Comparison of Backbones for Microscopic Object Detection Algorithms

Natthaphon Hongcharoen[1], Parinya Sanguansat[2], and Sanparith Marukatat[3]
[1,2]Faculty of Engineering and Technology, Panyapiwat Institute of Management, Nonthaburi, Thailand
[3]NECTEC, AI Research Group, Pathumthani, Thailand
E-mail: 627210108@stu.pim.ac.th, parinyasan@pim.ac.th, sanparith.marukatat@nectec.or.th

Received: April 4, 22, 2021 / Revised: April 21, 2023/ Accepted: April 22, 2022

***Abstract*—Modern object detection methods are mostly comprised of feature extractor parts and detection parts. With the rise of Vision Transformers and more advanced variants of Convolutional Neural Networks, we present the comparative experimental results of using different feature extractors on the Cascade Faster R-CNN object detection technique. We also prove the significance of using the complete pre-trained weight for the entire object detection model over the slight increase in feature extractor performance but need to randomly initialize all detection layers. The trained models were evaluated using the mean Average Precision (mAP) metric on the unseen laboratory-generated data and also visual evaluation of real-world data from medical diagnoses. The modern Vision Transformer techniques such as PVT and Swin significantly outperformed the traditional Convolutional Neural Network model such as ResNet or ResNeXt with PVT V2 achieved 78% mAP at IOU 0.7 with only the feature extractor pre-trained on ImageNet dataset compared to 60.5% of ResNet 101 and 59.2% of ResNeXt 101-64x4 with similar weight initialization. The results also show a significant increase in the accuracy of using the pre-trained model entirely as a weight initializer in every layer but the final output. ResNet 50 and ResNet 101 achieved 75.6% and 77.2% mAP respectively. A significant improvement over 59.5% and 60.5%. ResNeXt with a pre-trained detector also achieved 78.8% and 79.2% on 64 and 32 cardinality sizes respectively, actually better than PVT V2 with only random weight initialized on the detector part.**

***Index Terms*—Deep Learning, Microscopic Images, Object Detection, Vision Transformer**

I. Introduction

Liver cancer is one of the leading causes of cancer death [1]. Opisthorchis Vivertini or OV (also known as liver fluke) is one of the causes of liver cancer. Liver fluke infection is generally caused by raw fish consumption which is very common in the northeastern region of Thailand. The accumulation of liver fluke will eventually lead to liver cancer if not properly treated.

While it is possible to detect parasite infection by analyzing fecal slides, liver fluke is not the only parasite detectable by this method. Other parasites such as Minute Intestinal Flukes (MIF) has very similar eggs compared to liver fluke as shown in Fig. 1. However, they, infect different organs and cause different health problems.

With the similarity of different parasites detectable by the same method, visually differentiating them would be time-consuming even for experts. With the advances in computer vision research, it is possible to automatically detect and classify the parasite eggs in the digital images of fecal slides.

II. Literature Review

Many modern object detection techniques such as RetinaNet [2], Cascade R-CNN [3], DETR [4], and Yolact [5] rely on the use of large convolutional neural networks as feature extraction part of the detector (also known as backbones). Several convolutional neural networks of various depths such as ResNet or ResNeXt that has been trained on image classification task can be used. The results from several works showed that more accurate backbones made higher accuracy detectors at the cost of the lowered speed.

Recently, new techniques such as Vision Transformer [6] surpassed both ResNet and ResNeXt in the image classification task. Therefore Transformer-based backbones seem to be an interesting approach for object detection as well.

This paper aims to explore the difference in the performance of an object detection technique with different backbone configurations when used in microscopic images of parasite eggs.

III. Data

The dataset was divided into 2 parts, the laboratory-generated images with bounding box annotations for training and comparative evaluation and the real-world images from medical diagnoses but without annotations for visual testing.

There were 3,237 annotated images in total, with 1,684 being MIF and 1,553 being OV. They were split into 2,465 training images and 772 evaluation images.

The real-world image data consist of 1,320 images with some of them duplicated. We test all the images and select a sample of them that showed distinctions between each model.

The examples of the data are shown in Fig. 1 the real-world images are shown in Fig. 2.

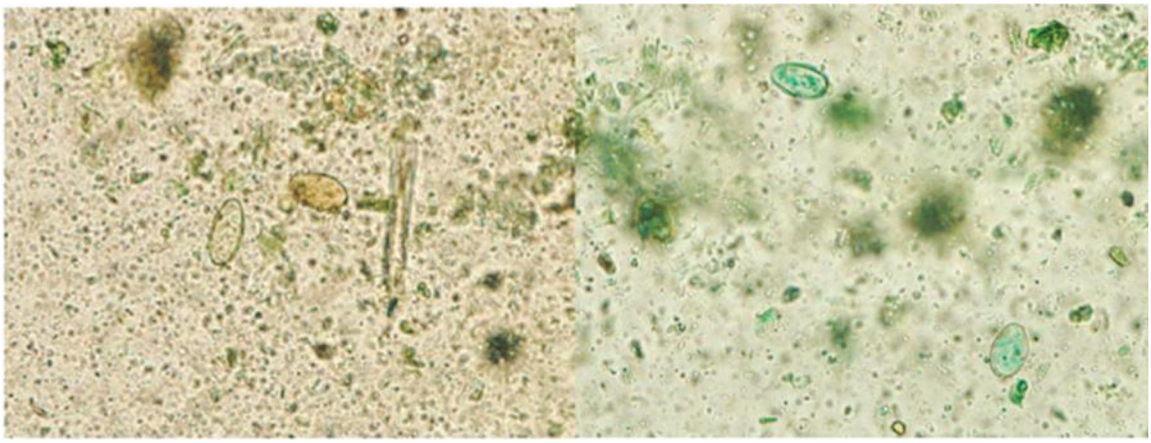

Fig. 1. Example images from the laboratory-generated data, the left image contains only MIF eggs and the right image contains OV eggs.

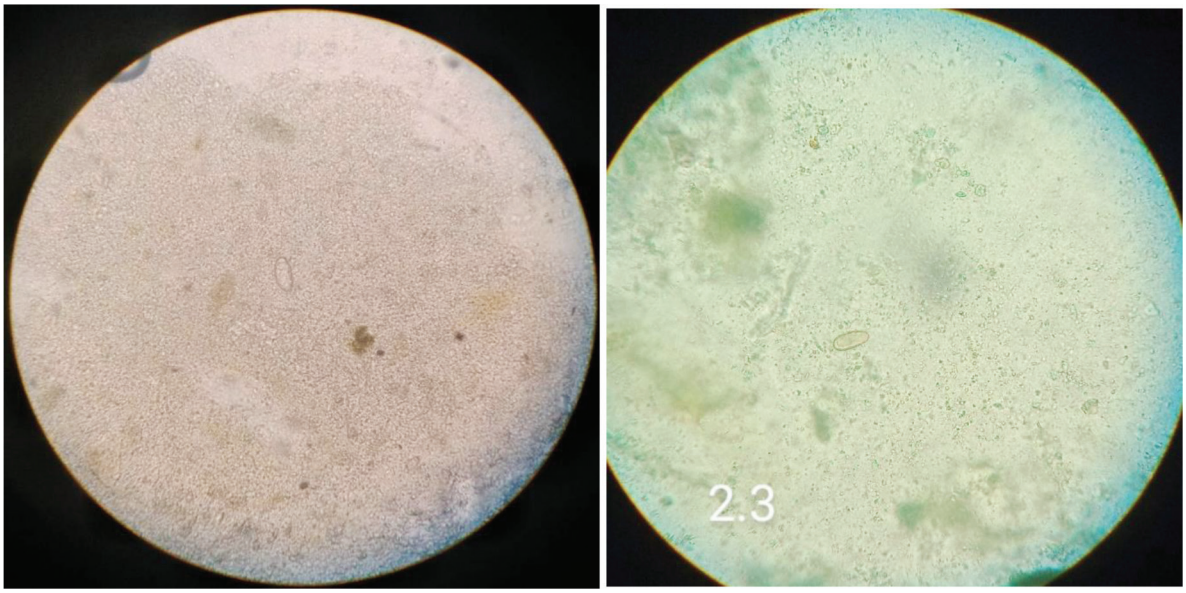

Fig. 2. Example images from the real medical diagnosis data. The left image is MIF while the right image is OV.

IV. Experiment

We selected a set of Convolutional Neural Networks and Vision Transformers that perform well in the object classification task. Then use each of them as the feature extraction part of an object detection model and compare the results using object detection metrics.

We used the modular object detection framework MMDetection [7] to train and test all models for consistency. We chose the Cascade Faster R-CNN [3] as the base object detection technique and ResNet 50 as the baseline backbone. The comparisons of Cascade Faster R-CNN and other detectors are shown in Fig. 4 to Fig. 5 and Table I

The other configurable parameters for each backbone such as the type of optimizer and initial learning rate are based on the backbone's original paper while batch size, number of epochs, input image size, learning rate schedule, and data augmentation are the same for all experiments.

We used Google's Colaboratory for training. The weights of the last epoch of the trained models were then used for further testing and evaluations.

Each model was trained for 20 epochs, with the learning rate divided by 10 at epochs 16 and 19. We also used a learning rate warm-up by using the initial learning rate of 1/1000 of the base learning rate and gradually increased to the base learning rate at 500 iterations.

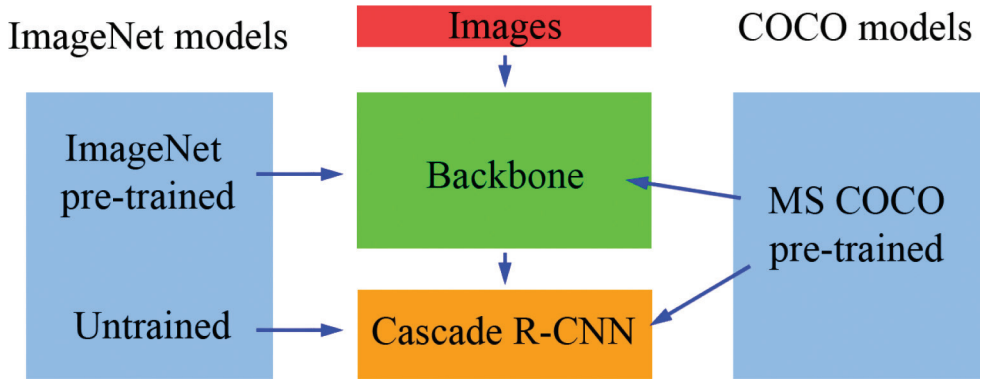

Fig. 3. The diagram of the Cascade R-CNN detector and its backbone, also shows the differences between the COCO models and ImageNet model.

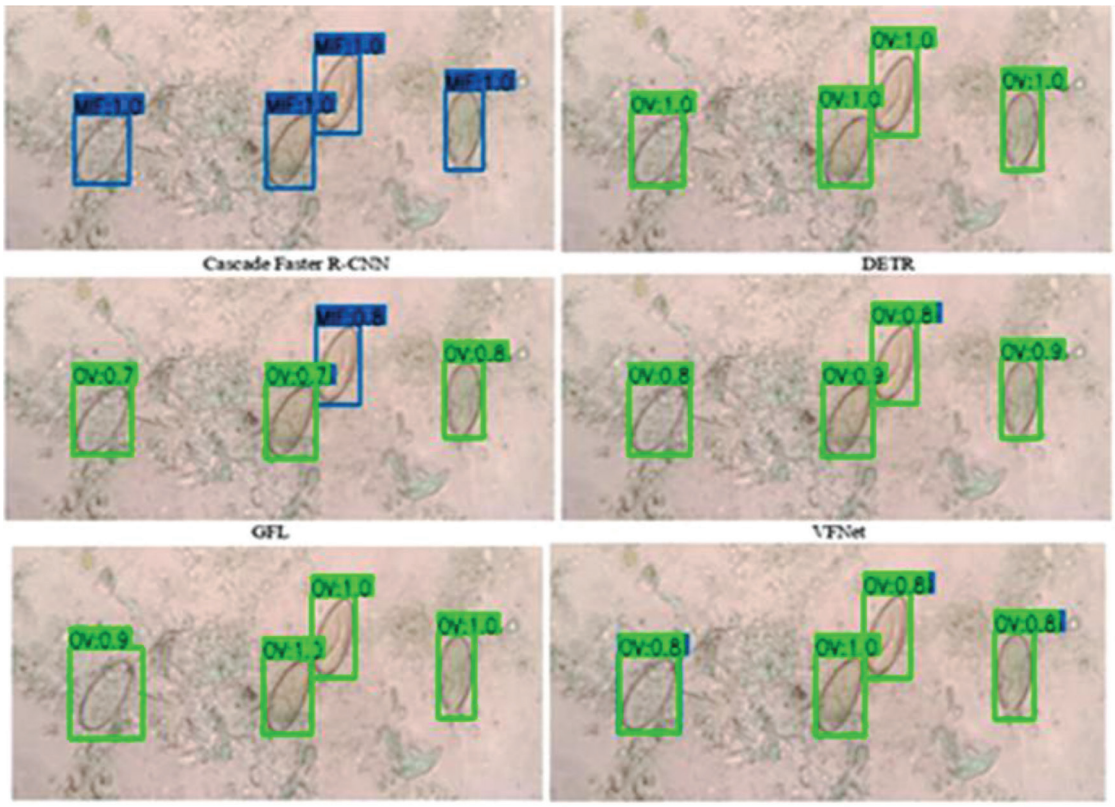

Fig. 4. The comparison of object detection techniques on MIF eggs

TABLE I
MEAN AVERAGE PRECISION AND COMPUTATION TIME OF EACH TECHNIQUE

Technique	Backbone	Mean Average Precision (mAP)				Speed (Frames/ Sec)
		0.3	0.5	0.7	0.9	
Deformable DETR	ResNet 50	0.847	0.814	0.517	0.001	4.519
YoloV3	DarkNet 53	0.845	0.835	0.723	0.097	8.364
RetinaNet	ResNet 101	0.875	0.838	0.737	0.402	4.898
Faster R-CNN	ResNet 101	0.873	0.853	0.751	0.426	4.666
Cascade Faster R-CNN	**ResNet 101**	**0.860**	**0.830**	**0.752**	**0.462**	**4.040**
RetinaNet	ResNet 50	0.888	0.857	0.755	0.433	5.660
Cascade Faster R-CNN	**ResNet 50**	**0.870**	**0.836**	**0.765**	**0.482**	**4.681**
GFL	ResNeXt 101 DCN	0.919	0.880	0.773	0.366	3.858
DETR	ResNet 50	0.916	0.898	0.800	0.096	5.996
Faster R-CNN	ResNet 50	0.912	0.885	0.807	0.457	5.474
VFNet	ResNet 50	0.944	0.927	0.840	0.565	5.313
GFL	ResNet 101 DCN	0.941	0.935	0.875	0.533	4.409
VFNET	ResNeXt 101 DCN	0.945	0.937	0.878	0.611	2.506
GFL	ResNet 50	0.945	0.941	0.882	0.518	5.770
VFNet	ResNet 50 DCN	0.953	0.944	0.897	0.597	4.729

The table is sorted by mAP at IOU 0.7.

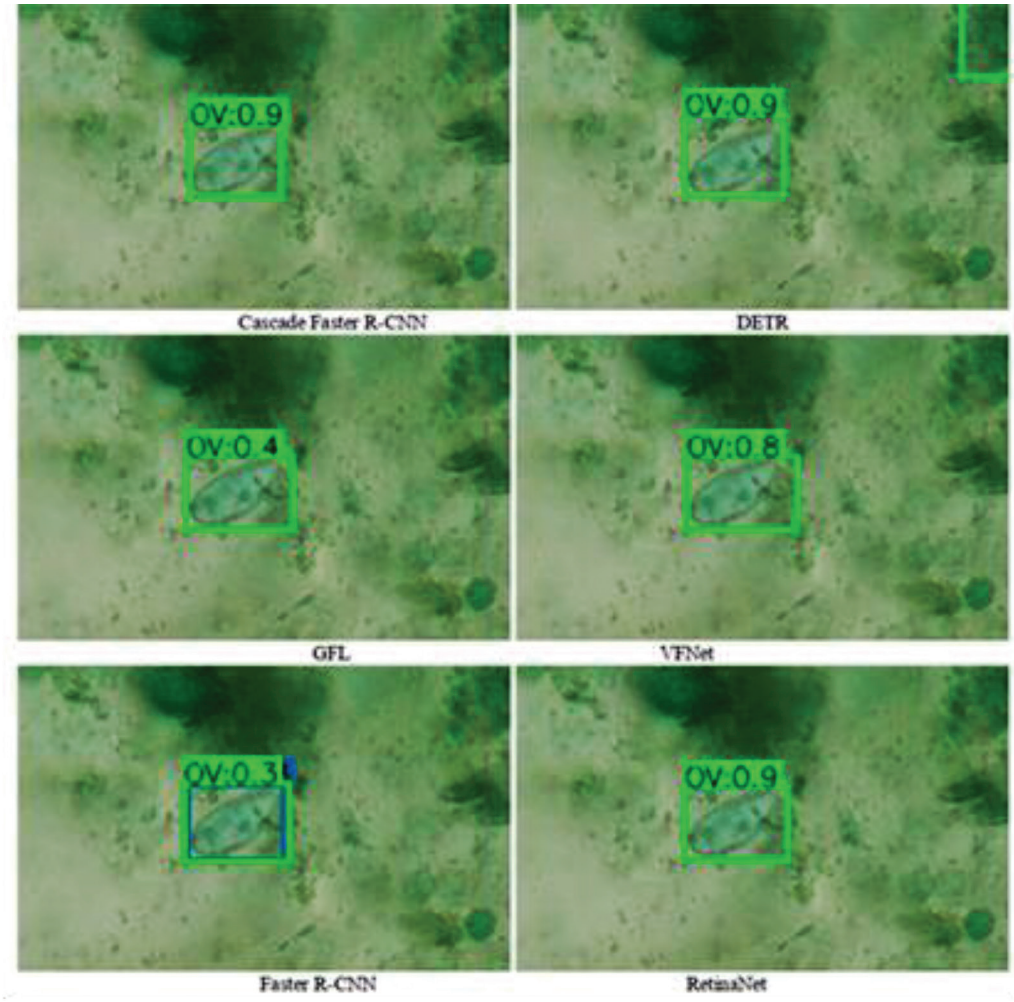

Fig. 5. The comparison of object detection techniques on an OV egg

We also experimented with the effectiveness of pre-trained models. Because most modern and high-performance backbones only had pre-trained weights available for the backbones (all of the ones we chose were pre-trained from the ImageNet dataset [8]), not the detector. So we trained another set of models for each backbone if pre-trained weights for the detector were available (all of them were pre-trained from the MS COCO dataset [9]) by using every layer except the ones that output bounding boxes and classes from COCO weight. The diagram of the experimental methodology is shown in Fig. 3.

A. ResNet

ResNet from the paper "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun from Microsoft Research published in 2015. It is one of the major breakthroughs of machine learning by introducing the use of residual connection which neutralizes the "vanishing gradient" problem while training very deep models. With sufficient accuracy and a lot of flexibility, it is the de facto standard image classification and is also widely used as the backbone of most object detection after 2015, such as Faster R-CNN [10], Cascade R-CNN [3], RetinaNet [2], and DETR [4].

We selected the intermediate size ResNet 50 and ResNet 101 for the task. They are small enough to be able to run at a reasonable speed on a personal computer while also being accurate enough for our use.

With ResNet being the baseline backbone for Cascade R-CNN originally, pre-trained weights of the object detectors were available for both ResNet 50 and ResNet 101. We trained 2 models for each depth, one using only ImageNet pre-trained backbone with the rest of the detector part randomized, and another one with the entire detector pre-trained on the MS COCO dataset with only the detection layers randomized.

All models were trained using Stochastic Gradient Descend (SGD) optimizer with a base learning rate of 0.005, weight decay of 0.0001, and momentum of 0.9.

B. ResNeXt

ResNeXt from the paper "Aggregated Residual Transformations for Deep Neural Networks" by Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He from Facebook AI Research [11]. It is the follow-up work of ResNet by increasing the cardinality of the model, the technique is more efficient than increasing the depth or width of bigger models.

As the technique relies on making ResNet bigger, we chose a depth of 101 like the deeper model of ResNet we chose previously, and 2 cardinality sizes of 32x4 and 64x4.

As ResNeXt also had MSCoco pre-trained weights like ResNet, we trained 2 sets of models with ImageNet and MSCoco pre-trained weights. All models were trained with the same setup as the ResNet.

C. ResNeSt

ResNeSt or "Split-Attention Networks" [12]. The technique utilized Attention Layers on the model that is essentially a ResNeXt. With Attention mechanism significantly improve the accuracy of the model without much speed decrease.

We chose 2 models with 50 and 101 depths, much like the ResNet.

Only ImageNet weights were available for ResNeSt. Both models were trained using the same setup as ResNet.

D. PVT

PVT or Pyramid Vision Transformer from the paper "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions" [13]. The original Vision Transformer (ViT) [24] was specifically designed for the image classification task and typically yields lower resolution outputs and was not suitable to be used as the backbone of object detection, segmentation, or similar models. PVT solved that by performing dimensional reduction similar to standard convolutional neural networks.

The models were then trained using the Adaptive Moment Estimation optimizer with weight decay (AdamW). We used a base learning rate of 0.0001 and a weight decay of 0.0001.

E. Swin

Swin from the paper "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows" [14]. It is a vision Transformer technique that improves the memory consumption of attention layers by using shifted windows and computing only non-overlapping areas.

The models were then trained using the Adaptive Moment Estimation optimizer with weight decay (AdamW). We used a base learning rate of 0.0001 and a weight decay of 0.0001.

F. PVT v2

An improved version of Pyramid Vision Transformer by the same authors [15]. The technique made improvements with the addition of overlapping patch embedding, the use of convolution layers within the transformer block, and linear-complexity attention layers.

Although the technique offers many different sizes of the model ranging from the smallest B0 to the biggest B5, B3 is the biggest we can use in our environment. Thus, 2 models were chosen, the smallest B0 and B3.

Both models were trained using the same setups as the original PVT.

V. Comparative Results

We evaluated the accuracy of the trained models using the mean Average Precision (mAP) metric on the laboratory data and measured the processing time per image processed on an Nvidia GTX1080 GPU with a batch size of 1. The mAP and speed comparison is shown in Table II.

We also measured the models' specificity (true negative over the total number of pictures without an object) and sensitivity (true positive over the total number of pictures with at least an egg) scores. The results are shown in Table III.

Originally we did not plan to use COCO weights as only a few backbones had full pre-trained detectors and instead, we chose to standardize at ImageNet weights for the backbones and train the detection part from scratch. Although using the pre-trained weights generally give better result than training from scratch especially for small dataset. But with the detection part of an object detection model being much smaller than the backbone part we did not expect much difference. The results however showed otherwise. The one model that we have tested before with COCO weights, Cascade Faster R-CNN with ResNet 101 as the backbone, significantly underperformed when trained with only ImageNet weights on the backbone part and training the detection part from scratch. Thus, we decided to include COCO weights whenever possible.

A. Lab Data

We evaluated the precision of the trained models using the mean Average Precision (mAP) score on 772 validation images. The score was computed by the ratio between the true positive (correctly predicted an object) over the total number of objects. The predicted boxes were considered true positive when the area overlapped with the ground truth boxes more than a certain threshold, this is called Intersection Over Union (IOU). We select 4 IOU thresholds, 0.3, 0.5, 0.7, and 0.9. We also measure the models' inferencing speed using GTX 1080 GPU with 1 image per batch. The speed was measured using only the average of the models' processing time of all images.

As shown in Table II. All Vision Transformer models significantly outperformed the original ResNet and ResNeXt when using ImageNet weights. PVT V2 B0 actually had more precision with ImageNet weights than ResNet with COCO weights on top of being fractionally faster than ResNet 101. Note that ResNeXt had more precision than ResNet when trained using COCO weights but worse than ResNet when trained with ImageNet weights.

We then evaluate the sensitivity and specificity scores of the trained models. The sensitivity score is calculated by using true positive over the total number of pictures with at least an egg while specificity is calculated by true negative over the total number of pictures without an object. More sensitivity means fewer parasite eggs were left undetected while more specificity means fewer false alarms. The results are shown in Table III.

TABLE II
Accuracy in Mean A Verage Precision and Inference Speed of Each Technique.

Backbone	**Pre-trained Dataset**	**Mean Average Precision (mAP)**				**Speed (images/sec)**
		0.3	0.5	0.7	0.9	
ResNeXt 101-32	ImageNet	0.730	0.674	0.578	0.134	3.944
ResNeXt 101-64	ImageNet	0.742	0.681	0.592	0.127	3.085
ResNet 50	ImageNet	0.745	0.685	0.595	0.157	5.221
ResNet 101	ImageNet	0.752	0.691	0.605	0.157	4.506
ResNeSt 50	ImageNet	0.816	0.765	0.671	0.170	3.441
ResNeSt 101	ImageNet	0.826	0.771	0.674	0.200	3.014
PVT Tiny	ImageNet	0.830	0.800	0.732	0.340	4.346
Swin Tiny	ImageNet	0.865	0.820	0.733	0.398	4.010
Swin Small	ImageNet	0.866	0.831	0.742	0.386	3.332
PVT Small	ImageNet	0.844	0.805	0.745	0.359	2.766
ResNet 50	**COCO**	**0.873**	**0.838**	**0.756**	**0.472**	**5.134**
ResNet 101	COCO	0.874	0.850	0.772	0.482	4.441
PVT V2 B0	**ImageNet**	**0.880**	**0.847**	**0.775**	**0.340**	**4.547**
PVT V2 B3	ImageNet	0.885	0.852	0.780	0.417	2.535
ResNeXt 101-64	COCO	0.883	0.862	0.788	0.493	3.133
ResNeXt 101-32	COCO	0.887	0.864	0.792	0.497	3.736

The table is sorted by mAP at IOU 0.7. The technique highlighted in bold fonts is the ones with highest score of speed multiplied by mAP at IOU 0.7 for each dataset.

TABLE III
SENSITIVITY AND SPECIFICITY SCORE OF EACH TECHNIQUE

Backbone	Pre-trained dataset	MIF		OV	
		Sensitivity	Specificity	Sensitivity	Specificity
ResNeXt 101-32	ImageNet	0.753	0.941	0.766	0.593
ResNeXt 101-64	ImageNet	0.786	0.947	0.801	0.622
ResNet 50	ImageNet	0.782	0.958	0.805	0.639
ResNet 101	ImageNet	0.804	0.945	0.826	0.659
ResNeSt 50	ImageNet	0.841	0.963	0.840	0.676
ResNeSt 101	ImageNet	0.863	0.964	0.826	0.684
PVT Tiny	ImageNet	0.876	0.981	0.840	0.801
Swin Tiny	ImageNet	0.892	0.965	0.851	0.819
Swin Small	ImageNet	0.914	0.973	0.840	0.830
PVT Small	ImageNet	0.884	0.978	0.833	0.789
ResNet 50	COCO	0.927	0.996	0.858	0.901
ResNet 101	COCO	0.937	0.987	0.862	0.908
PVT V2 B0	ImageNet	0.908	0.991	0.865	0.836
PVT V2 B3	ImageNet	0.929	0.980	0.837	0.805
ResNeXt 101-64	COCO	0.943	0.993	0.872	0.918
ResNeXt 101-32	COCO	0.949	0.991	0.862	0.924

B. Real-World Data

We select some images that each technique predicts differently. For the ImageNet pre-trained models, the same technique with different model sizes all had an interesting tendency to predict roughly the same result as shown in Fig. 7 to Fig. 9. Although the ImageNet models in general sometimes struggle to predict anything at all as shown in Fig. 6 compared to the same image predicted by the COCO models in Fig. 10. The other results of the ImageNet model compared to the COCO model are shown in Fig. 10 to Fig.13.

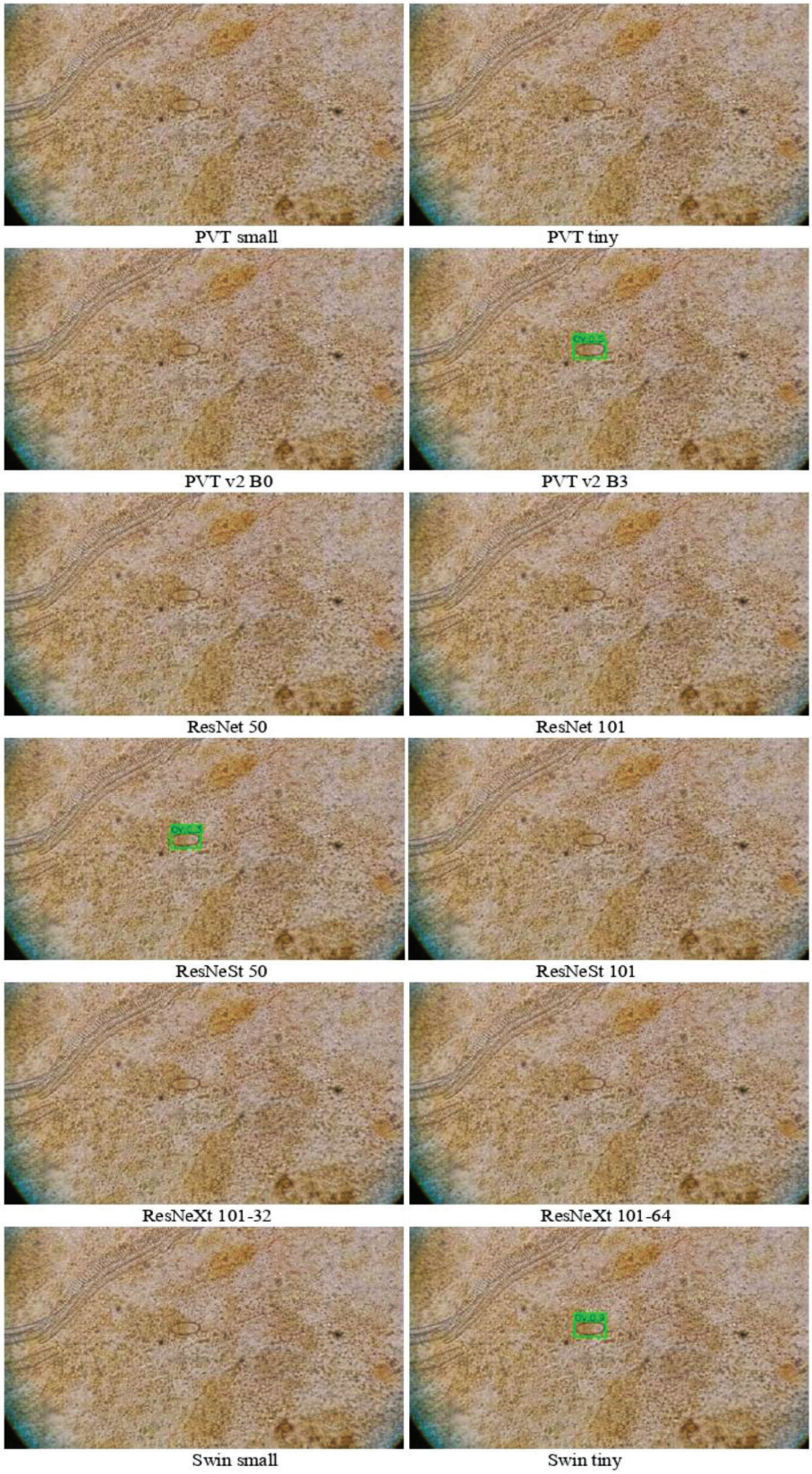

Fig. 6. Example results of an image with a MIF egg from ImageNet models

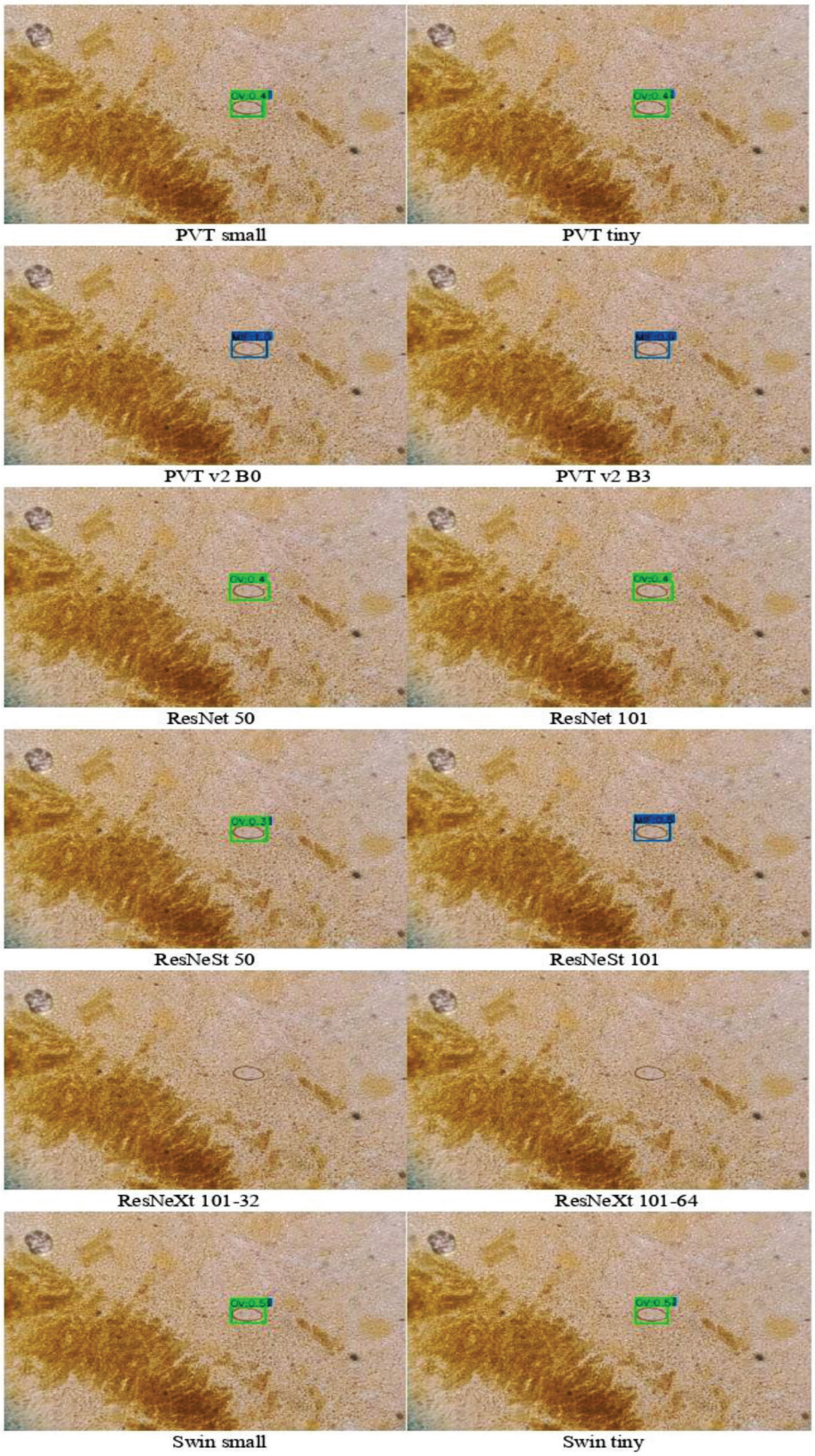

Fig. 7. Example results of an image with a MIF egg from ImageNet models

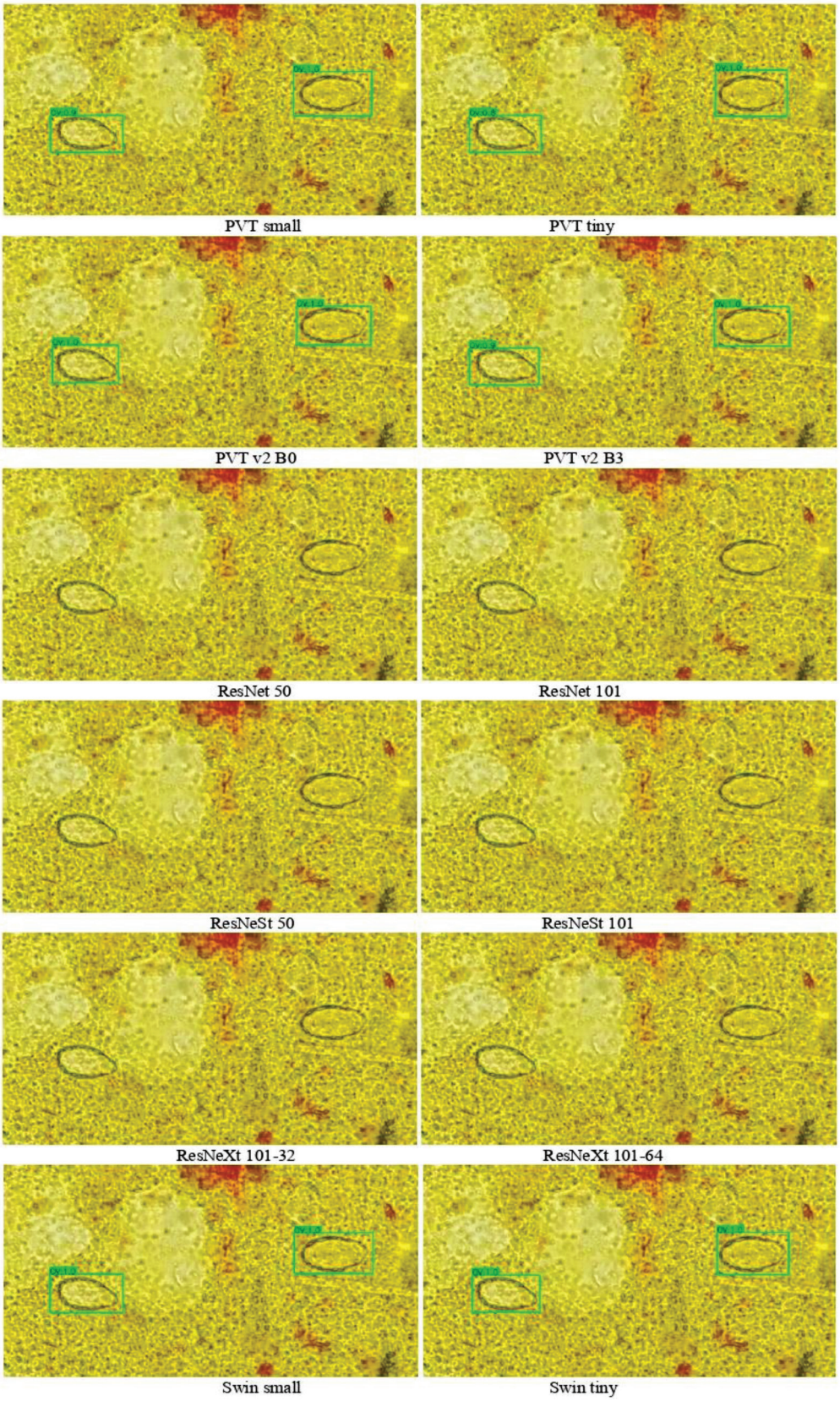

Fig. 8. Example results of an image with OV eggs from ImageNet models

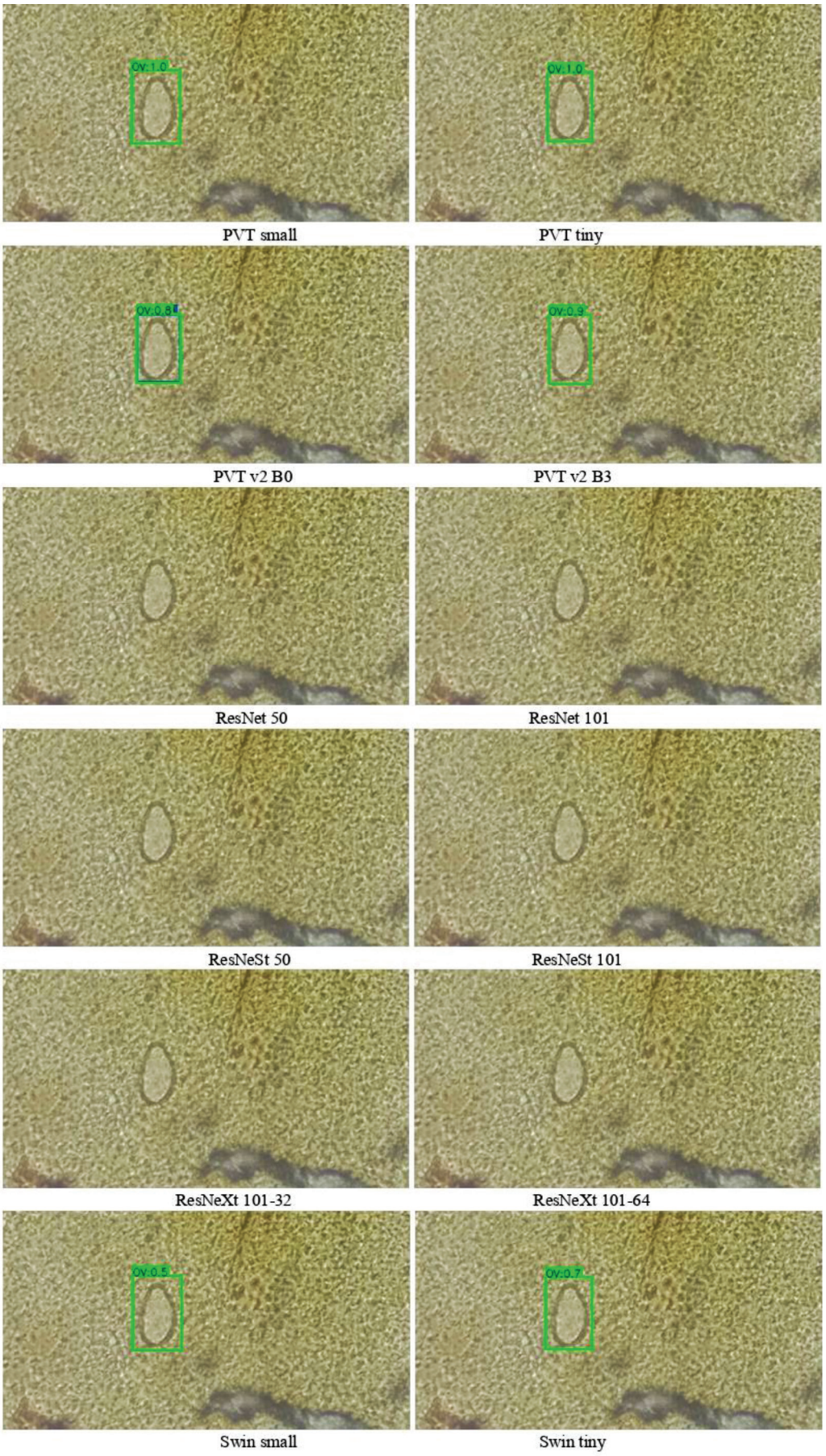

Fig. 9. Example results of an image with an OV egg from ImageNet models

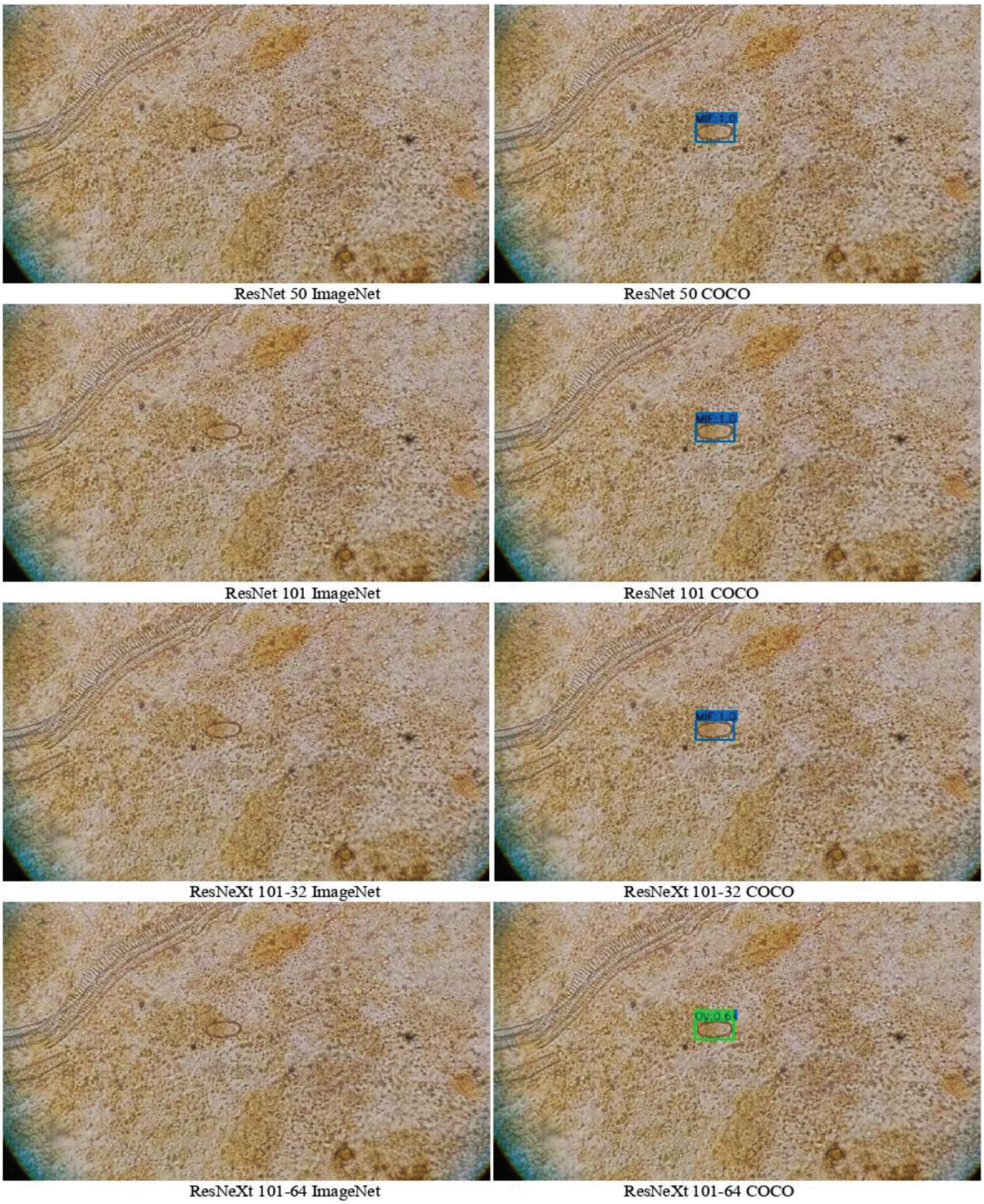

Fig. 10. Comparison result of a MIF egg between only ImageNet pre-trained backbones with randomized detector weight on the left and the entire model pre-trained on MS COCO on the right

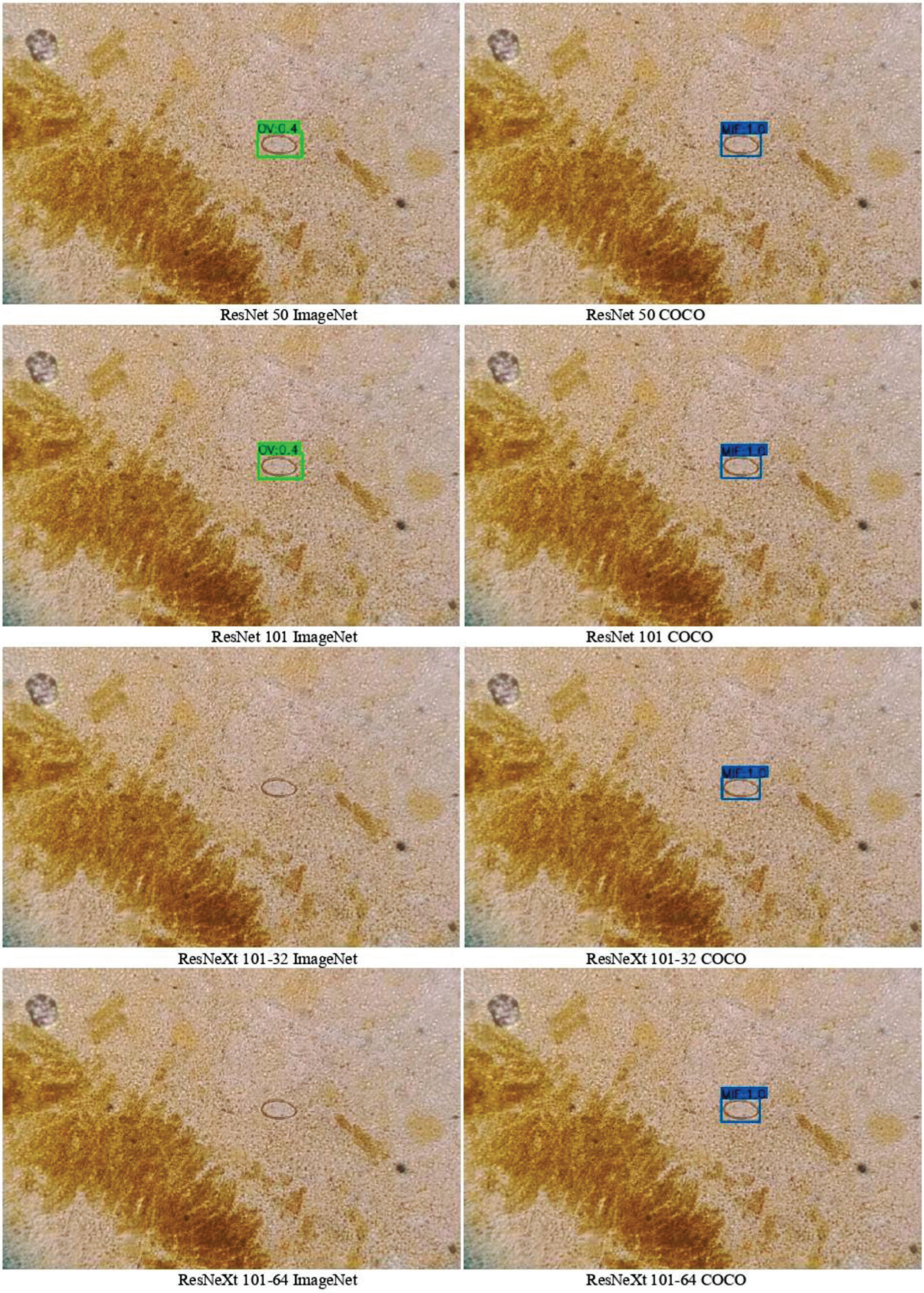

Fig. 11. Comparison result of a MIF egg between only ImageNet pre-trained backbones with randomized detector weight on the left and the entire model pre-trained on MS COCO on the right

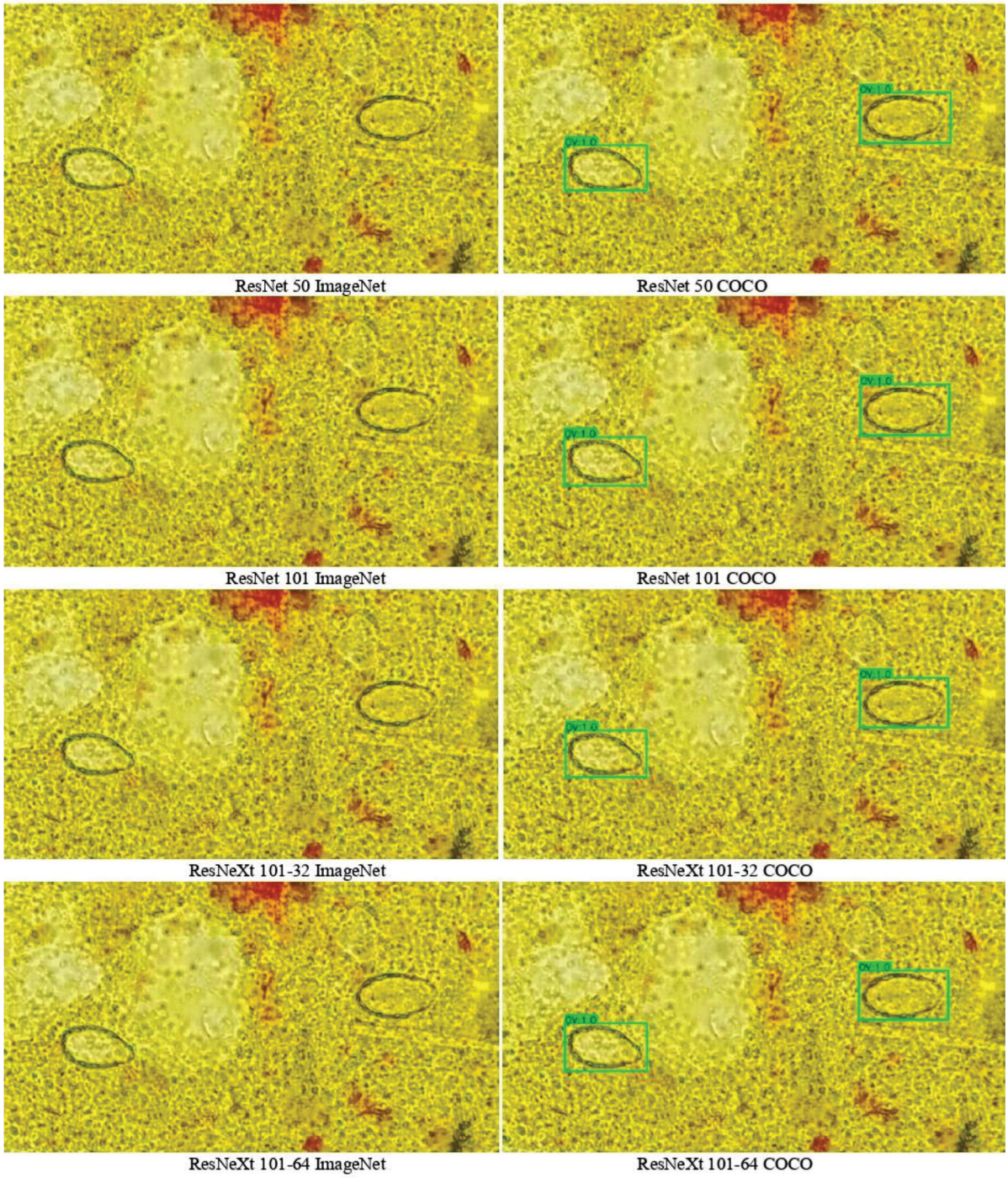

Fig. 12. Comparison result of OV eggs between only ImageNet pre-trained backbones with randomized detector weight on the left and the entire model pre-trained on MS COCO on the right

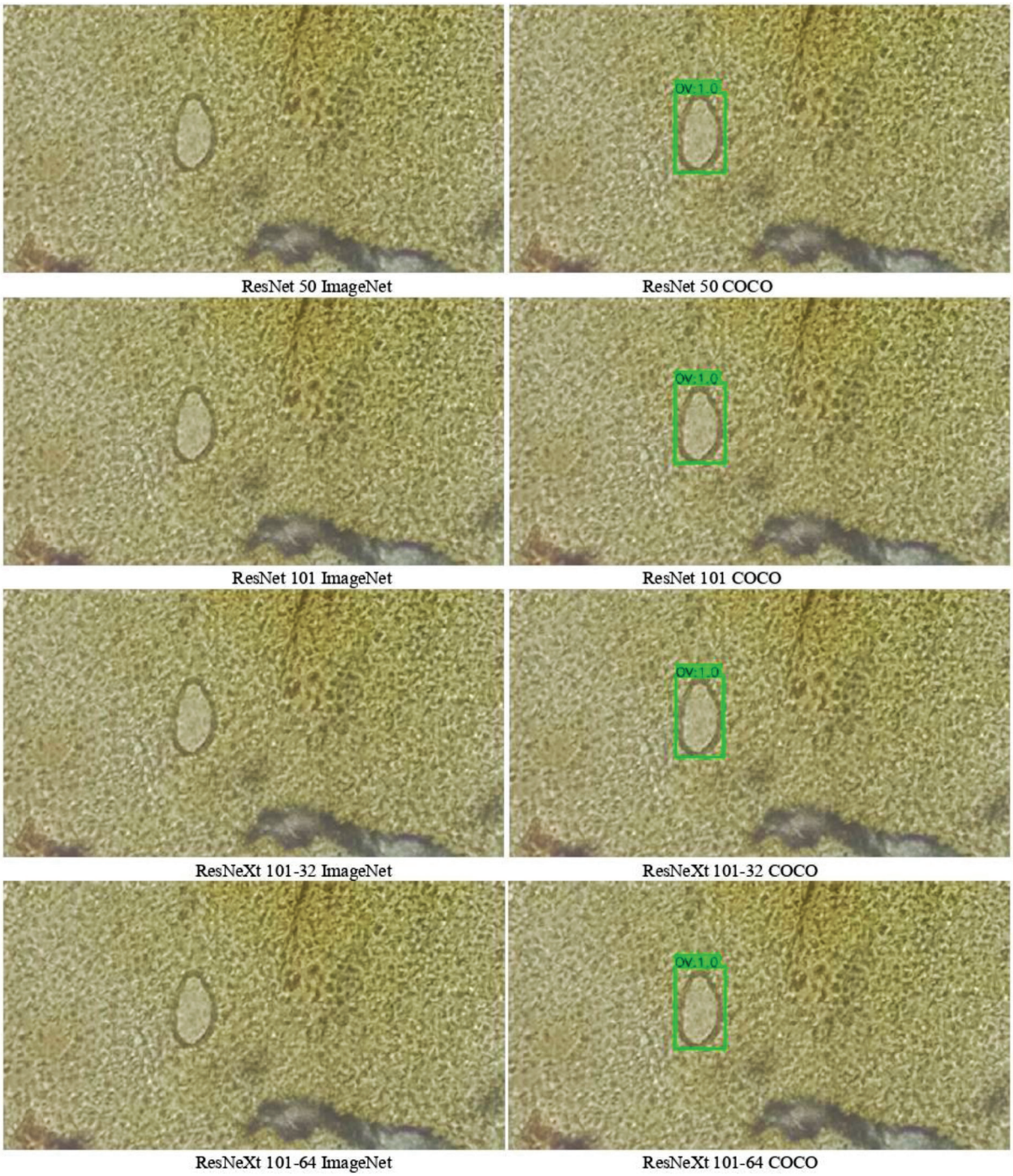

Fig. 13. Comparison result of an OV egg between only ImageNet pre-trained backbones with randomized detector weight on the left and the entire model pre-trained on MS COCO on the right

Like in the lab data, Vision Transformer models were significantly better than the original ResNet and ResNeXt. PVTv2 in particular showed good classification accuracy on top of, most of the time, correct object localization even without pre-trained weight on the detector part. PVT and Swin also had good localization but not so much for classification.

VI. Conclusions

Both results from lab data and real-world data showed that using a higher-performance backbone can significantly improve the performance of the detector. But for backbones with a similar level of performance like ResNet 50 and ResNet 101, the complete pre-trained weights are much more important than a little bigger backbone.

While all Vision Transformer models outperformed the original ResNet. The fact that the ResNeXt actually had less precision than ResNet with just ImageNet while performing better with COCO weights suggests that the use of pre-trained weights gives different performance increases for each backbone. Thus, the result between Vision Transformers themselves may be inconclusive. Pre-trained weights for the entire detector for each backbone would be needed for more complete results.

ACKNOWLEDGMENT

Scholarship received from Thailand Graduate Institute of Science and Technology (TGIST) of the National Science and Technology Development Agency (NSTDA). Scholarship ID SCA-CO-2562-9836-TH.

REFERENCES

[1] F. Bray, J. Ferlay, I. Soerjomataram et al. (2018, Sep). Global Cancer Statistics 2018: Globocan Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*. [Online]. *68(6)*, pp. 394-424. Available: https://acsjournals. onlinelibrary.wiley.com/doi/abs/10.3322/caac.21492

[2] T. Y. Lin, P. Goyal, R. Girshick et al. (2018, Aug). *Focal Loss for Dense Object Detection. Computer Vision Foundation*. [Online]. *1*, pp. 2980-2988. Available: https://arxiv.org/abs/1708.02002

[3] Z. Cai and N. Vasconcelos. (2019, Nov). Cascade R-CNN: High-Quality Object Detection and Instance Segmentation. *IEEE*. [Online]. *43(5)*, pp. 1483-1498. Available: http://dx.doi.org/10.1109/tpami.2019.2956516

[4] N. Carion, F. Massa, G. Synnaeve et al. (2020, May). *End-to-End Object Detection with Transformers. European Conference on Computer Vision*. [Online]. *12346*, pp. 1-17. Available: https://arxiv.org/abs/2005.12872

[5] D, Bolya, C, Zhou, F, Xiao et al. (2019, Apr. 4). YOLACT: *Real-Time Instance Segmentation. Computer Vision and Pattern Recognition*. [Online]. Available: https://arxiv.org/abs/1904.02689

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al. (2022, Jan. 10). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. [Online]. Available: https://arxiv.org/abs/2010.11929

[7] K. Chen, J. Wang, J. Pang et al., (2022, Jan. 10). *Mmdetection: Open MMlab Detection Toolbox and Benchmark*. [Online]. Available: https://arxiv.org/abs/1906.07155

[8] O. Russakovsky, J. Deng, H. Su et al. (2022, Jan. 10). *ImageNet: A Large-scale Hierarchical Image Database*. [Online]. Available: https://ieeexplore.ieee.org/document/5206848

[9] T. Y. Lin, M. Maire, S. Belongie et al. (2022, Jan. 15). *Microsoft coco: Common Objects in Context*. [Online]. Available: https://arxiv.org/abs/1405.0312

[10] S. Ren, K. He, R. Girshick et al. (2022, Jan. 15). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. [Online]. Available: https://arxiv.org/abs/1506.01497

[11] S. Xie, R. Girshick, P. Dollár et al. (2022, Jan. 15). *Aggregated Residual Transformations for Deep Neural Networks*. [Online] Available: https://arxiv.org/abs/1611.05431

[12] H. Zhang, C. Wu, Z. Zhang et al. (2022, Jan. 15). *ResNeSt: Split-Attention Networks*. [Online]. Available: https://arxiv.org/abs/2004.08955

[13] W. Wang, E. Xie, X. Li et al. (2022, Jan. 15). *Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions*. [Online]. Available: https://arxiv.org/abs/2102.12122

[14] Z. Liu, Y. Lin, Y. Cao et al. (2021, Jan. 15). *Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows*. [Online]. Available: https://arxiv.org/abs/2103.14030

[15] W. Wang, E. Xie, X. Li et al. (2022, Jan. 15). *PVTv2: Improved Baselines with Pyramid Vision Transformer*. [Online]. Available: https://arxiv.org/abs/2106.13797

Natthaphon Hongcharoen received his B. Eng. degree from the Department of Computer Engineering, Faculty of Engineering and Technology, Panyapiwat Institute of Management in 2019. He has experience in internships at the National Electronics and Computer Technology Center (NECTEC), and the National Science and Technology Development Agency (NSTDA) in the Image Technology Research Laboratory. And won a gold medal at Super AI Engineer Camp Season 1 by the Artificial Intelligence Association of Thailand (AIAT). His research areas include Machine Learning, Image processing, and Computer Vision.

Parinya Sanguansat is an Associate Professor in Electrical Engineering and head of Computer Engineering and Artificial Intelligence at Panyapiwat Institute of Management (PIM), Thailand. He graduated B.Eng., M.Eng., and Ph.D. in Electrical Engineering from Chulalongkorn University. His research areas include Machine Learning, Image processing, and Computer Vision. He got many research grants from both private and public organizations. He has written several books about Machine Learning and MATLAB programming.

Sanparith Marukatat graduated in Computer Science in 1998 from the University of Franche-Comté, Besançon, France. In 2004, he completed his doctoral thesis on online handwriting recognition at the University of Paris 6, France. Currently, he is Principal Researcher in the AI Research Group at the National Electronics and Computer Technology Center (NECTEC). Dr. Marukatat's research interests include machine learning, pattern recognition using statistical tools, and deep learning.