Arable & Non-Arable land DetectionUsing Deep Learning

Salman Farshi
Dept of ECE
North South University
Dhaka, Bangladesh
salman.farshi@northsouth.edu

Md. Iftakarul Islam Suvo
Dept of ECE
North South University
Dhaka, Bangladesh
islam.suvo@northsouth.edu

Md.Ashiquer Rahman
Dept of ECE
North South University
Dhaka, Bangladesh
ashiquer.rahman@northsouth.edu

Shahnewaz Siddique
Dept of ECE
North South University
Dhaka, Bangladesh
Shahnewaz.siddique@northsouth.edu

Abstract

Bangladesh is a small country with a huge population. There are also many arable lands here which are not being cultivated. They are fallen as barren land. If we can cultivate these abandoned lands all across the country by the aid of the Government officially, we can reduce the food crisis of the Bangladeshi people. In order to do so, we intend to detect arable & non arable lands by deep learning process and also to differentiate arable lands from the non arable ones in order to produce a collected of the arable lands in Bangladesh.

Introduction

Food crisis & non arable lands responsible for it in a major issue in Bangladesh when it comes to food problem. As we are pondering upon how to solve this food crisis of Bangladesh, we are detecting arable lands. We are using various deep learning algorithms for it. First of all, we take satellite images plain lands of different districts of Bangladesh. Arable, non arable, cultivated, and farmed lands are there. We detach only arable and non arable lands from all other types of lands. We use a software called ‘LabelImg’ to label these two types of land into two distinct classes we named ‘Arable’, and ‘Non Arable’. Then we use them in Google Colab to run algorithm over them. That provides us with an output of Data accuracy.

Related Works

As far as we know, there is no researcher team or researcher individual has done the exact research of ours not only in Bangladesh but also all over the world. So, getting such working on lands is quite impossible. However, there are some works on various other things applying the same method as we do here.

CONVOLUTIONAL NEURAL NETWORK FOR DETECTION OF RESIDENTIAL PHOTOVOLTAIC SYSTEMS IN SATELLITE IMAGERY

It was a research done by Matthew Moraguez, Alejandro Trudillo and their team where they tried to detect photovoltaic systems through image processing in satellite imagery. They also used convolutional neural network for larger images. They had a precision of 91.9% & a recall of 92.4% in PV classification in Californian cities [1].

SHIP DETECTION AND FINE-GRAINED RECOGNITION IN LARGE-FORMAT REMOTE SENSING IMAGES BASED ON CONVOLUTIONAL NEURAL NETWORK

It was a research done by Jin grun Li and his team. There they tried to detect ships and fine grain recognition in large format remote sensing images. They use convolutional neural network. They got a precision of 0.9, recall of 0.95 for Res34. In 'ours', they got precision of 0.925 & recall of 0.99. They used 30 images in total with a resolution of 3000*3000 [2].

Google Earth Engine Cloud Computing Platform for Remote Sensing Big Data Applications: A Comprehensive Review

A review research done by Mehsan Amani, Seyed Ali Ahmadi and their team to review how good Google earth engine cloud computing platform performs in remote sensing big data applications [3].

Why the problem is interesting

Food crisis is a common problem in Bangladesh. Its also one of the most dangerous problem all across the country. Food crisis is giving birth new crimes almost everyday in middle class and lower class people of the country. The government is trying to reduce the food crisis at any how. If we can detect all the arable lands of Bangladesh and cultivate them using Government fund, we hope to reduce our food crisis problem to 50% or even less than that. We can implant this method in countries like India, Pakistan, Central Asian countries where lands are available with zero cultivation. This process can be used not only in Bangladesh but also in the entire world. As we are the first team to do so, we hope this research will bring a huge respect to our country. It will also encourage other Bangladeshi researchers to research something using deep learning. Entirely, it will open a new door of the sectors of machine learning and deep learning researches in our country.

Methodology

This section provides an overview of the methodology of the research. Data collecting process, labelling process, algorithm running process are discussed here. Necessary diagrams, tables are provided in this section.

Our work starts with collecting Satellite images of different districts of Bangladesh such as Dhaka, Comilla, Chandpur, Manikgonj, Pabna etc.

We have used Google Earth satellite view to collect images.

Fig-1 shows the Google Earth interface while capturing an image from
Pabna city

Fig -1: Google Earth interface

Fig-2: A fresh image without labeling showing arable land in countryside

Fig-3: 'LabelImg' interface, detecting arable lands in blue and non arable in light green.

After labelling these images, we got a .txt file for each images which contain some numbers that generated through labelling and our algorithm will run through it.

All object detectors take an image in for input and compress features down through a convolutional neural network backbone. In image classification, these backbones are the end of the network and prediction can be made off of them. In object detection, multiple bounding boxes need to be drawn around images along with classification, so the feature layers of the convolutional backbone need to be mixed up and held up in light of one another. The combination of backbone feature layers comes

What is YOLO?

YOLO is the abbreviation for You Only Look Once. YOLO is a state-ofthe-art, real-time object detection system framework. It was invented by Joseph Redmon. It is a real-time object recognition system that can recognize multiple objects in a single frame at a time. YOLO has evolved into newer versions over time. They are viz., YOLOv2, YOLOv3, and YOLOv4. YOLOv4 is the latest version that has released so far.

What is YOLOv4?

YOLOv4 is an object detection algorithm framework that is an improved version of the YOLOv3 model. The YOLOv4 method was invented by Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. It is twice times faster EfficientDet with comparable performance. In addition, AP (Average Precision) and FPS (Frames Per Second) in YOLOv4 have updated by 10% and 12% respectively comparing to YOLOv3. YOLOv4's architecture is composed of CSPDarknet53 as a backbone, spatial pyramid pooling additional module, PANet pathaggregation neck, and YOLOv3 head.

YOLOv4 uses many new features and puts together some of them to achieve state-of-the-art results: 43.5% AP (65.7% AP50) for the MS
COCO dataset at a real-time speed of ~65 FPS on Tesla V100. [4]

How does YOLO work?

YOLO is made on a single Convolutional Neural Network (CNN). The CNN divides an image into more than one region and then it predicts the boundary boxes and probabilities for every region. It simultaneously predicts multiple bounding boxes and probabilities for those classes. YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. in the neck.

It is also useful to divide object detectors into two categories: one-stage detectors and two stage detectors. Detection comes in the head. Twostage detectors decouple the task of object localization and classification for each bounding box. One-stage detectors make the predictions for object localization and classification at a time. YOLO is a one-stage detector, as the name of it derives 'You only look once.' [5]

Results & output

After collecting images through Google earth and labelled data; we divided the data into two parts. First part is 80%, training data; rest 20% are testing data.
We have run 6000 iterations through the training data. Without 6000 iterations a full result wasn't possible. We got the most accuracy at 2000 iterations. The accuracy at that point was 76%. After that point, the accuracy went downwards and stopped at 60%-65%. So, we stored the primary 2000 iterations as best weight. The precision was 78%, recall was 77% F1 score was 76%. Mean average precision or accuracy was 76%.

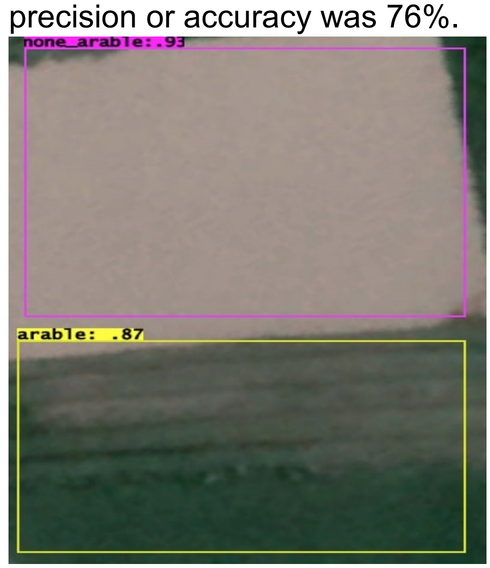

Fig 8: Output sample

In these iterations, we saw that, the framework can detect Arable lands more accurately almost 80%. Non Arable land detection was a little bit lower than that, 73%. The reason behind getting 80% is, the image resolution was quite good for Arable lands. The green color was more accurately detected by the algorithm. The non arable (not pure green) lands were less light reflected and were not so bright. We could have more accuracy if we had used a huge amount of images like 20,000 – 30,000. Because the algorithm gives more accurate result for more images.

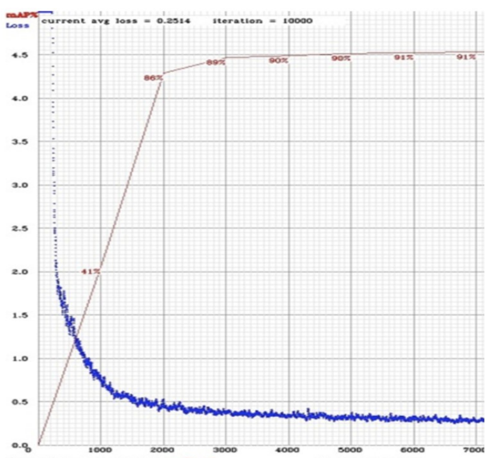

Fig-6: accuracy vs iteration

Precision	Recall	F1-Score
0.78	.77	.76

Fig-7 Result Table

```
mean average precision
(mAP@0.50) = 0.76, or 76%

arable:80%

none_arable:73%
```

Problems we faced

We had to struggle a lot for data collecting. The Google Earth images weren't pretty clear. That's why we had to face problems while labelling them. When we run the algorithm over the images, most of the images came out bogus and not acceptable for the result. So, we had to take data for the second time. The running took almost 88 hours to reach to the final result. Overall, the project wasn't an easy task. But we are happy that we completed that successfully.

Future work

We have completed our research successfully. Our algorithm could Arable & non Arable lands successfully upto 76%. In the near future, we would like to add a bigger picture to the research. We will collect 15,000-20,000 images in order to measure accuracy. Hopefully, our result will be more accurate then. As we mentioned before, if our research becomes successful in Bangladesh, we will do the same for the countries alike to Bangladesh. We hope to achieve our goal very soon.

Conclusion

Using the 'YOLO v4' algorithm to detect lands was a great achievement for us. We did label the images pretty much correctly with 'LabelImg'. Our data was unique also as it is based on districts on Bangladesh. Hopefully, this result will contribute to reduce the food crisis problem of Bangladesh perfectly and very soon we will be citizens of a country where there is no food crisis.

References

[1]. Matthew Moraguez, Alejandro Trujillo, Olivier de Weck, and Afreen Siddiqi "CONVOLUTIONAL NEURAL NETWORK FOR DETECTION OF RESIDENTIAL PHOTOVOLTAIC SYSTEMS IN SATELLITE IMAGERY"

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA

[2]. Jingrun Li, Jinwen Tian_, Peng Gao, Linfeng Li "SHIP DETECTION AND FINE-GRAINED RECOGNITION IN LARGE-FORMAT REMOTE SENSING
IMAGES BASED ON CONVOLUTIONAL NEURAL NETWORK"

School of Artificial Intelligence and Automation, Huazhong University of Science and Technology,

[3]. Meisam Amani *, Senior Member, IEEE*, Arsalan Ghorbanian , Seyed Ali Ahmadi , Mohammad Kakooei , Armin Moghimi , S. Mohammad Mirmazloumi*, Student Member, IEEE*, Sayyed Hamed Alizadeh Moghaddam , Sahel Mahdavi, Masoud Ghahremanloo, Saeid Parsian, Qiusheng Wu , and Brian Brisco "Google Earth Engine Cloud Computing Platform for Remote Sensing Big Data Applications: A Comprehensive Review"

[4] https://towardsdatascience.com/whats-new-in-yolov4-323364bb3ad3
[5]. https://blog.roboflow.com/a-thorough-breakdown-ofyolov4/?fbclid=IwAR3BsoDULrZseFguHACS1AUSdl2AYqbws2_b bmvo_77g UjqXKwyMluaoIq4
[6]. https://github.com/AlexeyAB