Powering up pedestrian segmentation with initial detection stage

Sharare Zolghadr
sharare.zolghadr@studenti.unipd.it

Donatas Vaiciukevičius
donatas.vaiciukevicius@studenti.unipd.it

Kateryna Skurativska
kateryna.skurativska@studenti.unipd.it

Abstract

In this work, we have experimented in the field of Semantic Segmentation. The idea is to compare the standard approach of semantic segmentation with a combination of object detection and segmentation. Our method works by feeding the input image into a fine-tuned object detection model, extracting object crops, and feeding them into a segmentation model, the outputs of which are stitched together to produce a full-resolution segmentation mask. We show that this approach improves the quality of the segmentation in some particular scenarios, but further experiments are required to fully confirm the advantages of this approach.

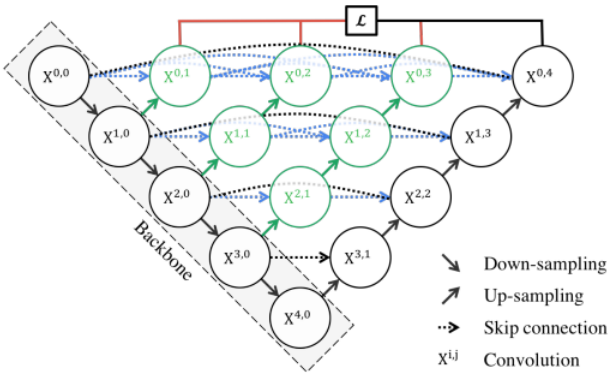

Figure 1. U-Net++ architecture

1. Introduction

In the dynamic realm of computer vision, the Pen-Fudan dataset stands as a crucial benchmark, challenging algorithms in crowded urban object detection. Our focus is on precise pedestrian detection in complex urban scenarios, addressing challenges like crowded scenes, occlusions, scale variations, and annotated masks. Overcoming these challenges is pivotal for practical applications such as urban planning, traffic management, and public safety, contributing to the reliability of computer vision systems. Our study presents a tailored approach to the Pen-Fudan dataset, aiming to advance current techniques and enhance system robustness in real-world environments.

2. Related Work

2.1. Faster R-CNN

Our pedestrian detection model is built upon the Faster R-CNN (8) architecture, renowned for its excellence in object detection. This architecture operates in two stages: an initial Region Proposal Network (RPN) suggests candidate regions, followed by a Region of Interest (RoI) pooling layer extracting features for final classification. Specifically configured components include an RPN optimized for detecting pedestrians and a RoI pooling layer, which we have fine-tuned to extract features crucial for accurate pedestrian classification.

2.2. U-Net and U-Net++

The semantic segmentation models we have based our combined approach on are U-Net (9), a model proposed back in 2015 for biological image segmentation and an evolution of it, U-Net++ (12). The U-Net architecture is based on two main parts - the contracting path downsamples the input images to capture the high-level features, while the expansive path slowly reconstructs the image into a segmentation mask using transposed convolutional layers. Skip connections are used to pass feature maps from the contracting path to the corresponding layers of the expansive path. In the end, a 1x1 convolutional layer is used to squeeze the final feature maps into logits for the desired number of classes.

U-Net++ mainly expands on this by introducing nested convolutional layers across skip connections, this way increasing model size and reducing the complexity of operations across layers. The main differences between U-Net and U-Net++ can be seen in Figure 1. The components in black are parts of both the original model and U-Net++, and elements in green, blue, and red are added in the latter. In

our scenario, we are using Unet++ with backbone layers from ResNet-34 (2) pre-trained on ImageNet (1).

3. Data

For this experiment, The Penn-Fudan Pedestrian Dataset (as used in (11)) is our preferred choice for research in object detection and pedestrian analysis. Developed by the University of Pennsylvania and Fudan University, this dataset offers unique features crucial to our study, including rich annotations, challenges in pedestrian occlusion and crowded scenes, diverse environments, varied perspectives, and real-world complexities. This dataset was chosen mainly because it is quite small, and given that we're working with limited resources (we have used Google Colaboratory Notebooks throughout this project), it speeds up the process of exploring different ideas.

4. Method

The primary problem addressed in this section is the accurate detection and segmentation of pedestrians in complex urban environments. Afterward, multiple segmentation models are tested for standard pedestrian segmentation tasks. Finally, we present a novel mixed detection-segmentation model to enhance pedestrian detection in the Penn-Fudan pedestrian dataset which serves as a benchmark for evaluating algorithms designed to address the challenges posed by object detection in crowded scenes and realistic scenarios.

4.1. Object detection

4.1.1 Model Selection

After evaluating various pre-trained models, we found Faster R-CNN to be optimal for Pen-Fudan pedestrian detection. We initialized it with pre-trained weights, fine-tuned the output layer for the dataset's unique characteristics, and enhanced generalization through diverse data augmentation techniques.

4.1.2 Choice of Backbone Model

The pedestrian detection model uses ResNet50 FPN V2 serves as the chosen backbone model, leveraging its ability to capture hierarchical features essential for detecting pedestrians. The Feature Pyramid Network (FPN) architecture aids in overcoming scale variations present in pedestrian instances within the dataset.

4.1.3 Modifications or Enhancements

1. Learning Rate Scheduling: Integrate dynamic learning rate scheduling mechanisms to adaptively fine-tune the learning rate throughout the training process. This implementation facilitates faster convergence and mitigates the risk of overshooting optimal parameter values

2. Post-Processing Techniques: Implement post-processing techniques like Non-Maximum Suppression(3) (NMS) were implemented to refine pedestrian detection results. This step contributed to the reduction of redundant bounding boxes by filtering out redundant bounding boxes with low scores, enhancing the accuracy of localized pedestrian instances.

3. Anchor Box Optimization: Experiment with anchor box sizes and aspect ratios. Customize anchor scales to better match the distribution of object sizes in your dataset

4. In addition, we employed the stochastic gradient descent (SGD) optimizer with specific hyperparameters: learning rate (lr) set to 0.005, momentum at 0.9, and weight decay of 0.0005. For the training phase, we opted to implement it from scratch following the guidelines outlined in the COCO engine training section. This decision resulted in faster training performance and ultimately yielded superior results compared to using the COCO engine.

4.1.4 Loss Function Used for Training

In particular, for the Faster R-CNN framework, the loss is composed of several components, and these components are calculated and combined into a final loss(the sum of all these individual losses). Here are the main components:

Region Proposal Network (RPN) Loss: The RPN is responsible for proposing candidate object regions. The loss associated with the RPN consists of two parts.

- Anchor Classification Loss (Binary Cross-Entropy Loss): Measures how well the RPN classifies anchors as foreground (containing an object) or background.

- Anchor Regression Loss (Smooth L1 Loss): Measures the difference between the predicted bounding box coordinates and the ground truth coordinates for positive (foreground) anchors.

Region-based ROI Heads Loss: After the RPN proposes candidate regions, the RoI (Region of Interest) heads are responsible for refining these proposals and predicting the class of the objects within them. The loss components include:

- Classification Loss (Cross-Entropy Loss): Measures how well the model classifies the objects within the proposed RoIs.

- Regression Loss (Smooth L1 Loss): Measures the difference between the predicted bounding box coordinates and the ground truth coordinates for the objects within the proposed RoIs.

$$\text{SmoothL1Loss}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (1)$$

4.2. Semantic segmentation

Before the segmentation training could start, the dataset had to be preprocessed. The images varied in size, therefore zero-padding in dimensions of the maximum size of the image was applied to all of the images. This resulted in a consistent image dimensions of 608*1024. As the ResNet model used for the Unet++ backbone was pre-trained on ImageNet, we have normalized the images to match its mean and standard deviation. Random cropping augmentation was used to reduce the GPU memory usage and increase training speed. Crops of size 576x576 were generated, which both matched the input size to the U-Net in the original paper, as well as fit the model's requirement of input dimensions being divisible by 32.

We did not perform fine-tuning for the models, as after initialization, only model encoder weights were pretrained, while the other layers were randomly initialized and had to be trained either way. The model was trained using Adam (4) optimizer with the initial learning rate of 0.0005. and Focal loss (6) as the loss function. The idea behind it is that it increases the loss for harder-to-classify classes and in this way tries to compensate for the class imbalance in the data (for example if a pedestrian takes up only a small part of the image).

$$\text{Focal Loss} = -\alpha_t (1 - p_t)^\gamma \log(p_t) \quad (2)$$

The strategy of reducing the learning rate based on a validation metric was used with the patience of 10 epochs and a reduction factor of 0.1. The model would usually train for around $70 - 90$ epochs - training would end due to early stopping (with the patience factor of 20 epochs).

4.3. Combined segmentation

The idea of combined segmentation required us to train two models on separate tasks. First, a pedestrian detection model had to be trained to produce bounding boxes against the regions of interest in the picture. The second model was to be trained on crops of the original image to simulate the detected regions.

4.3.1 Inference pipeline

We have defined an inference pipeline that starts with an object detection model that is fine-tuned to detect pedestrians. The detected boxes are used to make crops of the original image - each crop is expanded in width and height by a factor of two, then upsized by padding dimensions up to values that are divisible by 32 (as we're still using U-Net++ as a segmentation model). The upsizing is done to compensate for inaccuracies of the detection model where a predicted bounding box does not cover all parts of the pedestrian. These crops are then fed through a segmentation model. Then, an empty mask is initialized for the entire image, and unpadded crop masks are stitched into it using max blending.

The detection model was trained as described in 4.1. The training data for the segmentation model was created by taking each image of the dataset with its corresponding bounding boxes, expanding these boxes by a factor of two, and upsizing to fit the dimension requirements of the U-Net++. In this training, as before with the full-size images, they are normalized to match ImageNet. For consistent batch sizes, these crops then are randomly cropped again to the dimensions of 288x288 during training. The loss function, optimizer, and learning rate scheduler were kept as described in 4.2 to keep the model training consistent and evaluate the advantage of segmenting crops.

5. Evaluation Metrics

5.1. Object detection

To assess the performance of our object detection model, we employed a comprehensive set of evaluation metrics, according to the standards established by the MS COCO Dataset(7). Our evaluation metrics are IoU (Intersection over Union), Average Precision, Average Recall, and the Precision-Recall curve. Additionally, we measure the centeredness and IoU of our bounding boxes against their classification scores.

IoU is the overlap (intersection) of the predicted and ground truth bounding boxes divided by the union of the predicted and ground truth boxes. If the IoU surpasses 0.5(minimum IoU threshold), it indicates a successful detection (true positive).

mAP, or mean Average Precision, refers to the average precision calculated across various IoU values, typically ranging from 0.50 to 0.95. This principle holds similarly for Average Recall as well.

Center-ness is computed to assess the precision of our predicted bounding box locations using the below formula following the measurement of Center-ness for FCOS(10)(Fully Convolutional One-Stage Object Detection):

$$\text{Center-ness*} = \sqrt{\frac{\min(l,r)}{\max(l,r)} \times \frac{\min(t,b)}{\max(t,b)}} \quad (3)$$

Where (x, y) is the center of the predicted bounding box, $l = x - x_1$ (left ground truth coordinate), $r = x_2 - x$ (right

ground truth coordinate), $b = y - y_1$ (bottom ground truth coordinate), and $t = y_2 - y$ (top ground truth coordinate).

5.2. Semantic segmentation

The two main metrics used for the evaluation of the semantic segmentation were IoU (it measures how well the predicted segmentation aligns with the actual ground truth boundaries of the object.) and F1 Score. Compared to IoU, F1 Score focuses more on predicting the correct class on a pixel-by-pixel basis, rather than the general shape of the prediction. For the span of this project, we have evaluated both of these metrics for the segmentation performance.

6. Results and Improvements

6.1. Pedestrian Detection

In our comprehensive experiment, we systematically explored a range of hyperparameters to optimize our object detection model. Various architectures, including MobileNet, SSD, FCOS, and more, were considered, each configured with diverse backbones and experimented with fine-tuning or transfer learning. Through meticulous evaluation, involving scrutiny of IoU plots, classification scores, precision-recall curves (see fig 2), and Mean Average Precision (mAP) (see table 6.1) on the test set, we identified a standout model that consistently outperformed its counterparts.

This selected model showcased exceptional performance, attaining the highest area under the precision-recall curve and mean average accuracy. Notably, it demonstrated remarkable proficiency in detecting small objects. Additionally, it achieved top-tier scores in classification accuracy and IoU, affirming its efficacy for pedestrian detection tasks in intricate and challenging scenarios.

6.1.1 Failure Cases Analysis in Pedestrian Detection

Despite the overall commendable performance of our pedestrian detection model on the Pen-Fudan dataset, it is crucial to delve into instances where the model faces challenges and exhibits suboptimal results. The most frequent failures arise in the following scenarios:

Single Detection Recall at Low IoU (maxDets=1) One notable area of concern arises in scenarios where the model is constrained to output only one detection per object suggests a challenge in accurately localizing objects when limited to a single detection.

Small Objects Small objects typically exhibit fewer distinguishable features, making precise detection challenging.

6.1.2 Improvements

To enhance the model's performance on small objects, the following strategies were implemented: Feature Extraction

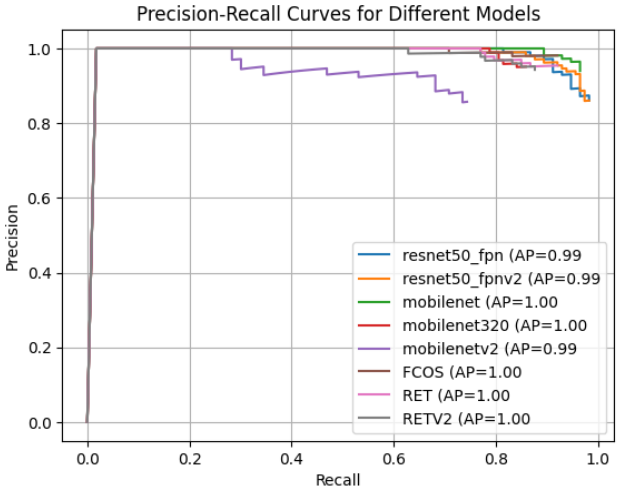

Figure 2. Precision-recall curves comparison among different object models

Enhancements (utilizing ResNet FPN V2 (5) as our backbone to provide valuable features), Augmentation Techniques, and anchor size optimization. This allowed us to achieve an Average Precision (AP) at IoU threshold [0.50:0.95] with maxDets=100 for small objects, recorded at 0.750. This metric showcases the model's effectiveness in accurately detecting and localizing pedestrians, particularly in scenarios where objects are relatively small within the image.

For single-object detection, Post-Processing Refinement, such as non-maximum suppression (NMS)(see fig 5), was applied to address overlapping or redundant detections that could contribute to decreased recall. Additionally, IoU Threshold Optimization was performed to explore the impact of different IoU thresholds on recall and assess the trade-off with precision. The outcomes of the enhanced model are presented in the table 6.1.2

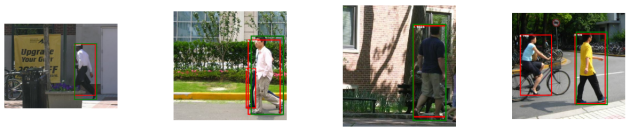

Figure 3. The detections with NMS

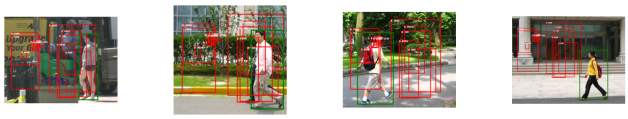

Figure 4. The detections without NMS

Figure 5. Selecting the Right Bounding Box Using Non-Max Suppression

Model Name	Backbone Name	AP@[0.50]	AP@[0.75]	AP@[small]	AP@[medium]	AP@[large]
FasterRCNN Model 1	ResNet50 FPN	0.990	0.923	0.525	0.717	0.799
FasterRCNN Model 2	ResNet50 FPN V2	**0.991**	**0.943**	**0.75**	**0.767**	**0.834**
FasterRCNN Model 3	MobileNet V3 Large FPN	0.975	0.906	0.000	0.648	0.781
FasterRCNN Model 4	MobileNet V3 Large 320 FPN	0.904	0.678	0.000	0.227	0.604
FCOS Model 5	ResNet50 FPN	0.980	0.927	0.501	0.591	0.749
RetinaNet Model 6	ResNet50 FPN	0.984	0.883	0.402	0.662	0.708
RetinaNet Model 7	ResNet50 FPN V2	0.974	0.859	0.701	0.669	0.738
FasterRCNN Model 8	MobileNet V2 Large	0.794	0.219	0.000	0.103	0.412

Table 1. Precision Metrics of Object Detection Models on Pen-Fudan Dataset

Model	AP@[IoU=0.75]	AP@small
Simple Faster R-CNN	0.643	-1.000
Improved Model	**0.943**	**0.75**

Table 2. Comparison of Simple Faster R-CNN and Improved Model

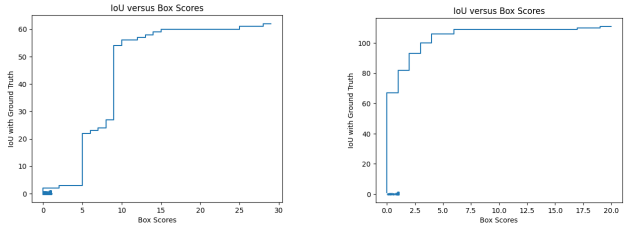

Figure 6. The intersection over unioun against classification score before and after applying improvements

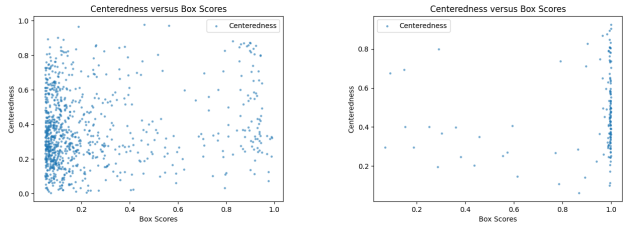

Figure 7. The Center-ness against classification score before and after applying improvements

6.2. Semantic segmentation

In total, 4 models were tested for semantic segmentation on full images - U-Net and U-Net++ with ResNet-18 and ResNet-34 backbones. The results (table 3) were a bit surprising at first sight - the most simple, smallest model, performed the best. But upon closer inspection, it became clear that due to the size and complexity of the other models, they overfit the training data more - U-Net with ResNet-18 backbone had the lowest metric deltas between training and test sets.

After these results, we decided to train different versions of this U-Net - without pre-trained weights and twice with pre-trained weights, freezing the backbone layers one of the times. The idea was to test how much of a difference pre-trained weights bring to the model. The results can be seen in the Table 6.2. This experiment did not show as many surprises as the initial training and the results were quite clear - training the entire model brought the best results, and using pre-trained weights did show an improvement. Freezing the backbone reduces the performance of the trained model slightly, but the margin is tiny to the point where the difference is negligible.

Looking at some segmentation examples makes the picture clearer - while there are artifacts in the segmentation, the ground truth masks have issues that weren't initially obvious when looking into the dataset. Example a) of Figure 8 shows a relatively nice segmentation result, however, the model is penalized for not segmenting an empty area between the person's legs - the ground truth mask is not precise enough and covers the general outline of the pedestrian. In case b) we can see a side effect of these imperfect masks - the model was exposed to similar imprecise masks during training and it made it struggle more at generalizing person-defining features - therefore the tree-supporting structure that could roughly resemble the legs of a human being get segmented as well. The worst-case scenario for evaluation is example c) - In this particular case, the input is an image of a crowd, however of of 15 people in the image, the ground truth mask only contains 3. This pretty much tells us that metrics do not show the full picture of the model's performance, although they do give a strong suggestion (as the less performant models did have more artifacts like example b).

Regardless, we moved on to segmentation on pedestrian crops, hoping to reduce the number of artifacts present in the predicted mask by localizing the segmentation targets.

6.3. Semantic segmentation on crops

The approach for the segmentation of crops was analogous to full images - we trained the same 4 models on the crops made from the original training data subset. In this scenario U-Net++ with a ResNet-18 backbone outperformed the rest, but again, by quite a low margin. The results can be seen in Table 6.3.

When taking a look at the generated predictions (Figure 9) from example a) we can see that localizing the image targets can produce very good segmentation masks, however, the model tends to over-segment and some small artifacts still show up. Example b) highlights how poor ground-

Model	Backbone	IoU	F1 Score	Train-Test IoU Delta	Train-Test F1 Delta
U-Net	ResNet-18	**0.608**	**0.748**	0.267	0.180
U-Net	ResNet-34	0.550	0.702	0.269	0.188
U-Net++	ResNet-18	0.588	0.733	0.282	0.192
U-Net++	ResNet-34	0.584	0.729	0.272	0.188

Table 3. Semantic segmentation on full-size images, test set

Condition	IoU	F1 Score
Random weights	0.559	0.711
ImageNet weights	**0.608**	**0.748**
ImageNet weights + frozen encoder	0.601	0.746

Table 4. U-Net performance depending on encoder weight initialization

Figure 8. Full-size image segmentation examples

Model	Backbone	IoU	F1 Score
U-Net	ResNet-18	0.626	0.745
U-Net	ResNet-34	0.607	0.731
U-Net++	ResNet-18	**0.637**	**0.756**
U-Net++	ResNet-34	0.631	0.751

Table 5. Semantic segmentation on cropped images, test set

truth masks misrepresent the performance of the model in the metrics.

6.4. Pipeline evaluation

As we designed the pipeline to match its interface with existing segmentation models, the evaluation process remained consistent with previous semantic segmentation approaches. Our pipeline achieved an average IoU of 0.629 and an F1 score of 0.765 on the test set. The improvement was marginal but expected when taking into consideration segmentation performance on image crops. Visually inspecting the predictions has confirmed that dividing

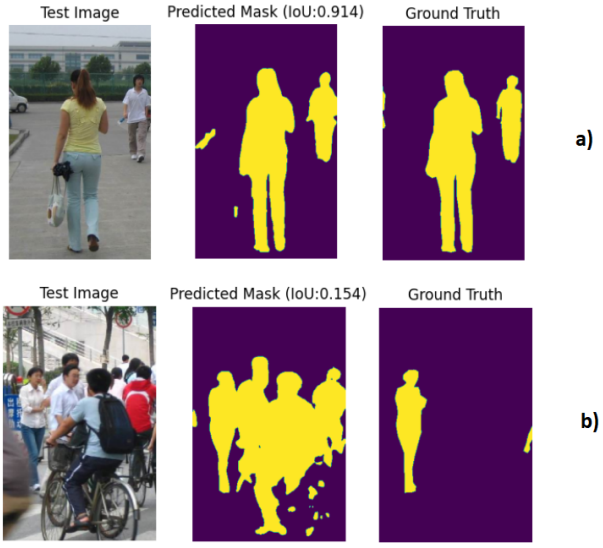

Figure 9. Cropped image segmentation examples

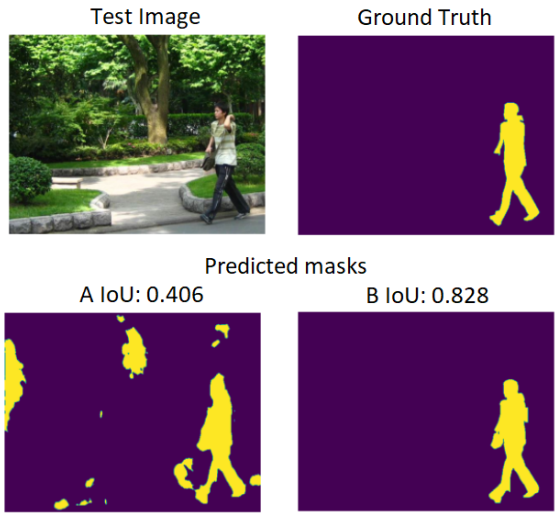

Figure 10. Segmentation comparison of standard semantic segmentation with U-Net (A) and pipeline (B).

the image into smaller crops to segment did not eliminate all the segmentation artifacts. However, it did help in some particular cases, an example we can see in Figure 10. The pedestrian detection model correctly identified the region

of interest in the image, therefore segmentation model only ran on the pedestrian and did not generate additional artefacts. That same segmentation model did show artifacts on tree branches in prior testing, but having a strong detection model helped to avoid them in the final segmentation.

In general, it has to be noted that this kind of pipeline is generally a less efficient approach - to begin with, inference takes longer because we're running two models at least once. Secondly, the number of times the segmentation model has to be run directly depends on the amount of crops generated by the detection model. In a perfect scenario, if there's only one detection box per pedestrian, therefore inference only runs on the regions of interest once and the approach has promise to be faster, given that the detection model is lightweight enough. However in most cases, there are multiple boxes generated per pedestrian, and that, added together with false predictions that have no pedestrians in them, makes the entire approach slower than the segmentation equivalent.

7. Conclusion

In this study, the semantic segmentation study revealed that in the case of the Penn-Fudan dataset, where the quality of provided ground truth segmentation masks is subpar and the dataset is quite small, smaller U-Net Models performed the best (although, based on the quantitative metrics, marginally). Meanwhile, our segmentation pipeline powered by an object detection stage achieved slightly better quantitative results by focusing the segmentation on image regions containing pedestrians and therefore reducing the amount of generated false positives. However, that was done with a performance penalty. To better judge the performance of such architecture, another dataset should be used for more reliable quantitative results. While celebrating these accomplishments, we recognize the potential for future research to explore additional optimization avenues and diverse datasets. This study provides valuable insights for achieving more efficient pedestrian detection in real-world scenarios.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515, 2017.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021.

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[10] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[11] Liming Wang, Jianbo Shi, Gang Song, and I-fan Shen. Object detection combining recognition and segmentation. In *Asian conference on computer vision*, pages 189–199. Springer, 2007.

[12] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018.