Project Planning Worksheet

To pass this course, you'll need to create a project that matches this criteria:

> *"Based on your understanding of the material, you're required to build and submit an open-source project that uses NVIDIA Jetson and incorporates elements of AI (machine learning or deep learning) with GPU acceleration, along with a video demonstrating the project in action. For example, you could collect your own dataset and train a new DNN model for a specific application, add a new autonomous mode to JetBot, or create a smart home / IoT device using AI - these need not be limited only to topics covered in the course. For inspiration, see the Jetson Community Projects page - the possibilities are endless!"*

To pass the certification, your project will be reviewed based on the following criteria:

- **AI (5 points)** - The project uses deep learning, machine learning, and/or computer vision in a meaningful way and demonstrates a fundamental understanding of creating applications with AI. Factors include the effectiveness, technical complexity, and performance of your AI solution on Jetson.
- **Impact / Originality (5 points)** - The concept of your project is novel and applies AI to solve or address challenges or issues faced by yourself or society. Also, our ideas and work are either original or derivative in a significant way.

- **Reproducibility (5 points)** - Any plans, code, and resources needed for someone else to build and use the project are included in the repository and are easy to follow.
- **Presentation and Documentation (5 points)** - The video effectively demonstrates and explains various aspects of the project, and there exists a clear, complete README in the repository that documents any steps needed to build/run the project, along with diagrams and images.

Follow these steps to plan out your project

Part One: Brainstorming

Write down 3-5 ideas for problems that you see in the world around you that you could create an AI to help solve. You can use student example projects or community example projects for inspiration or look back on past lessons that you enjoyed.

1. Chess Opening Recognition Tool - The computer can look through various types of common chess openings used by players, and then let the opposing player know what move the opponent has made. BONUS: The computer can also suggest the best move for the player to make in order to reduce the opponent's chances of winning based on the opening

2. Weeds in the Yard - Uses image recognition to identify different types of common backyard weeds. This can be used by yard owners to determine whether or not they have invasive weed species growing in their backyard. BONUS: The computer can also identify endangered plant species growing in backyards to prevent unwitting destruction of plants.

3. Sign Language Translator: Allow the computer to identify various types of sign language signs through images and video (because phrases and certain words can consist of one or more dynamic sign language signal), and then produce a general translation of the phrase so people who cannot understand sign language can communicate with those who are limited to sign language.

4. Guitar Chord Recognition: Feed the program multiple pictures of guitar chords/notes on the guitar fretboard, along with various different finger positions on the guitar to identify chords and keys in order for guitar players to use the exact keys from live guitar performances.

5. Project Mini Baymax - identify the physical symptoms of certain diseases and differentiate between similar diseases to help healthcare professionals provide accurate treatments.

Part Two: Details

Write down the answers to these questions for your **two favorite** ideas:

AI: How would the AI work? Technically speaking what kind of network is it and how does this network work?

Idea 1 - Chess Openings:

- Image Recognition - uses existing images of chess openings + names, and returns the closest match of the opening with confidence level.
- Dataset: no image datasets exist online, instead can be created through pictures of openings on chess board, chess.com, etc.
- Idea for program:
    - Imagenet
- Use the video camera - dev/video0 - to provide input for the board
- Create data set to use the 8 most commonly used chess openings, as there are over 3000 existing chess openings

Idea 2: Mini Baymax:

- Image recognition software - uses uploaded images of patient photos to detect symptoms of similar diseases and returns the closest match
- Dataset: https://www.kaggle.com/datasets/joydippaul/mpox-skin-lesion-dataset-version-20-msld-v20?select=Original+Images
- Command used: imagenet
- Dataset has 6 different image classes, 5 diseases and 1 class to recognize someone as healthy

Impact: What impact would this project have? Who does it impact and in what ways?

Idea 1 - Chess Openings:

- Using an AI model to identify different types of chess openings can help beginner chess players learn the primary types of openings that many players including grandmasters use. By learning to recognize openings and learn the best responses to these openings, beginner chess players can advance their chess skills far faster just by using the AI and playing against other people.

Idea 2 Project Mini-Baymax:

- Using an AI model to identify injuries can help both patients and doctors determine what kind of injury, and what the best action would be towards healing the wound. For example, a treatment for chicken pox and acne is different, but because they can look similar, it will be hard for the program to differentiate between the two.

Part Three: Resources

Now that you have thought out the impact and technical aspects of how the AI will work, it is time to map out what resources are going to be needed to complete your project.

Docs from jetson-inference: Add your documentation or tutorial link below

Step 1: Train a model following the steps from this doc

Step 2: Write a classification program using steps here

Also possible to train on browser using this colab notebook

Example code: Add your example code below

Using the file based of the my-recognition.py file:

Import the imagenet files from the jetson inference library

Specify the input/output, and the model used as well (also ensure that the resnet18.onnx model is saved)

Datasets: If applicable, add the dataset that you will be using below

https://www.kaggle.com/datasets/joydippaul/mpox-skin-lesion-dataset-version-20-msld-v20?select=Original+Images

Miscellaneous: Add any other resources you might need for your project below.

Part Four: Documentation

Video: Write down any key points that you want to add into your video below

- What the program does - the program is modeled after the character Baymax from the movie Big Hero 6. The program uses image recognition software to differentiate between the physical symptoms of different types of diseases such as monkeypox and measles, and can also identify if a person is sick or healthy due to their physical appearance
    - Can help doctors identify different types of similar looking diseases to help them provide the correct treatments for diseases

- How to build the program from scratch - show the kaggle documentation and the folder from which the dataset was obtained
    - Step 1: Folder file download and then zip (show fold 1)
    - Step 2: Upload the zip file to VS Code and then unzip with the unzip command in the directory jetson-inference/python/training/classification/data
    - Overcommit memory: echo 1 | sudo tee /proc/sys/vm/overcommit_memory
    - Step 4: Enter the docker container (./docker/run.sh)

    - Step 5: cd to j p c and then run: python3 train.py --epochs=10 --batch-size=2 --model=models/filename data/filename
        - NOTE ABOUT THE BATCH-SIZE AND THE CANNOT COMMIT MEMORY ISSUE, AND HOW THAT'S REDUCED THE ACCURACY (see next point)
    - Once model is finished training, export the resnet18.onnx model to the model file: python3 onnx_export.py --model-dir=models/filename

- Issues faced by the program:
    - Batch-size problem on my desktop and how that was resolved
    - NOTE: the batch size issue is one that led to a decrease in accuracy, and this was due to the limits faced by the VS Code software on my OS, which is why the accuracy of the program isn't very high
    - To increase accuracy, increase the number of epochs or train in the browser to run a better model.

- Run the program with images on the file
    - Cd to the my project directory and upload an image (use pre uploaded image)
    - Run: python3 my-rec monkeypox-1.jpg

Documentation: Write down any key points that you want to make sure are in your readme doc.

- What the program is - what problem can be solved using this model?

- Steps to create program (see below) + dataset link

- How to run the program

Reproducibility: How could your project be reproduced or run on another machine. Make sure to remember all steps that make your project work.

1.

2.

3.

4.