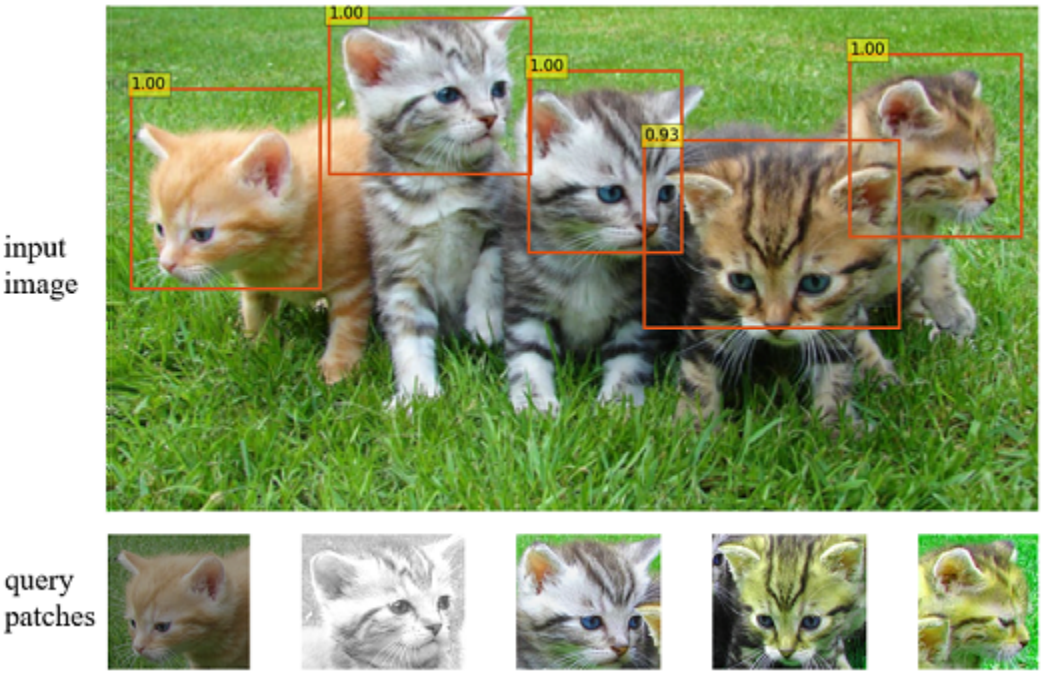
Final Report: UP-DETR

Abigail Lambert and Kendall Ruth

Abstract

UP-DETR is a successful object detection model that we attempted to use to identify objects in satellite imagery instead of the everyday objects it normally identifies. We approached this problem by researching the UP-DETR model to understand its workings. We then looked for ways to switch the data set that it is trained on. This included generating an annotations file and editing other parts of the code so that it could run on the DOTA data set which is composed of satellite imagery. We had minor success training on a subset of the dataset locally but ran into complications as we scaled up on the Google virtual machine and set up our environment. We then tried to train the DETR model which removes the unsupervised learning aspect of the model and continued to have issues with the environment. In the end, we decided to not move forward with the UP-DETR or the DETR model and instead switched to helping with the SimCLR model of the other group.

Methods



The most in-depth resource we found for UP-DETR was the paper. It outlined the workings of the model and discussed how to analyze the results. We mostly used the paper to inform our understanding of the model and to interpret the results we got from training on a subset of the data. There were a few other online resources we referenced as we looked at switching out the data set, but they were not geared toward satellite imagery and the formatting we had.

To switch out the data sets we needed to modify the DOTA data set to be similar to the COCO data set which UP-DETR was originally trained on. To do this we had to change the key file of the DOTA data set to the same format as the annotations file the COCO data set uses. This annotations file needed to be a json that has each image id, the bounding boxes of that image enclosing the objects, and the labels of those bounding boxes. Previously the key file had one line for each bounding box. Generating this annotations file for the DOTA data set was a key step in moving forward.

In addition to generating the annotations file we also made slight edits to the code available in the UP-DETR GitHub repository. These edits mostly were adjusting the file path the model used to find the training and testing data. Before training on all 100,000+ images, we wanted to test our changes in a smaller environment. We trained locally on 1,000 images and evaluated with 2,000+ images for several epochs and saw improvement for the few epochs we were able to train it as shown in our results section.

The next step was to scale up and train our model on all of the DOTA images. To do this we used a Google virtual machine. We cloned our repository onto the vm and brought in all of the images. We then tried to train the model but repeatedly ran into issues with the environment. There were issues connecting to the Nvidia driver and then many issues relating to packages and permissions to alter the environment.

As we were nearing the end of the semester we looked for other possible ways to get results. We tried using the DETR model which does not use an unsupervised learning pre-training as UP-DETR does. However, we encountered similar issues to our attempt with UP-DETR. The paper Facebook wrote about DETR does emphasize its unique ability to identify objects it has not been trained on. Knowing this, we decided to see how well DETR would be able to identify objects in the DOTA dataset that it had never seen before. DETR was able to identify some of the objects and their locations with varying success but was unable to classify them as it was the first occurrence.

Results

In our initial testing in a google colab notebook, we were able to come up with some basic results before attempting to move over to the google cloud instance. These basic results were only on a subsection of the data, and with less epochs than the original dataset. Here is a screenshot of the results. The average precision stands for how accurate the model is in defining each of the boxes in the images:

```
Accumulating evaluation results...
DONE (t=0.47s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.007
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.026
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.002
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.003
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.012
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.007
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.007
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.025
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.045
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.034
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.091
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.086
```

To give a brief explanation of these results and what they mean, the main ones that we look at to tell accuracy are middle ones with accuracy small,medium and large, along with the overall accuracy, which is the first one. The rest of the accuracies are not as important as the rest of them. These sizes correspond to the sizes of the patches taken from the images. Here is a table with comparison between the model run on the COCO dataset (Which was originally used with the UP-DETR model) and the DOTA dataset.

Name	**Fine-tuned**	**Epochs**	**Box AP**	**AP_S**	**AP_M**	**AP_L**
UP-DETR	COCO	300	43.1	21.6	46.8	60.9
UP-DETR	DOTA	108	0.7	0.3	1.2	0.7

As you can see on this, while we were able to get results with our subsection of the data, they are very little compared with the results that the model achieved using the COCO dataset. This is why we were interested in running our data on the google cloud. If we would be successful in running our code using the cloud, we would have been able to run our code for more epochs and use all the images in the dataset. Unfortunately as discussed in this writeup, we encountered many issues with that that prevented us from using the google cloud to get the results that we wanted, and in the end we ended up switching to help the other half of our group with the SimCLR model.

Recommendations

One of the first things that we would recommend people who are working with the UP-DETR model to do would be to actually first work with the DETR model. For a little background, the DETR model was first created by Facebook as a way to do object detection on images using transformers. The DETR model is more commonly used, and there are more resources available using the DETR model. Using the DETR model first will allow for better understanding of the basis of what the model is trying to do. This will make it easier when switching to the UP-DETR model.

Another thing that we would change in the future would be when we are working with our google cloud. When we got to the point that we tried to run on a model in the google cloud, we had to play around to set up the environment that we were running it on. In the process of editing the environment, we had to try a lot of different ways to get the code to run correctly. This included installing multiple different packages in many different formats. In the process of doing this, it made the cloud process more complicated. In the future when working on this project again, we would recommend being more careful with the way the model was set up.

One last final thing that we would try would be going along with trying the DETR model first would be to simply use the attention mask process. This process is run on images to decipher where important objects are located in them. The code uses pretrained models that have been trained to identify objects in images. Based on the shapes and aspects of each of the images, this model creates an accuracy score. This accuracy score helps to tell the user with what accuracy the model estimates the shape in the image to be. While we were finishing up our project, we attempted to run some basic attention masking in identifying the satellite images that were in the different images in the DOTA dataset, and we were able to see some results in that. It would be very interesting to see any more results from trying this process, and it would especially be interesting to try this first in future, even before running any of the DETR algorithm.

Works Cited

Z. Dai, B. Cai, Y. Lin, and J. Chen, 'UP-DETR: Unsupervised Pre-training for Object Detection with Transformers', *arXiv [cs.CV]*. 2021.

Z. Dai, B. Cai, Y. Lin, and J. Chen, 'UP-DETR: Unsupervised Pre-Training for Object Detection With Transformers', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1601–1610.