Trail Terrain Detection using Machine Learning

Anthony Carr, Brice Brosig, Cameron Donner, Ephraim Jackson, Alexandra Triampol
University of North Texas

Abstract

Saddleye is a company focused on producing solutions to save bicyclists from future biking accidents. They have worked on a product that uses machine learning to create a computer vision model that can recognize objects such as cars, trail boundaries, and terrain classification.

Spare Parts has worked on the terrain classification feature by visiting different trails all around Texas, creating a tool that labels all trail data, and training a machine learning model that learns the difference between gravel, asphalt, sidewalk, and off-trail terrains.

Spare Parts

Majors: Computer Science
Graduation: May 2020

Members:
- Anthony Carr
- Brice Brosig
- Cameron Donner
- Ephraim Jackson
- Alexandra Triampol

Sponsor:
- Ed Pichon

Data Extraction and Labeling Tools

The trail labelling tool in **Fig. 1** is written in C++ using Qt Creator. It takes MP4 files and extracts JPEGs at a user-defined frames per second. The tool dynamically stores the JPEGs in terrain specified folders and renames the files based on the file name and an ascending number i.e. ***filename_1.jpeg***. All frames extracted through this process become test data for the model.

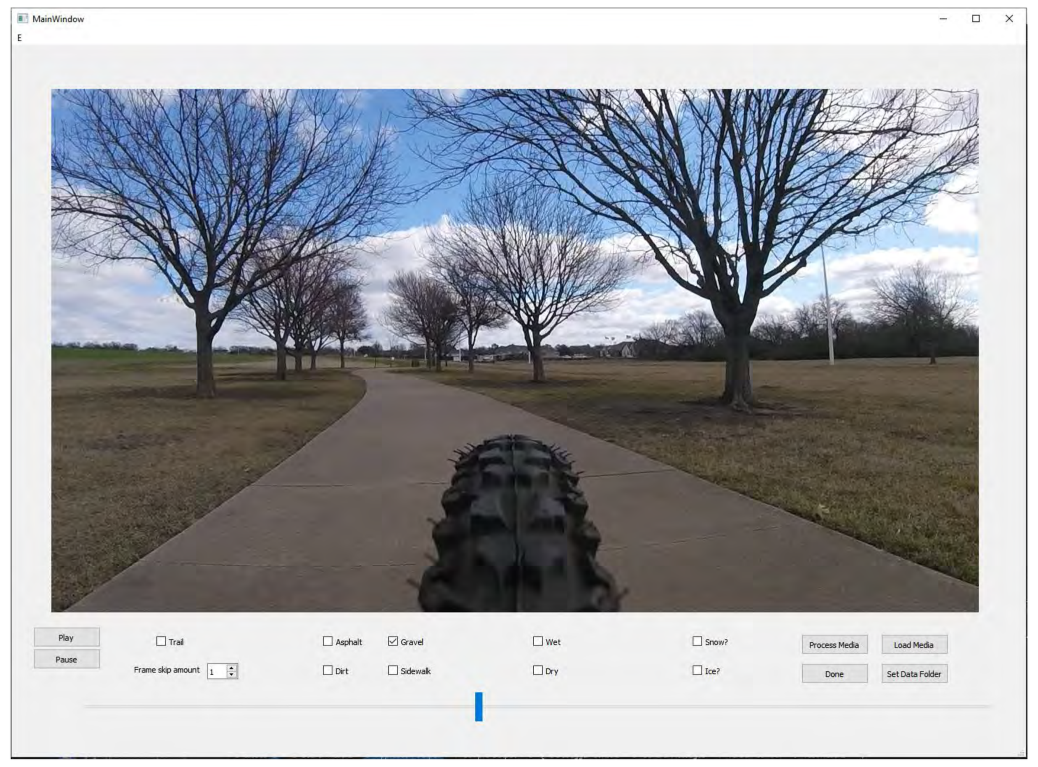

Fig. 1: Trail Labelling Tool

Terrain Classifying Model

The model written in Python uses the Keras library from TensorFlow detects terrain using a three-layer convolutional neural network (CNN) as represented in **Fig. 2**. These neural networks are great for image classification as they allow for more details of an image to be extracted versus a classifier that only takes pixels as input. A kernel or window of varying size is slid across the image and captures features from the image. This provides edge detection and other important details to be learned by the network. The size of each kernel, the number of filters, and the different types of kernels of each layer are tweaked as the model learns to classify terrain based on the given test data set. As the model learns to detect different terrains, it outputs the accuracy of its classification, allowing the user to determine if the model needs to add or remove epochs or iterations for the model to process through.

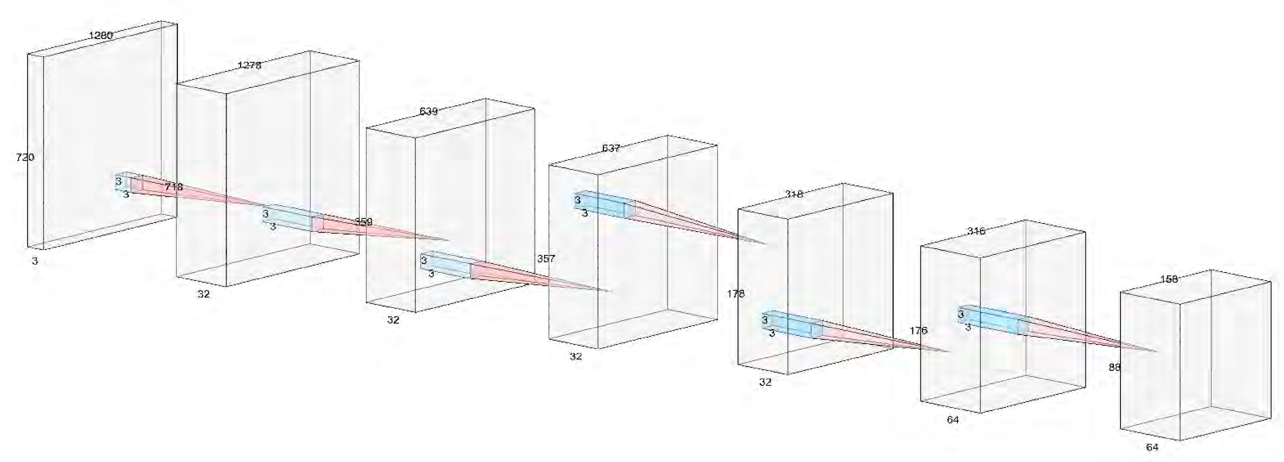

Fig. 2: Convolutional Neural Network (CNN) Used to Train Model

Fig. 3: Bean Device Used to Collect Video Data

Setbacks and Solutions

The first bicycle ride using the Bean Device (**Fig. 3**) gave the team unusable data. As depicted in **Fig. 4**, the device would droop, causing the frame to mostly capture the wheel instead of the trail. We modified the Bean case so that it's more stabilized to the bike.

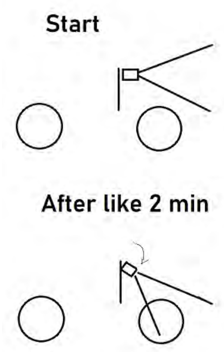

Fig. 4: Bean Device Droop

Cloud data storage became an issue due to being charged on upload and download requests and storage size. All data is now stored on OneDrive for Business because it provides 1TB of storage.

Conclusions

The model classifies the trails with a 28% accuracy. Any errors can be attributed to over fitting or under fitting data. The model was underfitting terrains because the data was not varied enough. A balanced training data set should fix this.

Future goals for the project include classifying wet and dry terrain. This would be a separately trained model which would use infrared images.

Sponsor Contact Information

Ed Pichon
Saddleye
ed.pichon@saddleye.com
http://saddleye.com/
(469) 733-4273

Spare Parts Contact Information

Anthony Car – AnthonyCarr@my.unt.edu
Brice Brosig – BriceBrosig@my.unt.edu
Cameron Donner – CameronDonner@my.unt.edu
Ephraim Jackson – EphraimJackson@my.unt.edu
Alexandra Triampol – AlexandraTriampol@my.unt.edu

Acknowledgements

We'd like to thank and acknowledge Ed Pichon, our sponsor, for providing our team with equipment to collect trail data. We'd also like to thank him for taking the time to come out to our meetings every Wednesday on campus to discuss our project.

© Poster Template by Genigraphics® 1.800.790.4001 www.genigraphics.com