

2017-06-06 Meeting notes

Date

06 Jun 2017

Attendees

[Shannon Bradley](#)

[Luigi Marini](#)

[Rob Kooper](#)

[Ankit Rai](#)

Agenda

- GI Identification - tool information gathering

Discussion items

GI Identification

Luigi

Shannon

Ankit

Rob

Notes:

Overview – algorithm – project for Luigi on the overhead

This is the final alg using for GI detector for BD

It takes RGB image – 1024/1024 – breaks it down to scale by half – 512x512 image and you get 4

Keep original as well

Input image can be of any resolution

Then for each level – have 3 different features

Create a window of nxn size – from each window – slide window from left top to right top

Extract features

Classifier

X and Y location recorded

Such as tree or bicyclist

Creates a location for box of that classifier

Detects multiple high priority regions

Creates a pyramid of images – x levels deep

Tests the classifier on those windows

Demo – 512x512 – but imagine hi rez and more images and populate locations

Much larger / complexity

Want to test model on BD on single level of data – 100,000 to 2,000,000 images

Test real time online detection

Local machine got stuck on a couple thousand images

Can BD handle this load?

Include this algorithm – he is testing deep learning models as well – 120 hours to train on Blue Waters

GPU clusters – all hand labeled for learning / training data

So far 76 to 88 percent accurate

Improve algorithmically? Or increase computational power?

BD helps you scale large collections by deploying many instances of extractors – if you put 1,000,000 images – then BD would scale up to many instances to handle that many – easy to parallelize since the images are independent

Need to tell the extractor – we can run this many instances at once – then upload all the images to it

Currently scaling testing

In this case we do not want to run every image extractor – we only want to run 1 extractor on all the photos

Can start doing this now – we can monitor together

Inside each extractor – you can modify model – we can make sure there are enough resources on each node – each node only has so much memory – so we need to make sure the images are not too large

Eventually this would be used in Dallas – scale it up for the city of Dallas

Could make it so you give xml with location of photos – so specific to that use case – at this point it is intensive in specificity

Need to be able to upload images to BD – send an image/images to BD

BD reacts to data coming in

In future – connect to local data instead of having to upload?

Green path – call to google API to get one image? Or chunk locally

Send the chunks

Get scores back

Then combine locally

Work with Marcus to refine and Ankit needs updated on the new API

Call just that one extractor – and allow that extractor to scale up (right now we only allow 5)

Will need to make sure the RAM instantiated with the docker VM is enough

Action items

