Fine-Tuning for One-Look Regression Vehicle Counting in Low-Shot Aerial Datasets

Aneesh Rangnekar[1(✉)], Yi Yao[2], Matthew Hoffman[1], and Ajay Divakaran[2]

[1] Rochester Institute of Technology, Rochester, NY, USA
aneesh.rangnekar@mail.rit.edu
[2] SRI International, Princeton, NJ, USA

Abstract. We investigate the task of entity counting in overhead imagery from the perspective of re-purposing representations learned from ground imagery, e.g., ImageNet, via feature adaptation. We explore two directions of feature adaptation and analyze their performances using two popular aerial datasets for vehicle counting: PUCPR+ and CARPK. First, we explore proxy self-supervision tasks such as RotNet, jigsaw, and image inpainting to re-fine the pretrained representation. Second, we insert additional network layers to adaptively select suitable features (e.g., squeeze and excitation blocks) or impose desired properties (e.g., using active rotating filters for rotation invariance). Our experimental results show that different adaptations produce different amounts of performance improvements depending on data characteristics. Overall, we achieve a mean absolute error (MAE) of 3.71 and 5.93 on the PUCPR+ and CARPK datasets, respectively, outperforming the previous state of the art: MAEs of 5.24 for PUCPR+ and 7.48 for CARPK.

Keywords: Proxy self-supervision · Low-shot aerial dataset · Vehicle counting

1 Introduction

Tremendous progress has been made in the last few years with respect to aerial scene representation learning - from datasets (DeepGlobe [7], xView [19], DOTA [37], SkyScapes [4]) to better network architectures (RA-FCN [25], ROI transformer [9], SCRDet [38]). Most of these approaches are developed for either object detection or semantic segmentation. [1,2,22]. Recently there has been developing interest in entity counting via regression, as opposed to commonly used via detection, is highly motivational as it requires comparatively fewer parameters while achieving similar, if not better, accuracy, especially for crowded scenes. However, regression-based entity counting has been explored mostly using ground imagery [20,22,24,28]. In this paper, we focus on entity (e.g., vehicle) counting from overhead imagery without relying on any localization information.

A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12668, pp. 5–18, 2021.
https://doi.org/10.1007/978-3-030-68793-9_1

More specifically, we try to answer the question: *what can we do to improve feature representations pretrained on ground imagery, e.g., ImageNet, for aerial vehicle counting?* In the same line of Aich and Stavness [1], we start by fine-tuning a pretrained VGG-16 network [34] on PUCPR+ and CARPK datasets and use it as our baseline. Singh *et al.* showed that a network trained with self-supervised semantic inpainting was able to outperform its ImageNet pretrained counterpart on aerial semantic segmentation by learning domain specific features [35]. Hence, it is possible to use self-supervision for improved feature learning in the aerial domain. However, PUCPR+ and CARPK contain only 100 and 989 images respectively in the training sets making it difficult to perform full-fledged self-supervision from scratch. Since background categories such as vegetation, roads and buildings dominate the content of these datasets, the likelihood of learning vehicle-specific features is lower than that of background categories.

In order to learn meaningful features and cope with vehicle-specific sample scarcity within the datasets, we propose to adapt representations pretrained using ground images under the premise that features learned from ground images capturing color, texture, edges can be reused for entity counting in aerial images. We investigate two alternative approaches for feature adaptation: 1) **proxy self-supervision tasks:** we apply the self-supervision tasks to the ImageNet pretrained VGG-16 network instead of its randomly initialized version (Sect. 3.1) and 2) **network modifications:** we experiment with squeeze and excitation blocks and active rotating filters as means for feature re-calibration (Sect. 3.2). We achieve better performance than the state of the art in regression-based entity counting. Comparing to detection-based approaches, our methods are more promising since they achieve counting while bypassing precise localization, which requires more complex network architecture to support the computation and large amounts of annotations to support the training.

2 Related Work

Vehicle counting has been tackled previously by using various approaches - from object detection [3,5,13,14] to regression and matching [1,2,22]. The former set of approaches involve designing complex networks with extensive hyper-parameter search (for example, the anchor scales, anchor ratios, learning rate), while the latter set are prone to making the training in an orderly fashion for the network to capture the dataset space. Hsieh *et al.* released two datasets captured from a drone - PUCPR+ and CARPK - with bounding box annotations, and used spatially regularized constraints to increase the localization performance [6,14]. Goldman *et al.* proposed the soft-IOU (intersection over union) layer as the third head of the RPN detection alongside object score and coordinates to help resolve densely packed object detections [13]. Amato *et al.* adapted the YoloV3 for aerial detection by jointly training the layers for maximizing the use of ImageNet and dataset-specific layers [3,8,32]. Cai *et al.* proposed the Guided Attention Network (GA-Net) which consists of foreground and background attention blocks, learned explicitly to extract discriminative features

within the imagery. They also propose a new method for data augmentation, to switch between different times of the day using brightness and Perlin noise, which leads to considerable boost in detection performance [5,29].

Aich and Stavness proposed the first approach for one-look regression on the dataset - they combined count regression with heatmap regulation of the network [1] . In their approach, the network is trained on two loss functions - an L1-loss for minimizing the count and a Smooth L1-loss for minimizing the corresponding class activation map with the ground truth object locations placed as Gaussians. We denote the network trained with and without heatmap regulation as VGG-GAP-HR and VGG-GAP in Table 1 respectively. Aich and Stavness further replaced the global average pooling layer at the end of VGG-16's convolutional backbone with a global sum pooling layer to achieve resolution invariance [2]. Lu *et al.* formulated counting as a template matching problem by learning a density map prediction over samples from the ImageNet-VID dataset [22,33]. They minimized the matching between a single snapshot of the object of interest and the whole frame, where the network was trained with weighted L2 loss on the output density map with the ground truth locations (similar to [1]). They used domain adapters to shift from the ImageNet-VID dataset to the CARPK dataset for vehicle counting [31].

Self-supervised Learning. has attracted a lot of research interest [18,21,35] in computer vision community as it is able to extract context directly from the design of pretext tasks as the supervisory signals, instead of relying on extensive labeling. Kolesnikov, Zhai and Beyer [18] explored the quality of representations learned from rotation [12], exemplar [11], relative patch locations [10] and jigsaw [26] using ImageNet [8] and Places205 [39] datasets. Singh *et al.* improved the performance of a ResNet-18 network trained from scratch by adding a self-supervised semantic inpainting loss wherein the network is forced to learn overhead-specific features for correctly filing the masked out regions [35]. Liu *et al.* [21] improved the performance of crowd counting networks by leveraging unlabeled data with a ranking loss - given two areas sampled from an image in a concentric manner, the network has to ensure that the count predicted for the smaller area is *smaller* than the count predicted for the larger area.

3 Methodology

We establish a baseline for vehicle counting by removing the last set of convolutional layers from VGG-16 network, pretrained on ImageNet, and retrofitting a single fully connected layer that predicts the final count (VGG-GAP [1]). Our focus, herein, is to *improve* the baseline performance by re-calibrating the features learned using ground imagery towards vehicle counting in satellite imagery. To this end, we describe the two unique approaches we investigated (summarized in Fig. 1). One, a data-driven or indirect scheme, encourages suitable features to be learned via introducing proxy self-supervised training. The other directly selects or imposes suitable feature properties via introducing additional network layers. While self-supervision has been widely studied as an unsupervised

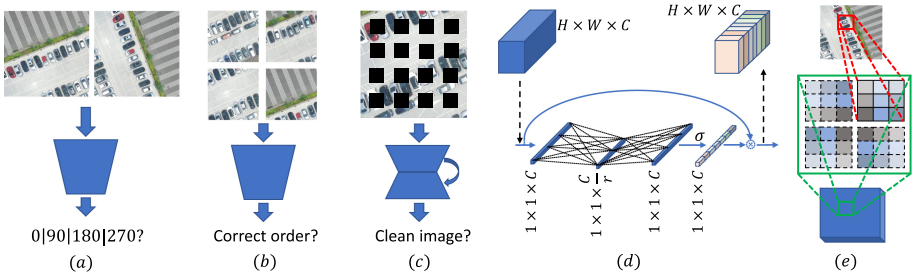

Fig. 1. An overview of all methods experimented in this paper: (a) shows crops with rotation invariance, (b) jigsaw solver, (c) semantic inpainting, (d) squeeze and excitation block, (e) active rotation filters.

representation learning method for various downstream tasks, its application to adapting features from ground imagery to aerial imagery has not been attempted at large, especially for scenarios with sparse annotation. Besides, the effectiveness of indirect and direct adaptation via self-supervision and network modification, respectively, has not been thoroughly investigated and compared in literature. We will address these two issues in this paper.

3.1 Proxy Self-supervision Tasks

Rotation Invariance (RotNet): Proposed by Gidaris *et al.* - the authors create four different copies of a single image by transposing and flipping it and then train a convolutional network to predict the geometric transform applied to the image from its original setting [12]. This helps the network learn informative features and focus on the most salient object in the scene as well as gauge its default appearance. However, we cannot directly apply the task to aerial imagery as there is no *de-facto* default appearance setting - for example, cars can be present with front facing the north or south direction and yet both are plausible settings. Hence, we modify the task and minimize the loss as shown in Fig. 1(a):

$$loss(X_i, \theta) = -\frac{1}{K}\sum_{y=1}^{K} log(F^y(g(X_i|y), X_i|\theta)), \tag{1}$$

where X_i is the sampled image from the dataset and $\{g(\cdot|y)\}_{y=1}^{K}$ applies the geometric transformation with label y to image X_i. $F^y(\cdot)$ and θ indicate the predicted probability distribution over y and the model F's learnable parameters respectively. We convert the problem into a siamese network - where the network receives an image and its rotation version as inputs and is tasked with predicting the rotation used to generate the rotated input. Following [12], we use 0, 90, 180, and 270° as the options for $g(\cdot)$ and discuss the rest of implementation details in Sect. 4.2.

Jigsaw Solver: proposed by Noroozi and Favaro to learn contextual representations by training the convolutional network to solve jigsaw puzzles [26]. This task helps the network to learn discriminative features as it has to find appropriate features that can place the randomly shuffled set of K patches (K = 9 by default) in the correct order. Practically, we implement this approach by minimizing the loss as shown in Fig. 1(b):

$$loss(X_i, \theta) = -\frac{1}{K}\sum_{y=1}^{K} log(F^y(g(X_i|y)|\theta)), \tag{2}$$

where $\{g(\cdot|y)\}_{y=1}^{K}$ splits the image X_i as per the tile configuration $y = (A_1, A_2, \ldots, A_9)$. The original paper used a subset of 1000 permutations based on high Hamming distance, and we use 15 of those combinations in our approach as we did not find using more permutations being a good trade-off in training time and network performance.

Semantic Inpainting: We use the Least squares generative adversarial networks (LS-GAN) with perceptual loss to learn the task of filling in the holes randomly placed within an image (Fig. 1(c)) [16,23]. Our motivation for this task is to encourage learning features that are based on strong contextual and relational information - for examples, if there are pixels to be filled around a red car, how do we make the network aware of what color would be filled in the corresponding pixels? We use a fixed hole grid of 4×4 centered around a mask of 12×12 with replications instead of random masks - unlike other datasets that contain a wide distribution of images, the datasets PUCPR+ and CARPK have a *slightly fixed* area of focus and hence we use fixed masks with extensive image rotations to capture better variance.

3.2 Network Modifications

Squeeze and Excitation Blocks: introduced by Hu *et al.* , these blocks perform feature re-calibration by adaptively weighting each channel of the feature maps (Fig. 1(d)) [15]. We hypothesize not all ImageNet-learned features contribute to aerial imagery, and hence apply SE blocks as channel attention over the features for adaptation. Assuming $\mathbf{v}_{H\times W\times C}$ is the output of a convolutional block where W, H, and C represent the height, weight, and channel, respectively, the squeeze operation applies a global average pooling layer to aggregate the channel-wise responses $\mathbf{z} = \{\mathbf{z}_c\}$ as

$$\mathbf{z}_c = \mathbf{F}_{squeeze}(\mathbf{v}_c) = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{v}_c(i,j), \tag{3}$$

where i, j, c are the indices for height, weight, and channel, respectively. The squeezed representations $\mathbf{z}$ are passed through two fully connected layers parameterized by $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$, $W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ to compute the inter-channel dependencies in the excitation operation as

$$\mathbf{s} = \mathbf{F}_{excite}(\mathbf{z}, W_1, W_2) = \sigma(W_2\delta(W_1\mathbf{z})), \tag{4}$$

where δ, σ and r represent the ReLU non-linearity, the sigmoid activation and the reduction ratio respectively. Finally, the initial features are scaled by the inter-channel weights to obtain the final scaled features as $\widetilde{\mathbf{v}} = \mathbf{s} \cdot \mathbf{v}$. We experimentally insert SE block after the last max pooling layer and before the last convolutional layer choose $r = 2$ to maximize the network prediction performance and minimize changes to the network architecture.

Active Rotating Filters: proposed by Zhou *et al.* and further developed by Wang *et al.* to produce rotation-invariant filters [36,40]. Active Rotating Filters (ARF) generate feature maps with orientation channels - during the convolution, each filter rotates internally and produces feature maps to capture the receptive field layout from K different orientations (for example, $K = 4 \rightarrow 0$, 90, 180, and 270° - Fig. 1(e)). This improves the generalization capacity of the network by learning for orientations that have not been seen before with significantly less need for data augmentation and hence, ARFs are a naturally viable candidate for aerial imagery where objects do not follow a default orientation. To assimilate all the gathered orientation information, Zhou *et al.* proposed ORAlign which calculates the dominant orientation and assigns the features in its favor [40]. Wang *et al.* developed it further into S-ORAlign with concepts from SE blocks and fixing the backpropogation to work with constant learning rate [36]. Experimentally, we adopt ARFs with S-ORAlign in our approach for feature adaptation by imposing the desired orientation invariance for improved performance in vehicle counting.

4 Experiments and Results

4.1 Datasets

The PUCPR+ dataset contains images captured from an altitude at a slanted view of a parking lot. It is a subset of the PUCPR dataset [6] and has images under different weather conditions including sunny, cloudy and overcast. This dataset contains 100 training images and 25 test images. The number of car instances varies from zero to 331 in the training set and from one to 328 in the testing set. The CARPK dataset was released along with the PUCPR+ dataset in [14]. It is the first large-scale aerial dataset for vehicle counting in parking lots under diverse location and weather conditions. This dataset contains 989 training images and 459 test images. The number of car instances varies from one to 87 in the training set and from two to 188 in the testing set. CARPK differs from PUCPR in two ways - 1) it has a diverse location setting compared to images in PUCPR overlooking the same region at all times and 2) it has a more complex count distribution. The images in both datasets are at 720×1280 resolution.

4.2 Experimental Settings

We use Pytorch for evaluating all proposed approaches on the PUCPR+ and CARPK datasets [14,27]. We drop the last set of convolutional layers from the VGG-16 network following previous works [1,14] with the presumption to have just enough downsampling to perceive all vehicles in the scene at the last feature map.

We downsample the images by a factor of 2: $720 \times 1280 \rightarrow 360 \times 640$ for all experiments, since we observe negligible performance difference between these two resolutions (this is also consistent with the approach adopted by VGG-GAP [1]). We also split 10% of the training set as validation set using stratified sampling so that the error metrics are more informative as compared to random sampling. We use the validation set for hyperparameter search and final model selection across all epochs. We train our networks on the task of count regression for 30 epochs with a learning rate of $1e-4$ and then 20 more epochs at a learning rate of $1e-5$ with a batch size of 16. Unless mentioned otherwise, we use the Adam optimizer [17] in all our experiments. We apply random horizontal flip, random vertical flip, and color jittering to both datasets. In addition, we observe that the orientation of vehicles in CARPK has more variance as compared to PUCPR+. Hence, we add data augmentation in the form of transposing the image to account for more car orientation, which we refer to as transposed augmentation in the following discussion).

For the proxy self-supervision tasks, we sample 10 random patches within [72 $\times$ 72, 90 $\times$ 90] resolution per image. For rotation invariance and jigsaw solver tasks, we train on a batch size of 50 for 30 epochs - we use an initial learning rate of $1e-3$ and drop the learning rate by a factor of 10 at 15th and 23rd epoch. For semantic inpainting, we use an initial learning rate of $2e-4$ for the generator and $2e-5$ for the discriminator. We observed that the discriminator learns at a faster rate and to even the curve, we use stochastic gradient descent (SGD) as the optimizer for the discriminator. We train the networks for 30 epochs after which, we discard the discriminator and use the encoder from the generator for count regression fine-tuning.

5 Evaluation Metrics

We use the Mean Absolute Error (MAE), Root-Mean-Sq. Error (RMSE), %Over-estimate (%OA) and %Under-estimate (%UA) for reporting all results:

$$MAE = \frac{\sum_i |y_i - x_i|}{N}, RMSE = \sqrt{\frac{\sum_i (y_i - x_i)^2}{N}}, \tag{5}$$

$$\%OA = \frac{\sum_i |y_i - x_i| I_{[(y_i - x_i) > 0]}}{\sum_i x_i} \times 100, \%UA = \frac{\sum_i |y_i - x_i| I_{[(y_i - x_i) < 0]}}{\sum_i x_i} \times 100, \tag{6}$$

Table 1. Performance of different methods on PUCPR+ and CARPK datasets. We highlight the best results in each group of methods - detection vs. regression - in bold.

Frameworks	PUCPR+		CARPK	
	MAE	RMSE	MAE	RMSE
Detection				
LPN (1000 proposals) [14]	8.04	12.06	13.72	21.77
YOLOv3 [32]	5.24	7.14	7.92	11.08
Soft-IOU [13]	7.16	12.00	6.77	8.52
GA-Net [5]	3.28	4.96	4.61	6.55
YOLOv3 - Amato [3]	**1.80**	**2.74**	**3.73**	**5.11**
Regression				
VGG-GAP [1]	8.24	11.38	10.33	12.89
VGG-GAP-HR [1]	5.24	6.67	7.88	9.30
Class agnostic counting [22]	–	–	7.48	9.99
Proposed methods	**3.72**	**6.32**	**5.93**	**7.90**

where y_i, x_i are the predicted and actual counts for the image sample i and N is the total number of image samples. Hence, we not only get an overall network performance from Eq. 5, but also get a comparative count of over limit and under limit predictions from Eq. 6. We use MAE as the primary metric of interest throughout our results discussion.

5.1 Results

We discuss the performance of our best performing method in comparison with other published methods in Table 1 and ablation study in Table 2. Our method achieves the best performance among regression-based methods [1]. While we observe about 2–3 increase in MAE and RMSSE with respect to the best performing detection-based method, our method requires only half the computation complexity and can be trained without the need of localization annotation, which is known to be expensive to acquire. This clearly demonstrates the effectiveness of our method for entity counting using aerial datasets with sparse annotation.

Table 2 compares the performance of our methods with different configurations on PUCPR+ and CARPK. We report results with transposed augmentation for CARPK. For PUCPR+, most of the configurations under study, both feature adaptation via self-supervision and feature selection via network modification, produce a better performance than the pretrained baseline. Particularly, RotNet-based self-supervision produces the most improvement followed by semantic inpainting and ARFs, demonstrating the efficacy of representation adaptation. We show activation maps from pretrained baseline and RotNet-trained network in Fig. 2. We observe that the latter version has finer activation

Table 2. Ablation study of all approaches discussed in Sect. 3 on PUCPR+ and CARPK datasets using a baseline VGG-16 ImageNet-pretrained network. We highlight the best results in bold with MAE as the metric of interest.

Task/Blocks VGG-16	PUCPR+				CARPK			
	MAE	RMSE	Over-est (%)	Under-est (%)	MAE	RMSE	Over-est (%)	Under-est (%)
Pretrained	5.84	8.51	2.58	1.15	6.88	9.40	0.85	5.80
Rotation	**3.72**	**6.32**	**1.00**	**1.38**	7.81	10.02	2.10	5.45
Jigsaw	4.56	6.07	2.35	0.56	9.05	11.64	3.35	5.40
Inpainting	4.16	6.33	1.96	0.69	6.31	8.25	2.70	3.40
SE Block	5.76	9.50	1.45	2.22	**5.93**	**7.90**	**1.79**	**3.94**
ARF	4.12	5.84	1.89	0.74	12.38	15.83	0.77	11.99

details as compared to the ImageNet-pretrained network. This is learned via proxy self-supervision tasks without using any localization information.

For more complex scenes in CARPK, we have two key observations if transposed augmentations is not used:

- the pretrained baseline gives an MAE of 11.5 ± 1.4. This demonstrates the simple effectiveness of understanding the training and test data distribution and adjusting with data augmentation. We also observe in Fig. 3 that the activations for vehicles have lower intensities when transposed augmentations are not used, especially in cases where the orientations do not match the training set distribution.
- RotNet and SE blocks gives an MAE of 9.3 ± 1.2 and 10.2 ± 0.7 respectively, thus proving that feature adaptation is essential for aerial imagery adaptation.

However, with transposed augmentation, we notice that only semantic inpainting and SE blocks, which are complementary to transformation-based augmentation can further improve the performance (Table 2). This further validates our hypothesis that not all ImageNet-learned features contribute to complex aerial imagery and feature adaptation is essential for good performance.

Additionally, we also performed an ablation study where we trained the network from scratch with rotation invariance as self-supervised task. The network trained from scratch without any self-supervision or localization information achieves an MAE of 124.75 on PUCPR+. The network with RotNet-based self-supervision achieves an MAE of 17.05. Although this does not match the MAE of 3.72 on a RotNet-based ImageNet-pretrained, self-supervised learning still leads to a significant difference in the MAE performance and hence strengthening the scope for aerial self-supervised learning. Figure 4 shows the comparison of three networks on the first convolutional layer - two of them based on pretrained ImageNet features and the third on self-supervised RotNet. We observe that the values of the pretrained and RotNet-proxy VGG-16 networks are identical for weights and biases. This visually confirms our hypothesis towards feature re-usage between the ground and aerial imagery as the weights appear to be looking for the same early set of features. For the network trained from scratch with RotNet as a self-supervised learning task, it is harder to interpret the

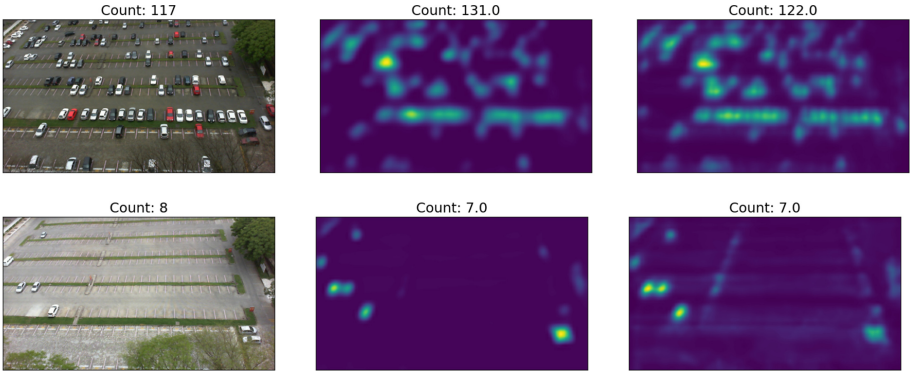

Fig. 2. Exemplar activation maps for images from the PUCPR+ dataset: input image with ground truth count (left), activation maps from pretrained network (middle), and activation maps from the network finetuned with rotation invariance proxy task (right).

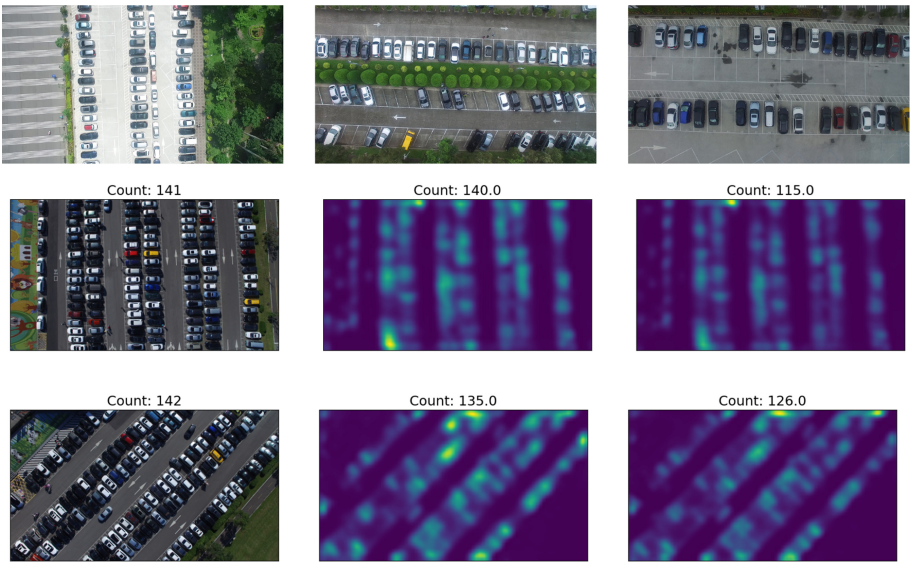

Fig. 3. Exemplar activation maps for images from the CARPK dataset observing the differences based on using orientation-based augmentation. The first row shows images sampled from the training set. Rows 2 and 3 display input image (left), activation maps from network with (middle) and without transposed-image augmentation (right) respectively.

information stored - however, we can still observe that the network is looking for some information towards edges and color given there is not a single weight that is monochromatic.

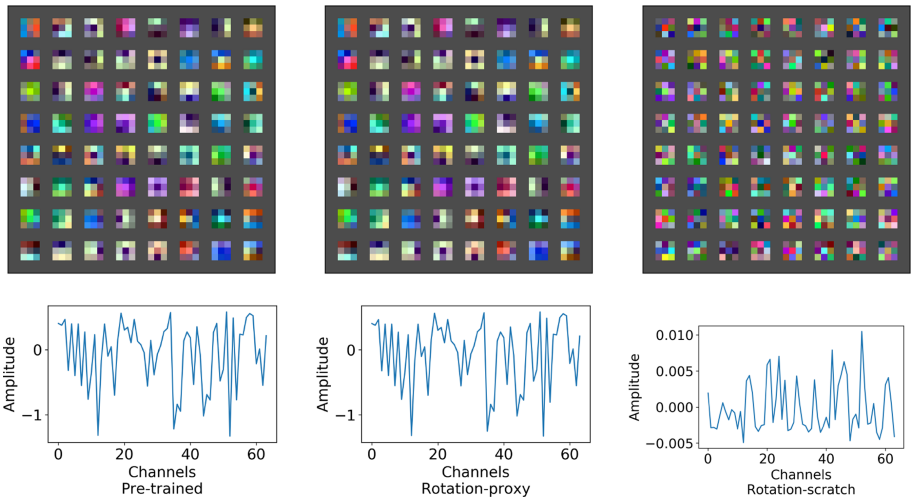

Fig. 4. Comparison between networks trained on PUCPR+ dataset. Top and bottom rows: weights and biases of the first convolutional layer from VGG-16. Left column: pre-trained ImageNet. Middle column: trained with RotNet proxy. Right column: trained with RotNet from scratch.

To further understand where the networks actually differ, we use singular vector canonical correlation analysis (SVCCA) [30] to compare the activations of the two networks on the fixed set of input images from the dataset. SVCCA uses a combination of singular value decomposition and canonical correlation analysis for interpreting similarity within different sets of feature maps without accounting for filter orderings. From Figs. 5, 6, we observe that the activations differ post the second max pooling further strengthening our hypothesis of feature re-usage.

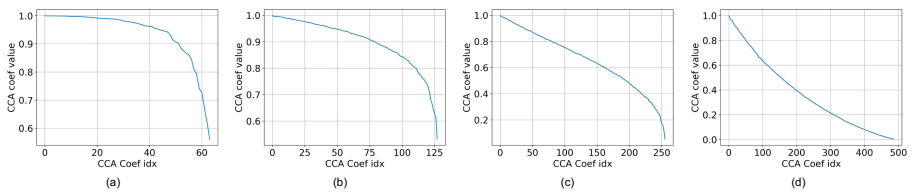

Fig. 5. CCA similarity amongst different layers of VGG-16 comparing the activations of pretrained and RotNet proxy on PUCPR+ dataset. (a), (b), (c) indicate the similarities at the *maxpool* stages and (d) indicates the similarities before the global average pooling layer.

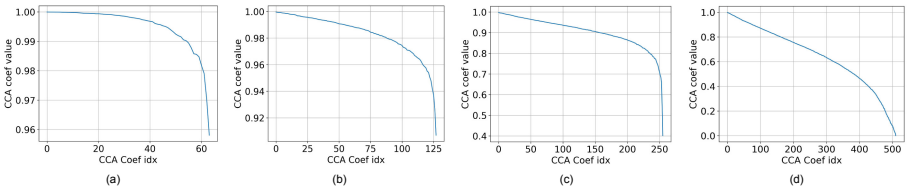

Fig. 6. CCA similarity amongst different layers of VGG-16 comparing the activations with and without transposed-image augmentation on CARPK dataset. (a), (b), (c) indicate the similarities at the *maxpool* stages and (d) indicates the similarities before the global average pooling layer.

6 Conclusion

We study a suite of approaches that help in learning better features for vehicle counting from aerial imagery with small scale datasets. Our study showed that different adaptation approaches induce different amounts of performance improvement depending on data characteristics. With a suitable adaptation scheme, we achieved substantial performance improvement on both PUCPR+ and CARPK datasets.

Acknowledgements. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2018- 18050400004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

1. Aich, S., Stavness, I.: Improving object counting with heatmap regulation. arXiv preprint arXiv:1803.05494 (2018)
2. Aich, S., Stavness, I.: Object counting with small datasets of large images. CoRR abs/1805.11123 (2018). http://arxiv.org/abs/1805.11123
3. Amato, G., Ciampi, L., Falchi, F., Gennaro, C.: Counting vehicles with deep learning in onboard UAV imagery. In: 2019 IEEE Symposium on Computers and Communications (ISCC), pp. 1–6. IEEE (2019)
4. Azimi, S.M., Henry, C., Sommer, L., Schumann, A., Vig, E.: Skyscapes fine-grained semantic understanding of aerial scenes. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
5. Cai, Y., et al.: Guided attention network for object detection and counting on drones. arXiv preprint arXiv:1909.11307 (2019)
6. De Almeida, P.R., Oliveira, L.S., Britto Jr., A.S., Silva Jr., E.J., Koerich, A.L.: Pklot-a robust dataset for parking lot classification. Exp. Syst. Appl **42**(11), 4937–4949 (2015)

7. Demir, I., et al.: Deepglobe 2018: a challenge to parse the earth through satellite images. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 172–17209. IEEE (2018)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR09 (2009)
9. Ding, J., Xue, N., Long, Y., Xia, G.S., Lu, Q.: Learning ROI transformer for oriented object detection in aerial images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2849–2858 (2019)
10. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1422–1430 (2015)
11. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 766–774 (2014)
12. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
13. Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J., Hassner, T.: Precise detection in densely packed scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5227–5236 (2019)
14. Hsieh, M.R., Lin, Y.L., Hsu, W.H.: Drone-based object counting by spatially regularized regional proposal networks. In: The IEEE International Conference on Computer Vision (ICCV). IEEE (2017)
15. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
16. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. arXiv preprint arXiv:1901.09005 (2019)
19. Lam, D., et al.: xview: objects in context in overhead imagery. arXiv preprint arXiv:1802.07856 (2018)
20. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Advances in Neural Information Processing Systems, pp. 1324–1332 (2010)
21. Liu, X., Van De Weijer, J., Bagdanov, A.D.: Leveraging unlabeled data for crowd counting by learning to rank. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7661–7669 (2018)
22. Lu, E., Xie, W., Zisserman, A.: Class-agnostic counting. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) ACCV 2018. LNCS, vol. 11363, pp. 669–684. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20893-6_42
23. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802 (2017)
24. Marsden, M., McGuinness, K., Little, S., Keogh, C.E., O'Connor, N.E.: People, penguins and petri dishes: adapting object counting models to new visual domains and object types without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8070–8079 (2018)

25. Mou, L., Hua, Y., Zhu, X.X.: A relation-augmented fully convolutional network for semantic segmentation in aerial scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12416–12425 (2019)
26. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 69–84. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_5
27. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32, pp. 8024–8035. Curran Associates, Inc. Red Hook (2019)
28. Paul Cohen, J., Boucher, G., Glastonbury, C.A., Lo, H.Z., Bengio, Y.: Count-ception: counting by fully convolutional redundant counting. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 18–26 (2017)
29. Perlin, K.: Improving noise. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, pp. 681–682 (2002)
30. Raghu, M., Gilmer, J., Yosinski, J., Sohl-Dickstein, J.: Svcca: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In: Advances in Neural Information Processing Systems, pp. 6076–6085 (2017)
31. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Efficient parametrization of multi-domain deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8119–8127 (2018)
32. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
33. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
35. Singh, S., et al.: Self-supervised feature learning for semantic segmentation of overhead imagery. In: BMVC (2018)
36. Wang, J., Liu, W., Ma, L., Chen, H., Chen, L.: Iorn: an effective remote sensing image scene classification framework. IEEE Geosci. Remote Sens. Lett. **15**(11), 1695–1699 (2018)
37. Xia, G.S., et al.: Dota: a large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3974–3983 (2018)
38. Yang, X., et al.: Scrdet: towards more robust detection for small, cluttered and rotated objects. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 8232–8241 (2019)
39. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **40**(6), 1452–1464 (2017)
40. Zhou, Y., Ye, Q., Qiu, Q., Jiao, J.: Oriented response networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 519–528 (2017)