DETRiminator: Using AI to Prevent the Spreading of Mosquito-borne Diseases

Katherine Zeng, katherinezhwc@gmail.com

Abstract

Being portrayed as a tiny flying nuisance by most, mosquitoes are in reality the deadliest animal to humans in the world, killing around 1 million people every year. According to the Centers for Disease Control and Prevention(CDC), one of the most effective ways to protect against this danger is to look towards the root cause: water-holding containers left outside that mosquitos require in order to lay eggs and reproduce.

In this project we trained an AI model to detect open water containers in the backyard of homes, which is one of the commonly overlooked breeding grounds for mosquitos. The training and evaluation dataset were created with images taken from a DJI Mini 4K drone. The images were then processed with the CVAT image annotation platform and exported to a format compatible with the COCO dataset. Our end-to-end object detection AI model is based on the state-of-the-art DETR model from Meta Research. We leveraged the pre-trained model from the HuggingFace Transformer library, then re-trained it with our own dataset using Pytorch Lighting on a Nvidia

GeForce RTX 4060 GPU. The re-trained model was uploaded to HuggingFace model hub and can be downloaded later for inference.

The complete source code is available at https://github.com/katherinezhwc/Detriminator.

Mosquito: The deadliest animal

Surpassing both the Killer Whale and Black Mamba snake, the mosquito is the world's deadliest animal to humans. In human history, more people have died of mosquito-caused diseases than of all other animals combined. For comparison, in 2023 only 14 people were killed in shark attacks worldwide while mosquitoes average on killing around 1 million people every year. In addition, mosquito-borne diseases infect up to 700 million each year - almost one in ten people.

Mosquitoes do not kill humans directly. Rather, they are a vector that transmit deadly diseases such as malaria, dengue, West Nile, yellow fever and Zika etc.

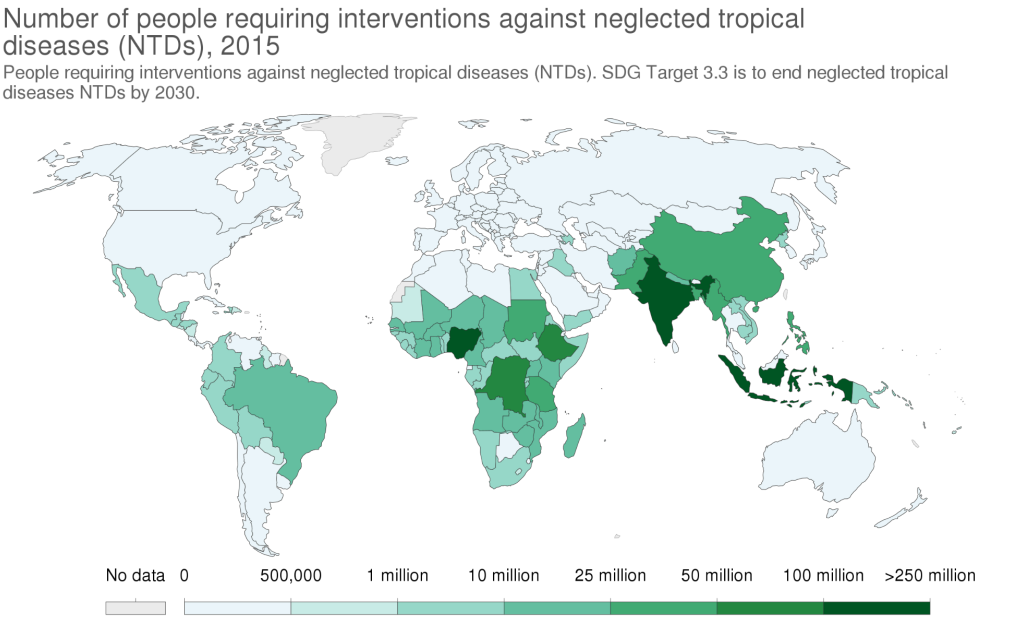
West Nile Virus in California

In California, West Nile virus is the most common and serious vector-borne disease. There have been more than 7,000 human cases and over 300 deaths reported in California since 2003. Preventive measures recommended by California department of health includes dumping and draining standing water in yards, changing the water at least once a week in places like pet dishes, birdbaths, and wading pools, and keep gutters free from debris. Restricting the mosquito population is the most effective

preventative measure, and involves a constant mindfulness of these potential still-water containers.

Disproportionate Impact on Low-income Communities

A significant portion of the diseases mosquitos transfer are classified as NTDs, or [Neglected Tropical Diseases]. NTDs are a group of diseases that usually affect lower-income populations and 3rd-world countries, but have been historically overlooked in terms of treatment development. Lower quality of living conditions lead to greater risk of being infected by a mosquito, leading poorer communities to receive the biggest hit, making them disproportionately affected. Furthermore, pharmaceutical systems are profit-oriented, and typically don't prioritize patients of lower income. Because of the lack of medical attention, the results of these diseases are made more severe, and include disability and death. Exposure to these diseases continues the cycle of poverty and suffering, preventing victims from being able to overcome either vulnerability.



Climate Change

As climate change grows more severe, the average temperature on Earth climbs higher and higher, incidentally also escalating the danger of mosquito-transferred diseases. Firstly, climate change incites a longer active season for mosquitoes, intensifying the overall exposure. Longer active seasons are a result of more favorable breeding conditions, including a warmer average temperature and increased rainfall. These conditions also result in a larger total mosquito population. Secondly, a study by the National Library of Medicine compared the incubation period of certain pathogens in mosquitos, and found that as temperature increased, parasites would develop quicker and there would be greater risk for diseases such as malaria. Thirdly, warmer temperatures allow mosquitoes to inhabit originally cooler regions that they previously were not able to survive in. This exposes regions previously not affected.

In a recent study, the London School of Hygiene and Tropical Medicine predicted that the suitability of diseases towards the environment would increase by up to 6 additional months to diseases like malaria and dengue due to climate change. This would put more of the human population at risk to those diseases, affecting up to 4-7 billion people by 2070.

Mosquito Control

Everyone can participate in helping lessen the impact of mosquitos. This is a list of CDC recommendations on deal with water-holding items:

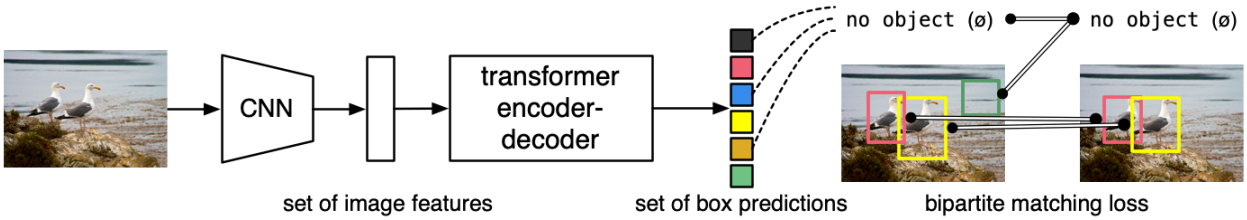
1. Throw out, clean, or cover the containers weekly.
2. Use wire mesh with holes mosquitoes are too large to get through to cover containers.
3. Fill tree holes.
4. Repair cracks or gaps in septic tanks.
5. These water-holding items include: tires, buckets, planters, toys, pools, birdbaths, flower pot saucers, trash containers, open vents, and plumbing pipes.

DETR Model for Object Detection

The DETR model was proposed in the 2020 paper "*End-to-End Object Detection with Transformers*". It has received extensive attention from the academia due to its streamlined architecture and elimination of hand-crafted components. For example, Deformable-DETR aims to accelerate training convergence with multi-scale features by enhancing the efficiency of the attention mechanism. Recently RT-DETR shows the model can outperform previously advanced YOLOs in both speed and accuracy

DETR Architecture

The DETR model starts with a CNN(convolutional neural network) backbone, then has a Transformer encoder-decoder which uses the features generated by the backbone to more efficiently complete object detection. It has proven to be simpler than a pure CNN model.



How DETR works

First, the DETR model uses ResNet-50 as the CNN backbone which outputs a feature map. Then, the dimensions of the feature map are changed to match the hidden dimension of the Transformer. This feature map is then flattened into tokens, which the Transformer encoder uses to further develop understandings of the input image.

After that, queries are sent to identify the directed objects in the image, which in our case is the water-holding containers. Self- and encoder-decoder attention layers will relay information to the embeddings. There will then be two heads executed to lead to the final output, the first being a series of layers that predict the bounding

box for each water-holding container. Then, the second head will identify each object as either a water-holding container or not a water-holding container.

The DETR model pretrained on the COCO dataset always predicts 100 objects, with each object having a prediction of the bounding boxes and label. L1 and IoU loss functions are used to optimize the final predictions.

Transfer Learning

Transfer learning is the reuse of a pre-trained model on a new problem. It's popular in deep learning because it can train deep neural networks with comparatively little data. This is especially useful in our case where it is time-prohibitive to collect and annotate a big and diverse dataset to train our relatively large object detection model.

Our AI model is based on the facebook/detr-resnet-50 model from Hugging Face that is pre-trained on COCO 2017 object detection task dataset.

We fine-tuning the pre-trained model for fast learning with our own much smaller dataset

Open water container is not in the original COCO objects so the pertain model won't be able to detect the container.

Hugging Face

Hugging Face is a machine learning (ML) and data science platform and community that helps users build, deploy and train machine learning models. The platform offers model hosting, tokenizers, machine learning applications, datasets, and educational materials for training and implementing AI models.

Transformer library

Hugging Face is known for its Transformers Python library, which provides a library of models that are based on the transformer deep-learning architecture. The library gives developers an efficient way to include one of the ML models hosted on Hugging Face in their workflow and create ML pipelines.

Hugging Face Transformer library offers a range of pre-trained models for different tasks, including natural language processing, computer vision, speech recognition and Multimodal modeles. Using pretrained models will reduce your compute costs, carbon footprint, and save your time and resources required to train a model from scratch. Our object detection model is based on DetrForObjectDetection from the transformer library.

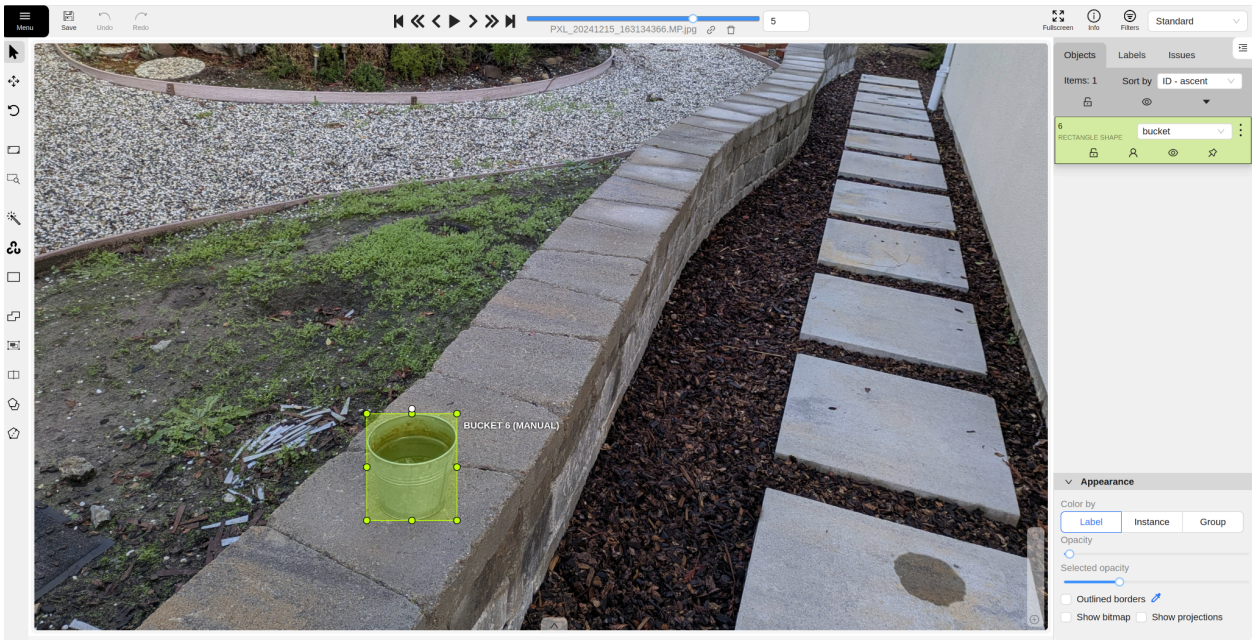
Dataset

Image collection

We collected 100+ images with a DJI Mini 4K drone. The images include open water containers with different colors and sizes. We take pictures with drones from different altitudes, different angles and under different light conditions. Below is an example of an open water container with water after overnight rain.

Image Annotation

In order to train our model, we need to annotate the images with bounding-boxes and labels. CVAT is a free, open source, web-based image and video annotation tool used for labeling data for computer vision tasks. The image below shows how we annotate the image with a bounding-box around the container and labeled it with "*bucket*"



Dataset Export

Microsoft's Common Objects in Context dataset (COCO) is the most popular object detection dataset at the moment. It is widely used to benchmark the performance of computer vision methods. Due to the popularity of the dataset, the format that COCO uses to store annotations is often the go-to format when creating a new custom object detection dataset.

The "COCO format" is a specific JSON structure dictating how labels and metadata are saved for an image dataset. After image annotation with CVAT, we export the dataset using the COCO 1.0 format.

Training and Validation dataset

During the export, we split the dataset to training dataset and validation dataset with 80:20 ratio. The training dataset is used to fit the model parameters. The validation dataset provides an unbiased evaluation of a model fit on the training data set while tuning the model's hyperparameters. We don't have a separate test dataset due the limited amount of data we have.

Model Training

Dataset Module

We created a custom pytorch module based on torchvision.datasets.CocoDetection to load our training and validation datasets. The module combines the raw image files and the annotation metadata to instantiate the dataset objects. Below is an example of an image with bounding-box overlay from our training dataset.

We then created train_dataloader and val_dataloader using torch.utils.data based on the two datasets.

DetrForObjectDetection Model

We then instantiate a pretrained model from the transformer library with resnet-50 backbone and our customised number of labels.

```
self.model =
DetrForObjectDetection.from_pretrained("facebook/detr-resnet-50",

revision="no_timm",

num_labels=len(id2label),

ignore_mismatched_sizes=True)
```

Pytorch Lighting

Our ObjectDetection model is then wrapped with the Pytorch Lighting module. Pytorch Lighting is a popular deep learning framework designed to create scalable deep learning models that can be easily trained on distributed hardware. We use the loss function defined by the base *DetrForObjectDetection* model and torch.optim.AdamW optimiser to train the model.

Training

We use pytorch lightning.Trainer with 200 max_steps to fit the model with our datasets. After training complete the model is uploaded to the hugging face model hub.

Model Inference

To inference we download the trained model from the hugging face hub and run the forward pass. Below is an example of the detected object overlaid with bounding-box and confidence score.

Future Works

Real time Detection: Deploy the AI model directly to the drone so the drone can do real time detection.
Detection of more types of breeding grounds: For example clogged gutters or stagnant water in old tires
Trace-back: Add the capability to save precise GPS location so the drone can guide the human back to the scene where the object is detected for proper treatment of the mosquito breeding spot.