Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра математики

Курсова робота

на тему: «**Domain Adaptation for Object Detection**»

Виконав: студент 3-го року навчання,

Освітньої програми «Прикладна математика», 113

Шпіганович Владислав Олегович

Керівник Швай Н.О.,_____

старший викладач, кандидат наук

Рецензент

_____

(прізвище та ініціали)

Кваліфікаційна робота захищена з оцінкою

_____

Секретар ЕК

_____

«____» _____

20____ р.

Київ – 20___

Календарний план виконання роботи:

№ п/п	Назва етапу кваліфікаційної роботи	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи	20.10.2022	
2.	Огляд літератури за темою роботи	10.12.2022	
3.	Опрацювання літератури	20.02.2023 – 15.03.2023	
4.	Перевірка експериментів	18.03.2023 – 26.03.2023	
5.	Написання текстової частини	травень 2023	
7.	Захист курсової роботи	23.05.2023	

Науковий керівник Швай Надія Олександрівна (ПІБ)

Виконавець курсової роботи Шпіганович Владислав Олегович (ПІБ)

Зміст

INTRODUCTION

Domain adaptation has become an increasingly important area of research in computer vision due to the challenges of adapting models to new tasks and datasets. The high cost of retraining and annotating models for new tasks and domains is a significant obstacle for researchers and practitioners alike.

By understanding and applying domain adaptation techniques, researchers can adopt machine learning models to new tasks and datasets without starting from scratch. This means that they can leverage the knowledge gained from previous models and adapt them to new domains, reducing the time and cost associated with building new models from scratch. However, as it can be seen from surveys made, most methods utilize two-stage detectors [3, 13], which are too slow for real-world applications.

The aim of this paper is to review domain adaptation techniques for object detection, specifically for one-stage detectors. First part describes commonly used domain adaptation methods, second part describes chosen YOLOv5 based method and the third, final part analyzes experiments we have made.

CHAPTER 1. DOMAIN ADAPTIVE OBJECT DETECTION, AN OVERVIEW

1.1. Domain adaptation as part of transfer learning

Using knowledge learnt by machine learning model while training for a particular task in order to apply it to another machine learning task is a sphere of transfer learning (TL) research problem. Given a domain , which consists of feature space F and marginal probability distribution $P(X)$, where $X = \{x_1, x_2, \dots, x_n\} \in F$. Considering domain is $D = \{X, P(X)\}\}$, the learning task can be defined as $\mathrm{T} = \{F, f: X \to Y\}$, where Y is a label feature space, and $f(\cdot)$ is an objective function which also represents conditional propability $P(X/Y)$. In general supervised approach, neural network learns how to map $x_i \to P(y_i/x_i)$ given training pairs $\{x_i, y_i\}$.

Pan et al. at their survey about transfer learning have categorized transfer learning methods into three main categories: transductive, inductive, and unsupervised [9]. Differences between them are shown in Table 1.1:

Table 1. Relationship between Traditional ML and TL settings. Source - [8]

Learning settings		Domains	Tasks
Traditional supervised learning		same	same
Transfer learning	Inductive	same	different
	Unsupervised	different	different
	Transductive	different	same

Domain adaptation is therefore defined as transductive transfer learning.

1.2. Unsupervised Domain Adaptation methods

1.2.1. General overview

Based on availability of data, domain adaptation techniques can be classified as supervised, when labeled data is in both source domain and target domain, unsupervised domain adaptation, when source domain is rich on labeled data, and target domain has no labels, but sufficient amount of images, and semi-supervised, when limited amount of labeled data from target domain is available. For purproses of this work, we focused on semi-supervised methods, which use unsupervised techniques.

According to survey about unsupervised domain adaptation [11], methods commonly used can be classified by the following categories: adversarial feature learning, pseudo-label based self-training, image-to-image translation, domain randomization, mean-teacher training and graph reasoning. Next parts describe them.

1.2.2. Adversarial feature learning

Adversarial feature learning is based on a gradient reversal layer proposed by Ganin et al. [4], this technique involves adversarial training of an object detector model using a domain discriminator. The model is trained to produce features that can fool the discriminator, while the discriminator is trained to correctly classify the features as either source or target domain. This process results in the production of domain-invariant features that can be used for detection in the target domain.

1.2.3. Pseudo-label based self-training

This technique uses highly confident predictions from a detector model, trained on source domain, to train it on the target domain. By utilizing these confident predictions, which have a higher chance of being correct in the target domain, this strategy progressively improves the performance of the model.

1.2.4. Image-to-image translation

This technique involves using an unpaired image translation model, such as Cycle-GAN [12], CUT[1], to convert target images into source-like images or vice versa. This approach helps to reduce distribution shift in the visual domain, making it easier for the detector to perform well on the source-like target images.

1.2.5. Mean-teacher training

Mean-teacher training is an effective way to improve model generalization by utilizing unlabeled data in a student-teacher framework. Teacher model has the same architecture as a student model. Given student model weights ω_s and teacher model weights ω_t, ω_t upgraded using exponential moving average of student weights:

$$\omega_t^i = \alpha * \omega_t^{i-1} + (1 - \alpha) * \omega_s^i \ (1.1),$$

where α is exponential decay factor. At the start, weigths of teacher are set as the same ones of a student, so $\omega_t^0 = \omega_s^0$.

According to paper, where they were used for domain adaptation [10], α = 0.99 is considered the most suitable for a majority of domain adaptation tasks, but later during training can be changed into 0.999. The reason behind this is that initially teacher model would gain more from short-term memory, as student model initially trains fast, and later slower, so teacher would help more from weights stability, if higher α value is set.

Further, consistency loss is calculated, based on predictions of student and teacher model. It is usually one of distance losses, such as Kullback-Leibler Divergence (KL Divergence), absolute (L1) and Euclidean (L2, realized using mean squared error) distances. It can be applied on feature representations, however for simpilicity it is commonly utilize detection losses of student and teacher models on target domain data.

CHAPTER 2. DOMAIN ADAPTATION FOR OBJECT DETECTION

2.1. Problem of two-stage detectors

In general, object detection models can be broadly categorized into two groups: the two-stage detectors and the one-stage detectors. The two-stage detectors, such as Faster R-CNN and Mask R-CNN, extract regions of interest (ROI, and then class classification and bounding box regression are performed. However, two-stage architectures suffer from slow inference time, which makes them inviable for tasks, where real-time object detection is required.

On the other hand, one-stage detectors like YOLO series directly output bounding boxes and class predictions from the predicted feature maps, utilizing pre-defined anchors. While they yield lower accuracy, than two-stage architectures, fast inference time and overall smaller sizes make them commonly used for modern computer vision applications.

However, despite the fact that domain-adaptive object detection is a popular research field, based on surveys made [3, 13], most adaptation methods are focused on the two-stage detectors, especially on Faster-RCNN[5], because this way they can utilize regions of interests generated by models for adaptation task. This makes researching methods of domain adaptation for one-stage architectures an important field, and because of this they started obtaining well-deserved attention only recently.

2.2. YOLO model

YOLOv5 is an object detection algorithm, develobed by Ultralitics team [8], that is part of the You Only Look Once (YOLO) family of models. It is designed to efficiently and accurately detect objects in real-time applications. YOLOv5 builds upon the success of previous YOLO versions, especially YOLOv3, incorporating several architectural improvements and advancements. For this work, we used release 5.0. Currently newest release is 7.0.

It's architecture contains 3 key components [7]:

- Backbone: CSP
- Neck: SPP
- Head: YOLOv3 Head

Detection loss can is defined the following way:

$$L_{det} = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{box} \ (2.1),$$

where

λ - hyperparameters. According to code implementation, $\lambda_1 = 0.3, \lambda_2 = 0.7, \lambda_3 = 0.05$.

L_{cls} – class classification loss, calculated using cross-entropy.

L_{obj} – objectness loss, which penalizes whether object existence was predicted correctly. Calculated using binary cross-entropy (BCE).

L_{box} – bounding box correctness loss. Calculated using complete IoU (CIoU) loss.

2.3. Domain adaptive YOLO

For purposes of this work, we have chosen semi-supervised domain adaptive YOLO, proposed by Zhou et al. in their paper [14].

The method is based on the release 5 of YOLOv5 detector and consists of four main components: the Mean Teacher model which was described in chapter 1.2, distillation loss, and the consistency loss. The pseudo cross-generated training images, generated by CUT[1], are used to alleviate image-level domain differences.

The distillation loss is used to remedy cross-domain discrepancy, and the consistency loss, which adopts L2 distance on detection losses of teacher model on target source-like images and student loss on target images, is used to redress cross-domain objectness bias. Labels from target domain are used only for validation, although they are available in datasets used. Architecture is shown in Fig. 2.1:

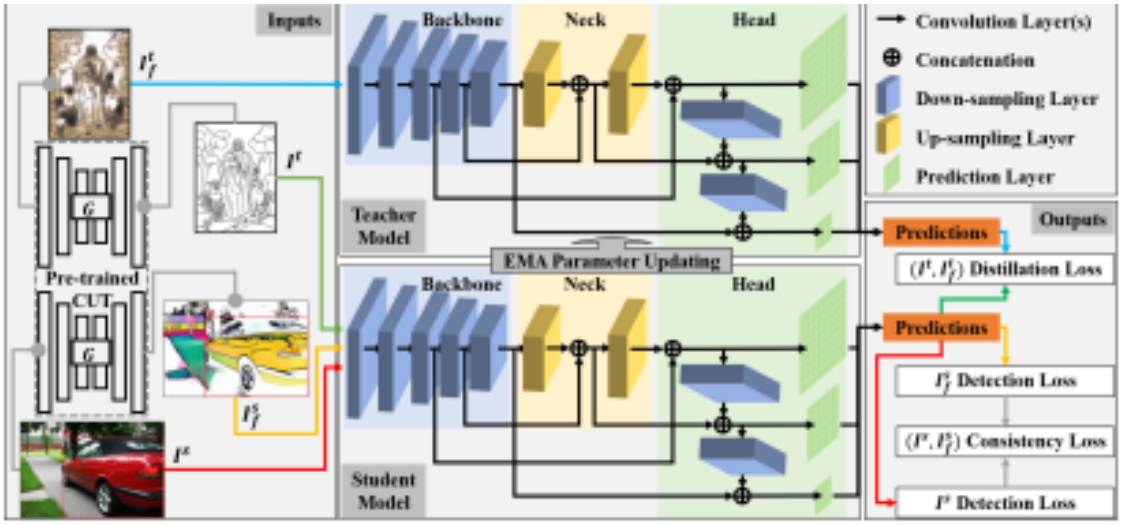

Fig. 2.1. SSDA-YOLO method architecture. Source - [14]

In this method, consistency loss is defined as L2 distance between detection losses, which are described in formula 2.1, for student model on source images I^s and source target-like images I_f^s. Distillation loss is defined as detection loss for student on target images I^t with usage of labels, obtained by teacher model on targes source-like images I_f^t. Formulas for these losses are 2.2 and 2.3:

$$L_{cons} = ||L_{det}(I^s, B^s) - L_{det}(I_f^s, B^s)||_2 \quad (2.2),$$

$$L_{dis} = L_{det}\left(I^s, P\left(I_f^s\right)\right) \quad (2.3),$$

where B^s- labels for source dataset, $P\left(I_f^s\right)$-filtered after process of non-maximum suppression predictions of teacher model, which are treated as ground-truth.

Overall loss function is:

$$L = L_{det}(I^s, B^s) + L_{det}\left(I_f^s, B^s\right) + \alpha * L_{cons} + \beta * L_{dist},$$

where α, β – hyperparameters. Authors of this method during experiments found out, that their best values are 2 and 0.005 respectively.

2.4. Model compressing in context of knowledge distillation

Apart from training, knowledge distillation techniques can be used for compressing model size. By transferring the knowledge from a larger teacher model to a smaller student model, the student model can achieve better performance than without it, while having a smaller size and lower computational requirements, than teacher model. This is useful for applications, where larger models are not suitable, because memory during inference time is limited.

In our experiments we tried using previously described architecture to transfer knowledge from YOLOv5l to YOLOv5s. Also, considering model sizes may be too big and therefore knowledge transfer may work poorly, as was found in paper [6], we used intermediate YOLOv5m architecture. This is further described in part 3.2.

CHAPTER 3. EXPERIMENTAL RESEARCH

3.1. Used datasets

3.1.1. PascalVOC

PASCAL VOC [11] is real-world dataset that consists of a diverse collection of general objects. It's merged versions from 2007 and 2012 are commonly used as one of benchmark datasets, accordingly to survey. It's train and validation splits in total contain 16551 images. Dataset contains 20 classes: airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, table, dog, horse, bike, person, plant, sheep, sofa, train, and TV.

Fig. 3.1. Sample from PascalVoc

3.1.2. **Clipart**

Clipart dataset, introduced by Inoue et al. in their Faster-RCNN based cross-domain method [2], contains a total amount of 1000 images, with them being split equally into train and validation. Classes are the same as in PascalVOC dataset.

Fig. 1.2. Sample from Clipart

3.1.3. Generated images.

For PascalVOC → Clipart style transfer images were generated in UMT method by CycleGAN, and for Clipart → PascalVOC images were generated by Zhou et al. [14], using CUT. These fake images help to reduce domain bias, which improves adaptation to target domain.

3.2. Experiments

Experiments were made, using released datasets with the discussed method code.

Following SSDA, experiments were made on PascalVOC and Clipart datasets, where PascalVOC version from 2007 and 2012 were merged and used as source domain D_s, and Clipart is used as target domain D_t. Images were resized into shape

(3, 640, 640). Experiments for YOLOv5 with large parameters (YOLOv5l) were made with batch size 36, for other smaller sizes it was set to 54. This way images can fit into 3 Nvidia RTX 3090. Baseline is set as a results when teacher exponential decay $\alpha = 0.99$ according to formula 1.1. Mean average precision (mAP) over all classes is reported.

Goals of experiments made can be categorized into following categories:

- Changing α for exponentialy moving average of teacher weights, where α is from formula 1.1.
- Changing confidence threshold for non-maximum suppression during post-processing of predictions, generated by teacher model on target source-like images. Tried values are default 0.8, 0.5 and 0.4. IoU threshold was remained the same at a value of 0.3. Overall, while lower confidence results in more predictions returned, it also means that there are more incorrect predictions.
- Experiments with knowledge distillation. Reasoning for these experiments was described previously in chapter 2.4. We decided to remain the same architecture, as in SSDA-YOLO, but exponential moving average was removed, and initial pseudo-labels are pre-computed, using YOLOv5 model with large parameters, trained at confidence threshold 0.5 and teacher $\alpha = 0.99$. This model was chosen, because it showed the best validation results among training experiments, and it is natural to choose labels, which are confident.

3.2.2. Baseline results for plain YOLOv5.

Model size	Train domain	Target mAP
Large	Source	41.9
Small		29.1
Large	Target	59.8
Small		51.0

3.2.3. Changing of alpha results comparison

alpha	Teacher mAP	Student mAP
$\alpha = 0.99$	41.4	44.9
$\alpha = 0.999$	41.5	45.3
α changes linearly from 0.99 to 0.999	41.6	43.4
α changes at epoch 100 from 0.99 to 0.999	42.2	44.3

3.2.3. Changing of confidence threshold results comparison

Confidence threshold	Teacher mAP	Student mAP
$t_{conf} = 0.8$	41.4	44.9
$t_{conf} = 0.5$	42.3	46.2
$t_{conf} = 0.4$	40.8	44.6

3.2.4. Knowledge transfer results comparison

Here baseline means best results, obtained under the settings $\alpha = 0.99$ and confidence threshold 0.5.

Transfer type	Student mAP
Large → medium	42.3

Large → small	35.7
Transferred medium → small	35.5
Medium, baseline	41.7
Small, baseline	34.6

3.3. Results analysis

Due to noisy training, it's hard to explain results for changing alpha, although it looks like when $\alpha = 0.999$ the model is training somewhat better.

However, for confidence table we can clearly see, that choosing threshold of 0.5 is the best for training. It means that we can get more target domain information by weakening filter limitations, however we should not get too greedy, as results for threshold 0.4 show. An example output of this model with 0.5 threshold is shown at Fig. 3.3.

Model compression has shown good results, and it was able to outperform normal training under domain adaptive method. The reason of lower results for medium → small transfer is that under method realization we do not use confidence scores for loss functions. This way "soften" in comparison to large model predictions are worse, because they are less accurate overall, although performance is still comaparable. This can be avoided by utilizing soft predictions, as was shown in paper [6].

Fig. 3.3. Predictions on target domain

SUMMARY

This work covers a topic of domain adaptation for object detection, reviews methods used for semi-supervised domain adaptation and researches method for domain adaptive object detection, which is based on one-stage YOLOv5, which is superior in inference time. During experiments, we evaluate possible good hyperparameter changing strategies and apply knowledge distillation based model compressing technique. The results show validity of discussed method and confirm, that knowledge transferring techniques may help in domain adaptation.

REFERENCES

1. Contrastive learning for unpaired image-to-image translation / T. Park et al. *Computer vision – ECCV 2020*. Cham, 2020. P. 319–345. URL: https://doi.org/10.1007/978-3-030-58545-7_19 (date of access: 10.05.2023).
2. Cross-Domain weakly-supervised object detection through progressive domain adaptation / N. Inoue et al. *2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Salt Lake City, UT, 18–23 June 2018. 2018. URL: https://doi.org/10.1109/cvpr.2018.00525 (date of access: 10.05.2023).
3. Deep domain adaptive object detection: a survey / W. Li et al. *2020 IEEE symposium series on computational intelligence (SSCI)*, Canberra, Australia, 1–4 December 2020. 2020. URL: https://doi.org/10.1109/ssci47803.2020.9308604 (date of access: 10.05.2023).
4. Domain-Adversarial training of neural networks / Y. Ganin et al. *Domain adaptation in computer vision applications*. Cham, 2017. P. 189–209. URL: https://doi.org/10.1007/978-3-319-58347-1_10 (date of access: 10.05.2023).
5. Faster R-CNN: towards real-time object detection with region proposal networks / S. Ren et al. *IEEE transactions on pattern analysis and machine intelligence*. 2017. Vol. 39, no. 6. P. 1137–1149. URL: https://doi.org/10.1109/tpami.2016.2577031 (date of access: 10.05.2023).
6. Improved knowledge distillation via teacher assistant / S. I. Mirzadeh et al. *Proceedings of the AAAI conference on artificial intelligence*. 2020. Vol. 34, no. 04. P. 5191–5198.

URL: https://doi.org/10.1609/aaai.v34i04.5963 (date of access: 10.05.2023).

7. Jocher G., Nishimura K., Mineeva T. YOLOV5 architecture summary. *Ultralytics YOLOv8 Docs*. URL: https://docs.ultralytics.com/yolov5/tutorials/architecture_description/ (date of access: 10.05.2023).
8. Jocher G., Nishimura T., Mineeva K. Yolov5: code repository. *GitHub*. URL: https://github.com/ultralytics/yolov5 (date of access: 17.05.2023).
9. Pan S. J., Yang Q. A survey on transfer learning. *IEEE transactions on knowledge and data engineering*. 2010. Vol. 22, no. 10. P. 1345–1359. URL: https://doi.org/10.1109/tkde.2009.191 (date of access: 10.05.2023).
10. Tarvainen,A. Valpola, H. Mean teachers are better role models: Weightaveraged consistency targets improve semi-supervised deep learning results. In NIPS, 2017.
11. The pascal visual object classes (VOC) challenge / M. Everingham et al. *International journal of computer vision*. 2009. Vol. 88, no. 2. P. 303–338. URL: https://doi.org/10.1007/s11263-009-0275-4 (date of access: 10.05.2023).
12. Unpaired image- to- image translation using cycle generative adversarial networks. *Regular*. 2020. Vol. 9, no. 6. P. 380–385. URL: https://doi.org/10.35940/ijeat.f1525.089620 (date of access: 10.05.2023).
13. Unsupervised domain adaptation of object detectors: a survey / P. Oza et al. *IEEE transactions on pattern analysis and machine intelligence*. 2023. P. 1–24. URL: https://doi.org/10.1109/tpami.2022.3217046 (date of access: 10.05.2023).
14. Zhou H., Jiang F., Lu H. SSDA-YOLO: Semi-supervised domain adaptive YOLO for cross-domain object detection. *Computer vision and image understanding*. 2023. P. 103649.

URL: https://doi.org/10.1016/j.cviu.2023.103649 (date of access: 10.05.2023).