भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Indian Institute of Technology Hyderabad

Semi Supervised Learning for Better Pedestrian Detection

SHORYA SHARMA
School of Electrical Sciences, Indian Institute of Technology Bhubaneswar
ss118@iitbbs.ac.in

Supervisor: **PROF. Dr. KRISHNA MOHAN**
Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad

July 2021

Abstract

Deep neural networks often require large amounts of labeled training data in order to successfully train the model. However, collecting large amounts of labeled data is both costly and time-consuming, while only a subset of it can be labeled by humans due to the effort needed for high quality annotation. In order to reduce this bottleneck, many studies utilize the technique of semi-supervised learning. Active learning is one such technique that can make the process of labeling data more efficient by selecting the smallest possible training set that can improve the model the most. Self-training is another such technique that reduces the labeling effort that enables the model to be trained on its own predictions. In this paper, we combine a novel method of different semi-supervised learning methods for pedestrian detection tasks with transfer learning in order to reduce the labeling effort and better utilize the availability of a large number of labels from open-source dataset. Link to the codebase: GitHub

1 Introduction

Deep neural networks have made significant advances in image classification and object detection tasks. Training deep convolutional neural networks (Krizhevsky et al. [2012]) with a large amount oflabeled data enables a successful classification in various domains. Pedestrian detection is a task in the field of computer vision that has applications in different domains such as autonomous driving, robotics, and person identification. A performance of Pedestrian detection is important because it will have an immediate and far-reaching impact on various applications. Such applications are the following: surveillance videos, robotics, assistive technology for the visually impaired, content-based indexing (e.g., Flickr, Google, movies), and advanced human machine interfaces and automotive safety. Among these applications, the impact on the techniques for autonomous driving is particularly huge as they have the potential to save numerous lives. There are a lot of benchmarks that are publicly available and they have contributed in expanding the interest and progress in this area of machine vision. One example of such benchmark is the INRIA pedestrian dataset (Dalal and Triggs [2005]). Although there are multiple existing datasets on the pedestrian detection, these existing datasets often contain a limited range of scale, occlusion, pose variation, and are fairly small. These limitations make it difficult for the model to evaluate real life problems.

Hasan et al. [2020] demonstrate how existing state-of-the-art pedestrian detectors generalize poorly from one dataset to another. They claim that these models over-fit on popular data sets and some popular datasets generally are not dense in pedestrians and diverse in scenarios. Accordingly, many studies utilize techniques like transfer learning with the availability of a large number of labels from datasets like ImageNet(Deng et al. [2009]), VOC(Everingham et al. [2014]), and COCO(Lin et al. [2015b]). Transfer learning reduces the required number of labels by reusing the knowledge learnt from a large dataset (source domain) and fine-tuning it on a smaller dataset (target domain) (Guo et al. [2018]).

Additionally, supervised learning methods usually require a large set of labeled images in order to successfully train the model. However, collecting many annotated data in different domains is costly and time-consuming. In order to reduce this bottleneck, many studies utilize the technique of semi-supervised learning. Semi-supervised learning is an approach that combines a small amount of labeled data with a large amount of unlabeled data during training in order to reduce the labeling effort. Common semi-supervised learning methods are active learning and self-training. Active learning is capable of exploiting the correlations within the unlabeled data which can help determining which data should be labeled. Subsequently, if a more informative subset of a dataset is selected, then a model would perform effectively, while saving annotation efforts. Self-training follows an iterative procedure to self-train a network that can pseudo-label data from the target domain. The self-trained network learns target-specific information from the source domain without the need for a lot of training labels from the target.

In this paper, we investigate different existing pedestrian detection methods with different semi-supervised learning methods in order to reduce the labeling effort. Our baseline models are YOLO(Redmon and Farhadi [2018]) and Faster R-CNN(Ren et al. [2015]). Transfer learning is performed on these models to reduce the required number of labels by reusing the knowledge learnt. We pre-train the model on different pedestrian datasets and fine-tune it on target data to improve the generalization ability of existed model. In order to further reduce the labeling effort, active learning and self-training are utilized. With the aforementioned techniques, the model improves on the generalization ability while learning target-specific information without additional labeling efforts.

2 Literature Review

2.1 Pedestrian Detection

Early works in pedestrian detection use handcraft features with classical machine learning methods. Dalal and Triggs [2005] propose a support vector machine classifier with histograms of oriented gradients (HOG) as features. Dollar et al. [2009] use the AdaBoost classifier with integral channel features (ICF). Dollár et al. [2014] use the AdaBoost classifier with aggregated channel features (ACF). Variations of these methods have appeared in the literature (Jiang et al. [2019], Xiao et al. [2020], Hong et al. [2015], Yang et al. [2012], Chavez-Garcia and Aycard [2016], Wang et al. [2009], Roncancio et al. [2012]). More recently, convolutional neural networks (CNNs) based approaches have shown improvements over conventional approaches in pedestrian detection. These deep learning-based pedestrian detection approaches involve either a two-stage or a single-stage approach. Examples of two-stage approaches include Region-based Convolutional Network (R-CNN) (Girshick et al. [2014]), Fast R-CNN (Girshick [2015]), and Faster R-CNN (Ren et al. [2017]). These approaches perform both region scanning and detection. Although these approaches have a high performance, their computational complexity is also high. Examples of single-stage approaches include SSD (Liu et al. [2016]) and YOLO (Redmon et al. [2015]). These approaches do not address region scanning. In general, these approaches have a high computational efficiency, but lower accuracy compared to other computationally heavy methods. Variations of the above deep learning-based approaches have also appeared in these papers: Song et al. [2018], Guan et al. [2019], Gonzalez Alzate et al. [2016], Hosang et al. [2016], Brazil et al. [2017]. Although there are a lot of studies in the field of object detection, pedestrian detection is still a challenging task because the datasets for the pedestrian detection are relatively small. Due to this limitation, the pedestrian detection methods tend to have the overfitting problem and have a poor generalization on a new dataset. In order to solve these bottlenecks, many studies utilize the transfer learning technique.

While the need for pedestrian detection is gaining, pedestrian detection using traditional detection in RGB camera has the limitation that RGB cameras only work well under certain lightning conditions. Thermal images, on the other hand, can be used in night, fog or smoke robustly, allowing the algorithms to be applied in different environments. Thus, some research tries to use thermal images to detect pedestrians. Lin et al. [2015a] presents a new feature extraction and classify the pedestrians in thermal images using a three-layer feed-forward neural network. John et al. [2015] presents an adaptive fuzzy C-means method to cluster and segment the images to retrieve the candidates and use CNN to learn the features and binary classification. Ghose et al. [2019] utilizes a saliency map as attention mechanism in faster R-CNN to improve the performance of pedestrian tracking in thermal images, especially for the images collected at daytime. However, compared with RGB camera, thermal cameras are less popular and the dataset is much smaller. To solve the problem, Guo et al. [2019] tries to learn an image transformer to generate synthetic thermal images for data augmentation to train the network. Some research also uses thermal images and RGB images together for pedestrian detection. For example, Li et al. [2019] uses faster R-CNN to detect pedestrian in multispectral images with different information fusion method. Cao et al. [2019] leverages the information from two different images to build an auto-annotation framework for pedestrian detection.

2.2 Transfer Learning

Transfer learning is a learning method relaxing the assumption that the target domain has similar data distribution with the training domain. And transfer learning on deep learning is an ongoing research area. Tan et al. [2018] divides deep transfer learning method into four categories: instances-based, mapping-based, network-based, and adversarial-based transfer learning, and reviews current research on transfer learning for deep neural network. Deep transfer learning has been applied in object detection as well. For example, Talukdar et al. [2018] explores the strategies to generate a synthetic dataset to train multi state-of-the-art deep neural networks and to transfer to detect real-world object. Roy Chowdhury et al. [2019] uses self-training to generalize a faster R-CNN (FRCNN) between different dataset. In pedestrian detection, some researchers use transfer learning to transfer the knowledge of a generic pedestrian detector to train a scene-specific detector with limited labels or without labels. Early works are done in traditional machine learning using semi-supervised (Xie et al. [2010]) or unsupervised (Wang et al. [2013], Ye et al. [2017]) method. Some latest works on deep transfer learning include: Zeng et al. [2014] proposes an unsupervised self-learning deep model to learn scene-specific features and distribution of the target domain. Geng et al. [2016] proposes a deep person re-identification model based on ImageNet and uses a two-step fine-tuning method. Guan et al. [2019] iteratively generates pseudo annotation to train the pedestrian detector for target domain.

2.3 Active Learning

Semi-supervised learning reduces the labeling effort required to prepare the training set by training the model with a small number of fully labeled examples with an additional set of unlabeled examples. One of the semi-supervised approaches is active learning technique. Active learning (AL) reduces the labeling cost by iteratively selecting the most valuable data to query their labels from the annotator (Settles [2009]). Active learning allows the model to achieve higher accuracy with fewer samples (Xu et al. [2013]). In relations to the field of machine learning, querying techniques by committee (Seung et al. [1992]) or expected error reduction (Roy and McCallum [2001]) are used to identify informative instances. These techniques, however, does not work well to deep learning and therefore many approaches have been proposed in order to integrate active learning and deep learning. Wang et al. [2017] proposed a method which uses CNN to pick informative instances while pseudo-labeling the instances that the model is very certain about.

2.4 Self-Training

Self-training is another technique in semi-supervised learning which allows a machine learning model to be trained on its own predictions. An initial model is constructed by using the fully labeled data. The model then pseudo-labels samples. If the confidence of a sample is above the threshold, it is added to the training set and the process repeats (Rosenberg et al. [2005]). This method assumes that high confidence predictions are correct despite the overall accuracy of the predictor (Radosavovic et al. [2017], Hinton et al. [2015]). Zhou and Li [2005] incorporate the idea of tri-training into the self-training technique. Tri-training simultaneously trains three classifiers on the source domain where each classifier is fine-tuned on target domain samples that are unanimously labeled by the other two classifiers. Tang et al. [2016] transfers he knowledge between similar objects from visual and semantic domains to improve the performance of semi-supervised training for object detection. Kim et al. [2019] uses adversarial background regularization in self-training to help extract features for backgrounds to reduce the domain shift, and use a weak self-training method to reduce the effects of false positive pseudo labels. Tang et al. [2020] presents a consistency-based proposal learning module and a self-supervised proposal learning module to encourage the learning of context information and the consistency of bounding boxes. Chen et al. [2020] improves the teacher model in knowledge distillation to allow the teacher model to also learn from the student model.

3 Contribution

In our project, we compare two semi-supervised transfer learning strategies with different models for pedestrian detection in thermal images. We apply self-training and active-learning, two popular semi-supervised transfer learning algorithms, in our task, and we explain why they have good or bad performance, and what the issues were when applying them in the pedestrian detection task. We test the algorithms on two object detection models: Faster R-CNN and YOLO, to show that these algorithms are not limited to the model architectures.

3.1 Models

Faster R-CNN: Faster R-CNN (Ren et al. [2015]) is a commonly used algorithm for object detection. It contains the convolutional networks as a feature extraction and the regional proposal networks to find the bounding box. The proposed regions and features are fed into a classification network after ROI pooling. In our case, we use ResNet50 as the backbone for the feature extractions. For Faster R-CNN, the loss is computed as the following:

$$Loss = L_{cls} + L_{reg} = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i*) + \frac{1}{N_{reg}} \sum_i p_i * L_{reg}(t_i, t_i*) \tag{1}$$

YOLO-v3: YOLO object detector predicts an objectness score for each bounding box using a logistic regression. In the training stage, it marks a bounding box containing an object if the bounding box overlaps a ground truth object by a threshold. If a bounding box prior is not assigned to a ground truth object, it incurs zero classification loss. Due to this factor, YOLO can be trained in a relatively short time. We use YOLOv3 with Darknet-53 (Redmon and Farhadi [2018]) as the baseline model for the pedestrian detection in RGB format pictures. Lastly, the model is pre-trained using COCO dataset.

3.2 Semi-supervised transfer learning

Self-training: We are using a method in Sohn et al. [2020] for self-training. The algorithm is to first train a teacher model with only the labeled images, and then do the pseudo labeling for training. The procedure to train the teacher model is the same as a standard supervised training. After the teacher model is trained, the algorithm then uses the teacher model to give the unlabeled data with pseudo labels. We only use the labels with a confidence score higher than a given threshold, which means that the teacher model is confidence about the output results. The instances with pseudo-labels are used to train the new model, and the new model pseudo-label the unlabeled data again. The loss of the self-training is slightly different from the classical supervised training. The loss from the pseudo-labels are multiplied with a lower weight, since they are not as reliable as the true labels.

Active learning: In active learning, we first train the model over a randomly selected labeled dataset. After a series of training steps, the trained model evaluates over unlabeled data to determine what data can help the model to learn most. In our experiment, we assume that images with highest sum entropy are the most informative. We calculate this by

$$entropy -= prob * \log_{number\_of\_classes} prob \tag{2}$$

where prob is the output softmax value for each class. In each iteration, a batch of images with highest sum entropy will be selected and labeled, and added to the training set for the next training iteration.

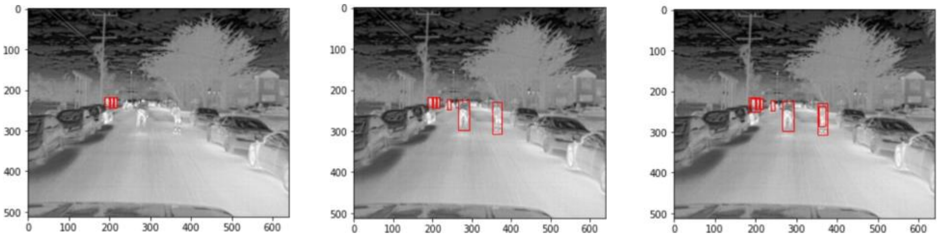
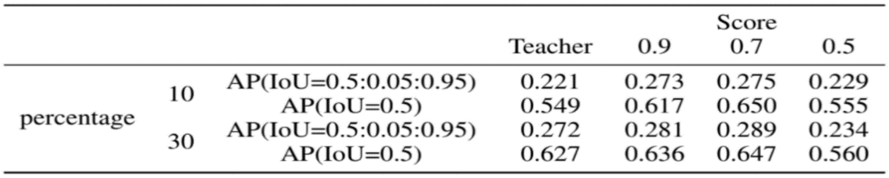


Figure 1: The quality of the pseudo labels under different score threshold. The teacher model is trained with only 10% labels and the score threshold is set as 0.9(left), 0.7(middle) and 0.5(right).



				Score		
			Teacher	0.9	0.7	0.5
percentage	10	AP(IoU=0.5:0.05:0.95)	0.221	0.273	0.275	0.229
		AP(IoU=0.5)	0.549	0.617	0.650	0.555
	30	AP(IoU=0.5:0.05:0.95)	0.272	0.281	0.289	0.234
		AP(IoU=0.5)	0.627	0.636	0.647	0.560

Table 1: The metrics of the models after self-training with different confidence scores.

In this way, theoretically, images that help the model learn most will be added to the training set batch by batch after each iteration and much less label efforts are required.

4 Experimental evaluation

4.1 Experiments

Self-training on Faster R-CNN: In this experiment, we use ResNet50 as the backbone for feature extraction and follow other structures demonstrated in the experimental details in Ren et al. [2015]. The model is pre-built in the open-source torch vision with pre-trained weights on COCO dataset. The score threshold is set as 0.9 and the weight for unlabeled loss is 0.2. When training the teacher model, we split the labeled training dataset to 70% for training and 30% for validation. We train the teacher model until the loss for validation set does not decrease. The training usually stops in 3-4 epochs, since the given number of images are small. We relabel the pseudo-labels in every training epochs for self-training. During our experiments, we found several issues that add difficulties in implementing self-training. Firstly, it is difficult to tell when to stop the training process. For supervised training, we stop the training when the loss on the validation set is not decreasing. But for self-training, the loss includes both the labeled loss and unlabeled loss. Therefore, it is hard to indicate how the model is working from the total loss, since we do not know the labels for the unlabeled data. The accuracy of the model on pseudo-labels is usually high even after long training epochs. It is possible that the given pseudo-labeling is only strengthening the original learned knowledge of the model. So instead of stopping the training when the performance on all the validation set is decreasing, we stop the training when the precision with IoU threshold as 0.5 is decreasing for the labeled validation dataset. Following this strategy, the training process stops early (2 4 epochs). We also ran experiments to test the model performance under different confidences scores for pseudo-labeling, to test the effects of different score threshold on self-training. The scripts are a combination of open-source code and scratch. We use the model and the coco-evaluator from the torch vision, and other parts, including the dataset, dataset splitting, teacher model training, self-training, and some helper functions to visualize the results.

Self-training on YOLO: In this experiment, we use Darknet-53 architecture for YOLOv3. In order to use the FLIR thermal dataset on YOLOv3, some pre-processing needs to be done. In the pre-processing phase, we select images that contain "person" object and save the map of bounded boxes information as annotations. We then transfer the image format from jpeg to jpg, map the 1 channel thermal image to 3 channels. For self-training, we first train our model on a subset of training set, then use this trained model to detect pedestrians on the unlabeled dataset. The model outputs the bounding box with its own confidence score, τ, for each pedestrian detection. The threshold is a hyper-parameter. If the detected confidence score τ is above the threshold, we mark it as a labeled detection and put it in the training set to train our model in the following loop. In this implementation, we use 20% of the labeled training set to initially train our model and use the threshold value of 0.5 to pseudo-label images for further training.

Active learning on Faster R-CNN: For active learning on Faster R-CNN, a deep fully convolutional network is used to propose regions, a ROI pooling is used to convert the ROI to a fixed feature map, and a Fast R-CNN detector (Girshick [2015]) is used to classify the object. The Faster R-CNN is implemented in Detectron2 (Wu et al. [2019]). Detectron2 also contains a large set of pre-trained R-CNN models trained on ImageNet-5k, which is suitable for transfer learning. We use ResNet-101 (Xie et al. [2016]) as a feature pyramid network (FPN) in the RPN because it has the highest accuracy on COCO dataset in the provided pre-trained models reported by Detectron2. For the dataset, we divide it into three parts: a fixed size test set, a randomly selected labeled training set, and an unlabeled set. First, we train the model with the initial training set. After each iteration, the trained model evaluates images in the unlabeled set. Based on the entropy of those images, our strategy selects a batch of the most informative images based on the summed entropy. We then label those images and add them to the training set. In our experiment, the initial training set size is 2000. We run it for 20 iterations, adding 300 most informative images to the training set at each iteration. We then record model performances in each iteration to see how active learning can speed up model learning.

Active learning on YOLO: For YOLO, similar to the experiments we did for self-training on YOLO, we use Darknet-53 architecture for YOLOv3 and do the same pre-processing. After pre-processing the data, we compute the entropy of the Bernoulli random variable at each position in the probability map of a specific class. After assigning a score to each image, the aggregation process is needed to select images with a lot of uncertain detected objects in them. Then, we select a batch of N samples based on the informativeness score computed in the previous step by selecting the top N scoring images for labeling and add them to the training set. In our experiment, we run it for 50 iterations, adding different percentage of most informative images to the training set at each iteration.

The entropy strategy we used in both architectures is called sument. This strategy calculates the sum of potential entropy in each image over the unlabeled data set and pick images with the highest sum entropy. We also used max entropy sampling and random sampling, but the performances are not as good as using sument.

4.2 Dataset

For pre-training process, we use COCO (Lin et al. [2015b]) dataset which is a large dataset with RGB images of different objects. For the target domain in transfer learning, we use FLIR thermal dataset (https://www.flir.com/oem/adas/adas-dataset-agree/). FLIR dataset includes images collected by thermal sensors for autonomous driving. Thermal sensors are not influenced by lighting as RGB cameras are. Therefore, it can be used in many challenging conditions for autonomous driving, such as darkness, fog, and smoke. Additionally, the FLIR thermal sensors have higher detection ranges which can be helpful for the autonomous driving. Also, the thermal sensors can be combined with RGB cameras for more robust sensing. The FLIR dataset includes 8862 training images and 1366 testing images. The data are saved as 8-bit gray-scale images.

4.3 Evaluation Metrics

We use the coco-evaluation metrics since it is a widely accepted metric for object detection tasks. We compare the average precision with different IoU thresholds (from 0.5 to 0.95), and the precision with IoU threshold being 0.5. A detection is considered successful only if the IoU with ground truth is larger than the threshold. We choose the first metric because it reflects the precision of detection under different IoU thresholds, and it is used in many object detection competition. We chose the second metric because it has a lower IoU threshold. By comparing it with the first metric, we can separate the performance of object class prediction and bounding box regression.

4.4 Description of Baseline

The baseline model is the model trained on the full labeled data (which is the teacher model in the self-training). For Faster R-CNN, the baseline result can be seen in Table 4, under the "teacher"

Model	Average IoU	Accuracy score	Recall
YOLO	0.3382	0.6083	0.7453
YOLO with 20% FLIR	0.4272	0.6500	0.6036
YOLO with self-training	0.4383	0.6723	0.6372

Table 2: Performance on YOLO with and without 20% of FLIR training set

AP(IoU=0.5) on different percentage of labels	30%	50%	70%
AL	70.2	71.8	72.4
random	67.7	68.2	69.5

Table 3: Performance on YOLO with active learning method in comparison to random data selection with 30%, 50%, and 70% instances queried respectively.

column. For YOLO, Table 2 shows the drop of performance when directly change the domain from RGB image to thermal image without any semi-supervised learning method. As seen in the table, the performance is improved when we apply 20% of the FLIR training datasets on the YOLO model.

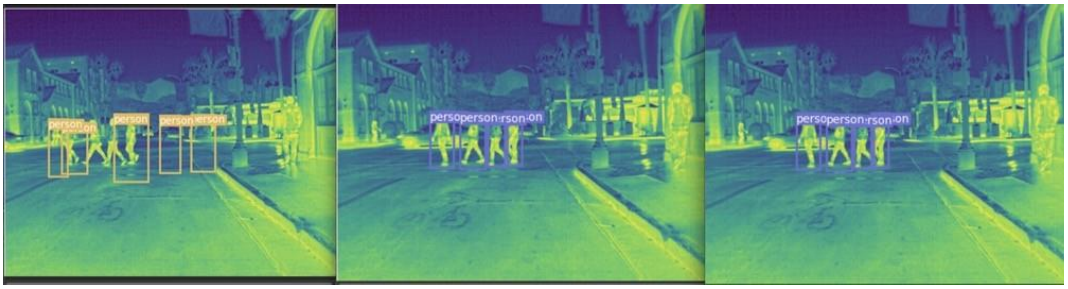


Figure 2: The image on the left shows the detection with pre-trained YOLO model, the image on the midle shows the detection with another 20% of FLIR training set, the image on the right shows the detection with self-training

5 Results and Discussions

Self-training on Faster R-CNN: The quantitative results can be seen in Table 4. As seen in the table, self-training can increase the precision. For the smaller labeled dataset, the performance of the model has higher improvement. This demonstrates how self-training allows the model to learn extra knowledge from the unlabeled domain. With self-training on 10% labeled data, we can achieve a similar training performance as on 30% labeled data. However, when the number of labeled data increases, the difference between the teacher model and the self-training becomes minimal. A possible reason might be because the pseudo-labeled data is repeating the knowledge already learnt by the teacher model. The two precision results have a large difference. While the precision with IoU with a threshold at 0.5 is decent, the mean precision across different IoU thresholding is small. The original paper Sohn et al. [2020] also shows that the mean precision over different IoU thresholding is low. One of the possible reasons is that the pseudo-bounding box is not accurate enough for the model to learn the correct bounding boxes. Object detection is a complicated task and the teacher model cannot learn enough knowledge to generate precise bounding boxes for self-training. When we increase the threshold of IoU, the precision decreases and the mean precision is small.

For the experiment of testing different score threshold, we first tested the quality of pseudo-labels and the result is in Figure 1. The result shows that the pseudo labels have the best performance when it is set as 0.7. We then ran self-training with different thresholds and ratio of labeled data, and the result is in Table 1. The result shows that the precision has the most increases when the threshold is set as 0.7, which is in accord with the image examples in Figure 1.

Percentage of labels	10%		30%		50%	
Models	teacher	self-training	teacher	self-training	teacher	self-training
AP(IoU=0.5:0.05:0.95)	0.221	0.273	0.272	0.281	0.337	0.345
AP(IoU=0.5)	0.549	0.617	0.627	0.636	0.758	0.765

Table 4: Selected metrics for Faster R-CNN with self-training. The AP stands for average precision. The percentage of labels are the labels used in training dataset. "Teacher" refers to the performance of the teacher model which is trained only with the labeled data, and "self-training" means the performance of the model after self-training.

Iterations	5		10		15		20	
Method	AL	Rand	AL	Rand	AL	Rand	AL	Rand
AP(IoU=0.5:0.05:0.95)	45.82	44.00	46.84	44.78	46.84	46.00	47.36	46.84
AP(IoU=0.5)	86.79	84.70	88.29	86.73	88.77	87.26	88.89	88.02

Table 5: Selected metrics for Faster R-CNN with active learning. Compare model performances with and without active learning

Self-training on YOLO: We compare the accuracy between the following 2 cases. First case, we train the pre-trained (with RGB COCO dataset) model with 20% of the thermal images (from FLIR_ADAS dataset) for 4 epochs. Second case, the self-training case, we train the same pre-trained model with 20% of the thermal images, and then, we do the detection for the rest 80% images without telling the model the ground truth label. With threshold of 0.5, the model generates pseudo labels for the rest images, then we use these pseudo labeled images together with the previous 20% labeled images to train the model again, then do the detection again, repeat this process for 4 times. The accuracy does not increase much. This may due to the reason that even take rest 80% of FLIR_ADAS images(7K) initialized as unlabeled dataset to realize self-training, the unlabeled dataset size is still very small, but it do helps improve the accuracy of the model and without let human do the label work. Thus, in reality, as unlabeled images are much cheaper to get compared to labeled images, the model can keep learning alone the time when it do the prediction. Thus, it still will be worth to utilize the self-training when the initial training set size is small, until it converge. But the process of doing self-training is relatively slow than doing fine-tuning or just directly use the pre-trained model, it may not suitable when the compute resource is limited.

Active learning on Faster R-CNN: In our experiments, we consider same metrics as above, AP at IoU = 0.5 and AP at IoU = 0.5:0.05:0.95. Intersection over Union (IoU) is used when calculating mean AP. It is a number from 0 to 1 that specifies the amount of overlap between the predicted and ground truth bounding box. When IoU=0.5, we treat our predictions as positive when the predicted bounding boxes have half overlap with the ground truth bounding boxes.
As stated before, we add a batch of most informative samples to the training set and train the model again. The results of some iterations are proposed in Table 5. As a comparison, we also propose results with random samples adding to the training set. From the results we can tell that active learning can speed up model converging and requires less labeled data to get a good enought performance.

Active learning on YOLO: The training details of active learning on YOLO is the same as that of the active learning on Faster R-CNN. We also compare the results of active learning to random data selection with different percentage of labeled data on YOLO. As seen in Table 3, active learning improves the overall performance for different number of labeled data queried. For both active learning and random selection, Data selection with 30%, 50%, and 70% instances are queried respectively.
Additionally, in order to increase the performance, we have tried ensembling different architectures with self-training and active learning techniques. Inspired by works done by Yoo and Kweon [2019], we have implemented the loss prediction module which learns to predict target losses of unlabeled inputs. Most active learning techniques are designed specifically for the given target tasks and architectures, but the active learning loss is task-agnostic. After predicting the losses of inputs, the module suggests data that the model is likely to incorrectly predict. We have successfully incorporated this learning loss module on an object detection task, but did not combine it with self-training due to various reasons. These reasons will be explained in Section 6.

6 Conclusion and Future Scope

From the active learning techniques, we have found that we can reduce the redundant information in the training set and that we can reduce the number of needed labels in the training set in order to generate comparable results to those of the full labeled dataset. The results demonstrate that active learning can decrease the labeling efforts overall. However, active learning still requires humans to manually add extra labels during the training procedures. Therefore, the labeling effort is larger than that of self-training.

On the other hand, the self-training techniques can improve the performance of the pedestrian detection network without extra labeling efforts. However, there are various hyper-parameters to be tuned and the improvement of the performance is not significant. We also found that as the number of labeled data increases, the improvement of self-training becomes smaller. It might be due to the fact that the teacher model already learns enough knowledge from the labeled data, hence the extra generated pseudo-labels do not provide significant amount of new information to the model. We also found that the bottleneck of the self-training as the teacher model cannot learn the bounding boxes well enough. For active learning, the bounding boxes used in training are accurate as the technique uses actual labels. For self-training, it needs to predict the bounding boxes and pseudo-labels them boxes which is a complicated tasks for the model to perform well. Due to these reasons, when the IoU threshold increases, the precision decreases greatly.

We initially thought about ensembling different architectures trained by different techniques such as active learning and self-training. However, because the performance of the models trained by self-training is much lower than that of the models trained by active learning, we have concluded that the ensemble learning is not the best way to improve the overall performance as self-training cannot give extra information to the models trained by active learning. As the performance of self-training is too low, it will only go lower if we use active learning because we query less instances.

Our future works include the following:

1. Explore different sets of parameters for self-training. Our future work will be adding an autonomous hyper-parameter selection in self-training. For example, we can evaluate the confidence scores on the labeled dataset in order to select the score threshold automatically. We then can change the score threshold in different epochs automatically which will improve the performance of self-training.

2. Combine active learning loss prediction module with self-training under different architectures. As of now, the performance of self-training is too low and will only go lower if we query less instances. However, if we can improve the performance of self-training, we can combine it with active learning. Ideally, this will reduce the labels that humans have to manually add during the training procedures as self-training can pseudo-label the queried instances.

References

Garrick Brazil, Xi Yin, and Xiaoming Liu. Illuminating pedestrians via simultaneous detection and segmentation. pages 4960–4969, 10 2017. doi: 10.1109/ICCV.2017.530.

Yanpeng Cao, Dayan Guan, Weilin Huang, Jiangxin Yang, Yanlong Cao, and Yu Qiao. Pedestrian detection with unsupervised multispectral feature learning using deep neural networks. *Information Fusion*, 46:206–217, 2019.

R. O. Chavez-Garcia and O. Aycard. Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):525–534, 2016. doi: 10.1109/TITS.2015.2479925.

Cong Chen, Shouyang Dong, Ye Tian abd Kunlin Cao, Li Liu, and Yuanhao Guo. Temporal self-ensembling teacher for semi-supervised object detection. *arXiv preprint arXiv:2007.06144*, 2020.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. doi: 10.1109/CVPR.2005.177.

J. Deng, W. Dong, R. Socher, L. 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, 2009. doi: 10.1109/CVPR.2009.5206848.

P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311, 2009.

P. Dollár, R. Appel, Serge J. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1532–1545, 2014.

M. Everingham, S. Eslami, L. Gool, C. K. Williams, J. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111: 98–136, 2014.

Mengyue Geng, Yaowei Wang, Tao Xiang, and Yonghong Tian. Deep transfer learning for person re-identification. *arXiv preprint arXiv:1611.05244*, 2016.

Debasmita Ghose, Shasvat M Desai, Sneha Bhattacharya, Deep Chakraborty, Madalina Fiterau, and Tauhidur Rahman. Pedestrian detection in thermal images using saliency maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. doi: 10.1109/CVPR.2014.81.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

Alejandro Gonzalez Alzate, Zhijie Fang, Yainuvis Socarras, Joan Serrat, David Vázquez, Jiaolong Xu, and Antonio López. Pedestrian detection at day/night time with visible and fir cameras: A comparison. *Sensors*, 16:820, 06 2016. doi: 10.3390/s16060820.

Dayan Guan, Xing Luo, Yanpeng Cao, Jiangxin Yang, Yanlong Cao, George Vosselman, and Michael Ying Yang. Unsupervised domain adaptation for multispectral pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

Tiantong Guo, Cong Phuoc Huynh, and Mashhour Solh. Domain-adaptive pedestrian detection in thermal images. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1660–1664. IEEE, 2019.

Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: Transfer learning through adaptive fine-tuning, 2018.

Irtiza Hasan, Shengcai Liao, Jinpeng Li, Saad Ullah Akram, and Ling Shao. Pedestrian detection: The elephant in the room, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

Gwang-Soo Hong, Byung-Gyu Kim, Youngsup Hwang, and Kee-Koo Kwon. Fast multi-feature pedestrian detection algorithm based on histogram of oriented gradient using discrete wavelet transform. *Multimedia Tools and Applications*, 75, 01 2015. doi: 10.1007/s11042-015-2455-2.

J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):814–830, 2016. doi: 10.1109/TPAMI.2015.2465908.

Yingjun Jiang, Jianxin Wang, Yixiong Liang, and Jiazhi Xia. Combining static and dynamic features for real-time moving pedestrian detection. *Multimedia Tools and Applications*, 78, 02 2019. doi: 10.1007/s11042-018-6057-7.

Vijay John, Seiichi Mita, Zheng Liu, and Bin Qi. Pedestrian detection in thermal images using adaptive fuzzy c-means clustering and convolutional neural networks. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 246–249. IEEE, 2015.

Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6092–6101, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Chengyang Li, Dan Song, Ruofeng Tong, and Min Tang. Illumination-aware faster r-cnn for robust multispectral pedestrian detection. *Pattern Recognition*, 85:161–171, 2019.

Chun-Fu Lin, Chin-Sheng Chen, Wen-Jyi Hwang, Chih-Yen Chen, Chi-Hung Hwang, and Chun-Li Chang. Novel outline features for pedestrian detection system with thermal images. *Pattern Recognition*, 48(11):3440–3450, 2015a.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015b.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016. ISSN 1611-3349. doi: 10.1007/978-3-319-46448-0_2. URL http://dx.doi.org/10.1007/978-3-319-46448-0_2.

Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning, 2017.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. URL http://arxiv.org/abs/1506.02640.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6): 1137–1149, 2017. doi: 10.1109/TPAMI.2016.2577031.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

H. Roncancio, A. C. Hernandes, and M. Becker. Vision-based system for pedestrian recognition using a tuned svm classifier. In *2012 Workshop on Engineering Applications*, pages 1–6, 2012. doi: 10.1109/WEA.2012.6220095.

C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, volume 1, pages 29–36, 2005. doi: 10.1109/ACVMOT.2005.107.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 441–448, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Aruni RoyChowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–790, 2019.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 287–294, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: 10.1145/130385.130417. URL https://doi.org/10.1145/130385.130417.

Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.

H. Song, I. K. Choi, M. S. Ko, J. Bae, S. Kwak, and J. Yoo. Vulnerable pedestrian detection and tracking using deep learning. In *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–2, 2018. doi: 10.23919/ELINFOCOM.2018.8330547.

Jonti Talukdar, Sanchit Gupta, PS Rajpura, and Ravi S Hegde. Transfer learning for object detection using state-of-the-art deep neural networks. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 78–83. IEEE, 2018.

Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.

Peng Tang, Chetan Ramaiah, Ran Xu, and Caiming Xiong. Proposal learning for semi-supervised object detection. *arXiv preprint arXiv:2001.05086*, 2020.

Yuxing Tang, Josiah Wang, Boyang Gao, Emmanuel Dellandréa, Robert Gaizauskas, and Liming Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2119–2128, 2016.

Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, Dec 2017. ISSN 1558-2205. doi: 10.1109/tcsvt.2016.2589879. URL http://dx.doi.org/10.1109/TCSVT.2016.2589879.

X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *2009 IEEE 12th International Conference on Computer Vision*, pages 32–39, 2009. doi: 10.1109/ICCV.2009.5459207.

Xiaogang Wang, Meng Wang, and Wei Li. Scene-specific pedestrian detection for static video surveillance. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):361–374, 2013.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Feng Xiao, Baotong Liu, and Runa Li. Pedestrian object detection with fusion of visual attention mechanism and semantic computation. *Multimedia Tools and Applications*, 79, 06 2020. doi: 10.1007/s11042-018-7143-6.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

Yao-Fang Xie, Song-Zhi Su, and Shao-Zi Li. A pedestrian classification method based on transfer learning. In *2010 International conference on image analysis and signal processing*, pages 420–425. IEEE, 2010.

Yan Xu, Fuming Sun, and Xue Zhang. Literature survey of active learning in multimedia annotation and retrieval. pages 237–242, 08 2013. doi: 10.1145/2499788.2499794.

Y. Yang, W. Liu, Y. Wang, and Y. Cai. Research on the algorithm of pedestrian recognition in front of the vehicle based on svm. In *2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering Science*, pages 396–400, 2012. doi: 10.1109/DCABES.2012.108.

Qixiang Ye, Tianliang Zhang, Wei Ke, Qiang Qiu, Jie Chen, Guillermo Sapiro, and Baochang Zhang. Self-learning scene-specific pedestrian detectors using a progressive latent model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 509–518, 2017.

Donggeun Yoo and In So Kweon. Learning loss for active learning. *CoRR*, abs/1905.03677, 2019. URL `http://arxiv.org/abs/1905.03677`.

Xingyu Zeng, Wanli Ouyang, Meng Wang, and Xiaogang Wang. Deep learning of scene-specific classifier for pedestrian detection. In *European Conference on Computer Vision*, pages 472–487. Springer, 2014.

Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. on Knowl. and Data Eng.*, 17(11):1529–1541, November 2005. ISSN 1041-4347. doi: 10.1109/TKDE.2005.186. URL `https://doi.org/10.1109/TKDE.2005.186`.