Fast and Efficient Model for Real-Time Tiger Detection In The Wild

Orest Kupyn *
Ukrainian Catholic University
Lviv, Ukraine
kupyn@ucu.edu.ua

Dmitry Pranchuk*
WANNABY
Minsk, Belarus
d.pranchuk@gmail.com

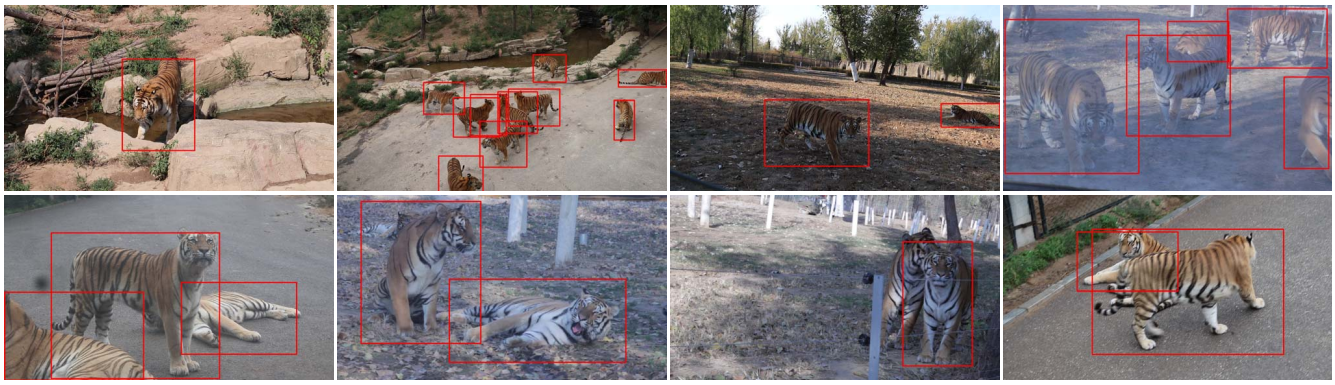

Figure 1: The examples of the predictions on the test set. Our model is able to correctly detect tigers even in complex scenarios and occlusions with a relatively small number of false positives.

Abstract

The highest accuracy object detectors to date are based either on a two-stage approach such as Fast R-CNN [9] or one-stage detectors such as Retina-Net [18] or SSD [19] with deep and complex backbones. In this paper we present TigerNet - simple yet efficient FPN based network architecture for Amur Tiger Detection in the wild. The model has ***600k parameters****, requires* ***0.071 GFLOPs*** *per image and can run on the edge devices (smart cameras) in near real time. In addition, we introduce a two-stage semi-supervised learning via pseudo-labelling learning approach to distill the knowledge from the larger networks [12][2]. For ATRW-ICCV 2019 tiger detection sub-challenge, based on public leaderboard score, our approach shows superior performance in comparison to other methods. The model and the code are available at :* `https://github.com/KupynOrest/AmurTigerCVWC`

1. Introduction

Wildlife conservation is critical for maintaining species biodiversity. Failure to protect endangered species on Earth may lead to imbalanced ecosystems and affect environmental health. This mission is increasingly depended on accurate monitoring of the geospatial distribution and population health of these endangered species, especially in the face of poaching and loss of habitats. Traditional methods of attaching transmitters to wildlife are prone to sensor failure, difficulties with scaling to large populations, and impossibility to measure how the wildlife interacts with its environment. Computer vision techniques are a promising approach to wildlife monitoring, especially with the use of unmanned aerial vehicles or camera traps to collect visual data. In particular, detection and re-identification (re-ID) is a core vision method required to obtain accurate population counts and track wildlife trajectory. However, the deployment of such systems is hampered by several challenges. First, resource constraints on the edge camera require low-power and accurate tiger detection to trigger the image capture and thus avoid that massive irrelevant image capturing consumes space of storage card and battery life. The recent advances of deep learning has led to significant progress in the object detection field. Still, most of the state of the art methods [18][19] use a multi-stage approach or heavy deep neural network architectures that require significant computational resources.

We introduce a new lightweight object detection architecture that meets all the needed requirements for edge devices deploymen with superior efficiency in terms of FLOPs and network size while still achieving comparable accu-

* These two authors contributed equally

2473-9944/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCVW.2019.00041

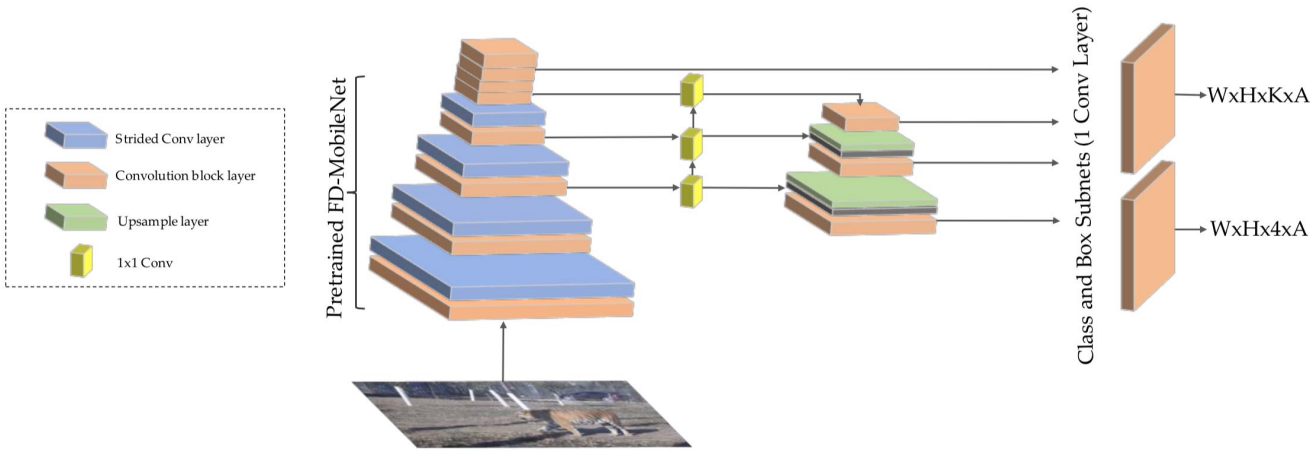

Figure 2: The proposed architecture with the pretrained FD-MobileNet encoder. We use large feature maps at the last network layers together with the slimmer box regression / classification subnets and Deptwise Separable Convolutions to create an efficient model for edge and mobile devices. In contrast to [18] the Class and Box subnets does not share the weight across the feature maps.

racy to deeper state-of-the-art methods. Our innovations are summarized as below

- We present a novel neural network architecture for object detection. The model is based on the Feature Pyramid Network (FPN [17]). To further increase efficiency, we extensively use Depthwise Separable Convolutions (originally introduced in [23] and used in Inception models [13]) and lightweight FD-MobileNet backbone [21]. Further, we modify the original RetinaNet architecture to produce larger feature maps at the last layers that allow us to train the network with high efficiency even on small image sizes that significantly reduce the computational requirements.
- We introduce the two-stage semi-supervised learning approach to distill information from larger deep neural network architectures. We train deep SE-ResNeXt-101 [11] based RetinaNet and use Pseudo-Label [12] approach to distill the knowledge from this network using unlabelled data into small FD-MobileNet model. We show that this approach helps to significantly boost performance and efficiently learn the lightweight model even with a small number of annotated samples.

2. Dataset

The Amur Tiger population currently has less than 600 wild individuals in the world. Capturing enough image data for free-roaming Amur tigers is infeasible as these tigers have an activity range over hundreds of kilometers. Instead, the training data for tiger detection challenge originate from images taken in multiple wild zoos in an unconstrained setting with time-synchronized surveillance cameras and tripod fixed SLR cameras. The dataset includes 4,434 images of high resolution (1920x1080) with 9,496 bounding boxes. To reduce storage, power, and networking consumption the data contains only a subset of frames with at least one Amur Tiger present. The annotations are provided in PASCAL VOC [7] format with bounding boxes assigned to each frame.

3. Model

The goal is to place tight bounding boxes around tigers from images/videos captured by cameras. The solution should be able to run on edge devices so the model should be optimized in terms of FLOPs and size. In this section, we introduce the novel lightweight FPN-like architecture for object detection and a semi-supervised training approach with pseudo-labels.

3.1. Architecture

The architecture we use is a modification of RetinaNet [16]. It generates multiple feature map layers which that different semantics and contain better quality information. FPN comprises a bottom-up and a top-down pathway. The bottom-up pathway is the usual convolutional network for feature extraction, along which the spatial resolution is downsampled, but more semantic context information is extracted and compressed. Through the top-down pathway, FPNs reconstruct higher spatial resolution from the semantically rich layers. The lateral connections between the bottom-up and top-down pathways supplement high-resolution details and help localize objects.

This model is agnostic to the feature extractor backbones choice. We use ImageNet-pretrained networks [6] to improve the generalization of the final model by introducing more general features from the backbone activations. To improve the efficiency of the model we use FD-MobileNet

[21] as the backbone network which is an efficient and accurate network for very limited computational budgets (10-140 MFLOPs) and outperforms original MobileNet [10] both in terms of accuracy and efficiency. To further reduce the complexity, we introduce Deptwise Separable Convolutions [5] instead of regular Convolutional Layer in the FPN. In contrast to the original RetinaNet, we created separate box and regression subnets for each feature map activation and limited the number of the layers in the subnet to 1. This allowed us to additionally decrease the number of FLOPs while stile achieving satisfactory results in terms of mAP.

We also discovered that original RetinaNet or SSD implementations fail to converge on smaller image sizes. As shown in table 2 the RetinaNet-MobileNetv2 baseline achieve the lowest mAP among all the experiments. Due to the Pooling or Stride Convolutional Layer after each feature map, the final activations have too small spatial resolution thus fail to learn good domain-independent features. For default image size = 300 the final activation map output is only 2x2. We removed poolings from the last layers of the model which let us to train and inference on smaller image sizes which in turn significantly improved the efficiency of the model in terms of FLOPs. The output feature maps have spatial resolution *1/8, 1/16, 1/32, 1/32, 1/32, 1/32* of input image size.

Loss Functions: RetinaNet output consists of several feature maps taken from different layers of the FPN network. Category prediction (background or tiger in our case) and bounding boxes coordinates are retrieved from feature maps at the post-processing step. We use separate loss functions for category classification and bounding box coordinates regression. For classification subnet output we use categorical cross-entropy loss:

$$CE = -\sum_{c=1}^{M} y_{o,c} log(p_{o,c})$$

where *M* - number of classes in classification task, *y* - binary indicator (0 or 1) if class label *c* is the correct classification for observation *o*, *p* - predicted probability observation *o* is of class *c*. For negative-positive balancing hard-negative mining strategy was used. According to this strategy, for each positive anchor we select η negative anchors with the greatest error. Classification loss equals to zero for all the rest negative anchors. For the final model we use $\eta = 3$.

For box regression subnet smooth L_1 loss [8] is used:

$$smooth_{L_1} = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

where *x* - residuals between ground truth and prediction values.

3.2. Semi-Supervised Learning using Pseudo-Labels

As in many other domains and applications, the task of this challenge is to create an efficient and lightweight model using a limited number of training samples. Due to the small size of the Amur Tiger population, it is complicated to create a large-scale dataset with various conditions (day time, weather, zones, scales, etc.) Thus, it is crucial to be able to learn from unlabelled data which is usually easier to find and collect/create. We propose to distill the knowledge from the larger networks via pseudo-labelling [12] on unlabelled data. This both helps to increase the generalization of the smaller model and utilize the information from the additional data. In particular, we train a large SE-ResNeXt-101 RetinaNet-like model on the labeled training set which produces 0.61 mAP. Further, we use the raw predictions on the unlabelled part of the dataset on this model as the ground truth labels along with the labeled training set and train the final FD-MobileNet object detection model. As shown in Section 6, this semi-supervised learning approach with knowledge distillation allows to significantly improve the generalization of the smaller model.

4. Experimental Evaluation

4.1. Model Training

We implemented all of our models using PyTorch [1]. We take the random 80% of images as the training set and the left 20% as the validation set to prevent the model overfitting. All models were trained on a single 1080-TI GPU, with Adam [14] optimizer and the learning rate of 10^{-4}. We use the Early Stopping criterion [20] with mAP on Validation set for 15 epochs. The learning rate is reduced by a factor of 2 each time the validation mAP stopped increasing for 5 epochs. The final model takes 3 hours to converge.

Augmentations: Due to the small size of the training dataset and it's sameness hard augmentations were used in the training process. We used vertical and horizontal flips, affine transforms (rotations, shifts, and scales), Gauss noise, different variants of blur (Gauss blur, median blur, motion blur), random rain and random shadow transforms, colors augmentations (gamma, brightness, contrast). All of the listed transforms were taken from *Albumentations* open-source library [4].

Additionally we use custom "tiger cutout" augmentation to increase the model performance in cases with multiple tigers present in the frame. We cut out tigers from any images and inserted them on random places of image. 3 illustrates examples of augmented images. **Post Processing:** To improve bounding box coordinates prediction we replaced the standard suppression algorithm with a blending strategy described in Bazarevsky*et al*. [3]. Also, we implemented test-time augmentations (blending predictions of

Figure 3: Network input after augmentations.

augmented images) but it gave us poor quality improvement and noticeably increased inference time.

4.2. Quantitative Evaluation on ATRW dataset

We compare our model with other participants within the CVWC 2019 challenge on the ATRW dataset [15]. Our model is superior to most of the solutions both in terms of Mean Average Precision and FLOPs. On the final mAP per FLOPs metric described above our solution clearly outperforms others. The result is shown in Table1. We achieve high accuracy with the small number of false positives while still maintaining high efficiency. The model can be further optimized using channel pruning and quantization up to 1Mb in size making possible the deployment on any platform or edge device.

Table 1: The results for CVWC Tiger Detection Challenge. The final metrics are different variants of PPF metric - precision per FLOPs. `https://cvwc2019.github.io/leaderboard.html`

	mAP	GFLOPs	PPF
Our solution	0.515	0.071	**0.630**
dcyhw (FCOS detector)	0.586	5.68	0.467
zdi (Faster RCN + HRN)	0.601	245.3	0.388
lazy-learners (FasterRCNN)	0.546	112.48	0.388
CVWC Team (SSD-v2)	0.476	1.25	0.277

4.3. Ablation Study and Analysis

Table 2: Ablation Study on the ATRW dataset.

	mAP	GFLOPs
SSD-MobileNet (solution baseline)	0.426	1.2
RetinaNet-MobileNetv2	0.33	0.9
+ hard augmentations	0.42	0.9
+ mean NMS	0.43	0.9
+ slimmer subnet and remove pooling	0.511	0.64
+ separable convolutions and FD-MobileNet	0.489	0.071
+ pseudo-labels	**0.515**	**0.071**

We perform an ablation study on the effect of specific components of the pipeline. Starting from the original RetinaNet with MobileNet-v2 [22] backbone, we gradually inject our modifications: adding mean NMS, depthwise separable convolutions, FD-MobileNet backbone, slim subnet, pseudo-labeling, etc. The results are summarized in Table 2. We can see that all our proposed components steadily improve either efficiency (FLOPs) or accuracy (mAP). In particular, the mean NMS module and large feature maps contribute most significantly in terms of accuracy, while DSC and FD-MobileNet significantly improve efficiency.

5. Conclusion

This paper introduces a new efficient neural network architecture and training pipeline for object detection. The introduced model can work in real-time and be deployed on edge/mobile devices while still maintaining high accuracy. The proposed solution achieves superior results on the CVWC challenge and had a potential application to wildlife conservation.

References

[1] PyTorch. `http://pytorch.org`.

[2] Yauhen Babakhin, Artsiom Sanakoyeu, and Hirotoshi Kitamura. Semi-supervised segmentation of salt bodies in seismic images using an ensemble of convolutional neural networks, 2019.

[3] Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. Blazeface: Sub-millisecond neural face detection on mobile gpus, 2019.

[4] Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, and Alexandr A. Kalinin. Albumentations: fast and flexible image augmentations, 2018.

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[7] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.

[8] Ross Girshick. Fast r-cnn, 2015.

[9] Ross B. Girshick. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[10] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv e-prints*, page arXiv:1704.04861, Apr 2017.

[11] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2017.

[12] Dong hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[15] Shuyuan Li, Jianguo Li, Weiyao Lin, and Hanlin Tang. Amur Tiger Re-identification in the Wild. *arXiv e-prints*, page arXiv:1906.05586, Jun 2019.

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[17] Tsung-Yi Lin, Piotr Dollr, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2016.

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollr. Focal loss for dense object detection, 2017.

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. 2015.

[20] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 55–69, London, UK, UK, 1998. Springer-Verlag.

[21] Zheng Qin, Zhaoning Zhang, Xiaotao Chen, and Yuxing Peng. FD-MobileNet: Improved MobileNet with a Fast Downsampling Strategy. *arXiv e-prints*, page arXiv:1802.03750, Feb 2018.

[22] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[23] Laurent SIfre and Stphane Mallat. Rigid-motion scattering for texture classification, 2014.