The Training Data that is output from the simulator has an excel file that shows the paths to images taken from the center of the car, the left side, and the right side for every segment of time.

The measurements it records are a Steering value (-1 to 1), a Throttle value (0 to 1), Brake value (all 0), and a speed value (0 to 30).

All these measurements can be useful to train your network, but let us start with only using the steering measurements.

In this case the Images are your Feature Set (THE INPUT), and the Label Set (Steering Measurements).

They will train the network to predict training measurements.

**Feature Set**

Each of the images shows the hood of the car and the distant background in the image. They are rgb images with a width of 320px and a height of 160px.

The top of the image has trees and stuff that might distract the model more than help it. At some point it can make sense to crop that section out of the images.

The reason a left, right, and center images are used to train your model is because it can help generalize the model. These images teach the network what it like to be off the center of the road, and what it takes to come back to the center.

**Validation Model:**

When your **model predictions are poor on both the training and validation set**, then you should either increase the number of epochs or add more convolutions to the network

When your **model predicts well on the training set but poorly on the validation set, then this is evidence of over fitting**, which can be handled with dropout or pooling layers, fewer convolution and fully connected layers, or collect more data or further augment the data set.

**Data Augmentation:**

Sometimes the data we collect can be biased to give a certain result. For example For this project we are driving around the track counter clockwise, which involves a lot of left turns to stay centered. This would bias the output of a model trained on this data to mostly take left turns. This can be corrected by augmenting our input data.

If we flip our images horizontally and multiply our steering measurements by -1, and train our model with this new data in conjunction with the data we acquired, we can train on a balanced dataset.

Augmentation has the benefit of providing us more data that we can train on, as well as the data we train on is more comprehensive.

Why use multiple cameras to get data for a car to autonomously drive with a Neural Network?

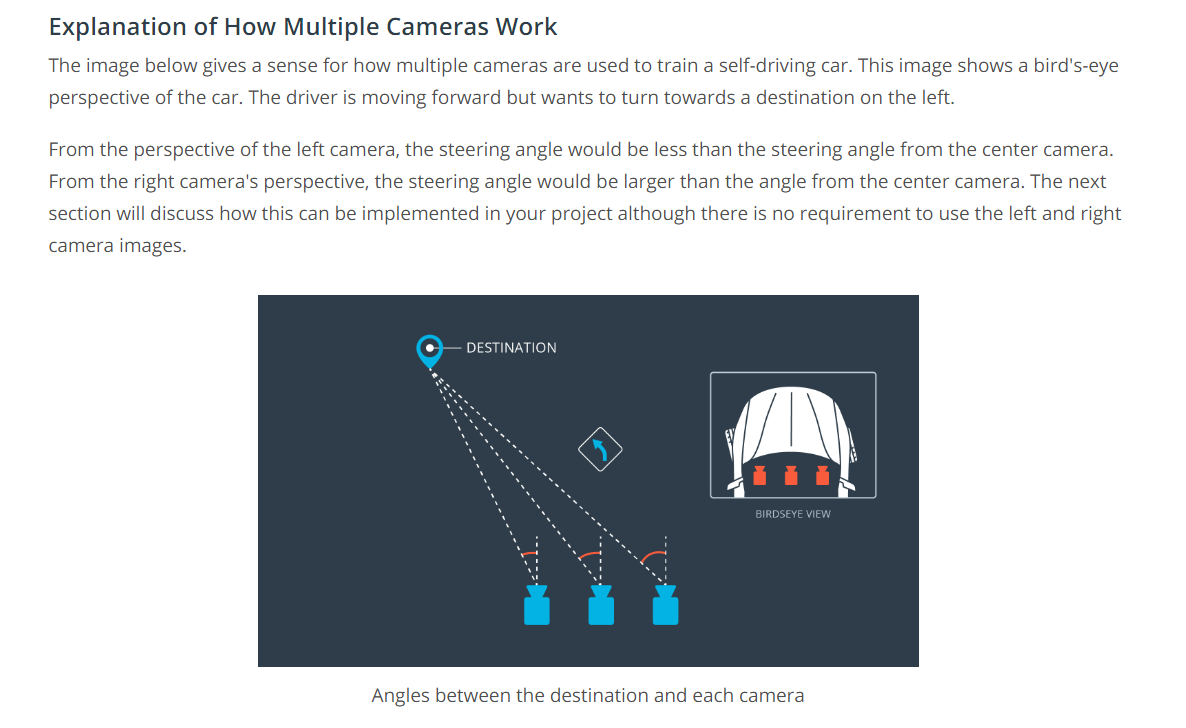
We are gaining camera images from three cameras in the simulator. We are getting images from cameras mounted on the center of the car, the left of the car, and the right of the car.

Now if we think about it, in a simulator we can weave all over the place to gain information, but in a real car we can’t do that.

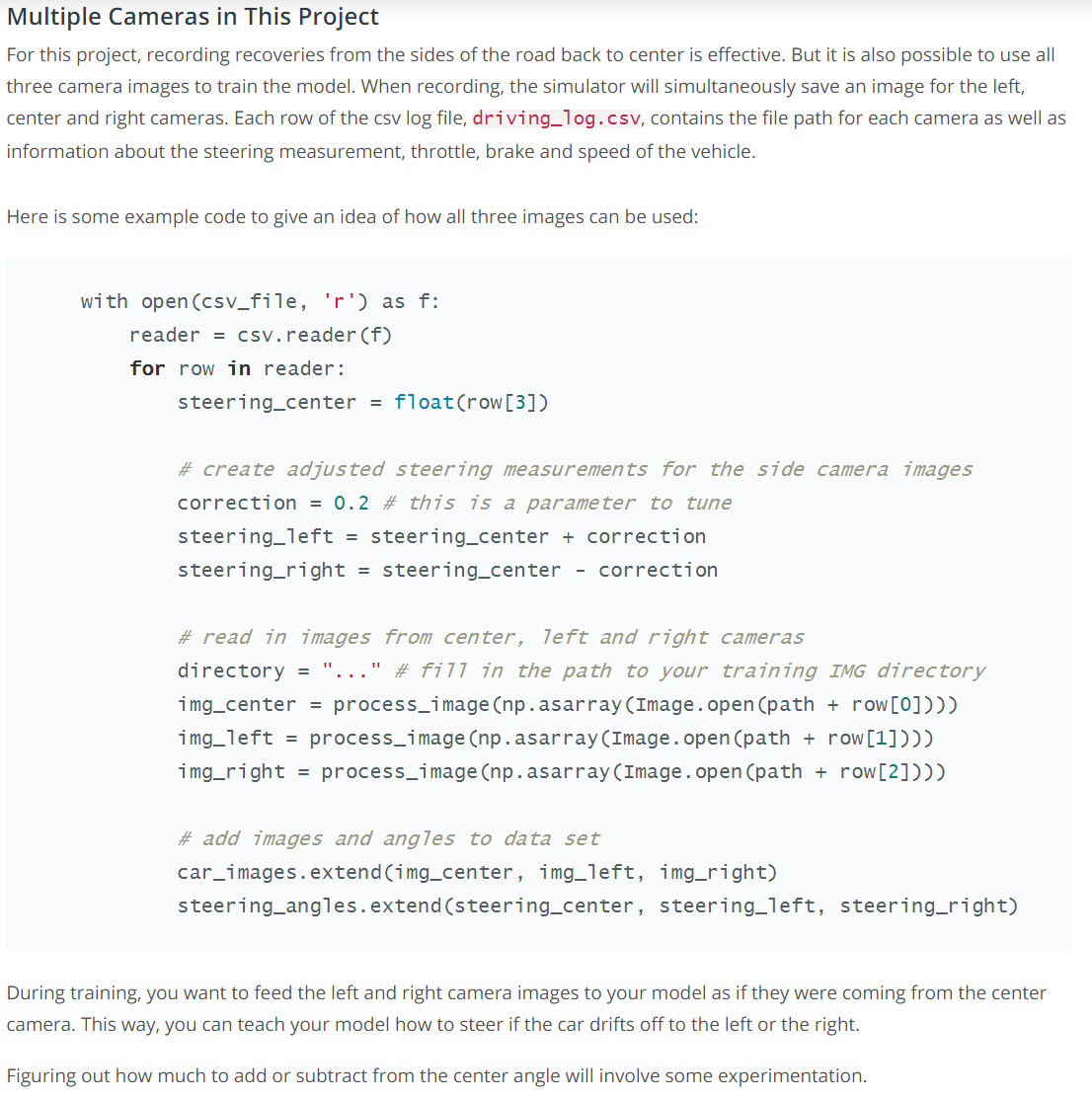
So on a real car, we use multiple cameras and map recovery paths from each camera.

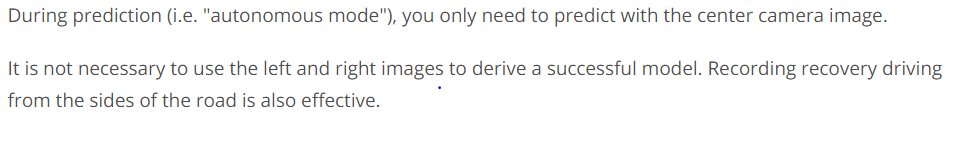
Having multiple cameras placed on the car lets you simulate your vehicle being in different positions of the road, if the left and right images are viewed by the center camera.

**How Multiple Cameras Work.**



The Idea is that if the your center camera sees the image from the left camera, then it should respond with a little bit of an extra right turn to get back to the center. You can handle this by adding a small correction factor to the left images steering measurement, and subtracting the same factor from the right images.





Generators:

When we look at the images we are using as our input, they are 160x320x3 which for a large data set like 10,000 images can take over 1.5 Gigs of memory. When we are preprocessing the dataset at the same time, it can change from integer values to float values for pixels, which can increase the size of the data by a factor of 4.

Generators are used for large amounts of data so that you don’t need to store the preprocessed data in memory all at once. Generators let us pull pieces of the data an process them on the fly only when we need them which is much more memory-efficient.

Generators are like co-routines in the fact that they run separately from the main routine. They are a useful python function.

Generators in python are defined like functions, but instead of returning a value, they use yield, which still returns the desired output values but saves the current values for all the generators variables as well.