(notes for me: typescript/atelier-lakesude-light, python/agate)

Chapters: (highlight is done)

1. How do I get a project idea?
2. What can ML be used for?
3. AI Background (optional)
4. Data setup
5. How to do ML (classification)
6. How to do ML (segmentation)
7. How to do ML for non image datasets
8. LLM background
9. Make your own LLM (Fine-tune)
10. How to use RAG on LLM
11. Put your chatbot on discord
12. How to make graphs on JMP / Compare Data
13. What can GIS be used for?
14. How to get data for GIS (remote sensing)
15. How to use GIS (raster calculations)
16. How to analyze terrain data (hydrology)
17. How to use machine learning in GIS
18. How to make presentations (script)
19. How to present
20. Resources - example boards/videos(ISEF WORTHY 🙀)

21. Datasets I found to do if you can't find (basically project ideas atp)

1. How to think of a project idea

You have a research teacher for a reason. However, as someone who has had to think of multiple research ideas, usually there are two ways to approach it. Firstly, try to find anything that relates to your life, honestly if you don't already have a project idea you shouldn't be in research anyway (yet I am someone who didn't have a project idea to begin with lol). For example, if you have any thoughts or think anything is unusual that you want to research such as the effect of coffee on sleep, or the effect of airpods on hearing deficiency. Always think of potential correlations between two things that you notice (like effect of air pollution on covid)
For innovation based solutions, think about what can make your own life better that you always had trouble with. For example you notice a ton of people with glasses, then maybe look for a solution for kids that stay indoors a lot to not lose eyesight (this is actually something I was curious in doing, as loss in sight is attributed to the cornea growing too long causing nearsightedness because the body isnt used to being in such a dim environment for so long, rather than staring at screens) like a light therapy that simulates a bright outdoors environment or a potential supplement that tells your body to send hormones to prevent cornea overgrowth (or maybe tell your kids to go outside more, I probably should have followed that advice).

The second (and more cynical) way to think of an idea (this only applies to engineering or programming projects) is to not think of one, but rather base an idea around a project you make. What that means is crafting a story around some sort of project. For example, I once was participating in a kaggle competition for machine learning to try to identify between stroke types (ischemic and hemorrhagic, I didn't even know what those were at the time of development) where they provide the data and you utilize it for cash prizes. So when someone came knocking on the door for a project since they had no clue what to do, I just repurposed the code for that into a project where people from lower income countries could use this program to classify between stroke types in a quick manner with just NCCT scans. Why did I spin it like this rather than just saying it can identify between stroke types? Because honestly current solutions were better due to them having way larger datasets, so instead of saying that due to my low amount of training data, I said that the accuracy that my model could reach with just this amount of data means that for areas with low amounts of medical data available(third world countries) this model could still achieve comparable accuracy against currently used models that utilize millions of images in their training dataset. Now why did I talk about

NCCT scans rather than just CT scans? Because my code was based off of non-contrast CT scans, which are basically the most vanilla version of CT scans. And since current solutions utilize CT scans that involve dyes and advanced medical procedure (of course they also had a NCCT scan option, but it's convenient to leave that out of the story) this further cemented how this code could be used in poorer areas that may not have the resources or the personnel available to complete these more advanced versions of CT scans.

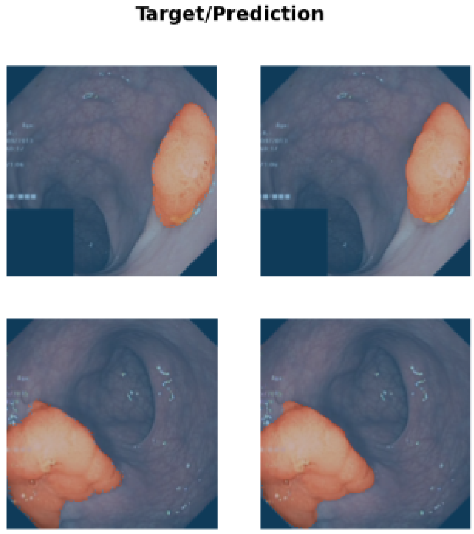
To apply this cynical thinking on research-based projects look at peer research projects for inspiration and try to find any sort of gap in knowledge and exploit it. Also preferably make sure that the issue is very impactful on our lives. For example the reason I chose soil erosion for my project topic was actually due to seeing a similar project in summer research at brentwood. Someone was looking at the effect of xanthan gum as a potential solution for soil erosion, and so what I did was see if soil erosion was a big issue. It surprisingly was a bigger issue than I originally thought (We lose all topsoil within 60 years, topsoil supports 95% of food supply and 50% of biodiversity and US farmers lose 9 billion annually importing soil) and then I immediately looked into the cost of xanthan gum and compared it to other potential solutions for erosion. How I found other potential solutions is that I looked at how construction companies solidify the ground, and then I found the ones that wouldn't be damaging to the environment and crops (which were organic tackifiers and rigid inclusions, organic polymers, etc) and chose the ones that were cheaper to use than xanthan gum which were rigid inclusions and tackifiers, and then I researched the effect of THOSE solutions on soil erosion. Technically it's not stealing a project, but using another's research as a stepping stool for your own. (By the way, this led me to convert my project into a more programming based erosion forecasting application, since research ain't my thing I prefer creating stuff).

2. What can ML be used for?

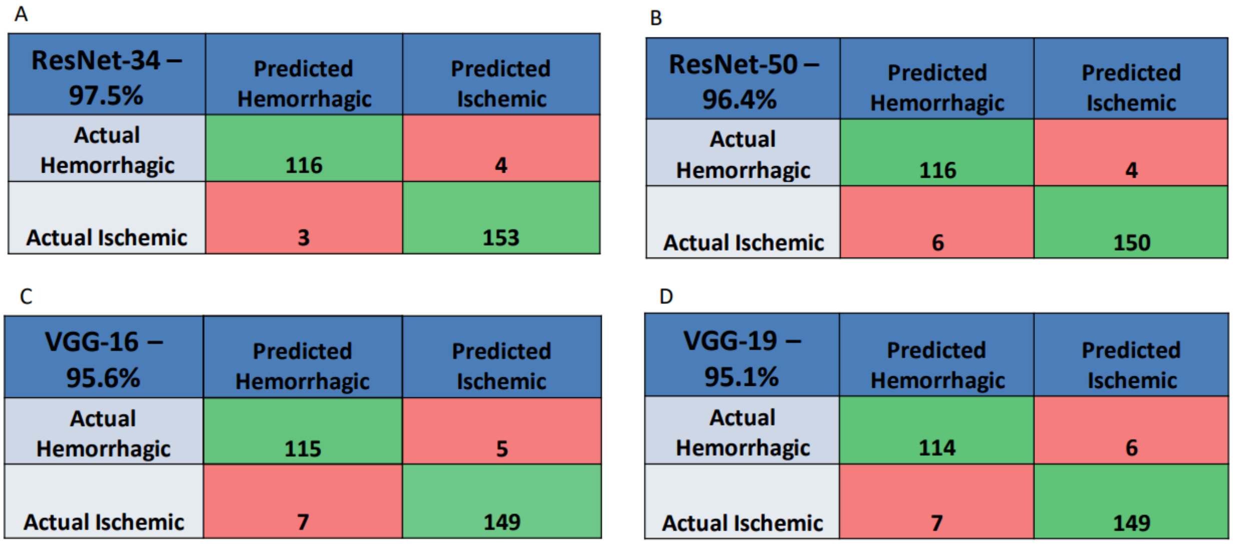
A lot of things. But firstly, what is Machine Learning (ML)? Machine Learning is, in layman's terms, an algorithm that uses data to perform a specific task to human-like capacity. Here, I'll provide the code to perform various ML tasks, so the data that you use is what matters. **The two ML codes I will be providing will do segmentation and identification. Identification will be a ML code that can identify between different things (for example a ML model that can differentiate between cats and dogs, or more scientifically, ischemic and hemorrhagic strokes from NCCT scans), whilst segmentation will be to "mask" or draw over where stuff is happening (like**

identifying where a duck is in the image, then drawing over the space is in the duck, or more scientifically, drawing a mask over where gastrointestinal polyps are to help doctors locate and identify them easily). Machine learning seems hard, but python libraries and the code that I'll be offering here makes the whole thing surprisingly easy, and can be applied to almost every project (or be used to create projects even if you really can't think of anything) The basic functions that this code will be able to accomplish are, image segmentation (image 1), confusion matrix (image 2), and receiving operating characteristic(ROC) curves (image 3-this is for correlation purposes).

Firstly, you will need some sort of coding notebook. This is because practically all AI libraries like pytorch or tensorflow utilize CUDA software, which only is available on nVidia graphics cards. In some cases (like mine) not everyone has those (AMD or MAC) and thus needs cloud processors. This can be done with google collab (although you would have to buy collab pro to do anything serious) however I much prefer kaggle, so firstly make a kaggle account (you can do so for free easily with google account or just email)



(Image 1-target on left, ML predicted image segmentation on right)



A

ResNet-34 – 97.5%	Predicted Hemorrhagic	Predicted Ischemic
Actual Hemorrhagic	116	4
Actual Ischemic	3	153

B

ResNet-50 – 96.4%	Predicted Hemorrhagic	Predicted Ischemic
Actual Hemorrhagic	116	4
Actual Ischemic	6	150

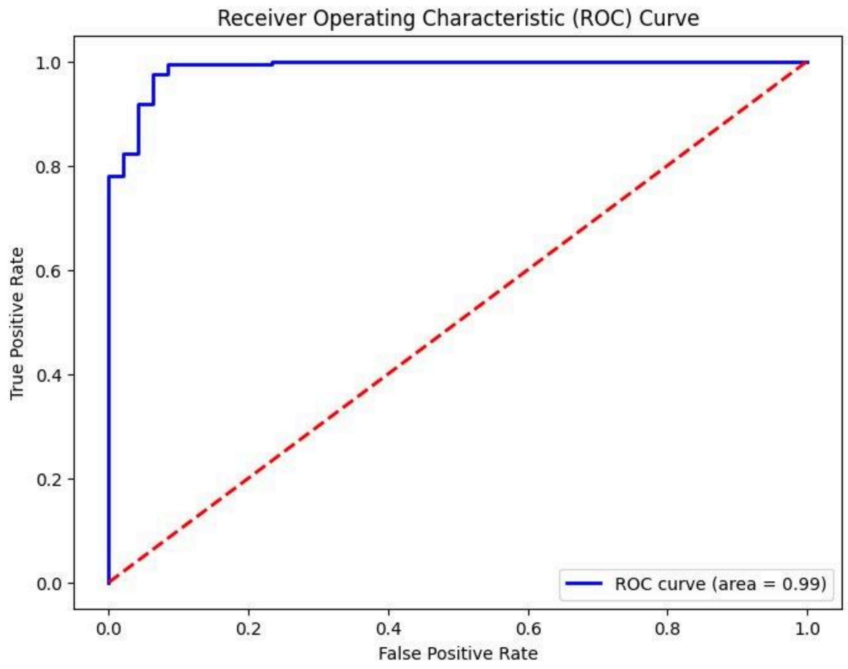
C

VGG-16 – 95.6%	Predicted Hemorrhagic	Predicted Ischemic
Actual Hemorrhagic	115	5
Actual Ischemic	7	149

D

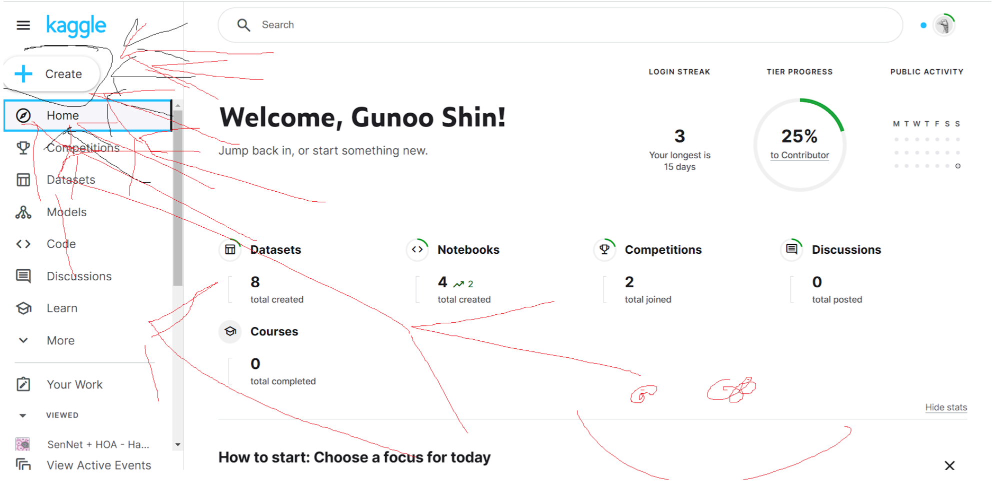
VGG-19 – 95.1%	Predicted Hemorrhagic	Predicted Ischemic
Actual Hemorrhagic	114	6
Actual Ischemic	7	149

(Image 2- confusion matrix)



(Image 3- ROC curve: basically higher area or furthest distance that roc curve is away from the dotted red line is best)

You are going to want to create a new notebook (image 4) and then make sure that internet and gpu p100 are both on (code will not work without these) (image 5).



Welcome, Gunoo Shin!

Jump back in, or start something new.

3 Your longest is 15 days

25% to Contributor

Datasets 8 total created

Notebooks 4 2 total created

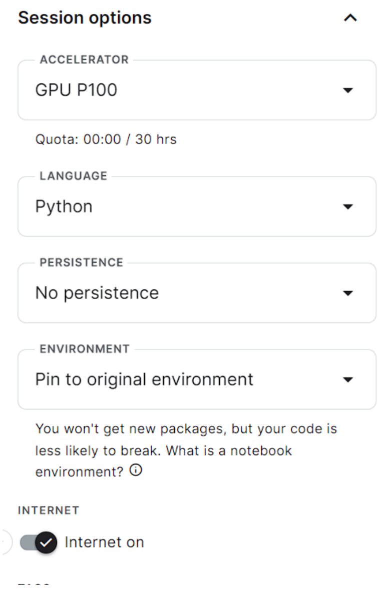
Competitions 2 total joined

Discussions 0 total posted

Courses 0 total completed

How to start: Choose a focus for today

(image 4 - create a kaggle notebook once signed in)



Session options

ACCELERATOR: GPU P100

Quota: 00:00 / 30 hrs

LANGUAGE: Python

PERSISTENCE: No persistence

ENVIRONMENT: Pin to original environment

You won't get new packages, but your code is less likely to break. What is a notebook environment?

INTERNET: Internet on

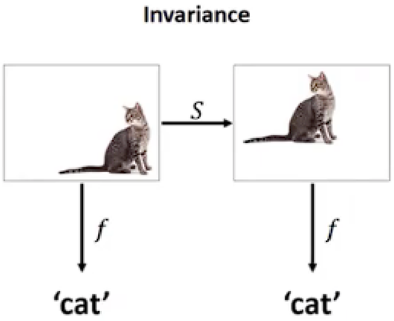
(image 5 - important specs for kaggle notebook once created)

3. Super nerdy ML background (not necessary but here if you’re asked questions or want to beef up presentation)

I am using convoluted neural network (CNN) for these examples that regard image comparison such as resnet-34 (if what you are comparing is words or something [for maybe a psychology study] either look into fine-tuning LLMs like mistral7b or just use a ANN from fastai, although I do not have that much experience in these).

The benefit of deep learning (with CNN's) is that it outperforms other conventional methods for image classification (Gautam and Raman, 2021).
Decision to Implement a CNN:

- Hierarchical Feature Learning - although this feature depends heavily on what model you use, basically what this is saying (and you can assume this for the CNN's used here) is that the model recognizes features in the data by simplest to highest complexity (resnet-34 does this and the number following resnet is just the number of layers in the model)
- Translation Invariance - This means that the position of the data does not matter, rather the pattern of it. This is particularly important for image classification so the ML model can recognize specific patterns of pixels more efficiently.



(Invariance is where the data is unaffected by position)

- Parameter Efficiency - This is for when you are fine-tuning (training) the model. The main idea is that you freeze the parameters of a pre-trained model, add some new parameters, and fine-tune the new parameters on a new (small) training dataset. This is done with U-net architecture for resnet and the other CNN models I mention.

Now wtf is U-net and resnet and whatever. U-Net architecture(image 6) relates to PEFT (parameter efficiency fine tuning) which makes the process of training a model for a specific task alot more efficient, which is important due the highly mechanically depending aspect of this (You need a ton of computing power for training a model, which is why current LLM development is just a hardware arms race for GPU's, which is why nvidias profit has exploded). U-Net is specifically for image segmentation, and relates to freezing and unfreezing pixels which keeps the image from distorting (quality control) and squeezes the most training out of the least data (in other words it's efficient). Now Resnet refers to residual neural networks, or in other words it's a pre-made model that we are fine-tuning specifically for our purposes. I usually use resnet-34, however there are other versions of resnet like resnet-50 or other models like VGG, but I use resnet-34 for sample sizes of 1000-3000 images usually because I see that it gives the best results (you would use resnet-50 or VGG for way larger datasets like 10k+ images, but you would rarely see those publicly available and you would need alot of computing power to handle those anyways). By the way if you ever want to state why you used resnet-34, I already did the tests for this claim so you can add to the presentation(image 7-8) if you want lol :).

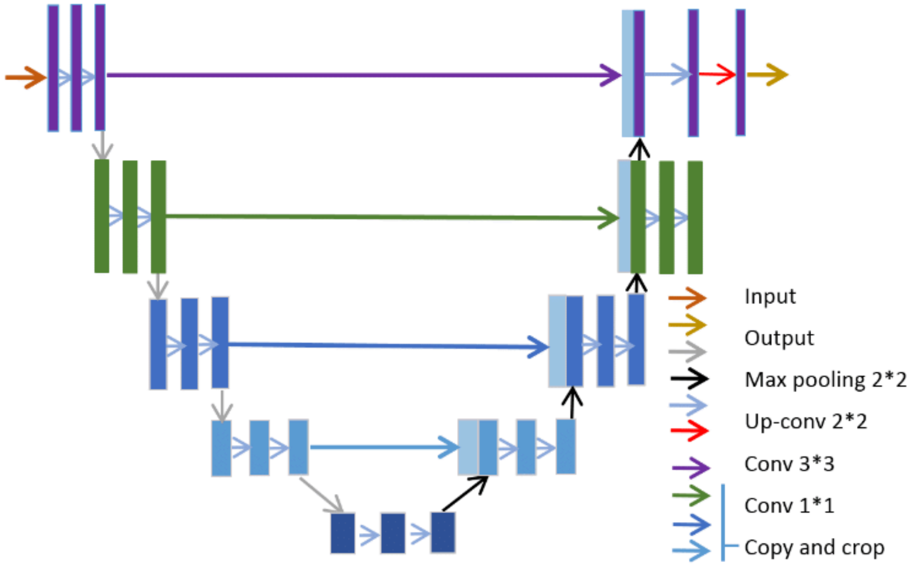
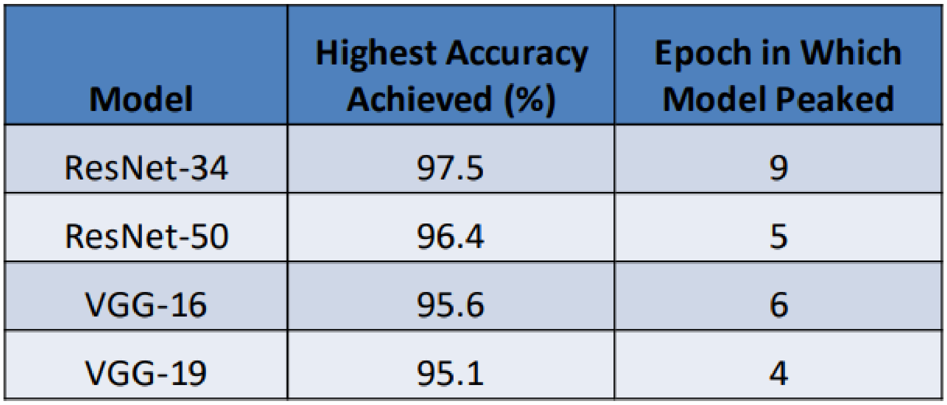
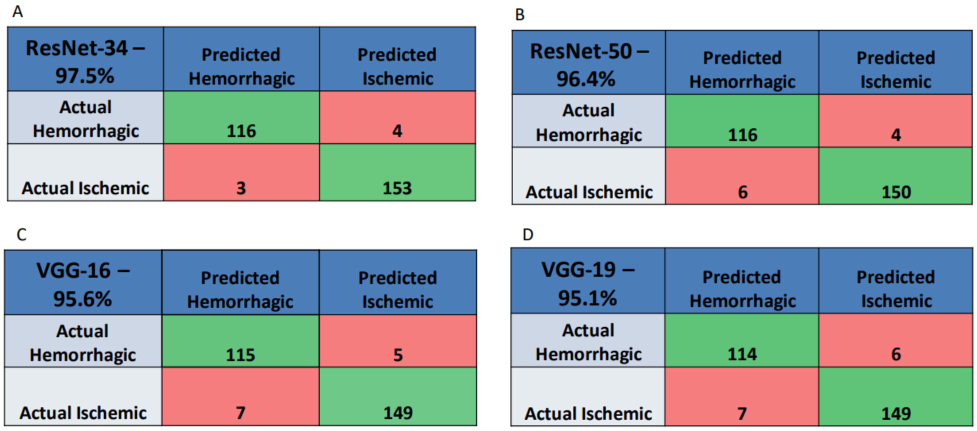


Image 6 (U-net architecture just search it up if you are really curious)



Model	Highest Accuracy Achieved (%)	Epoch in Which Model Peaked
ResNet-34	97.5	9
ResNet-50	96.4	5
VGG-16	95.6	6
VGG-19	95.1	4



A

ResNet-34 – 97.5%	Predicted Hemorrhagic	Predicted Ischemic
Actual Hemorrhagic	116	4
Actual Ischemic	3	153

B

ResNet-50 – 96.4%	Predicted Hemorrhagic	Predicted Ischemic
Actual Hemorrhagic	116	4
Actual Ischemic	6	150

C

VGG-16 – 95.6%	Predicted Hemorrhagic	Predicted Ischemic
Actual Hemorrhagic	115	5
Actual Ischemic	7	149

D

VGG-19 – 95.1%	Predicted Hemorrhagic	Predicted Ischemic
Actual Hemorrhagic	114	6
Actual Ischemic	7	149

Images 7/8 (My results for various CNN's on image classification [NOT SEGMENTATION])

Now other important stuff to know is transfer learning and data preprocessing stuff. Data pre-processing is basically getting all data to fit within a certain resolution (like 460x460 pixels) and also techniques to get the most out of a certain amount of data. These are called image augmentations, where you purposefully tinker with the image to artificially produce more training data out of a certain amount of data. For example, an image of a cat can be moved around, flipped, and cropped to produce 8 more images from an original 1 (image 9).

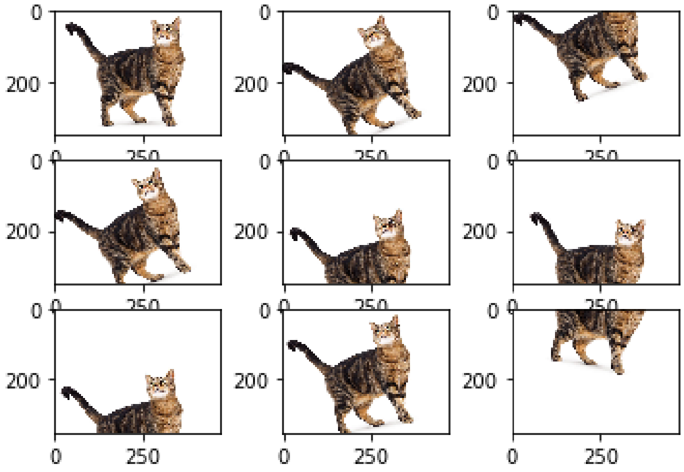


Image 9 (image augmentation example)

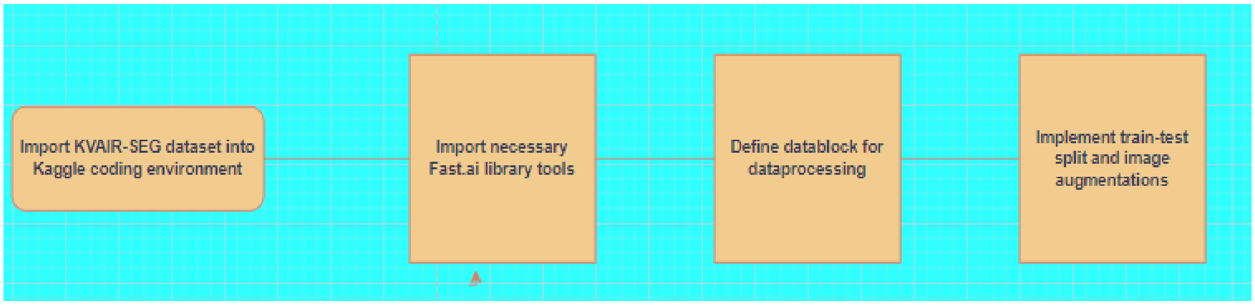
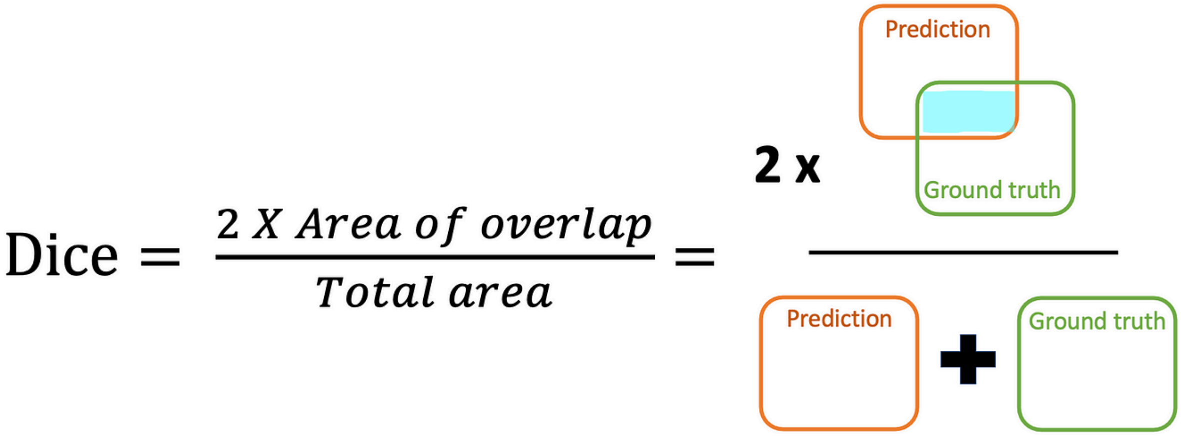

Import KVAIR-SEG dataset into Kaggle coding environment
Import necessary Fast.ai library tools
Define datablock for dataprocessing
Implement train-test split and image augmentations

Image 10 (workflow for data pre-processing)

Transfer learning is basically like double frying fried chicken, one fry is alright but two fries is a lot better. You are basically taking a pre-trained model (that's already been trained on a usually larger more generalized dataset that's still somewhat relevant to your cause) and then pre-train that model on a smaller data-set that's specific to your purposes. To use a scientific example, training a model on Imagenet dataset (which is very large) for general image classification then training that trained model on a specific batch of CT scans for stroke classification will generate way better accuracy than just getting a basic model and training it on your small batch of data. The first step of this though resnet-34 already does for you though, since resnet-34 is pre-trained on imagenet already, so you're doing transfer learning to begin with. In other words, you're doing it and can claim legitimate credit for doing it without actually really doing it. Lastly, to obtain accuracy for **image segmentation** (not identification, for identification it's a very basic process of just counting how many right images there are to how many wrong images) we use the mean dice metric, which is basically overlapping the original mask to the ML generated mask and comparing how many pixels overlap versus the pixels that do not overlap. (image below I am not making this a numbered image because I do not want to move all the other image numbers)



WHAT IS A EPOCH

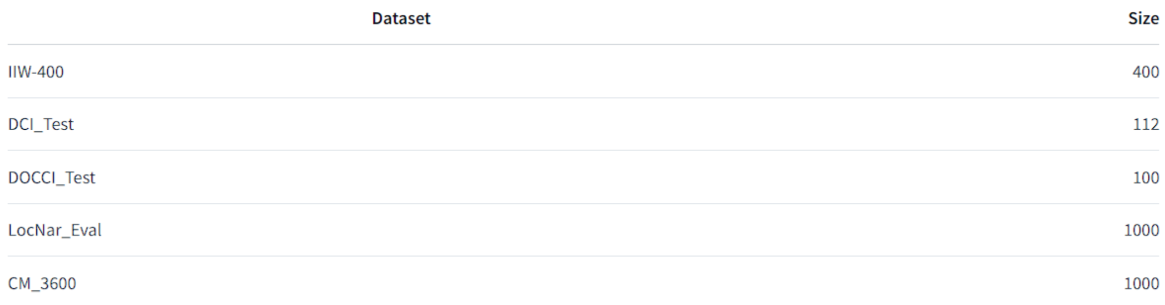
Epoch is just a full turn that the model trains through the dataset (it completely goes through it). Usually you want the epoch to be like 5-10 turns because after a while the accuracy will plateau with your dataset, especially when it's not that large. I'll specify the epochs in the code, but usually if you see a number in a fine_tune or any function starting with “fit” the first number will usually be the amount of epochs it runs (whole number). Example below.

```
learn.fit_one_cycle(10, 6e-3) #the first number is # of epochs
learn.fine_tune(5) #the number represents how many epochs you fine-tune
```

4. Data setup

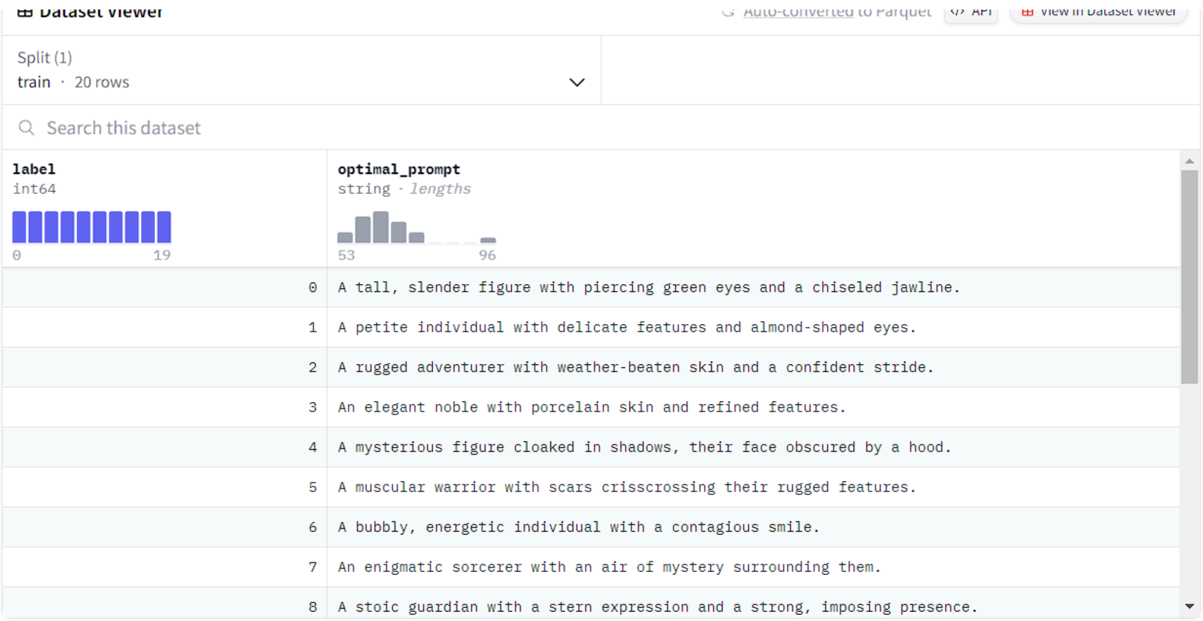
Before introducing the code or anything, you need a basic understanding of how ML works. In simple terms, ML learns via question and answer, so by continuously asking a model a question and then right after correcting/reaffirming its answer, it gets better and better at its task depending on the amount of data you feed it. However, this data must firstly be formatted in a way where it is “labeled”, or basically the “answer” is attached to all data points. For example, if you were to train a ML model to do math, all equations in your dataset (like 1+1, 2+2, etc) should have a label with its corresponding answer, so that the ML model can either learn from its mistakes or confirm its doing something right. In order to confirm you have “labeled” data (almost all datasets you find on kaggle/hugging face will be by the way, so this is a minor concern), make sure that the data is organized into proper files (to go back on my previous analogy: all equations that sum up to 7 would be in one file, while all equations that sum up to 5 in another, etc).

After this, the data should be separated further into “train” and “test” files (which will have the same organization of data within, if “train” has 5 and 7 answer files, so will “test”). The reason behind this is that a portion of the data will be used to train the model, whilst the other will be to confirm the accuracy of the model. The data used for both must be **organized** the same way, however the actual data must be **different.** If you were to label and separate your own home-made data, a good rule of thumb to follow would be 80% of data for training and the remaining 20% for testing. If the dataset you find does not have a “train” and “test”, either the naming is weird (image 11 - you will need to adjust accordingly later so keep this in mind), or the author is stupid (the dataset is probably for prompt engineering lol, if you are curious about this just email me - image 12)



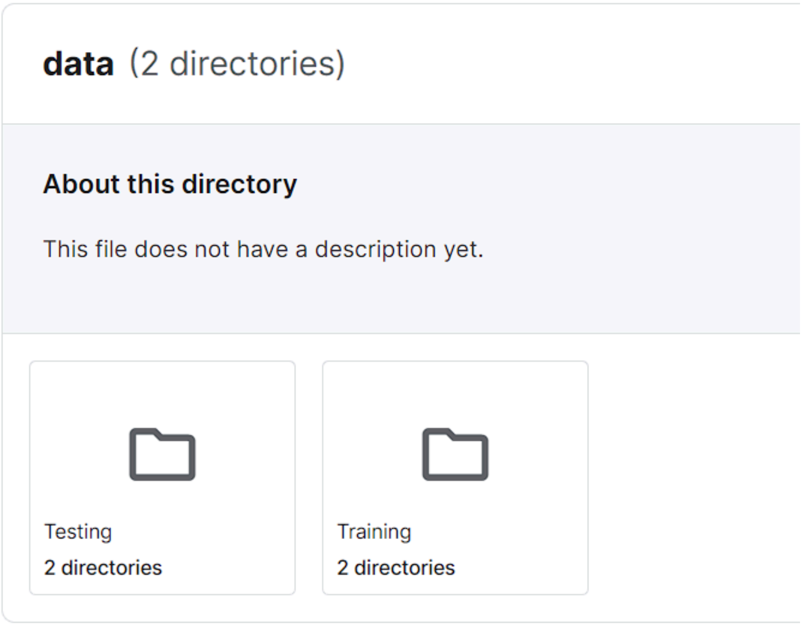
Dataset	Size
IIW-400	400
DCI_Test	112
DOCCI_Test	100
LocNar_Eval	1000
CM_3600	1000

(image 11 - weird names for test/train)



label int64	optimal_prompt string · lengths
0	A tall, slender figure with piercing green eyes and a chiseled jawline.
1	A petite individual with delicate features and almond-shaped eyes.
2	A rugged adventurer with weather-beaten skin and a confident stride.
3	An elegant noble with porcelain skin and refined features.
4	A mysterious figure cloaked in shadows, their face obscured by a hood.
5	A muscular warrior with scars crisscrossing their rugged features.
6	A bubbly, energetic individual with a contagious smile.
7	An enigmatic sorcerer with an air of mystery surrounding them.
8	A stoic guardian with a stern expression and a strong, imposing presence.

(image 12 - why is there no train/test ?????)



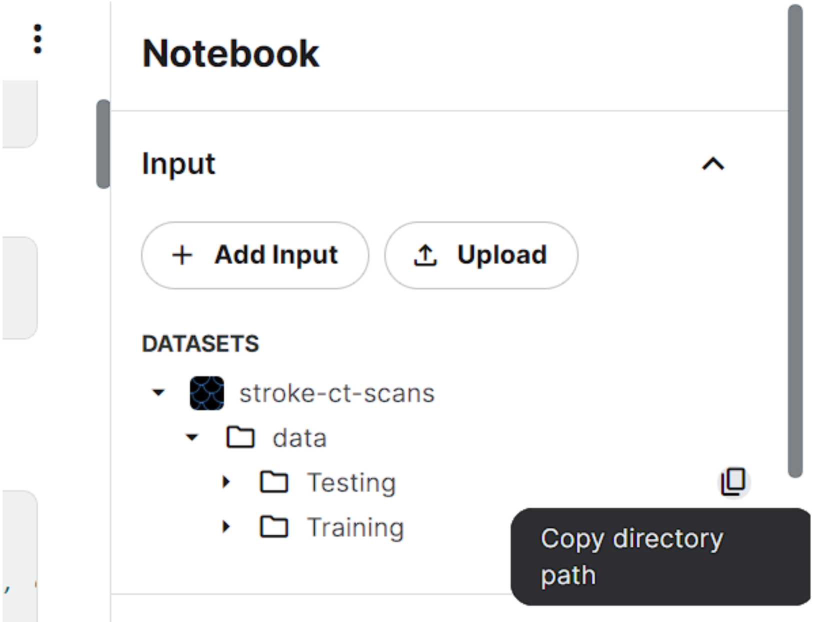
data (2 directories)

About this directory

This file does not have a description yet.

(image 13 - this is a scenario where test and train are there, but the files aren't named "test" and "train", in this case just be mindful of the path when you access these files via pandas/os later)

In order to add data into the kaggle notebook, download the dataset (kaggle, hugging face, github) as csv and upload it here (image 14) where it says "upload". If the dataset is from kaggle, click where it says add input and simply search up the dataset. Then you want to find the training and testing folders and copy the path to it by clicking “copy reference path”



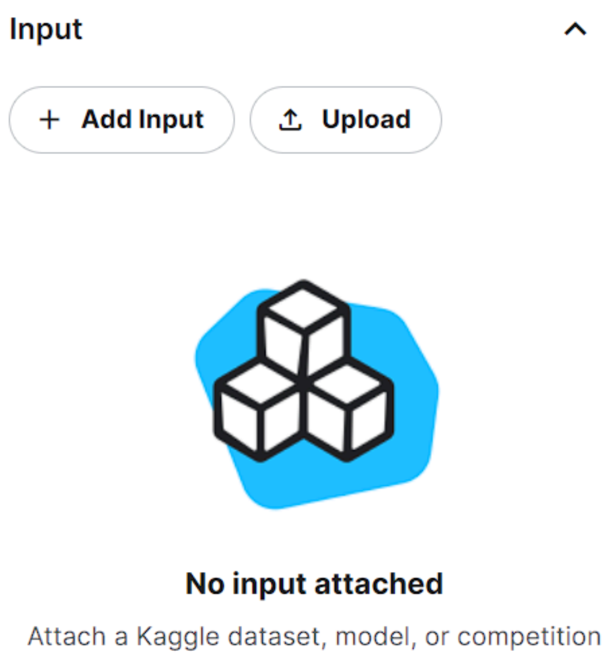


Image 14 (upload your dataset here in kaggle, it's on right hand side of notebook)

5. How to write the actual ML model (this is for a ML algorithm that can predict and identify stuff, not image segmentation)

Now this is where I actually tell you how to do this. This skill can be applicable in pretty much every aspect of science (My project was environmental engineering and this contributed, and two other projects that I gave to undisclosed individuals (one of which somehow got to ISEF for TMED lol) were almost entirely based around this while being in biomedical and translational medical science. So firstly you are going to want to install the necessary libraries:

Fastai - this basically does the heavy lifting for ML purposes and has many models compatible with it (SUPER IMPORTANT)

Os (THIS IS FOR NON-CSV DATASETS LIKE IF TRAINING IS JUST A FILE FULL OF IMAGES) - This is basically to read the data files and bring them into the code

Pandas(THIS IS FOR CSV FILES) - This is basically to read the csv(data) files and bring them into the code

Torch - this is the pytorch library, which is what fastai is based upon, and overall think of it as the windows operating software for ai basically.

```
!pip install fastai
```

```
import shutil
from fastai.basics import *
from fastai.vision.all import *
from fastai.callback.all import *
import os #this is if you are not using kaggle notebook
import pandas as pd #this is if you are using kaggle notebook
import torch
```

```

import shutil
from fastai.basics import *
from fastai.vision.all import *
from fastai.callback.all import *
import os #this is if you are not using kaggle notebook
import pandas as pd #this is if you are using kaggle notebook
import torch
```

```
torch.cuda.get_device_name(0) #this is literally just to see what
sort of graphics card you have, and whether or not your device is
compatible or not (you need NVIDIA)
```

Accessing the data withos (btw if you have any error due to path, just put r in front of it)
FOR OS

```
  path = '/kaggle/input/stroke-ct-scans/data'

train = os.path.join(path, 'Training') #or whatever your train file
is named, same with the test
test = os.path.join(path, 'Testing')
```

FOR PANDAS (DONT WORK, ONLY WORK IF U GET A CSV)

```
import pandas as pd
train = pd.read_csv("/kaggle/input/train.csv")  #example paths by the
way, everyone will have different paths
test = pd.read_csv("/kaggle/input/test.csv")
```

Model implementation

```
data = ImageDataLoaders.from_folder(train, test=test, valid_pct=0.2,
item_tfms=Resize(460), batch_tfms=[*aug_transforms(size=224),
Normalize.from_stats(*imagenet_stats)]) #this is necessary

data.show_batch() #unnecessary but will show cool stuff
```

```
valid_pct=0.2, item_tfms=Resize(460),
batch_tfms=[*aug_transforms(size=224),
Normalize.from_stats(*imagenet_stats)])
data.show_batch()
learn = vision_learner(data, models.resnet34, pretrained=True,
metrics=accuracy)
```

Model training/testing

```
learn.fine_tune(5) #the number represents how many epochs you
fine-tune
```

```
learn.freeze() #u-net type shi
learn.fit_one_cycle(10, 6e-3) #the first number is # of epochs
```

```
learn.unfreeze()
learn.fit_flat_cos(20, lr=1e-5) #the first number is # of epochs
```

```
preds, targets = learn.get_preds() #this is the testing part of it
```

Generating Receiving Operating Characteristic Curve

```
from sklearn.metrics import roc_curve, auc
fpr, tpr, thresholds = roc_curve(targets, preds[:,1])
roc_auc = auc(fpr, tpr)
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', lw=2, label='ROC curve (area =
%0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='red', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```

Getting accurate readings from folders for confusion matrix

```
# Get predictions for the validation set
preds, targets = learn.get_preds()

# Convert predicted probabilities to predicted classes
predicted_classes = preds.argmax(dim=1)

# Create dictionaries to count correctly and incorrectly classified
images for each class
correct_counts = {cls: 0 for cls in data.vocab}
```

```
incorrect_counts = {cls: 0 for cls in data.vocab}

# Compare predicted classes with true labels and count
correctly/incorrectly classified images for each class
for predicted, target in zip(predicted_classes, targets):
    if predicted == target:
        correct_counts[data.vocab[target.item()]] += 1
    else:
        incorrect_counts[data.vocab[target.item()]] += 1

print("Correctly classified images per class:")
for class_name, count in correct_counts.items():
    print(f"{class_name}: {count} correctly classified")

print("\nIncorrectly classified images per class:")
for class_name, count in incorrect_counts.items():
    print(f"{class_name}: {count} incorrectly classified")
```

6. Code for image segmentation ML model

For this I'll provide a example notebook to use if you just want to follow along, so use this (copy and edit top right corner): https://www.kaggle.com/code/gunooshin/fork-of-gastrointestinal-polyps

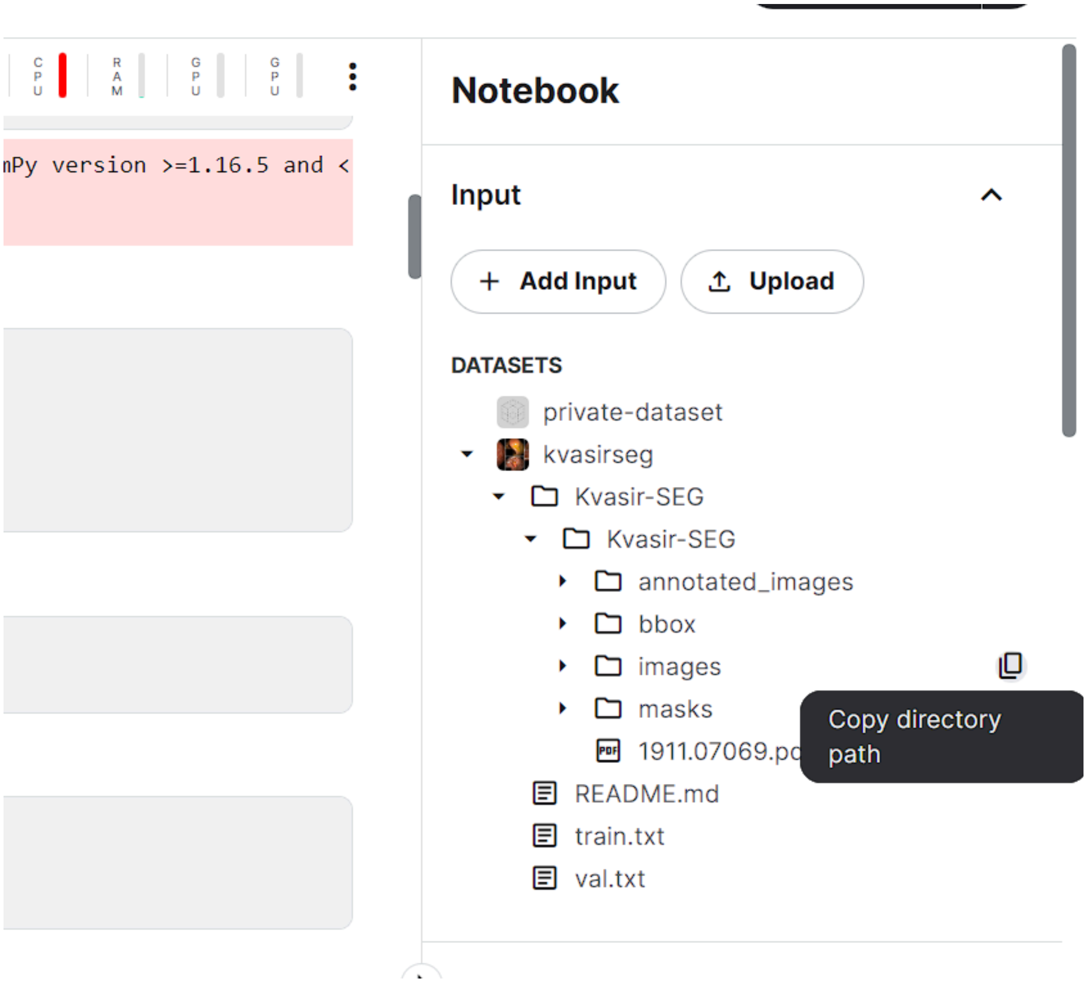
Here is a dataset (hasn't been used in comp yet) to use for yourself if you want to try it yourself with different data for practice/project:https://www.kaggle.com/competitions/blood-vessel-segmentation/data

Now if you want to just plug and chug the dataset for this code, you can quite literally do the same thing as the other model code. However, just keep in mind that rather than different strokes or labels being the ones identifying the data, the data will come in pairs of original data and the mask of the original data that was made manually. This is the train/test data format, because segmentation is training models to identify and then place masks over areas of interest, and thus keep this in mind when selecting datasets for image segmentation and make sure to adjust the paths as necessary here:

```
path_im = Path('/kaggle/input/kvair-seg/Kvasir-SEG/images') #this is
basic image directory
```

```
path_lbl = Path('/kaggle/input/kvair-seg/Kvasir-SEG/masks') #this is
labeled image directory with mask
```

SUPER IMPORTANT - HOW PUT IN DATASET:
KAGGLE - if your dataset is from kaggle like the dataset that the example notebook uses, just simply click add input on the right side of the notebook and search up the dataset. Then click the path to the images and masks and input them in by pressing “copy directory path”.



```
mPy version >=1.16.5 and <
```

SUPER IMPORTANT - HOW TO PUT IN DATASET: HUGGING FACE
I wouldn’t recommend this if you're a bum, but it's quite simple. Just download the images and masks file from the hugging face source and then click upload on the right side of kaggle notebook and upload the image and mask files. Just make sure you zip your image and mask files otherwise you won't be able to upload it to kaggle notebook

(it automatically unzips your files after download). Then just copy the path to both and plug them in.

Also usually the “test” files for image segmentations are named ‘val’ for validation, so keep this in mind.
SUPER IMPORTANT PART OF THIS BTW FOR MEAN DICE/CONFUSION MATRIX (AKA TESTING):

```
cancer = DataBlock(
    blocks=(ImageBlock, MaskBlock(codes)),
    get_items=get_image_files,
    splitter=RandomSplitter(valid_pct=0.2), #SUPER IMPORTANT VALID PCT
    get_y=get_msk,
    item_tfms=[Resize(180),FlipItem(p=0.5),RandTransform(p=1)],
    # Resize images to 800x800
    batch_tfms=[
        Normalize.from_stats(*imagenet_stats),
        IntToFloatTensor(div_mask=255)
    ]
)
```

This part of the code is super important, particularly “valid_pct=0.2”. Fastai doesn't have a way to mean_dice, so I made a way to do it. However, this means that it cannot take “test” and “train” both, so if there is a dataset for image segmentation with that sort of split, use just one folder (the one with more images or just combine them). This is because valid_pct splits the folder of data you input into train and test automatically, with that 0.2 meaning 20% is going to be used for validation/testing and the remaining 80% for training. This means just use the “train” folder or there will only be one folder providing all the data for image segmentation datasets.

7. How to make an ML model for non-image datasets?

For now I am pretty lazy but take a look at this presentation:https://drive.google.com/file/d/1EO368XeWoh8yN9B606v_sgxjfR_OnRBR/view?usp=sharing
And see if that's what you need. It basically takes a ton of coordinate points with different stats like humidity, wind speed, and fancy stuff like genesis potential index (GPI). Then they laze out and use sci-kit python library for K-nearest neighbors, Random Forests (you can use google EE smile randomforest for images by the way),

and AdaBoost (All stuff I can do btw, don't be thinking i'm some sort of bum now). Anyways, if you need me to basically recreate this persons project or you want to do this sorta thing for your data/project just hit me up preferably on discord (tragic thing to say) or email (good chance I don't respond) or mobile (strange but I definitely will respond) below:
Discord:boom120
Email:shija051@verizon.net
Mobile:(934)777-5552
I won’t require credit nor will I rat nor will I even ask for payment probably, just a lot of work to put all code here and explain.

8. LLM background

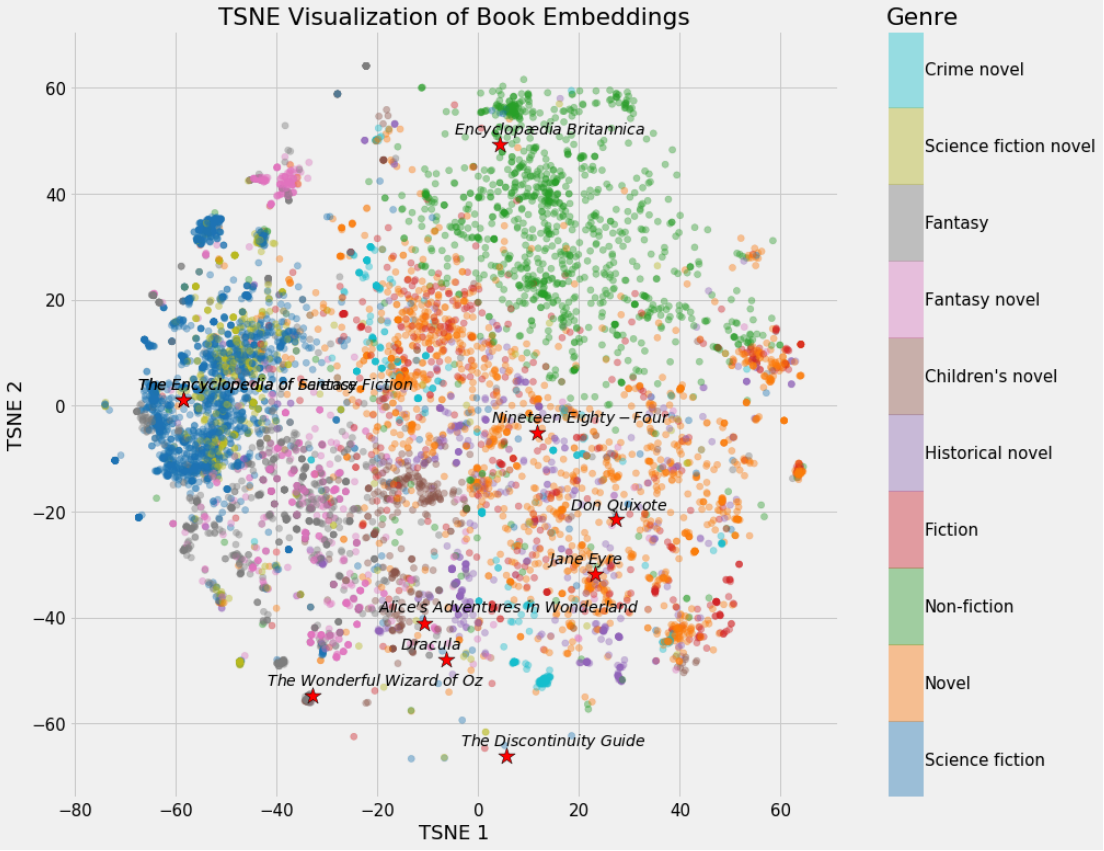
What is a LLM?
LLM stands for Large Language Model, and is a pre-trained text based model for text output. Before that though, you need to know what the difference between generative AI and LLMs, generative AI models are anything that produce output organically (or they create it) whilst LLM are just a subcategory of genai that has garnered a lot of fame recently with chatGPT, and are based around text. I’m only introducing LLMs here, but just keep in mind there are image, music, video, and countless other types of generative ai, (there's even one that translates brain waves into words: DeWave, this relates to a project idea I have that Im putting at the bottom of this that would be crazy). Although ChatGPT is the most famous of the LLMs, there are many open source and smaller LLMs out there that we can use to create something for our purposes. The two techniques that can be used to adjust LLMs for our sake are fine-tuning and retrieval augmented generation (RAG).

What is fine-tuning and why use it?
Fine-tuning a LLM is basically what we did before with the image classification models. The only difference is that before we were fine-tuning models to classify images, now you are fine-tuning for speech purposes. Fine-tuning is most useful for adjusting the personality and accuracy (of a specific topic) for a LLM, keep this in mind as in terms of being accurate with sources and references, RAG is far superior. But fine-tuning is used a lot to shape personalities, which is what I will show later. If you see in the data setup you will see an example of a dataset without test/train, but rather prompts and responses. Prompts are example inputs that the user can give, and the responses are pre-generated by either the user or a powerful model like gpt-4. The way fine-tuning works with these is that it basically gives the model just the prompt first, the model tries responding to the prompt, then the premade output is given to the model as the “correct” response the mode should have given, and after that the model attempts to answer further questions in that style. This is how the model fine-tunes on example

prompt/output datasets. This helps with personality, and this is important since you can potentially make chatbots cater your personality, a tv show character, a historical figure, or anyone you want. The hardest part is finding/making a dataset to work with.

What is RAG and why use it?

Retrieval augmented generation is essentially offering the LLM a database to refer to every time it is asked a question. For example, if you wanted to make a lawyer chatbot for texas(jayomaGPT would go hard, may work on this), you would give it a database of specifically texas law, so that the ai would reference texas law instead of going on the internet and referencing irrelevant pieces of legislature. How this works behind the scenes is with vectors. If you do not know what vectors are in math, they are an object with magnitude and direction. Why is this important for RAG? Because vectors are used to classify words in order to match up the relevant data in the database to the question that is prompted by the user. For example, words may be classified based on “furriness” and “barkiness” which would give any dog related word its own grouping whereas any feline words would get grouped up elsewhere.



Here is an example of different genres being vectorized with two vector categories (no idea what they are btw but it could be like amount of curse words, foreshadowing, symbolism, etc) which shows how only two vectors can be used to differentiate between genres. However in ML each word has thousands of vectors, each one showing a word's relationship to a certain trait. This requires a vector database, like pinecone, that has millions of premade values with vector attachments to match your input with. Then you need to vectorize the input, which you can use openai vectorizing tool for.

9. How to make your own LLM (fine-tune)

10. How to make your own LLM (RAG)

hmu

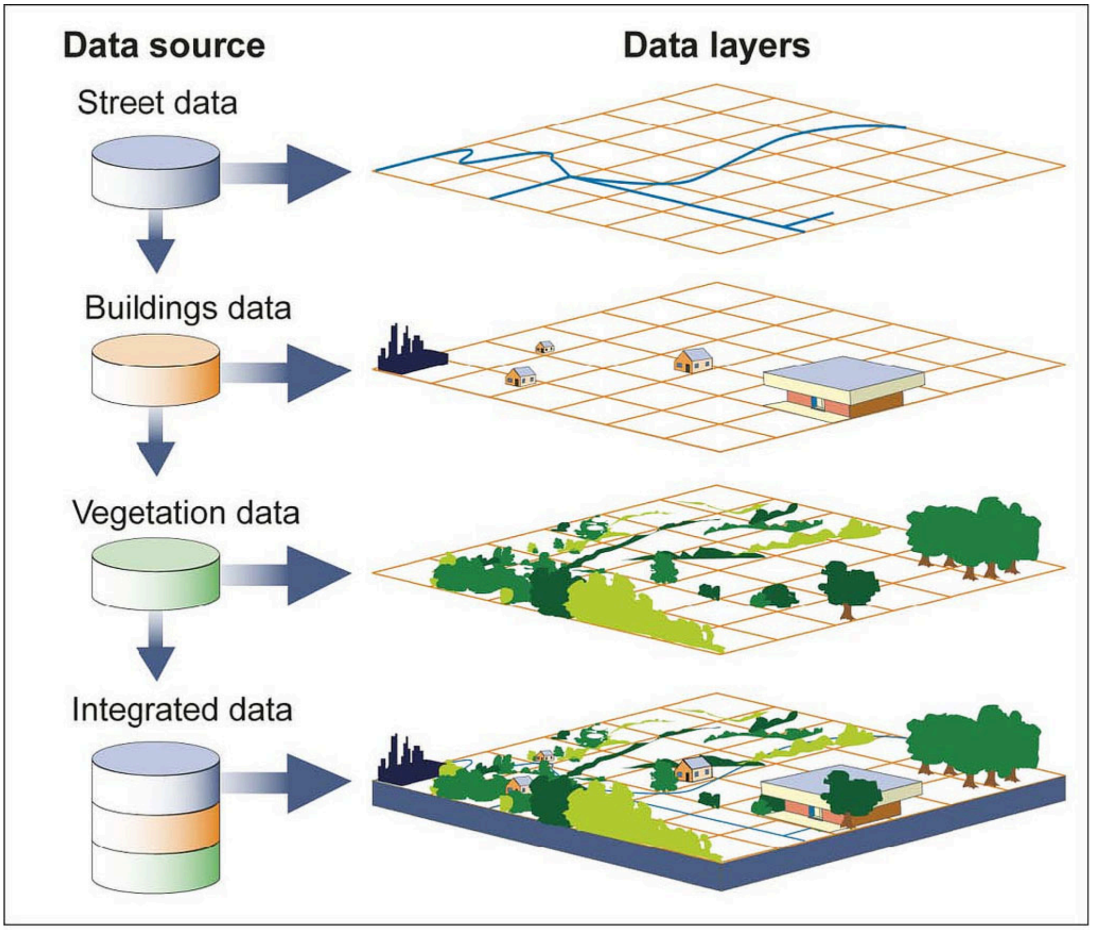
11. How to make discord chatbot

12. How to compare datasets / make graphs with JMP?

My friend Sudarshan has already done this, so I’ll refer you here:
https://docs.google.com/document/d/1_J-bKpYIGaFgq4Oa70jENivZI_IznvtvHbLFgNq8ijE/edit?usp=sharing

13. What can GIS be used for

What is GIS?
GIS stands for geographic information systems. GIS is like graphing software, sort of like JMP, but for satellite data. This means you can also process, calculate, and analyze the satellite data put in. The main softwares that you use with GIS are arcGIS and QGIS, with the latter being free to use therefore the tutorial will revolve around that.



Source: GAO.

(*Example of process of GIS)*

What can this be used for?

GIS can be used to track information including but not limited to forest coverage, forest loss, air quality, vegetation cover, plant health, and air pollution. GIS is also a great tool to use if you're trying to prove that geographic location plays a key role in a certain hypothesis. Example: If you're trying to prove that the ability to tolerate milk is based on a genetic trait found in a specific area of the world, you can create a standard data table that shows the person's lactose tolerance and their nationality, then plot where their family originates from on a map. If you see that people who can tolerate milk have a lot of ancestors from a specific part of the world, this could be great evidence to support your claim. GIS can also be used to find things such as soil, calculate how badly an area was burned, and how water flows throughout terrain.

14. How to get data for GIS (remote sensing)

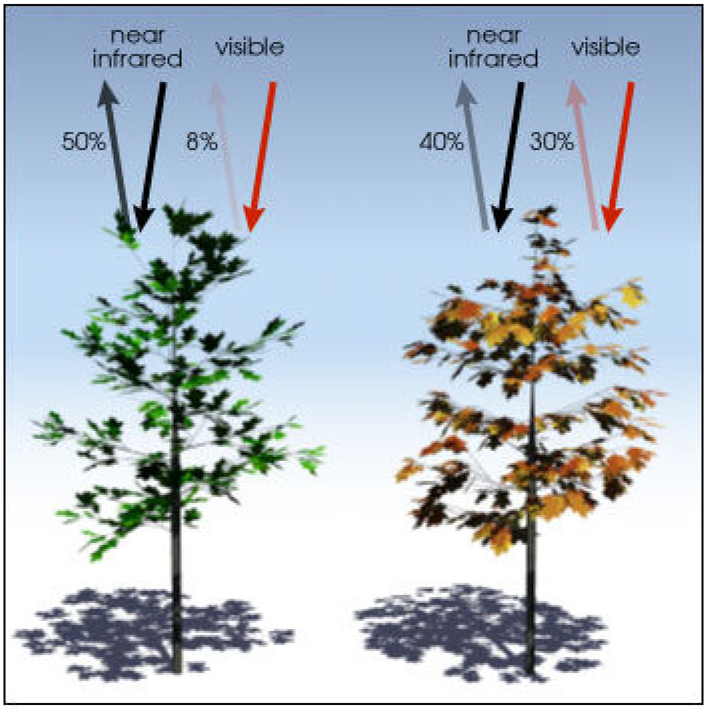
What is remote sensing?

Remote sensing is the use of satellites, planes, or drones equipped with multispectral cameras capable of taking georeferenced photographs of the earth. The camera comes with tons of different “bands” (Near Infrared, Blue, Red, Green) that act as a filter on the image. These bands can be put into different equations to highlight certain things about an image.

For example, the NDVI equation (Normalized Difference Vegetation Index) measures the density and health of plants on earth. It does this by using this equation:

$$\text{NDVI} = \frac{(\text{NIR - Red})}{(\text{NIR + Red})}$$

In this equation, you subtract the NIR (Near Infrared Band) from the Red Band, then divide it by NIR + Red. This will be demonstrated in 16. Raster Calculations.



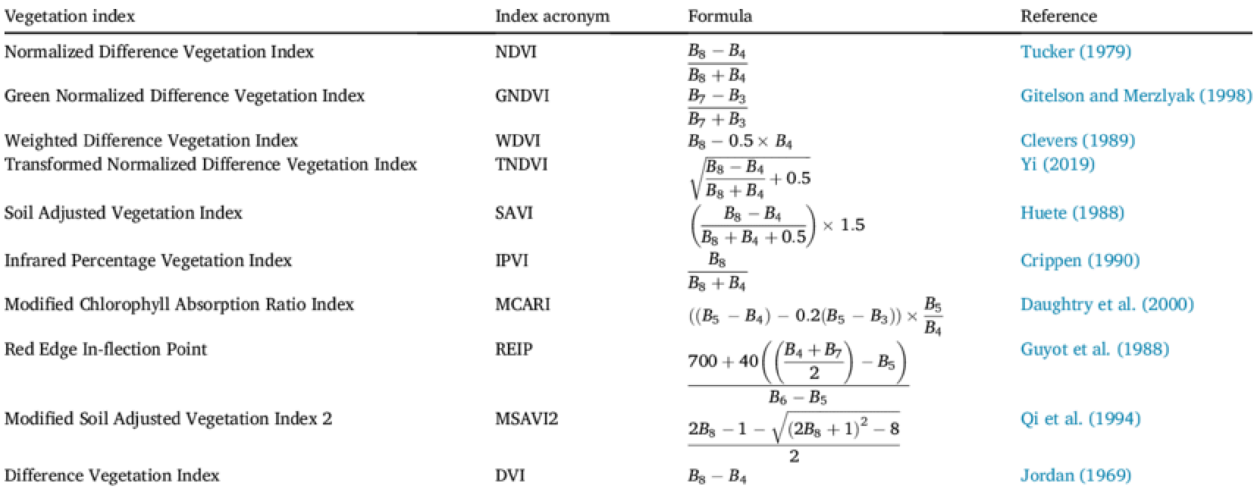
$$\frac{(0.50 - 0.08)}{(0.50 + 0.08)} = 0.72 \qquad \frac{(0.4 - 0.30)}{(0.4 + 0.30)} = 0.14$$

What data can we use for remote sensing?

For larger study areas that don't need high resolution data, you can use sentinel 2. Sentinel 2 has a spatial resolution of 10 meters by 10 meters, meaning that each pixel is 10x10 meters. Sentinel 2 is great if you need real-time up to date data. This is because sentinel 2 updates every 6 days.

If you want to sacrifice updated data for higher quality, you can use USDA's NAIP. NAIP is an agriculture program made by the USDA for taking high quality, multispectral imagery of the entire United states. It has a resolution of 0.35x0.35 meters and is great for doing very detailed remote sensing analysis. Keep in mind, the only extra spectral band NAIP has is NIR, so you should only use this if you plan on calculating NDVI.

There's a lot of other equations you can use for sentinel 2 such as



Vegetation index	Index acronym	Formula	Reference
Normalized Difference Vegetation Index	NDVI	$\frac{B_8 - B_4}{B_8 + B_4}$	Tucker (1979)
Green Normalized Difference Vegetation Index	GNDVI	$\frac{B_7 - B_3}{B_7 + B_3}$	Gitelson and Merzlyak (1998)
Weighted Difference Vegetation Index	WDVI	$B_8 - 0.5 \times B_4$	Clevers (1989)
Transformed Normalized Difference Vegetation Index	TNDVI	$\sqrt{\frac{B_8 - B_4}{B_8 + B_4} + 0.5}$	Yi (2019)
Soil Adjusted Vegetation Index	SAVI	$\left(\frac{B_8 - B_4}{B_8 + B_4 + 0.5}\right) \times 1.5$	Huete (1988)
Infrared Percentage Vegetation Index	IPVI	$\frac{B_8}{B_8 + B_4}$	Crippen (1990)
Modified Chlorophyll Absorption Ratio Index	MCARI	$((B_5 - B_4) - 0.2(B_5 - B_3)) \times \frac{B_5}{B_4}$	Daughtry et al. (2000)
Red Edge In-flection Point	REIP	$\frac{700 + 40\left(\left(\frac{B_4 + B_7}{2}\right) - B_5\right)}{B_6 - B_5}$	Guyot et al. (1988)
Modified Soil Adjusted Vegetation Index 2	MSAVI2	$\frac{2B_8 - 1 - \sqrt{(2B_8 + 1)^2 - 8}}{2}$	Qi et al. (1994)
Difference Vegetation Index	DVI	$B_8 - B_4$	Jordan (1969)

https://www.researchgate.net/figure/Calculation-formulas-of-vegetation-indices-based-on-Sentinel-2_tbl2_353126211

How can we get this data?

My friend Kaiden has already made an easy to use google colab notebook that lets you get sentinel 2 or USDA NAIP.

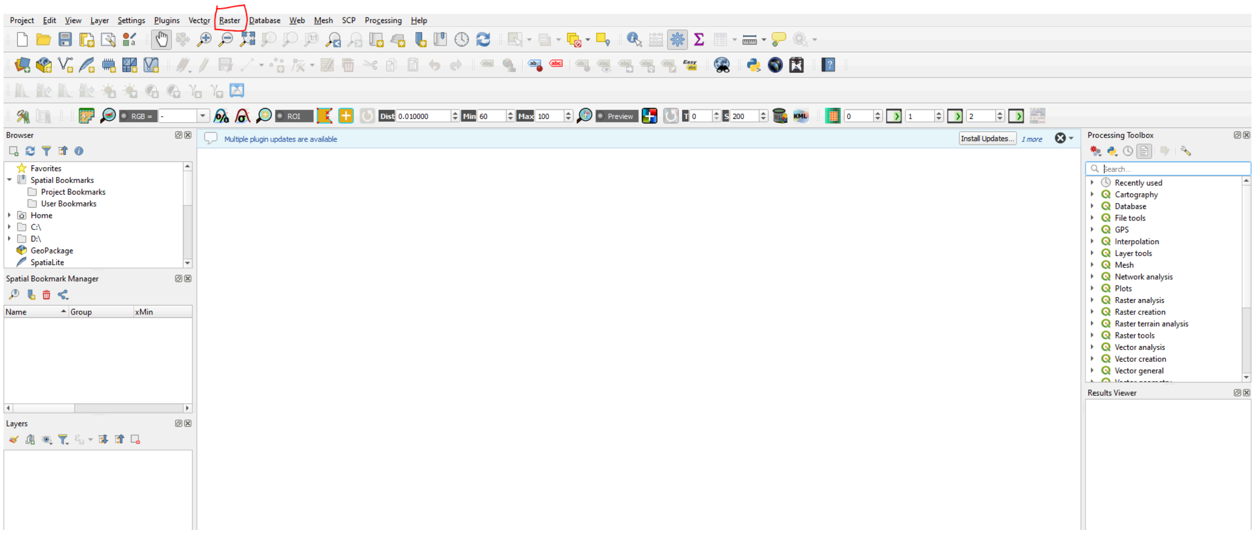
https://colab.research.google.com/drive/1VP5FOdgpP-zgqIwXK8kPdbIRjN8urj1Z?usp=sharing

15. How to use GIS (raster calculations)

For GIS, you will use QGIS since it’s the best free multi-purpose GIS software out there.

To perform a raster calculation, open QGIS.

Go to raster



Multiple plugin updates are available

Go to the drop down and click the raster calculator.

Follow this tutorial to understand the basics of how to do remote sensing equations on sentinel 2 in QGIS

16. How to analyze terrain data (hydrology)

Terrain Data

In GIS, Terrain Data is stored as 3d rasters that either contain bare earth elevation (DEMs) or terrain, trees, and vegetation (DTMs). DTMs (Digital Terrain Models) are better if you are trying to provide a precise spatial model for your area. DEMs (Digital Elevation Models) are better if your just trying to analyze things like slope, contours, or hydrology.

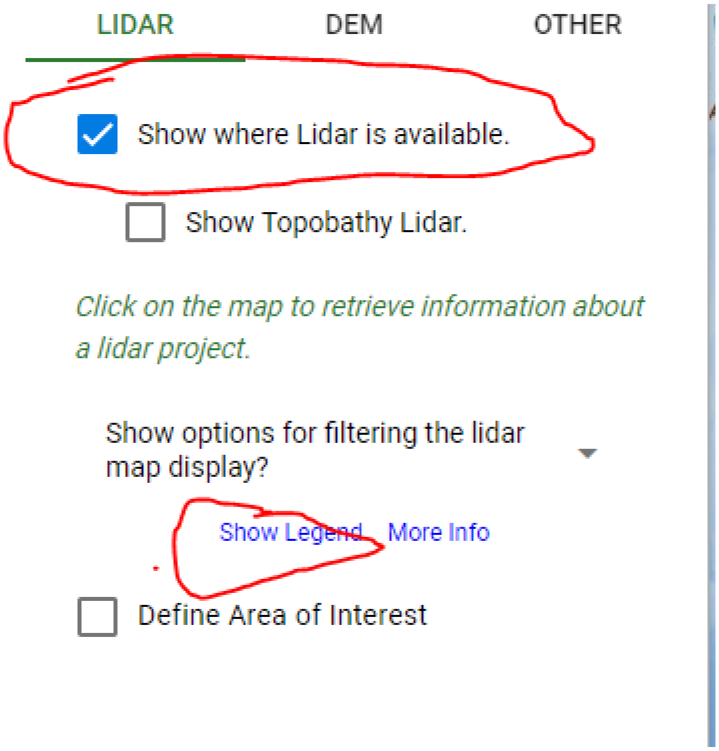
There are two main sources of elevation data: 3DEP and Lidar.

Lidar is better for DTMs because Lidar is hyper accurate, with accuracy down to the foot, ensuring the ability to view trees, plants, buildings, etc.
3DEP is better for terrain analysis due to it's focus on creating DEMS.

How do we get Lidar/3DEP?

Lidar

To get Lidar data, go to https://apps.nationalmap.gov/lidar-explorer/#/



☑ Show where Lidar is available.

□ Show Topobathy Lidar.

Click on the map to retrieve information about a lidar project.

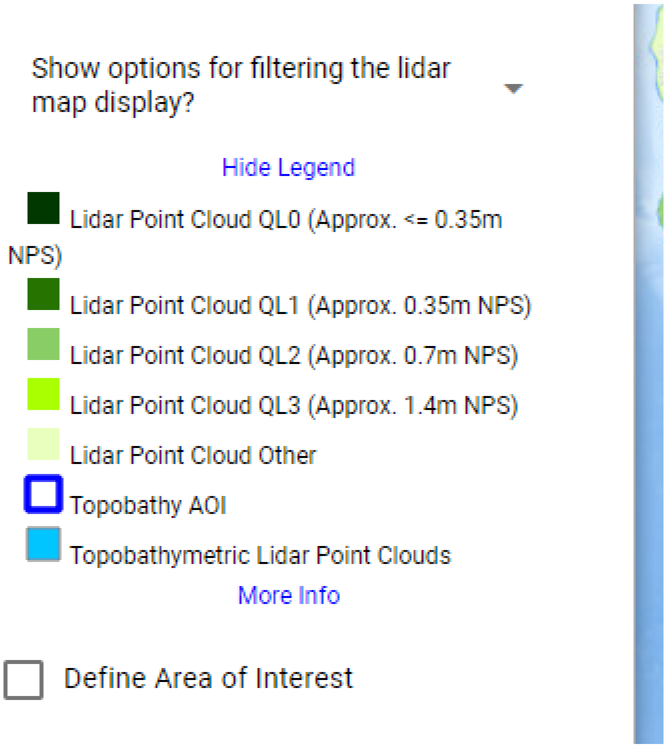
Show options for filtering the lidar map display? ▾

□ Define Area of Interest

To see where Lidar is available, select "Show where Lidar is available".

To see what the colors on the map mean, click "show legend"

Let's break these numbers down:



Show options for filtering the lidar map display? ▾

Hide Legend

🟩 Lidar Point Cloud QL0 (Approx. <= 0.35m NPS)

🟩 Lidar Point Cloud QL1 (Approx. 0.35m NPS)

🟩 Lidar Point Cloud QL2 (Approx. 0.7m NPS)

🟩 Lidar Point Cloud QL3 (Approx. 1.4m NPS)

🟩 Lidar Point Cloud Other

□ Topobathy AOI

🟦 Topobathymetric Lidar Point Clouds

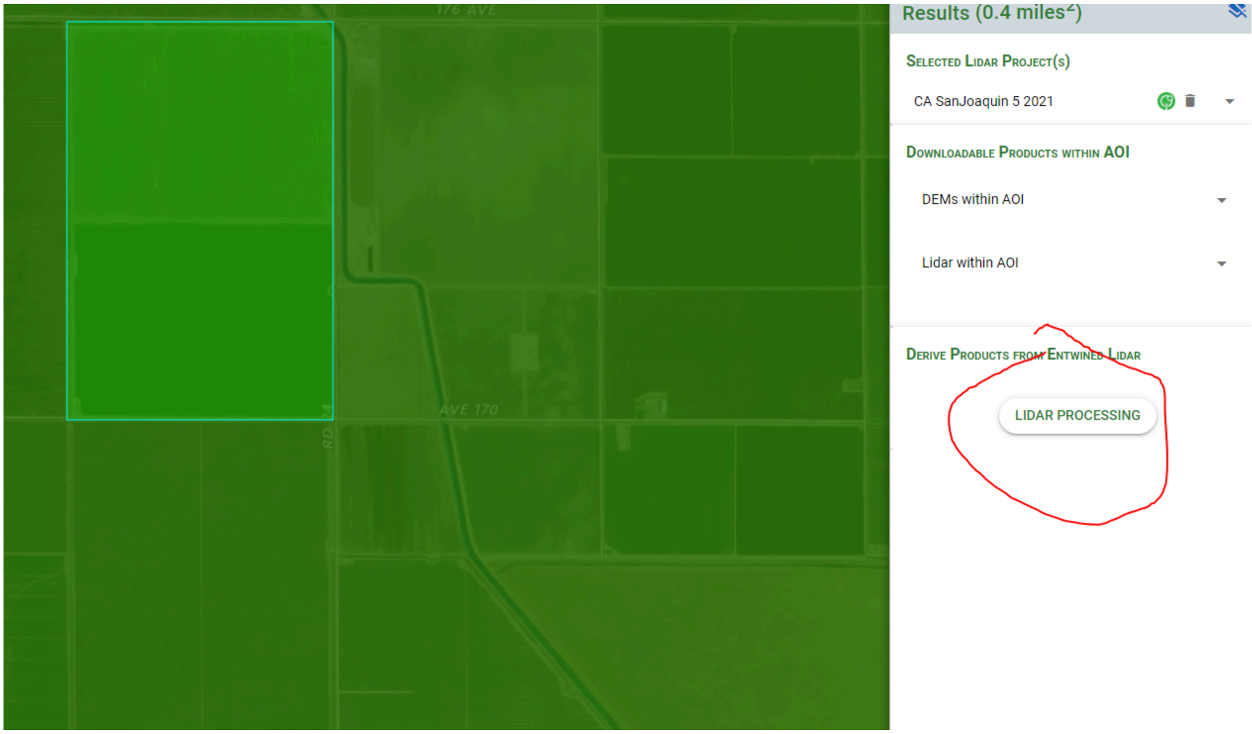
More Info

□ Define Area of Interest

The lower the number in m, the higher resolution. This number is telling you the pixel size in meters. Dark green is the highest resolution at 0.35 x 0.35 meters.

Let's select our study area. Zoom into the map, hold control and draw out an area (DO NOT MAKE IT TOO BIG, OR THE LIDAR EXTRACTION PROCESS WILL FAIL.)

Click "lidar processing"



Results (0.4 miles²)

Selected Lidar Project(s)

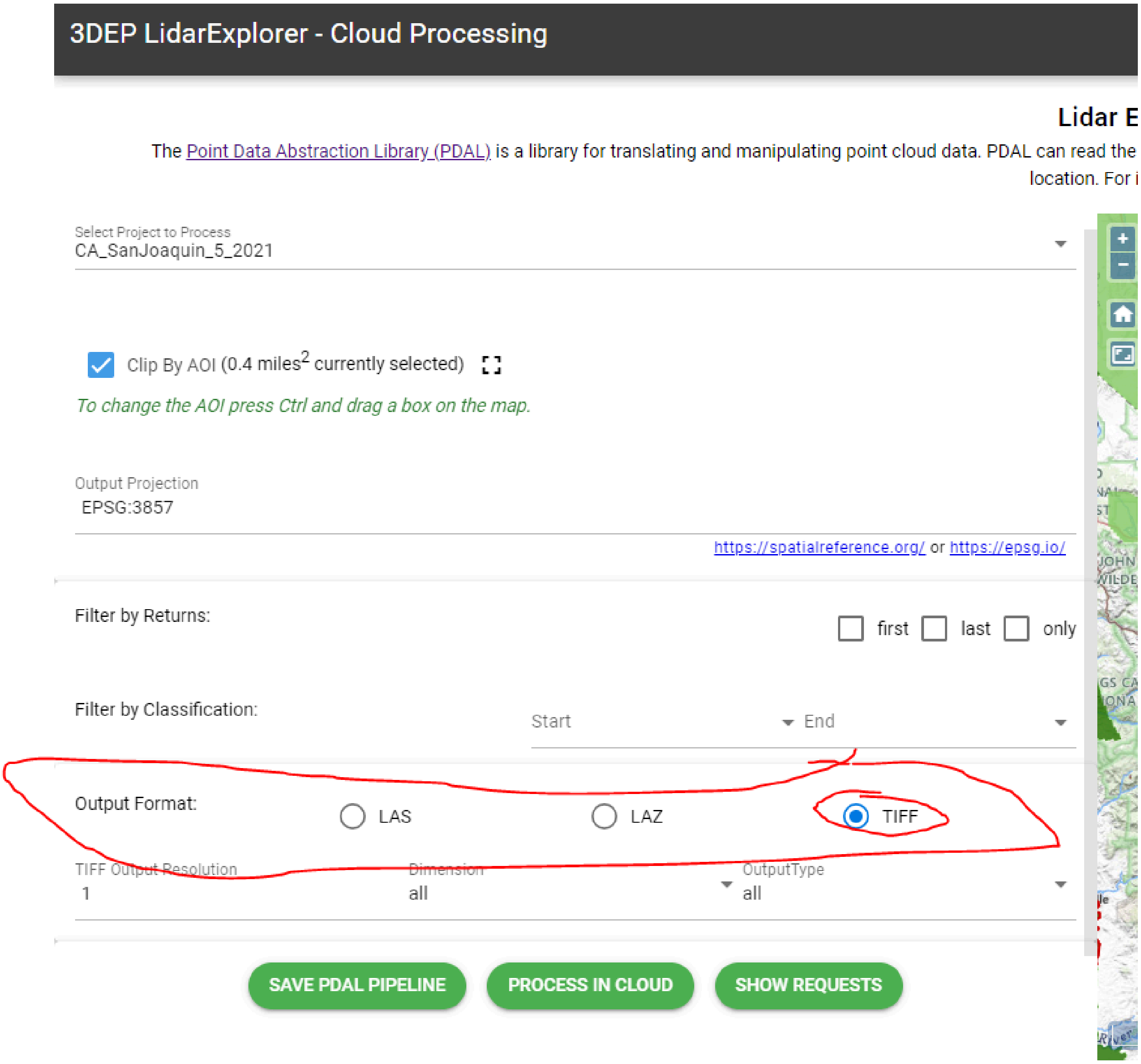
CA SanJoaquin 5 2021

Downloadable Products within AOI

DEMs within AOI

Lidar within AOI

Derive Products from Entwined Lidar



3DEP LidarExplorer - Cloud Processing

Lidar E

The Point Data Abstraction Library (PDAL) is a library for translating and manipulating point cloud data. PDAL can read the location. For i

Select Project to Process
CA_SanJoaquin_5_2021

☑ Clip By AOI (0.4 miles^2 currently selected)

To change the AOI press Ctrl and drag a box on the map.

Output Projection
EPSG:3857

https://spatialreference.org/ or https://epsg.io/

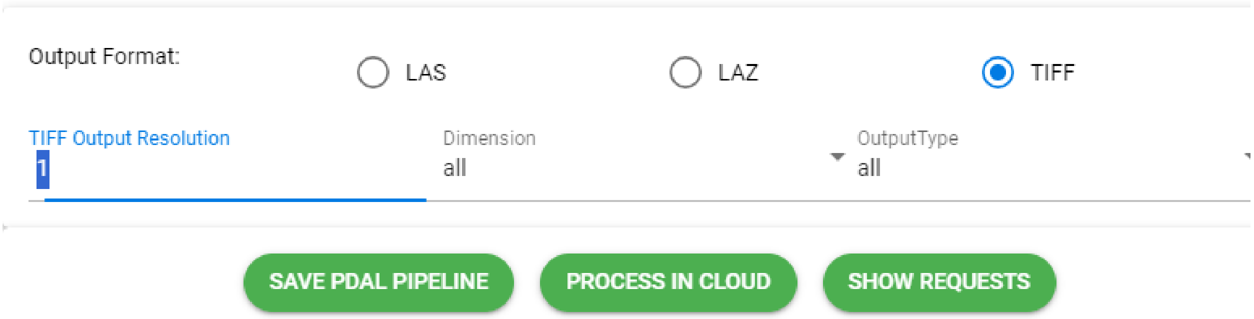
Filter by Returns: □ first □ last □ only

Filter by Classification: Start End

Output Format: ○ LAS ○ LAZ ● TIFF

TIFF Output Resolution: 1
Dimension: all
OutputType: all

Set the output format to TIFF.



Output Format: ○ LAS ○ LAZ ● TIFF

TIFF Output Resolution: 1

Dimension: all

OutputType: all

Set the output resolution to whatever number the color of your data corresponds with. Since this area is dark green, I will set it to 0.35

Click Process in Cloud

You can view your processes. Make sure to refresh every few minutes to see if it's done.

Once it's done, download it and drag and drop the geotiff file into QGIS.

Follow this tutorial to see how to model watersheds with DEMs in QGIS:

https://www.youtube.com/watch?v=c_fOhfXGsGA

In the tutorial, they will show them selecting an "outlet" point for watershed delineation.

For that, simply select the area you're trying to model water flowing and outputting into.

18. How to make a presentation? (video)

If your goal is placing well on science competitions (like I was originally when I started), this will ironically be the most important part of this all (as I found out when an undisclosed project that you cannot figure out got to ISEF by just doing this tutorial but having a good presentation). Remember that everything here is just my own personal opinion that I think works if your goal is winning, or just to make a good presentation.

Making a script:

The two things I'll emphasize here are the introduction and the conclusion, as that's really what the judge is really gonna understand and listen to. For the methods and whatnot just use buzzwords and sound as fancy and complicated as possible, which is easy if you follow this guide (rather than cnn you say convolutional neural network for example). Here I'll include an example script (THAT GOT TO ISEF BTW) from one of my projects, and outline the purpose and what went into each part. (this script is for the video but it applies the same to boards)

Introduction:

```
Hi, my name is ___ ____, and my project utilizes machine learning to
conduct image segmentation of gastrointestinal polyps for early detection
of colorectal cancer.
```

Just a basic introduction, start with your name and rather than regurgitating the project title, explain what the project accomplishes or what the project explores with what process (in easy to explain terms, as you will go in depth later)

```
Colorectal Cancer is a massive burden on our society today with it being
the fourth most common cancer in the world. With a 35 percent mortality
rate and causing 8.6 percent of all cancer deaths, it is one of the world's
most dangerous cancers. The American Cancer Society also finds that
colorectal cancer is the leading cause of cancer deaths among men under 50
and the second-leading cause of cancer death for women in the same age
group.  It is estimated that 4.1 percent of all American men and women will
be diagnosed with colorectal cancer, making it a top priority to find means
for early detection and polyp removal.
```

Next, introduce the importance of the subject. This is one of the most important parts of your presentation, as you want to emphasize the severity of the project, and will help the listener get an idea of how significant your project really is. I did this with my erosion project by blabbing about how topsoil supports 50% of earth's biodiversity and 95% of our food supply, and is estimated to be completely eroded within 60 years (which was the most generous estimate I could find for my purposes). I mean even with this "importance" paragraph I made here, we found the purpose **after** I made the actual project by connecting gastrointestinal polyps to cancerous polyps and then to colorectal cancer, which is something that gets attention and a related problem that is far more interesting than just saying "doctors can identify polyps better", so instead you say "doctors identify polyps better which can save millions from the deadliest form of cancer for men under 50". The story matters just as much as the actual project.

```
Gastrointestinal Polyps are masses of cells or tumors that form on the
inside linings of your stomach or colon. These polyps are the cause of
colorectal cancer, and are extremely common with 15-40% of American adults
developing them within their lifetimes. These polyps then have a 6-10%
chance of developing into malignant tumors, which results in colorectal
cancer.
```

After you hook their attention in with your significant part of your introduction, you lull them back to sleep with background information on your project. On a more serious note, your judges will very rarely actually know much about your subject, so explaining the subject more in detail will help them greatly. Never assume they know what you are talking about because judges will always try to look like they do. Also always use the most generous estimates (I am not saying to lie just utilize what helps)

```
The purpose of this project is to improve upon current methods of
gastrointestinal polyp detection. Studies have shown that 38.69% of
patients have at least one polyp missed during manual colonoscopy,
along with a general polyp miss rate of 17.24%. Furthermore, virtual
colonoscopies(or the other method of detecting polyps) only
identified about 65% of the patients who had polyps detected by
manual colonoscopy. Therefore, current methodologies are far too
inaccurate and unreliable and risk the safety of the patient. My
project aims to address this, using various machine learning models
to detect gastrointestinal polyps in a timely and accurate manner,
which will greatly aid the chances of colorectal cancer prevention.
```

This is where you grab their attention back again. Also the most important part of your presentation where you utilize all your previous background information on the severity of the issue and then flip it on its head by presenting an innovative solution (or if you're doing research, how you're researching something completely unknown/necessary). You also further emphasize the issue here, by discussing current solutions (or for research current findings) and their many shortcomings (cherry pick what you say of course) such as lack of effectiveness, accuracy, cost-effectiveness, and user-friendliness for engineering-based projects and small sample size, misleading data, lack of coverage, and inconclusive findings for research-based projects.

Methods:

```
A convolutional neural network or CNN is a type of neural network
```

> that specializes in different computer vision tasks such as image segmentation and classification. A CNN's convolutional layers each detect a certain feature of the desired segmentation, such as edges or colors which becomes more complex the more layers the model possesses. The rationale for using a CNN over an artificial neural network or ANN lies in its ability to utilize image datasets and CNN's being the most powerful model of AI for segmentation purposes.

The methods section is going to usually require its own background introduction, as you will need to inform the judges on what exactly you are using. For GIS you can explain how GIS is software that allows you to display and calculate geographical data from satellites. For research you usually don’t need to introduce the background of the process as much, since most judges will know laboratory technique, but if you were to use special equipment or utilize a unique data collecting technique (such as a soil runoff plot), you would need to introduce that before moving on. Remember you never want to make the judge feel completely lost, but rather slightly confused but knowing enough to be greatly impressed.

> For data processing, I chose Kaggle for my coding environment due to its availability of GPU accelerators. For my python library, fast.ai was chosen firstly for its efficient data preprocessing, which resized all my data to a consistent fit as well as image augmentation such as random rotation, cropping, and flips which allowed for an expanded dataset without the need for more data. Secondly it was used for its simplified integration with Pytorch allowing for more access and variability of different functions. Here is my code.

> For the model, UNet architecture was used for the convolutional neural network. This architecture is used to prevent image distortion by using the same process in image-to-tensor and tensor to image conversion. This is particularly important in the medical field, where accuracy is paramount to the patient's health. A freeze and unfreeze method was also used to specify training for specific layers such as specific or complex (like identifying the polyp from colon) task layers. Lastly, fine tuning transfer learning was used to minimize any further image distortion and maximize model accuracy.

Now using buzzwords and sounding fancy doesn't mean be insensible and just yap. Here in this paragraph you can clearly see that although I am using words that the judge would probably not know, I always follow up with some sort of reasoning, so that the judge is aware of the thought and reasoning put behind my processes. Remember that you don’t want the judge completely lost, just lost enough to be greatly impressed. So if you are going to use fancy terms, make sure to always follow up with reasoning.

```
After training, a peak accuracy of 96.8% was reached with the
resnet-34 model. As shown in the image to the right, the target and
prediction images are nearly identical showing the model's accuracy.
The model was also able to detect small polyps (or polyps smaller
than half a centimeter) which are normally missed by manual
inspection.
```

```
Here on the left, is another image of the results showing its
accuracy with regular-sized polyps (or half a centimeter radius or
above) And on the right is a confusion matrix for the model with the
numerical values representing the amount of pixels matching the
appropriate category of the matrix.
```

```
Here are more images of examples of the model generated predicted
polyp detection versus the target or original polyp dataset, which is
what the model should be generating at a hypothetical 100% accuracy
rate.
```

Not much here to explain except that you should always make the results relevant when they can’t speak for themselves. For example when I talk about a peak accuracy of 96.8%, that level of accuracy speaks for itself, especially when given the low accuracies that I talked about earlier in the presentation. However, when I compare the target mask dataset to the predicted mask dataset (refer to image 1), I explain that the target is what it should look like and the predicted mask is what the model generated, so that judges then can look at that and be like “wow the prediction is so close to the real thing!” rather than “wtf am I looking at”.

Conclusion:

```
In conclusion, my machine learning model will allow for great
```

```
progression in the field of colorectal cancer detection and thus
prevention. As stated prior, colorectal cancer is a massive issue
that plagues the health of Americans nationwide, and thus a model
that provides an accurate assessment of polyp detection in a timely
manner will greatly increase chances of preventing a malignant tumor
from developing. Such a model is provided in this project, and I
believe that utilization of my model will be key in prior detection
of gastrointestinal polyps.
```

Remember, judges are always tired around this time and they have just received a barrage of yap, so in a way “restate the thesis” (talk about the issue from introduction again) and then tie it back to your project and how it solves the horrifying issue that you originally introduced in a way that no one else has before. Then state with confidence that this solution is the best and has great potential to be used in the real world (whether or not it is or not, you need that level of confidence regardless for them to believe that you accomplished a great innovation). For research level projects, this is made a lot easier. Talk about how your findings have shed new light on an undiscovered area of science (again whether or not this is completely the case is irrelevant) and how your findings solve or tie up loose ends with the mystery brought up in the introduction. For example, if your research is on the effect of iPhone radiation, state how your findings can conclude that radiation can lead to increase in cancer risk when carried in pockets for example, and how that originally there was no real conclusive evidence of this before you (this is purely hypothetical but you get the point).

```
For future work, I want to implement this project in the real world. This
project would be able to advance by being integrated into an easily
accessible web app for doctors to implement during the manual colonoscopy.
While taking image data from the colonoscopy, the model can run
concurrently allowing for a quick and accurate segmentation of all polyps
in the gastrointestinal system. There is also the potential to be
integrated into education as a checking tool during simulations for medical
students. Finally further fine tuning with more complex patient data such
as gender and age would allow for the model to reach more in depth results
and conclusions such as probability of malignant tumor development and risk
factors.
```

For engineering based projects, future work should be a piece of cake. Almost ALWAYS talk about **REAL WORLD APPLICATIONS** and how you plan to and what industries/people can utilize the project that you have built. Now engineering projects have 4 real factors that make their value: User-friendliness, cost-effectiveness, accuracy, and effectiveness (for example how long it takes to work). Now not all engineering projects will hit all 4 of these factors in a satisfactory manner, so address your weakest link first before they can question you on that (or at least if they do, you will be prepared). For example, if your project is on iphone detectors for testing centers but cost-effectiveness isn't really a strong suite of your project, you can talk about how "projected" costs can potentially be cut 10 fold through industry of scale and replacing expensive parts with cheaper parts and somehow making that work.

For research based projects, future work is almost always a "what next" sort of thing (which is I guess what future work means lol). This means that now that you found out this, what else that is relevant to this can you explore. For example, going back to iPhone radiation, if you conclude that radiation can increase cancer risk, you can now see if there is a link between that and infertility, or maybe see if the effect is due to 5g rather than the iPhone itself (purely hypothetical I don't wear a tinfoil hat). You also can address potential shortcomings in your research, such as a smaller sample size or potential corruptors in your dataset. But remember, you are an aspiring high school student, so use that and spin the story as one where your shortcomings in your research is purely on your lack of resources as an independent high school researcher, and that in the future you hope to have the funds/resources necessary that are currently holding your wonderful research back.

19. How to present

Important things to consider:
LISEF round 1: 7-9 minutes to present / 3 judges (they have a lot of kids so they are very strict with timing, make sure your presentation can be comfortably said in 6-6:30 minutes)

LISEF round 2: 9-13 minutes to present / 5 judges (More lax now that 75% of kids have been cut, but they are still relatively strict on timing, but now you have more time for discussion and questions)

NYSSEF round 1: 7 minute video

NYSSEF round 2: 11-15 minutes to present / 3-5 judges (the most lax experience, which is a double edged sword to some. Most of the presentation will be discussion and questioning, and the day is also really fun since you can explore the hall of science. Some judges even get to 30 minutes)

Introduction:

I always like to ask the judge before I start on what scientific background they hail from. This makes it so that the presentation has a more personal feel to it and that the judge feels more comfortable now, it also helps you to gauge their comfort with the topic you're presenting and allows you to elaborate/skip through areas of your presentation appropriately. Overall I have only had positive reactions to this question, and so have my proxy presenters.

Also when talking about the importance, tone matters a lot because you do not want them to de-engage, which is very possible given the amount of presentation judges have to sit through. Emphasize through tone the significance of the issue and drive it home. Also I would even point to or use a pointer to help the judges know where in the presentation you are referring to when you speak, rather than them trying to guess what part you are talking about.

Methods/Results:
This mostly matters when you're discussing a research based project, always highlight the results and sample size when advantageous to you. Also make sure to alway compare your stats to others as reference when appropriate.

Conclusion:
This is where you wake them up from the snooze fest that is methods and results. Re-inject some energy at the last part of your presentation (do not speak too fast, but speak with vibrance) when restating the significance of the issue at the core of the project then emphasize how important your project is in comparison to

20. Resources

Example Videos: https://youtu.be/7kz1UkOvhAk (ISEF-BMED)
https://youtu.be/OZuWIxWBWH4 (NYSSEF 3rd place-TMED)

Example Boards:
https://www.mediafire.com/file/53r4bbhgtpym58y/NYSSEF_BOARD_PPT.pdf/file
(BIOMEDICAL ENGINEERING/SCIENCES)

https://www.mediafire.com/file/6k22g433gxl9tg3/nyssefboardfinal.pdf/file (EARTH AND ENVIRONMENTAL OR ENVIRONMENTAL ENGINEERING)

https://www.mediafire.com/file/qkww0htb90oftgp/Materials_and_Bioengineering_Justin_Cheung_ISEF_%25281%2529.pdf/file (MATERIAL SCIENCE) https://www.mediafire.com/file/kt0pdsjk4bzrj12/Computational_Biology_Rebecca_Alford3_%25281%2529.pdf/file (COMPUTATIONAL BIOLOGY)

https://www.mediafire.com/file/l7ivw8ngboousbl/Cell_and_Molecular_Biology_Scott_Massa_ISEF_2015_%25281%2529.pdf/file (CELL/MOLECULAR BIOLOGY)

https://drive.google.com/file/d/1EO368XeWoh8yN9B606v_sgxjfR_OnRBR/view?usp=sharing (SOME ML PROJECT NO IDEA WHAT CATEGORY)

https://drive.google.com/file/d/16pDvD3pALdyjM01q-ZSQarDkxjGhyUOi/view?usp=sharing (CELL AND MOLECULAR BIOLOGY)

https://drive.google.com/file/d/1T97_Yz1smZLt3kcgBCU78OBhnaeCaN4n/view?usp=sharing (ANOTHER ML PROJECT NO IDEA WHAT CATEGORY)

https://drive.google.com/file/d/1p10WDL3Vsk_Q7jLk7DnlI3Y58uQ9fH-H/view?usp=sharing (ELECTRIC ENGINEERING)

https://docs.google.com/presentation/d/18TijwEG6biQhUwSxMWZqLH53J26M6Am3/edit?usp=sharing&ouid=110550642057288758741&rtpof=true&sd=true (NO IDEA BUT IT'S ABOUT TEETH)

https://drive.google.com/file/d/1qGiFhIAOFEsmJoBpFisUe7JY3zQTjvvK/view?usp=sharing (MICROBIOLOGY)

Example Research Papers (I would NOT copy these for regeneron, I despise writing papers and the results show, however for LISEF and NYSSEF it don’t really matter and it’s not accounted for whatsoever, I would recommend looking for highly cited papers online instead if you want to see a really well written paper):
https://docs.google.com/document/d/1_Ncp-FamYFc2Ug5WoU385u32hhAClASUxkO8id5Fw9Q/edit?usp=sharing (this is mine)
https://docs.google.com/document/d/1KKI2QcP8e5uK4oVr-A2-ENSK5FnTU63UskQsqmwazu4/edit?usp=sharing (somehow got results I think from somewhere no clue just know someone was throwing a fuss about someone going somewhere because of something regarding this paper)

Websites:

Use the smithtown west school database with password/username “shsdb”

To access research papers use https://sci-hub.41610.org/ (sci hub keeps getting taken down since its technically illegally leaking papers, but everyone uses it unless you want to pay 100$ per paper, thats gonna add up after 20 citations assuming that you find the perfect paper to use all 20 times) and just paste the link to the research paper in whatever website you find through that link.

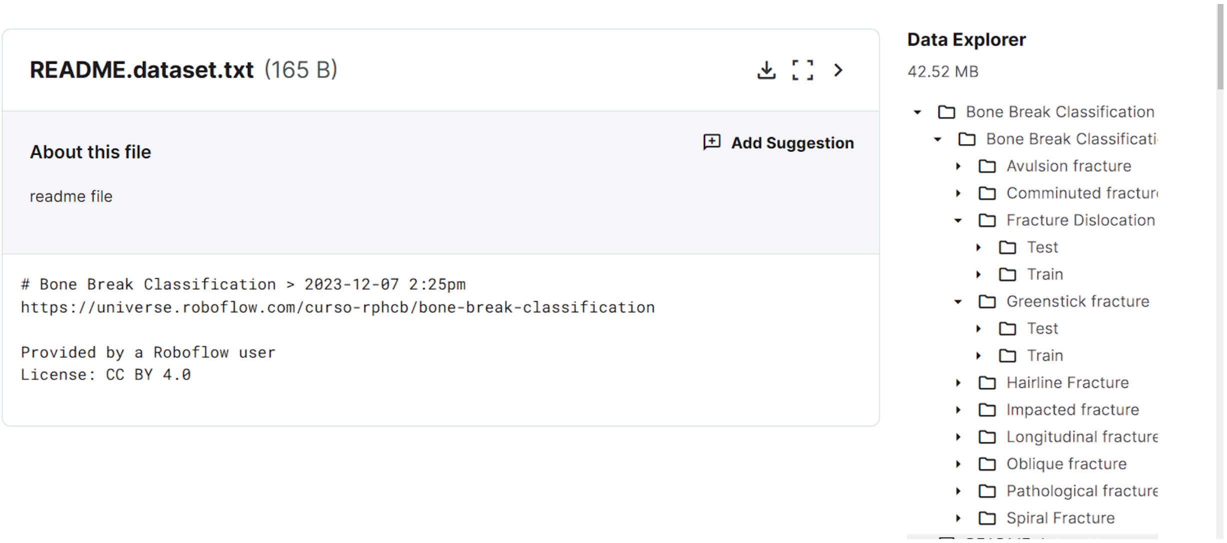
Search up “collegebored” on github and do what they (theplummer) says. You're welcome.

SAMPLE DATASETS (with project ideas, and what code to use)

This doesn't include the sample dataset already put with image segmentation

File navigation just in case:

Sometimes you're not gonna just see the train and test right in front of you, but don't be alarmed, you have to look for it. Scroll down on the data card and look to the right on data explorer. You don't have to download it when its on kaggle like this, but when you do the “add input” for your notebook you are gonna have to navigate to the right test and train you want and copy those paths for your code.



README.dataset.txt (165 B)

About this file

readme file

```
# Bone Break Classification > 2023-12-07 2:25pm
https://universe.roboflow.com/curso-rphcb/bone-break-classification

Provided by a Roboflow user
License: CC BY 4.0
```

Data Explorer

42.52 MB

- Bone Break Classification
  - Bone Break Classificati
    - Avulsion fracture
    - Comminuted fractur
    - Fracture Dislocation
      - Test
      - Train
    - Greenstick fracture
      - Test
      - Train
    - Hairline Fracture
    - Impacted fracture
    - Longitudinal fracture
    - Oblique fracture
    - Pathological fracture
    - Spiral Fracture

Image segmentation:

https://www.kaggle.com/datasets/faizalkarim/flood-area-segmentation?select=Mask - this can help develop a model that identifies flooded areas using pictures

https://www.kaggle.com/datasets/vpapenko/nails-segmentation - nails

https://www.kaggle.com/datasets/tapakah68/supervisely-filtered-segmentation-person-dataset - human segmentation, can help detect humans through images and cameras which can be used for security, self driving, etc.

https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery - (all data within the tiles, so your gonna have to either combine and make resolution the same or run it alot of times) something I found which could honestly be useful for my project. This helps identify different terrain features like land, water, roads, etc which can be used to help plan infrastructure, manage agriculture, and whatever you can think of.
https://www.kaggle.com/datasets/prathamgrover/3d-liver-segmentation/data - think this is for detecting liver tumors. Not actually that sure lol.
https://www.kaggle.com/datasets/tapakah68/segmentation-full-body-mads-dataset - another human segmentation dataset, probably more user friendly but smaller dataset than the other one.

Image classification/prediction:
https://www.kaggle.com/datasets/bmadushanirodrigo/fracture-multi-region-x-ray-data - identify what x-ray scans involve fractured bones or not
https://www.kaggle.com/datasets/pkdarabi/bone-break-classification-image-dataset - another bone breakage dataset, but it's thick. This could be a big project if you use it, but there are alot of different types of bone breakage files and test and train accompanying each one. You could make models for each of those, or even combine them for one big boner model.
https://www.kaggle.com/datasets/mdwaquarazam/microorganism-image-classification - This ones alot more tricky than the other ones, because it doesn't split the data into train and test for you. Therefore, you will need to split them manually into your own test and train files. I hope you are smart enough to do this, but if you are not, then ask someone else because me saying “I hope you can do this” will most likely off put you from contacting me for this question out of embarrassment. Oh and remember, 80% train 20% test. Also for the actual project this can help detect microorganisms, and since you may work with them in the lab for research, could be cool to have, or if you expand upon it maybe even your research project.

Non-image classification/prediction(no tutorial but I can just do it if you make a good enough case that I should): OH AND BTW ALL CSV DATA FILES ARE NON IMAGE.

https://www.kaggle.com/datasets/jainaru/parkinson-disease-detection - parkinson's disease has no definitive laboratory process to diagnose, and monitoring progression to diagnose requires a lot of labor and weekly check ups over a period of time. A model that can quickly identify PD using patient data would be greatly beneficial.
https://www.kaggle.com/datasets/alsaniipe/differentiated-thyroid-cancer-recurrence-dataset - Massive and amazing dataset that offers patient data in relation to thyroid cancer, this is an easy predict cancer project.
https://www.kaggle.com/datasets/uciml/mushroom-classification - mushroom classification
https://github.com/duanyiqun/DeWave - This is not a database, but rather a pre-trained model that converts brain waves into words/text at 40% accuracy. If you utilize this with some hardware that can read brain waves like eeg click, you have the foundations of a project that can help the mute, paralyzed, and anyone with a disability causing them to have speech problems. Just a cool idea I doubt anyone will dare to do, but you make this into a viable solution, and you have a very good shot at winning ISEF (50k) and more importantly having an actually viable product.

Contact
Discord:boom120
Email:shija051@verizon.net
Mobile:(934)777-5552