Daily Thoughts and Experience Diary

AutoML (formalising and automating the stages in training a Machine Learning algorithm)

Developing machine learning models to solve problems is often described as more of an art than a science. This is another way of saying that it is very hard to guess how well a given technique will work on a problem. AI developers therefore spend a lot of their time searching over different machine learning architectures and hyper parameters trying to find ones that work well for their problem. This can lead to messy code and create a feeling like you are building on sand, as soon as you decide that one part of the model is working well changing another part may invalidate that choice. This has led to the field of AutoML which is a process of automatically searching for good AI architecture and hyper parameters. This approach is a good way to approach AI problems where instead of writing a single algorithm that you change you develop a python script that can generate different AI experiments with different architectures and parameters. Using this approach you can automatically run lots of different experiments and automatically search for the best approach. As you work on the project you are adding different alternative architectures to your python script so that you can automatically search over more approaches. In this way you never take a step back, every choice you add is at worst making your system try unhelpful approaches but the optimal design can be easily produced from your configuration.

The goal of this project is to take one or more of the AI algorithms described as part of the FastAI course (the leading course on Deep learning which includes a number of state of the art models for solving common machine learning problems). For each problem express the choices available in how the algorithm is implemented using a csv format (like a row in excel).

Make the resulting training algorithms capable of running on google colab and as a docker gpu container so that the machine learning algorithms can be easily trained on a cloud server.

# Monday 18th January

* Having a first look at the fast.ai course and trying to get it set up.
* Spent around an hour trying to set up a jupyter notebook using pip and couldn’t get it working.
* Tried using anaconda to install the libraries and it seemed to work but when I got my notebook opened I couldn’t figure out how to import the files from the course into it.
* Decided to switch over to trying to use google colab to open the chapters of the course.
* Started reading the introduction chapter and it seems really interesting.
* Only some of the built in python scripts in colab seem to be working, a lot of them show errors of either a library not being found or a variable not being assigned.
* Because of this I'm not really able to run most of them.
* The first script for training a model worked but it took nearly an hour to train it, so I’m thinking that maybe i'm using the wrong GPU.
* Finished reading through the introduction chapter and enjoyed reading it. Looking forward to the next chapters, however I'm not sure if I would be able to write an algorithm myself yet or even how to go about that.

# Tuesday 19th January

* Just reading over the second chapter of the course today.
* Didn’t get much accomplished because I was lazy, only got part of the way through chapter two.
* Was sort of sidetracked by my Web Technologies module because I was watching the canvas videos and trying to do the lab practical.

# Wednesday 20th January

* Going to try to get stuck into chapter two of the course today.
* Decided I might switch over to using gradient instead of google colab. Trying to set up the notebook now.
* Switched over to gradient and it seems a lot better, things seem to be running correctly as of now.
* One of the scripts won’t run and so all of the scripts after won’t run. I’ve been looking around on the forums etc. and can’t find a solution. I will have to just not run the scripts. It is quite frustrating.
* Looked around more for a solution, none found.
* Watching youtube videos about the fastai library, I want to start experimenting with code ASAP to get my head around using the library and some simple model creation.

# Thursday 21st January

* Today my plan is to go over the video lectures for the fast.ai course while having the the notebooks open to experiment with the code
* I think I need to learn a bit more from the course before I try to start making a lot of code because otherwise I will pretty much just be copy pasting code without understanding the meaning behind it.
* Currently watching the fast.ai video lectures, noting down some important code and its meaning in a document so that I can go over it later and experiment.
* Starting to feel better about the practical side now as I’m getting some exposure into how the code will look.
* Trying to create a simple image recognition program for detecting different types of rodent. Running into a few roadblocks with syntax at the minute.
* I'm getting stuck with file handling and scraping the images from google images, I will go over it tomorrow and try to get it working.

# Friday 22nd January

* Had my first meeting with John today. It went well and he mentioned to me to contact him if I’m experiencing any roadblocks such as not being able to get certain code to run.
* We also determined what area of ML each person in our group will be focusing on. I will be focusing on Computer Vision which I am happy about. I find it interesting; it's very easy to see it working which I think is very useful and visually appealing.
* We discussed what each person would base their how-to guide on and I will be doing mine on how to get gradient and fast.ai up and running. This will be fun as I haven’t actually got it up and running yet. Although that is probably a good thing as I can list the problems I experience and that should be useful to anyone else who is trying to get gradient set up.
* Currently reading up a bit on python because I haven’t used it since GCSE Computer Science.

# Monday 25th January

* Not much progress as I was catching up with videos and work for other modules.
* Plan for this week is to try to get gradient up and working properly and start on a bit of code to get my head around fast.ai.
* Got an email from John with a pdf attached which went over the importance of AutoML and how it is used in Enterprise which was an interesting read. Took some notes from it such as the goal of AutoML and some key areas that should be focused on: usability, stability etc.

# Tuesday 26th January

* Went through one of my group members’ test projects in a notebook and everything ran perfectly, with what seemed to be identical code to the fast.ai notebook. Starting to think that my problems originated from imports not working correctly or me not running them for some reason.
* Originally I thought that the problem was my Bing search API key, however, even after swapping my group member’s key out with my own, the notebook ran perfectly. I’m now possibly over this roadblock in the code, but the only way to know for sure is to try to set up my own application.
* Also today, going over the lesson 3 lecture on the fast.ai course, which by now is getting more in depth in the code. Learning about pytorch syntax and about tensors in pytorch.

# Wednesday 27th January

* Rewatching part of yesterday’s fast.ai lecture to remind myself. Learning about gradient descent now and pytorch code.
* Today I think I will try to create the image recognition app.
* I'm trying to set up my app but for some reason now the pip installs and the imports aren’t working. It's confusing because the exact same code was running yesterday.
* I think the problem was that I was trying to create a new notebook from google colab that wasn’t connected to the main fast.ai repository meaning I couldn’t import properly.
* Started a new notebook from gradient and the code is now running.
* Ran into a problem when downloading images where I created a path but then an error occurred because of a spelling mistake in a method name. This led to the if statement not being activated because the condition was “if not path.exists()” meaning that the downloading wouldn’t happen.
* Fix: I just had to delete the empty directory from the main notebook repository so that the code would run and it works now.
* Running into an error when trying to train my model.
* The notebook kernel keeps dying and I’m getting memory allocation errors.
* I think I will finish for tonight and try to get the errors sorted tomorrow.
* I think possibly the problem with memory could be due to the host GPU, I just noticed the host was a CPU so I will need to try to change the host to a GPU and see if that fixes it.

# Thursday 28th January

* I have restarted my notebook and switched the machine type over to a free GPU after it had automatically switched to a CPU. It seems that the training is now working, so I’m assuming the CPU host doesn’t have enough memory or just isn’t powerful enough compared to the GPU host.
* I was able to train my model and clean my dataset. I incorporated voila to make into an actual usable application within a jupyter notebook. It classifies an image of rodents between 3 types of rodent: Rat, hamster and capybara.

# Friday 29th January

Meeting with John

* Todays meeting:
  + Discussed what we should be doing currently to progress.
    - What I need to focus on is trying to create an object detection program that will detect objects within an image and put a box around them with a label of what the object is.
    - The program will use the YOLO algorithm and John suggested using YOLOv4.
    - I will need to look into the YOLO algorithm and create a YOLO style application in a Jupyter notebook.
  + We also discussed the overall goal of our projects
    - To be able to take a labelled dataset and input it to our program
    - It will then compare different algorithms and find the most efficient for the problem and write that code in a jupyter notebook and run it.
* Read some of this paper: “YOLO Nano: a Highly Compact You Only Look Once Convolutional Neural Network for Object Detection”. Just trying to read a bit about YOLO and also learn a bit of the academic vocabulary.

# Monday 1st February

* I was reminded today by email that I have to complete and hand in my how-to guide this friday and I wasn’t aware of that, and so today I have been getting started on it by writing the basic introduction and researching other how-to guides that were linked on canvas.
* My how-to guide is called “A guide to set up fast.ai + Gradient for Machine Learning”
* I am currently researching fast.ai to figure out how I am going to explain it and describe its usefulness to the audience of the guide.

My objective with this how-to guide

* My goal for this guide is to help anybody who is trying to engage with Machine Learning by getting them started with the tools to begin their journey(fast.ai + Gradient).
* A lot of the facilities that are used in the field of Machine Learning development and Research are made for specialists. Therefore they are hard to use and hard to set up for non-specialists.
* During my start-up process, I encountered many bugs and errors with both fast.ai and also with Gradient. It was frustrating when I was trying to figure out how to fix them, but now that I have, I want to make it easier for those who are trying to accomplish the same thing in the future.

How do I plan to do this?

* I want to make this guide in a similar way to existing guides because they were clearly successful and helpful.
* One guide I had a look at was on the fast.ai website [here](https://course.fast.ai/start_gradient).
* I think that laying it out in clear steps would be effective at making the audience keep reading and be able to follow easily.
* However, I think I will try to make the steps as simple as I can, to increase accessibility further because more accessibility is generally better for more people to engage with content.
* Finally, based on the previously linked successful guide, I will need to include an attention-grabbing piece of text to engage the viewers and make them more inclined to read on and investigate my guide.

# Tuesday 2nd February

* Today I will again be focusing on writing up my how-to guide.
* I am now going through the instructions for creating a notebook.
* I plan to finish the project creation section and then go through the problems I faced during my journey and the solutions I found for them in order to help future users.
* ISSUE: Now when trying to set up a notebook it isn’t working so I’ll have to figure out how to fix it.
* FIX: I was trying to create a project rather than a Notebook.
  + This issue was an easy fix in terms of simplicity (I just had to create a Notebook instead) but it was an issue that didn’t show any errors, meaning it could be hard to figure out what the problem is.
  + Including small fixes like this will allow the viewers to get around the hurdles and get started faster than if they were trying to do this by themselves.
* Now I’m trying to figure out how I’m going to lay out this guide in terms of a website.
  + It needs to be clear and visually appealing to the viewer. (Otherwise they will just click away).
  + To do this I think I will have multiple named pages with clear purposes to allow for a concrete structure throughout.

# Wednesday 3rd February

* Downloaded one of the free templates from [Free responsive website templates | CSS, HTML5 web templates](http://www.html5webtemplates.co.uk/).
* I’m now trying to add all of my information from my how-to guide into the template.
* Firstly, I need to figure out the best layout for my material for any users.
* It is likely that my website will resemble a common website layout:
  + Home/../../Contact Info
  + This is because it is both used and works well on many other websites, so it is clearly a successful way to organise a website.
* It will probably consist of 3-4 tabs/pages.
  + I don’t think I want it to be a very large collection of pages as that may encourage viewers to quit if they think it will take too long or be too complicated.
* I'm obviously going to need a page for the tutorial itself- I’ll call it “Getting Started” or something similar.
  + Again, this is a standard name for such a webpage so it is likely going to be effective and also will be familiar to the viewer.
* Finished the “Home Tab”, it is just a basic overview/introduction of the tutorial including what both fast.ai and Gradient are.
  + I might need to add an attention grabbing piece of text as I mentioned in my plan.
* Started work on my “Getting Started” tab.
  + This will contain the tutorial content.

# Thursday 4th February

* Was at work all day today so my progress was impeded a bit.
* However, i’m going to try to finish up my how-to guide website tonight.
* Currently just finishing off the “Getting Started” tab by adding in the images and finishing off the text.
* I’m also noting down some ideas for information to include on a “My Goal” tab.
  + It will be a page about my reasons for creating this tutorial and what I hope to achieve with it (mainly to help students of the fast.ai course by allowing them to get started with the chapter notebooks and work alongside the lecturer).
* ISSUE: Now when I open the Notebook, none of the fast.ai material is in it for some reason.
* FIX: I just had to create a new Notebook and it seemed to work fine.
  + This is yet another bug that takes time to figure out how to solve by yourself. It took me at least 30 minutes and I was familiar with the process.
  + So hopefully, including this simple fix will help any readers of the guide if the error occurs for them.
* I added in the aforementioned attention grabbing statement at the top of the home page.
* I have finished the website now-
  + Layout is Home/Getting Started/My Goal/Contact Me
* Tomorrow I will be able to get back on track with trying to create a YOLO implementation for object detection. I’m eager to get a YOLO notebook up and running.

# Friday 5th February

Meeting with John

* We went over the how-to guide and how to get marks and create a valuable guide.
* Learned that I need to include more info in my diary.
* Changed some of my diary just by including more detail about what I had already written.
* Then we went over our current goals.
  + Mine is creating an implementation of the YOLO algorithm in a jupyter notebook.
  + A tip John mentioned was simplifying existing code by writing it out in plain english and then refactoring it.
* John also mentioned that as we are now getting more into the technical aspect of our projects we should share code (our own and existing) with him so he can give advice to help me understand it.

# Monday 8th February

* No progress- working on other modules today

# Tuesday 9th February

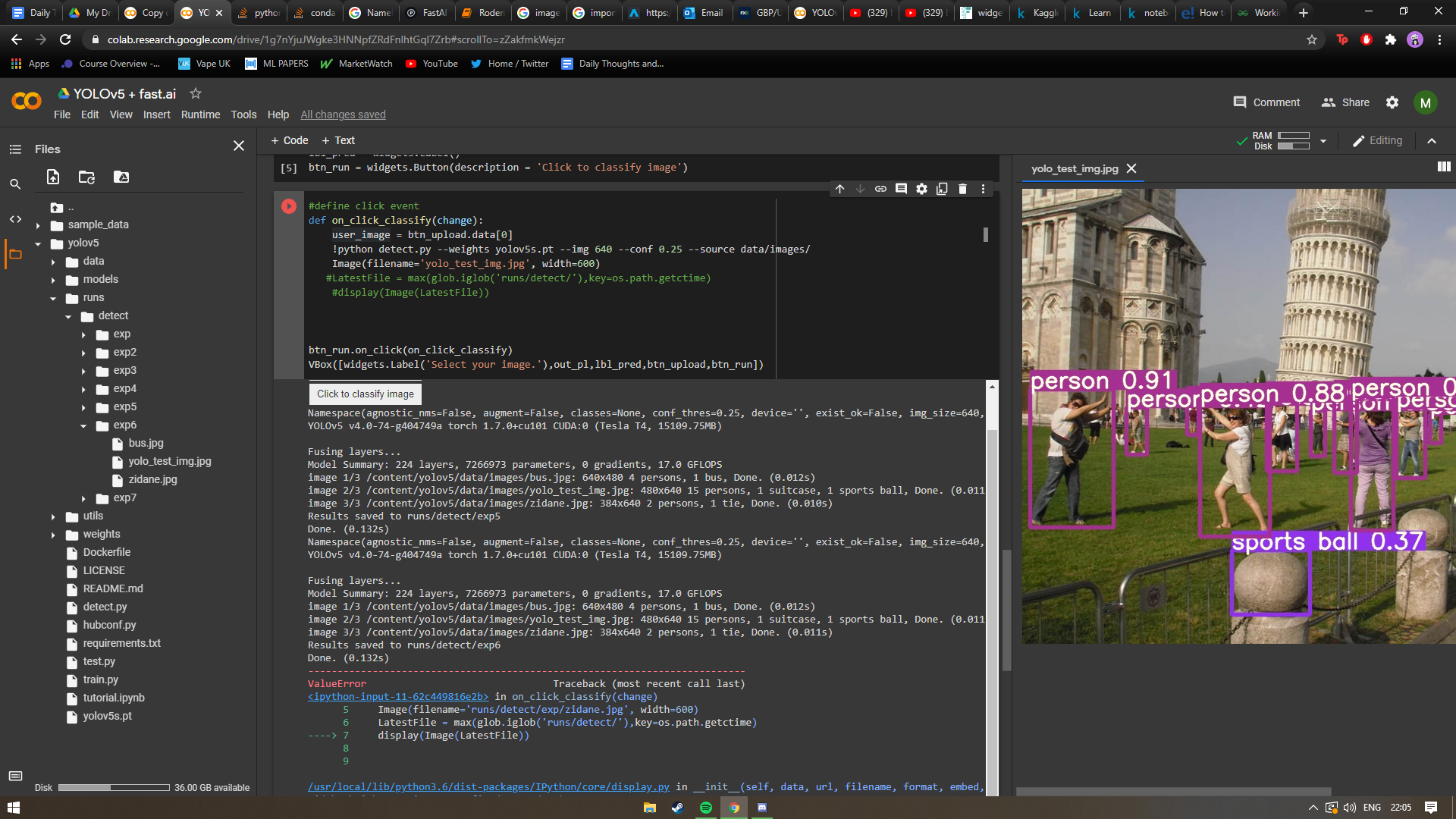
* Today I’m researching YOLO and implementations of it to try to get my head around it.
* I want to have an understanding of :
  + How YOLO works abstractly.
    - I am watching videos explaining the concept of YOLO and the main principles.
  + An appropriate structure for the code.
    - I need to look at existing implementations of YOLO.
* It appears that YOLO is built on top of a neural network architecture known as darknet.
* However, I believe that you only have to use the darknet to train the model, and then you can build the jupyter notebook based on whatever framework you wish (pytorch/fast.ai for me).
  + I’m following a [guide](https://pjreddie.com/darknet/yolo/) on getting started with yolo/darknet.
  + I am currently trying to install darknet and the associated YOLO weights file from a bash terminal on windows.
  + I’m just trying to see if I can install and use the YOLO model before trying to implement it into a jupyter notebook.
  + Some parts of this guide are not working for me, trying to get around them but I’m going to have to try to find fixes.
  + Turns out that the guide used only works on linux so certain commands will not work on my windows machine even though i'm using an Ubuntu terminal.
    - FIX: have to use software called CMake to overcome this issue.
    - CMake did not work, now I'm going to try VCPKG.
    - Looks like all of this was for compiling darknet for training on a local machine, but I want to compile it on a cloud GPU so I am going to have to look into methods for doing that.

# Wednesday 10th February

* Today I’m going to continue researching existing code implementations of YOLO.
* Also, I need to figure out a way to actually compile darknet without using my local machine (with gradient or google colab).
* I have found a [Yolov5 notebook](https://colab.research.google.com/github/ultralytics/yolov5/blob/master/tutorial.ipynb) that can be opened in colab, it's also implemented in pytorch, so it looks promising. I’m currently running the cells in it.
* When I train the model in the notebook, I have to create an account with [Wandb.ai](https://wandb.ai/) which allows me to use their API when I train the model and shows me all of the data and also graphs of the results.
* After finishing running that notebook. I’m not sure it's exactly what I’m looking for. I want a notebook that will simply run YOLO on an input dataset, and I’m not sure how I would do that with this notebook.
* I will either have to refactor it, or look for something else.
* This notebook also only uses pytorch and doesn’t use fast.ai, but at least it will be easier to refactor pytorch with some fast.ai rather than something else like TensorFlow/keras.

# Thursday 11th February

* I'm going to start off today by having a look at refactoring that notebook from yesterday, but if it doesn’t seem feasible I’ll have a look at other implementations.
* The plan is to combine parts of the code within this notebook to train the YOLO model and then use part of my simple image classification notebook’s code to allow for user input of an image.
* If I can get the user datasets to work, the next step will be to have my own object classes trained so that I can detect custom objects with the model.
* I have started making a notebook in colab, and have imported the fastai prerequisites as well as the YOLO github repo.
* I have also pasted in a test inference from the YOLO notebook to see if it would run and it was a success.
* Now I am going to try to upload an image of people and see if the model will detect the objects within the image.
  + I will need to create the upload button and run the detect function on it to try to make an inference on my custom image.
  + I will copy the upload button code from my image classifier and slightly alter it.
* Having trouble trying to convert the event function to run detect.py.
* Contacted John about the problem converting the event function and he helped me out, it was just an accidental copy/paste and a matter of restarting the notebook.
* I have now got the button working with the yolo model, but I need to change it so that the uploaded image is the one that the inference is made from.
  + I need to figure out how to get the user’s image from the upload control.
  + Then run the detect command on that image.
* I got it working with my own image:

*Before:*

*After:*

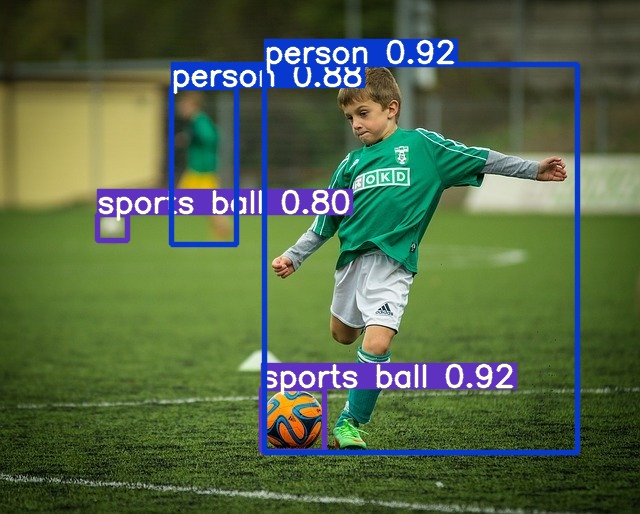
* The only problem is that it's really not that easy to use because you have to upload your image to the colab notebook and then insert its file name into the detect command.
* However, it was really meant for a proof of concept; just me testing if I could use my own image and get it to make an inference.

# Friday 12th February

Meeting with John

* Went over the three assessments in the module:
  + How-to guide
  + Blogpost
  + Diary
* My goal is :
  + Get the Yolov5 notebook running and try to train my own dataset of 128 images.
  + Want to use the existing pre-trained model, and get a dataset of obsequre images that wouldn’t be in the original training dataset.
  + Train and validate with that dataset to specialise.
  + Once that is done I will want to train yolo for new classes of objects, then I’ll want to make the notebook more understandable and easier to read.
* Possible blogpost inclusions/ advice from John:
  + Troubleshooting section- when I encounter a problem, mention it and its solution
  + Make notebooks easier for people to understand
  + Someone should come along and read the blogpost and be able to get up to speed much quicker than I did originally, then be able to easily carry on with the work on the notebooks etc. (accessible and easy to build on top of).
  + Want to write this as I'm going.
* Don’t forget to get help from John when needed.

# Wednesday 17th February

* Have been quite pre-occupied this week so I haven’t made any progress.
* Today I will have a look at refactoring my notebook to take an input dataset of around 128 images as discussed with John in the meeting on Friday.
* I have had a look at the notebook and it seems like it's going to be awkward because I am going to have to figure out a way to upload a dataset through the fastai.widgets library upload control. However, I will also have to upload that dataset to the colab notebook’s file structure so I will have to determine a way to integrate the upload control with google colab (I don’t imagine this should be terribly difficult.)
* I think for the sake of initial simplicity I am going to try to manually upload a dataset of images to colab just to test the model on them.
* I downloaded a small 10 image object detection dataset from [kaggle](https://www.kaggle.com/kkhandekar/object-detection-sample-images).
* I mounted my google drive to the notebook and uploaded the dataset.
* My drive and the data are showing but I cannot get the detect.py command to find them.
* Ok, I fiddled about with the path within the detect command and it worked and ran on all the images within the dataset.
  + However, it still weirdly gave me the file not found error even though it worked.
* Some of the images were classified pretty well:
* However others were pretty miserable:
* Looks like the model will need some training on more varied datasets so my next task will be to find a large dataset and figure out how to train the model and validate it etc. on that dataset.
* Tomorrow, I’ll need to ask John if there is a specific format of a dataset I will need to use for the train command. I will try to get it working before asking him.

# Thursday 18th February

* I have downloaded a free dataset of flowers across the UK, but I’m not sure how I will be able to go about training the model on this because i'm unsure about how to generate new classes.
* I downloaded the images, the data segmentations and the imagelabels.mat but i’m not sure how to work with these files to train the model. I'm assuming it isn’t going to work.
* It looks like this is going to take hours to upload to google drive. So it looks like I’m going to have to figure out what else to do while they are uploading.
* I’m not sure how to proceed in the mean team so I will have to do some work on my other modules.
* Upload was successfully finished, now I’m going to attempt to train just swapping out the file name in the train command.
* Unsurprisingly this didn’t work, I probably shouldn’t have wasted so much time trying to upload this dataset.
* I'm going to look for a small dataset with images and labels and try to get it trained on something just to see what format is acceptable. I can’t find out what way it has to be laid out from the tutorial notebook.
* I’m struggling to find a small labelled dataset, all of them seem to be large.
* During my search for a dataset I was looking into yaml files and it seems that for yolov5, a yaml file is needed for declaring training, validation, test subsets and classes.
* I think it should be feasible to copy and alter the coco yaml file to something that can make use of a dataset I get, but I will need to have a dataset that is labelled in classes.
* I'm thinking of creating my own dataset via web scraping.
* My current plan for the future is to possibly create my own labelled dataset(or find a suitably labelled one) and then refactor the default coco yaml file into one that will include my dataset’s information and then to train,validate and test using that yaml file.

# Friday 19th February

Meeting with John

Create Notebook for training yolov5 for custom datasets:

* Transfer learning- use a pre-trained model.
* Pass in a small number of images unused in the original training set in the write file format.
* Get it to run on a small size dataset.
* Pass in a training set to validate then use new images as well.
* Do it manually on a small dataset before going to a large dataset.
* Understand training code: file formats etc

## Goal

* Make self contained packages that will work for 10 years (long time).
* Overall system-
  + Users only need to know about adjusting hyperparameters and the system will compute everything else automatically

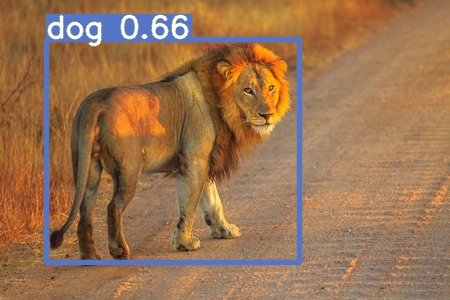
## Blogpost

* Explain the vision John was talking about : ai researcher with exe and self contained subsystems.
* Similar to fast.ai-
  + If the fast.ai course didn’t exist I would be spending a long time looking through pytorch documentation. Explain all of the things that I have learned and what helped me.
  + Fast.ai gets you going quickly and gives you a foundation, I want my product to be like that.
* Allow someone to be able to easily launch and build upon what I’ve done.
* If someone was given this, it should be user-friendly and someone could understand it and get to grips with what I’ve done in about a week of looking through it.
* The goal for the blogpost is to increase speed for someone to build upon my work.
* Lay out this vision of the project.
* Want it to be meaningful and long lasting.
  + Don’t want it to be abandoned after I’ve finished.
* Someone will be able to easily identify what needs to be done next.

# Tuesday 23rd February

* So today I'm reviewing the coco128.yaml file stored in the file structure of the yolov5 notebook. My plan is as above stated: figure out how to alter the yaml file and then have my own structure within it to be able to train my own dataset.
* I’m going to start by trying to create a small dataset of images of multiple classes and then label it in the correct format for the .yaml file.
* I’m still not exactly sure of the labelling format because it isn’t indicated in the yaml file. My assumption would be that within the main dataset directory each class has a named folder and the associated images will be within it.
  + This will be how I will structure it for my first attempt.
* I made 6 folders containing images of african animals(1 training, 1 validation for each):
  + Lion
  + Hyena
  + Zebra
* Now I’m going to upload them to google drive for use in the colab notebook.
* It looks like the yaml file accepts only one directory/file for training and one for validating, meaning that I'm unsure of how to use all 3 classes and also have them labeled.
* The names of the files in these folders however, follow a format such as e.g. Lion\_1.jpg, Lion\_2.jpg etc.
  + Perhaps just the name of the image could be used as a label and all of the training images could be put in a single folder and likewise for validation.
  + I’m still just not sure how the classes are associated with the image files, and it doesn’t seem like the yaml file controls that.

# Thursday 25th February

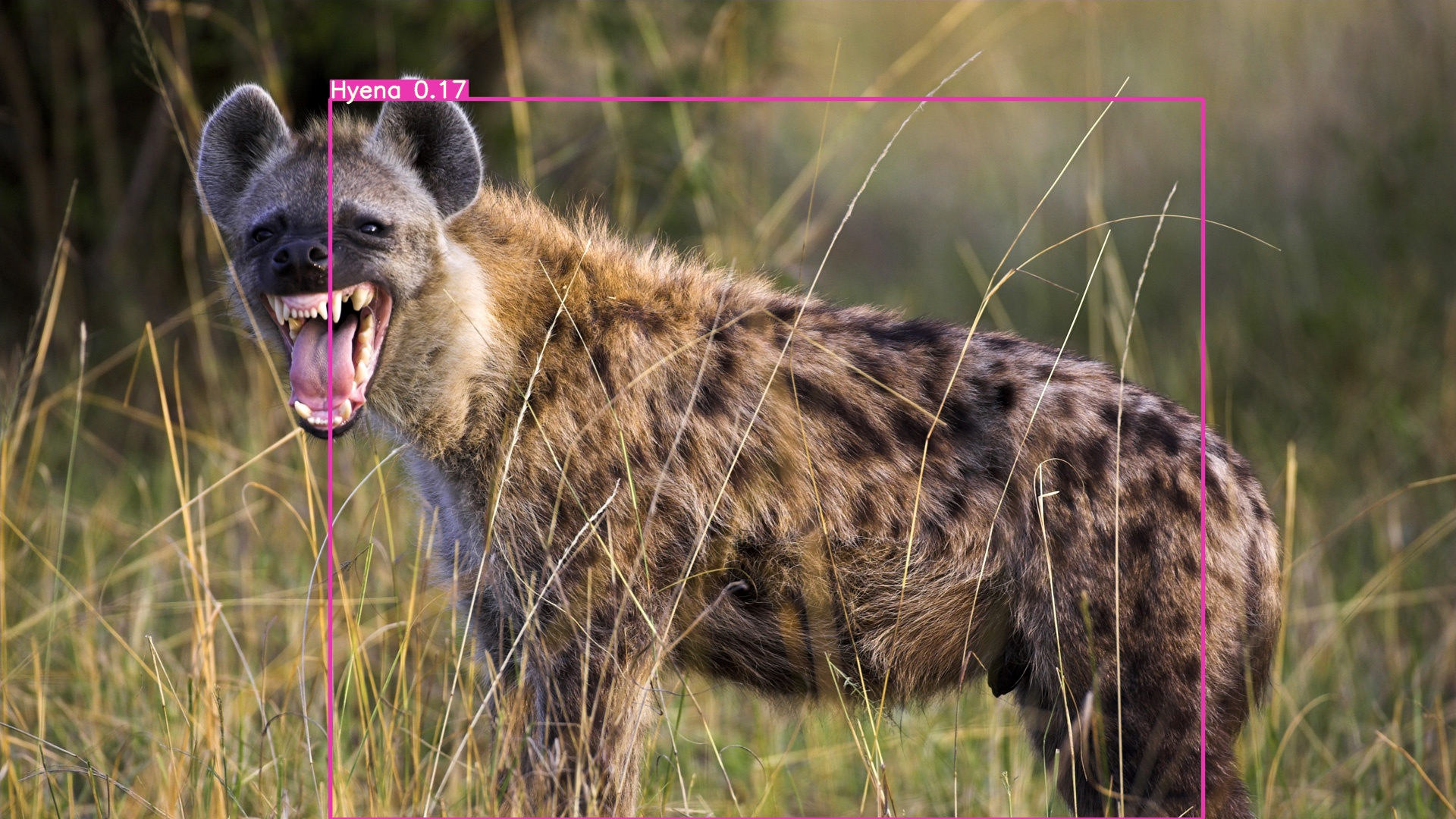
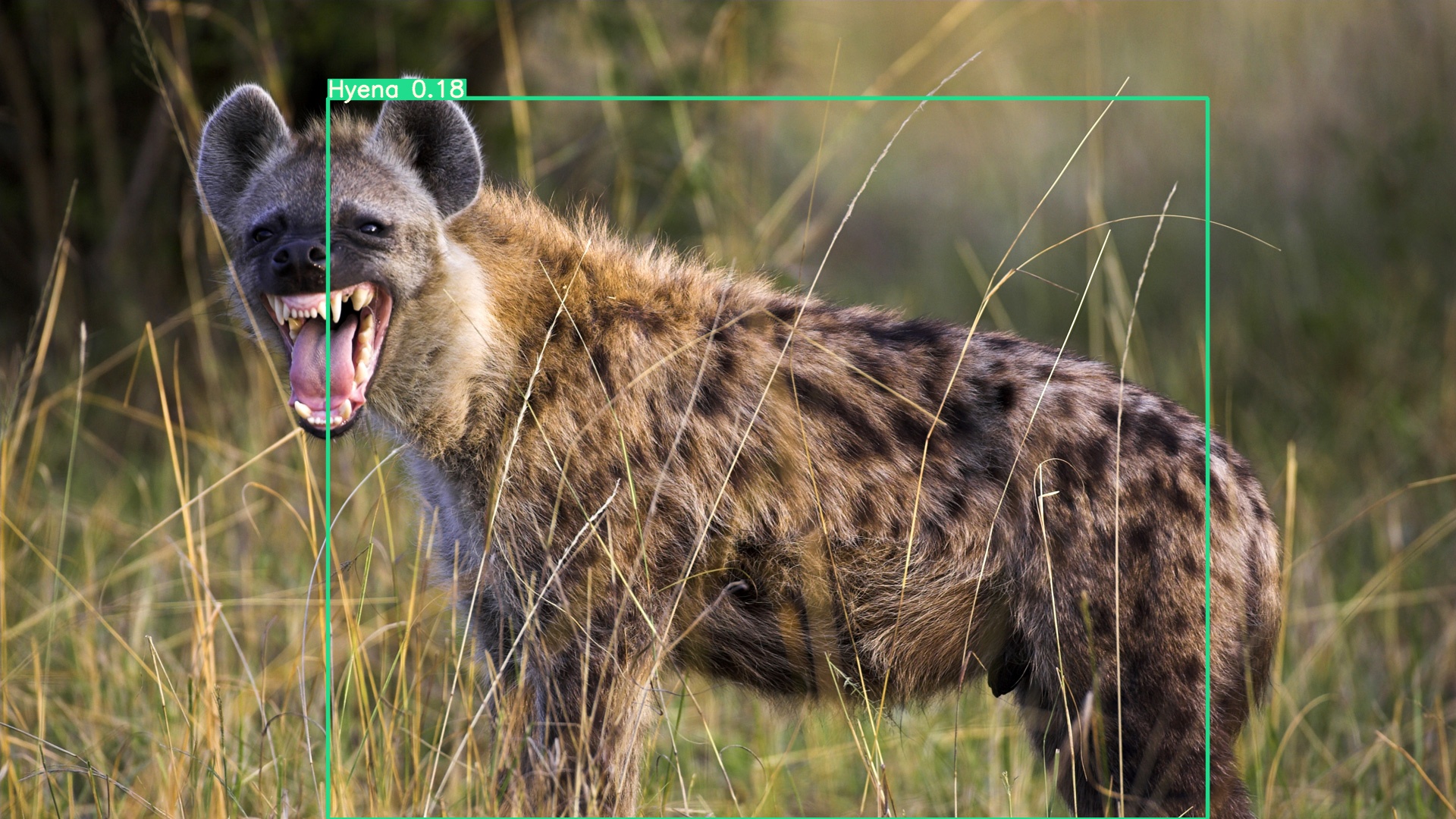
* Today I’m going to create a .yaml file for training
* After creating the yaml file and fiddling with the file paths I got it to attempt to train. However, it gave me an error indicating that it couldn’t find the labels for my images. I sort of expected an error like this because I didn’t give any labelling information.
* I contacted John with the issue of how the labels are inputted and he mentioned that in the coco.zip file there is both an image folder and a labels folder. Also, in the error it gave me a link to a github page for [training custom datasets](https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data) which mentioned some tools for labelling data.
* However, I went looking for a better labelling tool and found a [youtube video](https://www.youtube.com/watch?v=GRtgLlwxpc4&ab_channel=DeepLearning) that mentioned a [tool](https://www.makesense.ai/) that will export labels in the yolo format which is exactly what I'm looking for.
* So now I’m going to try to set up this tool and label my data.
* I have finished labelling the images now and have changed the structure of the dataset to fit the required format for YOLOv5.
* Now I just need to upload my new dataset to google drive and alter the yaml file to fit it.
* I keep getting an error when trying to train saying that it can’t find a .cache file.
* It seems that a .cache file has been created within the labels folder and it is trying to look for the labels in .cache file instead of in the labels folder. I’m going to delete the .cache file and see if that fixes anything.
* The .cache file is just created again at runtime.
* I have put the .yaml file in the same Yolov5 folder directory as the other .yaml files based on the youtube video previously mentioned.
* So it seems that the issue was caused by the file structure of my dataset directory. I had each class' images in a folder, rather than having all training images in the same folder eg:
  + Dataset->
    - Images->
      * Train/
        + Lion/
        + Hyena/
        + Zebra/
      * Val/
        + Lion\_val/
        + Hyena\_val/
        + Zebra\_val/
    - Labels
      * Train/
      * Val/
* However it should just be :
  + Dataset->
    - Images->
      * Train/
        + All training image files here
      * Val/
        + All validation image files here
    - Labels
      * Train/
      * Val/
* After this fix, it trained successfully.
  + Now I will see if I can make inferences on some new images.
* Before:
* After 5 epochs:
* So this is a pretty terrible inference, but I sort of expect that when I have only trained for 5 epochs. I was only testing to see if I could get the train.py to work with my files.
* Now it's time to train for around 100 epochs and see if it makes enough of a difference.
* I don’t want to overfit the model, being such a small dataset, so I will try only 100 epochs first and then compare to 5.
* After 100 epochs:
* So it seems like it made 0 difference contrary to my expectations.
* In the youtube video previously mentioned, the creator used the weights that are outputted after training to detect, so I tried that with no further training:
* So it's clearly not perfect but at least it is giving me confirmation that the classes are working. Maybe I just need to train for an additional few hundred epochs.

# Friday 26th February

Meeting with John

* End product: We want all of our code in a folder that will be free from external influence with training-
  + Sort of like an app with metadata.json etc.
  + Someone with no programming knowledge should be able to train custom models.
  + Some with programming knowledge but no ML knowledge should be able to train custom models and also integrate the libraries into their own projects.
* Put one of my training images through inference to see if it gets it correct- it should.
* Discuss with Jamie what i'm going to do with my yolo notebook.
* First half of the project is getting a trainable model.
* Second half is taking the notebook and making it user friendly for someone to deploy and use in this self contained, packaged way.
* The product should be able to be handed to someone and they can run in without internet access.
* My next thing to do is package it together and use a command line interface to generate a notebook from source code.
* User perspective:
  + Download zip file
  + A command line program can convert trained models into something they can use/ deploy.
  + They can do these steps without knowledge of programming etc.
* Additionally, John mentioned that the low accuracy of my model is likely due to the very small size of the dataset rather than the number of epochs trained.

# Monday 1st March

* Today I will begin by reviewing my notes from Friday's meeting with John.
* I think I might want to make a start on the blogpost today.
* Firstly, I’m going to quickly try to make an inference on one of my training images as John suggested in the meeting. It should have a high accuracy.
* It doesn’t look that my training has saved when I reconnected to my colab notebook so I’m training again for 100 epochs.
  + Original:
  + After training:
  + So it seems that even though this image was in the training set, meaning the bounding box was predefined, the confidence is only 0.17.
  + This leads me to believe there is something wrong here.
  + I’m going to train for another 100 epochs to see if any difference will be made.
  + After another 100 epochs:
  + So the confidence increased by 0.01. Not very impressive; I assume I'm doing something wrong with these inferences because this accuracy should be above 0.9 I believe, considering the bounding boxes are already defined.

# Wednesday 3rd March

* Today I will be working on my blogpost as it is due to be finished by Friday.

Approach

* My approach for this blogpost will be to improve upon my existing how-to guide based on the feedback I received from John.
* Building upon that, I will explain my work that has been completed after the creation of that guide and how I have got to this point.
  + I want to make the guidance succinct and clear for readers so as to allow them to follow it easily. My thought being that readers will simply get bored or uninterested if the material is too drawn out.
  + Moreover, clarity of the material will allow for the readers to get up to speed with my project much quicker.
* Additionally I will outline the goal of the project:
  + A self contained package that will allow non-technical users to be able to train and deploy custom datasets without knowledge of the underlying code.
  + This package will be a subsystem within a larger AI researcher that will automate all stages of creation and deployment of a ML model, including dataset creation, training and deployment.
  + I think that outlining this goal will be effective in giving the reader a context in which they can see the importance of the project. Once they can recognise that I believe that they will want to work on/improve my project
* The reader should be able to build upon my work:
  + This blogpost should be able to be given to someone and they will be able to get to grips with the content in around 1 week of time.
  + They should be able to easily identify what needs to be pursued next within the project and be able to make progress by themselves based on what they have learned from the blogpost.
  + The blogpost should be laid out in a user friendly way.
    - I want to keep engagement and reader attention high.
    - It's clear from other guides such as [this one](https://www.w3schools.com/html/html_examples.asp) from w3schools, that to make a successful guide a user friendly design is important.
  + The point of creating this blogpost is to ensure that my work will continue making progress without me, and to make the process of understanding how to do that quicker.

Issues

* My original plan was to build upon my how-to guide. However, my original guide was based on how to set up gradient and fast.ai in order to learn alongside the fast.ai lecture videos.
* Considering that I switched to google colab, and the differing nature of the blogpost and the how-to guide, I think it would be best if I made the blogpost and how-to guide separate and simply linked to my how-to guide within the blogpost.
* I am currently searching for a free template, I think I will go for a more visually appealing design so as to encourage people to want to read it/stay on the post.
* Decided on [this one](https://templated.co/monochromed) because it gives a clean look.

# Thursday 4th March

Blogpost

* Finishing my blogpost tonight.
* My index page is simply a brief “About” page.
  + It begins with an attention grabbing piece of text: “Are you looking for an exciting project in Machine Learning?”
  + I found through my research for my how-to guide that successful guides employ this technique, and so I decided to replicate it.
  + I thought that even though this is a blogpost, and not a how-to guide, I do still want people to both be interested in and also to engage with/build upon my work; and I believe this is an effective way to achieve these goals.
  + Moreover, I include a brief introduction explaining the project and mentioning my how-to guide.
  + In addition to this, I also include a brief explanation of my focus area (being object detection with YOLOv5).
* The next page I have is labelled “YOLOv5”.
  + This page will be the main page for explaining the content of my project.
  + So, I am explaining how I’ve progressed through the project.
  + I am splitting the content with headings, because as I was collating it I found that a “wall” of text was hard to look at and disengaging; which is the opposite of what I am aiming for.
  + After adding these headings I feel a lot better with the visual appeal of the page; because I think that if the page is more visually appealing, attention retention will be much higher.
* I am also adding in a “Project Goal” page that will explain the goal for my project and then the larger project around it.
  + I’m explaining the AI researcher concept. I think that this will give a good incentive for readers to want to work on my project because the AI researcher is an interesting concept and they would want to contribute to that.

# Friday 5th March

Meeting with John

## Overview of the package application

* ML becomes as simple as running a program, user friendly for non technical users so they can train and deploy custom ML models.
* Zip Files
  + Zip for experiment with json file containing settings with relative path for datasets and training code.
    - Train locally- json file for experiment will allow access for training code folder:
      * Venv(complete copy of python system + libraries) folder
      * Different venv folders for different OS’s.
      * Need to work out how to deal with CUDA- makes it more complicated.
    - Train with docker GPU
    - Train on Google Colab
      * Internet access needed.
      * If you have a restrictive firewall, Jamie's plugin will navigate to colab and do local upload from venv to google drive(or other).
      * Exe will build the appropriate number of cells which plugin will fill in.
      * Fill in hyper parameters from json file.
      * End up with a training folder, with storage of python code (i.e. in a notebook.)
        + .pkl folder
        + Enough code and exes that it can copy the appropriate things that will contain all necessary files:
        + Weights, json files, exes, etc. everything needed for this training folder to be a packaged model that can be used by itself.
        + Exes:

Run as docker gpu flask app- wrap up trained model as a web server and run on an API.

Run locally as a flask web server.

Run as a command line program- similar to OpenPose.

* + - * + Modify model with ONNX (aspiration).
  + Training data zip
  + Algorithm zip
  + Output for trained models as a zip, can be used for transfer learning.
  + Exes inside the zips that can be run absent an internet connection.
  + From a user's perspective, download zips and run exes that do what they say they do.
  + Users can edit the hyperparameters, but they shouldn’t have to alter code itself.
* Likely that it will start as a command line application, then progress into a GUI for management of packages, and creating formatted datasets.