Deep learning in remote sensing

Mohamed Yousif and Mohamed Jaafar

January 9, 2018

Overview

Overview

- Introduction
- Our work
- Difficulties we faced
- Next works

What is it like to be a GIS analyst

GIS is quite simple. Our dataset is always has something to do with the Earth. Hence, the G. Ofttimes, the end result of a GIS workflow is a map. However, that is not always the case.

It is very hard to separate GIS from remote sensing. Remote sensing can be thought of as the science behind interpreting information from maps, where GIS is the tools and techniques to make such *interpretations* possible.

GIS is not the same as ArcGIS.

Making interpretations from an image

Perhaps one of the main problems GIS tries to solve is generating maps from images. Images can be either from a satellite (satellite imagery), or from an airplane (aerial imagery). The former usually has multiple bands (more on that later), and also taken from very high altitudes, while the later uses same bands as our phone camera, though from a higher place.

Previously, people had to *survey* to make a map. Also, hence the name Surveying. You can imagine how boring is that task.

Make interpretations from an image

Since we are talking about images that captures places, we need to assign coordinates to these images. Aerial imagery usually came without embedded location information (georeferenced). So, we need to make that step. It is not hard though, we just need to transfer this image (1^{st} order polynomial transformations is the most common one) [**?**].

Making interpretations from an image

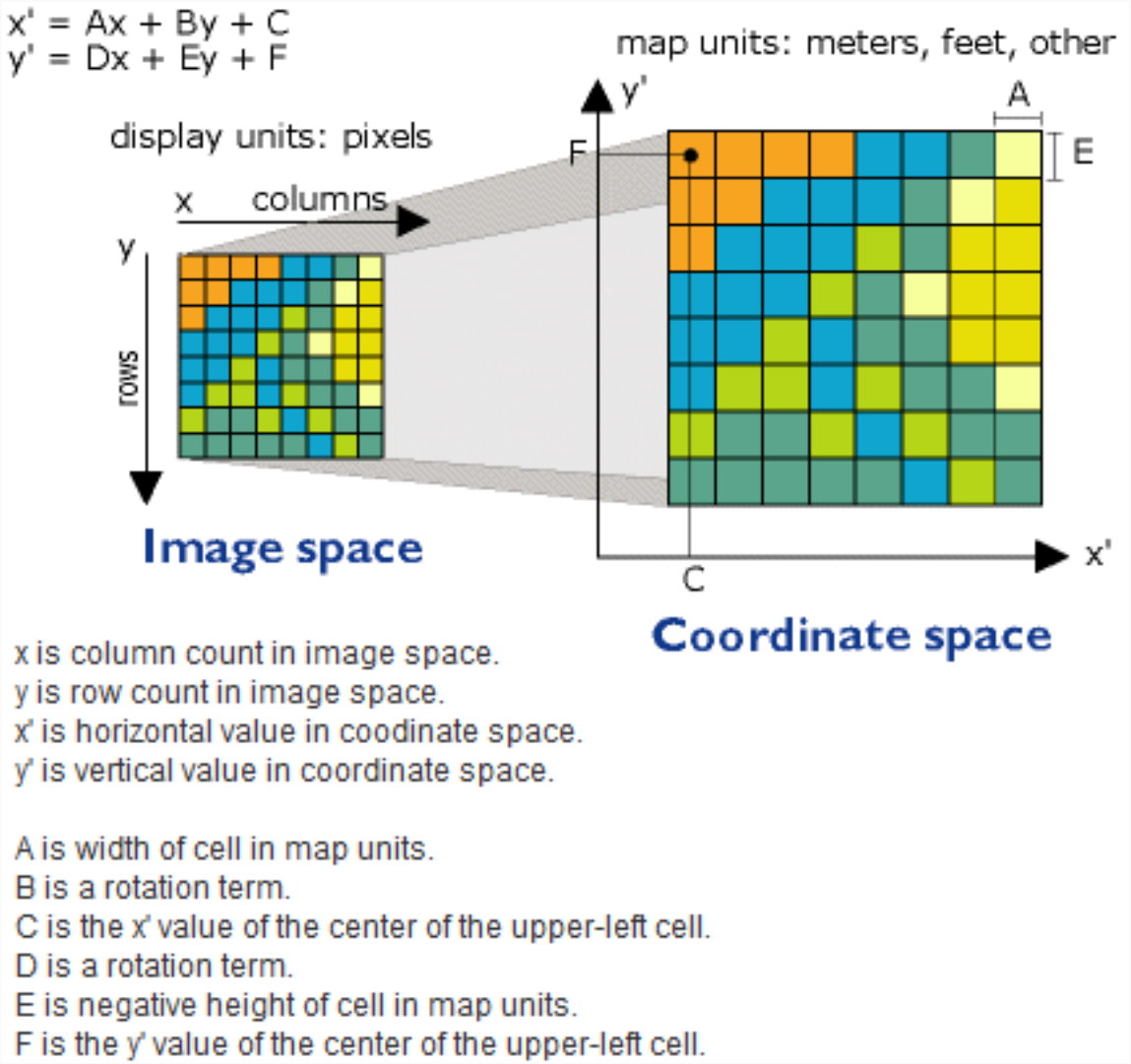

Figure 1: How georeferencing works. Image courtesy of Esri.

Making interpretations from an image

After that is where magic begins, the single most boring job description happens. You just have to manually trace each place of the image that you believe to be has some meaning (be a street or a lake). You manually have to convert this image into a vector representations of it. Not every information in the image is relevant to GIS tasks, it was not easy to build a program that can solve such problems.

We would like to extract streets from an image

Figure 2: An image from Google Earth. A few years ago these gray looking lines above each street were not available. Spoiler alert: They use deep learning.

ArcGIS

ArcGIS is the biggest application in GIS world. People often use GIS and ArcGIS interchangeably. The question is, why after all of these years Esri couldn't develop a tool to automate such tasks.

- Esri is a big corporate. It cannot just plugin experimental works on it.
- The field has accepted to use Esri tools as they are.
- Many labors were hired to do such lame tasks.

The technology was not mature enough, and the field didn't try to do anything to push that further. Proprietary software sucks.

Current state of GIS

"Nevertheless, despite over 30 years of effort, at the time of writing there was no commercial automatic or semi-automatic road detection system on the market and, to the best of our knowledge, no published method has been shown to work reliably on large datasets of high-resolution urban imagery." — Hinton

GIS $\neq$ tracing.

Just to be clear, GIS is not all about tracing. However, most of what is done in our case is really just that.

Remote sensing

Let us look at other examples and compare how remote sensing and GIS usually operate.

Figure 3: A satellite imagery of a vegetations. Image courtesy Google Maps.

Figure 4: A satellite imagery of a water body. It is hard to know what is

Using informations from remote sensing

NDVI (Normalized Difference Vegetation Index) is 'NDVI is an indication of the greeness in the image. - [**?**]'

$$NDVI = \frac{IR - R}{IR + R}, \tag{1}$$

where IR is the infrared band, R is the red band. The result of this equation is a number between -1:1. Numbers between 0.2-0.3 indicates grassland.

Using infortmations from remote sensing

Same applies in water. There is something called NDWI (replace vegetations with water).

$$NDWI = \frac{X_{nir} - X_{swir}}{X_{nir} + X_{swir}}, \tag{2}$$

where X_{nir} is the near infrared band, X_{swir} is the short-wave infrared band.

We have already mentioned that satellite imagery usually have multiple bands. You all now what RGB is, obviously. This what our eyes can see (cf. Stackexchange), and our camera can shoot, too. It turns out, objects experience different, distinct, behaviors on different wavelenghts.

Using infortmations from remote sensing

This is what remote sensing has contributed to us. Using specific wavelenght ranges, grass looks more greener, the water also becomes more clearer. So, we can actually detect water bodies, and vegetations using these indices from remote sensing. This is was one of our earlier works. More interestingly, the winner of DstI (a kaggle competition about satellite imagery), has used NDWI with a convolutional neural net to enhance his model quality. (the only different between the winner and most of the top 10 positions, was in using this index!)

Remote sensing and GIS

By now we should all be familiar with what remote sensing is, and what GIS is.

Let us restate the problem

We would like to detect x object from a satellite imagery. For water and vegetations, we have exploited intrinsic characteristics about the problem, and solved it. We do not have an NDSI (S for streets) index though. Also, hand made features don't work. It didn't work in any of computer vision problems.

Using a convolutional neural nets to in satellite imagery

We would like to detect streets from a satellite imagery. Let us explore previous works first.

- DeepOSM, from Open Street Maps
- DIGITS, from nVidia [object detection and semantic segmentation]
- Unet [Used extensively in Kaggle]

In our work we have used a modified version of Hinton's work, which is uses semantic segmentation instead of objects detection. Our network has shown to be perform very well using variety of image sizes.

Sample works

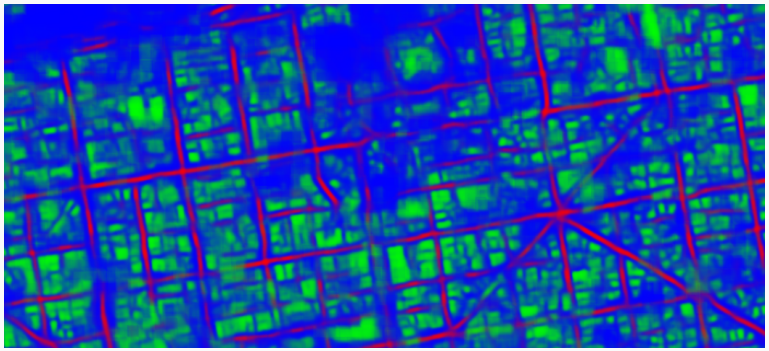

Figure 5: Satellite imagery, Khartoum

Figure 6: Satellite imagery, Khartoum

Different CNN architectures

Satellite imagery are taken from a different perspective, a bird view. The relative size of objects in satellite imagery is also different than that in other cases (this one is particularly devastating when using Yolo — a problem they have reported that their model suffers from.)

Other works has shown that, training a Faster RCNN from scratch has better mAP results than using pretrained versions on ImageNet, when applying this model in satellite imagery. However, the gained accuracy is not that huge.

Different CNN architectures

Satellite imagery usually are very high resolution (6773×6773) and more are very normal. Yolo was trained on a data set of sizes (416×416), VGG-16 on (225×225). So, you probably get the picture. You need to you reconfigure your data set.

You can train a Yolo on a data set of 416 × 416 to detect 1080 × 1080, but for sizes greater than that, it'd be much better to cut your images.

Deep learning on remote sensing — Problems we faced

If you have enough dataset, and computing resources, 30% of the times your models will work 100% correct. That was a joke.

One of the biggest problems we faced, is the lack of labeled training dataset. Satellite imagery are already available through USGS program for years, though you still need to label them. That's a problem.

In the past year, DigitalGlobe, and NASA too, have provided freely available satellite imagery for deep learning applications (came with labels). Actually, So, yes we have many of Imagenet for Earth dataset.

Preprocessing such dataset is another hurdle. Also, they are really very large.

Python — I have got you covered

The good news is, Python, as always has got us covered. Thanks to gdal and rasterio, you do not actually need to use any other GIS software to handle satellite imagery (in the case of OSM data set, you actually need to do that. If you would like to detect street, you also need to do that.)

Practical Advises

Problems we faced

We are still having many problems. Most of them are because this field is new in Sudan, so it is hard actually to find people who understand the potential of this field. No funding.

Most of the companies we worked with, they have a stupid tendency. They wanted us to build them something, without getting paid, after the work is done, we discus later how we can get paid [they either buy it from us, or share the revenue.]

In our case, our solution saves money, and have better accuracy. Yet, no ones care.

Problems we faced — GPUS

We do not have any GPU machines. We only use our laptops. That sucks.

You *need* a GPU to run it on your local experiment. We have tried not to do that, and the results were incredibly embarrassing. Use a GPU, save your life. (We have spent more than 10 hours trying to build caffe on a cloud computing provider, it was not good at all.)

You cannot use AWS or others to do your experimental works and final model computations! That's really just a huge mistake. Unless you have had such a large money (628$ per month for p2-large), which in that case, I'd advise you to invest in a GPU instead. AWS is not meant to to do any experimental works.

Using Amazon AWS

You can have upto to 150$ from AWS if you were a student (*.edu is sufficient). To activate AWS account, you need to be in any place except for Sudan (in the activation, they call you, and give you a number). Again, another problem. If your phone number, and your address didn't match (you wrote you were from Sudan, while your phone number was from Bahrain, your account is likely to be suspended–no worries, you can send them your ID, any bills to check for your address). Usually, you won't need any credit card to activate your student account, though because your account was suspended (the narrator: It will), you will need a payment method to activate your account.

Visa card

We have tried to use Wirex. Use it to activate a paypal account, and then after that activate your AWS.

Paypal doesn't work in Sudan. Use a VPN, and activate it on Saudi Arabia, or Somalia — it doesn't really matter.

If you have a relative, or a friend who lives outside Sudan, you can ask them for a Visa card, that will save you lots of time.

After that you probably have a working AWS. Now you need to use EC2 service. The steps are simple, you use an Amazon instance (DLAMI), which specify what GPU you need, IP address, hardisk, etc. Use p2-large (0.9$ per hour, 4CPUs), p2-x2large is more expensive, and the only different is in the CPU-you do not need that.

Oh, and after that you really need to get along with the command line.

Or,

Microsoft also gives 150$ for students on Azure. Google's Cloud Engine gives 200$ for every new user. All of them actually require a working visa card (not prepaid, nor gifted. You need someone outside this country, again.)

Paperspace

Paperspace is another cloud provider. They will give you a free 10$ when you create an account with them, and also a free hour. Wirex doesn't work on paperspace, you need to use your paypal account. Also, Paperspace uses a service to check the validity of your card. You need to use a VPN during that, since it will complain about you country.

Paperspace and Amazon: what to choose

Paperspace. For its simplicity. Literally, in just a few sec, you will find that you have created your instance (ProTip: remember to check for the static ip, and uncheck snapshot. Also, use Fastai pytorch version, and a hardisk of larger size than 50GB.)

Using AWS, you will be overwhelmed by the number of services they have (man, there are a lot of them). It is just very hard for starter to figure out what they want.

Paperspace is very simple. You will be asked about the very basic questions every cloud should ask: GPU version, hardisk, IP. That's it.

Paperspace and Amazon: what to choose

Paperspace is cheap compared to AWS. 0.6$ to 0.9$ using amazon. In 10 hours you spend 6 dollars using Paperspace, while 9 using Amazon.

Use Paperspace.

Training dataset revised

Now you need a data set to train your models. Amazon has this new feature called Earth, check it out. More or less the problem is solved.

Train your model on Imagenet to detect low level features such as contours, edges. Use a CNN architecture that is fast. Go for Yolo, very fast, and have reliable results. For small objects though, you need to modify a few things. Finetune it on your model, and you are probably good to go.

What's next

Revenue

You will do all of that, only to find out that no one's gives anything. Which is wrong. We need to fix that.

Big companies put it very simple: you build a product, we share the revenue. *On what universe does that make any sense?* If I can build it myself, what is your use? Marketing it? God Bless.

The natural way is that you have an idea or a prototype of something. You need to raise funding to support your work. If your work is novel, and can get money (which should be made clear through your business plan), it shouldn't be so hard to find sponsors for your work.

This is very difficult to solve, though such meetings can help us solve such problems.

Research, government, etc

No. Government won't support your work, nor there any grants for your research works. Pass.

Big corporate, Dal, Zain, Sudani. Haven't check any of them. Sometimes irrelevant, most of the times they higher the wrong guys. Pass, too.

You will probably have better chances to get a grant from Microsoft of Amazon or UNICEF on any poverty-related problems. You need to be either student, or in a research group.

Africa Center of Technology

I heard they offer computing services. Very interested to get in touch with them. A very good option, in case they work. GPU? I'm not sure.

Our future work

I don't know. It is very hard to tell. We will keep working anyway.

Data science community

Very nice idea. I'd propose to create Pydata Khartoum out of it, perhaps also make our first conference?

- Tutorials sessions on Numpy, scipy, etc will be very useful for the community.
- Solve the problem of Visa cards, cloud computing.
- Write. Archive your work, however small it was. It will help you in the future.

Practical advises on programming

- Always write scripts to automate your work.
- When it doesn't work, that means, you screwed up, not your machine.
- Do you really need to use Jupyter notebooks? For exploratory data analysis, sure. Most of the time it is useless. text editor + terminal = way better.
- Never use your cloud machine to do what can be done in your local environment.

Practical advises on programming

- Ask for help.
- Do not undermine yourself, nor your work.
- Sometimes, things only work on our machines. Everybody knows that. Do not be ashamed to use it as an excuse when things went wrong.
- If the problem is too hard, perhaps it might not be solve?

Final notes

This session was not meant to be technical. I have intentionally tried to be as less technical as possible. For that part, I'd rather prefer a tutorial. I have tried to focus on practical problems we faced, that every startup would face too. Solving these problems, will encourage more people to create startups, which means more job opportunities, and more resources.

May all your models somehow work, and your dependencies are all pip install

About our team — gndi

Mohamed Jaafar, Android Developer
Abdalsalam Albashier, GIS Analyst
Mohamed Yousif, -

We work on the field of geoinformatics. We analyse the data [MY] using GIS and remote sensing [AA], and deploy our analysis to end users [MJ].

We are certainly looking for other team members. Though, know it is quite hard to find any projects to work on.

We launched an internship program for college students. It was exclusive to UofK Surveying Engineering students. We wanted to start a systematic way of getting a job. We'd like other companies and startup to take similar steps.

Contact us

I'm always avilable through my email mmbusif@gmail.com
GitHub: https://github.com/adonese
Linkedin: https://www.linkedin.com/in/adonese
Location: Taif Block 22, St. 12