Data Reconstruction for Fault Tolerance in Autonomous Driving Systems

Alanna Zoscak
Department of Computer Science
University of Virginia
Charlottesville, VA
az5dd@virginia.edu

Madellyne Waugh
Department of Computer Science
University of Virginia
Charlottesville, VA
mew8ax@virginia.edu

Matthew Walsh
Department of Computer Engineering
University of Virginia
Charlottesville, VA
mw6es@virginia.edu

I. Introduction

A large problem facing vision-based autonomous vehicles is obstructions via rain, snow, or other obstacles that decrease the quality of data received by cameras or sensors. In 2021, Tesla removed radar systems from its cars instead opting for ”Tesla Vision”, a camera-only autonomous driving system. However, this system has had issues in heavy rain or when debris obstructs the view of the cameras, leading to a frequent disengagements of the autonomous driving system. Current lane detection models such as Tesla Vision struggle with analyzing images in poor conditions thus leading to a higher risk of accidents or inaccurate route prediction. This project proposes a system using denoising autoencoder (DAE) models trained on unobstructed photos to reconstruct and remove noise from obstructed photos. The goal of this project is to increase the accuracy of lane detection for autonomous vehicles and reduce risk when navigating in poor weather conditions.

II. System Description

In order to reduce the risk of incorrect guidance in autonomous vehicles, our system is designed to reconstruct faulty images captured by cameras on the autonomous vehicle in order to increase the accuracy of lane detection. First, a denoising autoencoder will be trained using both clear images and noisy images. The noisy images have had been perturbed with Gaussian noise, changing each pixel slightly to model the obstruction caused by poor weather conditions such as rain or snow. The model will learn a mapping to encode these noisy images to a low level latent space and will then decode the images back to an output space with less noise than the input space. These denoised images will then be fed into the lane detection algorithm to identify where lane lines are and thus where the vehicle should maneuver.

Fig. 1: Overview of System Design.

III. Design and Implementation

A. Denoising Autoencoder

Figure 2 shows the architecture of a Denoising Autoencoder (DAE). An autoencoder performs an unsupervised learning task in which it attempts to learn how to efficiently reconstruct the input image. Specifically, an autoencoder consists of an encoder, which learns a mapping from the input image to a lower dimensional latent space, and a decoder, which learns a mapping from the low dimensional latent space to a final reconstructed image. An autoencoder takes in an image and outputs a reconstructed image. The goal during training is to minimize the loss between the input image and its reconstruction. The key difference between the DAE and a vanilla autoencoder is the DAE injects noise into the input at training time. Given less input information, the DAE still attempts to reconstruct the original, noise-free image. In this way it learns to perform the denoising task, and becomes more robust to faulty input images.

We implement a Convolutional Denoising Autoencoder (CDAE) to serve as a fault tolerance mechanism in front of a lane detection algorithm like would be found in an ADS. The CDAE consists of convolutional layers in its encoder and decoder, which enable more efficient computations and are able to effectively learn spatial features in data, a characteristic that makes them preferred in computer vision tasks. The loss function we use is Mean Squared Error, which performs pixel-wise subtraction of the input and reconstruction, pixel-wise squaring, and takes the average to get a final scalar loss. This loss function is used to train the model, as well as to evaluate the impact of the CDAE on the lane detection algorithm. Since all of our experiments are implemented with the CDAE architecture, we refer to the models as CDAE and DAE interchangably.

*1) **Datasets**:* One version of the DAE is trained on frames from a single ideal-case video to demonstrate feasibility of the fault tolerance technique. This video contains 1260 frames.

The second version of the DAE is trained on a subset of the CULane lane detection benchmark dataset [4]. The dataset consists of 7642 training images, 1906 validation images, and

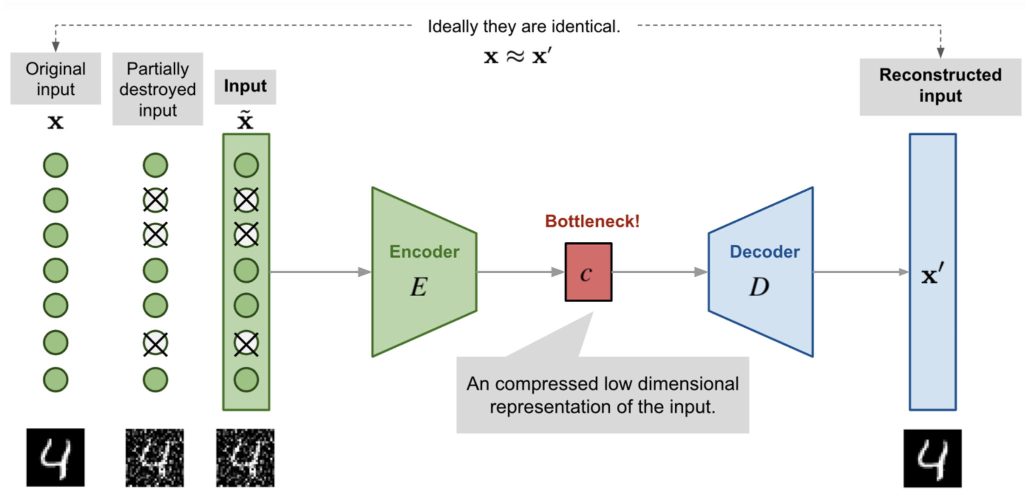


Fig. 2: Denoising Autoencoder Architecture [3]

838 test images. Working with this larger dataset involved implementing additional data loading methods.

B. Lane Detection Algorithm

We implemented a lane detection algorithm in order to test the usability of reconstructed images from the DAE. This algorithm uses color thresholds to differentiate between the road and the lane lines. First, the image is converted from RGB format to 8-bit greyscale format as shown in Figure 2.

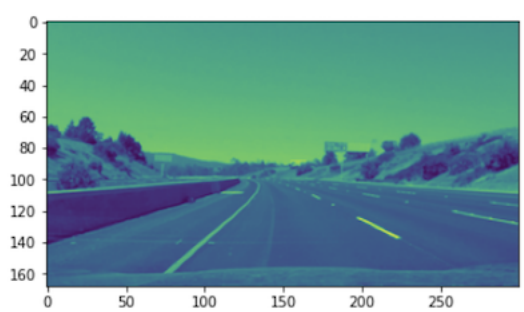


Fig. 3: Grayscale Lane Image.

Then a mask is applied to filter out and greyscale values darker than 150, which will leave all shades of white and yellow and remove darker areas such as the roadway.

To differentiate between lane lines and other white or yellow pixels such as the sky or lights, we manually defined an area in the image where we expect only to find lane lines. This area was defined as the triangle between the midpoint of the image and two lower corners, as this is where the roadway will be in the image.

Using OpenCV's Hough Line transform function, which can detect straight lines in images. We take the average of all the lines found through this transform and draw lines back onto the image to represent the lane lines we have isolated. The final result of this lane line detection algorithm can be seen in Figure 5.

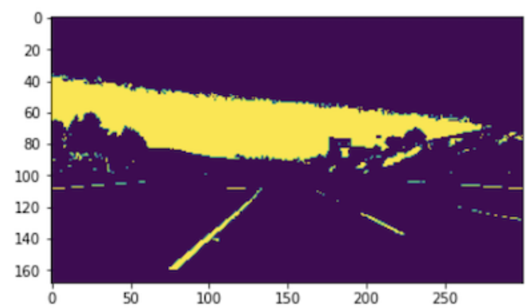


Fig. 4: White and Yellow Pixels

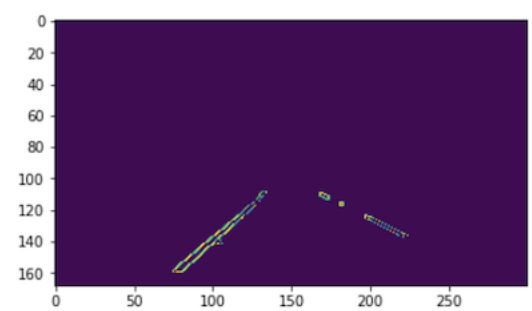


Fig. 5: Isolated Lane Lines.

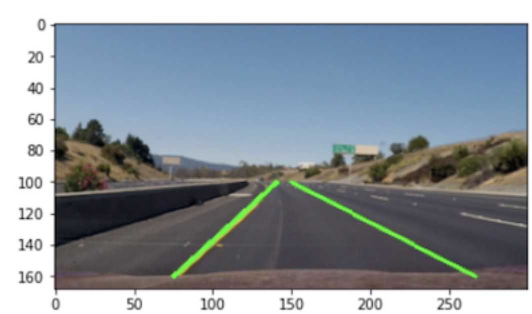


Fig. 6: Resulting Detected Lane Lines.

IV. Evaluation and Results

The evaluation of our models depends on how well we can detect lane lines after reconstructing noisy images. Therefore, in order to test the models we compared the detected lane lines in images without noise versus images reconstructed by the DAE. To measure both the ability of the model to reconstruct the image and the usability of the detected lane lines, we used our lane detection algorithm to draw lane lines on the ground truth image as well as on the reconstructed image. We also used the lane detection algorithm to attempt to draw lane lines on the noisy images as a proof of concept that the DAE is necessary. These images were then compared using mean squared error. The mean square error between actual images and noisy images was measured to be 9447.07. Between actual images and reconstructed images, error dropped drastically to 387.05.

A. Convolutional Denoising Autoencoder Trained on CULane Dataset

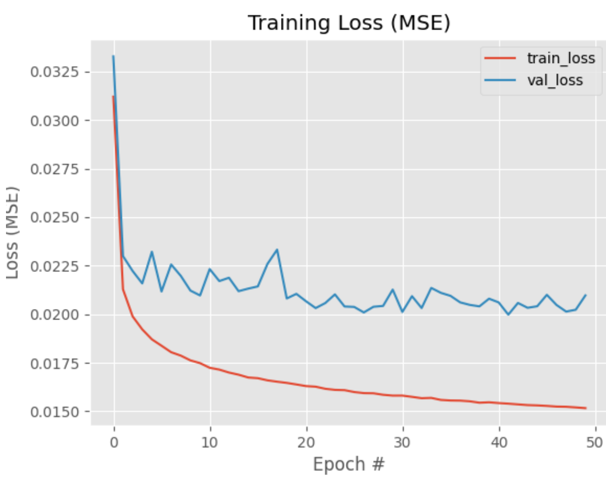


Fig. 7: Training Results for CDAE on CULane Dataset

Figure 8 shows isolated qualitative reconstruction results and test set loss for differing levels of noise in the test dataset. These results demonstrate fuzzy reconstruction results, with blurry lane lines visible for some reconstructions. The model struggles to reconstruct the right lane line, indicating that there may not have been enough of these examples in the dataset. The reconstruction also does not capture objects such as other vehicles; this is not important for the perception task of lane line detection, and actually helps extrapolate lines despite vehicles making lane changes.

The autoencoder mapping to a lower dimensional latent space enables the DAE to produce similar reconstructions for differing levels of noise, and there is no discernable trend between the level of noise added and the reconstruction loss. This suggests that a better DCAE is necessary to obtain better reconstructions, in addition to being agnostic to the noise level of the input, as observed in these results.

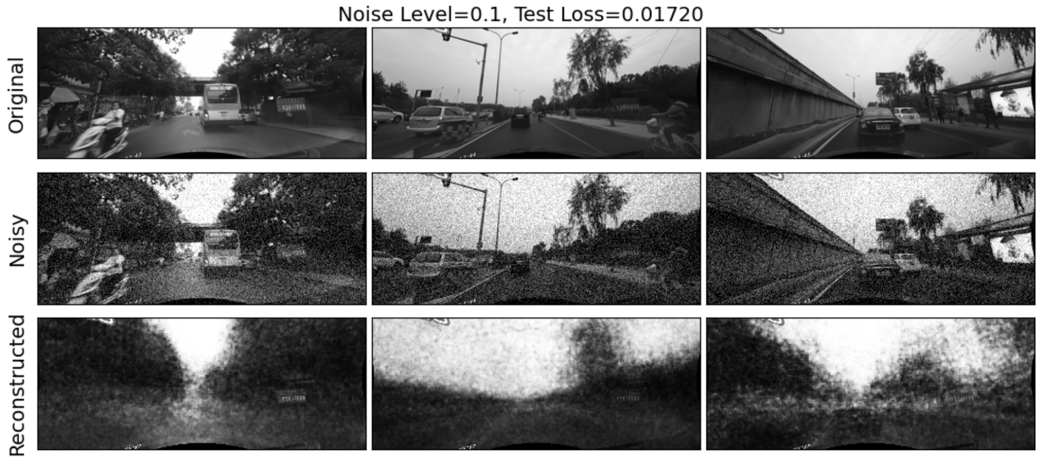
Another experiment we conducted was testing the DAE on out of distribution data, as shown in Figure 9. Specifically, we tested the model that was trained on the CULane dataset on the video frame dataset and found that it had much higher reconstruction loss. It also injected elements from the CULane dataset, such as the hood of the vehicle, into the reconstruction. This demonstrates the poor generalizability of the DAE to out of distribution data.

The CDAE was incorporated into the full pipeline, serving as a fault tolerance technique before the lane detection algorithm, and sample results are displayed in Figure 10. The overall test set reconstruction MSE, as well as the baseline MSE of the noisy data are reported in the titles of the plots. These results highlight the need for more complex lane detection techniques, as many of the clearly visible lane lines are not identified by the detection in noise-free data. For noise levels greater than 0.2, the CAE enables lane detection results that better align with the baseline performance on noisy images, indicating the potential to improve the fault tolerance of the lane detector. However, for lower noise levels, such as 0.1, the reconstructed image actually obscures the image more than the noise, which calls for better denoising models.

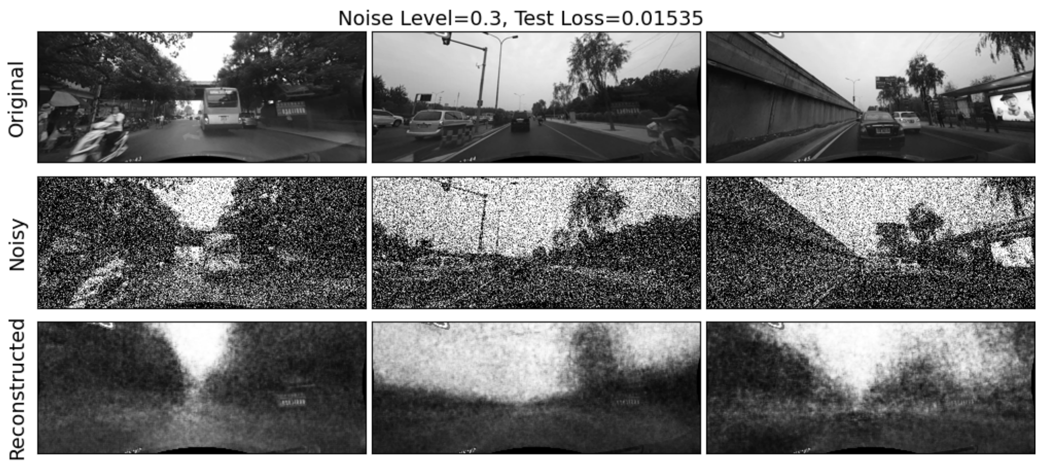
V. Lessons Learned

We learned many lessons as we worked through this project. As we were all new to denoising autoencoders, a lot of the optimizations we made were led by trial and error. Additionally, we learned about the difficulties of sourcing data for lane line detection. Though our project aims to mitigate risks associated with poor quality of input images, it was still quite hard for us to find usable data. This is due to many factors, but mainly that most data sets were completely unlabelled and included images from night time, poor weather conditions, and other unwanted images for our training data. When creating the lane detection algorithm, we learned that we would likely have to use color thresholds as our main differentiator for lane lines, and this was only complicated by the aforementioned data diversity issues because even if lane lines are the same colors in the real world, different image would perceive these colors differently thus making it difficult to define color boundaries for the lines.

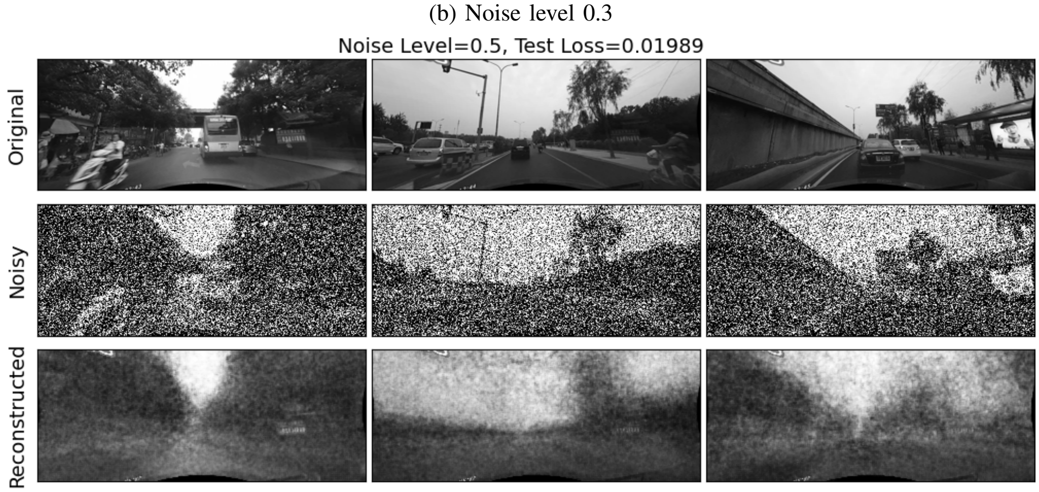
Since we tested two separate models with two different sets of input and training data, it became clear to us that diversity in the input state will drastically effect the accuracy of the autoencoder. Our model that was trained on a non-diverse training and validation set was highly accurate and was able to clearly reconstruct images from similar distributions. The other model was trained on more diverse data with less data in each driving scenario but more data overall. This meant it had a better representation of real world data, but it was not able to reconstruct images as well. This makes sense since the mapping to the latent space is more difficult with a broader input space, but we learned how greatly the variance in input data effects the overall accuracy of an autoencoder.



(a) Noise level 0.1



(b) Noise level 0.3



(c) Noise level 0.5

Fig. 8: Sample Images, Noisy Images, and Reconstructions

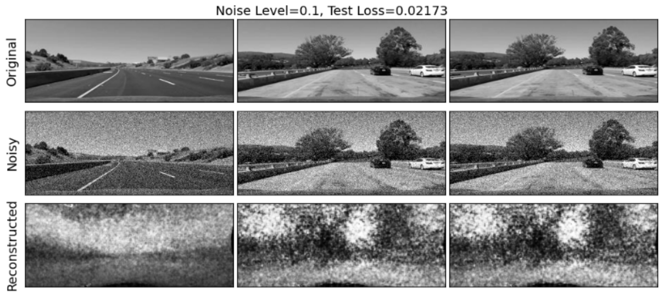


Fig. 9: CDAE on Out of Distribution Test Data

VI. Team Responsibilities

Matt was responsible for defining the autoencoder to be trained on the video data. This included sourcing the image frames and preprocessing the video into usable data, as well as training the DAE to reconstruct these images. He also implemented the lane detection algorithm to isolate lane lines from images as well as test the accuracy of the models.

Alanna was responsible for working with the larger CULane dataset and applying the autoencoder architecture to that data. This included a new data loading and preprocessing style and training of this model on CS department machines which enabled the model to train on the large scale dataset. She created command line tools for the model training, evaluation, and larger scale result visualization. She also conducted experiments for differing levels of noise.

VII. Conclusion

In conclusion, we implemented and evaluated a deep-learning-based approach for increasing the fault tolerance of the lane detection module of an autonomous driving system's perception component. The denoising convolutional autoencoder demonstrated improved lane line detection over the base lane line detection for data with moderate to high levels of noise. The poor generalizability of the model to out of distribution data and poor reconstruction of right-side lane lines call for a more diverse testing dataset. Future work with less computational constraints could explore a much more complex model on a much larger training set as a means of alleviating the issues observed with our model. Additionally, situational indicators such as steering, location, and speed of the car could be used to label incoming data to decrease the size of the input space and better define the latent space mapping.

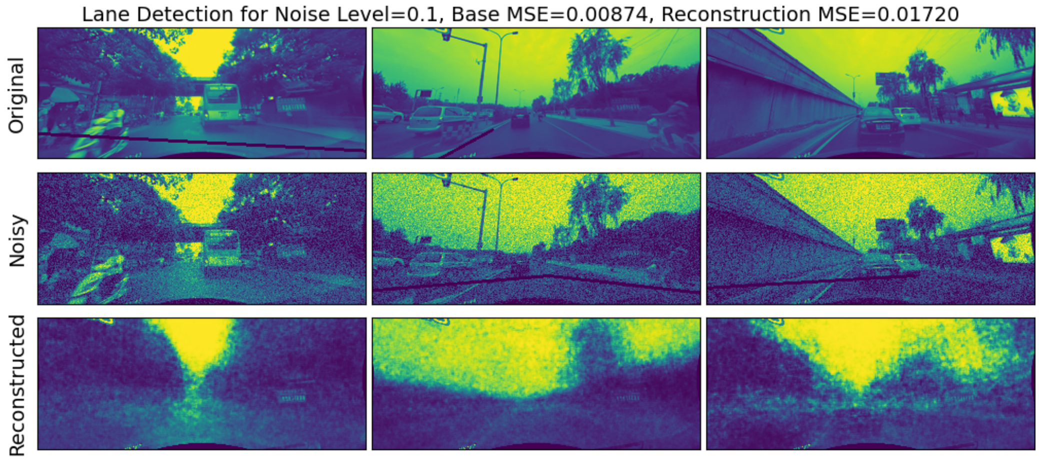
References

[1] A. Parukkatt, "Building a lane detection system," *medium.com*, Apr. 29, 2021. [Online]. Available: https://medium.com/analytics-vidhya/building-a-lane-detection-system-f7a727c6694

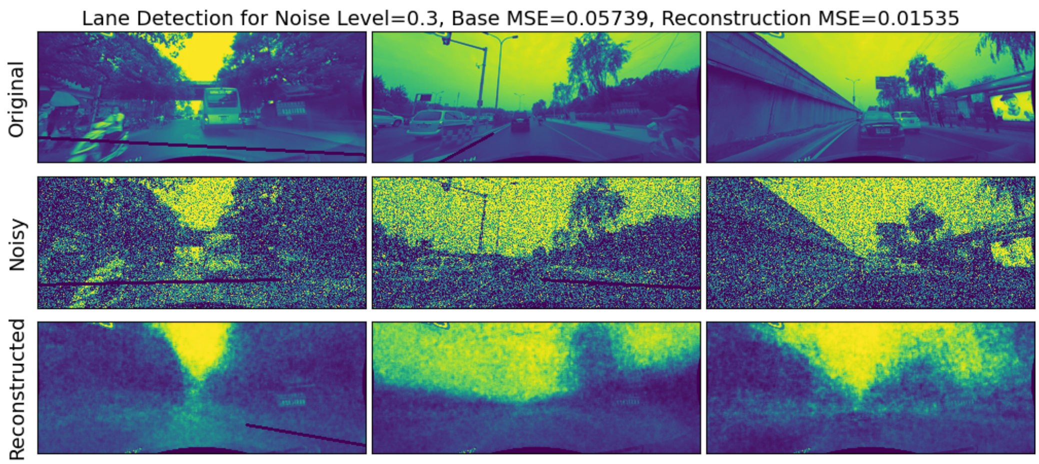
[2] A. Rosebrock, "Denoising autoencoders with Keras, TensorFlow, and Deep Learning," *pyimagesearch.com*, Feb. 22, 2020. [Online]. Available: https://pyimagesearch.com/2020/02/24/denoising-autoencoders-with-keras-tensorflow-and-deep-learning/

[3] L. Robinet, "Autoencoders and the Denoising Feature: From Theory to Practice," *towardsdatascience.com*, Nov. 26, 2020. [Online]. Available: https://towardsdatascience.com/autoencoders-and-the-denoising-feature-from-theory-to-practice-db7f7ad8fc78

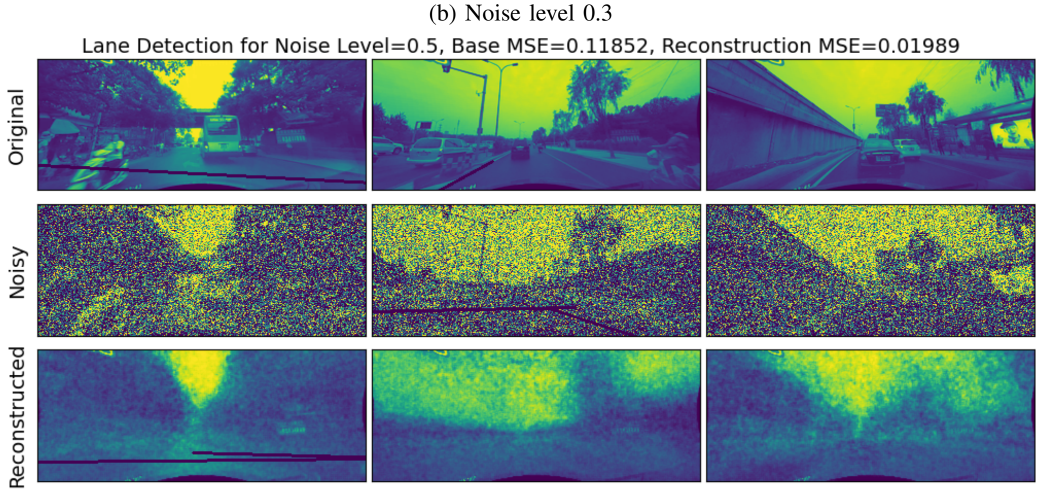
[4] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Spatial As Deep: Spatial CNN for Traffic Scene Understanding", February 2018. *AAAI Conference on Artificial Intelligence (AAAI)*. Available: https://xingangpan.github.io/projects/CULane.html



(a) Noise level 0.1



(b) Noise level 0.3



(c) Noise level 0.5

Fig. 10: Lane Detection Results