**Machine Learning Project Checklist (page 507)**

**The Problem:**

Essentially, I’m creating a program that uses a phone’s camera to identify survival plants, fungi, animals, feces, tracks, stars and medicinal plants. It can also hear sounds related to animals and identify them. It’ll be a ML program for survival, farming, and military fieldcraft uses. This is going to be a full spectrum program, with many facets to it: I’ve got to figure out things like GUI, the networking and security of multiple phones connecting to a server, linking the output to a Wikipedia, an encyclopedia or search engine, the actual machine learning algorithm, and the ranking/social proof system to allow to contribute. There are a lot of things and to be honest, I’m in over my head. But I can break down the problem into small parts and just attack that small part. So for now, I’ll focus on the Machine Learning Algorithms. Just being able to do this will be an achievement.

So, where the hell can the tool of machine learning come into play? These are the areas in which a machine learning algorithm is used:

* The ML model that can identify images harvested off the internet as well as images taken with the cell phone camera.
* A search algorithm that can determine what kind information is relevant to the user.
* An ML model that can determine the relationships between plants and group them into useful classifications.

There are a lot of problems that I’ll have to face, but right now, I’ve got to figure out how to simplify and limit the scope of the problem I’m solving. Don’t try to do too much. Specialize in a niche and become really good at it. I want to use Machine Vision and Hearing to solve everything, but right now, I’ve got to just focus.

The good thing is that there are already Machine Vision apps out there. It’s not far-fetched. What I’ve got to do is just make the best product I can make. I don’t have to be super original. In fact, there is already a program out there named **Leafsnap**. What I can do is make it better, more useful and expansive and with better marketing… Limit your scope and do the limited thing the best.

**Get the Data:**

Frankly, there are a lot of plants and fungi. At least 400,000 types. This is too big of a scope at the beginning. I need to structure the Model to be able to handle this many species… But at first, I’ve got to choose a small handful of plants and fungi in which to train the algorithm. Start small and expand from there. So where do I get the data???

I could get a lot of unlabeled data off the internet using a script. I have no idea how to write this script…. But I can learn!!! What is important is that I build an expandable and versatile framework.

Also, I can build a couple scripts that meaningfully alter the data (no Gaussian noise). It could be to alter the lighting, flip the image, stretch the image, etc.. I’ll have to research the specifics of this.

How much will be labeled and how much will be unlabeled? Can I train the lower layers of a Neural Network on unlabeled data and then the higher levels on the labeled data?

Is there already a preexisting database of plants and fungi or other images that I can train the lower layers on? Most likely there is.

Also, what is the minimal amount of data to train a successful algorithm?

**But this is a question I will probably need to ask the professors.**

Eventually I might have to go out and take pictures.

**Explore the Data:**

So I’m going to have a ton of pictures. Maybe hundreds of gigabytes of images. So I keep the data in a separate Solid State Hard drive.

I’ll have to make a copy of the data to play around with so I don’t fuck up the original data. As I explore the data, I’ll need to create a Jupyter Notebook to document the things I’ve done to explore the data. I’ll have to study Chapter 2 of the Machine Learning book to figure shit out.

Specifically, I’ll need to study and quantify some of these things:

* Name
* Type?
* % of Missing Values
* Noisiness and type of noise(stochastic, outliers, rounding errors, etc..)
* Possibly useful for the tasks?
* Type of distribution(Gaussian, uniform, logarithmic, etc...)
* Always document what I’ve learned

**Prepare the Data:**

Actually, this is the part where I apply different meaningful transformations to multiply the amount of data I have. Make sure to keep the original version of the data and then apply transformations to the copies of the data.

I’ll have to make sure to write functions for all data transformations for these reasons:

* In order to easily prepare the data, the next time you get a fresh data set
* To apply the same transformations to future projects
* To clean and prepare new data instances once your solution is live
* To make it easy to treat your preparation choices as hyperparameters.

I want to standardize or normalize the data. For me, I want to probably get the pictures to a standard sizing.

**Using the Deep Neural Network Model**

If the data is huge, you may want to sample smaller training sets so you can train many different models in a reasonable time. Try to automate this as much as possible. Automate as much as possible. Then compare and measure the performance. Next analyze the types of errors the models make. Think about what data would a human have used to avoid the mistake.

Also what I want to do is to train the lower layers of the model on pre-existing Computer Vision Data sets. Copy those lower layers. And then refine the model on specific data for my task at hand.

Next have a quick round of feature selection and engineering. Retest and then short-list the top three to five models.

Essentially I have a Computer Vision and Computer Hearing problem. What I’ve got to use is a **Deep Neural Network Model**. But which ones specifically? Possibly a **convolutional neural network.**

Issues to deal with in Computer Vision:

* Lighting Variability
* Pose Variability
* Intra-Class Variability: Alot of variability within a class and very little variability between classes.
* Occlusion. When part of the object is blocked(or occluded)

Some useful keywords that will be useful in my search:

* K-Nearest Neighbors
* Convolutional Neural Networks (Most accurate for this task)
* Spacial Invariance
* Local Connectivity
* Covolutional NN: Pooling
* ImageNet: 14 million images with many classes for learning lower layers.
* Capsule Networks

**Fine-Tune the System**

To fine tune the system, I need as much data as possible, especially towards the end of the fine-tune process. Of course, as in all programming, I want to automate and simplify as much as possible.

I want to Fine-Tune the hyperparameters using CROSS-VALIDATION.

Make sure to treat my data-transformation choices as hyperparameters, especially when unsure about them (e.g. should I replace missing values with zero or with the median value? Drop Rows? Resize images?

Unless there are very few hyperparameter values to explore, prefer random search over grid search. If training is very long, you may prefer a Bayesian optimization approach.

**Present the Solution and Validate in the Real World**

It’s an ongoing process of improvement and expansion. Also, the Machine Learning is only one part of the entire program. There also has to be the Android/PC/IOS app to take pictures, to communicate with other people, to form a community, etc… Also, I have to worry about things like building the infrastructure for the individual programs on people’s personal devices to network over 5G, wifi or whatever into a central server. There’s sales. There’s marketing. There is a lot. But I just need to do this and learn as I go. Move past limits. Be overwhelmed. And move forward.