Image processing:

The image processing part includes crops, flips (horizontally) and more, and we resize our images to a shape of 64x64 which we believe to be decent enough.

Data information:

We trained our model using 1024 (a round number😊) images of cars and non-cars (split 50-50). 80% is used for training and 20% for testing. The 50-50 ratio between the images of cars and non-cars is maintained for both training and testing.

We tried adding a few extra layers to our network (4 extra unit layers), but surprisingly, not only did it not increase our accuracy, but it made it worse, not to mention the fact that it took hours to train for a few epochs. I expect that this could just be an error on our part, or that we just got unlucky or did not train our model enough but after many hours of unsatisfactory results we decided to drop this change.

Performance:

We decided to compare our model to MobileNetV2 using the weights trained on the big ImageNet dataset. MobileNetV2 is similar to MobileNet, except that it uses inverted residual blocks with bottlenecking features. It contains a surprising number of layers, however, they are very thin to compensate for their number to reduce computational complexity .Since this model contains many class predictions that would be considered “cars”, we determined a list of classes that would be considered a “car” for our interests.

We compared our model with the MobileNetV2 using 500 images of cars and 500 images of non-cars, and we computed some metrics. Needless to say, we did not include any images that were part of the training process. The results were better than expected. Our model performed slightly better than the MobileNetV2. It reached an accuracy of 92% for the car images and 98.4% for the non-car images, obtaining the combined accuracy of 95.2%, whilst our model reached a combined accuracy of 95.8%. This being said, we were positively satisfied with the performance of our model, given that it was tested on a “fresh” batch of images.