Steering Angle Prediction for Autonomous Driving Based on Image Recognition

Shuyang Du[1], Haoli Guo[2], and Andrew Simpson[3]

Background

Self-driving vehicles have expanded dramatically over the last few years. A lot of Artificial Intelligence technologies have been applied to autonomous driving, such as CNN and RNN methods, VAE and GAN models, deep reinforcement learning, etc. Udacity has an ongoing challenge to create an open source self-driving car. In the second challenge Udacity released a dataset of images taken while driving along with the corresponding steering angle and ancillary sensor data for a training set. The goal is to build a model that, given an image taken while driving, minimizes the RMSE (root mean squared error) between the predicted steering angle and the actual one produced by a human driver.

Problem Statement

In this project, we investigated CNN and RNN models that take video frames as input and output predicted steering angles. We explore a variety of structures including 3D CNNs, RNNs using LSTM, ResNets, etc. to output predicted steering angles in numerical values. We also produced saliency maps to obtain a better understanding of the features in the model. The performance of the models is evaluated by the RMSE between predicted steering angles and actual steering angles provided in the test data set.

Datasets

- The dataset we used is provided by Udacity, which is generated by NVIDIAs DAVE-2 System. Three cameras (left, center, right) are mounted behind the windshield of the data-acquisition car. Time-stamped video from the cameras is captured simultaneously with the steering angle applied by the human driver. The videos were taken in various light, traffic and driving conditions.
- **Training data set** contains 101397 frames and corresponding labels including steering angle, torque and speed. We split this data set into training and validation in a 80/20 fashion. **Test set** contains 5615 frames. The original resolution of the image is 640*480.

Fig. 1. Training data set. Typical images for different light, traffic and driving conditions. (a) Direct sunlight, (b) shadow, (c) sharp left turn, (d) uphill, (e) straight, (f) heavy traffic

Models

We developed two types of model structures. The first one uses 3D convolutional layers (no pretrained 3D convolutional model available). The second one uses 2D pretrained convolutional layers from transfer learning.

- 3D Convolutional Model

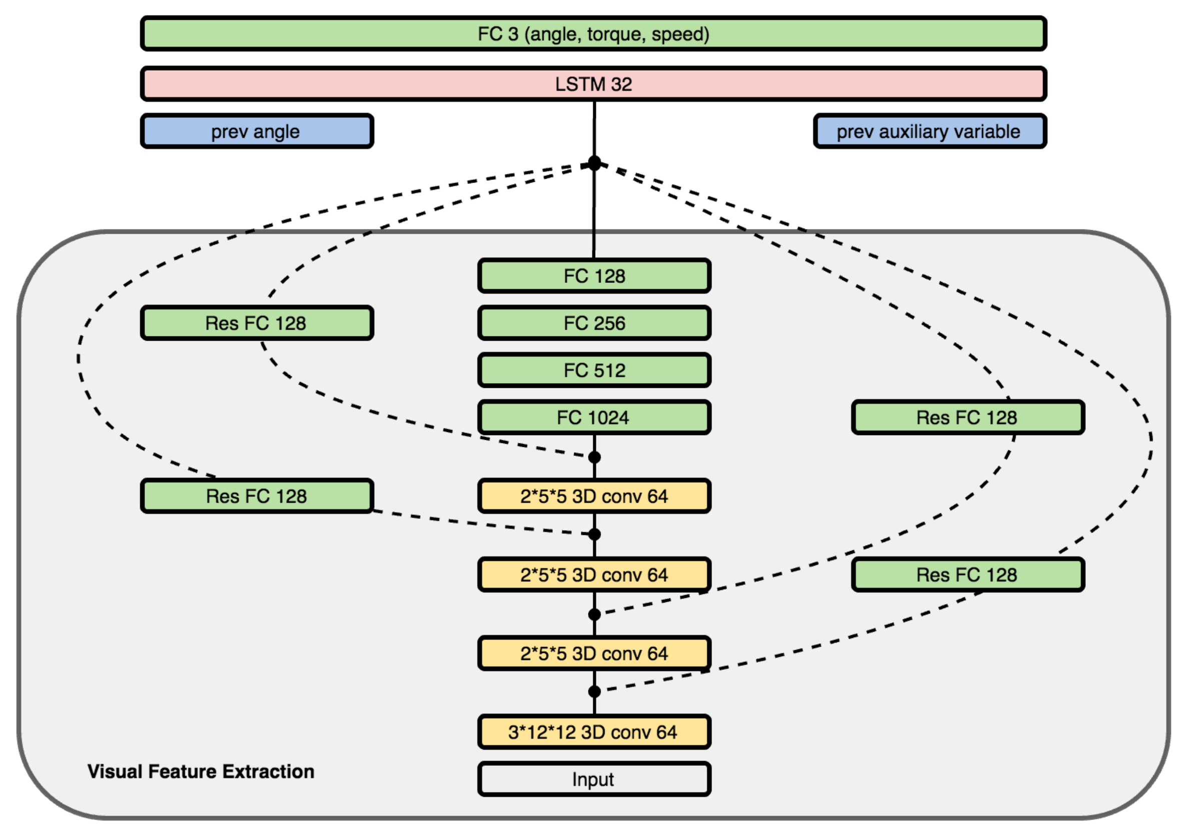

Fig. 2. 3D convolutional model architecture

(1) 3D filter. (2) Residual Connection. (3) Spatial Batch Normalization. (4) Layer Normalization

- Transfer Learning

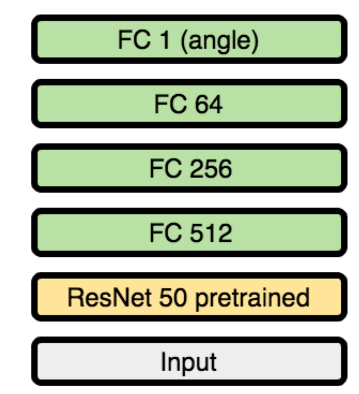

Fig. 3. 2D pretrained convolutional model from transfer learning

It uses pretrained ResNet 50 to extract visual features and processes those features with stacked fully connected layers.

Experiments and Findings

- Data Augmentation

Table 1. Summary of data augmentation

	No augmentation	Small augmentation	Heavy augmentation
Cropping and flips	✓	✓	✓
Angle shift	✗	Small change	Large change
Shadow	✗	✗	✓
Validation error	**0.09**	**0.10**	**0.19**

- No augmentation produces the best results.
- Cropping and flips help.
- Augmentation could be more important in smaller datasets or for more epochs.
- Dataset had a lot of variation may account for augmentation not being very useful.

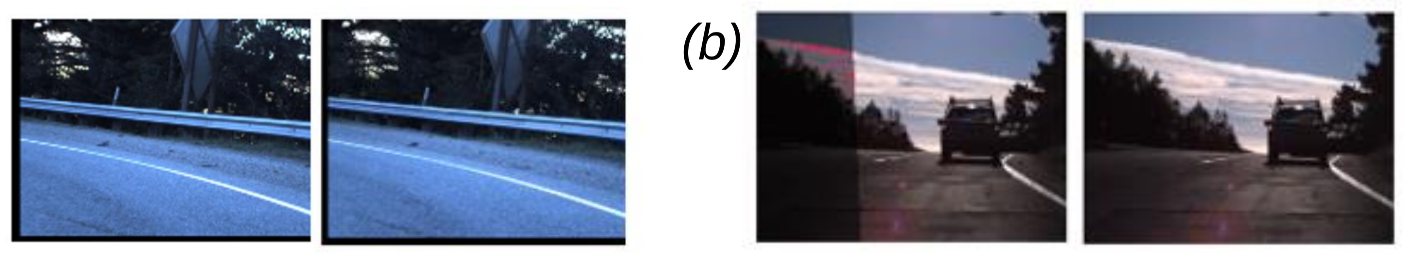

Fig. 4. Examples of data augmentation. (a) Angle shift, (b) shadow augmentation

Results and Discussion

- Training Process

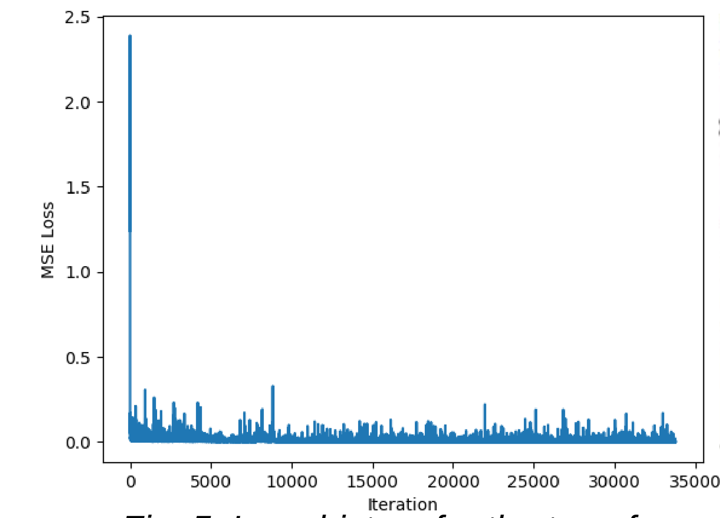

Fig. 5. Loss history for the transfer learning model.

- Prediction Visualization

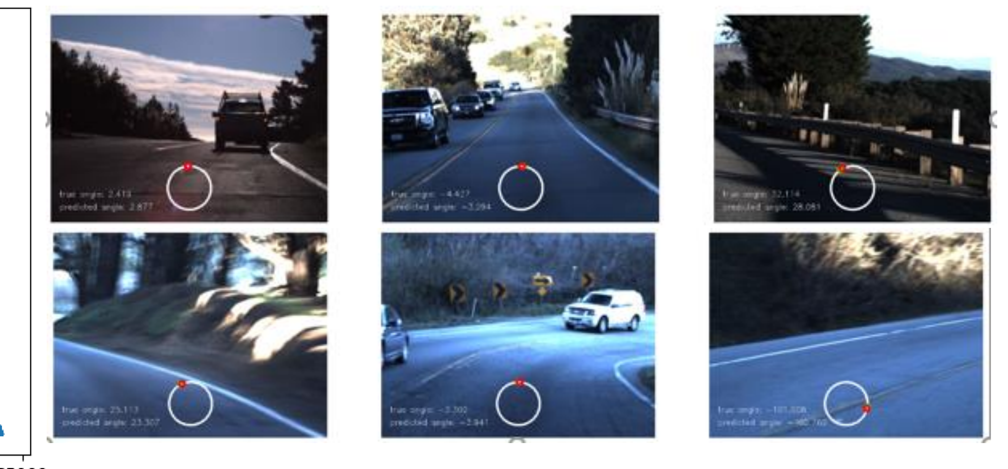

Fig. 6. Prediction visualization of steering angles. The green circle indicates the true angle and the red circle indicates the predicted one.

- Test Results

Table 2. Test results compared with baseline model by NVIDIA (2016)

	3D Convolutional Model	2D by Transfer Learning	NVIDIA Model (2016) Baseline
Test error (RMSE)	0.1123	0.0709	0.0986
Leader board	10th place	4th place	--

- Feature Visualization

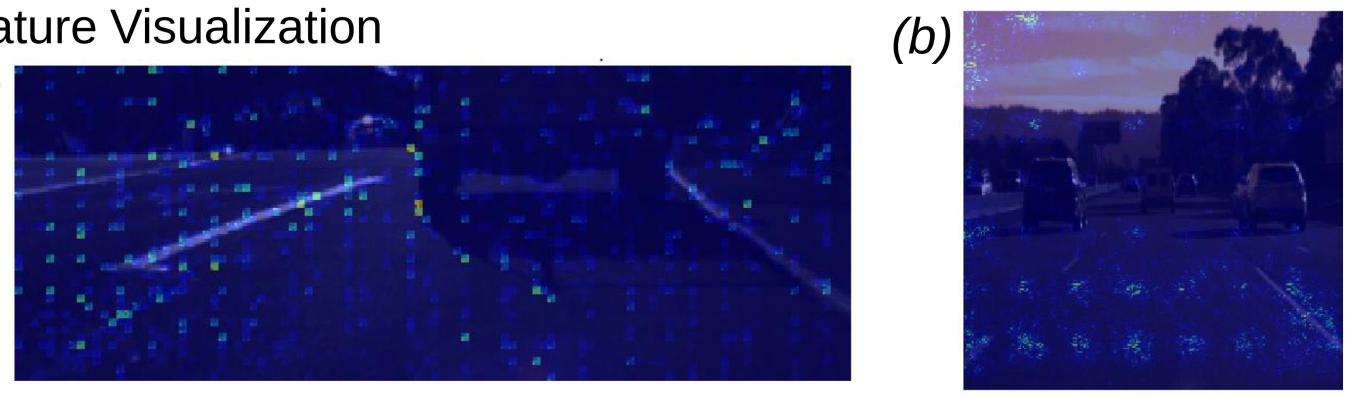

Fig. 7. Saliency maps for (a) 3D convolutional model with LSTM and (b) Transfer model

Conclusions and Future Work

- Conclusions
  - Our 2 models are competitive with others.
  - Transfer learning with a model trained on ImageNet is very effective on this dataset. Our result outperforms NVIDIA's model.
  - 3D convolutional layers with LSTM layers shows promise in using temporal information.
- Future work
  - Use a smoothing function after the output, training longer, better augmentation, etc.
  - Use deep reinforcement methods with a simulator.
  - Use GANs for more training data and winter or other weather scenes.

- Reference: M. Bojarski et al. arXivpreprint arXiv:1604.07316, 2016.

1. shuyangd@stanford.edu 2. haoliguo@stanford.edu 3. asimpso8@stanford.edu