Detailed Design: (Tasks)

* Add justification for scope limit:
* Go though the basic how it works and why.
  + Create a basic flowchart on how it works and this can be used in the final report.

NOTES:

* A basic overview of the control system is provided below.
* The detailed design and control algorithm is going to be shown in a flowchart. Emily Neil is going to help me work though that tomorrow (Wednesday Night) to see what software needs.

Detailed Design – Controls AI – Image Recognition

# Section1: Objective

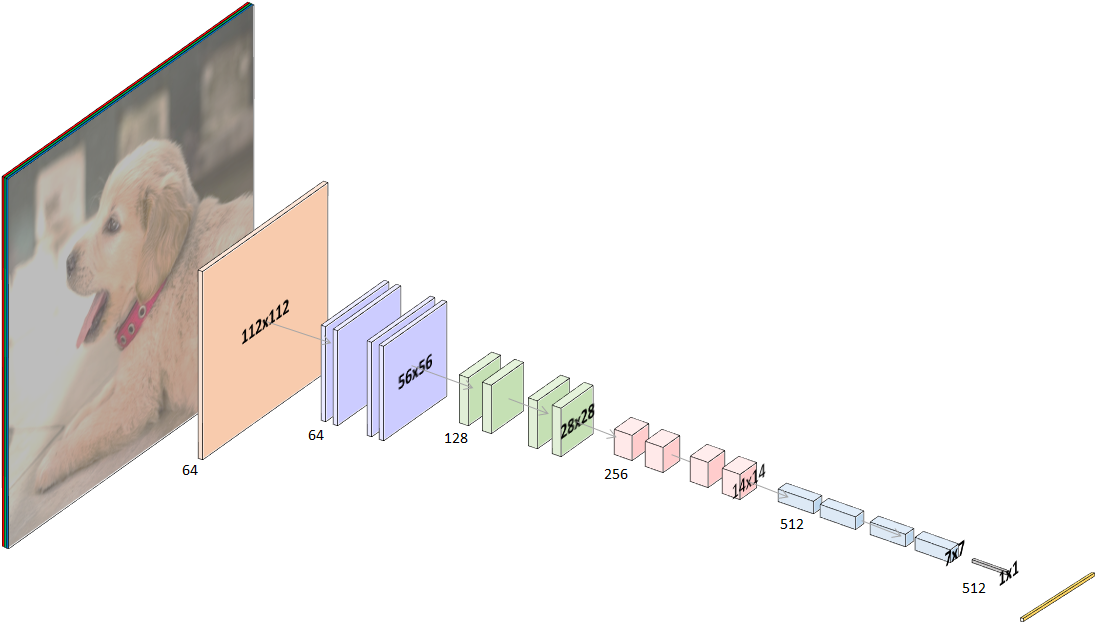
* Objective of the project is to increase the number of recyclables engineering the recycling stream
* To do this we need to identify the type of item and process it accordingly
* The current and best method found to identify the recycling was using computer vision since most of the recycling that we will be sorting is common household items, this includes but is not limited too cans, plastic containers, paper, cardboard boxes, amazon packages and many more.
* Since most of the items that will be recycles are common items, AI though the form of image recognition was though to be the best method since the library could be created with common household items.

# Section 2: General Overview

The Image Recognition will be performed on the Jetson Nano 2GB with a USB V4L2 camera. The Jetson Nano will use Image Classification techniques to determine the recycling object and determine the material composition. To use the image classification a model will need to be created to compare the photo to it.

The basic process of the jetson is to determine the type of recycling and then trigger the sorter mechanism to move and sort the object into the correct bin. The Jetson has GPIO pins which are going to be used to control the output and sorting mechanism. GPIO pins are general purpose input output pins. These pins are assignable to be able to read or write digital or analog signal to other instruments or sensors.

The image classification model will be trained using transfer learning, this is a technique for re-training a DNN model on a new dataset. Instead of using a dedicated server or cloud instance to train an entire new model, the Jetson can be used to train a custom model of household recyclables by using a pre-trained network. The figure below shows how the model will be trained by looking at smaller sections of the image and then keep moving into larger sections to ensure that it understands and can recognize the image properly.



A simple python script will be used to run the images coming from the USB camera though the trained model to determine what the object is that is placed into the camera view. The python script will overlay the image class as well as the confidence level. The confidence level is how accurate the Jetson thinks the object is one of the classes. As stated in our objectives and constraints of the project, if the Jetson is not confident to a level of 80% the item will be sorted into the garbage. The model may not be accurate if the object is too different from the training database images or the lighting or other factor could cause the lack of certainty. The object will be placed into the garbage if the certainty level is not high enough to ensure that no waste is sorted into the wrong bin. It is better to not recycle something if you are unsure than to contaminate the entire pile of sorted recycling.

# Section 3: Detailed Overview

The following detailed overview section will go over a few different reasons and the process to using the Jetson Nano for the image classification process.

Sections:

1. Image Collection and Sorting

To create and build an image classification model, you will need images to train and test your model with to ensure its accuracy. For the project PyTorch was used to train the model used for the recycling sorter. The model needs hundreds and thousands of images to be able to have a good understanding of what the different classes are as well as how to distinguish between them.

PyTorch the training technique that we are using for creating the model requires the images to be split into 3 sections for each class of images. The 3 sections for each class that the images need to be split up into are, train, test, and validation. Each of the sections has an important purpose for training and creating the model. The train image-set should be able 80% of the total images that you have, this dataset is used for training the model and finding features and similarities between the different images to be able to create the model.

The validation image set is used to test but mainly validate the model that is created. PyTorch creates many models during the training processes and the computer needs a way to test itself to determine which model is the best. The training process uses the validation image set to test the model it has just created and comes up with a certainty rating. Based on the certainty rating the model can be kept or re-trained to be able to increase the certainty rating to a satisfaction level.

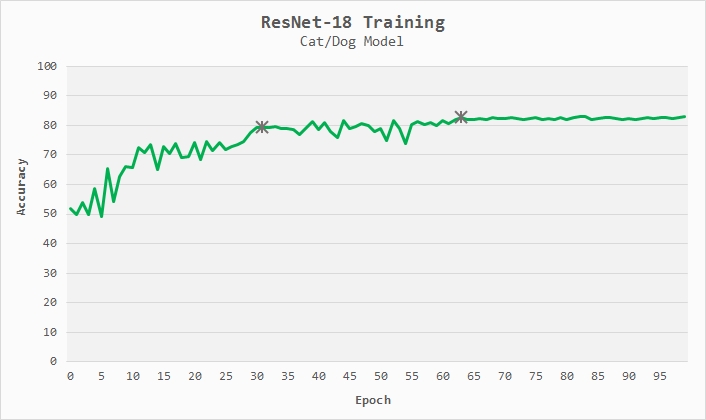
The test image data set is used for the programmer to test the model yourself. The computer will be able to test the model based on the validation data and determine if the model is accurately detecting the correct images. It is also useful to test different images to see how the model will respond.

1. Image Classification Model Training

To train the image classification model, transfer learning with PyTorch was used. Transfer learning was used since it takes less time and resources to train a model based on a new dataset compared to creating a model from scratch. When training with transfer learning, the weights of a pre-trained model are tuned to classify the new dataset. So, in our case, the ResNet-18 pre-trained model was used to help train the new model of recycling images.

During the training process, there are a few settings that could be changed to help train the model to improve accuracy. Some of the settings were changed to test if it would improve accuracy but then were returned to default values because they were providing the best results.

An important setting that needed to be changed was epoch. An epoch is one complete pass though the dataset of images though all classes. The default number of epoch’s is 35 which provides a great starting point for models because that is usually wear the model accuracy starts to level off as shown in the figure below for a Cat/Dog trained model. The figure shows that at around 30 epochs the dataset reaches about 80% accuracy and at 65 epoch’s the dataset converges at about 82.5% accuracy. The additional training time is helpful to improve model accuracy but the best way to improve model accuracy is to add more images to the training dataset.



* It is important to know that the accuracy of the training model is determined by the validation image set, the model tests the image set with these validation images, so if you have images that are not performing well then you can put them into the validation image set to help pick the better model.

1. Image Classification Running

To be able to run the created image classification model, we need to convert it from the re-trained (transfer trained) ResNet-18 model to ONNX which is an open model format that supports most of the popular machine learning frameworks. A simple python script was used to convert the ResNet-18 model to a ONNX model that can be read by TensorRT which is what we use to process the images on the jetson in a python script.

The image classification technique that we are using supports both live camera feed though V4L2 USB cameras as well as .jpg image processing. The .jpg image processing was useful for testing of the model to ensure accuracy as well as to send a folder of images in at once to determine how it would react to them as well as keep a record of the images and the percentages for future training purposes.

1. Python Script – GPIO Pins – Output

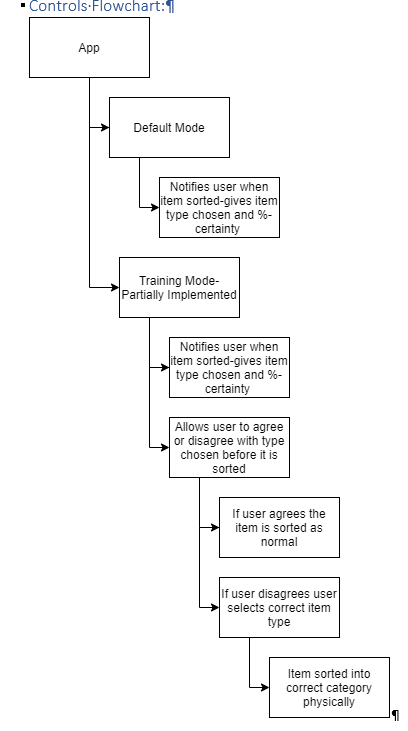
A python script is used to run the image processing network which also puts the class it thinks it is as well as the accuracy level. The script will also determine when the item in the hopper has changed from just the background to an item class so that it will be able to trigger the motors and actuators to move the item into the correct bin. The sorting process will only begin after 10 confident guesses of 80% or more accuracy to ensure an item is ready to be sorted as well as ensure that the image is classified correctly. If the software detects that an object is there but the confidence interval on the type of recycling is not high enough to meet the threshold of 80% for 20 guesses (frames) then the item will be sorted to garbage to ensure that that no containments will be put into the recycling and cause problems later though the recycling process.

Within the python script the GPIO (input and output signals) pins are able to be signaled to provide the input and output signals to the motors and actuators to which are used to control the movement of the sorting mechanism. The script will determine when an item is ready to be sorted and placed into a bin after following the criteria shown above.

The python script will also send the information to the app over a Bluetooth connection. The app is what is used for the user to interact with the Jetson Nano (controller). The python script though the help of a Bluetooth USB dongle will open a connection with the phone and then be able to send the information to the phone app. The information that will be sent to the phone is the recycling class for the current image as well as the accuracy percentage, the user will also be able to see on the app the status of the current and past 100 items recycled and sorted to be able to check their log and how accurate the detection method is.

# Section 4: Process Detailed

Emily Neil and I are going to work on some more flowcharts and process steps tomorrow (Wednesday Night) to ensure that Bill will understand and know what the controls is planning to do as well as how to interact with the software (app).



# References:

<https://towardsdatascience.com/how-to-train-an-image-classifier-in-pytorch-and-use-it-to-perform-basic-inference-on-single-images-99465a1e9bf5>

<https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-transfer-learning.md>