598	88,516	132,926	70.41	61.31	28.9
Persolo **(ours)**	0.61	255,237	434,833	88,287	36.9	74.3	29.4
Synth-FRCNN (Fabbri et al., 2021)	0.62	206,902	28,202	136,622	88.0	60.2	-
Synth-FRCNN FT[1] (Fabbri et al., 2021)	0.72	241,056	**23,465**	102,468	**91.1**	70.2	-
ViPeD20 (Ciampi et al., 2020)	0.80	297,101	139,111	46,277	86.5	68.1	11.2
GNN-SDT (Wang et al., 2021)	0.81	304,236	31,677	39,288	90.6	88.6	1.2
Persolo FT[1] **(ours)**	0.87	314,578	181,416	28,946	63.4	91.57	29.4
Persolo MB[1] **(ours)**	**0.88**	**317,139**	185,547	**26,385**	63.1	**92.3**	**29.4**

[a]FT stands for fine-tuning on the MOT20 dataset, and MB stands for mixed-batch fine-tuning as described in section 4.2
The measurement of TP, FP, FN, recall, and precision is calculated based on the whole submission detections regardless of the predicted score threshold. That explains certain models have a high number of false positives.

class bias in the YOLO-V4 without degrading its performance when cross-evaluated on another dataset, Oxford Town Centre, that neither has seen. We show the precision-recall curve for the experiment in Fig. 4.

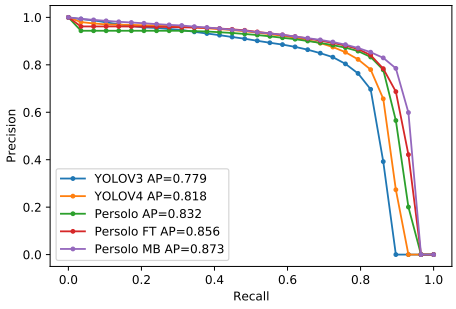


Figure 4. Precision-Recall Curve on Oxford Town Centre Dataset (cross-evaluation).

6. Conclusion

After observing the shift towards using a general object detector as the base for pedestrian detection models and the inspiring results of these models, we modify the YOLO-V4 with a dense training pipeline to focus on pedestrian detection instead of general-purpose detection. We propose a model that achieves optimal speed and accuracy with no specific data bias and can be used in real-time settings. We fine-tune our model with two different techniques on the MOT20 benchmark, and we conclude that mixed-batch fine-tuning (MB) helps our model to generalize best because it reduces the contextual bias of densely crowded datasets. Researchers are working to solve the robustness problem in the general object detection models by optimizing against adversarial attacks, pre-training on synthetic data, and moving the training towards a decentralized learning approach, federated learning, to collect more data that can help in the training pipeline without exposing data privacy. In the future, we will leverage the semi-supervised and weakly-supervised algorithms in our model to improve the performance and robustness of the pedestrian detection task, utilizing synthetic and weakly labeled datasets.

References

Benfold, B. and Reid, I. Stable multi-target tracking in real-time surveillance video. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3457–3464, 2011. doi: 10.1109/CVPR.2011.5995667.

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, 2016. doi: 10.1109/ICIP.2016.7533003.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. 4 2020. URL http://arxiv.org/abs/2004.10934.

Cai, Z. and Vasconcelos, N. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43:1483–1498, 5 2021. doi: 10.1109/TPAMI.2019.2956516.

Ciampi, L., Messina, N., Falchi, F., Gennaro, C., and Amato, G. Virtual to real adaptation of pedestrian detectors. *Sensors*, 20(18), 2020. ISSN 1424-8220. doi: 10.3390/s20185250. URL https://www.mdpi.com/1424-8220/20/18/5250.

Dendorfer, P., OsÌ†ep, A., Milan, A., Schindler, K., Cremers, D., Reid, I., Roth, S., and Leal-TaixÃ©, L. Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision 2020 129:4*, 129:845–881, 12 2020. ISSN 1573-1405. doi: 10.1007/S11263-020-01393-0. URL https://link.springer.com/article/10.1007/s11263-020-01393-0.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. Ieee, 2009.

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision 2009 88:2*, 88:303–338, 9 2009. ISSN 1573-1405. doi: 10.1007/S11263-009-0275-4. URL https://link.springer.com/article/10.1007/s11263-009-0275-4.

Fabbri, M., Brasó, G., Maugeri, G., Cetintas, O., Gasparini, R., Ošep, A., Calderara, S., Leal-Taixé, L., and Cucchiara, R. Motsynth: How can synthetic data help pedestrian detection and tracking? In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10829–10839, 2021. doi: 10.1109/ICCV48922.2021.01067.

Girish, D., Singh, V., and Ralescu, A. Understanding action recognition in still images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1523–1529, 2020. doi: 10.1109/CVPRW50498.2020.00193.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, 9 2014. doi: 10.1109/CVPR.2014.81.

Hasan, I., Liao, S., Li, J., Akram, S. U., and Shao, L. Generalizable pedestrian detection: The elephant in the room. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11323–11332, 2021. doi: 10.1109/CVPR46437.2021.01117.

Hosang, J., Omran, M., Benenson, R., and Schiele, B. Taking a deeper look at pedestrians. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4073–4082. IEEE Computer Society, 10 2015. doi: 10.1109/CVPR.2015.7299034.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL http://arxiv.org/abs/1704.04861.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., and Ferrari, V. The open images dataset v4. *International Journal of Computer Vision 2020 128:7*, 128:1956–1981, 3 2020. ISSN 1573-1405. doi: 10.1007/S11263-020-01316-Z. URL https://link.springer.com/article/10.1007/s11263-020-01316-z.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T. (eds.), *Proc. of the European Conf. Computer Vision (ECCV)*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015. URL https://github.com/.

Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., and Yeh, I.-H. Cspnet: A new backbone that can enhance learning capability of cnn. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1571–1580, 2020. doi: 10.1109/CVPRW50498.2020.00203.

Wang, Y., Kitani, K., and Weng, X. Joint Object Detection and Multi-Object Tracking with Graph Neural Networks. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Wu, J., Cao, J., Song, L., Wang, Y., Yang, M., and Yuan, J. Track to detect and segment: An online multi-object tracker. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12352–12361, June 2021.

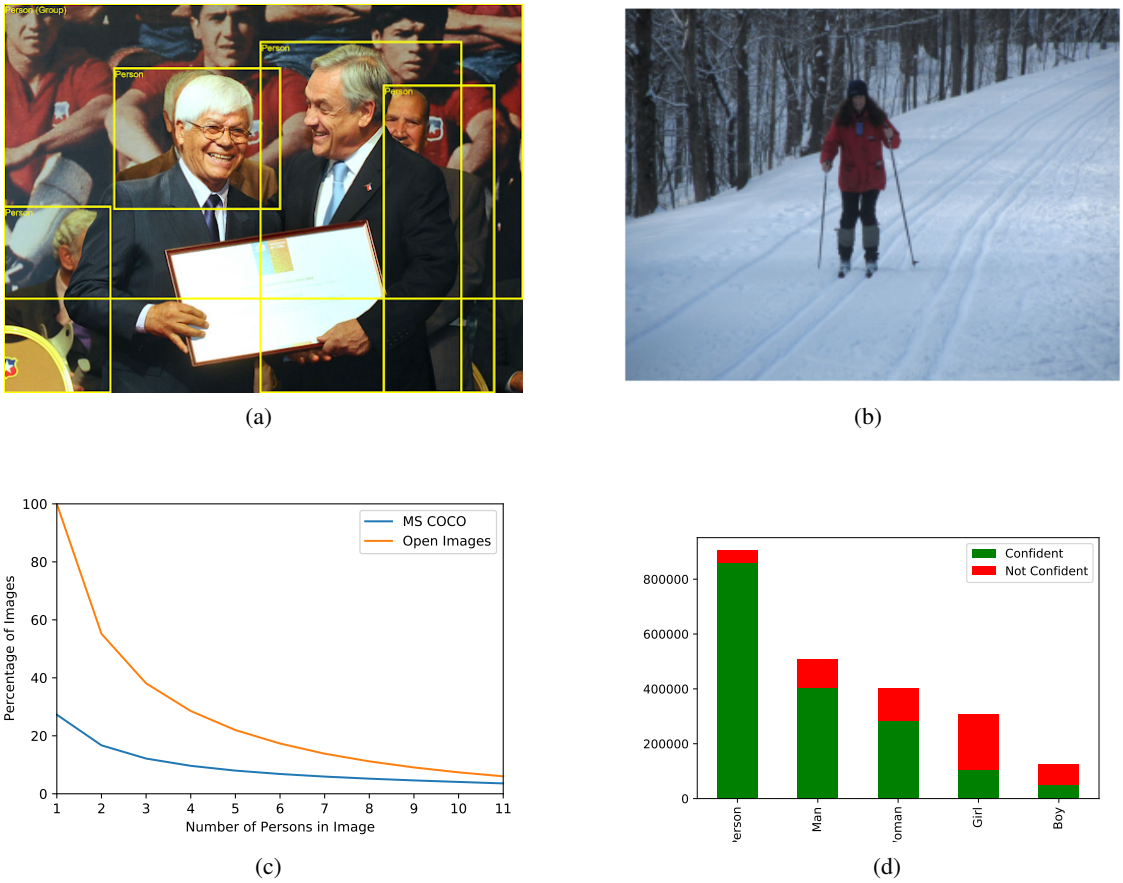
Yang, Z., Li, J., and Li, H. Real-time pedestrian and vehicle detection for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 179–184, 2018. doi: 10.1109/IVS.2018.8500642.

Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., and Zheng, N. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1109–1118, 2020. doi: 10.1109/CVPR42600.2020.00119.

Zhang, S., Benenson, R., Omran, M., Hosang, J., and Schiele, B. How far are we from solving pedestrian detection? In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1259–1267, 2016. doi: 10.1109/CVPR.2016.141.

A. Appendix

A.1. Datasets



(a) (b) (c) (d)

Figure 5. (a) A sample person image in Open Images dataset. (b) A sample person image in MS COCO dataset. (c) Percentage of images containing n persons in MS COCO and Open Images Dataset. (d) Classes of Interest in Open Images Dataset.

(a) (b) (c) (d)

Figure 6. (a) Original Sample Image. (b) Random Cropping. (c) Random Horizontal Flipping. (d) Random Translation.

A.2. Persolo Detailed Architecture

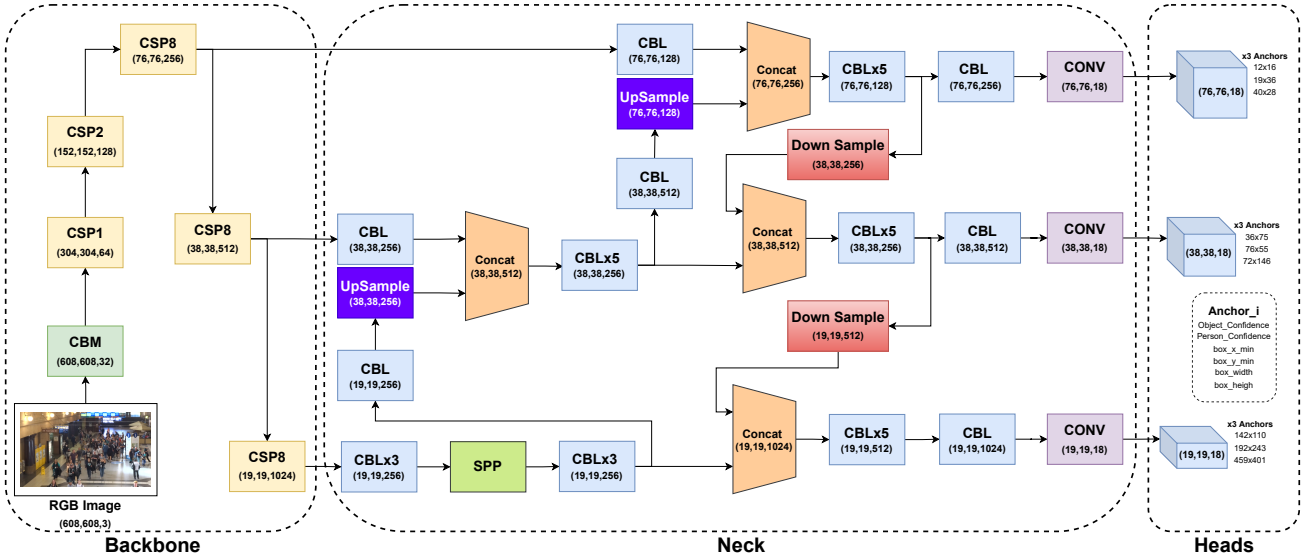


Figure 7. Model Abstract Architecture.

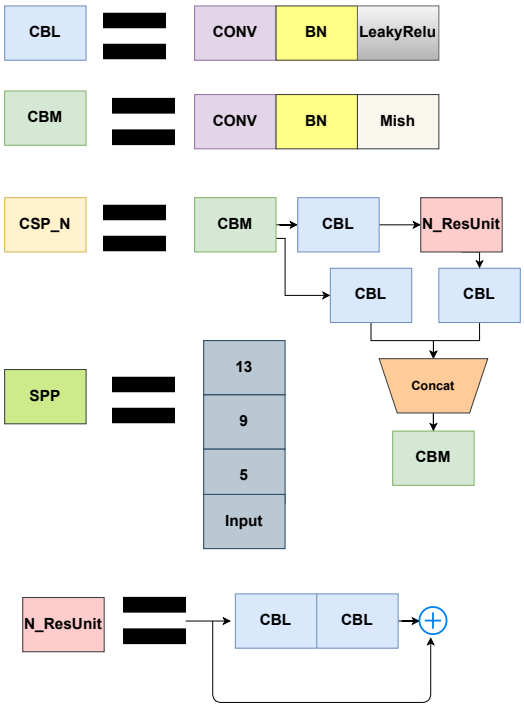


Figure 8. Helper blocks showing abbreviation for Persolo Architecture. BN abbreviates batch normalization.

A.2.1. Backbone

The backbone network is the state-of-the-art cross-stage-partial convolution network with darknet53 backbone (Wang et al., 2020). Cross-Stage-Partial Network (CSP) increases the receptive field by downsampling the input by factors of 8, 16, and 32 through 53 convolutional layers, where each scale follows a separate route.

A.2.2. NECK

The neck network consists of a path aggregation network that takes the generated routes from the backbone. In addition, we use an extra spatial pyramid pooling (SPP) block at the aggregation path of the third route generated from the backbone as it increases the receptive field with no reduction of the network speed.

A.2.3. HEAD (DETECTOR)

We use a one-stage detector head, YOLO, with output from three different routes with three different dimensions, 76×76×18, 38×38×18, and 19×19×18. Let the detector output dimension be donated as (S × S × f), where f is the number of filters, and S is the dimension along an axis. Each pixel is represented by f values. The number of filters, f, that represent the pixel is equal to:

$$f = (C + P_c + X_c + Y_c + RW + RH) \times number\ of\ anchors, \tag{1}$$

where C is the number of classes present in the system, one class in our case, P_c is the probability that there is an object in that pixel, X_c and Y_c are the X and Y coordinates of the center of bounding box relative to that grid cell. RW and RH are also the width and height of the bounding box relative to the grid cell. In our case, for the last three convolutional YOLO layers in the head, we choose filters equal to 18 to detect one class in our system, the person class.

A.3. Performance Metrics

We use average precision (AP) taken for different thresholds, precision, recall, F1-score, and speed in frame per second to assess our experiments. The average precision calculated (AP) is the precision for 11 equally spaced recall values [0, 0.1, 0.2, 0.3 . . . 0.9, 1.0] on the Precision-Recall curve for our person class, a Pascal VOC metric described in equations 2 and 3 (Everingham et al., 2009).

$$AP_11 = \frac{1}{11} \sum_{R \in [0, 0.1, \ldots, 1.0]} P_{intrep}(R) \tag{2}$$

$$P_{intrep} = max_{r':r'>r}(P(r')) \tag{3}$$

Table 2. Open Images Performance Comparison.

Model	AP	TP	FP	FN	Precision	Recall	F1-Score [1]
YOLO-V3	0.5601	23846	25622	8580	0.48	0.73	0.58
YOLO-V4	0.5870	**24905**	26740	**7521**	0.48	**0.77**	0.59
Persolo (ours)	**0.6408**	23824	**14120**	8602	**0.63**	0.73	**0.68**

[a]The TP, FP, and FN used for precision, recall, F1 score is taken at a score threshold of 0.25 and IOU threshold of 0.5.

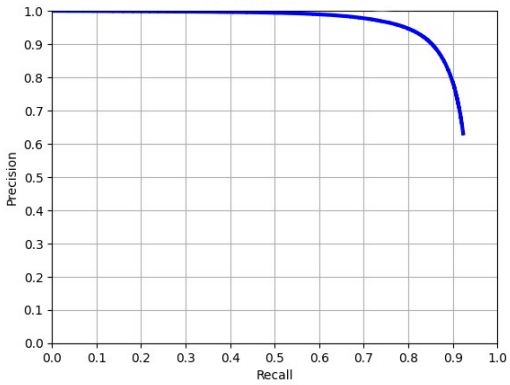


Figure 9. Precision Recall Curve on the MOT20 Detection Challenge.

Table 3. Oxford Town Centre Performance

Detection Model	AP
YOLO-V3	0.779
YOLO-V4	0.818
Persolo (ours)	0.832
Persolo FT (ours)	0.856
Persolo MB (ours)	**0.873**

A.4. Detection with Tracking

Figure 10. Examples of Persolo on different video sequences.

In this experiment, we test our detection model, Persolo MB, with a popular simple online detection-based tracking algorithm, SORT, that uses a control theory algorithm with Kalman Filter prediction to predict the trajectory path of each individual in the system, using state-space representation (Bewley et al., 2016). The experiment validates the efficiency of our detection model when used in a real-time tracking domain. We generate the detections for Persolo MB on the MOT20 test set. Then, we utilize the tracking algorithm SORT with our model. We execute evaluation on the benchmark server. The experiment shows that our model succeeded in raising the detection-based tracking algorithm performance as shown in Table 4 compared to the public detections provided by the benchmark.

Table 4. MOT20 Tracking Benchmark

Detection Model[1]	MOTA	MT	ML
Public Detections	$42.7_{\pm 18.6}$	208	326
Persolo MB (ours)	$\mathbf{59.5_{\pm 20.7}}$	**696**	**114**

[a]Detections are taken for a score threshold of 0.25 and IOU threshold of 0.5.