Fluke Eggs Detection and Classification Using Deep Convolution Neural Network

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: October 14, 2021 / Revised: December 19, 2021 / Accepted: January 27, 2022

***Abstract*—We present the experimental results of utilizing object detection to solve the problem of detecting and also classifying the parasite eggs in the fecal slides. We experimented with different detection techniques and different sizes of the backbone part. The trained models were evaluated using standard mean Average Precision (mAP) on the results from labeled data collected from the closed environment and also manual evaluation on the results from field data collected from actual medical diagnoses that do not have accurate labels. On the lab data, VFNet achieved a very good 0.897 mAP at the Intersection over Union (IoU) of 0.7 thresholds but performed rather poorly in the classification part on the field data. The relatively older technique Cascade Faster R-CNN had a little below average result in the lab data but had a very good classification accuracy on the field data. The backbone part also had conflicting results on the lab data and field data. The smaller backbones performed better in the lab data but lost to the bigger backbones on the field data.**

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

I. Introduction

According to World Health Organization (WHO), liver cancer is the fourth leading cause of cancer death in 2018 [1]. With liver fluke or Opisthorchis Vivertini (OV) being one of the causes of liver cancer and one of the causes of OV infection is raw fish consumption which is common in the north-eastern region of Thailand. The accumulation of OV will eventually lead to liver cancer in advanced age if not treated properly.

Parasite infection in humans can be detected by analyzing the fecal slide but OV is not the only type of parasite eggs found in human feces. Minute Intestinal Flukes (MIF) are in fact very similar to OV as shown in Fig. 1 but they infect different organs, OV infects livers whereas MIF infects small intestines.

While finding the parasite eggs in the fecal slide is achievable, it is very difficult to distinguish them and would need experts to do the job.

This paper aims to utilize the object detection methods and create models that can accurately detect and classify the types of parasite eggs in the fecal slides which would free up the experts for other tasks. And without the need for experts, the diagnosis can be done on a larger scale easily as medical students, nurses or clinicians can perform the task as well.

II. Related Work

Bruun et al. [2] used an elliptic filter to detect the eggs to detect the eggs. The filters were designed using the average size of the eggs and manually selected rotation angles. Yang et al. [3] used the classic artificial neural network (also known as multi-layer perceptron) to classify the parasite eggs. Two models were used in this work, one for determining whether or not the object is a parasite egg and another one for classification. Akintayo et al. [4] used a convolutional autoencoder for the localization and classification of microscopic nematode eggs.

III. Data

A. Training Data

To train an object detection model, the images and corresponding annotations of the objects inside the images are needed. We used the data obtained from a medical laboratory and the two types of parasites (OV and MIF) that were cultivated in a closed environment. Then the eggs were added to the fecal slides; one type per slide either OV or MIF. This ensured that all the eggs in the images were correctly classified and reduced the problems with incorrect labeling.

The fecal slides were then passed through the Kato-Katz method. The images were taken from the microscope with about 40 times magnification using hand-held digital cameras and smartphone cameras. This increased the variation of the obtained images to some extent. The locations of the eggs in these images were then manually tagged by domain experts from the Department of Disease Control (DDC), Ministry of Public Health.

In total, there were 1,684 MIF images and 1,553 OV images with 2,439 MIF eggs and 3,510 OV eggs. These images were then separated into 2,465 training images and 772 validation images. The examples of the data are shown in Fig. 1.

B. Field Data

We also used real images obtained from actual medical diagnoses. As it is possible that the images gathered from the closed environments using cultivated parasites may not represent the real-world problem well enough. But with the difficulty of classifying the eggs resulted in many incorrect labeling and conflicting classification of the images from the same slides from multiple sources, we used them only as test data instead. The examples of the field data are shown in Fig. 2.

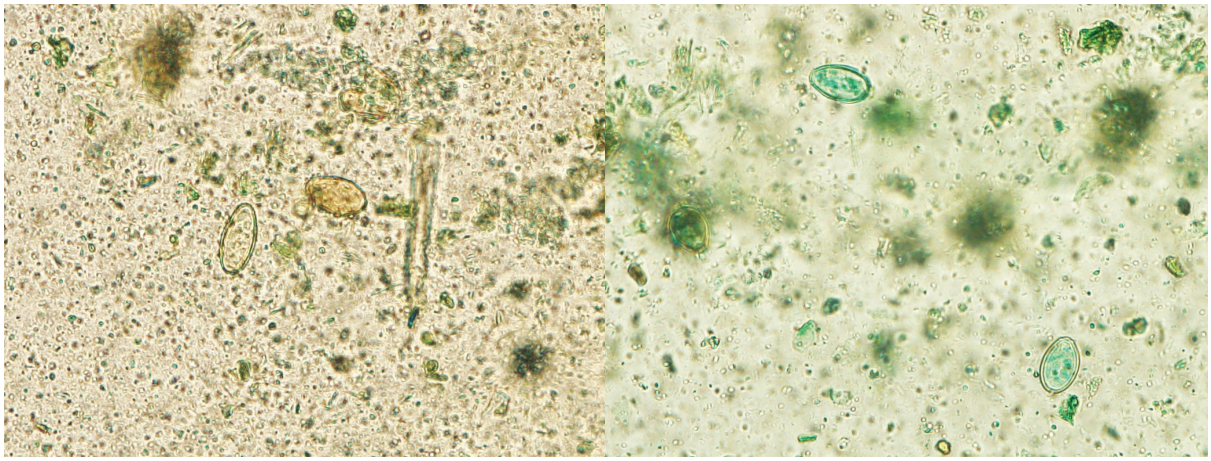


Fig. 1. Example images from the lab data, with MIF on the left and OV on the right. Note that the color difference is per slice basis and cannot be used to distinguish the eggs in the picture.

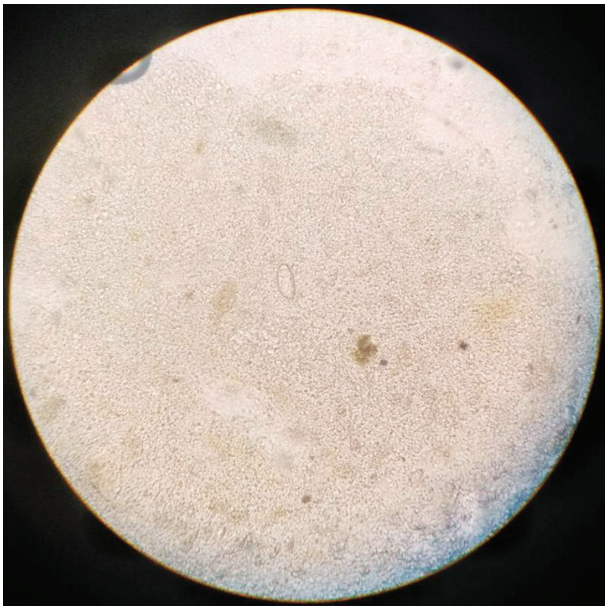


Fig. 2. Example images from the field data. With MIF on the left and OV on the right.

IV. Experiment

We experimented with standard object detection, having the detectors do both localizing and classifying the objects. The option of using object detection to only locate the objects and use sophisticated classification models is also considered but we choose not to use it because that would require near-perfect object localization for the models to be accurate.

We compared different object detection algorithms in terms of performance and speed. The performance of the algorithms was measured using mean Average Precision (mAP) and measure the speed with how many images can the algorithms process in one second during the inference phase.

We chose to standardize on one single framework that contained many algorithms in order to reduce the differences between each experiment to only the differences in the algorithms as much as possible. MMDetection [5] was chosen for this.

All techniques except the YoloV3 used the same data augmentation in the training phase. Each input image was resized to have a smaller size equal to 800 pixels and keep the aspect ratio after resizing the bigger size was not bigger than 1333 pixels, if it was bigger, we center cropped it to 1333 pixels. Then the images were randomly flipped with 0.5 probability both vertically and horizontally. The input images were then normalized using MSCOCO [6] mean and standard deviation. And as the images needed to be stacked as batches for training, we pad the smaller images with zero to make the size equal to the biggest image in each batch, if that particular model needed the input to be divisible by 32 then we padded all images further to meet that requirement.

The optimizer and learning rate schedule for each model was configured similarly to the original paper. Except with a smaller learning rate as we used smaller batch sizes for our smaller GPUs.

TABLE I
Mean Average Precision and Computation Time of Each Technique.

Technique	Backbone	Mean Average Precision (mAP)				Speed
		0.3	0.5	0.7	0.9	(frames/sec)
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.

TABLE II
Sensitivity and Specificity Score of Each Technique

Technique	Backbone	MIF		OV	
		Sensitivity	Specificity	Sensitivity	Specificity
Cascade Faster R-CNN	**ResNet 101**	**0.931**	**0.981**	**0.848**	**0.912**
Cascade Faster R-CNN	ResNet 50	0.933	0.993	0.851	0.903
DETR	ResNet 50	0.961	0.971	0.908	0.667
RetinaNet	ResNet 50	0.931	0.978	0.833	0.893
RetinaNet	ResNet 101	0.924	0.976	0.837	0.889
Faster R-CNN	ResNet 101	0.933	0.993	0.840	0.893
Faster R-CNN	ResNet 50	0.963	0.993	0.890	0.922
YoloV3	DarkNet 53	0.939	0.967	0.862	0.936
Deformable DETR	ResNet 50	0.818	0.999	0.816	0.934
GFL	ResNet 50	0.984	0.983	0.915	0.897
GFL	ResNet 50	0.984	0.988	0.936	0.893
GFL	ResNet 101 DCN	0.99	0.965	0.915	0.854
GFL	ResNeXt 101 DCN	0.951	0.977	0.787	0.807
VFNet	ResNet 50 DCN	0.984	0.990	0.911	0.893
VFNet	ResNet 50 DCN	0.984	0.988	0.915	0.914

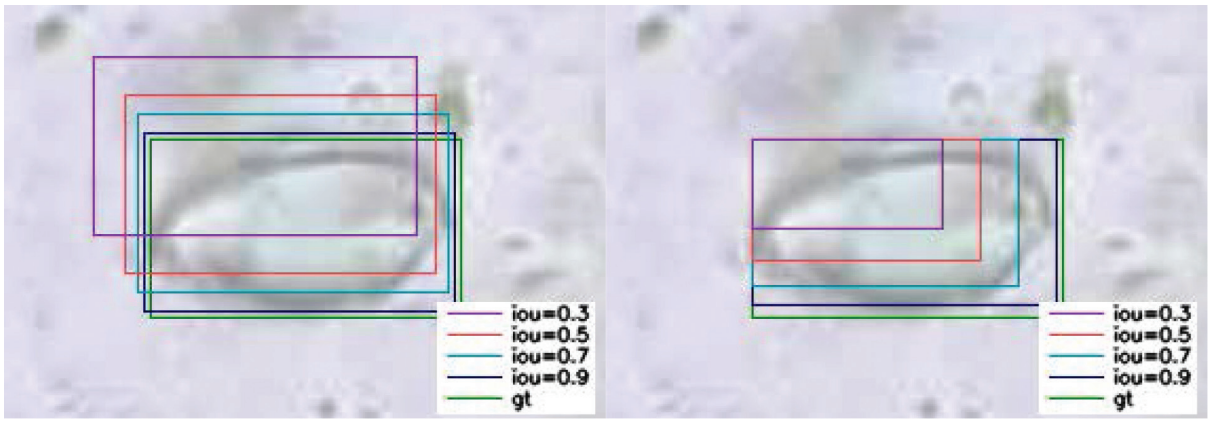


Fig. 3. A comparison of the 4 chosen IOU. IOU 0.3 is purple, 0.5 is red, 0.7 is yellow, and 0.9 is blue compared to the ground truth (green). The left image shows similar bounding boxes in different locations while the right image shows different bounding boxes with the same starting points.

We also experimented with the effectiveness of using bigger feature extractors as part of the algorithms (also known as backbones). As backbones like ResNet [7] can be made more accurate by simply adding more layers but this would increase processing time and thus reduce the speed. We intended to see whether or not using a bigger backbone helps increase performance in measurable metrics and how much the speed decreases. The ResNet 50 [4], ResNet 101 [4], and ResNeXt 101 [8] are chosen. With ResNet 50 as the baseline, ResNet 101 is twice the depth of ResNet 50, and ResNeXt is 101 a larger version of ResNet 101. We also trained some models using Deformable Convolution [6] layers in the detection stage.

We then measure the inference speed of each technique using an Nvidia GTX 1080 GPU with a batch size of 1 and the same resizing as the training stage.

A. RetinaNet

RetinaNet introduces Focal Loss. A method to reduce the imbalanced data problem where there are too many background objects in the ground-truth compare to the foreground objects by reducing the contribution of score from the background class [10]. Two models were trained. One with ResNet 50 as the backbone and one with ResNet 101. Both models were trained for 24 epochs with a batch size of 2. We used Stochastic Gradient Descent as the optimizer with a base learning rate of 0.01, the a momentum of 0.9, and a weight decay of 0.0001. The warm-up procedure was by using a 0.00001 initial learning rate and linearly increased by 0.000002 after every batch until it reached the base learning rate (500 batches). The learning rate was then reduced by a factor of 0.1 at epochs 16 and 22.

B. YOLO v3

Yolo v3 predicts the center point of each foreground object and uses a set of handcrafted anchor boxes to help determines the object boxes' width and height. Then classify the objects on the same output head as the bounding boxes head [11].

The Yolo v3 was trained with a different pre-processing procedure than other models as it is the only model that was trained using square input in the original paper. And with our expected performance of Yolo being a fast model with some accuracy compromise, we decided to use the square resizing as well and also used Yolo's unique backbone DarkNet 53. The input images were randomly cropped to the minimum of 30% of the original size while keeping the center of all the bounding boxes. The images were then center-cropped with randomized sizes between 320x320 and 608x608.

The model was trained using Stochastic Gradient Descend (SGD) as the optimizer with a base learning rate of 0.00025 and momentum of 0.9. The learning rate was warm up by starting at 0.000025 and linearly increase to the base value after 2000 batches. The model was trained for 100 epochs and the learning rate was reduced by a factor of 0.1 at epoch 80 and 90.

C. Faster R-CNN

Faster R-CNN is a two-stage network. An improved version of R-CNN and Fast R-CNN. It utilizes Region Proposal Network (RPN) to predict regions of interest after the backbone stage. Then uses the Region of Interest Pooling network (ROI Pooling) to choose the relevant boxes and then predict the bounding boxes and classify the objects [12].

Two models were trained with one using ResNet 50 and one with ResNet 101 as backbones. The models were trained using SGD with a base learning rate of 0.005, a momentum of 0.9, and the a weight decay of 0.0001. The warm-up train phase was using an initial learning rate of 0.000005 and linearly increased to the base learning rate at 500 batches. Training for a total of 24 epochs with a batch size of 2. The learning rate was reduced by a factor of 0.1 at epochs 16 and 22.

D. Cascade Faster R-CNN

An improved version of Faster R-CNN. It is designed to solve the problem of positive samples vanishing when increasing the IOU threshold and when the testing phase uses a different IOU than during the training phase [13].

The training configurations were the same as the ones that were used to train Faster R-CNN.

E. DETR

DETR [14] utilizes the Transformer [15] for the prediction stage in order to get away with the hand-crafted knowledge such as anchor boxes and Non-Maximum Suppression.

The Decoupled Weight Decay Regularization version of the Adam (AdamW) optimizer was used with an initial learning rate of 1e-4 and weight decay of 1e-4. The input images were heuristically resized to have the a larger size equal to 1333 pixels and then random croped to have a smaller size between 480 to 800 pixels. The batch size was 2. The model trained for 24 epochs and the learning rate was divided by 10 at epochs 16 and 22.

F. Deformable DETR

A proposed improved version of DETR to increase the convergence speed and feature spatial resolution [16].

The model was trained using AdamW like DETR and similar other setups. In addition to using a smaller learning rate with 0.1 factor at the backbone, sampling offsets, and reference points.

G. Generalized Focal Loss

An improved version of RetinaNet and its Focal Loss. Instead of calculating the IOU score on the bounding box head separate from both the bounding box and classification and then perform Non-Maximum Suppression on the joined value of all three. GFL combines the IOU and classification score and NMS suppression over this value not on the bounding box [17].

As it is being an improved version of RetinaNet, we trained GFL using the same configuration. With the exception of an additional ResNeXt 101 backbone and using DCN [6] on both ResNet 101 and ResNeXt.

H. VarifocalNet

As the name implies, VarifocalNet or VFNet is another variation of Focal Loss [7] similar to the GFL combines the star-shaped bounding boxes. The VFNet replaces the classification score of the ground-truth class with the IOU score between predicted boxes and ground-truth boxes [18]. Very similar in concept to the GFL but with the difference is being that GFL combines both bounding boxes and classification scores together whereas VFNet keeps them separated like the original Focal Loss.

VFNet was trained using the same configuration as RetinaNet and GFL. With the backbone being ResNet 50, ResNet 50 DCN, and ResNeXt 101 DCN. In order to measure the performance of DCN, we trained VFNet with ResNet 50 and ResNet 50 DCN two separate models each.

VI. Comparative Results

A. Lab Data

We evaluate the models on the validation dataset using the Mean Average Precision (mAP) metric. The score was calculated by computing the intersection over union (IOU) between the predicted bounding box and ground truth. The IOU threshold was set at 0.3, 0.5, 0.7, and 0.9. The predicted bounding box with over the chosen IOU score compared to a ground-truth bounding box was considered as matched. Precision was the ratio between the correctly predicted bounding boxes over the total number of predicted bounding boxes.

Table I shows the mAP for the five detectors sorted by mAP at IOU 0.7. VFNet models outperformed everything in terms of precision while YOLO v3 was significantly faster than other models. The scores of VFNet with ResNet 50 and ResNet 50 DCN are the average of the two models of each backbone.

To visualize the differences between the IOU numbers, Fig. 3 shows bounding boxes with IOU 0.3 to IOU 0.9 compare to ground truth. While the bounding boxes with IOU 0.9 were very similar to the ground truth but we decided that boxes with IOU 0.7 are good enough for our usage, which as shown in Table I faster models had a significant drop in precision at IOU 0.9 compared to IOU 0.7.

We then measure each technique's specificity and sensitivity performance. The specificity score was calculated by the number of pictures without the eggs from the specific class and the models did not predict any object over the total number of pictures without the eggs from the specific class. The sensitivity score was calculated by the number of pictures with eggs from each class and the model correctly detected over the total number of pictures with eggs from the specific class. Because the specificity test did not need

any label, we included the field data along with using the data from the different classes. The results are shown in Table II.

B. Field Data

Without the usable labels as stated in section III-B, we only had the models predict these images and manually verify the bounding boxes' precision and classification accuracy on some images with known classes.

We selected a small portion of images that contains eggs from only one class. The results are as shown in Fig. 4 to Fig. 9. Contradicted to the lab data and mAP table, GFL performed worse than other hi-mAP models by tending to not predict anything like Fig. 7. The best performing in this test is Cascade Faster R-CNN with its remarkable classification accuracy on top of good detection as well. The most surprising result of this test would be Yolo v3, considering its performance in the lab data it did very well in the field data, especially in classification. Every model tested was capable of reliably detecting most of the eggs but often fail to classify them and predict everything as OV. This suggests there should be some kind of biases in the training data.

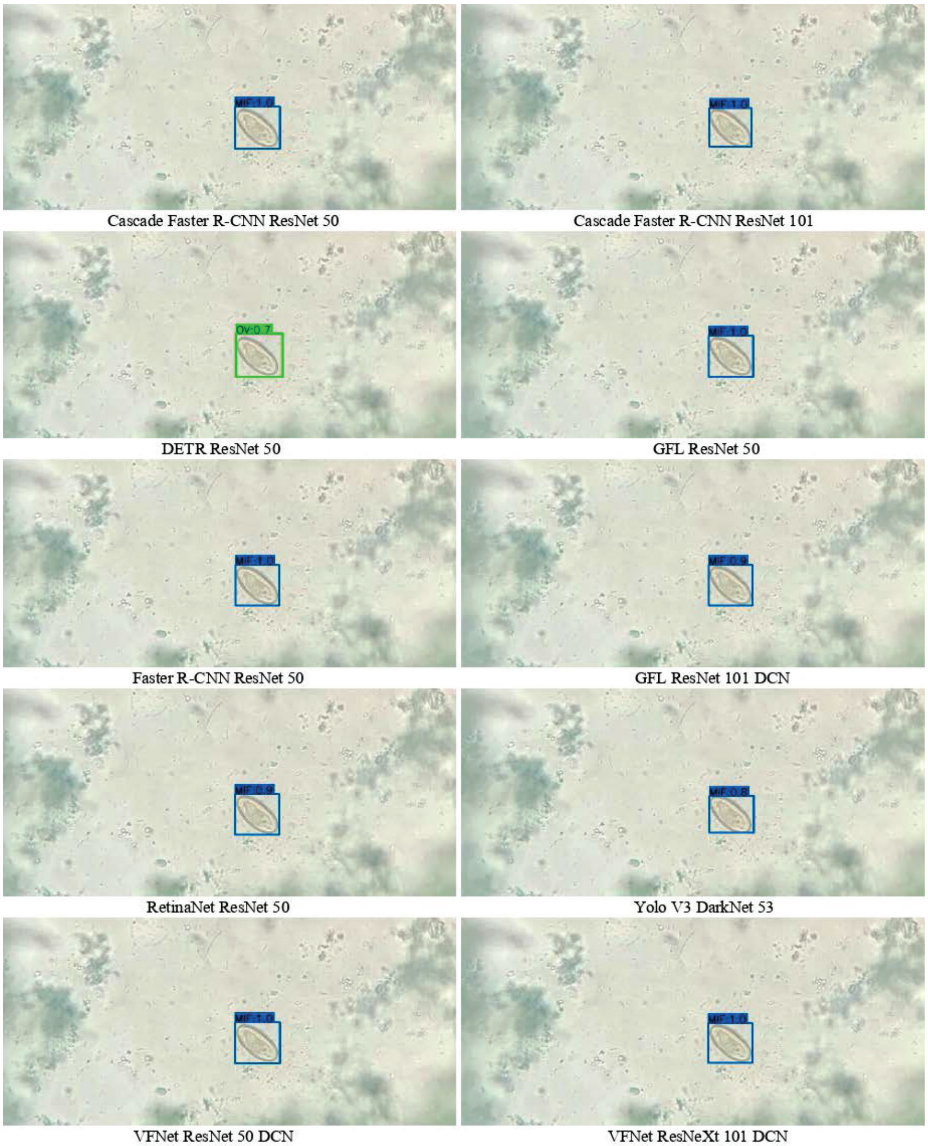


Fig. 4. Example result of an image with a MIF egg from the field data

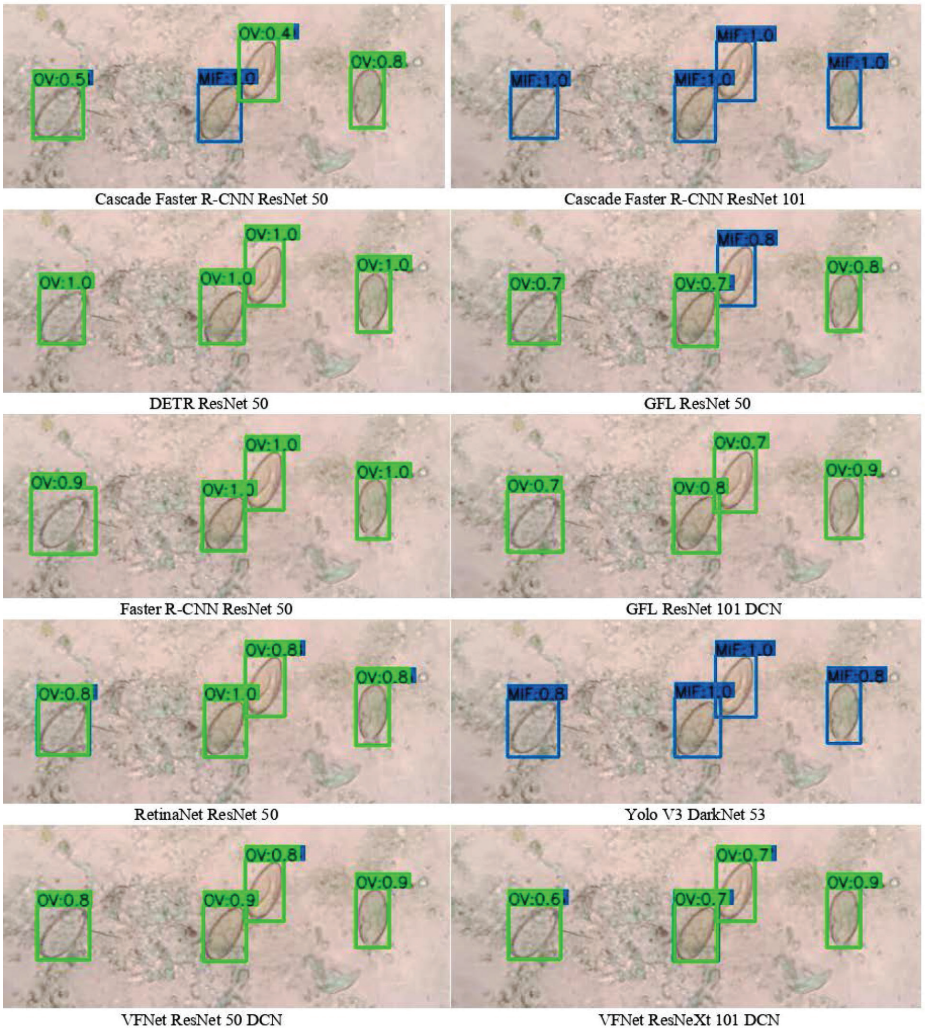


Fig. 5. Example result of an image with a MIF egg from the field data

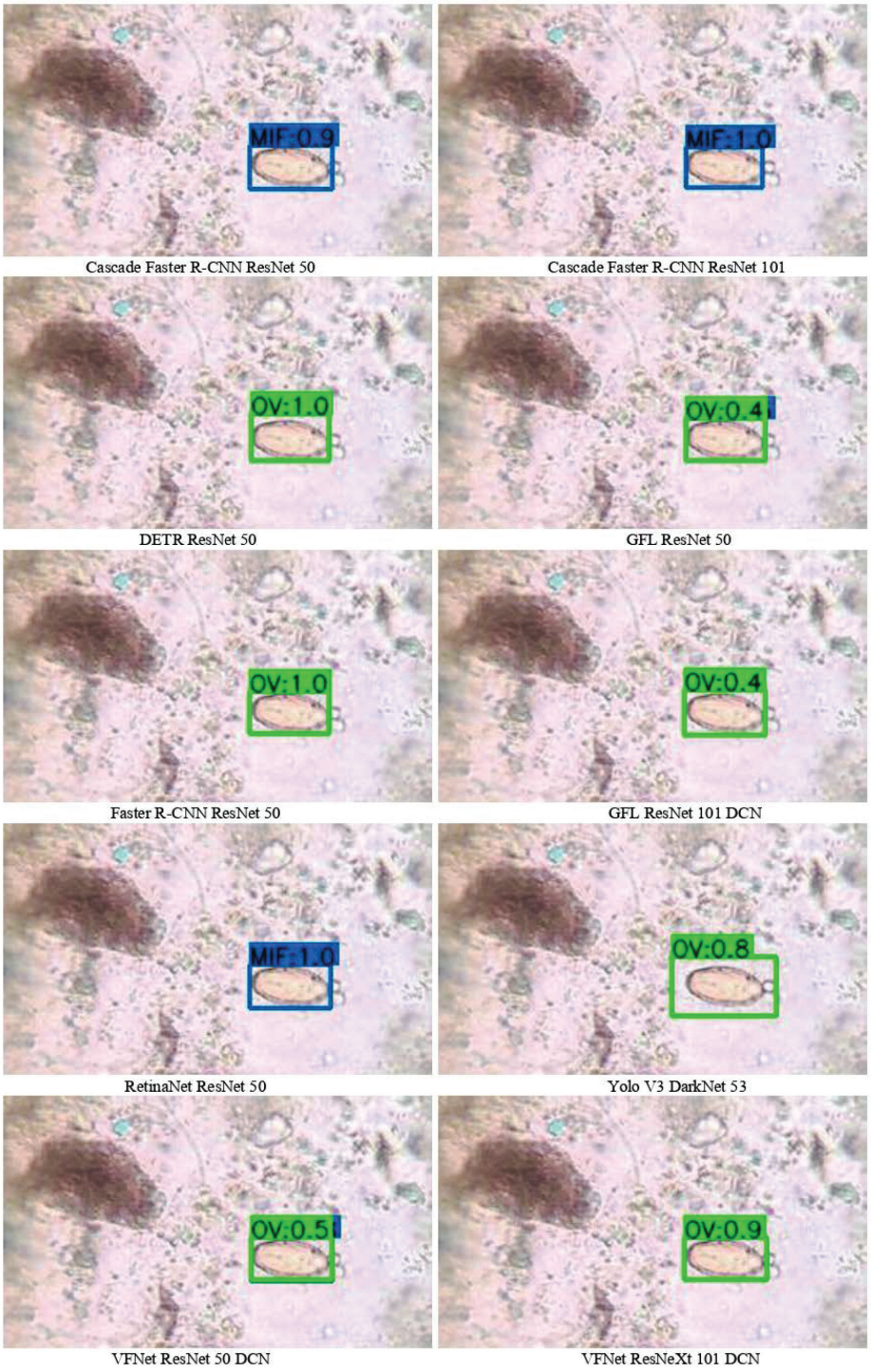


Fig. 6. Example result of an image with a MIF egg from the field data

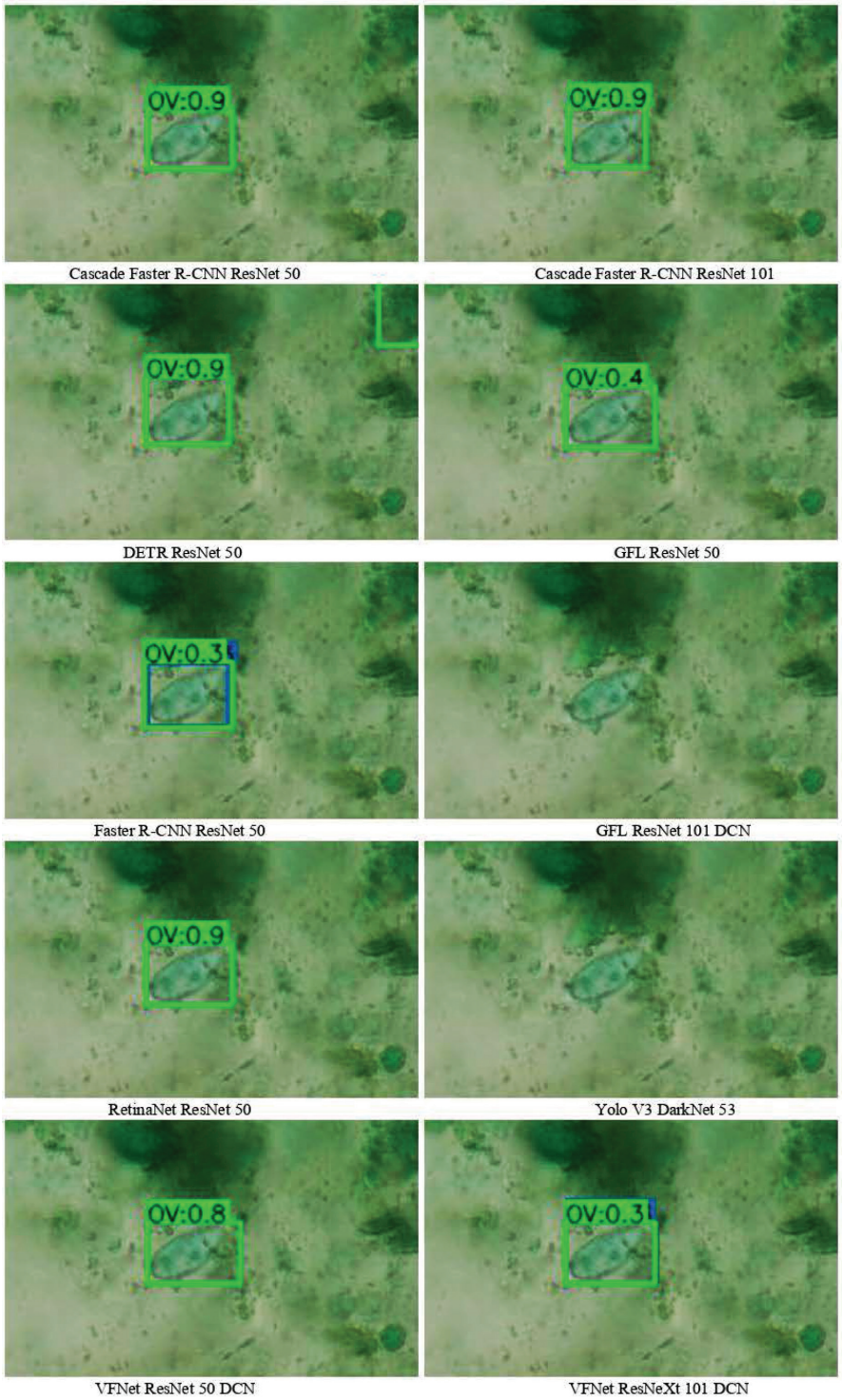


Fig. 7. Example result of an image with an OV egg from the field data

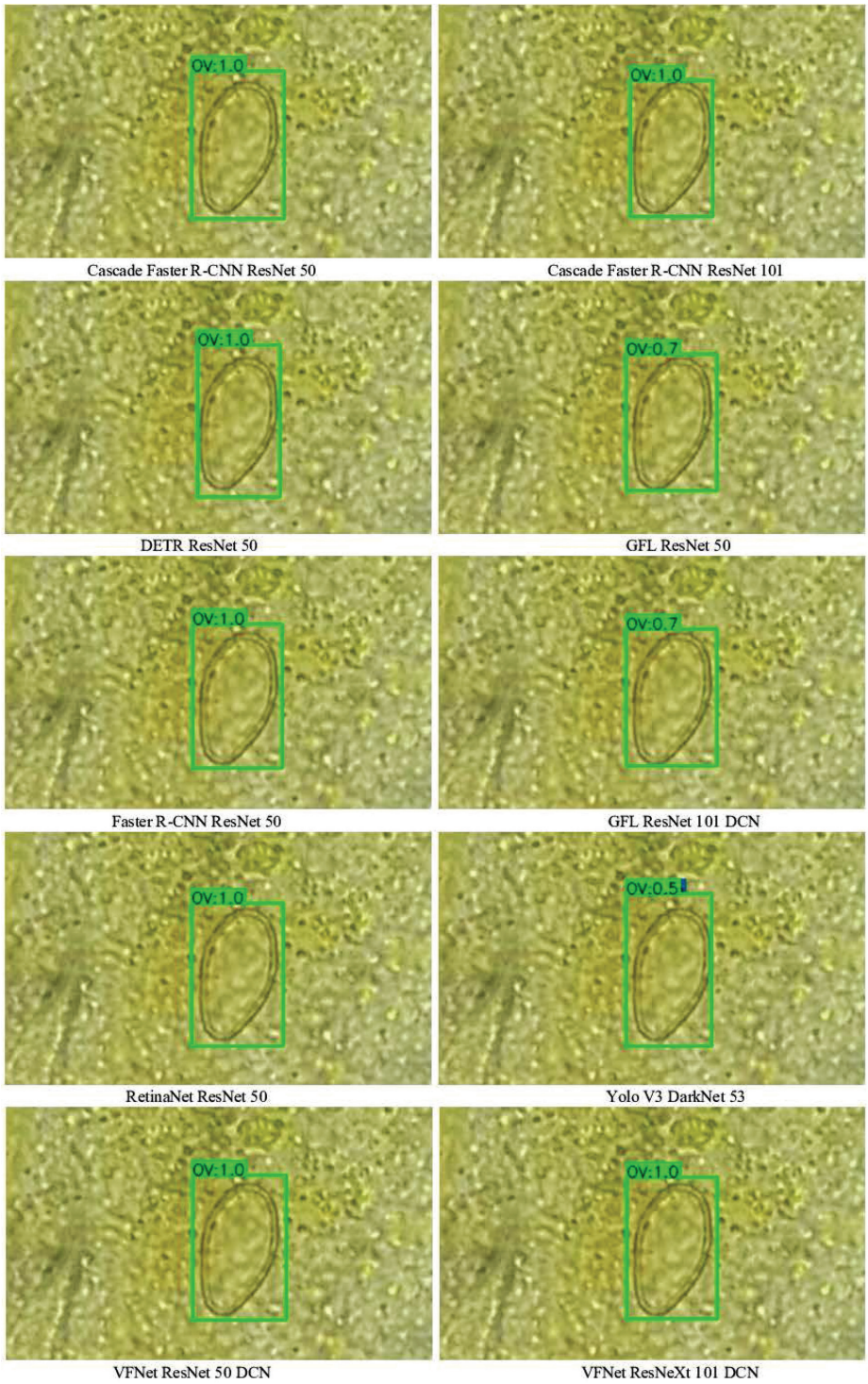


Fig. 8. Example result of an image with an OV egg from the field data

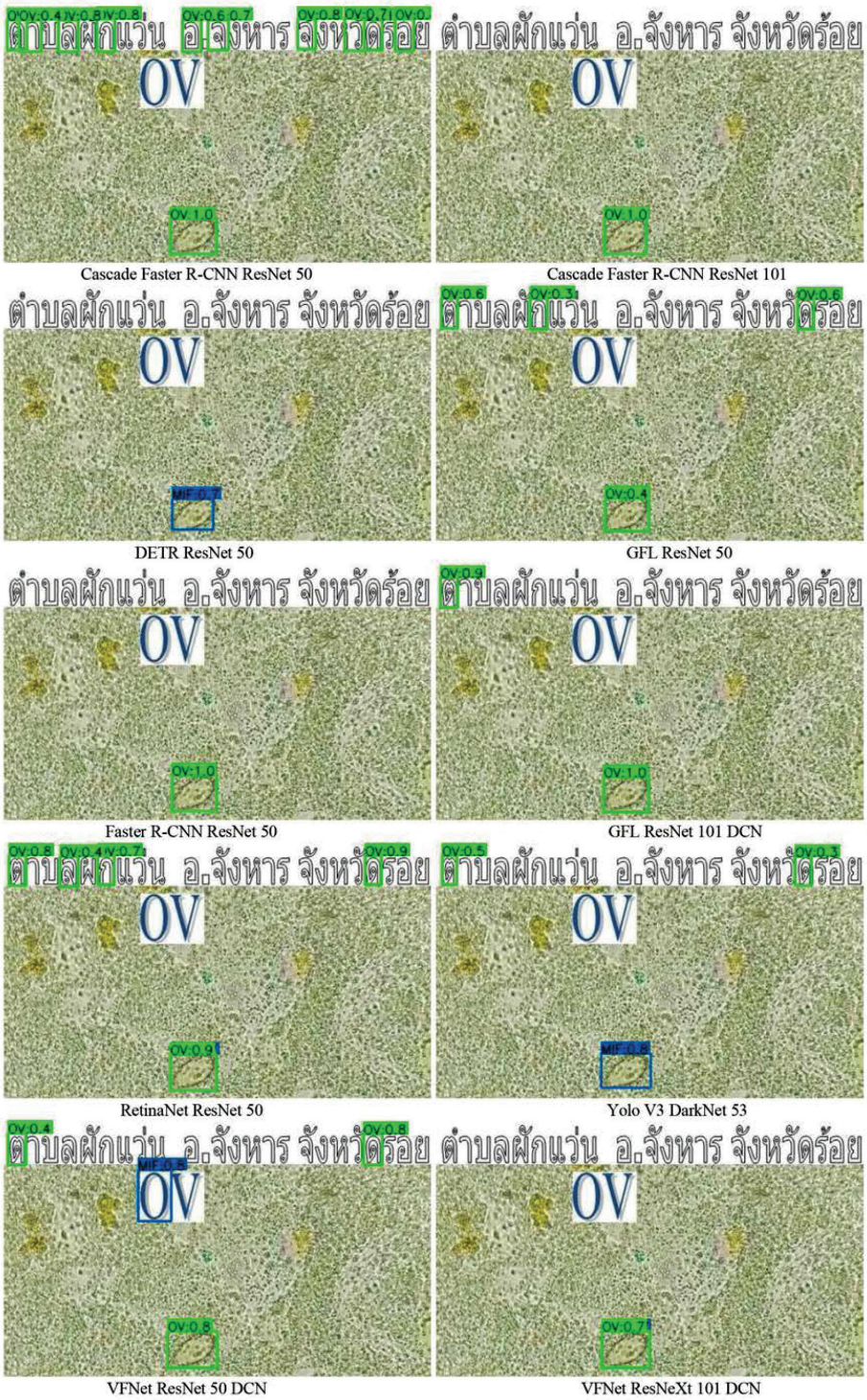


Fig. 9. Example result of an image with an OV egg and unrelated noises in the forms of overlay characters

C. Backbone Comparison

In the lab data, the smaller backbone performs significantly better than the bigger ones, with the baseline ResNet 50 outperforming other backbones in all techniques except VFNet in which the best is still ResNet 50 but with DCN.

In the field data, however, the results are somewhat different. The bigger backbone in Cascade Faster R-CNN performed noticeably better than the smaller counterpart in terms of classification. Also, for most models, the ones with a bigger backbone are less likely to predict false positives, especially on noises that are not in the training data such as overlay characters as shown in Fig. 9.

The DCN showed good results in the Lab data as in Table I but no noticeable difference in the field data. The side-by-side comparisons of VFNet with ResNet 50 with and without DCN are shown in Fig. 10 and Fig. 11.

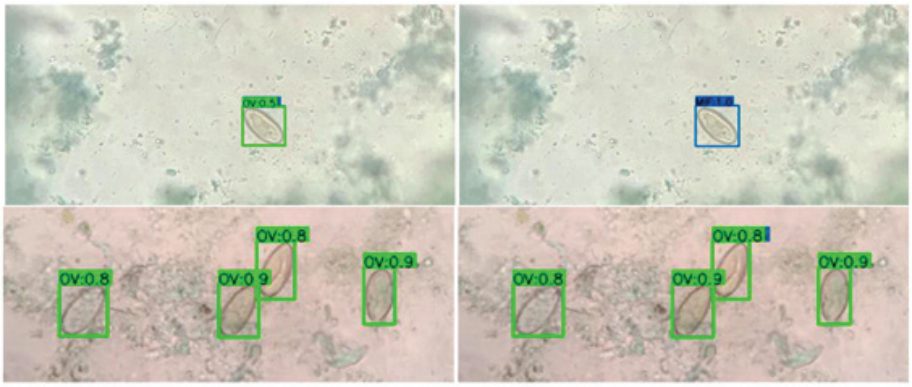
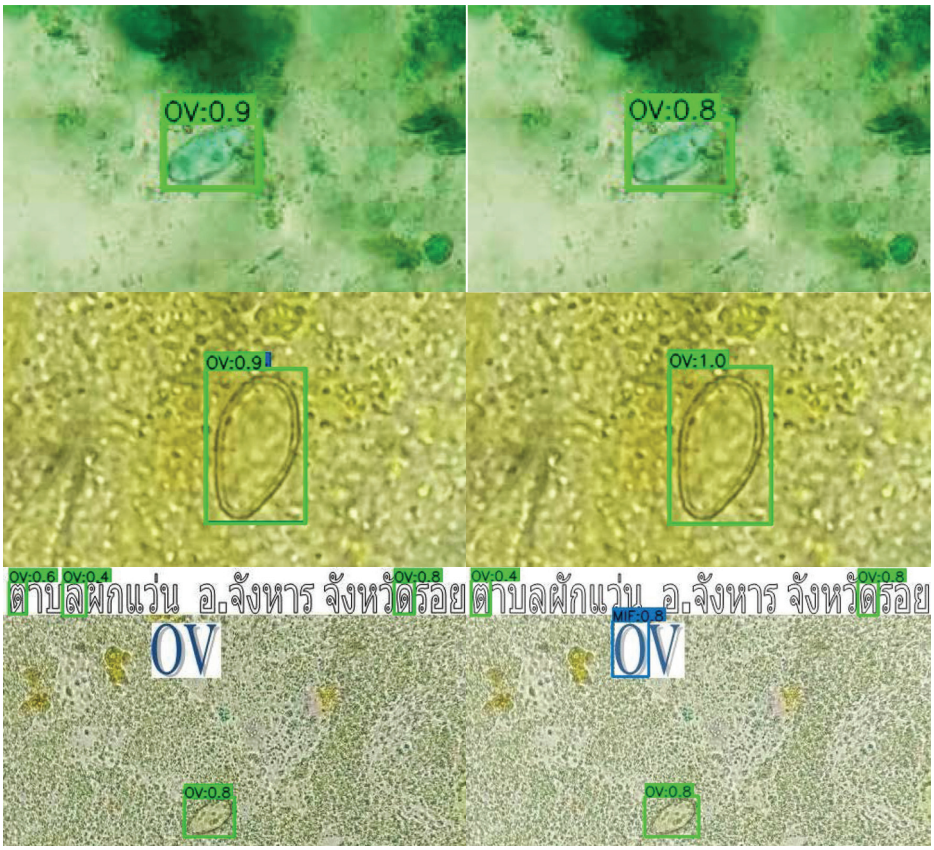


Fig. 10. Side-by-side comparison between base ResNet 50 and ResNet 50 DCN using VFNet technique on MIF images



Base ResNet 50 ResNet 50 DCN

Fig. 11. Side-by-side comparison between base ResNet 50 and ResNet 50 DCN using VFNet technique on OV images

Conclusion

The task of detecting and classifying parasite eggs is certainly possible using object detection techniques. The performance of each technique however may be different from the usual object detection task like COCO, possibly due to the training data we had been different from the real data used for evaluation. We determined that Cascade Faster R-CNN [10] is the best-performing technique for this task primarily because of its classification accuracy in test data. The bigger backbones showed worse results than the smaller counterpart in the lab data but with manual evaluations on the test dataset, some techniques like Cascade Faster R-CNN and RetinaNet had better results with bigger backbones. DCN showed satisfactory results on evaluation data but no noticeable difference in the test data. These points may need further prove with fully labeled test data.

The improved versions of Focal Loss on GFL [14] and VFNet [15] had significantly improved performance over the original RetinaNet [7]. The Deformable DETR on the other hand did not work very well on our task even when compared to the original DETR. The Transformer based DETR [11] tended to predict the noises and also had poor classification accuracy resulting in a rather poor specificity score. We suspected that the model had to be fine-tuned on some unseen data first.

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., "Global Cancer Statistics 2018: Globocan Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," *CA: A Cancer Journal for Clinicians*, vol. 68, no. 6, pp. 394–424, Sep. 2018.

[2] J. M. Bruun, C. M. Kapel, and J. M. Carstensen. (2012, Jun. 21). *Detection and Classification of Parasite Eggs for Use in Helminthic Therapy*. [Online]. Available: https://ieeexplore.ieee.org/document/6235888

[3] Y. S. Yang, D. K. Park, H. C. Kim et al. (2001, Jun. 2), *Automatic Identification of Human Helminth Eggs on Microscopic Fecal Specimens Using Digital Image Processing and an Artificial Neural Network*. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/11396601/

[4] A. Akintayo, G. L. Tylka, A. K. Singh et al. (2021, Jul. 2). *A Deep Learning Framework to Discern and Count Microscopic Nematode Eggs*. [Online]. Available: https://www.nature.com/articles/s41598-018-27272-w

[5] K. Chen, J. Wang, J. Pang et al. (2019, Jun. 15). *MMDetection: Open MMLab Detection Toolbox and Benchmark*. [Online]. Available: https://arxiv.org/abs/1906.07155

[6] T. Y. Lin, M. Maire, S. Belongie et al. (2015, Jun. 15). *Microsoft Coco: Common Objects in Context*. [Online]. Available: https://cocodataset.org

[7] K. He, X. Zhang, S. Ren et al. (2015, Jun. 15). *Deep Residual Learning for Image Recognition*. [Online]. Available: https://arxiv.org/abs/1512.03385

[8] S. Xie, R. Girshick, P. Dollár et al. (2017, Jun. 20). *Aggregated Residual Transformations for Deep Neural Networks*. [Online]. Available: https://arxiv.org/abs/1611.05431

[9] X. Zhu, H. Hu, S. Lin et al. (2018, Jun.21). *Deformable Convnets V2: More Deformable, Better Results*. [Online]. Available: https://arxiv.org/abs/1811.11168

[10] T. Y. Lin, P. Goyal, R. Girshick et al. (2018). *Focal Loss for Dense Object Detection*. [Online]. Available: https://arxiv.org/abs/1708.02002

[11] J. Redmon and A. Farhadi. (2021, Jun. 20). *Yolov3: An Incremental Improvement*. [Online]. Available: https://arxiv.org/abs/ 1804.02767

[12] S. Ren, K. He, R. Girshick et al. (2021, Jun. 20). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. [Online]. Available: https://arxiv.org/abs/1506.01497

[13] Z. Cai and N. Vasconcelos. (2021, Jun. 21). *Cascade R-CNN: High Quality Object Detection and Instance Segmentation*. [Online]. Available: http://dx.doi.org/10.1109/tpami.2019.2956516

[14] N. Carion, F. Massa, G. Synnaeve et al. (2021, Jun. 20). *End-to-end Object Detection with Transformers*. [Online] Available: https://arxiv.org/abs/2005.12872

[15] A. Vaswani, N. Shazeer, N. Parmar et al. (2021, Jun. 2). *Attention is All You Need*. [Online] Available: https://arxiv.org/abs/1706.03762

[16] X. Zhu, W. Su, L. Lu et al. (2021, Jun. 21). *Deformable Detr: Deformable Transformers for End-to-End Object Detection*. [Online] Available: https://arxiv.org/abs/2010.04159

[17] X. Li, W. Wang, L. Wu et al. (2021, Jul. 2). *Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection*. [Online] Available: https://arxiv.org/abs/2006.04388

[18] H. Zhang, Y. Wang, F. Dayoub et al. (2021, Mar. 4). *VarifocalNet: An IoU-aware Dense Object Detector*. [Online] Available: https://arxiv.org/abs/2008.13367

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 has completed the doctoral thesis on online handwriting recognition at University 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.