Democratizing
Autonomous Driving

Cloudy Liu

Brandon Griffin

Anthony Clemens

Mission Statement:

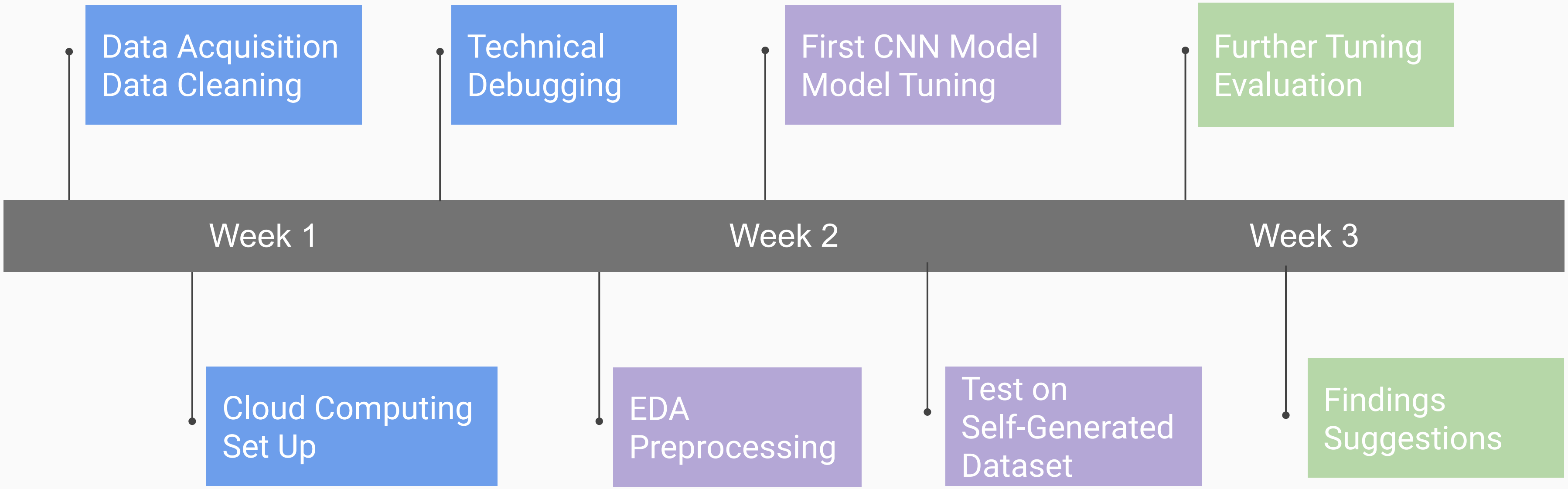
Break monopoly/oligopoly.
Encourage more companies to join the field
by building a self-driving car model from scratch.

Scenarios Considered

Can we...

1. train a model using CNN to make the car to finish a lap?
2. use a smaller dataset to achieve the similar result?
3. break through the hardware limitations?
4. optimize the model to drive more like a human, rather than swerve in the lane?
5. optimize the model to drive as fast as possible while still being able to stay in the lane?

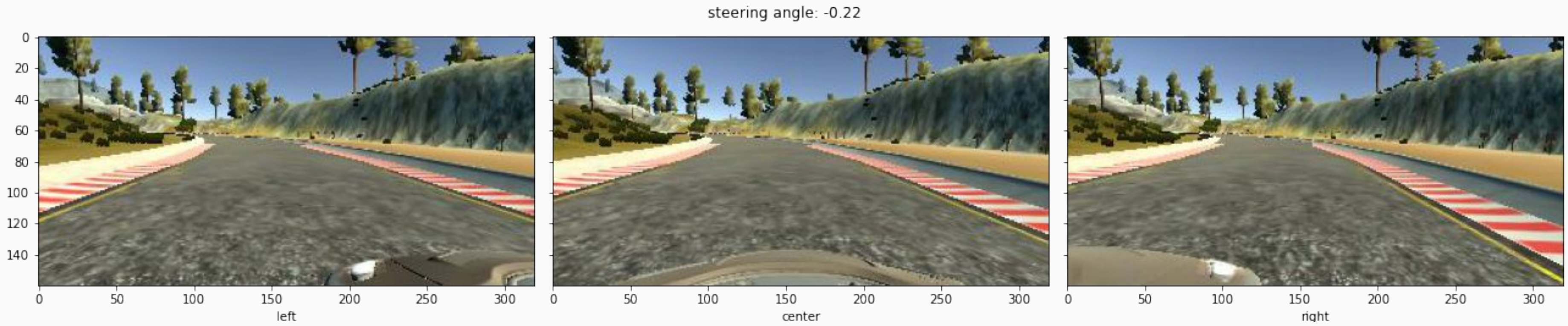
The Process


Data Acquisition
Data Cleaning
Technical
Debugging
First CNN Model
Model Tuning
Further Tuning
Evaluation
Week 1
Week 2
Week 3
Cloud Computing
Set Up
EDA
Preprocessing
Test on
Self-Generated
Dataset
Findings
Suggestions

Data Collection

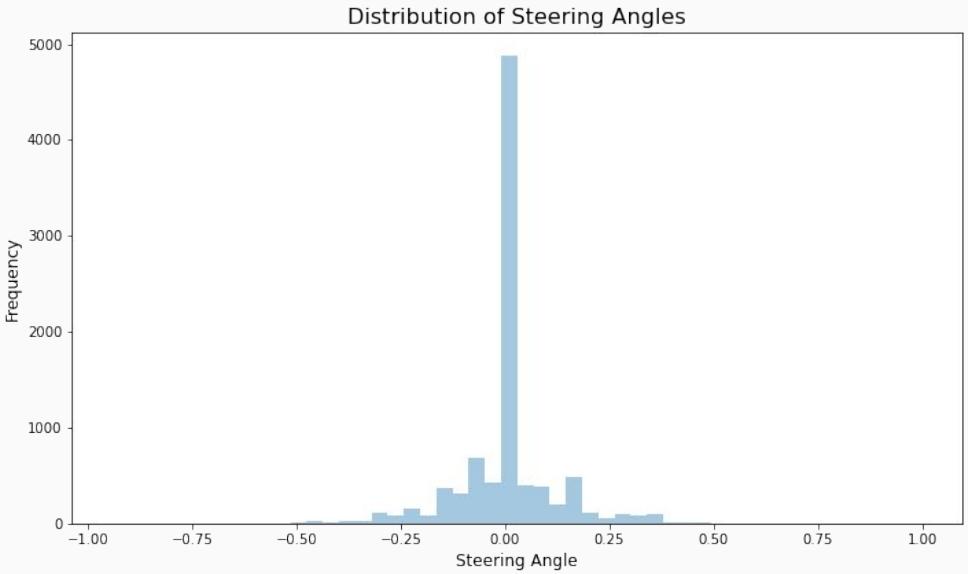
- Source: Udacity driving simulator
- Dataset used:
  - Udacity-released driving data (big scale)
  - Self-generated data (smaller scale)

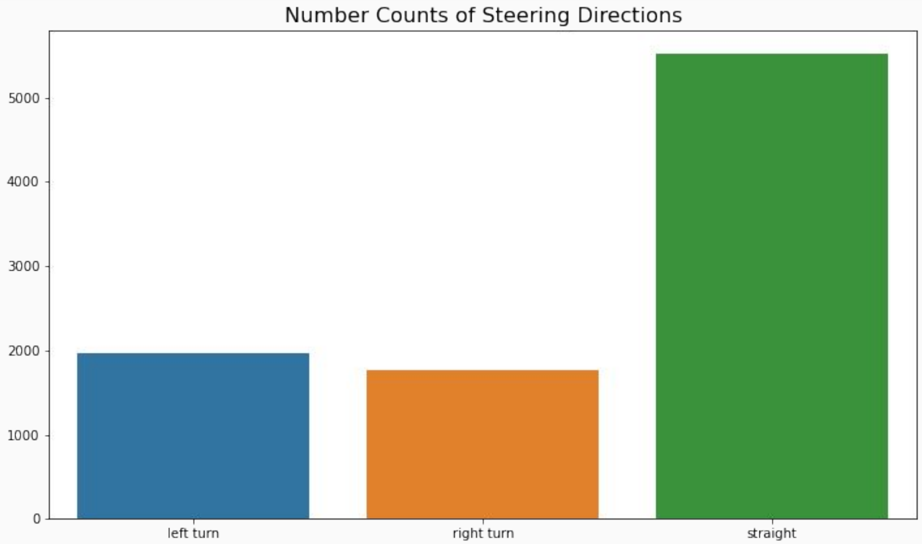
Exploratory Data Analysis



- Input: Center camera image
- Output: Steering Angle
- Assistance: Left and right camera images, using corrections

Exploratory Data Analysis - cont.





- Unbalanced data: calls for image preprocessing & augmentations

Image Preprocessing

- Before being fed into the neural network each image requires preprocessing
    - Cropping
    - Resizing
    - Converting RGB to YUV color space
- Purpose: to increase the efficiency of input data

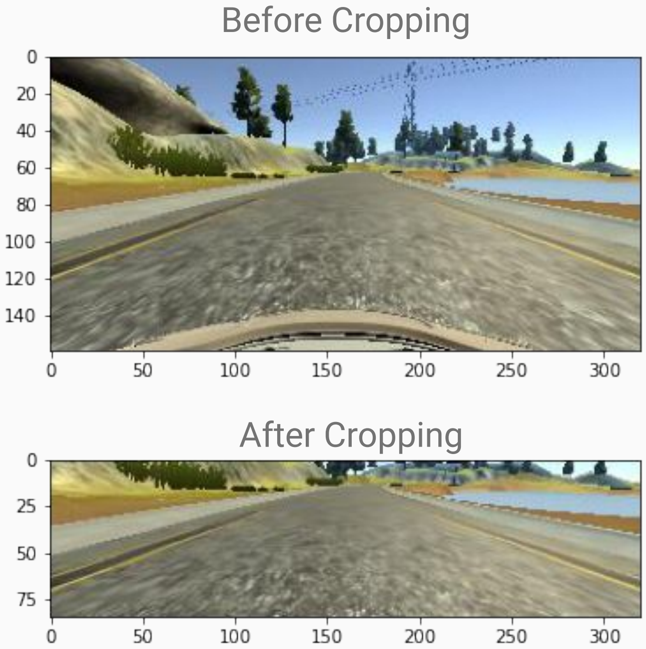
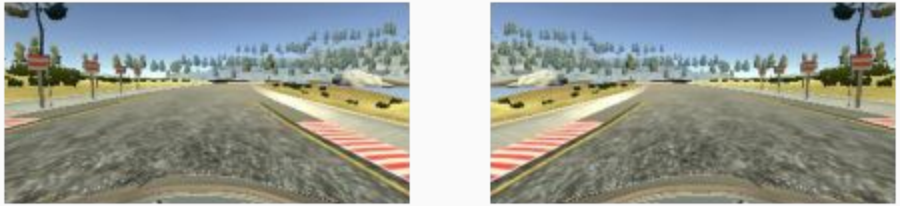


Image Augmentation

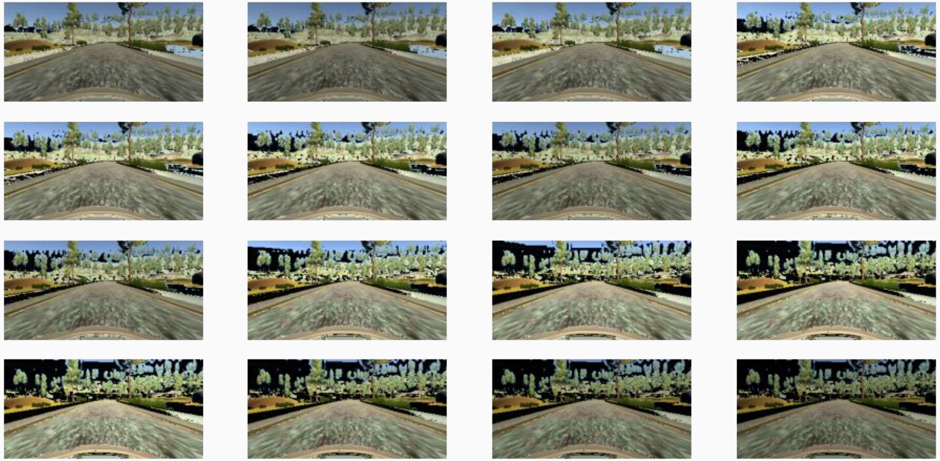
- Images are augmented ~60% of the time, including:
  - random flip
  - random translation
  - random shadow
  - random brightness
- Purpose: artificially introducing variance into the training, combat overfit



Example of random flip transformation on images

Image Augmentation

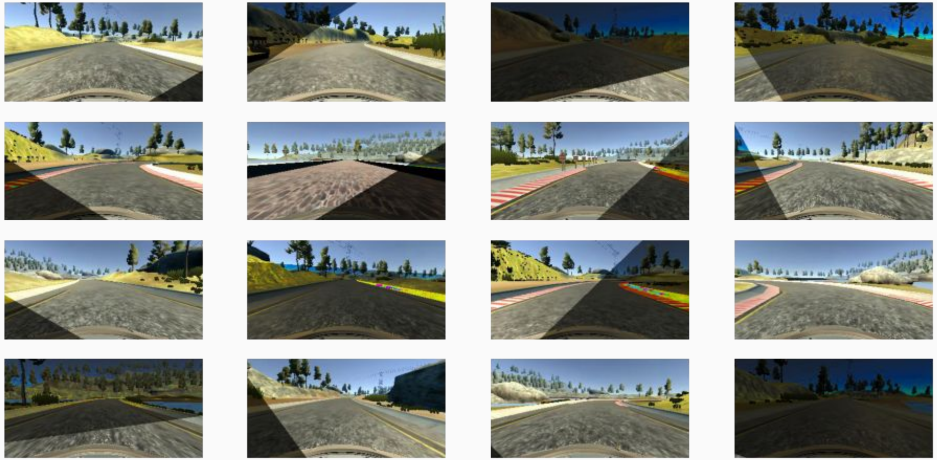
- Images are augmented ~60% of the time, including:
  - random flip
  - random translation
  - random shadow
  - random brightness
- Purpose: artificially introducing variance into the training, combat overfit



Example of random brightness transformation on images

Image Augmentation

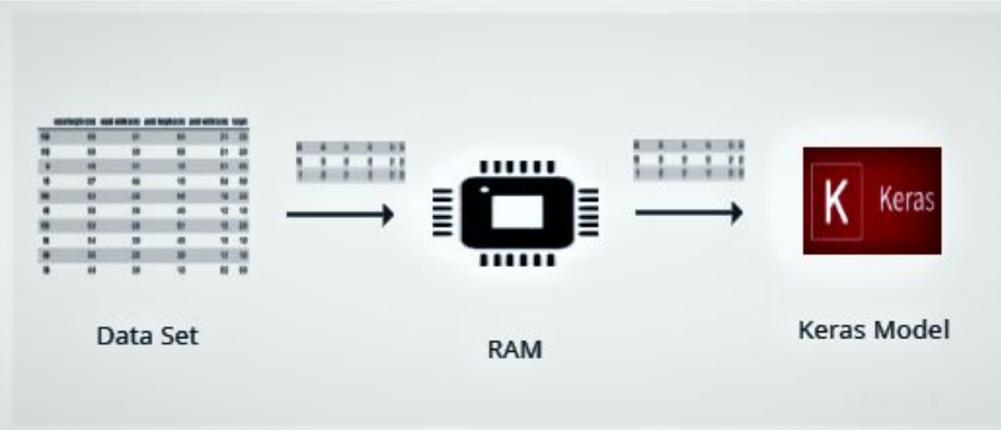
- Images are augmented ~60% of the time, including:
    - random flip
    - random translation
    - random shadow
    - random brightness
- Purpose: artificially introducing variance into the training, combat overfit



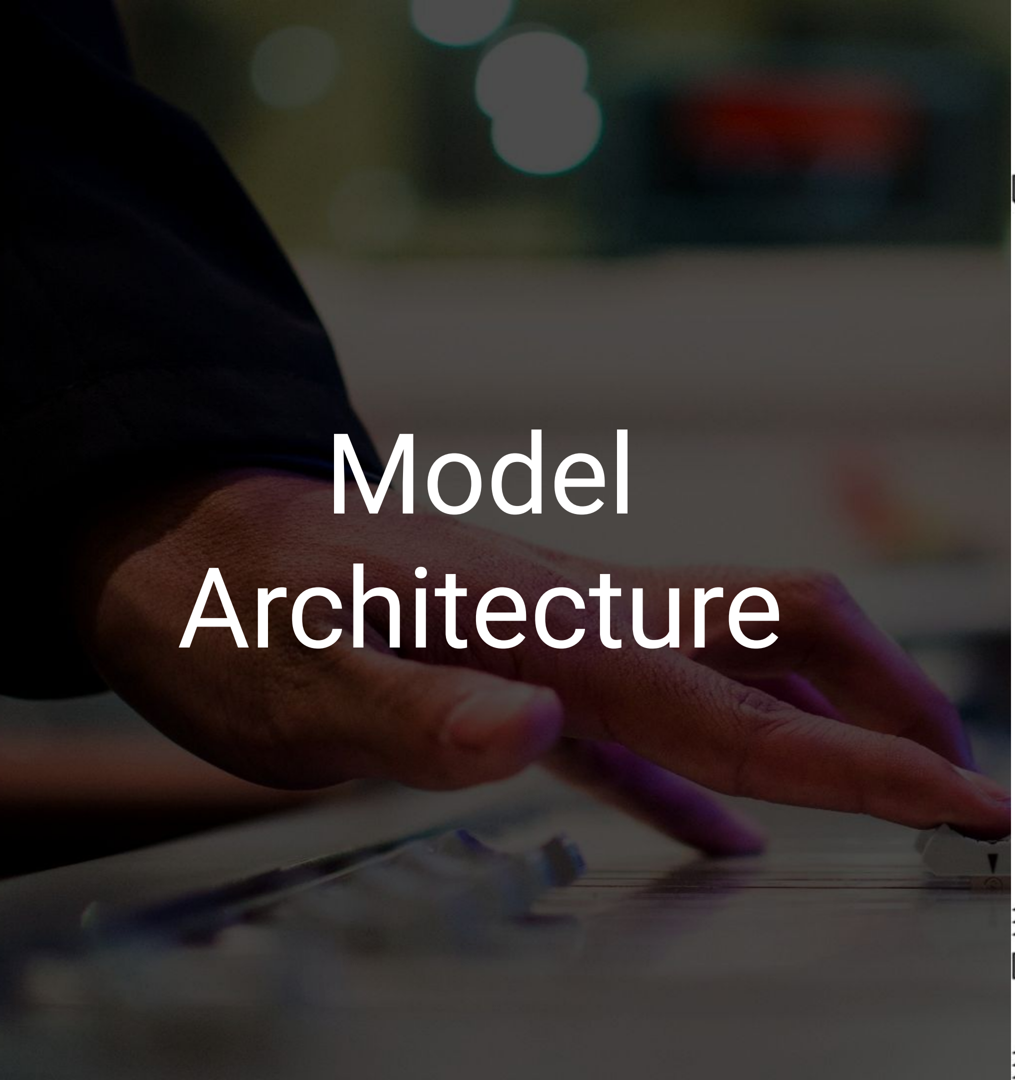
Example of random shadow transformation on images

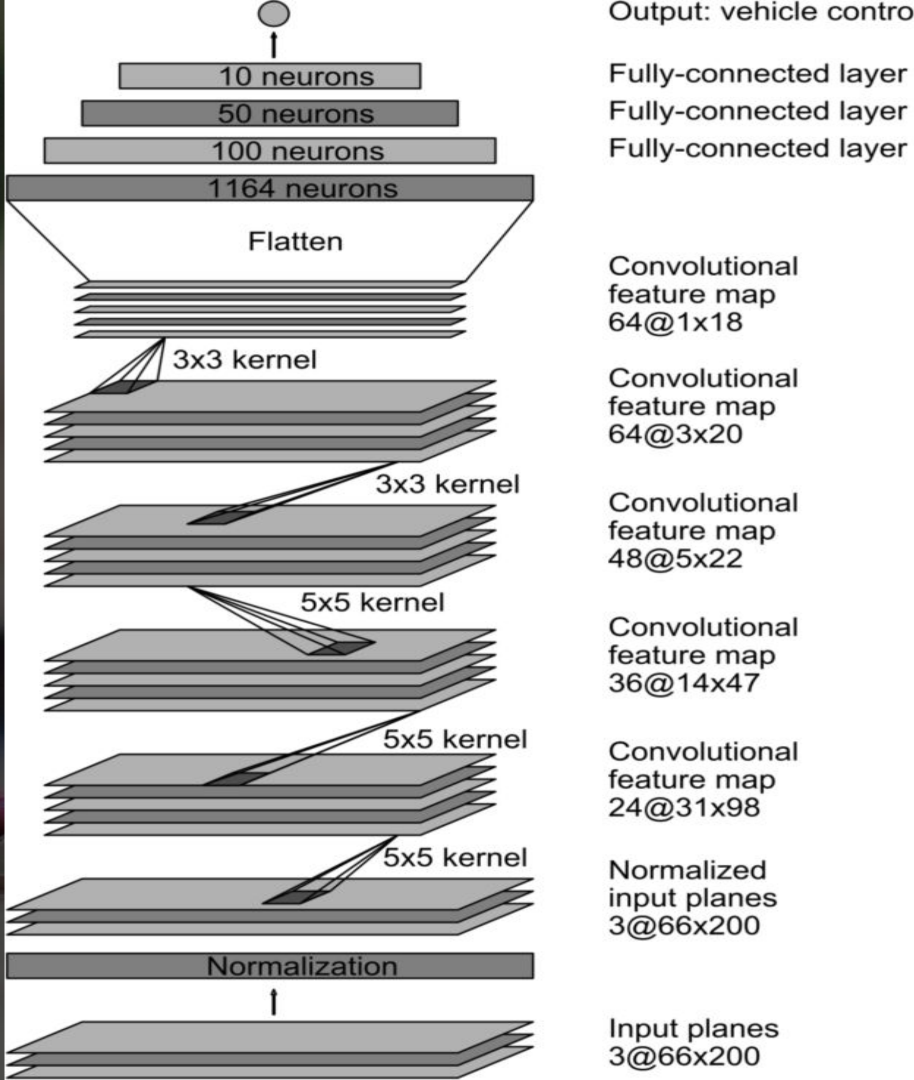
Generator

- Generator allows us to partition data into bite size chunks that are able to be loaded into memory.



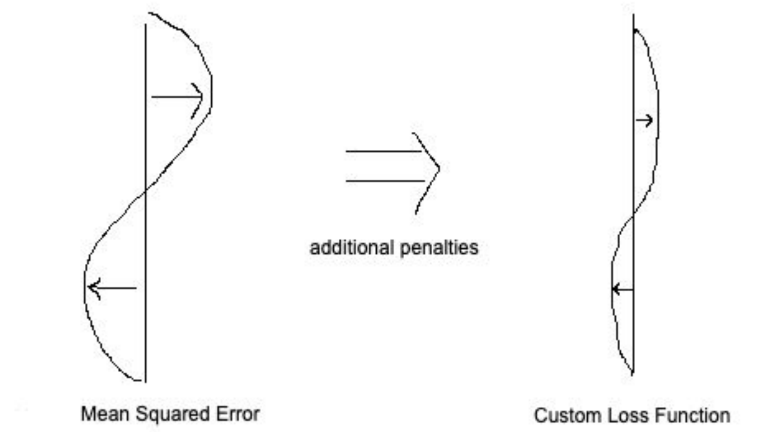
Model Architecture





Custom Loss Function

- Other models such as MSE, and Huber, featured a lot of swerving
- **Idea:** add additional penalties to the cost function in an attempt to change the behavior of the car and make it drive more human-like.



SHOW TIME (with speed limit)

Loss: HUBER

Able to finish the lap in 1 min 52s

CUSTOM

Lasted 1 min before crash (⅔ of the track)

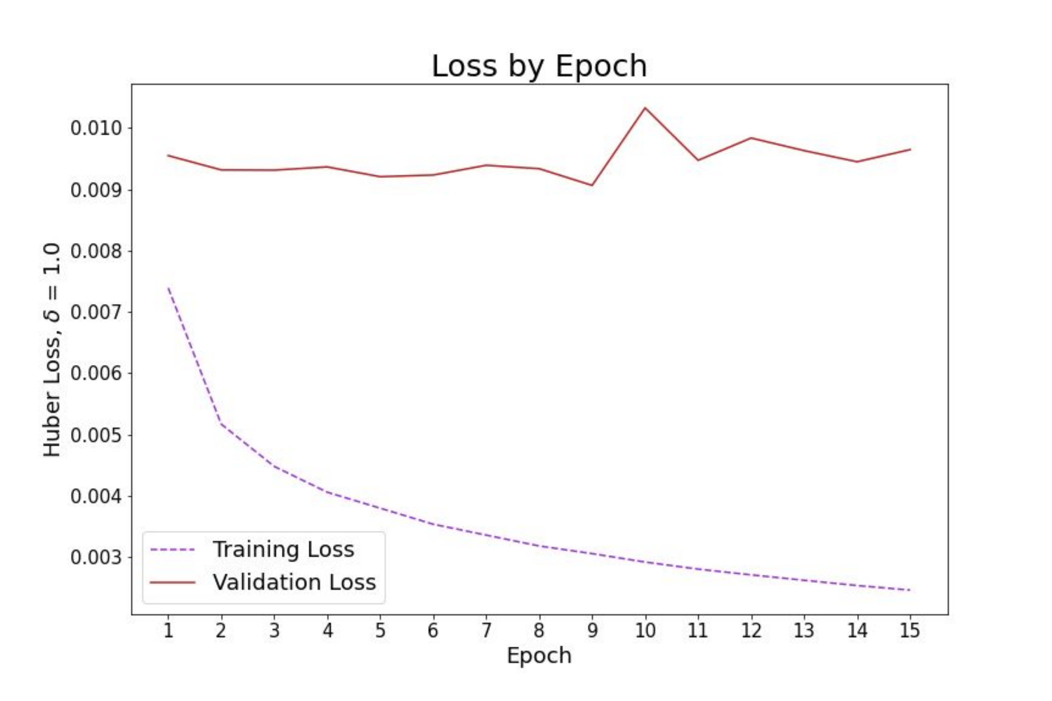
SHOW TIME (without speed limit)

Loss: HUBER | CUSTOM

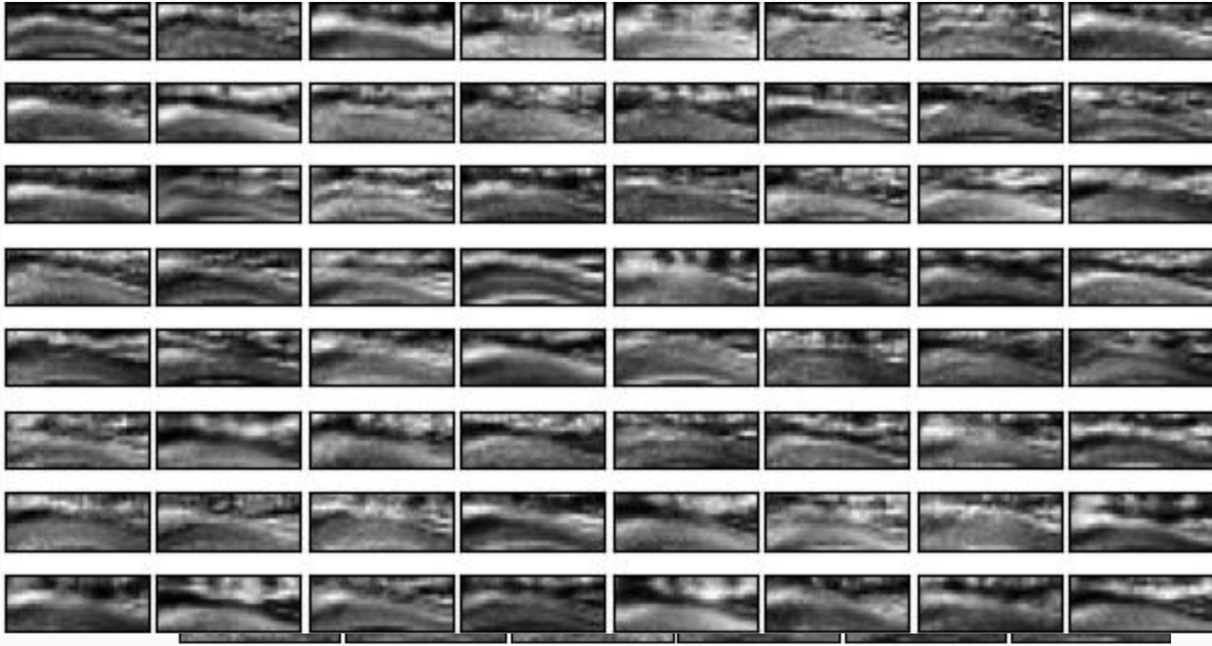
Lasted 9s before crash

Lasted 45s before crash
(⅔ of the track)

Model Evaluation



Model Evaluation



Conclusion

- Automated detection of important road features
    - Successful at maintaining lane without lane marks
- Single lap completion with speed limit enforced
    - Less than 30 mins of steering data sufficient for training the model
- Capital expenditure on Research & Development is reduced
    - Accelerate workloads: Entry level workstation GPU < $350
    - Offload work entirely: cloud services < $75 to run for 40 hours
- Model optimized to drive more “human-like”
    - Single lap completion at high-speeds not observed with current training parameters

Recommendations

- Mid-sized car companies:
  - Don’t feel intimidated to compete in auto-tech market!
- Companies with long-hauls & fixed routes:
  - Postal service, trucking industry, ground transportation
- Driving co-pilot for all:
  - Inexpensive accident averting detection device
  - Reduce fatigued-driving related fatalities

Next Steps

Step 1

Increase the epochs of training to evaluate the model performance on a different track

Step 2

Further augment the images to include different weather conditions

Step 3

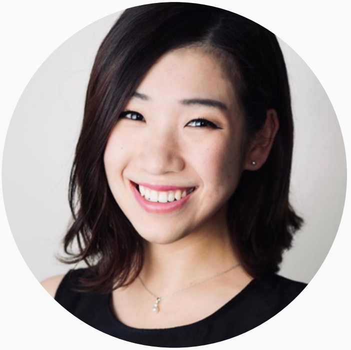
Take advantage of pre-trained models and experiment with more complex road conditions (city driving, change lanes, parking, etc)

Limitations

- Due to simplistic nature of design, model’s applications are best suited for predictable road conditions
  - Remote highway ✔️
  - Rush hour traffic ⛔
- Long train times:

  ~40 min per epoch

THANK YOU



Cloudy Liu

Brandon Griffin

Anthony Clemens