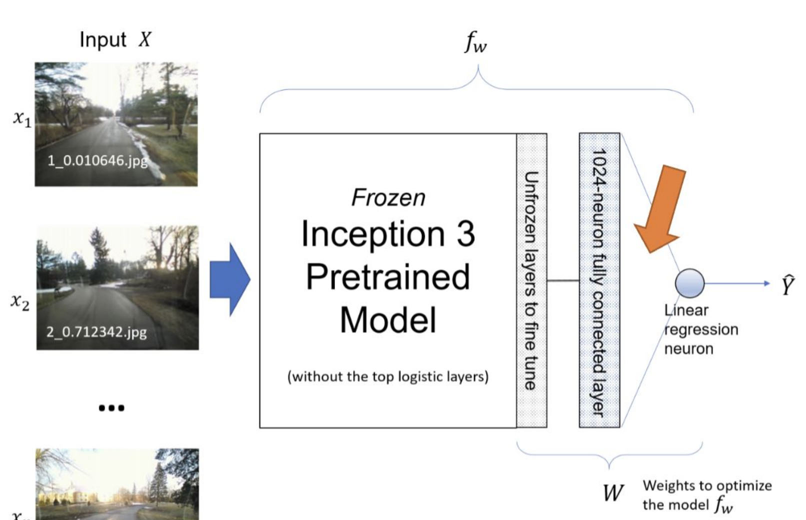
Teaching Cars to Self-Steer on Unmarked Roads with Convolutional and Recurrent Deep Learning

Austin Ramsey, Giuseppe DeRose, Justin Dombecki, Nicholas Paul, CJ Chung

College of Arts & Sciences, Lawrence Technological University

INTRODUCTION

This research presents an autonomous steering solution for vehicles on unmarked roads using inexpensive sensors, low computational resources, and a combination of machine vision and machine learning based on a fine-tuned InceptionV3, Recurrent Neural Network (RNN), and Dense Neural Network (DNN). Despite challenges in computation limitations and environmental conditions, the team acquired data through a forward-facing webcam and an Ubuntu laptop, with images pre-processed to reduce environmental effects. Building upon previous research [1, 2, 3, 4], a RNN layer was added to the InceptionV3-based architecture, which predicted the steering angle needed for navigation.



$\hat{Y} = f_w(X)$ Model f_w's prediction with input X

$Y = <0.010646, 0.712342, \ldots, -0.804531>$
Desired steering wheel angles by a human driver

$$E = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$W' = W - \alpha\frac{\partial E}{\partial W}$$

Our research goals are to add an RNN layer to the prior Deep Steer Project research model and to see if the added RNN layer will improve the accuracy of the pre-trained InceptionV3 based model. Testing was performed with the physical ACTor vehicle on and near main campus. The results were compared to the previous model from Fall '22 [1] to see if there was improvement.

METHODS

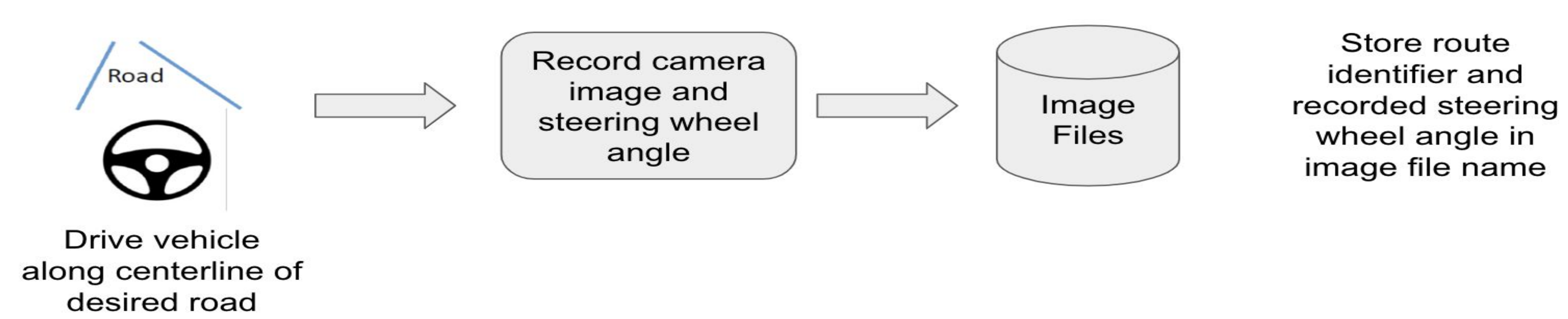
Reusing training, validation and testing data from previous research – this research brings a new model architecture and better control of how data is fed into the neural network during training.

Setting and context

The results from fall research showed great success. The smallest MAE in degrees found was 4.25 degrees of error when steering The goal is to generate better results by adding an RNN layer.

The dataset used was collected in the Fall 2022 for the previous Deep Steer research. The data was collected from several different routes near LTU campus. Routes in green are the ones used for training and the route in red is used for the application data set.

Each "frame" collected during data collection included the steering angle in the filename to be used later for training purposes.



New model architecture

To achieve the goal, the model from the fall needed rearchitecture. A reshape layer and SimpleRNN layer were added to the Keras model.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inception_v3 (Functional)    (None, 6, 8, 2048)        21802784
_________________________________________________________________
global_average_pooling2d (Gl (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              2098176
_________________________________________________________________
reshape (Reshape)            (None, 1, 1024)           0
_________________________________________________________________
simple_rnn (SimpleRNN)       (None, 128)               147584
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 24,048,673
Trainable params: 2,245,889
Non-trainable params: 21,802,784
_________________________________________________________________
```

Controlling data feed

Skip and append function takes a dataframe and start index as input. Returns a dataframe starting at the specified index. includes only every n images.

Batch size is number of images fed into the neural network at a time, during training. This is significant when working with a RNN layer as the model optimizes with the memory of the previous image.

RESULTS

Figure 1: Skipping *N* images during training (all with batch size 32)

Model	Train MAE	Validation MAE	Test MAE	Test Route MAE
Every 3	2.72 deg	2.91 deg	2.90 deg	4.37 deg
Every 5	**2.59 deg**	**2.76 deg**	**2.73 deg**	**4.36 deg**
Every 10	2.78 deg	2.90 deg	2.94 deg	4.37 deg

Figure 2: Batch size comparison (all with skipping 5 images)

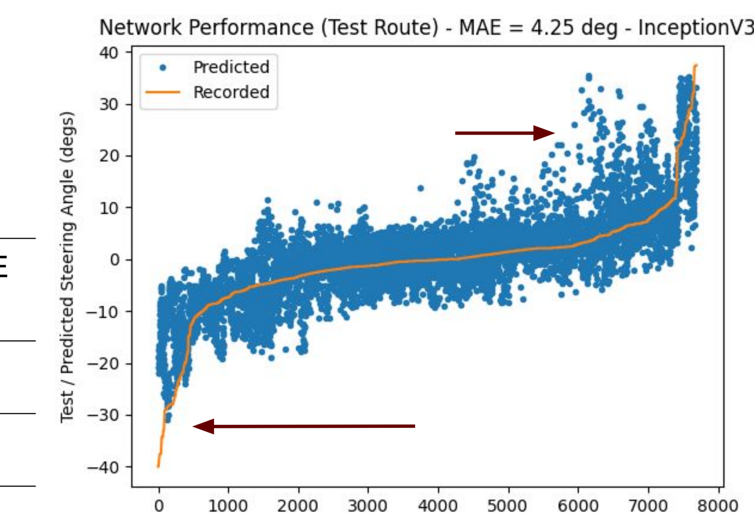
Model	Train MAE	Validation MAE	Test MAE	Test Route MAE
Batch Size 2	3.27 deg	3.45 deg	3.44 deg	4.16 deg
Batch Size 4	2.69 deg	2.83 deg	2.82 deg	4.09 deg
Batch Size 6	2.76 deg	2.95 deg	2.91 deg	4.14 deg
Batch Size 8	**2.51 deg**	**2.64 deg**	**2.67 deg**	**3.82 deg**
Batch Size 16	2.70 deg	2.81 deg	2.81 deg	3.93 deg
Batch Size 32	2.59 deg	2.76 deg	2.73 deg	4.36 deg

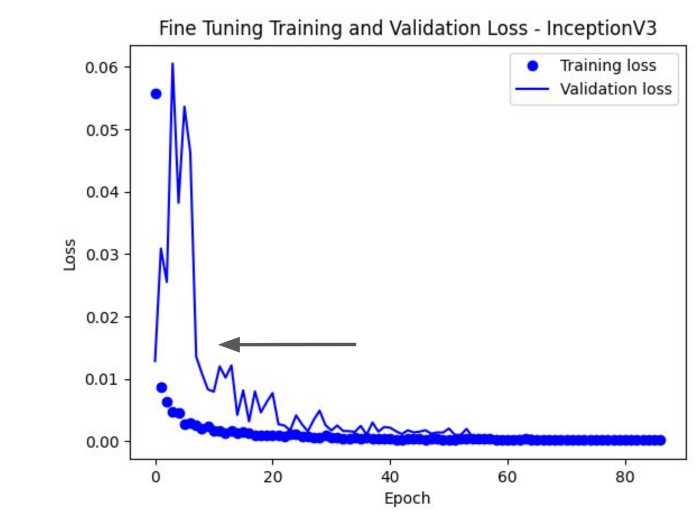
Figure 3: MAE of Fall '22 and Spring '23 models

Performance Set	Fall '22 MAE	**Spring '23 MAE**
Train	2.96 deg	**2.51 deg**
Validation	3.10 deg	**2.64 deg**
Test	3.10 deg	**2.67 deg**
Test Route	4.25 deg	**3.82 deg**

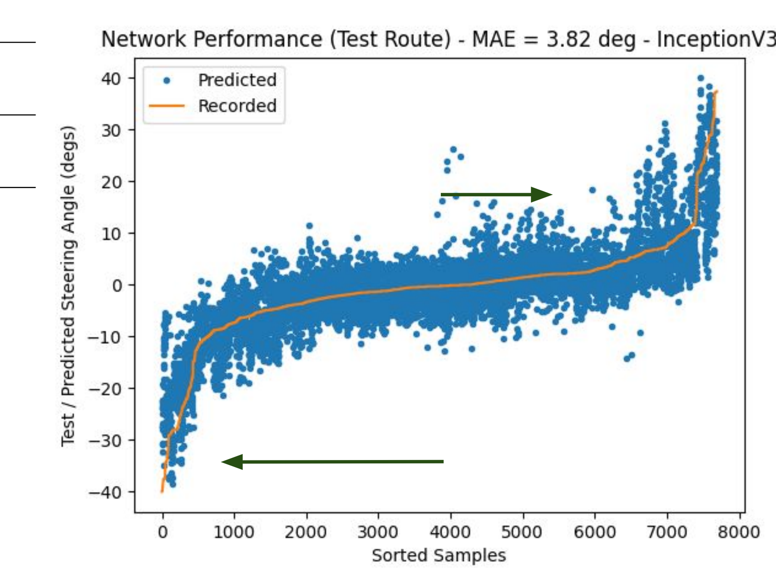
Figure 4: Test Course MAE

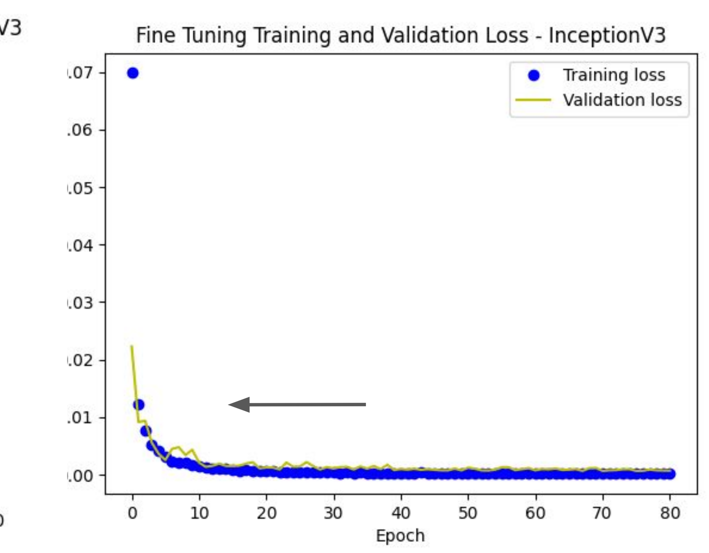
Figure 5: Train and Validation Loss





Fall '22 model





Spring '23 model

The first iteration involved training the models by skipping *N* images from the training dataset. It can be seen in the Figure 1 that the model trained by skipping every 5 images performed the best. The second iteration built on the best model of the first iteration and focused on training the models by changing the batch size used. It can be seen in Figure 2 that the model trained using a batch size of 8 performed the best. Comparing the results from Figure 3, the new model performed best in all performance sets with a final test route MAE of 3.82 degrees. When examining Figure 4, the new model was able to predict steering angles +-30 degrees and beyond more accurately than the old model. As well, there are less outliers in the curve compared to the old model, showing there was less false predictions. In Figure 5, the validation loss improved significantly during training as shown by how the curve fits better with the training loss. The final models achieved a Mean Absolute Error (MAE) values between 3.82 deg and 4.36 deg, on an unseen dataset. The solution successfully navigated the unseen route with satisfactory road-following behavior [5].

DISCUSSION

Adding a RNN layer to the previous research significantly improved the accuracy of the vehicle's ability to self-steer on unmarked roads. As suspected, predicting the steering wheel angle with the previous prediction in context stabilizes the system. In addition, the new architecture allows better predictions at high angles.

Further research:

- Use adaptive optimizers
- RNN architecture optimization
- Calculate the slope between steering angle predictions to determine if the RNN layer improves the steering returnability

REFERENCES

1. Giuseppe DeRose Jr., Austin Ramsey, and Chan-Jin Chung, Teaching Cars to Self-Steer on Unmarked Roads with Deep Learning, submitted to IEEE IROS 2023
2. Mark Kocherovsky, Guiseppe DeRose, Nicholas Paul, Ian Timmis, Chan-Jin Chung, Autonomous Vehicle Steering through Convolutional and Recurrent Deep Learning, Autonomous Vehicles and Systems: A Technological and Social Perspective, River Publishers (To be printed in 2023)
3. Mark Kocherovsky, Giuseppe DeRose, Nick Paul, CJ Chung, DeepSteer: Autonomous Driving through Convolutional and Recurrent Learning, April 20, 2022, Research Day Poster, Lawrence Technological University, http://qbx6.ltu.edu/chung/papers/43_Kocherovsky_DeepSteer.pdf
4. Ian Timmis, Nicholas Paul, Chan-Jin Chung, Teaching Vehicles to Steer Themselves with Deep Learning, 2021 IEEE International Conference on Electro/Information Technology, May 14 - 15, 2021, Central Michigan University, Mount Pleasant, MI
5. (2023). *YouTube*. Retrieved April 16, 2023, from https://youtu.be/yGqs5YsmQiU.