Driving Scene Segmentation: Lane Detection For Autonomous Car

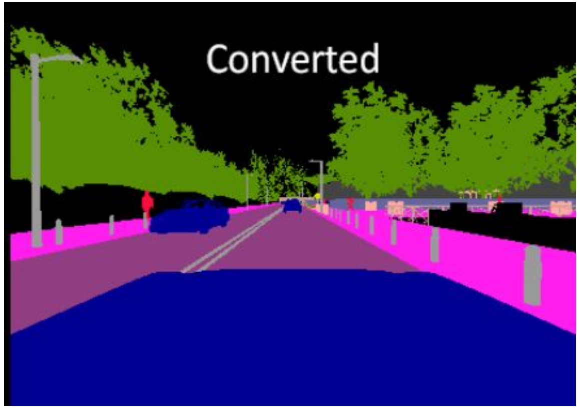
Team 6: Tejas Mahale, Chaoran Chen, Wenhui Zhang

Introduction

The goal of this project is to conduct lane detection in case of quick steering scenario of AutoCars.

Dataset

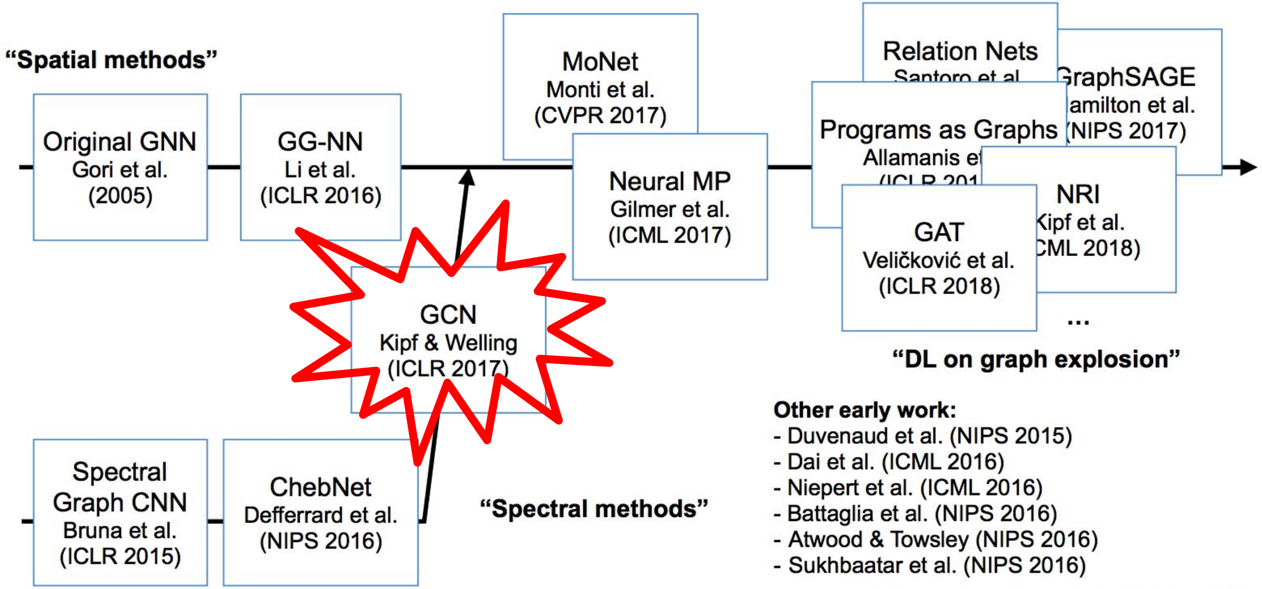
- Carla simulator dataset
  - open-source simulator for autonomous driving research
  - 3,000 images along with semantic labels with resolution 600 x 800 x 3



Dataset

- Carla simulator dataset
  - Drive a car in urban conditions to get train images and corresponding encoded label images
  - Almost all types of weather conditions and different time of day for the images.

Related Works on Structured DL: Deep learning on graphs etc.



Other early work:
- Duvenaud et al. (NIPS 2015)
- Dai et al. (ICML 2016)
- Niepert et al. (ICML 2016)
- Battaglia et al. (NIPS 2016)
- Atwood & Towsley (NIPS 2016)
- Sukhbaatar et al. (NIPS 2016)

(slide inspired by Alexander Gaunt's talk on GNNs)

Challenges

1) Classification: object associated to a specific semantic concept
→ use label map with GCN;
2) Localization: classification label for a pixel not aligned with score map
→use camera to world space projection;
3) Noisy Gradient Prediction:
→ use Batch Norm and Boundary Refinement, with optimization method of Adam.

Data Preprocessing

- Cropping car hood and sky portion

600 x 800 x 3

360 x 800 x 3

Data Preprocessing

- Decodify

R	G	B	int24	
00000000	00000000	00000000	0	min (near)
11111111	11111111	11111111	16777215	max (far)

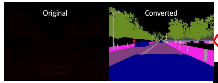
1.To decodify our depth first we get the int24. `R + G*256 + B*256*256`

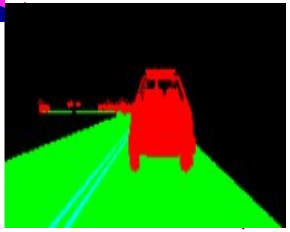
2.Then normalize it in the range [0, 1]. `Ans / ( 256*256*256 - 1 )`

3.And finally multiply for the units that we want to get. We have set the far plane at 1000 metres. `Ans * far`

Data Preprocessing

- Encoded image to Segmented image
  - Every pixel of label image is red channel class label
  - Retained class labels with road and vehicle and converted them to green and blue respectively





Value	Tag
0	None
1	Buildings
2	Fences
3	Other
4	Pedestrians
5	Poles
6	RoadLines
7	Roads
8	Sidewalks
9	Vegetation
10	Vehicles
11	Walls
12	TrafficSigns

GCN and Adam

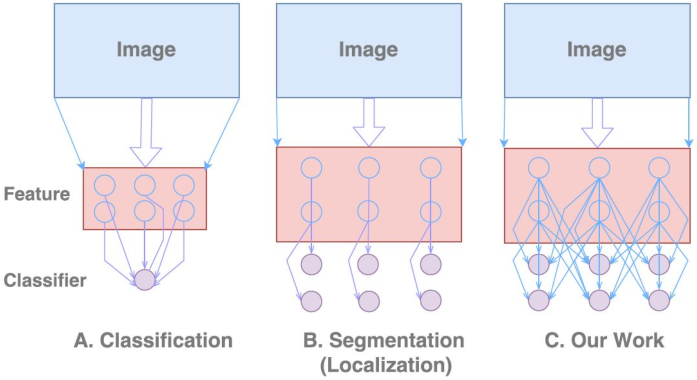
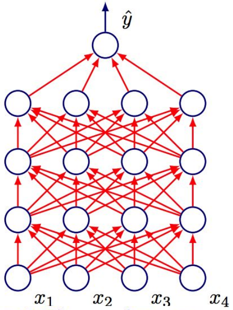


Fig 1: Classification network; B: Conventional segmentation network, mainly designed for localization; C: Our Global Convolutional Network.



Require: ϵ (set to 0.0001), decay rates ρ_1 (set to 0.9), ρ_2 (set to 0.9), θ, δ

Initialize moments variables $\mathbf{s} = 0$ and $\mathbf{r} = 0$, time step t $= 0$

while stopping criteria not met **do**

  Sample example $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ from training set

  Compute gradient estimate: $\hat{\mathbf{g}} \leftarrow +\nabla_\theta L(f(\mathbf{x}^{(i)};\theta), \mathbf{y}^{(i)})$

  $t \leftarrow t + 1$

  Update: $\mathbf{s} \leftarrow \rho_1 \mathbf{s} + (1 - \rho_1)\hat{\mathbf{g}}$

  Update: $\mathbf{r} \leftarrow \rho_2 \mathbf{r} + (1 - \rho_2)\hat{\mathbf{g}} \odot \hat{\mathbf{g}}$

  Correct Biases: $\hat{\mathbf{s}} \leftarrow \frac{\mathbf{s}}{1-\rho_1^t}, \hat{\mathbf{r}} \leftarrow \frac{\mathbf{r}}{1-\rho_2^t}$

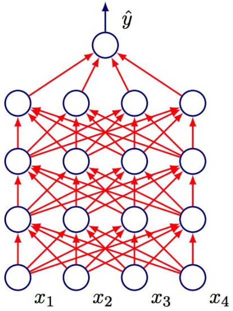
  Compute Update: $\Delta\theta = -\epsilon \frac{\hat{\mathbf{s}}}{\sqrt{\hat{\mathbf{r}}} + \delta}$

  Apply Update: $\theta \leftarrow \theta + \Delta\theta$

end while

Fig 2: ADAptive Moments

Batch Norm



Reparameterize a deep network to reduce co-ordination of update across layers

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \dots & h_{1k} \\ h_{21} & h_{22} & h_{23} & \dots & h_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & h_{m3} & \dots & h_{mk} \end{bmatrix}$$

Let H be a design matrix having activations in any layer for m examples in the mini-batch

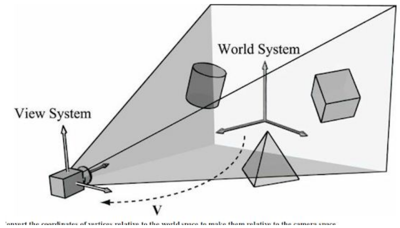
$$\mu = \frac{1}{m}\sum_j H_{:,j}$$

$$\sigma = \sqrt{\delta + \frac{1}{m}\sum_j (H - \mu)_j^2}$$

- μ is a vector with μj the column mean
- σ is a vector with σj the column standard deviation
- Hi,j is normalized by subtracting μj and dividing by σj

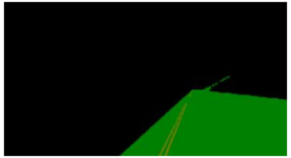
$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \dots & h_{1k} \\ h_{21} & h_{22} & h_{23} & \dots & h_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & h_{m3} & \dots & h_{mk} \end{bmatrix}$$

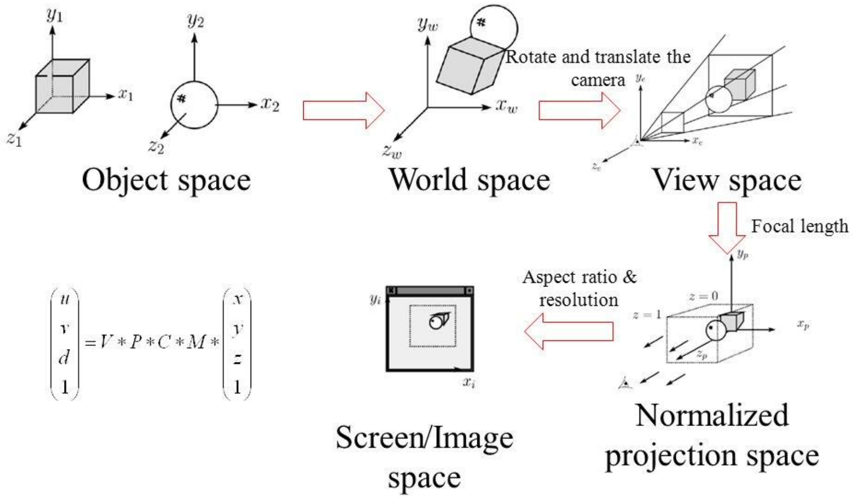
The new H allows convergence on extremely large datasets



Data Augmentation

- Rotation (0, 30 degree) and shifting (Width = (0, 0.2), height = (0, 0.1))

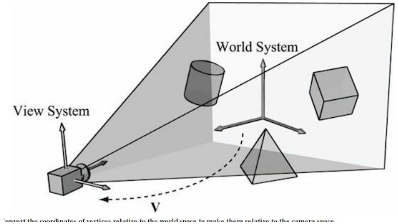


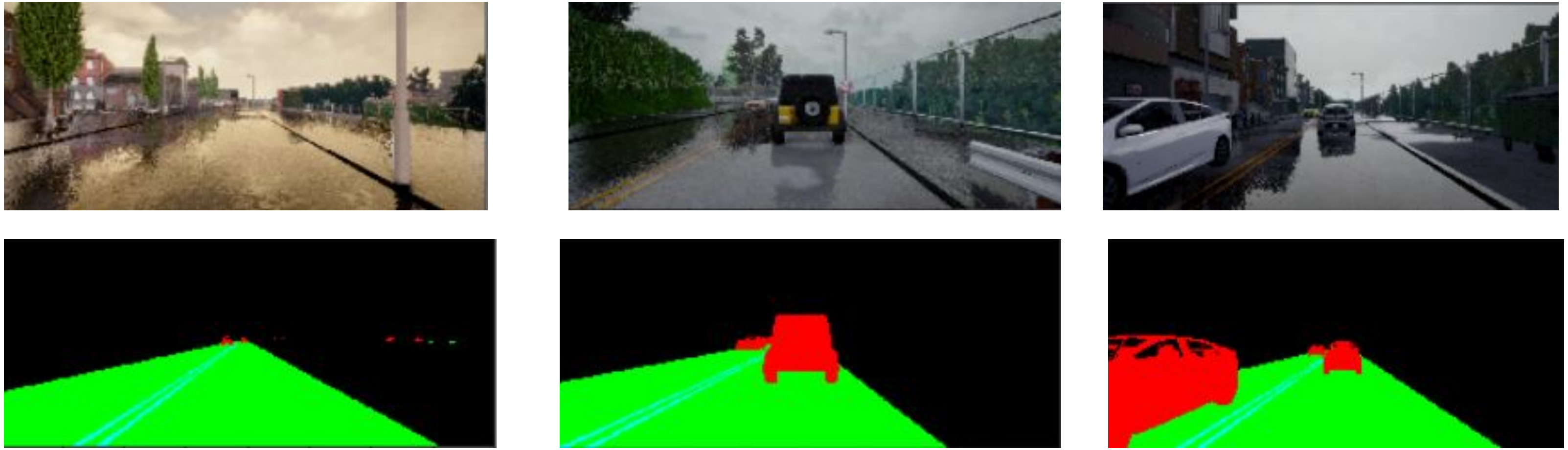


Data Distribution

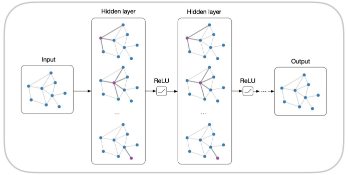
Training	**2400 images**
Validation	**300 images**
Testing	**300 images**

Data Visualisation



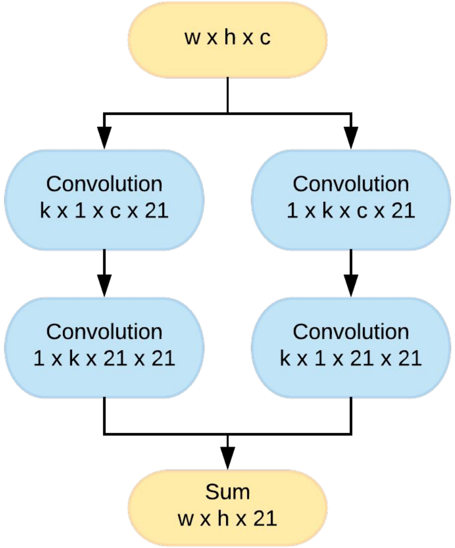


The "semantic segmentation" camera classifies every object in the view by displaying it in a different color according to the object class

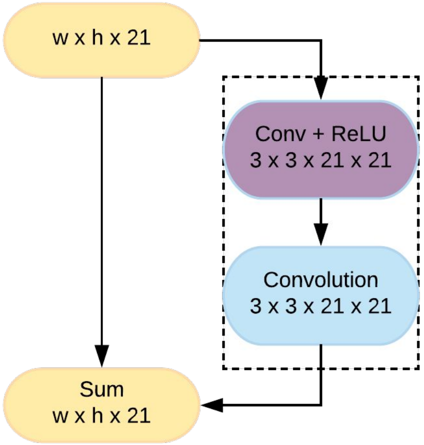


Method : GCN and Boundary Refinement

Metric = RMSE

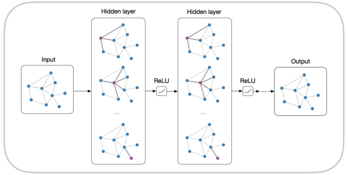


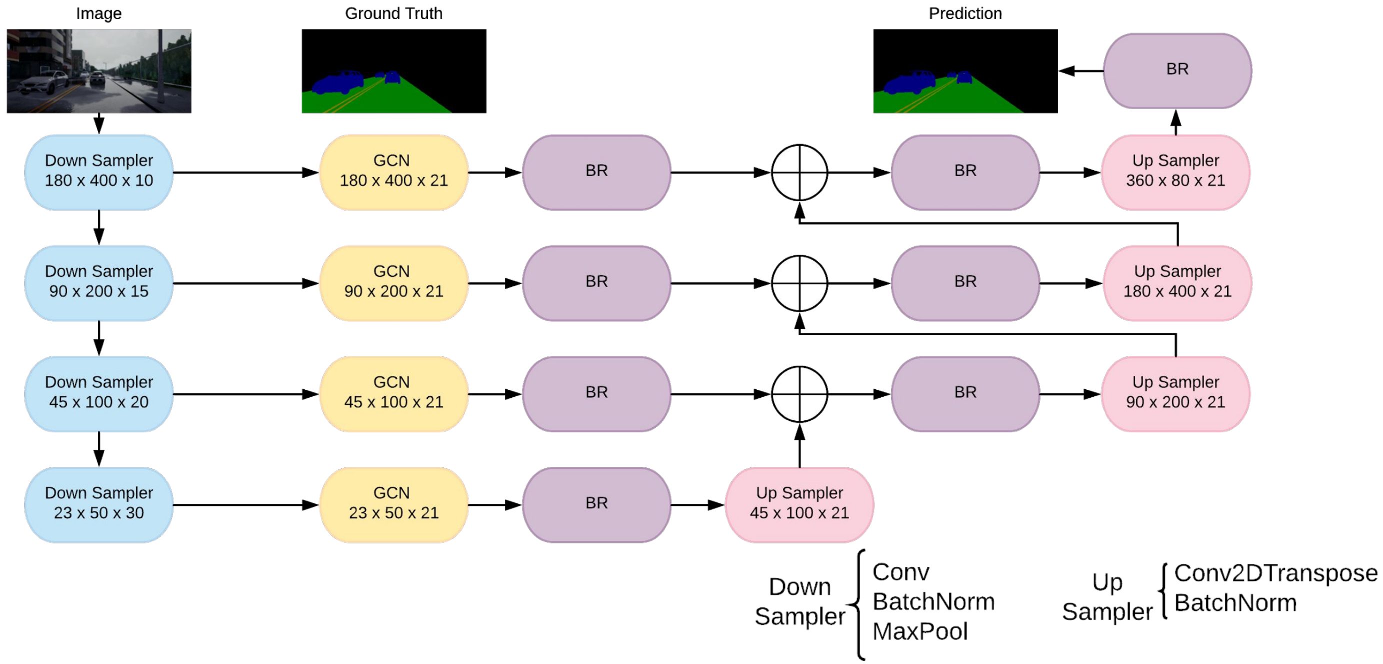
Global Convolution Network



Boundary Refinement

Architecture Model



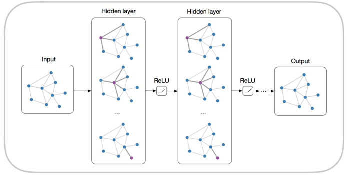


Metric

- Mean Absolute Error:
    - Absolute pixel to pixel difference between two images(in our case ground truth image and predicted image)

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|$$

- Mean Square Error:
    - $$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$



Parameters

Parameter	Value
Optimizer	Adam
Learning rate	0.001
Number of Epoch	38
Number of Filters	8, 16, 20, 32

Result

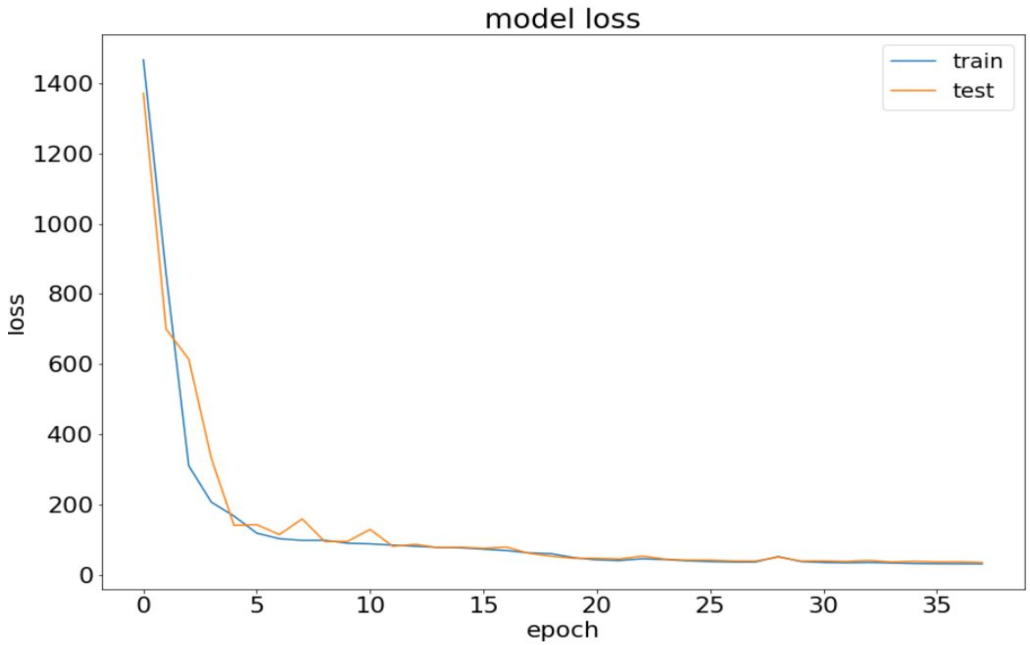
	Training		Validation	
	Minimum Square Error	**Mean Absolute Error**	**Minimum Square Error**	**Mean Absolute Error**
Initial Loss	1592.1951	13.3800	1443.7032	13.3908
10 Epoch	192.1637	3.1369	337.5560	6.8221
20 Epoch	102.3213	2.9696	119.4761	3.7389
30 Epoch	74.0532	2.0202	89.0769	2.8999
38 Epoch	42.7771	1.7011	61.4360	2.4999

Result

Results on Test data :

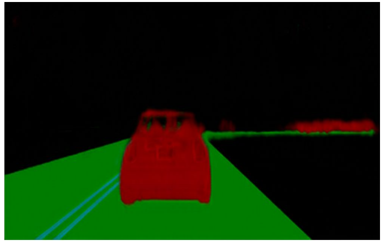
Minimum Square Error	57.5875
Mean Absolute Error	2.2104

Result


model loss
train
test
loss
1400
1200
1000
800
600
400
200
0
0
5
10
15
20
25
30
35
epoch

Results

Original Image

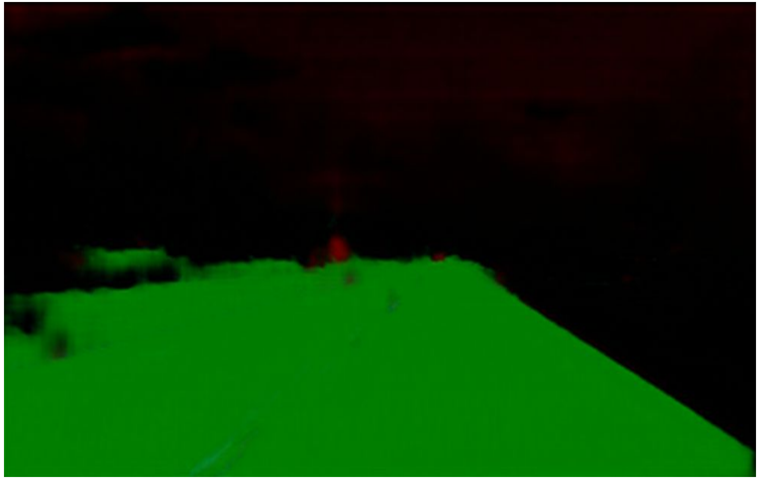


Predicted image

MSE = 33.1541

MAE = 1.1523

Result



References

[1] Cordts, Marius, et al. “The cityscapes dataset for semantic urban scene understanding.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
[2] Geiger, Andreas, et al. “Vision meets robotics: The KITTI dataset.” The International Journal of Robotics Research 32.11 (2013): 1231-1237.
[3] Wang, Panqu, et al. “Understanding convolution for semantic segmentation.” arXiv preprint arXiv:1702.08502 (2017).
[4] Ilg, Eddy, et al. “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks.” IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
[5] M. Everingham, A. S. M. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes challenge: A retrospective,” IJCV, vol. 111, iss. 1, 2014.
[6] Xu, Huazhe, et al. “End-to-end learning of driving models from large-scale video datasets.“IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
[7] Neuhold, Gerhard, et al. “The mapillary vistas dataset for semantic understanding of street scenes.” Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy. 2017.
[8] Brostow, Gabriel J., Julien Fauqueur, and Roberto Cipolla. “Semantic object classes in video: A high-definition ground truth database.” Pattern Recognition Letters 30.2 (2009): 88-97.
[9] Oh, Sangmin, et al. “A large-scale benchmark dataset for event recognition in surveillance video.” Computer vision and pattern recognition (CVPR), 2011 IEEE conference on. IEEE, 2011.
[11] Sandipan Narote, Pradnya bhujbal, “A review of recent advances in lane detection and departure warning system” , Volumn 73, Jan 18 Pattern Recognition Elsevier
[12] Ze Wang, Qiang Qiu. “LaneNet: Real-Time Lane Detection Networks for Autonomous Driving”, Computer Vision and Pattern Recognition arxiv.org
[10] Chao Peng, Gang Yu, “Large Kernel Matters-improving semantic segmentation”, CVPR 2017