CROSS-CONNECTED NETWORKS FOR MULTI-TASK LEARNING OF DETECTION AND SEGMENTATION

Rei Kawakami[1] *Ryota Yoshihashi*[1] *Seiichiro Fukuda*[1] *Shaodi You* [2] *Makoto Iida*[1] *Takeshi Naemura*[1]

[1]The University of Tokyo [2]CSIRO-Data61

ABSTRACT

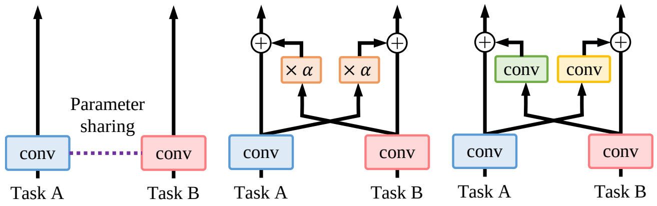
Multi-task learning improves generalization performance in neural networks by sharing knowledge among related tasks. Existing models are for task combinations annotated on the same dataset; research on how to utilize the knowledge of successful single-task convolutional neural networks (CNNs) that are trained on individual datasets is limited. We propose a cross-connected CNN, an architecture that connects single-task CNNs through convolutional layers that transfer useful information to their counterparts. We evaluated the architecture with a combination of detection and segmentation using datasets of two targets: pedestrians and wild birds. Experiments demonstrate how well our CNN learns general representations from multi-task learning.

Index Terms— Multi-task Learning, Pedestrian Detection, Bird Detection, Semantic Segmentation

1. INTRODUCTION

Multi-task learning improves the generality of performance by mutually utilizing information of related tasks [1]. The most common way to achieve this is to share parameters in feature representation layers and branch several top layers for task-wise prediction [2, 3, 4, 5, 6, 7, 8] as illustrated in Fig. 1 (a). However, this architecture can be restrictive because sharing choices, either by hard or soft sharing [9, 10, 11], are discrete, and the number of shared layers are fixed among all tasks. Performance in each task may be harmed because feature representations, particularly in upper layers, must be specialized for each task [12, 13]. A cross-stitch network [14] alleviates this problem with a more general architecture, as shown in Fig. 1 (b). It utilizes element-wise linear combinations of activation maps from each task stream, while retaining individual network parameters. It is, however, limited in that it considers only the combination of channels with corresponding indices.

In this paper, we propose a cross-connected convolutional neural network (CNN), as shown in Fig. 1 (c). We cross-connect intermediate layers of single-task CNNs via convolutional (conv) layers, which learn the importance of each activation map for the other task and determine which information to be sent to which destination. This enables the task-wise streams communicate with each other by exchanging their activation maps. To verify the effectiveness of the cross-connected CNN, we compare its performance with that of several baseline CNNs. We used object detection and semantic segmentation as two tasks to combine, a combination that may benefit from multi-task learning. We particularly investigated multi-task learning from *different* datasets of two tasks to exploit more diverse information sources, while existing studies have only used datasets with annotations for *both* tasks [15, 16]. In the experiments, we tested our proposed network with two types of object that are common to the two tasks: pedestrians and wild birds. Experiments on pedestrian data show the proposed CNN produces better detection performance by leveraging knowledge of segmentation even when the training datasets differ between tasks. In the experiments on pedestrian and wild-bird data, our CNN achieved a higher generalization performance compared to that of the baselines.



(a) Parameter sharing (b) Cross-stitching [14] (c) Cross-connection (ours)

Fig. 1: Comparison of multi-task learning architectures. (a) The layers learn the same representations suitable for both tasks. (b) Two scaling layers model the channel-wise weighted sum of activation maps from conv layers. (c) Two cross-connecting layers model a linear combination of activation maps utilizing all channels.

This paper has the following contributions. (1) A new architecture, the cross-connected CNN, is proposed for multi-task learning. Convolutional layers that cross-connect two single-task CNNs can model cross-channel and cross-layer feature interaction between tasks. The proposed model is a generalization of existing ones. (2) To our knowledge, we make the first attempt to perform multi-task learning of object detection and semantic segmentation using different datasets between tasks.

This work was supported by grants from the Japan Society for the Promotion of Science (JSPS; KAKENHI Grant Number JP18K11348 and Grant-in-Aid for JSPS Fellows JP16J04552).

Related work Although parameter sharing [1] is successful in various tasks [2, 3, 4, 5, 6, 7, 8], the combinations of tasks are based on either of the following two assumptions. First, one task is auxiliary to the other, such as pose estimation and action recognition [8] or facial landmark detection and attribute prediction [6]. Second, one task has insufficient training data and is thus helped by the annotations of the other task, as in depth estimation and surface normal prediction [3]. In all of the studies, multiple tasks have the *same* training data with multiple labels.

Focusing on multi-task learning of object detection and semantic segmentation, MultiNet [15] incorporates three single-task CNNs for classification, detection, and segmentation by parameter sharing. UberNet [17] and BlitzNet [16] aggregate activation maps from middle layers of a single CNN via task-wise skip connections, as similarly done in [18], but because all the task-wise streams use the activation maps from the same single CNN, they are within the classical paradigm of parameter sharing. We differ from them in two respects. First, we construct a multi-task CNN by integrating two single-task CNNs pre-trained on each task. Although consuming more memory due to increased parameters, our CNN can easily reuse existing networks. Second, we use different training datasets between tasks, while the previously discussed CNNs are trained on the same dataset.

Instance segmentation [19, 20, 21, 22] is also a candidate task to combine with object detection, which can distinguish individual object areas of the same class. However, fewer annotations for instance segmentation are available than those for semantic segmentation because more annotation effort is required. Learning from partial annotation [22] can mitigate this labor, but at the cost of segmentation accuracy.

Apart from multi-task learning, late-fusion-based output refinement [23, 24] is promising for simultaneously improving multiple outputs from deep networks for multiple tasks. This type of method is useful for correlated predictions, such as segmentation and optical flow [23], and object and action detection [24]. Those methods differ from the one in this paper in their motivations, as their aim is to improve performance by integrating two correlated outputs, while ours is to improve generalization of feature-level representations.

2. CROSS-CONNECTED CNN

We explain our method by taking two tasks, Task A and B, as examples. As shown in Fig. 2, we start with two single-task CNNs and cross-connect their feature extraction layers via convolutional layers.

Cross-connected layers Cross-connected layers are designed to effectively share knowledge between the combined tasks. The cross-connected layers are represented as a stack of the basic unit, as illustrated in the dotted rectangle in Fig. 2. It shows how the n-th unit receives input maps and passes output maps to the $n+1$-th unit. The unit consists of original conv layers derived from single-task CNNs (drawn in blue and red blocks) and additional conv layers connecting two CNNs (drawn in green and yellow blocks). The connecting conv layers have as many kernels as the number of channels in the output maps of the other stream. The kernel size is chosen to be one-by-one because the connection is for knowledge transfering rather than feature extraction. The connections represent linear transformation and learn the importance of each activation map for the other task. We denote input maps for n-th unit as $\boldsymbol{x}_n^A$ and $\boldsymbol{x}_n^B$ and the transformations learned by the original conv layers and ReLU as f_n^A and f_n^B, respectively. Assuming the cross-connection layers and ReLU learn transformations g_n^A and g_n^B, then $\boldsymbol{x}_{n+1}^A$ and $\boldsymbol{x}_{n+1}^B$ are computed as

$$\begin{cases} \boldsymbol{x}_{n+1}^A = f_n^A(\boldsymbol{x}_n^A) + g_n^A(f_n^B(\boldsymbol{x}_n^B)) \\ \boldsymbol{x}_{n+1}^B = f_n^B(\boldsymbol{x}_n^B) + g_n^B(f_n^A(\boldsymbol{x}_n^A)). \end{cases} \tag{1}$$

Activation maps are added in an element-wise manner. The second terms $g_n^A(f_n^B(\boldsymbol{x}_n^B))$ and $g_n^B(f_n^A(\boldsymbol{x}_n^A))$ have information considered useful for one task on the basis of knowledge obtained in the other task. In the first unit, both $\boldsymbol{x}_1^A$ and $\boldsymbol{x}_1^B$ are equal to an input RGB image.

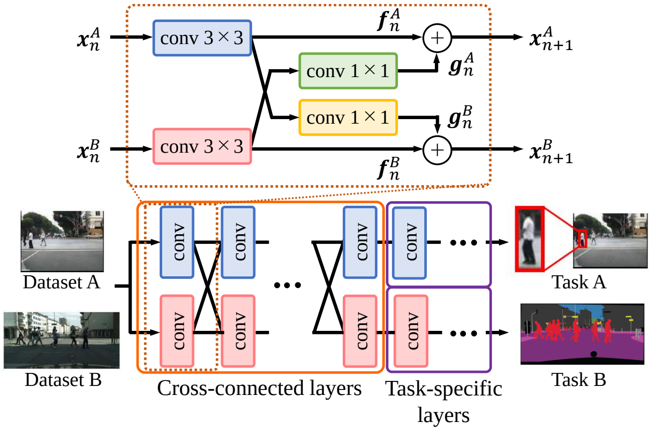


Fig. 2: We integrate two single-task CNNs using cross-connections. Cross-connected layers are common to both tasks, and one-by-one conv layers assign weights on useful information for the other tasks. All conv blocks are composed of a conv layer and an activation function ReLU.

Task-specific layers Task-specific layers are prepared for each task and trained to be specialized on task-wise output generation. For example, object detection branches its network into two paths, which are responsible for bounding box regression and its classification. Semantic segmentation generates a map with pixel-wise labels. These layers are expected to perform different functions between tasks, so their architectures must also be designed differently. Therefore, generally, they cannot be cross-connected due to the difference of the shape of their activation maps. Task-specific layers take feature maps from cross-connected layers and separately process them without communication between tasks.

Training procedure The training of cross-connected CNNs consists of two steps: single- and multi-task learning. For the single-task learning, we first pre-train CNNs for each task independently with individual datasets without cross-connections. By pre-training single-task networks, we can more easily utilize task-specific knowledge of one task for the other task during multi-task learning. Each CNN is trained by minimizing task-specific loss functions: L_A for Task A and L_B for Task B. To cross-connect them, we have to select CNNs with a similar structure.

For the multi-task learning, having pre-trained single-task CNNs, we start to train the cross-connected network. Layers in the network are initialized by weights from pre-trained single-task networks, except for cross-connecting conv layers, which are initialized by random weights. They learn to transform and transfer activation maps of one task to the others after being updated by multi-task training.

As in [17], we use the sum of task-specific losses as a multi-task loss. We denote it as L_{all} that satisfies the following expression: $L_{all} = L_A + \lambda L_B$. All layers in the network are updated in an end-to-end manner w.r.t. the gradient of both loss functions. To enable multi-task learning in different datasets between tasks, we switch training datasets at a constant interval. We compute a loss of only one task and set the loss of the other task that has no annotations to zero. That is, $L_{all} = L_A$ for Task A, and $L_{all} = \lambda L_B$ for Task B.

3. EXPERIMENTAL EVALUATION

To examine the effect of a cross-connected CNN on multi-task learning of object detection and semantic segmentation, we compare its performance with single-task CNNs as well as those of existing models. We show that in these tasks, detection tend to be enhanced by rich contextual information from segmentation. We particularly show that our CNN achieves a higher generalization performance by leveraging more general representations from multi-task learning. Specifically, we present experiments on two domains: pedestrians and wild birds.

Network implementation We first prepare two single-task CNNs, one is specialized for object detection and the other is specialized for semantic segmentation. We use a region proposal network (RPN) [25] based on VGG16 [26] for detection, which has been shown to be effective in pedestrian recognition. As in [25], we use a smooth $\ell 1$ loss for bounding box regression and a cross-entropy loss for its classification.

For semantic segmentation, we construct a VGG16-based pyramid scene parsing network (PSPNet) [27] by combining the convolutional layers from VGG16 and a pyramid pooling module [27]. Although the original PSPNet is based on ResNet, we use VGG16-based ones to clarify the effect of multi-task learning. We use a cross-entropy loss summed over all pixels. Those two single-task CNNs are also used as baselines for comparison.

Having two single-task CNNs, we construct the cross-connected CNN. We cross-connect the first 10 conv layers (conv1_1–conv4_3) and assign the rest as task-specific layers. We set λ as 1.0, following related studies [15, 16]. We fine-tune the cross-connected network using the training procedure in Sec. 2. Because our architecture has more parameters than the single-task CNNs, performance improvement may be natural. To clarify the improvement gained from multiple datasets from that gained by having more parameters, we also fine-tune cross-connected CNNs only with a single dataset as a baseline. These baselines are denoted as 'single-task cross-connected,' and they are fine-tuned only with a detection dataset when evaluating detection and with a segmentation dataset when evaluating segmentation.

We implement two types of multi-task CNNs as the multi-task baselines to compare with ours: hard parameter sharing networks, and a cross-stitch network [14]. Both of them are applied to VGG16-based RPN and PSPNet and trained in the procedure in Sec. 2. For parameter sharing, we test four types of CNNs. Each of these four CNNs shares layers up to the first (2 layers), second (4 layers), third (7 layers), and forth pooling layer (10 layers), respectively. We refer to them as Share 1, Share 2, Share 3, and Share 4, respectively, on the basis of the index of the top pooling layer. A cross-stitch network, denoted as cross-stitch, is implemented by replacing cross-connected layers with scale layers, which learn channel-wise multiplicative scaling factors.

Evaluation metrics We use log-average miss rate (MR) on false positives per image (FPPI) within a defined range to evaluate detection and use intersection over union (IoU) to evaluate the segmentation accuracy of target areas.

Pedestrian detection and segmentation To evaluate the CNNs, we first selected pedestrians because of their frequency in public datasets. We used the Caltech Pedestrian dataset [28], a collection of videos of urban road scenes taken with VGA resolution, for detection. Following [28], we used 42,782 images (set 00–set 05) for training and 4,024 images (set 06–set 10) for testing, among which we used the reasonable subset that includes those taller than 50 pixels and $\geq$ 65% visible. The test set includes 1,802 pedestrians in total. MR was calculated on FPPI in $[10^{-2}, 10^{0}]$ after filtering with non-maximum suppression (NMS) with a threshold of 0.7.

The Cityscapes dataset [29] was used for semantic segmentation. It consists of 5,000 images with 2048-by-1024 pixels. We used 2,975 images for training as in [29], and 500 validation images for testing because official test sets are not disclosed. We focus on person and rider as target classes among 19-class labels assigned to each pixel. We integrated the two labels for label consistency with the Caltech. Details of the training are provided in the supplementary material.

The KITTI [30] dataset was used to evaluate generalization of the detector trained on the Caltech. It has 7,481 training images with 1224-by-370 pixels. Since ground truths of the test set are concealed, we used the training set for testing.

Table 1 shows the evaluation results for pedestrians. The cross-connected CNN achieves an MR of 19.38% on the Cal-

Table 1: Results of detection (MR) and segmentation (IoU) of pedestrians on the Caltech, Cityscapes, and KITTI.

	Caltech	City	KITTI
	MR	IoU	MR
Single-task (RPN)	21.47	-	55.29
Single-task (VGG16-PSPNet)	-	**76.68**	-
Single-task cross-connected (Det)	22.69	-	60.25
Single-task cross-connected (Seg)	-	76.24	-
Share 1	22.15	75.23	50.85
Share 2	22.40	75.31	N/A
Share 3	23.33	75.47	N/A
Share 4	23.21	75.39	N/A
Cross-stitch	22.37	75.39	52.06
Cross-connected (proposed)	**19.38**	75.33	**48.61**

tech dataset, which is 2.09%-points better than the single-task CNN. Single-task cross-connected networks, while having the same number of parameters, do not surpass single-task CNNs. Hard parameter sharing suffers from less flexibility, and the cross-stitch network fails in utilizing the knowledge because the interaction of activation maps is limited.

The IoU in the Cityscapes was not improved by multi-task learning, though Caltech has nearly 10 times more instances of persons than Cityscapes [29]. Information may be richer in Cityscapes, as it has annotations for 19 classes, while Caltech only has pedestrians. Because the bounding boxes have poor information on region shapes, their contribution to segmentation, especially near region boundaries, is limited.

When evaluated on the KITTI dataset, the proposed cross-connected CNN outperforms 6.68%-points than the single-task CNN, and it achieves the best among the multi-task baselines. Table 1 also shows that the generalization performance is not obvious from the in-domain testing. More results are shown in the supplementary material.

Wild-bird detection and segmentation We also evaluated the CNNs on their wild-bird detection and segmentation performance. We aim to detect birds in landscape images to understand whether our method works for different types of real scenes. Unlike the previous experiments, we trained CNNs on the same dataset between the two tasks and verified the generalization performance of detection with another dataset.

We used three datasets constructed for wide-area surveillance of wild birds [31, 32, 33]. Two of them were collected in the Kinki region in Japan and were used for the training and testing of object detection [31] and semantic segmentation [32]. The third is a dataset collected in Tomamae, Hokkaido, and was used for checking the generalization performance of object detection [33]. The Kinki dataset consists of 32,445 landscape images with 5616-by-3744 pixels taken under fine weather. We used only the right half of them (2808-by-3744 pixels), which shows the surroundings of a wind turbine as in [32]. We selected 138 images that have ground truths both for detection and segmentation, where each pixel is annotated into 4 classes: bird, forest, sky, and wind turbine. We used 40 images for training and 98 images for testing, which include 46 and 113 birds taller than 15 pixels, respectively. Due to their sparse distribution, we set the NMS threshold to 0.1. MR was calculated on FPPI in $[10^{-2}, 10^{2}]$.

Table 2: Results of detection (MR) and segmentation (IoU) for birds in the Kinki and Tomamae datasets.

	Kinki		Tomamae
	MR	IoU	MR
Single-task (RPN)	18.59	-	41.35
Single-task (VGG16-PSPNet)	-	33.89	-
Single-task cross-connected (Det)	21.44	-	34.13
Single-task cross-connected (Seg)	-	34.65	-
Share 1	17.74	34.07	44.06
Share 2	18.57	34.39	42.02
Share 3	**16.95**	34.78	39.56
Share 4	24.45	31.74	41.34
Cross-stitch	18.54	34.01	40.84
Cross-connected (proposed)	18.88	**35.45**	**30.36**

The Tomamae dataset consists of 2,222 images with 3840-by-2160 pixels capturing landscapes under bad weather. It is a more challenging dataset due to the images' complex backgrounds. We selected 980 images where relatively more birds appear and used all of them for evaluating the detection performance. 615 birds taller than 15 pixels were included in the test. Evaluation conditions were the same as those in the Kinki dataset. Training details are provided in the supplementary material.

Table 2 shows the evaluation results for the birds. When tested in the Tomamae dataset, the cross-connected CNN achieved a 10.99% better performance than that of the single-task CNN and a 9.2% better performance than that of Share 3, the best of the multi-task baselines. Our CNN obtains a higher generalization performance than parameter sharing and cross-stitching. We do not consider the performance difference for detection in Kinki to be significant, as the total number of birds tested is not statistically large.

4. CONCLUSION

We have proposed a cross-connected CNN, a multi-task CNN consisting of two inter-connected single-task CNNs. In our architecture, two single-task streams pass their activation maps to each other via cross-connecting convolutional layers. These layers enable activation maps to interact across their channels and learn how to utilize the knowledge obtained by task-wise pre-training. We evaluated the CNNs using a combination of object detection and semantic segmentation. Experiments were conducted on two targets, pedestrians and wild birds. In pedestrian detection and segmentation, we demonstrated that our CNN outperforms baselines in detection performance and leverages knowledge of segmentation. In experiments with wild-birds and pedestrians, we demonstrated that our CNN acquires more general knowledge applicable to another dataset than has previously been possible. Future work will focus on more flexible feature re-usage using dense cross connections and its application to other combinations of tasks.

5. REFERENCES

[1] Rich Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 41–75, 1997.

[2] Benjamin Bischke, Patrick Helber, Joachim Folz, Damian Borth, and Andreas Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," *arXiv preprint arXiv:1709.05932*, 2017.

[3] David Eigen and Rob Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *ICCV*, 2015, pp. 2650–2658.

[4] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Pedestrian detection aided by deep learning semantic tasks," in *CVPR*, 2015, pp. 5079–5087.

[5] Yueying Kao, Ran He, and Kaiqi Huang, "Deep aesthetic quality assessment with semantic information," *TIP*, 2017.

[6] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang, "Facial landmark detection by deep multi-task learning," in *ECCV*, 2014, pp. 94–108.

[7] Alex Kendall, Yarin Gal, and Roberto Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018.

[8] Georgia Gkioxari, Bharath Hariharan, Ross Girshick, and Jitendra Malik, "R-cnns for pose estimation and action detection," *arXiv preprint arXiv:1406.5212*, 2014.

[9] Sebastian Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[10] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook, "Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser," in *ACL-IJCNLP*, 2015.

[11] Yongxin Yang and Timothy M Hospedales, "Trace norm regularised deep multi-task learning," in *Workshop track - ICLR*, 2017.

[12] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[13] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in *NIPS*, 2014, pp. 3320–3328.

[14] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert, "Cross-stitch networks for multi-task learning," in *CVPR*, 2016, pp. 3994–4003.

[15] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *Intelligent Vehicles Symposium*, 2018, pp. 1013–1020.

[16] Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid, "Blitznet: A real-time deep network for scene understanding," in *ICCV*, 2017.

[17] I. Kokkinos, "Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *CVPR*, 2017.

[18] Carl Doersch and Andrew Zisserman, "Multi-task self-supervised visual learning," in *ICCV*, 2017.

[19] Jifeng Dai, Kaiming He, and Jian Sun, "Convolutional feature masking for joint object and stuff segmentation," in *CVPR*, 2015, pp. 3992–4000.

[20] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár, "Learning to segment object candidates," in *NIPS*, 2015, pp. 1990–1998.

[21] Jifeng Dai, Kaiming He, and Jian Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *CVPR*, 2016, pp. 3150–3158.

[22] Ronghang Hu, Piotr Dollr, Kaiming He, Trevor Darrell, and Ross Girshick, "Learning to segment every thing," in *CVPR*, 2018.

[23] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *ICCV*, 2017.

[24] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid, "Joint learning of object and action detectors," in *ICCV*, 2017.

[25] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He, "Is faster r-cnn doing well for pedestrian detection?," in *ECCV*, 2016, pp. 443–457.

[26] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[27] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, "Pyramid scene parsing network," in *CVPR*, 2017.

[28] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona, "Pedestrian detection: An evaluation of the state of the art," *TPAMI*, vol. 34, no. 4, pp. 743–761, 2012.

[29] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016, pp. 3213–3223.

[30] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[31] Ryota Yoshihashi, Rei Kawakami, Makoto Iida, and Takeshi Naemura, "Construction of a bird image dataset for ecological investigations," in *ICIP*, 2015, pp. 4248–4252.

[32] Akito Takeki, Tu Tuan Trinh, Ryota Yoshihashi, Rei Kawakami, Makoto Iida, and Takeshi Naemura, "Combining deep features for object detection at various scales: finding small birds in landscape images," *IPSJ Trans. CVA*, vol. 8, no. 1, pp. 5, 2016.

[33] Tuan Tu Trinh, Ryota Yoshihashi, Rei Kawakami, Makoto Iida, and Takeshi Naemura, "Bird detection near wind turbines from high-resolution video using lstm networks," in *World Wind Energy Conference*, 2016.