RESIDUAL CONNECTION LSTM SINGLE-IMAGE DEPTH ESTIMATION

Aidan Mitchell *amitc173@students.kennesaw.edu*
Ariel Vidal *avidal4@students.kennesaw.edu*
Jaskirat Sohal *jsohal@students.kennesaw.edu*

Department of Computer Science, Kennesaw State University
Marietta, GA

***Abstract* – This paper introduces a novel approach to estimating depth from a single image by adding Long Short-Term Memory (LSTM) functionality to a double hourglass Convolutional Neural Network (CNN). Estimating depth from a single image is a challenging task due to the lack of explicit depth information. The proposed method aims to overcome this limitation by leveraging the temporal information captured by LSTM and the spatial information extracted by the double hourglass CNN. The proposed method, augmenting the double hourglass CNN with LSTM, improves depth estimation by effectively modeling sequence-dependent temporal dependencies, outperforming current methods in terms of accuracy and robustness.**

***Keywords – CNN*, LSTM, Residual Block, Hourglass Module, Depth Estimation**

I. INTRODUCTION

A. Depth Estimation Overview

Using depth estimation as part of computer vision is an important task because it allows robots and computers to understand their environment. By using depth estimation, they are able to estimate distances and sizes of objects as well as identify obstacles and plan routes. By using estimated depth, you can avoid the high cost of SLAM sensors or, in the case of imaging systems such as the Kinect, you can avoid the need for structured light and time of flight, which is not reasonable in outdoor applications, such as autonomous cars.

B. Current Approaches

Estimating the depth of a single image is challenging because it requires estimating the three-dimensional representation of a two-dimensional image. Current approaches rely on the combination of computer vision and machine learning techniques, including optical flow and deep learning algorithms. General CNN-based methods have been built upon networks such as AlexNet, VGG, and ResNet, however these networks are built for high-level vision problems, and are not directly applicable, and often struggle in this use case. Other methods such as Eigen et al. and Liu et al. struggle when encountering noisy input images, which is often the case in real-world scenarios. A FCNN was also tested in the AutoDepth paper, but the results were lacking compared with the final hourglass model. For AutoDepth, CNNs with Residual Blocks are coupled with Encoder-Decoder networks with hourglass modules to estimate depth from a single RGB image.

C. Proposed Method

We introduce a new approach where LSTM is used to augment the double hourglass CNN, allowing the model to better capture temporal dependencies for improved depth estimation results. This method should significantly enhance the accuracy and

robustness of the depth estimation process, as it is able to utilize the previous frame information from the RGB image data. We achieve this by introducing a new Long Short-Term Memory (LSTM) structure in the model, aiming to boost depth estimation accuracy using previous frames' information. As robotics applications require continuous depth estimation, this model is especially beneficial. To achieve this, the image sensor would provide a stream of data for the model to process.

II. PROPOSED METHOD

A. Model Overview

The model proposed in this paper creates an addition to the model proposed in "AutoDepth: Single Image Depth Map Estimation via Residual CNN Encoder-Decoder and Stacked" [1]. The model proposed provides an addition of an LSTM layer containing 3 LSTM units following the CNN encoder. The output of which is then passed directly to and residually connected with the output of the hourglass module respectively.

The schematics of the proposed model can be seen in Fig. 1. The built in CNN encoder contains 7 Convolution blocks, using the ReLU as the activation function. The outputs of the convolution blocks are residually connected after every two convolution layers. The number of filters (3) used in the convolution layers remained the same throughout the entire model until Conv13, a convolution layer contained within the CNN decoder, to produce an image with a single channel. The number of filters are 3 and 1 respectively. The output of the CNN encoder is then reshaped into an LSTM input format while maintaining the number of units obtained from the CNN encoder. The LSTM layers contains 3 different LSTMs units, 1 for each channel for the image fed to the model. The output of the LSTM layer is then reshaped into a CNN input format and fed into the hourglass module. Before the values from the hourglass module are fed directly into the CNN decoder, they are residually combined with the output obtained from the LSTM layer, allowing for the features extracted from the LSTM layer to supplement the features extracted from the hourglass module.

The hourglass module is made up of residual blocks which contain residual connections from the bottom up to the top-down structure, shown in Fig. 2. The hourglass module contains 9 different residual blocks, aptly named R1, R2, R3, R4, R5, R6, R7, R8 and R9. The kernel size of the residual blocks is 256x256, 128x128, 64x64 for R1, R2, and R3 respectively [1]. The kernel size for the R4, R5, and R6 is maintained at 32x32. The kernel size for the top-down section of the module containing R7, R8, and R9 have kernel size of 64x64, 128x128, and 256x256 respectively [1].

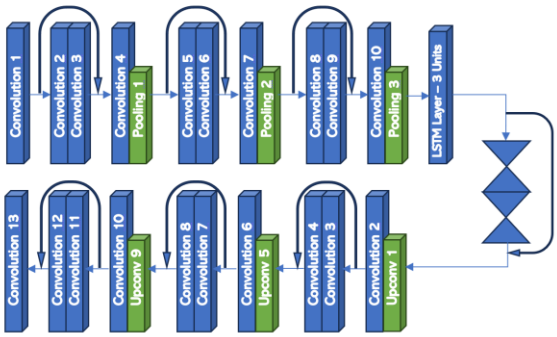


Figure 1: Residual Connection LSTM Model Schematics

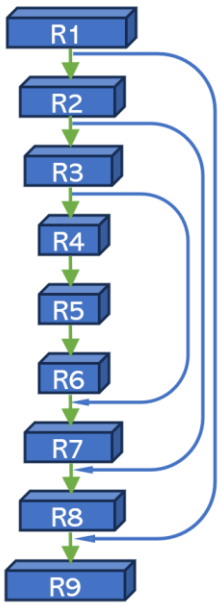


Figure 2: Hourglass Module Schematics

B. Hyperparameters

Adam was chosen to train CNN because of its reputation as an extremely powerful optimizer. In order to minimize the risk of exploding gradients in CNN training, the learning rate was set at 0.001 when specified, or left to the optimizer's discretion to minimize the risk of exploding gradients. It was also decided that a dropout rate of 0.2 would be used after each convolutional layer in order to ensure that overfitting was prevented by balancing learning and generalization.

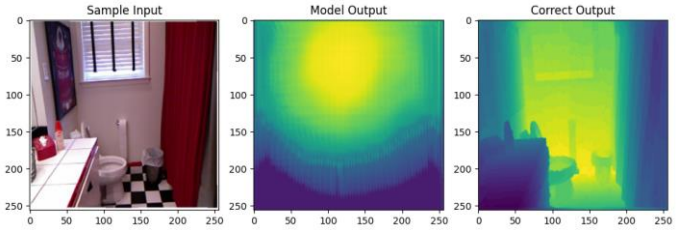
C. Loss Functions

In our experimentation, the loss function seriously affected the results of the model. Therefore, we carefully compared the loss functions to determine an optimal loss function. Specifically, when using exclusively MSE, the model would lack details in its output, which was undesired. To combat this, we utilized perceptual loss, relying on imagenet to calculate the difference between the two images and generate the perceptual loss. Based on AutoDepth's implementation, we calculated loss as follows:

$$Loss = MSELoss + 0.5 * PerceptualLoss$$

An example comparison of using strictly MSE Loss and the combined loss function can be seen in Fig. __, where the output of the model with exclusively MSE Loss can be seen to give results without any details, whereas the model with the combined MSE and Perceptual Loss was able to produce significantly more detail.

Exclusively MSE Loss:



Combined MSE and Perceptual Loss:

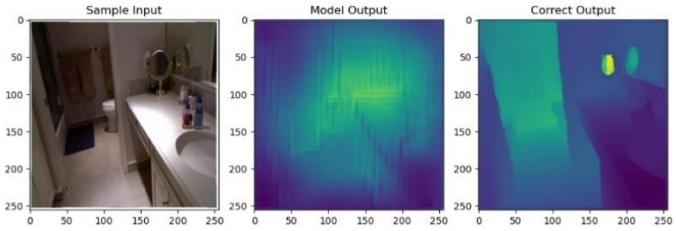


Figure 3: Comparison of MSE loss and combined loss

Finally, we also introduced an additional term which punished the model for having a small standard deviation of values for its output. This effectively punished the model for outputting a depth map consisting of only one color, which was a common occurrence when training, and resulted in a model which would never recover from this issue.

III. EXPERIMENTAL DESIGN

A. Dataset

During our experiment, we used the NYU V2 Depth dataset, a collection of indoor scenes captured using the Kinect sensor. This dataset consists of 464 classes of indoor scenes, offering a broad spectrum of diverse visual situations. In order to avoid significant loading times, we were only able to use 20 percent of the total dataset due to the limitations imposed by our hardware.

Within the training subset, there are 249 scenes with each scene having a sequence of frames taken by a video camera and a Kinect sensor. From these scenes, we randomly selected 795 training images and depth maps. Through this approach, we were able to maximize the utility of our data, accommodating our specific computational capabilities while maintaining training integrity.

It is worth noting that the AutoDepth paper used both the IKEA and NYU V2 datasets, while we opted to use exclusively the NYU V2 dataset. This was because between the two datasets the NYU dataset contained the most diverse environments. Additionally, the NYU V2 dataset explored typical NYU interior rooms, which is an expected use case for this model.

B. Training

Training was executed in two distinct phases: the initial phase involved training utilizing the Mean Squared Error (MSE) loss, followed by the second phase where the model was trained with a combination of MSE and perceptual loss.

In the first phase, the model underwent training for 250 epochs with a batch size of 8 and 100 steps per epoch. Both the base model and its modified LSTM model were trained using these parameters.

The second phase of training used 50 epochs, with a consistent batch size of 8 and 100 steps per epoch. This phase introduced a modified loss function combining MSE and perceptual loss, which enabled the model to better learn details.

During the training process, the base model exhibited notably faster convergence than the LSTM-equipped model. Specifically, the base model achieved an average training time of 8,750 seconds, contrasting with the LSTM model's extended training period averaging 38,700 seconds.

IV. RESULTS

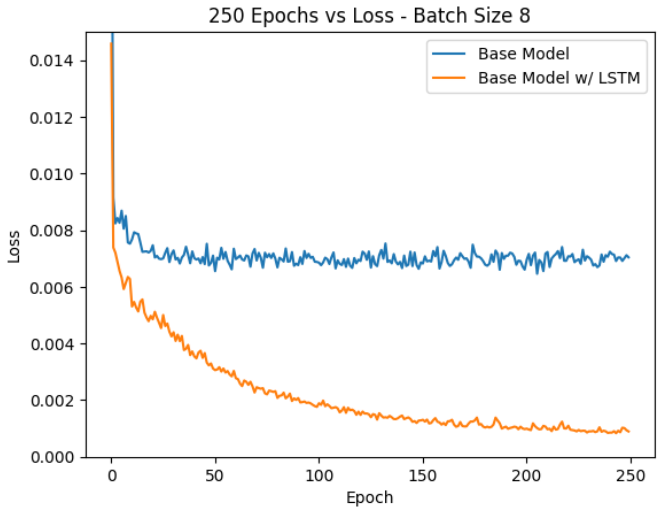
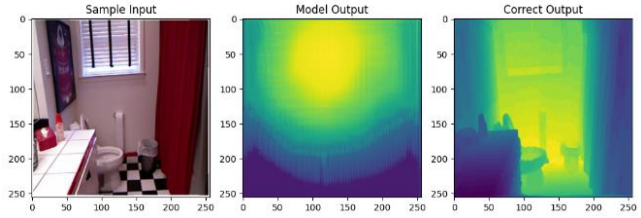


Figure 4: MSE Training Loss for base model vs. Proposed Model

Extrapolation information from the results of training the base and the proposed model using 250 epochs, with the batch size of 8 and Means Squared error as the loss function, the proposed model is able to outperform the base model in terms of training loss. Although the proposed model required five times as much time for training the proposed model is also able to reach a stable decline in loss during training comparatively to the base model.

Base Model (MSE)



Proposed Model (MSE)

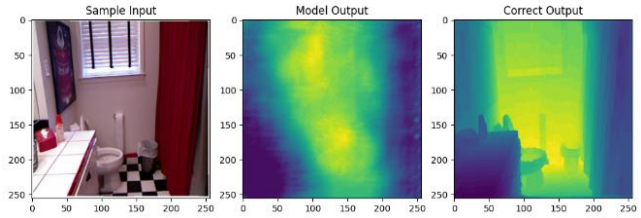
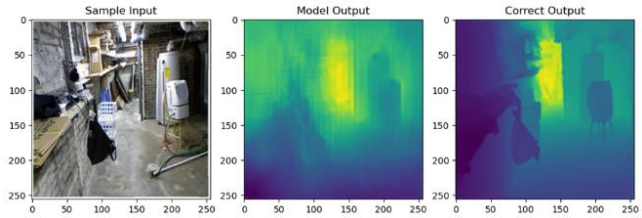


Figure 5: Comparison of base model versus proposed model with MSE loss

For further comparison the pretrained models were once again trained upon the combined loss. Due to time constraints the models were trained using 50 epochs while maintaining the batch size of 8, and 100 steps per epoch.

Base Model (Combined Loss)



Proposed Model (Combined Loss)

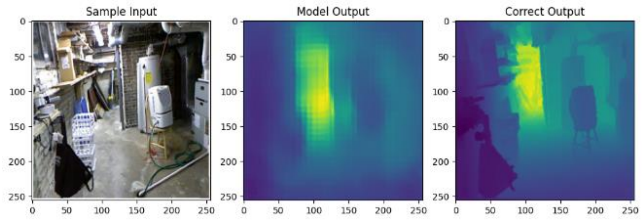


Figure 6: Comparison of base model versus proposed model with combined loss function

As seen in Figure 6, the proposed model is able to produce high levels of detail compared to the base model even at a lower number of epochs. As seen during training for the Mean Squared Error, it is possible for the proposed model to outperform the based model, given higher epochs.

V. CONCLUSION AND ANALYSIS

Our study has yielded promising results, particularly in the comparative evaluation between the base model and the proposed model. Notably, the proposed model demonstrated an enhancement in the output produced, presenting results that surpass those of the base model. While our achieved results in comparison to the target may be incremental, they represent progress toward our research goals.

Although we acknowledge the current limitations, specifically the use of only 20 percent of the NYU V2 dataset, we recognize substantial improvement can be achieved through broader exposure. In the future, we will incorporate a larger dataset into our training. By incorporating the Ikea dataset into our training set, we will diversify the model's exposure to a wider range of scenes.

In fine-tuning our model, architectural experimentation remains a focal point. The strategic placement of the Long Short-Term Memory (LSTM) cells within the model architecture stands out as a compelling avenue for exploration. Additionally, we also plan on experimenting with the impact of varying batch sizes on model performance. We recognize the significance of this parameter in optimizing training dynamics.

While our current results demonstrate promise, we readily acknowledge the potential for improvement. The identified areas for future work hold the potential to further elevate the model's efficacy. We remain optimistic that these targeted experiments will contribute to the continued

refinement and optimization of our single image depth estimation model.

VI. REFERENCES

[1] S. Kumari, R. R. Jha, A. Bhavsar, and A. Nigam, ‘Autodepth: Single image depth map estimation via residual cnn encoder-decoder and stacked hourglass’, in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 340–344.

[2] D. Eigen and R. Fergus, ‘Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture’, in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[3] F. Liu, C. Shen, and G. Lin, ‘Deep convolutional neural fields for depth estimation from a single image’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5162–5170.

VII. ANNEXURE