https://doi.org/10.2352/EI.2024.36.8.IMAGE-240

Motion-based domain randomization for detecting honey bees inside a hive

Chih-Hsing Ho[1], Izaak R. Gilchrist[2], Keirstyn A Amponsah[2], Amanda Stoltz[3], Brock A. Harpur[2], Steve Cantley[3], Amy R. Reibman[1]
[1] Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, U.S.
[2] Department of Entomology, Purdue University, West Lafayette, U.S.
[3] BroodMinder, Stoughton, Wisconsin, U.S.

Abstract

In this paper, we address the task of detecting honey bees inside a beehive using computer vision with the goal of monitoring their activity. Conventionally, beekeepers monitor the activities of honey bees by watching colony entrances or by opening their colonies and examining bee movement and behavior during inspections. However, these methods either miss important information or alter honey bee behavior. Therefore, we installed simple cameras and IR lighting into honey bee colonies for a proof of concept study whether deep-learning techniques could assist in-hive observations. However, the lighting conditions across different beehives are diverse, which leads to varied appearances of both the beehive backgrounds and the honey bees. This phenomenon significantly degrades the performance of detection using Deep Neural Networks. In this paper, we propose to apply domain randomization based on motion to train honey bee detectors for inside the beehive. Our experiments were conducted on the images captured from beehives both seen and unseen during training. The results show that our proposed method boosts the performance of honey bee detection, especially for small bees which are more likely to be affected by the lighting conditions.[1]

Keywords: Object Detection, Honey Bee Detection, Domain Randomization.

Introduction

Beekeeping contributes over $15 billion to the United States economy, through both honey production and plant pollination [3]. Multiple threats can kill a honey bee colony quickly [20]. Therefore, regular monitoring of hives and their activity levels is essential. Hives can be monitored at their entry [26, 31], by opening the hive [15], or by using sensors in the hive. In this paper, we consider a camera system that is installed inside the beehive, combined with automated image processing, to monitor honey bees in their natural environment.

Previous approaches to image and video analysis for honey bees have considered honey bee tracking [2, 14], but these have used a specially designed observation hive to expose the interior of the hive. Additional approaches have considered honey bee re-identification [4] and detecting Varroa mites on honey bees, each by observing honey bees entering or exiting the hive. A hive architecture that was optimized to make video recordings was designed in [27] to detect specific behaviors.

In this paper, we explore automated honey bee detection, deep inside any generic, intact hive. We deploy a camera system designed to transform any colony into an observation hive. The inside of a beehive is dark, and honey bees will seal their colonies from any lights they can see. In addition, white (natural) light may excite the honey bees, which could alter the very behavior we are trying to observe. Therefore, our camera system is designed with its own illuminators. We selected an infra-red illuminator that operates at a wavelength of 830 nanometers (nm) for our system, which is notably longer than the upper limit of honey bee vision at 650 nm [12].

While our camera system enables real-time monitoring inside the hive, the infrared illumination creates additional challenges. The hue, contrast, and brightness of all objects are altered relative to their appearances in natural light. Moreover, each level of each beehive has a distinct appearance, and their diversity makes object detection more challenging. Figure 1 shows several collections of images captured from different beehives under 830 nm illumination. Each row of images corresponds to a different level of three different hives. It is clear that each scenario has a distinct background, and that the different strains of honey bees in the second row, relative to the first and third rows, will create additional challenges for a typical machine learning object detector.

One solution to design effective object detectors across diverse environments is to create manual annotations, or labels, for a sufficient amount of data for every scenario. However, data collection and data labeling are laborious and time-consuming. In this paper, we would like to design a machine learning image analytics method that can cope with the diverse environments, without requiring multi-environment annotations. Therefore, we develop and present a method called motion-based domain randomization (MBDR) to apply the principles of domain randomization [5, 29] to real-world non-synthetic data. We demonstrate that this method requires only a single well-annotated dataset, and that it can improve the performance of honey bee detection in environments for which it was not trained.

Related Work

Object Detection

The state-of-the-art object detection methods can be categorized into two areas: one-stage methods that emphasize inference speed, such as YOLO series [1, 21–23] and SSD [18], and two-stage methods that emphasize detection accuracy, such as Fast R-CNN [8], Faster R-CNN [24], and Mask R-CNN [11]. Specifically, one-stage detectors merge the tasks of object localization

[1]This research was supported by a grant from the National Science Foundation STTR Phase I: Automated Continuous Varroa Mite Monitor for Honey Bee Hive Health, Grant #2127468.

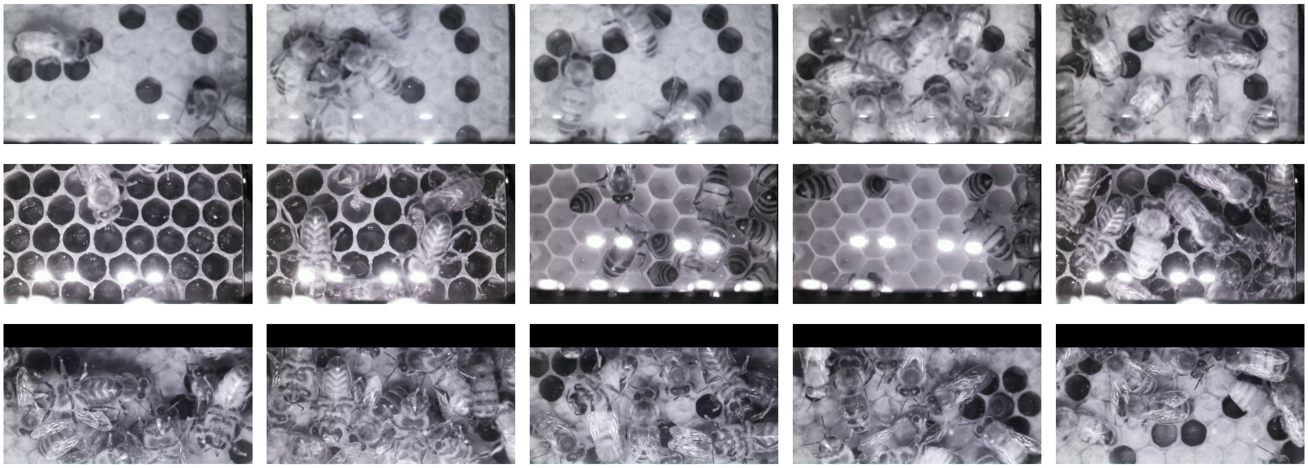

Figure 1: Images collected from various levels of various hives. All the images are collected under illumination with a wavelength of 830 nm wavelength. The appearance of the honey bees and the backgrounds are quite varied. In addition, the honey bees appear both ventrally and dorsally.

and image classification into a regression problem by predicting class probabilities and bounding box coordinates simultaneously. These methods can be further categorized into anchor-based [8, 9, 24] and anchor-free methods [6, 21, 28, 32], which characterize whether there is a predefined set of bounding boxes for the model to adapt to or not. Specifically, anchor-based approaches leverage predefined multiple-sized anchor boxes to detect objects with different scales and aspect ratios, while anchor-free detectors truncate anchor boxes and directly detect the vital keypoints, such as centers and corners of the object [7,21,28]. Recently, [34] points out that one difference between anchor-based and anchor-free detection is the method of defining positive and negative training samples during training. They propose a new method to adaptively select positive and negative samples according to statistical characteristics of objects. However, the performance of one-stage object detectors is lower than the two-stage object detectors. Therefore, we apply a classical two-stage detector Faster-RCNN as our object detector.

Pre-training in AI

The desired approach is to train a neural network using data that is statistically identical to the data for which we want to do inference. This approach is possible if we can guarantee the distribution of data is identical between the training dataset and the evaluation dataset. However, it is not always plausible to collect a large amount of labeled data for all the environments one might encounter. A popular approach is to pre-train a model using similar (but not identical) data that is readily available, and only fine-tune the network using the specific hard-to-obtain data. However, even with this approach, the fine-tuned model can perform poorly if the evaluation environment is too different from the training data.

Another popular solution to enlarge the training data is data augmentation, which randomly adjusts the contrast, brightness, and size of an image using a predefined amount of randomness. However, we found for our application that the foreground (honey bees) and the background of the captured pictures in the beehive have different amounts of variations in the lighting. Therefore, a straightforward application of data augmentation to the entire image may not be effective.

Domain Shift, Domain Adaptation and Domain Generalization

Deep learning typically imposes a strong, implicit assumption that training data and evaluation data samples are from the same statistical distribution data. However, when they are not from the same distribution, we call this domain shift [19,33]. The source domain refers to the training data, and the evaluation data is refered to as the target domain, or equivalently, as being out-of-distribution (OOD). When domain shift occurs, the performance of the model may be significantly degraded.

There are two mainstream approaches to address domain shift: domain adaptation and domain generalization. Domain adaptation [13,30,36] requires well-annotated source domain data and either labeled or unlabeled target data during the training process. This allows the model to learn the difference between the distribution of the source domain and the target domain. However, since the target domain has to be provided before training, and we are unable to obtain data in advance from all the beehives where we would like to deploy our camera system, this approach cannot be applied to our situation.

Another method is domain generalization [10, 16, 25, 35], which requires multiple related source domains during training. This enables the model to learn the common patterns among the domains, and then generalize effectively to OOD data. Nevertheless, collecting and thoroughly labeling multiple source domains is still time-consuming and laborious. Therefore, we consider another approach, domain randomization, which also aims to generalize well on OOD data.

Domain Randomization

Domain Randomization (DR) creates training data that expands the variability of the distribution space [5, 29]. Its goal is to expose a machine learning model during training to many dif-

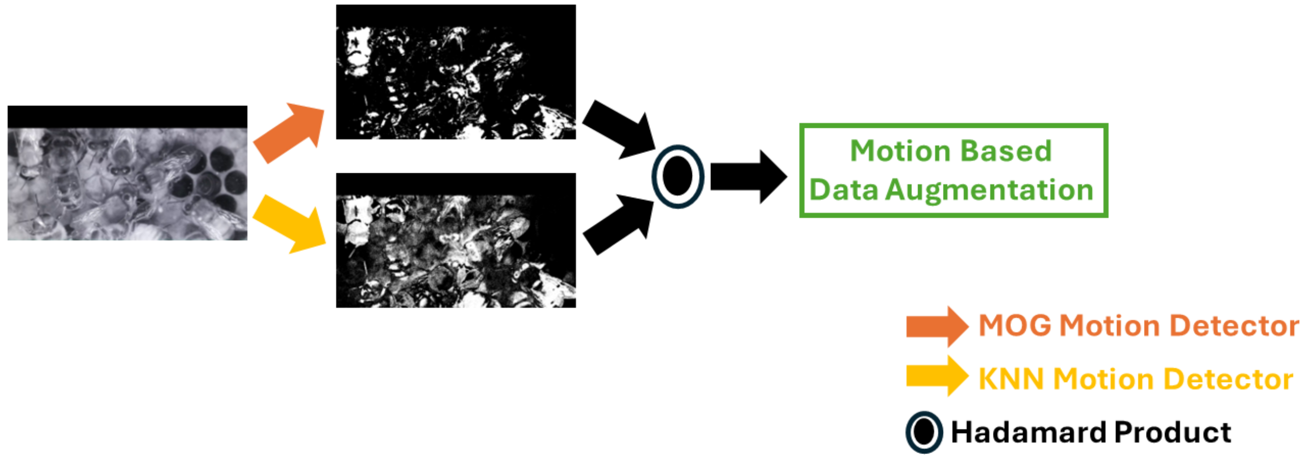

Figure 2: An overview of our proposed method to create training data, motion based domain randomization (MBDR).

ferent data domains. Ideally, DR exposes the model to so many domains in the set of possible distributions, that the model learns to become robust to any unseen distribution, because any new domain it might encounter is just one more domain. As such, training data need not be limited to the accessible physical environment.

In the reinforcement learning and robotics communities, domain randomization approaches have become popular because the complexity of real-world environments makes it difficult to collect data from every environment. Instead, data is synthetically generated separately for the target object, distractor objects, and the environment and its lighting. The shape, texture, position, and orientation of objects are varied, as is the lighting of the environment. However, these variations must be generated using simulators [5, 29]. This requires that simulators for the desired environment already exist; unfortunately, as yet there are no simulators for honey bees or their beehives. To circumvent the disadvantages of leveraging expensive simulators and also to avoid the reality gap, we propose a new method called motion based domain randomization (MBDR).

Proposed Method

In this section, we describe our proposed method to detect honey bees using motion-based domain randomization (MBDR). We are motivated by the observation that the appearance of the honey bees in different domains are more similar to each other than is the appearance of the backgrounds. Therefore, to train a honey bee detector that is effective across different beehives, we would like to generate training images with additional variability in the backgrounds relative to the honey bees themselves. We describe our proposed MBDR method, and then describe our honey bee detector using MBDR.

Motion-based domain randomization (MBDR)

We apply domain randomization to our real-world situation by using motion to segment each image into foreground objects and background. With this segmentation, it is now possible to apply different random augmentations to the objects and to the backgrounds. This has the advantage of only requiring a single well-annotated dataset so that our model can generalize well to unseen domains. In particular, we can supplement the existing data without the need to create a simulator for either the target objects, or for the background and its lighting.

To segment the video into objects and background, we apply two foreground-background segmentation algorithms: the mixture of Gaussians (MOG) [37] and the k-nearest neighbor algorithms (KNN). The MOG motion detector classifies the foreground and the background based on the mixture of 3 Gaussian distribution models of each background pixel. The weights for each Gaussian model are proportional to the duration that the color stays on the pixel. Therefore, if the weights of the Gaussian models for a given pixel are low, the pixel is categorized as foreground. The KNN motion detector computes updated weights of the Gaussian mixture model iteratively by computing the Euclidean distance of each element in the segmentation map; it then selects the pixels that belong to each foreground.

Fig. 2 illustrates the overall pipeline of our proposed method. The video is processed by both background segmentation algorithms. Because the motion maps from each are rather noisy, we apply a Hadamard (pixel-wise) product to the two binary motion maps to obtain a more accurate segmentation. Thus, we decide a pixel belongs to the foreground only if each algorithm estimates it to be a foreground pixel. Synthetic training data is then produced by applying different amounts of random augmentations to the identified foreground and background. In particular, stronger augmentations are applied to the background than to the foreground.

Bee detection using MBDR

We create an image-based honey bee detector using Faster-RCNN, where the training is supplemented by MBDR. Weaker random augmentations are applied to the honey bees, and stronger augmentation is applied to the background. The detector uses ResNet-50 as its backbone feature extractor. The loss function for training is a combination of a classification loss and a regression loss. Since there are only two classes (honey bee and background) in our setting, the classification loss is the binary cross entropy loss, L_{bce}, which is defined for each sample as

$$L_{bce} = -(y\log(p) + (1-y)\log(1-p)). \quad (1)$$

Here if $y = 1$ indicating the sample is a honey bee, then p denotes the predicted probability this sample is a honey bee; similarly if $y = 0$ indicating background, then p denotes the predicted probability the sample is background. The regression loss computes the mean square error loss L_{reg} of the prediction of the x and y coordinate of the bounding box, and also its width w and height h:

$$L_{reg} = \sum_{i}^{D}(i-\hat{i})^2, D \in \{x,y,w,h\}. \tag{2}$$

Finally, the overall objective loss function L for our training is defined as $L = L_{bce} + L_{reg}$.

Experiment and Results

In this section, we describe our experimental setup, including data collection, training, evaluation metrics, and results.

Parameter Settings

In our experiments, we compare the honey bee detection results of MBDR with the results both without data augmentation and with data augmentation. For the data augmentation approach, we augment the entire image with the same randomness, where brightness, contrast and saturation are between 0.8 to 1.2, and hue is between -0.1 to 0.1. For MBDR, we leverage different randomness on the foreground and the background since we find that the backgrounds are more complex than the honey bees. Therefore, we set a higher randomness for the background with brightness, contrast and saturation between 0.8 to 1.2, and hue between -0.2 to 0.2. For the foreground, we set brightness, contrast and saturation from 0.9 to 1.1, and hue from -0.1 to 0.1.

Datasets

We record videos from two beehives at the Purdue Apiary during August and September 2022. Colonies were kept queenright and maintained in double-deep Langstroth boxes with 10 frames each and a super. We use the data from September for both training and testing, and use the data gathered in August for testing only. Hence, we consider September to be our source domain, and both September and August to be our target domains. In September, video was recorded for 55 hours, and we extracted one frame every 500 frames to obtain 1200 image frames. We also computed the foreground-background segmentation as described above and sampled this at the same rate. Images were also extracted at a similar rate from the August videos, to create a similar number of testing images.

Evaluation Metrics

Object detection is typically evaluated using the Average Precision (AP) metric. AP computes the fraction of predicted bounding boxes that are true; for a bounding box to be true, it must have an Intersection over Union (IoU) that is greater than a given threshold. While a typical IoU threshold is 0.5, here we create a more stringent requirement by setting our IoU threshold to be 0.7 in our experiments. This is motivated by our application. We anticipate that once a honey bee is detected, further processing will be applied to the detected honey bees. With a higher IoU threshold, more of the bounding box pixels will correspond to an actual honey bee.

In addition, we also report AP results based on the size of the detected object. We follow the definitions of the MS COCO dataset [17], which computes AP small, AP medium, and AP large for bounding boxes with area smaller than $32*32$ pixels, from $32*32$ to $96*96$ pixels, and larger than $96*96$ pixels, respectively. Note that for our data, there are no honey bees that are categorized as small; therefore, we only report results for medium and large bounding boxes.

Within-domain results

In this experiment, we trained our honey bee detector using the September training data with three different approaches. The first uses the training data from September with no data augmentation. The second incorporates traditional data augmentation into the training by applying a single strength of randomness to both foreground and background. For the third approach, we apply our proposed MBDR, where stronger data augmentation is applied to the background and weaker to the foreground.

The results are shown in Table 1. First, we see that data augmentation improves performance by only a small fraction. This improvement is slight because data augmentation alone cannot alleviate the wide variations across the different beehives. Next, considering the AP of our proposed MBDR, we see that we obtain solid improvements of 0.0282 and 0.0268 relative to a detector trained both without and with data augmentation, respectively.

Our experiments also analyze the performance of the detector for honey bees of different sizes. For medium-sized honey bees, the MBDR approach outperforms the other methods by a significant margin, with improvements of 0.0676 and 0.0572, respectively. For large honey bees, the improvement of the MBDR approach relative to the other methods is 0.0081 and 0.0064. Smaller honey bees are generally more difficult to detect than larger honey bees; however, our proposed method is highly effective at improving the detector for these smaller honey bees even in the within-domain scenario.

Table 1: Honey Bee Detection Results: for September source domain

	No data aug.	With data aug.	MBDR
AP (all)	0.9062	0.9076	0.9344
AP (medium)	0.6949	0.7053	0.7625
AP (large)	0.9563	0.9580	0.9644

Across-domain results

Next, we explore the performance of our method when we evaluate on the target domain, which is to say, when we train on the September dataset and evaluate on the August dataset. Results are shown in Table 2. As can be seen, our proposed method increases performance by greater than 0.0122. For medium-sized honey bees, our proposed method outperforms no data augmentation by 0.0318, which verifies that MBDR can significantly boost the performance of detecting smaller honey bees, identical to our earlier findings. With the experiments conducted in Table 1 and Table 2, since the precision of large honey bees is already reasonably high even without MBDR, the improvement of our method is limited. However, for the more challenging task of detecting smaller honey bees, our method provides significant cross-domain improvements.

	No data aug.	MBDR
AP (all)	0.8756	0.8878
AP (medium)	0.4933	0.5251
AP (large)	0.9585	0.9576

Conclusion

In this paper, we presented a camera system that operates inside a beehive without disruption. Infrared light that is invisible to the honey bees was used for illumination; however, that reduced the contrast of the captured images. Because our system can be placed in any hive, we were able to obtain videos from multiple levels of the hive, and from hives with different strains of honey bees. This created the challenging task of detecting honey bees despite their highly varied appearance and the varied backgrounds.

To this end, we presented a novel method called motion-based domain randomization (MBDR), which designs a detector that performs well across multiple environments. We leverage the motion information in the videos to partition the honey bees from the background, and apply stronger data augmentation to the backgrounds to mimic the greater variations we observe. We demonstrated the method improved performance, particularly for smaller honey bees, when applied to out-of-distribution data.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv: 2004.10934*, 2020.

[2] Katarzyna Bozek, Laetitia Hebert, Yoann Portugal, Alexander S Mikheyev, and Greg J Stephens. Markerless tracking of an entire honey bee colony. *Nature communications*, 12(1):1733, 2021.

[3] Nicholas W Calderone. Insect pollinated crops, insect pollinators and us agriculture: trend analysis of aggregate data for the period 1992–2009. *PloS one*, 7(5):e37235, 2012.

[4] Jeffrey Chan, Hector Carrión, Rémi Mégret, Jose L Agosto-Rivera, and Tugrul Giray. Honeybee re-identification in video: New datasets and impact of self-supervision. In *VISIGRAPP (5: VISAPP)*, pages 517–525, 2022.

[5] Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. Understanding Domain Randomization for Sim-to-real Transfer, 2021.

[6] Yihong Chen, Zheng Zhang, Yue Cao, Liwei Wang, Stephen Lin, and Han Hu. Reppoints v2: Verification meets regression for object detection. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[7] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[8] Ross Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[10] Yunye Gong, Xiao Lin, Yi Yao, Thomas G. Dietterich, Ajay Divakaran, and Melinda Gervasio. Confidence calibration for domain generalization under covariate shift. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[12] N Hempel de Ibarra, Misha Vorobyev, and Randolf Menzel. Mechanisms, functions and ecology of colour vision in the honeybee. *Journal of Comparative Physiology A*, 200:411–433, 2014.

[13] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. MIC: Masked image consistency for context-enhanced domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[14] Panadda Kongsilp, Unchalisa Taetragool, and Orawan Duangphakdee. Individual honey bee tracking in a beehive environment using deep learning and kalman filter. *Scientific Reports*, 14(1):1061, 2024.

[15] Hong Gu Lee, Min-Jee Kim, Su-bae Kim, Sujin Lee, Hoyoung Lee, Jeong Yong Sin, and Changyeun Mo. Identifying an image-processing method for detection of bee mite in honey bee based on keypoint analysis. *Agriculture*, 13(8), 2023.

[16] Yumeng Li, Dan Zhang, Margret Keuper, and Anna Khoreva. Intra-source style augmentation for improved domain generalization. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

[17] 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. In *The European Conference on Computer Vision (ECCV)*, 2014.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot Multibox Detector. In *The European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[19] Willi Menapace, Stéphane Lathuilière, and Elisa Ricci. Learning to cluster under domain shift. In *The European Conference on Computer Vision (ECCV)*, 2020.

[20] Peter Neumann and Norman L Carreck. Honey bee colony losses. *Journal of Apicultural Research*, 49(1):1–6, 2010.

[21] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7263–7271, 2017.

[23] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[24] 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.

[25] Yangjun Ruan, Yann Dubois, and Chris J. Maddison. Optimal representations for covariate shift. In *International Conference on Learning Representations*, 2022.

[26] Stefan Schurischuster, Beatriz Remeseiro, Petia Radeva, and Martin Kampel. A preliminary study of image analysis for parasite detection on honey bees. In *Image Analysis and Recognition: 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018, Proceedings 15*, pages 465–473. Springer, 2018.

[27] Paul Siefert, Rudra Hota, Visvanathan Ramesh, and Bernd Grünewald. Chronic within-hive video recordings detect altered

nursing behaviour and retarded larval development of neonicotinoid treated honey bees. *Scientific reports*, 10(1):8727, 2020.

[28] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[29] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

[30] Thanh-Dat Truong, Ngan Le, Bhiksha Raj, Jackson Cothren, and Khoa Luu. Fredom: Fairness domain adaptation approach to semantic scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[31] Gang Jun Tu, Mikkel Kragh Hansen, Per Kryger, and Peter Ahrendt. Automatic behaviour analysis system for honeybees using computer vision. *Computers and Electronics in Agriculture*, 122:10–18, 2016.

[32] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[33] M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, and C. Finn. Adaptive risk minimization: Learning to adapt to domain shift. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[34] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[35] Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. Towards unsupervised domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[36] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, 2019.

[37] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006.