Transfer learning methods for training person detector in drone imagery

Saša Sambolek[1] and Marina Ivašić-Kos[2]

[1] High school Tina Ujevića, Kutina, Croatia
[2] University in Rijeka, Department of Informatics, Rijeka, Croatia
sasa.sambolek@gmail.com, marinai@uniri.hr

Abstract. Deep neural networks achieve excellent results on various computer vision tasks, but learning models require large amounts of tagged images and often unavailable data. An alternative solution of using a large amount of data to achieve better results and greater generalization of the model is to use previously learned models and adapt them to the task at hand, known as transfer learning.
The aim of this paper is to improve the results of detecting people in search and rescue scenes using YOLOv4 detectors. Since the original SARD data set for training human detectors in search and rescue scenes are modest, different transfer learning approaches are analyzed. Additionally, the VisDrone data set containing drone images in urban areas is used to increase training data in order to improve person detection results.

Keywords: transfer learning, YOLO v4, person detection, drone dataset

1. Introduction

Deep learning methods have been successfully applied in many computer vision applications in recent years. Unlike traditional machine learning methods, deep learning methods allow automatic learning of features from data and reduce manual extraction and presentation features. However, it should be emphasized that the deep learning model is highly data-dependent. Large amounts of data are needed in the learning set to detect patterns among the data, generate features of the deep learning model, and identify the information needed to make a final decision.

Insufficient data to learn deep learning models are a significant problem in specific application domains such as search and rescue (SAR) operations in non-urban areas. The process of collecting relevant image data, in this case, is demanding and expensive because it requires the use of drones or helicopters to monitor and record non-urban areas such as mountains, forests, fields, or water surfaces. The additional problem is that scenes with detected casualties rarely appear on the recorded material, which is the most useful for learning the model for detecting an injured person. Besides, the data collected should be processed, each frame inspected, and each occurrence of a person marked with a bounding box and labeled, which is a tedious and time-consuming process.

One way to overcome the problem of data scarcity is to use transfer learning. Transfer learning allows a domain model not to be learned from scratch, assuming that the learning set data is not necessarily independent and identically distributed as the data in the test set. This assumption makes it possible to significantly reduce the amount of data required in the learning set and the time required to learn the target domain model.

This paper aims to detect persons on the scenes of search and rescue (SAR) operations. Today, it has become commonplace to use drones in SAR missions that fly over the search area and film it from a bird's eye view. They can capture a larger area at higher altitudes, but then the people in the image are tiny and take up only a few pixels. People can be detected more efficiently at lower altitudes, but in that case, the field of view is smaller. People who are searched for are very often barely noticeable because of the branches and trees, occluded by some vegetation, in the shadow, fused with the ground, which further complicates the search even for favorable weather conditions. During SAR operations, the drone operator has a demanding task to analyze the recorded material in real-time to detect a relatively small person on a large, inaccessible surface that requires great concentration, so automatic detection can be valuable.

We used the YOLOv4 model for the person detector trained on the MS COCO dataset, which proved to be the most successful in previous research after additional learning on domain images [1-3].

To train the YOLOv4 model, we used the custom-made set of SARD scenes that were shot in a non-urban area with actors simulating injured people and prepared for machine learning. To increase the set, we have generated the Corr-SARD set from SARD scenes by adding atmospheric conditions. Since tailor-made SARD and Corr-SARD datasets were relatively small for learning deep learning models, we have additionally used the VisDrone dataset to include more images of people taken by drone, although not in non-urban areas.

This paper examined three different transfer learning methods for building YOLOv4 models for detecting persons in search and rescue operations. In the next section, three different methods of transfer learning will be presented. In the third section, the experimental setup is given along with the description of image data sets SARD, Corr-SARD, VisDrone, and basic information about the YOLO4 detector. In the fourth section, the experimental results of applying different transfer learning methods will be presented and compared. In conclusion, we list important characteristics regarding the impact of different transfer learning approaches on person detection in search and rescue scenes and a plan for future research.

2. Transfer Learning

Transfer learning involves taking a pre-trained neural network and adapting that neural network to a new distinct set of data by transferring or repurposing the learned features. Transfer learning is beneficial when learning models with limited computing resources and when a modest set of data is available for model learning.

Many state-of-the-art models took days, or even weeks, on powerful GPU machines to train them. So, to not repeat the same procedure over a long time, learning transfers allow us to use pre-trained weights as a starting point.

Different levels and methods of applying deep transfer learning can be classified into four categories according to [4]: network-based transfer learning, instance-based transfer learning, mapping-based transfer learning, and adversarial-based transfer learning, which we will not examine here.

2.1 Network-based deep transfer learning

Network-based deep transfer learning refers to the reuse of a part of the network (without fully connected layers) previously trained in the source domain and is used as part of the target network used in the target domain [4].

The CNN architecture contains many parameters, so it is difficult to learn so many parameters with a relatively small number of images. Therefore, for example, in [5], the network is first trained on a large set of data for classification (ImageNet, source domain), and such pre-trained parameters of the inner layers of the network are transferred to the target tasks (classification, detection, domain target). An additional network layer was added and trained on the labeled target set data to minimize the differences between the source and the target data regarding various image statistics (object type, camera position, lighting) and fit the model to the target data task.

Suppose the source domain and the target domain differ in scenes. In that case, the objects' appearance, lightings, background, position, distance from the camera, and similar lower detection results can be expected on target sets than achieved on the source. For example, the original model of the YOLO object detector trained on the COCO data set was used for detecting players in video frames of handball sports [6] and for person detection on thermal images [7]. In the case of player detection in handball scenes, the original YOLO model achieved an AP of 43.4%, which is often better than person detection in thermal images, where an AP of 19.63% was achieved. Lower results on thermal images are due to significant differences between thermal and RGB images. Lower detection results on handball scenes were achieved since the detector did not accurately identify the player and often drop to mark a high-raised hand or leg in the jump, as handball-specific poses did not exist in the original set.

2.2 Instances-based deep transfer learning

Instance-based deep transfer learning refers to a method in which a union of selected instances from the source domain and instances of the target domain is used for training. It is assumed that regardless of differences in domains, the source domain's instances will improve detections in the target domain.

In deep learning, the approach of fine-tuning models on the target domain, which are pre-trained on large benchmark datasets of source domains, is standard to improve results in other similar target domains. The authors in [8] use an instance-based deep transfer approach to measure each training sample's impact in the target domain. The

primary purpose was to improve the model's performance in the target domain by optimizing its training data. In particular, they use a selected pre-trained model to assess each training sample's impact in the target domain. According to the impact value, remove negative samples and thus optimize the target domain's training set.
In the previously mentioned research in the sports domain [6] and thermal images [7], it was shown that additional learning at the appropriate set and fine-tuning the parameters of the pre-trained model to tasks of interest could significantly improve the detection results at the target set. Thus, the basic model's AP on the set of thermal images with AP 19.63% with additional adjustment on the customized set of thermal images achieved AP of 97.93%. In additional learning in the handball scenes, AP increased from an initial 43% to 67%. Similar results after fine-tuning with state-of-the-art backbone deep neural networks such as Inception v2, ResNet 50, ResNet 101 were also reported in [9].

2.3 Mapping-based deep transfer learning

Mapping-based deep transfer learning refers to mapping instances from the source domain and the target domain to a new data space [4]. Mapping-based deep transfer learning finds a common latent space in which feature representations for the source and target domains are invariant [10]. In [11], a CNN architecture was proposed for domain adaptation by introducing an adaptation layer for learning feature representations. The maximum mean discrepancy (MMD) metric is used to calculate the overall structure's distribution distance concerning a particular representation, which helps select the architecture's depth and width and regulate the loss function during fine-tuning. Later, in [12] and [13], a multiple kernel variance of MMD was proposed (MKMMD) and joint MMD (JMMD) to improve domain adaptation performances. However, the main limitation of the MMD methods is that the computational cost of MMD increases quadratically with the number of samples when calculating Integral Probability Metrics (IPM) [14]. Therefore, Wasserstein distance has recently been proposed in [15] as an alternative for finding better distribution mapping.

2.4 Adversarial-based deep transfer learning

Adversarial-based deep transfer learning mainly refers to introducing adversarial technology inspired by generative adversarial networks (GAN) [16] to find transferable representations that apply to both the source and target domain but can also refer to the use of synthetic data used to enlarge the original dataset artificially.

In adversarial networks, the extracted features from two domains (source and target) are sent to the adversarial layer that tries to discriminate the features' origin. If there is a slight difference between the two types of features, the adversarial network achieves worse performance, and it is a signal for better transferability, and vice versa. In this way, general features with greater portability are revealed in the training process.

In the case of using synthesized data in order to increase the learning set of the deep learning model, it is necessary to analyze the content of the reference video scene and select elements to be generated on the virtual scene taking into account the background, objects on the scene and accessories, such as [17].

3. Experimental Setup

3.1 Dataset

In this paper, three datasets were used: the publicly available VisDrone dataset, custom-made SARD dataset and synthetically enlarged SARD dataset, Corr-SARD datasets.

From the VisDrone dataset [18] containing images of urban scenes taken by the drone, we selected 2,129 images that include a person or pedestrian tag. We combined both labels into one class: person. The obtained dataset was divided into a training set (1,598 images) and a test set (531 images). The selected dataset from the VisDrone set includes shots of people taken under different weather and lighting conditions in different urban scenarios such as roads, squares, parks, parking lots, and the like.

The SARD dataset [19] was recorded in a non-urban area to show persons in scenes specific to search and rescue operations. The set contains footage simulating poses of injured people found in inaccessible terrains in the hills, forests, and similar places by searching and rescuing actions and standard poses of people such as walking, running, sitting. The set contains 1,981 images divided into two subsets, a training set containing 1,189 images and a test set with 792 images.

The Corr-SARD dataset is derived from the SARD set so that the effects of snow, fog, frost, and motion blur are added to the SARD images. The training set has the same number of images as the SARD training set, while the test set has slightly fewer images (714) because images in which no persons are seen after adding the effect have been removed.

For the experiment, we created an additional three datasets containing images of the sets mentioned above.

Fig. 1. Example of images from SARD dataset.

The SV refers to a mixture of SARD and VisDrone sets. Similarly, the SC is a mixture of SARD and Corr set, and SVC is a mixture of SARD, VisDrone, and Corr test set.

3.2 YOLOv4 person detection model

Detection of persons in high-resolution images taken by a drone is a challenging and demanding task. People who are searched for due to loss of orientation, fall, or dementia are very often in unusual places, away from the road, in atypical body positions due to injury or fall, lying on the ground due to exhaustion, covered with stones due to slipping or landslides (Figure 1). On top of all that, the target object is relatively small and often camouflaged in the environment, so it is often challenging to observe.

In this experiment, for person detection, we used the YOLOv4 model [20]. YOLOv4 uses CSPDarkNet53 as a backbone [21] that includes the DarkNet53, a deep residual network with 53 layers, and the CSPNet (Cross Stage Partial Network). To increase the receptive field without causing a decrease in velocity, the authors added Spatial Pyramid Pooling SSP [22] as the neck, and PAN, Path Aggregation Network [23] for path aggregation, instead of the Pyramid Feature Network (FPN) used in YOLOv3. The original YOLOv3 network is used for the head [24].

In addition to the new architecture, the authors also used training optimization called "Bag of Freebies" to achieve greater accuracy without additional hardware costs, such as CutMix, Mosaic, CIoU-loss, DropBlock regularization. There is also a "Bag of Specials" set of modules that only slightly increase the hardware costs with a significant increase in detection accuracy.

To train and evaluate the YOLOv4 model, we used the Darknet framework [25], an open-source neural network framework written in C and CUDA that supports CPU and

GPU computing. For the experimentation, we used Google Colab [26], a free tool for machine learning and local computer Dell G3 i7-9750H CPU, 16 GB RAM, GeForce GTX 1660 Ti 6 GB, with Ubuntu 16.04. 64-bit operating system.

3.3 Evaluation Metrics

We use average accuracy (AP) to evaluate the detection results. AP is a metric that considers the number of correctly and incorrectly classified samples of a particular class and is used to determine the detection model's overall detection power, not just accuracy [27]. In this experiment, we have used three precision measures in the MS COCO format that takes into account detection accuracy (IoU):

- AP thresholds of 10 IoU (0.5: 0.05: 0.95),
- AP50 at IoU = 0.50,
- AP75 in IoU = 0.75.

The original COCO script was used to calculate the results.

4. Results of Transfer Learning Methods and Discussion

This section presents the overall performance results from the conducted experiments. It is worth mentioning that the pre-trained YOLOv4 with weights (yolov4.conv.137 [25]) learned on the MS COCO [28] dataset was trained on three training datasets with different transfer learning methods to identify the transfer learning variant that provides the best solution for person detection in SAR scenes.

In all cases, the YOLOv4 model was trained with a batch size of 64, a subdivision of 32, and iterations of 6000. The learning rate, momentum, and decay for the training process were set to values of 0.001, 0.949, 0.0005, and width and height to value 512.

Before training, the parameters of the original model should be changed and adapted to our domain. The first step is to change the number of classes from 80, which corresponds to the number of MS COCO classes, to 1 class, a person in this experiment. After defining the class size, each Conv filter must be set to 18 as defined in (1), where the class corresponds to the number of classes (class = 1 in our case).

$$x\ filters = (classes + 5)\ x\ 3 \tag{1}$$

The impact of applying each of the transfer learning methods in training the detection model on the detectors' results in search and rescue operations is given below.

4.1 Fine-tuning the YOLOv4 model to the target domain

In the network-based deep transfer learning, the pre-trained YOLOv4 model trained on the COCO source domain was fine-tuned to the target domain: SARD, VisDrone, or Corr-SARD dataset. The sketch of network-based deep transfer learning is shown in Fig. 2.

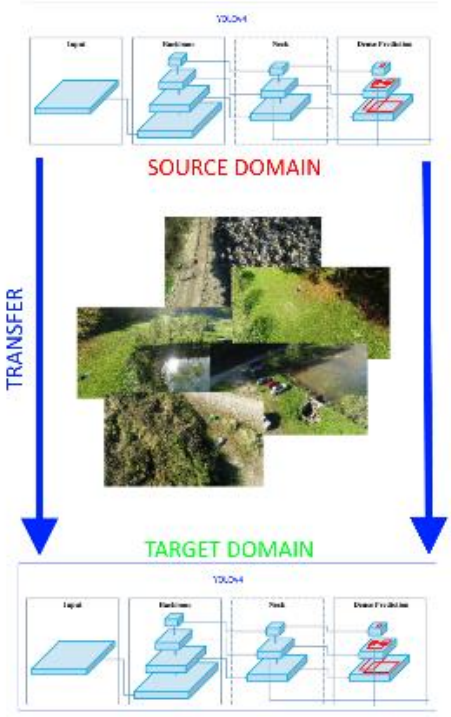

Fig. 2. A network-based deep transfer learning: the first network was trained in the source domain (in our case MS COCO), and then the pre-trained network was fine-tuned on the target domain (SARD dataset).

For a more straightforward presentation of the results, the model trained on the SARD training dataset was designated as the SARD model. The model labeled COCO refers to the pre-trained model on the MS COCO dataset.

Table 1. shows the results of person detection on SARD images concerning the AP metric with the original YOLOv4 model and the YOLOv4 model that was further trained on SARD images. The results show a significant improvement in AP (Imp 37,9) and Ap50 and AP75 metrics of the detection results after fine-tuning the model to the SARD dataset.

Table 1. Results of YOLOv4 models on SARD test dataset in case of network-based deep transfer learning

Model	AP	AP_{50}	AP_{75}	Imp
COCO	23.4	40.2	25.3	
SARD	61.3	95.7	71.7	37.9

4.2 Instances-based deep transfer learning with SARD, Corr-SARD, and VisDrone datasets

After we applied the network-based transfer learning, we applied several instance-based deep transfer learning to train further the YOLOv4 model, including a series of sets (VisDrone and Corr-SARD and SARD).

Using the VisDrone set, we selected only those instances from that set relevant to our target domain, i.e., those that contained a person. In the VisDrone training set that we used, there is approximately the same number of images as in the SARD training set, but in the VisDrone set, there are 25,876 objects more than in the SARD dataset that is 29,797 marked persons in VisDrone and 3,921 marked persons in SARD dataset.

In the first case of instance-based transfer learning, the original model was trained first on a selected part of the VisDrone dataset and then fine-tuned on the SARD training dataset (V+S model). The sketch of instances-based deep transfer learning with VisDrone and SARD dataset is shown in Fig. 3.

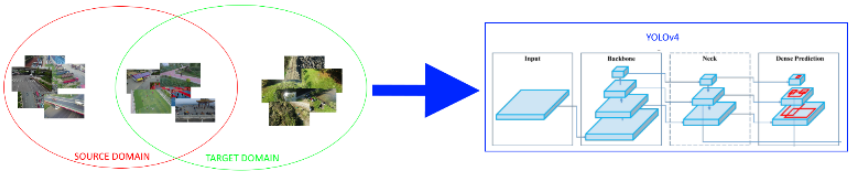

Fig. 3. Instance-based deep transfer learning. We selected only images relevant to our target domain and trained the model with it from the source domain. In the second step, the model was trained on the SARD dataset.

According to the results presented in Table 2, additional model training on the VisDrone set (model V+S) did not affect the detection results obtained on the SARD model. However, it improved the results compared to the original model (Imp 37,9).

Training on the Corr-SARD training dataset contributed to a slight improvement in detection results concerning the SARD model and significant AP improvement to the original model (Fig. 4).

Also, the results show that transfer learning is not commutative and that the order of the sets used to train the model affects the detection results. The best results are achieved when the model is fine-tuned on the dataset on whose examples it will be tested, so the V + S model achieves significantly better results than the S + V model.

We also tested instance-based deep transfer learning using three datasets so that the original model was fine-tuned on the SARD training set after training on VisDrone, and the Corr-SARD datasets (V+C+S model).

Table 2. Results of YOLOv4 models on SARD test set to build with instance-based transfer learning

Model	AP	AP_{50}	AP_{75}	Imp
S + V	22.8	41.7	23.7	-0.6
V + S	61.3	95.8	70.6	37.9
V + C + S	**62.0**	**95.9**	**71.9**	**38.6**

Table 3. shows the individual detection results on the SARD test set obtained when the original model was additionally trained on the VisDrone and Corr-SARD sets. For an easier results notation, a model trained on the VisDrone dataset is designated as VisDrone, and the model trained on the Corr-SARD as Corr-SARD.

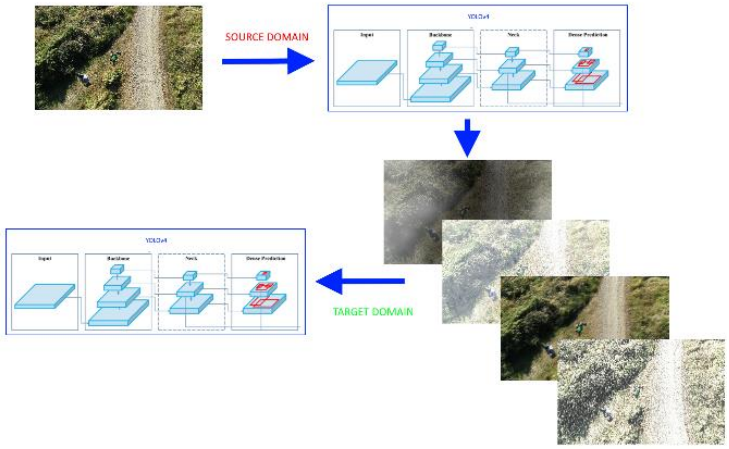

Fig. 4. Using Corr-SARD dataset for transfer learning. After training on the SARD dataset, the model was re-train with the same images with added effect.

The results are interesting and show that fine-tuning the original model to the VisDrone set even lowered the detection results even though the original COCO dataset does not include shots of people taken by the drone. The VisDrone set includes them just like the target SARD test set, but in urban areas.

The use of the synthetic Corr-SARD set contributed to improved person detection outcomes in the SARD test set.

Table 3. Results of YOLOv4 models on SARD test dataset after learning on the VisDrone set and Corr-SARD set

Model	AP	AP_{50}	AP_{75}	Imp
VisDrone	18.9	33.2	20.5	-4.5
Corr-SARD	54.9	90.5	61.9	31.5

4.3 Mapping-based deep transfer learning with images from SARD, Corr-SARD, and VisDrone datasets

In mapping-based deep transfer learning, several new sets were made for training the model as a union of images from the VisDrone, SARD, and Corr-SARD training sets. These are the SV sets created as a union of images from the SARD training set and VisDrone set, the SC model created by merging images from the SARD training set and Corr-SARD, and the SVC set created as a union of images from all three sets. A sketch of mapping-based deep transfer learning is shown in Fig. 5.

The results in Table 4. show that transfer learning on newly created sets (SV, SC, SCV) significantly contributed to the improvement of the detection result concerning the original model with a relatively high AP score achieved: for SC model 59.4%, SV 55.4%, and SVC 56.4%. The AP increase after transfer learning the model on new sets is 32 to 36 percent higher than with the original model (Imp column in Table 4.).

However, it can be noticed that the results of the model trained on the newly created sets SV, SC, SCV are comparable but still slightly lower than the case when the model was fine-tuned only on the training data from the target set (model SARD).

Table 4. Results of YOLOv4 model on SARD test set to build with mapping-based transfer learning methods

Model	AP	AP_{50}	AP_{75}	Imp
SV	55.4	92.5	60.8	32.0
SC	59.4	94.7	67.4	36.0
SVC	56.4	93.6	63.1	33.0

From the obtained results, it can be concluded that in the case of deep transfer learning based on mapping, relatively good AP results were achieved, but that results are still worse compared to deep transfer learning based on instances and network transfers. Overall, the best AP score of 62.0% was achieved with the V + C + S model, and immediately afterward, with the AP 61.3%, a SARD model was fine-tuned only on the SARD training set.

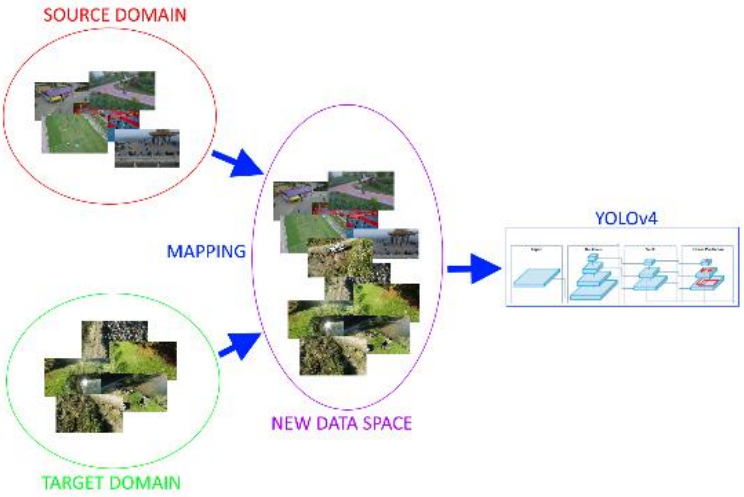

Fig. 5. Mapping-based deep transfer learning. Images from the target SARD dataset are mapped with images from the VisDrone and Corr-SARD datasets.

Additionally, to evaluate the performance of the SV, SC, SCV models built with mapping-based transfer learning on the appropriate test sets, additional testing of the models was done on the test sets generated in the same way as SV, SC, SCV training sets but from the corresponding test sets.

Table 5. Results of YOLOv4 models build with mapping-based transfer learning on appropriate test sets

Model	Test set	AP	AP_{50}	AP_{75}
SV	SV test	29.7	61.7	24.6
SC	SC test	55.8	91.6	61.7
SVC	SVC test	31.7	64.4	27.9

The obtained results of the models obtained with the mapping-based transfer learning tested on the testing part of SV, SC, SCV sets are shown in Table 5 and have worse results than when tested only at the set SARD test set.

The SC model achieved a minor difference in performance on the SC test set, comparing the SARD test set's detection results. This was expected because the Corr-SARD set images included in the SC test set are those from the SARD set only with the added effects of bad weather.

5. Conclusions

In this paper, transfer learning approaches to improve person detection on drone images for the SAR mission were examined. We have fine-tuned the YOLOv4 model using

different transfer learning methods on three datasets: a tailor-made SARD set for SARD missions, a VisDrone drone-recorded dataset in urban places, and a Corr-SARD dataset with synthetically added weather effects on SARD images.

We compared and discussed the impact of the transfer learning methods used in YOLOv4 model training on detection results. Testing was performed on the target dataset SARD and the newly created datasets SV, SC, and SVC, created by merging the initial sets.

The results show that the best detection results are achieved on the target SARD domain using network-based transfer learning when the set on which the model is fine-tuned is equally distributed as the set on which the model is tested. The best results were achieved by applying the network transfer learning method, which transmits features obtained on large data sets, and the instance-based transfer learning method, in which the model is trained on images of the domain corresponding to the images on which the model will be tested. The use of synthetic image instances further improved the performance of the model.

From the results, we also see that the worst results were obtained when the datasets were merged because, in that case, the model could not fully adapt to the data of interest. However, this way, by increasing the learning data, a more general model can be achieved. It has been shown that when training models with multiple datasets, it is not insignificant whether we train with all images simultaneously or individually on each set and the sets' order during training.

For future work, we plan to explore the impact of different transfer learning methods on various application domains and determine the key characteristics of learning datasets that positively impact model performance. Also, we are interested in further exploring different network strategies for selecting, merging, and changing network layers to improve detection results.

Acknowledgment

This research was supported by Croatian Science Foundation under the projects IP-2016-06-8345 “Automatic recognition of actions and activities in multimedia content from the sports domain” (RAASS) and IP-2018-01-7619 “A Knowledge-based Approach to Crowd Analysis in Video Surveillance (KACAVIS) and by the University of Rijeka (project number 18-222).

References

1. Sambolek, S., & Ivasic-Kos, M.: Detection of toy soldiers taken from a bird's perspective using convolutional neural networks. In International Conference on ICT Innovations (pp. 13-26). Springer, Cham. (2019, October).
2. Sambolek, S., & Ivasic-Kos, M.: Person Detection in Drone Imagery. In 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech) pp. 1-6. IEEE. (2020, September).

3. Kristo, M., Ivasic-Kos, M.. & Pobar, M.: Thermal Object Detection in Difficult Weather Conditions Using YOLO, IEEE Access, 2020, pp. 125459-125476
4. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C.: A survey on deep transfer learning. In International conference on artificial neural networks (pp. 270-279). Springer, Cham (2018, October).
5. Oquab, M., Bottou, L., Laptev, I., & Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1717-1724. (2014).
6. Buric, M., Pobar, M., Ivasic-Kos, M.: Adapting YOLO network for ball and player detection, 8th International Conference on Pattern Recognition Applications and Methods, 2019, pp. 845-851
7. Ivasic-Kos, M., Kristo, M., & Pobar., M.:Human detection in thermal imaging using YOLO, 5th International Conference on Computer and Technology Applications 2019, p. 20-24
8. Wang, T., Huan, J., & Zhu, M.: Instance-based deep transfer learning. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 367-375). IEEE (2019, January).
9. Pobar, M., & Ivasic-Kos, M.: Active Player Detection in Handball Scenes Based on Activity Measures, Sensors, 20 (5), 2020, pp. 1475.
10. Cheng, C., Zhou, B., Ma, G., Wu, D., & Yuan, Y.: Wasserstein distance based deep adversarial transfer learning for intelligent fault diagnosis. arXiv preprint arXiv:1903.06753 (2019).
11. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., & Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014).
12. Long, M., Cao, Y., Wang, J., & Jordan, M.: Learning transferable features with deep adaptation networks. In International conference on machine learning (pp. 97-105). PMLR (2015, June).
13. Long, M., Zhu, H., Wang, J., & Jordan, M. I.: Deep transfer learning with joint adaptation networks. In International conference on machine learning (pp. 2208-2217). PMLR (2017, July).
14. Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A.: A kernel two-sample test. The Journal of Machine Learning Research, 13(1), 723-773 (2012).
15. Arjovsky, M., & Chintala, S.: Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 7 (2017).
16. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y.: Generative adversarial networks. arXiv preprint arXiv:1406.2661. (2014).
17. Buric, M., Paulin, G., Ivasic-Kos, M.: Object Detection Using Synthesized Data, ICT Innovations 2019, Web Proceedings, 2019.
18. Zhu, P., Wen, L., Bian, X., Ling, H., & Hu, Q. Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437. (2018).
19. Sambolek, S., & Ivasic-Kos, M.: Detecting objects in drone imagery: a brief overview of recent progress. In 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO). pp. 1052-1057. IEEE. (2020).
20. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. (2020).
21. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H.: CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 390-391. (2020).

22. He, K., Zhang, X., Ren, S., & Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916. (2015).
23. Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J.: Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8759-8768. (2018).
24. Redmon, J., & Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. (2018).
25. Darknet, https://github.com/AlexeyAB/darknet, last accessed 2021/02/21.
26. Google Colab, https://colab.research.google.com/, last accessed 2021/02/21.

27. Padilla, R., Netto, S. L., & da Silva, E. A.: A survey on performance metrics for object-detection algorithms. In 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). pp. 237-242. IEEE. (2020, July).
28. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L.: Microsoft coco: Common objects in context. In European conference on computer vision pp. 740-755. Springer, Cham. (2014, September).