Attacking the Out-of-Domain Problem of a Parasite Egg Detection In-The-Wild

Nutsuda Penpong[1,4], Yupaporn Wanna[1], Cristakan Kamjanlard[2], Anchalee Techasen[3], Thanapong Intharah[1,5]

Abstract

Out-of-domain problem (OO-Do) has been hindered machine learning models especially when the models are deployed in real-world situation. The OO-Do happens at the test time when a learned machine learning model have to make a prediction for an input data belongs to a class that has not been seen at the training time. In this work, we tackle the OO-Do in object detection task specifically a parasite egg detection model being used in real-world situation. First, we introduced an in-the-wild parasite egg dataset to evaluate the OO-Do-aware model. The in-the-wild parasite egg dataset was constructed by conducting a chatbot test session with 222 Medical Technology students, which contains 1,552 images uploaded through the chatbot, including 1,049 parasite egg images and 503 non-parasite egg (OO-Do) images. Moreover, we propose a data-driven framework for constructing a parasite egg recognition model for in-the-wild applications to address the issue. The framework describes how we use publicly available datasets to train the parasite egg recognition model about in domain and out-of-domain knowledge. Finally, we compare integration strategies for our proposed two-step parasite egg detection approaches on two test sets: standard and in-the-wild datasets. We also investigate different thresholding strategies for robustness to OO-Do data. In the experiments, we found that concatenating a

[1]Visual Intelligence Laboratory, Department of Statistics, Faculty of Science, Khon Kaen University, Khon Kaen, Thailand
[2]Cholangiocarcinoma Research Institute, Khon Kaen University, Khon Kaen, Thailand
[3]Faculty of Associated Medical Sciences, Khon Kaen University, Khon Kaen, Thailand
[4]First Author: Nutsuda.penpong@kkumail.com
[5]Corresponding Author: thanin@kku.ac.th

classification model that is fine-tuned to be aware of OO-Do after the object detection model and using Softmax and G-mean achieved outstanding performance for detecting parasite eggs in the two test sets. The framework gained 7.37% and 4.09% F1-score improvement from the baselines on Chula$_{test}$+Wild$_{OO-Do}$ dataset and in-the-wild parasite egg dataset, respectively.

Keywords: Out-of-Domain, Parasite Egg Detection, Computer Vision In-the-Wild, Data Driven Framework, Chatbot

1. Introduction

The out-of-domain problem (OO-Do) occurs when a test image comes from a class outside the training data. It differs from the well-known out-of-distribution problem.In this work, we clearly define that the OO-Do problem occurs when the test images come from a class outside the training set, while the out-of-distribution problem occurs when the test images are of one of the trained classes but come from different distributions from the training data [1]. Fig. 1 illustrates the differentiation between the OO-Do and the out-of-distribution problems in computer vision.

When presented with the OO-Do data, the trained predictor will always generate wrong answers; otherwise, there are some mechanisms to detect the OO-Do before the predictor makes a prediction. It is necessary to draw a clear border between out-of-distribution and out-of-domain because the problems need different treatments. The out-of-distribution needs a decent transfer learning strategy to allow the model to adapt to the extended distribution. However, OO-Do is needed to be treated differently depending on the tasks. For object detection in self-driving vehicle, the detection model should be aware of the OO-Do, *i.e.* self-driving vehicle that has never seen an elephant should not ignore it and run into it. The vehicle should be aware of an unknown object on the track and avoid a collision. On the other hand, for parasite egg detection in an open chatbot application, the detection model should completely ignore unseen classes, especially the ones that look similar to the trained class.

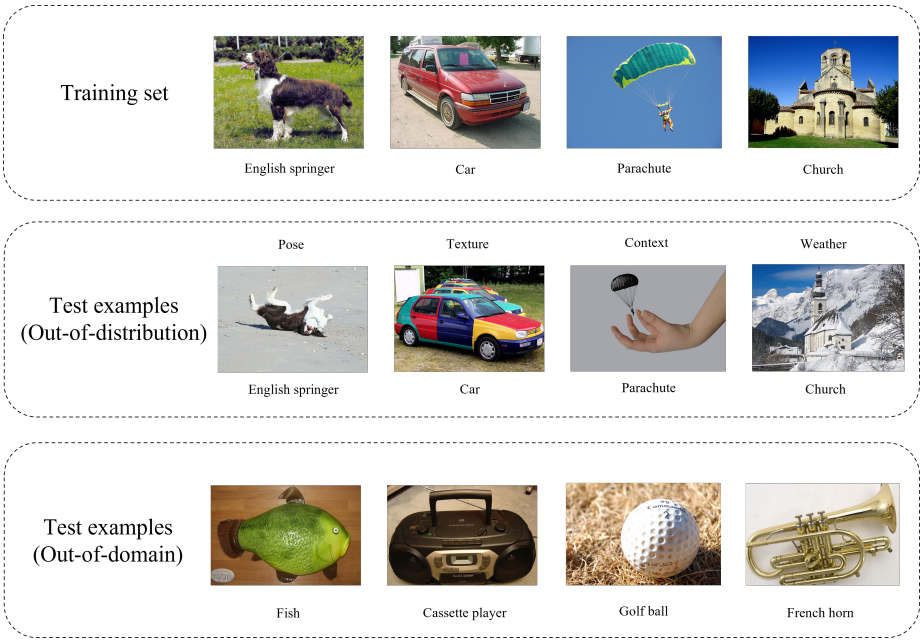


Figure 1: Differences between out-of-domain test data and out-of-distribution test data where the training data consists of four classes: English springer, car, parachute, and church.

Nowadays, AI is becoming a part of several people's working routines. When presented with a new AI application, one naturally tries to evaluate its intelligence and probe for its limitations. Testing an AI with OO-Do examples can give the impression to users whether the AI is too naive to be tricked by the users or able to pass this simple test. In this work, we design a chatbot to help Medical Technology students learn about parasites.

The main goal of the chatbot is to provide information about the parasites based on text queries. Moreover, the chatbot has a function to identify parasite eggs of input images captured from light microscopes. This parasite egg recognition module used a CNN-based object detection model without a mechanism to avoid the out-of-domain problem. It could detect and classify eleven parasite eggs in the microscopic input image. We ran a beta test of the chatbot on 222 students for two weeks and found that 727 OO-Do and 825 in-domain images were uploaded to the chatbot. The result raised the issue that when users ex-

ploited the OO-Do issue of the parasite egg recognition module, they uploaded as many OO-Do images as the expected domain images.

A baseline mechanism [2] to overcome the OO-Do is to introduce a threshold on prediction probability or confidence at the final prediction. In this work, we proposed a data-driven framework to solve the problem, starting by collecting data from different sources, mining the data for parasite egg recognition and OO-Do recognition, and evaluating two-step parasite egg detection algorithms that are aware of the OO-Do through rigorous testing experiments. Our two-step algorithms are based on the baseline mechanism where the classification step has a thresholding module to screen out the OO-Do samples. However, we couple an OO-Do-aware classification with a parasite egg detection model and investigate the best integration strategy.

In this paper, we constructed a dataset, which will be called the in-the-wild parasite egg dataset, by conducting a chatbot test session with 222 sophomore-associated Medical Technology students of Faculty of Associated Medical Sciences, Khon Kaen University, Thailand for two weeks in 2022. The dataset contains 1,552 images uploaded through the chatbot, which include 1,049 parasite egg images and 503 non-parasite egg images. The data was then manually labeled by medical technologists. This in-the-wild parasite egg dataset is used for evaluating parasite egg recognition algorithms.

To train our parasite egg recognition models, we used an existing dataset from Chula-ParasiteEgg-11 [3] to train CNN-based models. In the experiment, we compared two approaches by swapping the order of the models between the classification model and the detection model. We also investigated different thresholding strategies, namely without threshold strategy, with SoftMax threshold, and with ODIN threshold for robustness to OO-Do data.

From our experiments, concatenating a classification model which finedtune to aware of OO-Do with SoftMax thresolding stategy behind the object detection model achieved the best F1-score for detecting parasite eggs on Chula_{test}+Wild_{OO-Do} dataset. In addition, the framework achieved the second best F1-score on in-the-wild parasite egg dataset. The framework recov-

ered 14.97% precision while sacrificed 5.92% recall from the baseline method on Chula$_{test}$+Wild$_{OO-Do}$ dataset. In addition, the framework recovered 15.90% precision while sacrificed 10.02% recall from the baseline method on in-the-wild parasite egg dataset.

Contributions of this paper are as follows,

- We introduce two OO-Do test sets for parasite egg recognition.
- We describe a data-driven framework for constructing a parasite egg recognition models for in-the-wild applications.
- We investigate integration strategies for the two-step parasite egg detection on two OO-Do test sets.

The outline of this paper is as follows. We provide an overview of the out-of-domain problem and the motivation for tackling the problem in this Section. In Section 2, we summarize relevant literature on parasite egg recognition and proposed solutions for the out-of-domain problems in classification and detection tasks. Section 3 describes four datasets used in this research in detail, including the data collection process, classes, and number of images in datasets. There after, we introduce the data-driven framework to avoid the OO-Do in Section 4. The experiments to evaluate OO-Do are described in Section 5. A summary and discussion about the outstanding approach from the experiments for dealing with the OO-Do problem are presented in Section 6.

2. Related Work

In this section, we review existing related work in two aspects: parasite egg recognition and attempts to solve OO-Do in image recognition models: classification and object detection.

2.1. Parasite Egg Recognition

A digital image processing scheme [4] for protozoan parasite detection involves several steps that are applied sequentially color space transforming, gamma

equalization, median–mean filter, two-classes edge enhancement, morphological opening operation and the largest independent component detection. The experiment was tested on 112 protozoan parasite microscopic images provided by Dr. Tsai at National Chung-Hsing University, which the proposed scheme achieved an average correct rate of 96.64%. The average error rates were about 0.04 for misclassification error (ME), 0.45 for region non-uniformity (RN), and 0.06 for relative foreground area error (RFAE).

Mask R-CNN model [5] trained using images of both uninfected and *Plasmodium falciparum*-infected red blood cells to generate segmentation masks on top of bounding box classifications and count the number of *Plasmodium falciparum*-infected red blood cells. Dataset consists of 297 images, which contains five classes, but only four of them are labeled: uninfected reticulocyte, ring, trophozoite, and schizont. The Normocyte class, which represents healthy red blood cells, was not labeled to save time on manual annotations due to the large number of healthy normocytes present in each smear. The model achieved 94.57% accuracy in detecting infected cells with only 0.55% error.

A hybrid approach [6] that combines DS1, fast handcrafted image feature extraction and support vector machine (SVM) with DS2, Vgg-16 deep neural network, to improve the accuracy of image classification. The datasets used for the experiments consist of a total of 51,919 images of the 15 most common species of human intestinal parasites in Brazil, as well as similar fecal impurities. The results of the experiments demonstrate that the proposed hybrid approach achieves high levels of accuracy, with an average Cohen's Kappa of 94.9%, 87.8%, and 92.5% on helminth eggs, helminth larvae, and protozoa cysts, respectively.

A new dataset called Chula-ParasiteEgg-11 [3] was introduced for the ICIP 2022 Challenge, which contains 11 types of parasitic eggs from fecal smear samples and a total of 13,750 microscopic images. The training and testing datasets were randomly divided, with 1,000 images per class for training and approximately 250 images per class for testing. All training images were released along with 2,200 unknown labelled testing images for the competition. All methods used deep learning techniques such as YOLOv5, Fast-RCNN, EfficientDet, Cas-

cade R-CNN, CBNetV2, CenterNet2, Task-aligned One-stage object Detection (TOOD), and RetinaNet. The winner of the challenge, NEGU, achieved a mIoU of 0.942 and F1-score of 0.995 on the hidden data.

Transformer-based architectures such as Swin-Transformer-Base and Swin-Transformer-MoBY-Tiny is used as feature extraction backbones for object detection methods namely DETR, Deformable-DETR, Mask-RCNN, Cascade Mask-RCNN [7]. The Chula-ParasiteEgg-11 dataset has been used to evaluate the ability of different detection methods and backbones in solving the parasitic egg detection task in microscopic images. The result show that Cascade Mask-RCNN with the SwinTransformer-Base backbone achieving up to 0.875 mIoU score and 0.955 F1-score on the test subset.

A deep learning model uses EfficientDet object detector with EfficientNet-v2 backbone to localize and classify parasitic eggs and EfficientNet-B7+SVM model to enhance classification performance [8]. The model was evaluated using a new dataset called Chula-ParasiteEgg-11, which contains 11 different types of parasitic eggs. The reported accuracy of 92% and F1-score of 93% indicate that the model performed well on this task.

The combination of FasterRCNN, TOOD, YOLOX, and Cascade with Swin-Transformers [9] reduce the rate of missed detections. Each image is run through all five models, and then the highest confidence score is used to determine the final detection result. The dataset used for evaluation is the Chula-ParasiteEgg-11 dataset, and the proposed method achieves an IoU score of 0.915 and an F1 score of 0.974. This is a significant improvement over the performance of each individual method.

N. Butploy [10] used a convolutional neural network (CNN) architecture with varying numbers of convolution layers to identify the optimal number of layers for each type of egg and selected three specific convolution layers for further experimentation and applied ReLu activation, maxpooling, and dropout regularization techniques to fine-tune the CNNs. The dataset used in the study was freshly prepared from A. lumbricoides-infected stool samples obtained from the Department of Parasitology at the Faculty of Medicine Khon Kaen University in

Thailand. The model were able to achieve an accuracy of 93.33% in classifying the three types of A. lumbricoides eggs using deep learning techniques.

2.2. Out-of-domain Problem

2.2.1. OO-Do in classification

D. Hendrycks [2] presented thresholding the maximum softmax probability of a model's output on a given an example. Suppose the maximum softmax probability is below a certain threshold. In that case, the example is classified as out-of-distribution, and if it is above the threshold, the example is classified as correctly classified. The baseline method was evaluated on several architectures such as computer vision, natural language processing, automatic speech recognition, and numerous datasets such as MNIST, TIMIT, CIFAR-10, and IMDB.

OpenMAX [11] proposed a way to handle unknown classes by adding an extra class to the classifier's output, called the "unknown" class, by measuring the mean activation vector for each known class and using it to attenuate the softmax probabilities. The paper evaluates OpenMAX on the ILSVRC 2012 dataset, which has 1K known classes. The results show that OpenMAX improves the accuracy of open set recognition by nearly 4.3% over Softmax with an optimal threshold and 12.3% over the base deep network.

P. Perera [12] proposes a novel approach to training deep neural networks for multi-label image classification. Instead of using the standard softmax layer and cross-entropy loss, they use a membership loss function that directly models the probability that each class is present in the image. The proposed method is evaluated on four popular image datasets: Caltech256, Caltech-UCSD Birds 200 (CUB 200), Stanford Dogs, and FounderType-200. Two popular CNN architectures, VGG16 and AlexNet, are used for evaluation. The results show that the proposed method outperforms the standard softmax/cross-entropy approach on all four datasets, with the best performance achieved on Caltech256, where an AUC of 0.947 is obtained for the ROC curve.

P. R. Mendes Júnior [13] introduced Nearest Neighbor Distance Ratio (NNDR)-based Open-Set NN (OSNN) classifier, which uses a threshold to classify test samples into known or unknown classes based on their nearest neighbors and two new evaluation measures, normalized accuracy (NA) and open-set f-measure (OSFM), to assess the quality of methods in multi-class open-set recognition problems. The datasets used in the experiments are 15-Scenes, Letter, Auslan, Caltech-256, ALOI, and Ukbench. Based on the experimental results presented, the proposed OSNN method performs better than the baseline classifiers in most cases for several datasets.

19 different unsupervised anomaly detection methods, including k-nearest neighbors (KNN), support vector machines (SVM), and clustering methods, are commonly used in unsupervised anomaly detection [14]. KNN identifies anomalies by finding the samples farthest away from their nearest neighbors. SVM constructs a boundary around the normal data points; samples outside this boundary are considered anomalous. Clustering methods group similar data points together and consider data points that do not belong to any cluster as anomalous. The evaluation of unsupervised anomaly detection methods on various datasets such as Breast Cancer Wisconsin, Speech Accent Data, and Object Images (ALOI). In general, nearest-neighbor-based algorithms, such as KNN, perform better than clustering algorithms.

Reciprocal Point Learning (RPL) [15] presented the concept of reciprocal points. The activations of a model's penultimate layer are fed into a softmax function. The activations are computed based on the distance of the input image to each reciprocal point. If an input image is close to a reciprocal point, its activation will be low for the corresponding class and otherwise. The paper evaluates RPL on several standard image classification datasets, including MNIST, SVHN, CIFAR10, CIFAR+10, CIFAR+50, and TinyImageNet, which sample both known and unknown classes. The results show that RPL outperforms other state-of-the-art methods based on encoder architecture in all datasets, especially in the unknown classes.

The MC-Dropout [16] can be used to estimate the model's uncertainty. This

can be achieved by sampling the dropout masks at test time and obtaining a distribution of predictions. This distribution's variance can be used to measure the model's uncertainty, with higher variances indicating greater uncertainty. In the context of classification tasks using the MNIST dataset, it has been shown that MC-Dropout can significantly outperform other models regarding both root mean squared error (RMSE) and test log-likelihood.

DeepEnsemble [17] uses an ensemble of five neural networks trained with an adversarial-sample-augmented loss to measure predictive uncertainty and reject samples with high uncertainty. The MNIST, SVHN, and ImageNet datasets are commonly used benchmarks for evaluating the performance of classification models. DeepEnsemble have been shown to lead to lower classification error and better predictive uncertainty compared to probabilistic backpropagation and MC-Dropout, as evidenced by lower NLL and Brier score.

A modified version of the PixelCNN generative model [18] can be used as a density estimator for anomaly detection tasks. The model uses a discretized logistic mixture likelihood function to estimate the probability density of the input samples. A density estimator that can accurately model the distribution of the input data, the model can flag low measure samples as anomalies. The paper evaluate the performance of PixelCNN++ on the CIFAR-10 dataset The PixelCNN++ model achieved state-of-the-art results in terms of log-likelihood on the CIFAR-10 dataset.

2.2.2. OO-Do in detection

J. Nitsch [19] a method for detecting out-of-distribution (OOD) samples in autonomous driving using a GAN-based approach. The key idea is to train a GAN architecture to synthesize out-of-distribution data that the discriminator would incorrectly classify as in-distribution (ID) data. The discriminator is optimized to distinguish between the generated OOD samples and real ID samples. The discriminator loss, the generator loss and the object classification loss are jointly trained. The proposed approach employs an classification model and post hoc statistics measures to detect OOD samples. The proposed

approach is evaluated on two real-world automotive in-domain datasets: KITTI and nuScenes, and ImageNet is chosen as an out-of-domain dataset. The results show that the proposed approach achieves the best AUPR-in performance, with a score of 89.86% on KITTI and 91.51% on nuScenes.

Y. Li and J. Košecká [20] proposed a new method for detect out of distribution objects in autonomous driving. The method uses a two-step proposal segmentation to detect pixels belonging to background classes (e.g. road, vegetation, building), which a radial basis function network (RBFN) will produce high uncertainty in classifying all object's pixels into any known class. The proposed method was trained on the Cityscapes dataset and evaluated on three outdoor scenes datasets (L&F, FS, and RA) and two indoor scenes datasets (ADE20K and AVD). The evaluation results showed that the proposed method achieved an accuracy of 92.1% on the L&F dataset.

Monte-Carlo DropBlock (MC-DropBlock) [21] on the convolutional layer of YOLO model that is capable to exhibit high uncertainty on OO-Do data and do not produce high confidence to test data from different distribution from training dataset. The model is trained on the Pascal VOC dataset for ID dataset and testing on the COCO dataset for OO-Do dataset. Experimental results show that mAP values for Object detection on the COCO dataset Entropy is the established measure of capturing uncertainty. The mAP values for object detection on the COCO dataset Entropy are the established measure of capturing uncertainty. The experimental results reveal that the YOLOv4/v5 models achieve the best mAP values for train-time DropBlock at 65% and 71%, respectively.

YolOOD [22] that convert the YOLO object detection network into a multi-label image classifier. Detection heads are replaced with a 3D tensor. A single cell grid is identical to the original YOLO and predicts the objectiveness score and classes. The experiment is evaluated on PASCAL VOC and MS-COCO for in-distribution datasets and a subset of images from the ImageNet-22K and Textures for out-of-distribution datasets. YolOOD achieved 98.42 for AuPR and state-of-the-art performance compared to the baselines.

The margin entropy (ME) loss for 2D object detection [23] detect out-of-distribution samples in Safety-Critical applications. A subset of the Amazon AOT dataset is used as in-distribution dataset to train the neural network. The proposed method outperforms the baseline by 129 % by means of separability.

Previous research involves detecting out-of-domain samples in classification and modifying a pre-trained model for object detection to make it capable of detecting out-of-domain samples. The modifications are composed of changing the original model's architecture, loss function, or training data. However, our proposed two-step approaches aim to maintain the performance of the original model while specifically enhancing its ability to detect objects in out-of-domain samples without requiring any significant modifications to the original model. These two-step approaches typically involve a separate process or model to identify out-of-domain samples and then use this information to improve the detection performance of the original model on those out-of-domain samples.

3. Datasets

Four datasets are involved in constructing our data-driven parasite egg recognition frameworks that are aware of OO-Do. The conceptual model of the four datasets in the data-driven framework are illustrated in Fig. 2. The main dataset is used for training, fine-tuning and in-distribution testing. The in-the-wild testing dataset is ideally collected during the real-world testing of the recognition model trained with the main training dataset and will be used for testing the model in real-world situation. The out-of-domain dataset is considered out-of-domain data which contains images that are not related to the task. The hard negative out-of-domain dataset contains images of a closely related task which considered having similar appearance to the main task. Details of the datasets used in our experiment are described next.

3.1. Chula-ParasiteEgg-11

For the parasite egg detection task, we used Chula-ParasiteEgg-11 [3] dataset [24] as the main dataset. Chula-ParasiteEgg-11 is a large set of microscopic im-

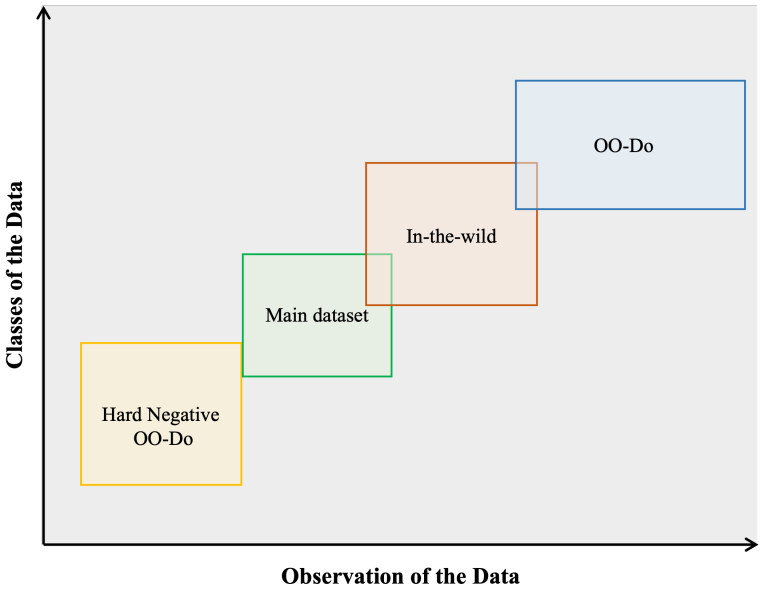


Figure 2: Conceptual model of the relation between four datasets in the data-driven framework for training an object detection that aware of the out-of-domain problem. Hard negative OO-Do are the dataset that has similar observation, appearance in the case of computer vision task, to the main task, but has no common class. The main dataset is the dataset that only contains data from the task-focused classes. In-the-wild are the dataset which aimed as test set that contains both data from the task-focused classes but might have slightly different observation and data from unseen classes. OO-Do are the dataset that contains wilder set of unseen classes

ages, which consists of 13,200 microscopic images taken through different devices ranged from mobile phone cameras through DSLR cameras. It containing 11 types of parasitic eggs from fecal smear samples, namely *Ascaris lumbricoides*, *Capillaria philippinensis*, *Enterobius vermicularis*, *Fasciolopsis buski*, Hookworm egg, *Hymenolepis diminuta*, *Hymenolepis nana*, *Opisthorchis viverrine*, *Paragonimus* spp, *Taenia* spp. egg and *Trichuris trichiura*. It comprises 11,000 images of the Chula-ParasiteEgg-11 training set and 2,200 images of the Chula-ParasiteEgg-11 unknown test set. The Chula-ParasiteEgg-11 training set is well-balanced, with each type of parasite egg having 1,000 images.

In our experiment, we built the Chula_{train} by sampling 800 images from each

class from the Chula-ParasiteEgg-11 training set. The rest 2,200 images were formed the Chula$_{val}$. Finally, the Chula$_{train}$ contains 8,800 images, while the Chula$_{val}$ contains 2,200 images. These training and validation sets were used to train and finetune the models. Lastly, the Chula-ParasiteEgg-11 unknown test set was used to test the constructed models' performance, which will call it Chula$_{test}$.

3.2. In-the-wild parasite egg dataset

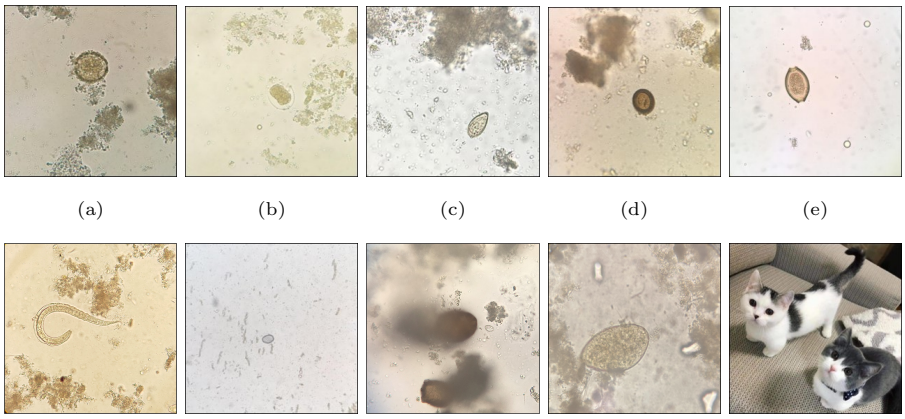
Table 1: Distribution of images in the in-the-wild parasite egg dataset.

In-domain images		Out-of-domain images	
Group	Count	Group	Count
Ascaris lumbricoides	188	Adult parasite	31
Hookworms	98	Arbitrary	139
Opisthorchis viverrine	128	Artifact	70
Taenia spp.	281	Unclear	294
Trichuris trichiura	130	Other parasite eggs	193
sum	825	sum	727

For the in-the-wild testing dataset of the parasite egg detection task, we obtained images through a real-world test session. A total of 1,552 parasite egg images were collected from a test session of parasite egg learning chatbot which ran on 222 students from the Faculty of Associated Medical Sciences, Khon Kaen University, for two weeks in 2022.[6] The microscopic images taken with mobile phone cameras through eyepieces of microscopes were uploaded to the chatbot to identify the parasite eggs in the images. The parasite egg detection model was trained to recognize 11 parasite egg, namely *Ascaris lumbricoides*, *Capillaria philippinensis*, *Enterobius vermicularis*, *Fasciolopsis buski*, Hookworm egg,

[6]This study was conducted according to the guidelines of the Declaration of Helsinki and the ICH Good Clinical Practice Guidelines and approved by the Human Ethics Committee of Khon Kaen University, Khon Kaen, Thailand,(HE642104; approved date June 21, 2021.)

Hymenolepis diminuta, *Hymenolepis nana*, *Opisthorchis viverrine*, *Paragonimus* spp, *Taenia* spp. egg and *Trichuris trichiura*. Table 1 illustrates the distribution of classes of the uploaded images. We can roughly divide the images into in-domain and out-of-domain groups, see figure 3. For in-domain images, 825 images consisted of five types of learned parasite eggs.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 3: **(a)** *Ascaris lumbricoides* **(b)** Hookworms **(c)** *Opisthorchis viverrine* **(d)** *Taenia* spp. **(e)** *Trichuris trichiura* **(f)** Adult parasite **(g)** Artifact **(h)** Unclear image **(i)** Other parasite egg **(j)** Arbitrary

The images that did not belong to the trained parasite egg classes, through out this paper will be called Wild_{OO-Do}, consisted of 727 images which can be grouped into adult parasite images, arbitrary images, artifact images, unclear images, and other parasite egg images. Adult parasite images included images of adult parasites uploaded to the system, such as *Strongyloides stercoralis*. Arbitrary images referred to images irrelevant to microscopic images, such as human faces, cats, food, and buildings. Artifact images were microscopic images of pseudoparasites, including undigested leftovers or coincidentally or purposely ingested nonparasitic organisms or their parts. Unclear referred to images which were not clear enough to be labeled as any specific class. Other parasite egg images included parasite images that did not belong to the 11 trained classes, such as *Echinostoma* eggs and Minute Intestinal Flukes (MIF) eggs. Examples

of images from the OO-Do group are illustrated in Fig. 3f-3j.

3.3. Imagenette$_{OO-Do}$

For the out-of-domain dataset, we use a subset of the Imagenet dataset as a representative dataset. The Imagenette dataset [25] is a subset of 10 easily classified classes from Imagenet [26]. The classes in the Imagenette dataset consist of French horn, English springer, cassette player, chain saw, church, tench, garbage truck, gas pump, golf ball, and parachute. Please note that all images in the dataset are not related to parasite eggs nor are microscopic images. We randomly sampled 40% of the images from each of seven classes, namely French horn, English springer, cassette player, church, gas pump, golf ball, and parachute, from the validation set of the Imagenette dataset. 1,100 images were sampled and assigned as the Imagenette$_{OO-Do}$. This dataset is used to finetune the model.

3.4. Malaria$_{OO-Do}$

For the hard-negative out-of-domain dataset, we carefully select a microscopic image dataset that does not contain any relevant object in the parasite egg detection task. The malaria microscopic image dataset, which is one subset of the Broad Bioimage Benchmark Collection (BBBC) dataset [27] were used as the hard-negative out-of-domain dataset for the parasite egg detection task. The dataset comprises 1,364 malaria microscopic images labeled as two uninfected cell classes, namely red blood cells (RBCs) and leukocytes, and four infected cell classes, namely gametocytes, rings, trophozoites, and schizonts. We used this dataset to finetune our model and named it Malaria$_{OO-Do}$.

4. Proposed Data Driven Frameworks

We propose and evaluate two data-driven frameworks. The frameworks comprise of the data-driven model construction process and recognition model architecture. The distinction between the two frameworks is the order of the recog-

nition models: OO-Do image classification model and object detection model. This section describes our proposed data-driven steps for both frameworks.

4.1. Classification-first

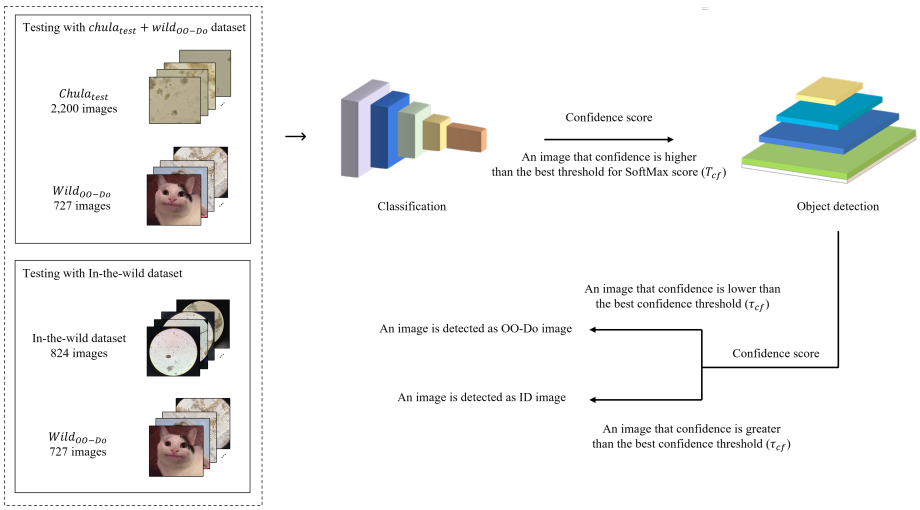


Figure 4: Overview of testing process for the classification-first framework.

For the classification-first framework, we use the OO-Do-aware classification model to screen the input images before passing the images to the object detection model, the architecture is shown in Fig. 4. Images are first passed through the classification model, which predicts the class and provides the image's class probability (SoftMax score) as output. Next, the image with a class probability below the threshold T_{cf} is regarded as out-of-domain, and the image with a class probability above the threshold T_{cf} is passed to the object detection model to obtain the bounding boxes with a confidence score. If the confidence score for a bounding box is greater than or equal to the threshold τ_{cf}, the object is detected. On the other hand, if the confidence score is less than the threshold τ_{cf}, the image is considered not detected.

The framework focuses on thresholding strategy for the classification T_{cf} and for the object detection τ_{cf} to determine whether the object is one of the

training classes and the image is in-domain or out-of-domain, respectively. The overall training process is shown in Fig. 5.

For the parasite egg image classification model, we trained the EfficientNet-B2 [28] model with the transfer learning technique on the Chula$_{train}$ data, where the pre-trained weights were trained on the ImageNet dataset. To find the best threshold, T_{cf}, for the class probability, we combine three datasets: the Chula$_{val}$, the Imagenette$_{OO-Do}$, and the malaria$_{OO-Do}$ dataset. The system first runs classification on these datasets using the trained EfficientNet-B2 model. The classification model will output confidence scores class labels for each image. We plot an ROC curve using the images' confidence scores and class labels. To determine the optimal threshold that balances false positive and true positive rates, we use the G-mean, a metric that considers both sensitivity and specificity. The G-mean is the square root of the product of sensitivity and specificity defined as

$$G - Mean = \sqrt{Sensitivity * Specificity} \tag{1}$$

The threshold with the highest G-mean is the optimal threshold used to distinguish between in- and out-of-domain images in our experiments.

For the parasite egg detection, we trained object detection with a pre-trained YOLOv5 [29] model on the Chula$_{train}$ data. For the object detection threshold, we find the best detection threshold τ_{cf} that produced the best F1-score for the Chula$_{val}$ data.

4.2. Classification-later

In contrast to the first framework, the classification-later framework places the OO-Do-aware classification model behind the object detection model, shown in Fig. 6. Images are first input into the object detection model, which generates a set of prediction bounding boxes and their corresponding confidence scores. Each image is cropped based on the bounding boxes with the confidence scores higher than the detection threshold value τ_{cl}. The cropped image is then

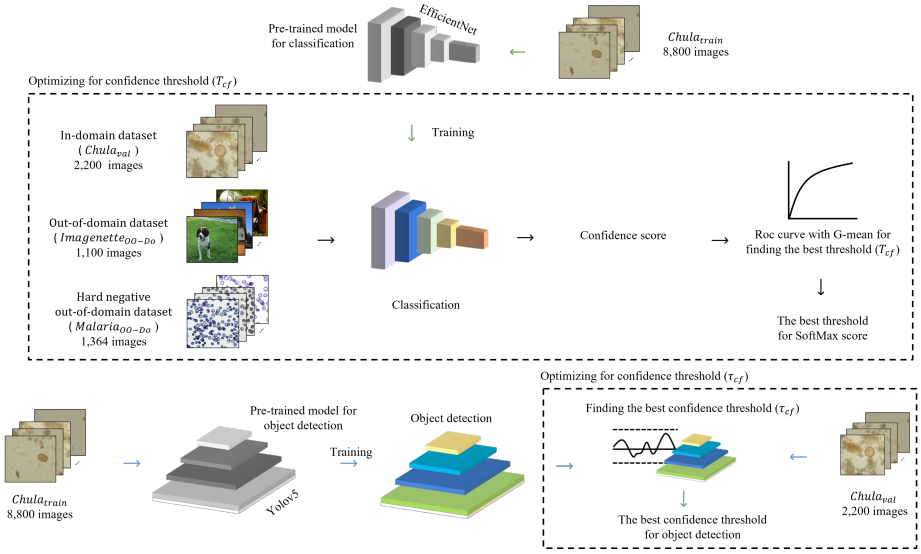


Figure 5: Overview of training and finding thresholds for classification-first framework.

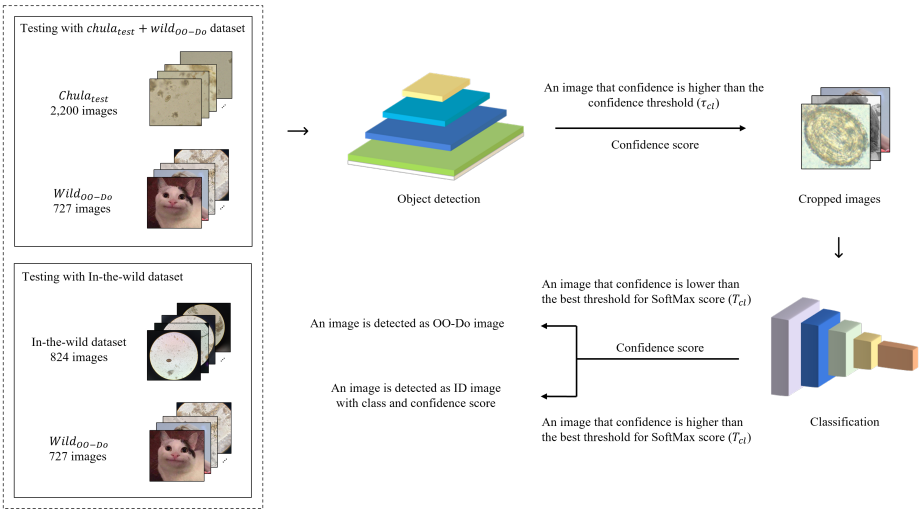


Figure 6: Overview of testing process for the classification-later framework.

classified by a trained classification model. The image that has class probability less than the classification threshold T_{cf} is deemed as an OO-Do image.

The data-driven framework for the classification-later is illustrated in Figure 7. We trained the parasite egg detection model by performing the transfer

learning step on the pre-trained the YOLOv5 object detection model with the Chula$_{train}$ data. The detection thresold τ_{cl} is chosen by selecting the highest confidence threshold that produces zero false negatives for the Chula$_{val}$ data.

We trained the EfficientNet-B2 classification model with a transfer learning strategy on a cropped version of the Chula$_{train}$. The pre-trained weights for EfficientNet-B2 were obtained from training on the ImageNet dataset. To set the classification threshold T_{cf}, the Chula$_{val}$, Imagenette$_{OO-Do}$, and malaria$_{OO-Do}$ dataset feed into an YOLOv5 object detection model. Finally, we use the images' confidence scores and class labels to plot a ROC curve. The classification threshold T_{cf} is then set to the threshold that produces the highest G-mean score.

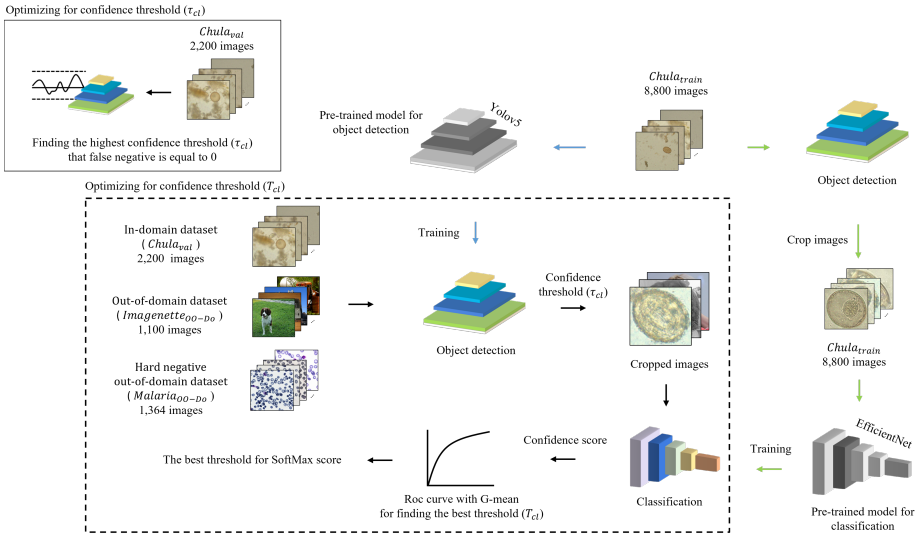


Figure 7: Overview of training and optimizing for thresholds for the classification-later framework.

5. Experiments and Results

5.1. Experimental setup

This section describes the specific setup and hyperparameters used for training two parasite egg recognition models: EfficientNet-B2 for image classification

and YOLOv5 for object detection.

For EfficientNet-B2, the model is trained using Adam optimizer and cross-entropy loss for 200 epochs with a batch size of 32, a dropout rate of 0.3, and a learning rate of 0.00125. For YOLOv5, the initial weight is yolov5l6, and the model is trained for 100 epochs with a batch size of 2, using the default parameters of the YOLOv5 setup. Six model variations were trained on the same dataset and tested on the Chula$_{test}$+Wild$_{OO-Do}$ dataset and the in-the-wild parasite egg dataset. SoftMax thresholding involves using the SoftMax function to normalize the confidence scores and then setting a threshold for accepting predictions. ODIN thresholding involves using a temperature scaling technique to calibrate the confidence scores before applying a threshold.

To summarize, the six model variations for the parasite egg recognition framework are:

- Object detection is a baseline object detection.
- Classification-later (without thresholding strategy) is a classification-later framework without performing OO-Do-aware thresholding stategies on both object detection step and classification step.
- Classification-later (SoftMax threshold) is a classification-later framework described in subsection 4.2. In this framework, the confidence score produced by the classification model is the product of SoftMax function.
- Classification-later (ODIN threshold) is a classification-later framework with the thresholding strategy but the classification model is modified with ODIN [30].
- Classification-first (SoftMax threshold) is a classification-first framework described in subsection 4.1. In this framework, the confidence score produced by the classification model is the product of SoftMax function.
- Classification-first (ODIN threshold) is a classification-first framework with the thresholding strategy but the classification model is modified with ODIN [30]

5.2. Problem of OO-Do

In this subsection, we experimentally show that OO-Do image hurt the precision of the baseline parasite egg detection model. The precision measure of how accurate the model made positive predictions, i.e., the proportion of positive predictions that are correct. Lower precision means that the model is making more false positive predictions, which can reduce its reliability and its usefulness for users. First, we performed an experiment by adding the Wild_{OO-Do} to the Chula_{test} to see how did the object detection model trained with Chula_{train} hold when the model was presented with OO-Do data along with in-distribution data. Second, we evaluated the baseline detection model through the in-the-wild parasite egg test set which composed of OO-Do data and out-of-distribution data, the data were acquired differently such as different specimen preparations and different image acquisition tools.

Table 2 and Table 3 show that precisions decreased significantly when tested on Chula_{test}+Wild_{OO-Do} test set and in-the-wild parasite egg test set. Without surprise, overall performance of the models on in-the-wild parasite egg test set were significantly lower than the Chula_{test}+Wild_{OO-Do} test set because of additional out-of-distribution data. When OO-Do is presented, the precision of the baseline object detection model decreased from 81.29 to 64.38 for object detection on Chula_{test}+Wild_{OO-Do} test set and from 74.64 to 45.31 for object detection on the in-the-wild parasite egg test set. We can clearly see that recalls of the models were not affected by out-of-domain because recall only regards true positive that the model can identify.

Adding classification model after the object detection to reclassify all detected boxes, Classification-later (without threshold), can improve overall performance of the baseline, 67.28% over 64.38% on precision and 99.00% over 96.69% on recall for the Chula_{test}+Wild_{OO-Do} test set and 53.37% over 45.31% on precision and 72.82% over 62.99% on recall for the in-the-wild parasite egg test set.

Table 2: Comparison of different approaches on Chula$_{test}$+Wild$_{OO-Do}$ dataset for out-of-domain experiments. All values are percentages. Bold numbers are superior results. (numbers in the table are represented as **top**, 2nd-top, 3rd-top, and regular)

Method	Parasite egg without OO-Do	Parasite egg with OO-Do	OO-Do
(Chula$_{test}$+Wild$_{OO-Do}$ dataset)	Precision / Recall / F1-score		
Object detection	81.29 / 96.69 / 88.33	64.38 / 96.69 / 77.30	**100.00** / 17.61 / 29.94
Classification-later (without threshold)	84.61 / 99.00 / 91.24	67.28 / **99.00** / 80.11	**100.00** / 17.61 / 29.94
Classification-later (SoftMax threshold)		79.35 / 90.77 / **84.67**	73.00 / 43.88 / 54.81
Classification-later (ODIN threshold)		80.14 / 74.38 / 77.15	46.16 / 54.61 / 50.03
Classification-first (SoftMax threshold)		**90.57** / 57.85 / 70.60	42.92 / 89.27 / **57.97**
Classification-first (ODIN threshold)		81.82 / 10.85 / 19.15	27.08 / **100.00** / 42.61

5.3. Chula$_{test}$+Wild$_{OO-Do}$ results

Based on the results of six different approaches on the Chula$_{test}$+Wild$_{OO-Do}$ dataset in Table 2, it appears that the classification-first approach using SoftMax threshold achieved the highest precision score of 90.57. This means that this method had the highest proportion of correctly classified positive results out of all the methods tested. The second highest precision score of 81.82 was obtained by the classification-first approach using the ODIN threshold and the third highest precision score of 80.14 was achieved by the classification-later approach using the SoftMax threshold.

Regarding recall score, the classification-later approach without threshold strategy had the highest score of 99.00. This means this method had the highest proportion of correctly identified positive results out of all the tested methods. The Object detection approach obtained the second-highest recall score of 96.69, and the third-highest recall score of 90.77 was achieved by the classification-later approach using the SoftMax threshold.

The F1-score, which measures the balance between precision and recall,

was highest for the classification-later using the SoftMax threshold approach at 84.67. This suggests that this method achieved the best overall balance between precision and recall and may be the most effective approach for this particular dataset. The second-highest F1-score of 80.11 was obtained by the classification-later approach without threshold strategy, and the third-highest F1-score of 77.30 was achieved by the Object detection approach.

Overall, the results indicate that the object detection, classification-later using SoftMax threshold and classification-later approach without threshold strategy are effective methods for dealing with out-of-domain images in the Chula$_{test}$+Wild$_{OO-Do}$ dataset. This conclusion is based on the evaluation of precision, recall, and F1-score metrics. The object detection approach, classification-later approach without threshold strategy and classification-later using the SoftMax threshold approach also had high scores in recall and F1-score and ranked in the top three.

To evaluate which model is more effective, we use OO-Do's precision, recall, and F1-score metrics, where out-of-domain images are considered positive and in-domain images are considered negative.

The second-highest precision score of 73.00% for classification-later approach using SoftMax threshold indicates that a large proportion of images classified as out-of-domain are correct. The higher recall score of 43.88% indicates that a larger proportion of actual out-of-domain images are correctly identified by this method. Finally, the higher F1-score of 54.81% suggests that this method is more effective overall in detecting out-of-domain images.

It can be concluded that classification-later using SoftMax threshold is more effective than classification-later without threshold strategy and object detection approach in detecting out-of-domain images on the Chula$_{test}$+Wild$_{OO-Do}$ dataset.

Therefore, the experimental results indicate that classification-later using SoftMax threshold is the most effective method for dealing with out-of-domain images in the Chula$_{test}$+Wild$_{OO-Do}$ dataset.

Table 3: Comparison of different approaches on In-the-wild parasite egg dataset for out-of-domain experiments. All values are percentages. Bold numbers are superior results. (numbers in the table are represented as **top**, 2nd-top, 3rd-top, and regular)

Method	Parasite egg without OO-Do	Parasite egg with OO-Do	OO-Do
(In-the-wild parasite egg dataset)	Precision / Recall / F1-score		
Object detection	74.64 / 62.99 / 68.32	45.31 / 62.99 / 52.71	**100.00** / 17.61 / 29.94
Classification-later (without threshold)	82.07 / 72.82 / 77.17	53.37 / **72.82** / **61.60**	**100.00** / 17.61 / 29.94
Classification-later (SoftMax threshold)		61.21 / 52.97 / 56.80	57.48 / 43.88 / 49.77
Classification-later (ODIN threshold)		**66.06** / 41.28 / 50.81	53.44 / 53.37 / 53.41
Classification-first (SoftMax threshold)		28.94 / 2.81 / 5.12	45.20 / 89.27 / **60.01**
Classification-first (ODIN threshold)		20.00 / 0.15 / 0.29	47.09 / **100.00** / 54.02

5.4. In-the-wild results

Based on the experimental results we provided in Table 2, it can be concluded that classification-later approach using the ODIN threshold achieved the highest precision score of 66.06% on the in-the-wild parasite egg dataset. The second-highest precision score of 61.21% was obtained by the classification-later approach using the SoftMax threshold, and the third-highest precision score of 53.37% was achieved by the classification-later without threshold strategy.

For the recall scores, the classification-later approach without threshold strategy had the highest score of 72.82. The Object detection approach obtained the second-highest recall score of 62.99, and the third-highest recall score of 52.97 was achieved by the classification-later approach using the SoftMax threshold.

The F1-score was highest for the classification-later approach without threshold strategy at 61.60. The second highest F1-score of 56.80 was obtained by the classification-later approach using the SoftMax threshold, and the Object detection achieved the third highest F1-score of 52.71.

The experimental results on the in-the-wild parasite egg dataset indicate that the classification-later using SoftMax threshold and classification-later approach without threshold strategy are also effective methods for dealing with out-of-domain images in the in-the-wild parasite egg dataset. These two methods had high scores in all three metrics and ranked in the top three.

To evaluate which model is more effective, we use OO-Do's precision, recall, and F1-score metrics, where out-of-domain images are considered positive and in-domain images are considered negative.

The second-highest precision score of 57.48% for classification-later approach using SoftMax threshold indicates that a large proportion of images classified as out-of-domain are correct. The higher recall score of 43.88% indicates that a larger proportion of actual out-of-domain images are correctly identified by this method. Finally, the higher F1-score of 49.77% suggests that this method is more effective overall in detecting out-of-domain images.

It can be concluded that classification-later using SoftMax threshold is more effective than classification-later without threshold strategy in detecting out-of-domain images on the the Chula$_{test}$ set.

Therefore, the experimental results indicate that classification-later using SoftMax threshold is the most effective method for dealing with out-of-domain images in the in-the-wild parasite egg dataset.

5.5. On the OO-Do images

The experimental results in the previous section have shown that classification-later using SoftMax threshold is the best approach for dealing with OO-Do images. This section presents the experimental results of detecting OO-Do images using classification-later with SoftMax threshold. The results are presented as a confusion matrix in Table 4, which shows the number of images seen as OO-Do for each class. The results suggest that the classification-later approach with SoftMax threshold effectively detects OO-Do images in some classes. Specifically, the detection rates for OO-Do images were highest for the adult parasite class (93.55%), followed by the arbitrary class (68.35%) and artifact class

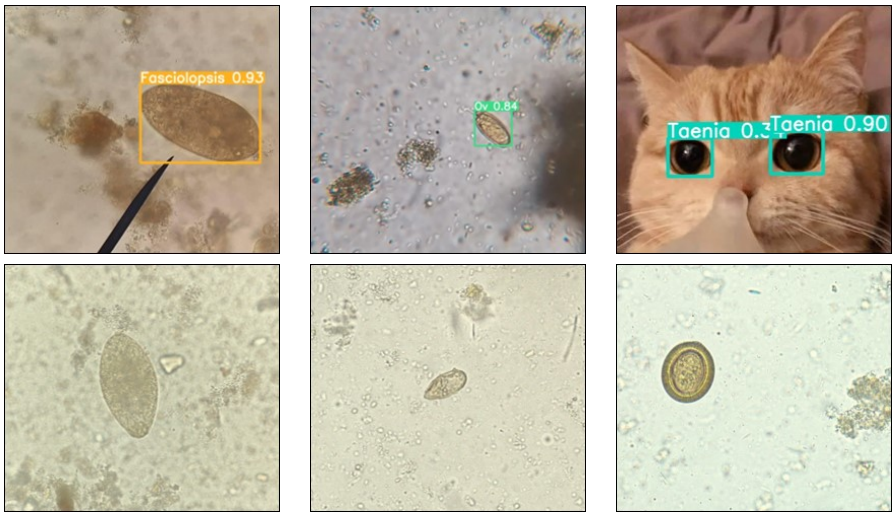
Table 4: Confusion matrix of OO-Do images that were detected by classification-later using SoftMax threshold

Misclassified classes	Adult parasite	Arbitrary	Artifact	Unclear image	Other parasite egg	Total
Ascaris lumbricoides	0	0	3	11	8	22
Capillaria philippinensis	0	1	0	5	0	6
Enterobius vermicularis	0	2	0	2	0	4
Fasciolopsis buski	1	5	6	19	73	104
Hookworm egg	0	13	2	38	5	58
Opisthorchis viverrine	1	3	9	33	53	99
Paragonimus spp.	0	1	4	32	6	43
Taenia spp.	0	13	2	28	0	43
Trichuris trichiura	0	1	4	19	1	25
Hymenolepis diminuta	0	1	1	2	0	4
Hymenolepis nana	0	4	5	21	0	30
Sum(%)	2(6.45)	44(31.65)	36(51.43)	210(71.43)	146(75.65)	438(60.25)
OO-Do(%)	29(93.55)	95(68.35)	34(48.57)	84(28.57)	47(24.35)	289(39.75)
Total (images)	31	139	70	294	193	727

(48.57%). An example of the classification model recovered the mistake of the detection model which mistakenly detected cat eyes as *Taenia* spp. eggs is illustrated in Fig. 8c. However, the detection rates for OO-Do images were much lower for the unclear image class (28.57%) and other parasite egg classe (24.35%). For other parasite egg class, there are two commonly misclassified classes due to the OO-Do classes have similar appearances to the trained classes. Fig. 8a and 8b demonstrate that *Echinostoma* spp. and *Fasciolopsis buski* share some similarities in their physical appearance, they can be distinguished based on the surface and their size while *Opisthorchis viverrini* (OV) and Minute Intestinal Flukes can be distinguished based on the position of the operculum.

6. Discussion and Conclusion

In this work, we aimed to address the challenge of detecting out-of-domain (OO-Do) images in the context of parasite egg object detection. Specifically, we proposed two-step approaches for detecting OO-Do images, including classification before(classification-first) or after object detection(classification-later). The experimental results suggest that classification-later using the SoftMax



(a) (b) (c)

Figure 8: Examples of misclassified and correctly classified classification cases. (a-b) are examples of the misclassified cases. An images of *Echinostoma* spp. (a above) and Minute Intestinal Fluke (b above) in the "other parasite egg" class was incorrectly classified as *Fasciolopsis buski* (a below) and OV (b below), respectively, which could lead to lower detection rates for OO-Do images in this class. (c) is an example of the classification-later framework correctly classified OO-Do images. An image of cat eyes (c above) was detected as Taenia spp.(c below) by the object detection. However, this image was correctly rejected by classification with the SoftMax threshold.

threshold approach is the most efficient approach for addressing OO-Do problems in this context.

Moreover, we developed a data-driven framework, which combines three datasets, namely Chula$_{val}$, Imagenette$_{OO-Do}$, and malaria$_{OO-Do}$, to find the optimal threshold for fine-tuning the two-step approach constructing a parasite egg recognition model for in in-the-wild applications.

Lastly, we presented 2 datasets for evaluation purposes. The Chula$_{test}$ +Wild$_{OO-Do}$ dataset, which combines parasite egg images from the Chula$_{test}$ with the OO-Do images from our in-the-wild parasite egg dataset, could be particularly useful for evaluating the robustness of OO-Do detection models to variations in the input data. In addition, we proposed a new dataset called the in-the-wild parasite egg dataset, which included parasite egg images and OO-Do images collected from running a parasite egg learning chatbot test session. This dataset could help evaluate OO-Do detection models under more realistic conditions.

7. Acknowledgment

This research project was financially supported by the Fundamental Fund of Khon Kaen University, fiscal year 2022, the National Science, Research and Innovation Fund (NSRF), Thailand.

References

[1] B. Zhao, S. Yu, W. Ma, M. Yu, S. Mei, A. Wang, J. He, A. Yuille, A. Kortylewski, Ood-cv: A benchmark for robustness to out-of-distribution shifts of individual nuisances in natural images, Proceedings of the European Conference on Computer Vision (ECCV).

[2] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, arXiv preprint arXiv:1610.02136.

[3] N. Anantrasirichai, T. H. Chalidabhongse, D. Palasuwan, K. Naruenatthanaset, T. Kobchaisawat, N. Nunthanasup, K. Boonpeng, X. Ma, A. Achim, Icip 2022 challenge on parasitic egg detection and classification in microscopic images: Dataset, methods and results, in: 2022 IEEE International Conference on Image Processing (ICIP), IEEE, 2022, pp. 4306–4310.

[4] C.-H. Lai, S.-S. Yu, H.-Y. Tseng, M.-H. Tsai, A protozoan parasite extraction scheme for digital microscopic images, Computerized Medical Imaging and Graphics 34 (2) (2010) 122–130.

[5] D. R. Loh, W. X. Yong, J. Yapeter, K. Subburaj, R. Chandramohanadas, A deep learning approach to the screening of malaria infection: Automated and rapid cell counting, object detection and instance segmentation using mask r-cnn, Computerized Medical Imaging and Graphics 88 (2021) 101845.

[6] D. Osaku, C. F. Cuba, C. T. Suzuki, J. F. Gomes, A. X. Falcão, Automated diagnosis of intestinal parasites: a new hybrid approach and its benefits, Computers in Biology and Medicine 123 (2020) 103917.

[7] A. Pedraza, J. Ruiz-Santaquiteria, O. Deniz, G. Bueno, Parasitic egg detection and classification with transformer-based architectures, in: 2022 IEEE International Conference on Image Processing (ICIP), IEEE, 2022, pp. 4301–4305.

[8] N. AlDahoul, H. A. Karim, S. L. Kee, M. J. T. Tan, Localization and classification of parasitic eggs in microscpic images using an efficientdet detector, in: 2022 IEEE International Conference on Image Processing (ICIP), IEEE, 2022, pp. 4253–4257.

[9] J. Ruiz-Santaquiteria, A. Pedraza, N. Vallez, A. Velasco, Parasitic egg detection with a deep learning ensemble, in: 2022 IEEE International Conference on Image Processing (ICIP), IEEE, 2022, pp. 4283–4286.

[10] N. Butploy, W. Kanarkard, P. Maleewong Intapan, et al., Deep learning approach for ascaris lumbricoides parasite egg classification, Journal of parasitology research 2021.

[11] A. Bendale, T. E. Boult, Towards open set deep networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1563–1572.

[12] P. Perera, V. M. Patel, Deep transfer learning for multiple class novelty detection, in: Proceedings of the ieee/cvf conference on computer vision and pattern recognition, 2019, pp. 11544–11552.

[13] P. R. Mendes Júnior, R. M. De Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, O. A. Penatti, R. d. S. Torres, A. Rocha, Nearest neighbors distance ratio open-set classifier, Machine Learning 106 (3) (2017) 359–386.

[14] M. Goldstein, S. Uchida, A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data, PloS one 11 (4) (2016) e0152173.

[15] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, Y. Tian, Learning open set network with discriminative reciprocal points, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, Springer, 2020, pp. 507–522.

[16] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, PMLR, 2016, pp. 1050–1059.

[17] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, Advances in neural information processing systems 30.

[18] T. Salimans, A. Karpathy, X. Chen, D. P. Kingma, Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications, arXiv preprint arXiv:1701.05517.

[19] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, C. Cadena, Out-of-distribution detection for automotive perception, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, 2021, pp. 2938–2943.

[20] Y. Li, J. Košecká, Uncertainty aware proposal segmentation for unknown object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 241–250.

[21] K. Deepshikha, S. H. Yelleni, P. Srijith, C. K. Mohan, Monte carlo dropblock for modelling uncertainty in object detection, arXiv preprint arXiv:2108.03614.

[22] A. Zolfi, G. Amit, A. Baras, S. Koda, I. Morikawa, Y. Elovici, A. Shabtai, Yolood: Utilizing object detection concepts for out-of-distribution detection, arXiv preprint arXiv:2212.02081.

[23] Y. Blei, N. Jourdan, N. Gählert, Identifying out-of-distribution samples in real-time for safety-critical 2d object detection with margin entropy loss, arXiv preprint arXiv:2209.00364.

[24] D. P. K. N. T. K. T. H. C. N. N. K. B. N. Anantrasirichai, Parasitic egg detection and classification in microscopic images (2022). doi:10.21227/vyh8-4h71.
URL https://dx.doi.org/10.21227/vyh8-4h71

[25] J. Howard, Imagenette.
URL https://github.com/fastai/imagenette

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[27] V. Ljosa, K. L. Sokolnicki, A. E. Carpenter, Annotated high-throughput microscopy image sets for validation., Nature methods 9 (7) (2012) 637–637.

[28] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.

[29] G. Jocher, A. Stoken, A. Chaurasia, J. Borovec, NanoCode012, TaoXie, Y. Kwon, K. Michael, L. Changyu, J. Fang, A. V, Laughing, tkianai, yxNONG, P. Skalski, A. Hogan, J. Nadar, imyhxy, L. Mammana, AlexWang1900, C. Fati, D. Montes, J. Hajek, L. Diaconu, M. T. Minh, Marc, albinxavi, fatih, oleg, wanghaoyang0106, ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support (Oct. 2021). doi:10.5281/zenodo.5563715.
URL https://doi.org/10.5281/zenodo.5563715

[30] S. Liang, Y. Li, R. Srikant, Enhancing the reliability of out-of-distribution image detection in neural networks, in: 6th International Conference on Learning Representations, ICLR 2018, 2018.