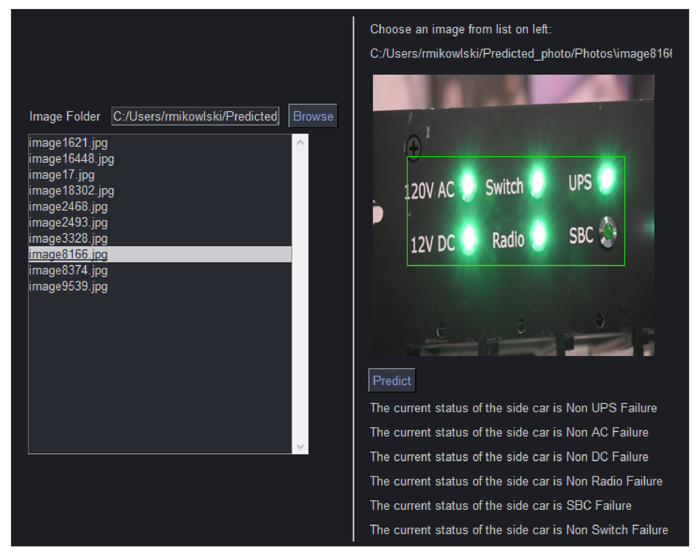
Predictive Troubleshooting of the OADMS sidecar using Machine Vision and Deep Learning

Robinson Mikowlski

Introduction

The Engineering team at MRIGlobal is trying to delve into the world of Artificial Intelligence and Machine learning, the first such work in this was the Machine Vision project involving the status of the OADMS side car. This poster will go into detail explaining the process used to create the predictive interface, the different types of algorithms used in each step of the process, as well as how work like this can be expanded upon in the future to bring a brand new capability to the institute as a whole.

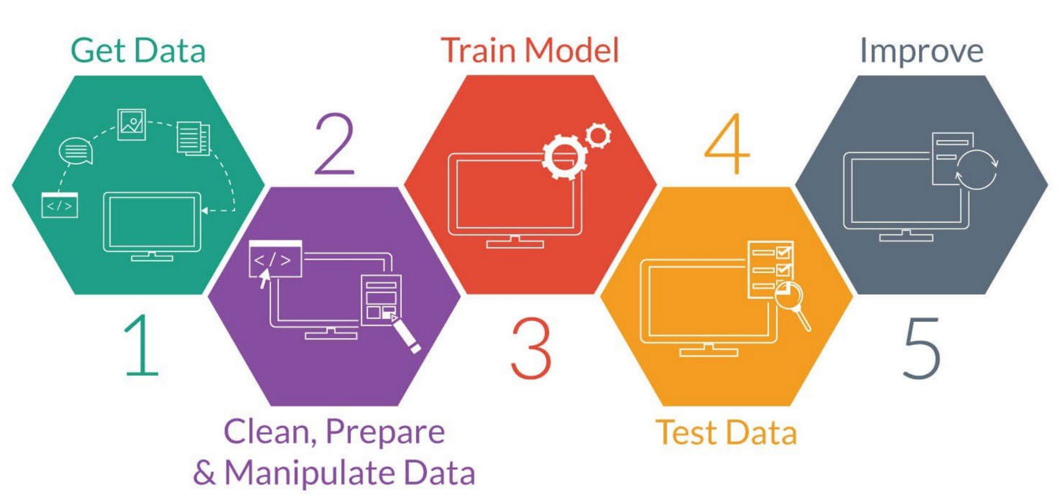
The Project here involved a multi-step process, involving object recognition using an OpenCV cascade classifier to first identify the location of the status lights for the device, then leading into 6 stacked neural networks with binary output variables to predict the status of each individual status light and provide an assessment of the current status of the machine.

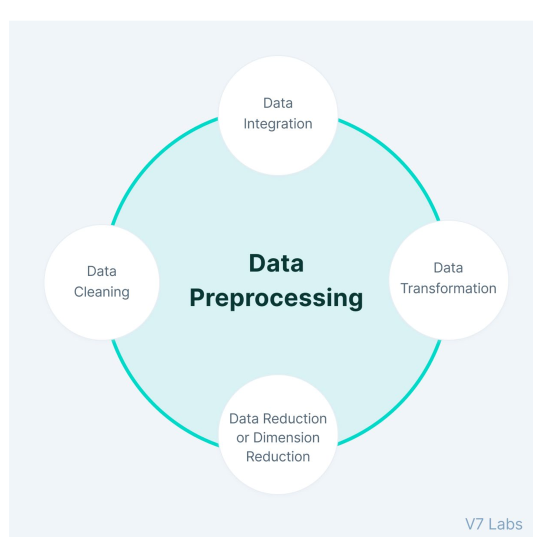


Machine Learning Overview

For understanding of this project, the first step is to create a base understanding of what machine learning/deep learning is and how it can be utilized. Machine learning involves the training of algorithms using specific training data, to then be able to predict aspects of new data based on what was learned from the training input.. For any type of machine learning project, the data goes through a few different steps.

1) The first step is gathering the data for the algorithm to train on, in this case, multiple thousands of pictures of the side car were taken to build a sufficient sample size to be able to create inferences from the input.
2) After gathering data, the data is separated into groups, and then preprocessed to prepare it for use within the training model. This can involve shrinking the data, changing the orientation of the data, as well as changing the color of the data. This step is to ensure the samples be diverse enough to use to tell the difference between data.
3) At this point, the task is creating and training the model, this can vary from simple linear regression, to multiple layered neural networks. Once built, the data is fed through the model through multiple iterations, to then teach the model how to fit the weights of the calculations to the data best.
4) Once the model is created, it must be tested to evaluate the fit to real work data. For this it is fed non trained samples and predict for these, it is then scored to see how well it predicted for this input.

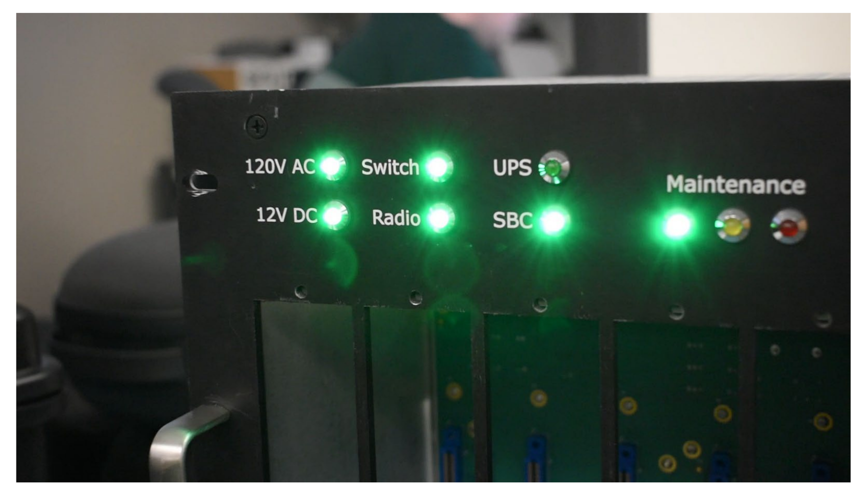




Object Recognition Portion

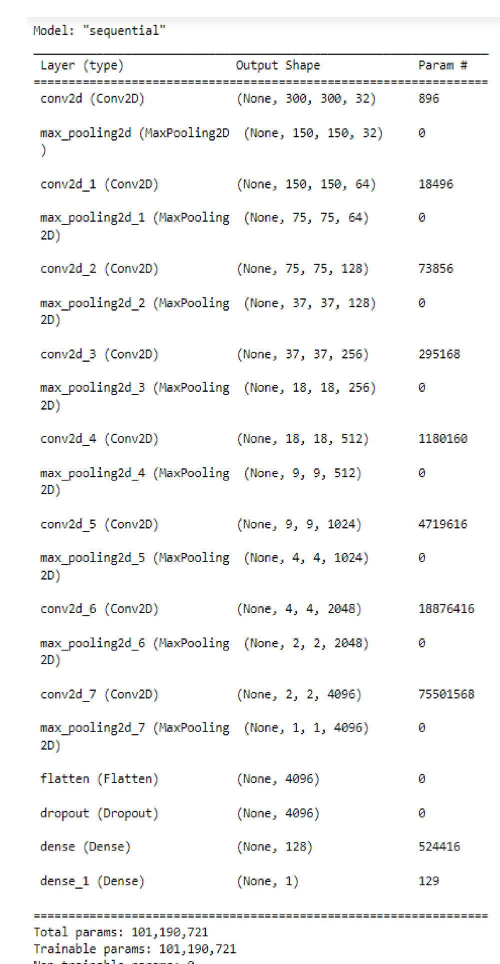
For the Object recognition portion of the project was done using an OpenCV cascade classifier. the objective was to be able to identify the status lights of the side car (shown by a green rectangle. For the classifier, images were separated to positive and negative first, the positive were then annotated to recognize the position of the status lights, then the positive and negative images were fed into the classifier itself to create the predictive model. The model was able to show success when attempting to identify the lights when the image was at a close to medium distance, but struggled at long distance.

OpenCV



Neural Network Training

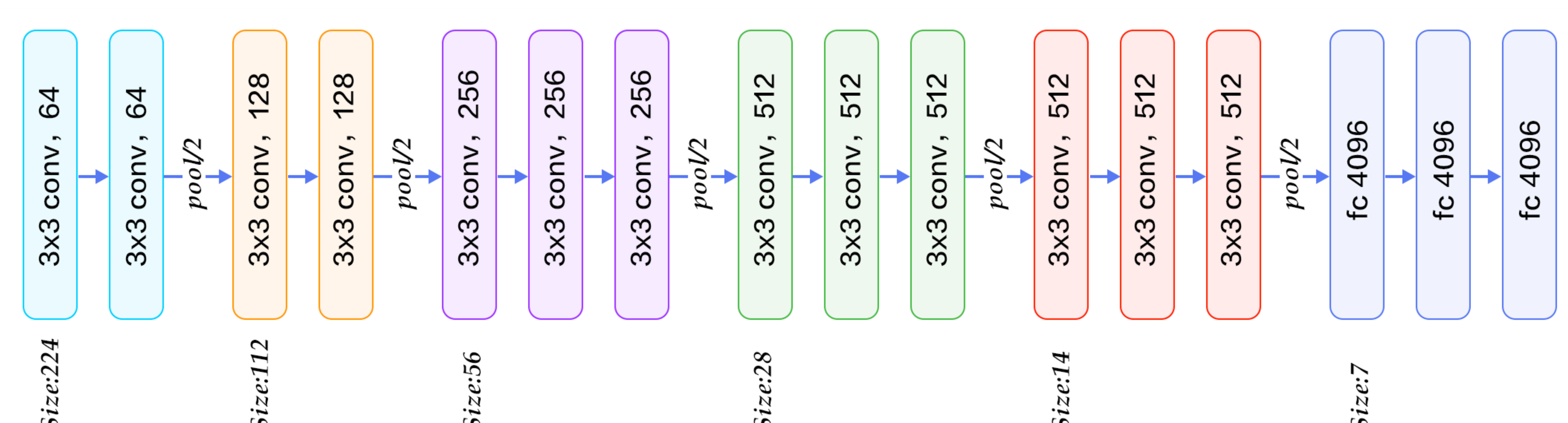
For the status prediction portion, six distinct neural networks were created, each predicting for a binary outcome (weather the light they were focusing on was on or off). For the training, The implemented model used was the VGG16 model architecture. The model is depicted below, and is identified by its recurrent convolutional layers, which are then followed by a max pooling layer to ensure not too much is lost when cutting down the input, which is then always followed by another set of convolutional layers where the neurons in the convolutional layers has doubled from the last set. Once the convolutional layers are done, flattening and dropout layers are added to enhance generalizability by removing overtrained versions, and finally a 1 neuron dense layer with a sigmoid activation for the binary predicted variable.



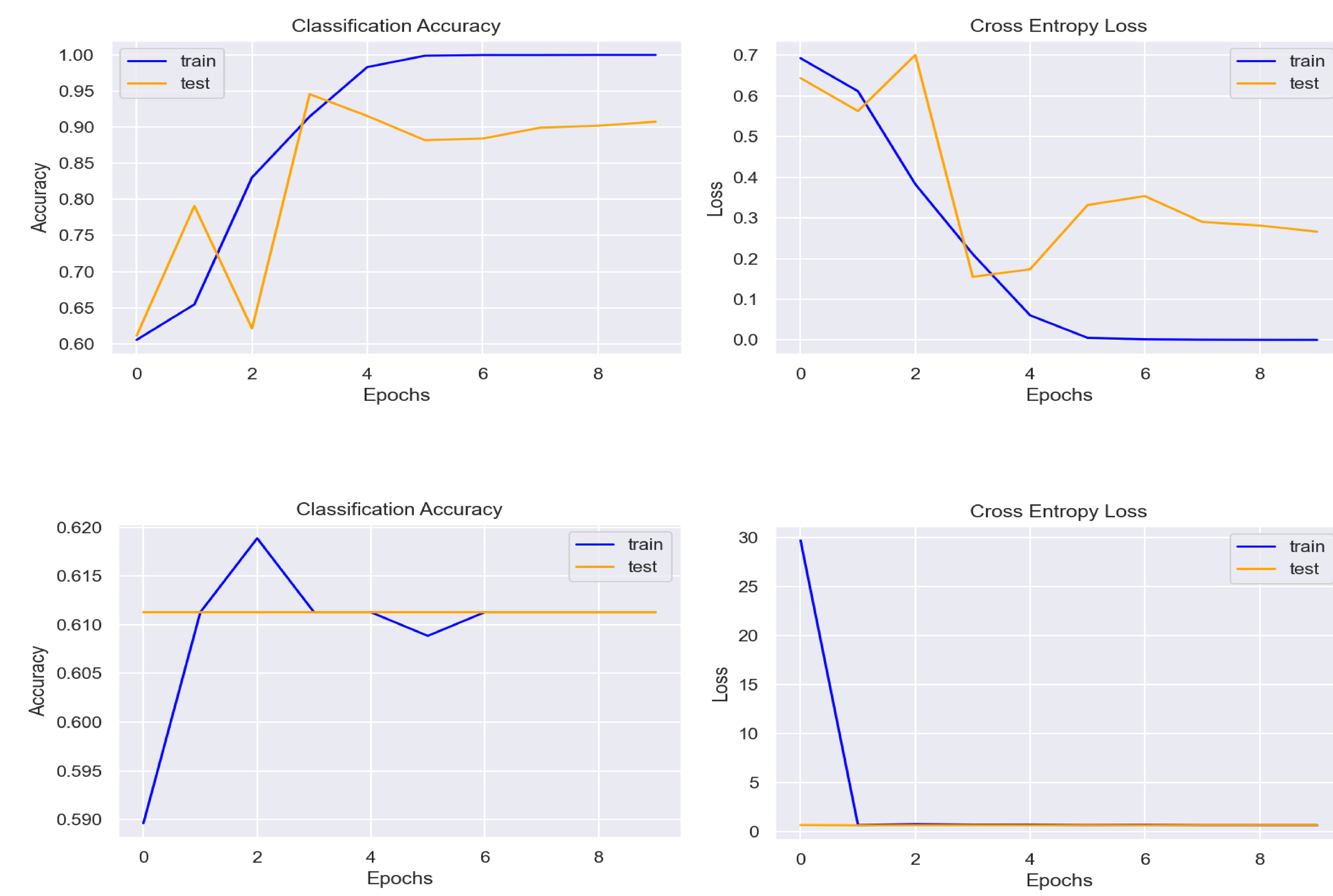
Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 300, 300, 32)	896
max_pooling2d (MaxPooling2D)	(None, 150, 150, 32)	0
conv2d_1 (Conv2D)	(None, 150, 150, 64)	18496
max_pooling2d_1 (MaxPooling 2D)	(None, 75, 75, 64)	0
conv2d_2 (Conv2D)	(None, 75, 75, 128)	73856
max_pooling2d_2 (MaxPooling 2D)	(None, 37, 37, 128)	0
conv2d_3 (Conv2D)	(None, 37, 37, 256)	295168
max_pooling2d_3 (MaxPooling 2D)	(None, 18, 18, 256)	0
conv2d_4 (Conv2D)	(None, 18, 18, 512)	1180160
max_pooling2d_4 (MaxPooling 2D)	(None, 9, 9, 512)	0
conv2d_5 (Conv2D)	(None, 9, 9, 1024)	4719616
max_pooling2d_5 (MaxPooling 2D)	(None, 4, 4, 1024)	0
conv2d_6 (Conv2D)	(None, 4, 4, 2048)	18876416
max_pooling2d_6 (MaxPooling 2D)	(None, 2, 2, 2048)	0
conv2d_7 (Conv2D)	(None, 2, 2, 4096)	75501568
max_pooling2d_7 (MaxPooling 2D)	(None, 1, 1, 4096)	0
flatten (Flatten)	(None, 4096)	0
dropout (Dropout)	(None, 4096)	0
dense (Dense)	(None, 128)	524416
dense_1 (Dense)	(None, 1)	129

Total params: 101,190,721
Trainable params: 101,190,721
Non-trainable params: 0



Neural Network Evaluation



Summary

When locating the status lights of the side car, as mentioned before there were some failures, but on a whole the algorithm was in most cases successful, and as for the predictions of status, all 6 of the predictive algorithms had a greater than 90% accuracy when predicting for their specific light when tested against non trained data.

The project here showed promise, and while not extremely practical in use, gave light to the capability for future machine learning projects (be it machine vision or otherwise). The future possibilities for machine learning at MRIGlobal are vast, with future opportunities to grow the capability and cater to a wider market of clients.

Contact Information

Robinson Mikowlski
T:(816) 326-5438
E:rmikowlski@mriglobal.org

MRIGlobal
454 Dr. Martin Luther King Blvd
Kansas City, MO
64110