Depth and Contact Prediction for Tactile Sensors

Vaibhav Sanjay
vsanjay@terpmail.umd.edu

Eric Liau
eliau@terpmail.umd.edu

Chibueze Nwade
cnwade@terpmail.umd.edu

Varun Lagadapati
vlagadap@terpmail.umd.edu

Leo Li
leo2003@terpmail.umd.edu

Abstract

This project explores the application of deep neural networks in predicting 3D contact geometry from monocular images obtained through a GelSight tactile sensor. Our primary goal is to establish an inverse sensor model capable of reconstructing local 3D geometry from tactile images. Two neural networks, ContactNet and TactileDepthNet, are designed for contact and depth prediction, respectively. Utilizing the TactileDataset in the experimental setup, the results exhibit promising outcomes in terms of contact and depth predictions. However, challenges such as model overfitting and unexpected prediction results are discussed. Overall, this project provides valuable insights into the training of networks for depth estimation and identifies potential areas for future enhancements.

1. Introduction

Tactile sensors are devices designed to measure information arising from physical interaction of robots with their environment and detect stimuli resulting from mechanical stimulation, temperature variations, and pain-like responses. However, recent sensor developments have predominantly concentrated on capturing the 3D geometry of contact. Depth prediction is the task of measuring the distance of each pixel relative to to the camera and depth is extracted from either monocular (single) or stereo (multiple views of a scene) images.

In this project, we use deep neural networks to predict 3D contact geometry from monocular images of a GelSight tactile sensor, which is also called vision-based tactile sensor [5]. Specifically, we aim to acquire the inverse sensor model to reconstruct local 3D geometry from a tactile image. This involves training the model in a supervised manner to predict local height-maps and contact areas from tactile images.

2. Literature Review

The most notable work that relates best to our approach is presented in the paper *"Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network"* by Eigen et al. In their work, they handle depth estimation by using two deep network stacks, "one that makes a coarse global prediction based on the entire image, and another that refines this prediction locally" [1]. They then refine their predictions by using scale-invariant error to "help measure depth relations rather than scale" [1]. Laina et al. follows a similar approach as Eigen et al. as they utilize "a fully convolutional architecture, encompassing residual learning, to model the ambiguous mapping between monocular images and depth maps" [3]. The main difference between the two approaches is that while Eigen et al. uses two deep network stacks, Laina et al. uses a single architecture "that is trained end-to-end and does not rely on post-processing techniques" [3].

Ma et al. also focus on dense depth prediction by training their model using a sparse set of depth measurements (supplemented by additional sparse depth estimations to improve the model's robustness) and a single RGB image [4]. Their proposed model is a single deep regression network that learns directly from the RGB-D raw data of its inputs and evaluates the impact of depth samples on its prediction accuracy [4]. Godard et al. takes a contrary approach by utilizing binocular stereo footage to train their singular convolutional neural network [2]. The model is then trained based on the constraints of epipolar geometry as they generate disparity images by training their network with an image reconstruction loss [2].

Wang et al. is also another work that address the issue of depth estimation, but they do this in a different way from the previous approaches as they have a different goal. Their approach involves using a sensor called the GelSight Wedge sensor, which is used to achieve high-resolution 3D reconstruction with the goal of achieving pose tracking

in 3D space [5]. To find the depth of the scene, they first calculate the mapping of the image from color (RGB) to horizontal and vertical surface gradients and then apply a fast Poisson solver to integrate gradients and get the depth [5].

3. Methodology

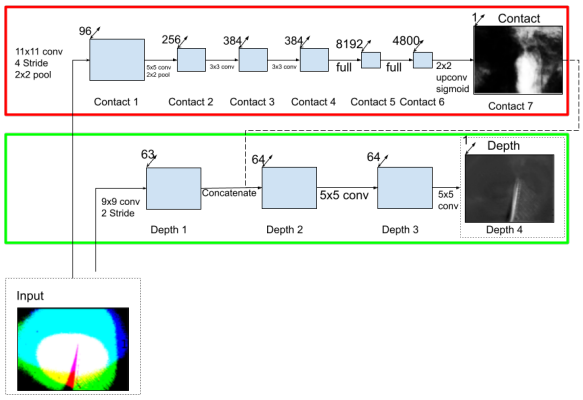

Figure 1. Network Diagram

Figure 1 shows our network architecture. We designed two neural networks by using one network for contact prediction which we called the ContactNet and another network for depth prediction which the referred to as the TactileDepthNet. This architecture takes inspiration from [1] where a similar double network is constructed. ContactNet contains five feature extraction layers of convolution, two fully connected layers, max pooling layer, dropout layer, and transposed convolution layer. After the dropout layer, the input data has been resized using anti-aliasing and the transposed convolution layer applies the sigmoid activation function to the input by squashing the input values between 0 and 1. This output can be thought of as the likelihood that the contact mask is 1 at this position in the image. The result of the contact is concatenated to the depth after the fully connected layers and this concatenation is upsampled back to the original size using the transposed convolution layer (ConvTranspose2D). The layers in ContactNet take the tactile image and output the predicted contact mask. TactileDepthNet contains three feature extraction layers of convolution. The layers in TactileDepthNet take the tactile image and output the predicted depth (heightmap). Between each network layer, we have a ReLU activation. In terms of the parameter configurations, we used the Binary Cross Entropy (BCE) loss for ContactNet and the Euclidean Loss function (also called the L2 Norm loss) for the TactileDepthNet. The BCE loss was chosen for the ContactNet since the resulting output is binary, so this loss function is fittiing. In the context of TactileDepthNet, the L2 norm loss measures the difference between predicted and true depth values. To be specific the loss was defined as the average of the matrix norms of order 2 of the difference between the predicted depth image and the ground truth depth image. The learning rate is set to 0.001 and we used Adam optimizer for both networks. We trained for 10 epochs and produced the loss graphs.

4. Experimental Setup

We used the TactileDataset which we saved in the folder mini_depth_dataset and split the dataset into training and validation sets. We stored the depth and tactile images into their respective folders for both sets where the training set represents 80% of the dataset and validation represents the remaining 20% of the dataset.

Before the training is done, we apply some precprocessing to the training images. We first normalized the images before training. It is known that normalization helps reduce the time and computational resources needed to train CNNs. Additionally, we apply some data augmentation techniques to the dataset. The techniques we applied are

1. **Random Rotation**: We randomly rotated the images between -5 to 5 degrees, as was done in [1].

2. **Random Horizontal Flip**: We randomly flipped the images horizontally with probability 50%.

These data augmentations are important for enhanced performance of the neural networks.

5. Results

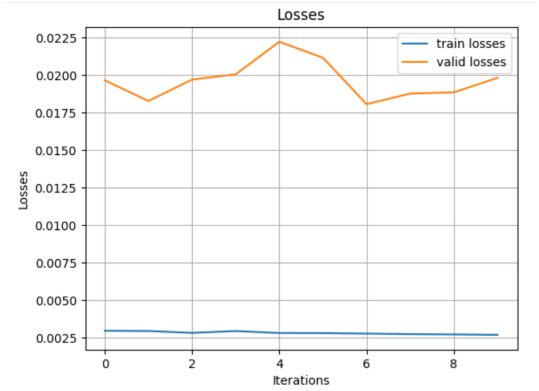

Figure 2. Loss graph for ContactNet Model.

Figure 2 shows the ContactNet loss graph and 3 shows the TactileDepthNet loss graph. Upon further examination of the ContactNet graph, the train losses are low however the validation losses are much higher and don't follow the

same trend. The train losses converge at around 0.0025 while the validation losses are a bit more random, but likely converges at around 0.02.

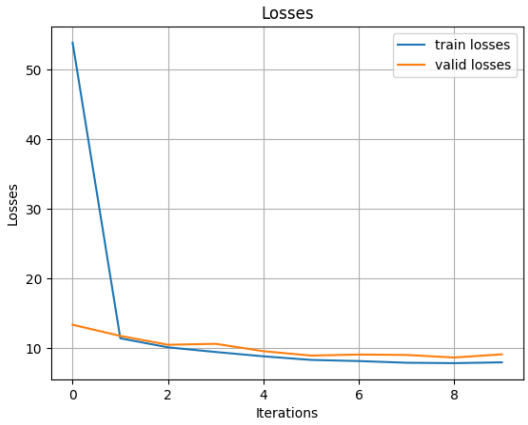

Figure 3. Loss graph for TactileDepthNet Model.

For the depth loss graph, both the train and validation losses start off at a higher value and converge to some value below 10. This is a good sign that our training is working over the 10 epochs.

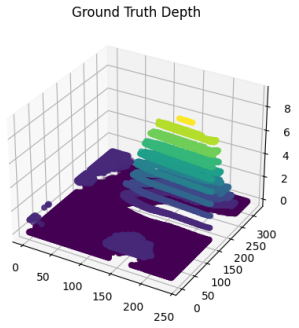

Figure 4. Ground truth depth corresponding to a tactile image.

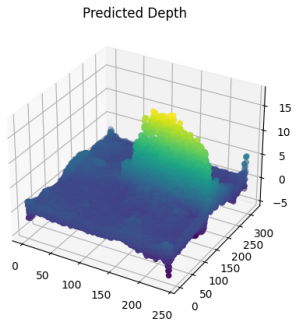

Figure 5. Predicted depth corresponding to a tactile image.

The ground truth depth and its corresponding prediction are shown in figures 4 and 5 respectively. The resulting depth prediction doesn't look bad. The shape of the depths seem correct however we are over-predicting the depths in many of the images.

6. Discussion

The resulting contact model loss graph shows that our data is a slightly overfitting model (as discussed in part 5 of the report), though not by much and can still be considered within the margin of error. Meanwhile our depth loss graph suggests accurate training and correct depth perception. In comparison to our project 1, where we analyzed clusters of data, as well as masking and pinpointing a specified object, the test result are similar in the concept of perception of depth of an object, in our case a data set. We don't have any training loss comparisons between the 2, but from the 3D graph outputs, you can tell that the implementation for this model is a lot more detailed. As unlike project 1, where we determine the position of an object simply by segmenting the object from its background using the Gaussian Mixture Model, we derive our prediction from using both the depth and contact model, and using a tactile sensor implemented through PyTorch. An unexpected observation that we encountered is prediction results. From the loss graph, it was assumed that we had an overfitting model. However, as we have told in section 5, our data was actually over-predicting, which is counter-intuitive to our loss graph. Though in this case, it may simply be the case of our depth loss graph in particular.

For the image that we were given to predict on Piazza, shown on figure 7, our resulting depth prediction is shown in figure 8.

7. Conclusion

In this paper, we detailed how a pair of Convolutional Neural Networks can be used for depth prediction given a tactile image. We discussed preprocessing of the given images, including normalization of tactile images and the generation of the contact images from the ground-truth depth. Our architecture involved the use of two neural networks, one for contact prediction and the other for depth prediction. The depth model used information from the contact model to make more accurate predictions of the depth.

A key discovery for this project was the training for the depth estimation of a tactile image, which would have been impossible without the use of PyTorch, which allowed us to process the images to obtain the datasets required for our model. By training tactile images, our implementation has reconstructed a local 3D geometry model from the given tactile images. However, our loss graph does show that our implementation has overfitted the inital data, which is shown in our contact image output where not all the new data from the contact image was correctly identified for our

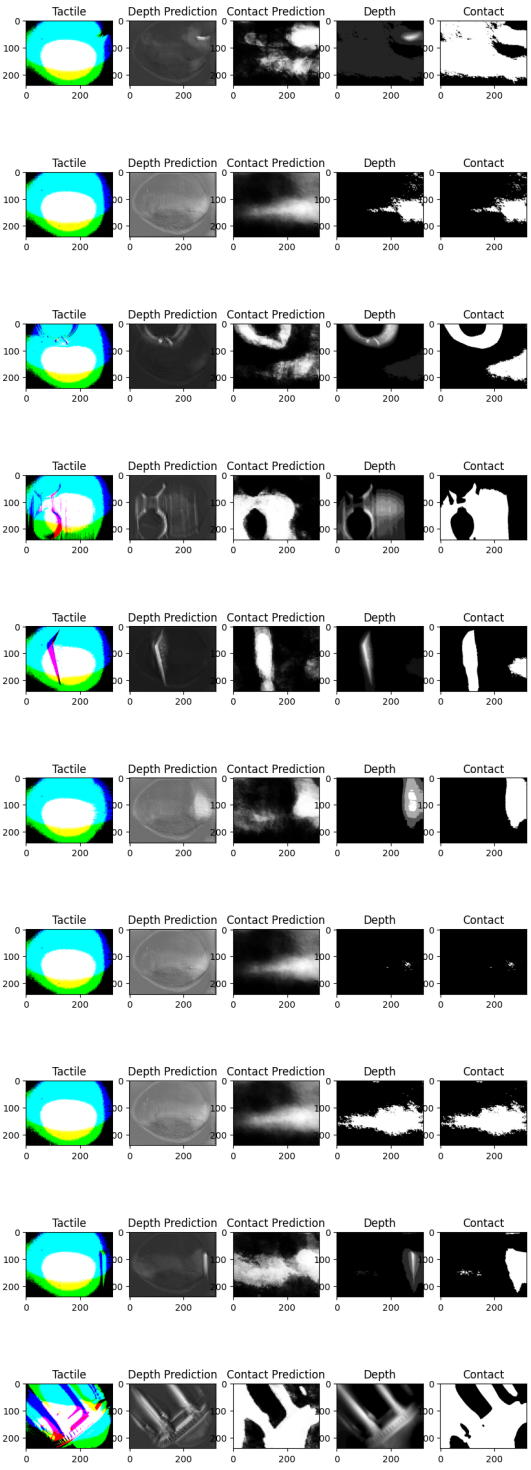

Figure 6. Predicted and ground truth depth and contact images for a set of 10 given tactile images.

final 3d geometry.

Figure 7. Tactile Image

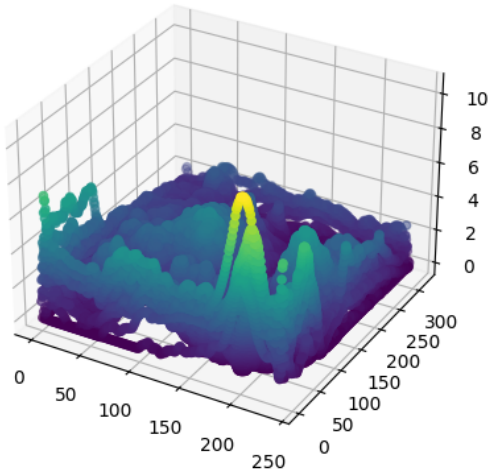

Figure 8. Depth Prediction

8. Future Work

To improve on our current results we plan to implement our models using different layer combinations. One approach we had in mind was using more fully connected layers, as that approach would best replicate the original approach that was provided as a guide. Another task we have in mind to improve our models' capabilities is to improve their robustness. During the training phase, we noticed that the models (especially the contact network) had very low loss outcomes for the training set in comparison to the validation set. This is likely as a result of overfitting and our models inability to properly generalize to new cases. Although it did not greatly affect our test results, it affected them enough to alert us of the possibility that our model may be overfitting.

9. Takeaways

A key takeaway we had from working on this project was how to train our networks for the purpose of depth estimation, with the main focus being how we evaluated our models' performances. While working on the project, we had to test a variety of loss functions to determine which functions worked best for our task as depth estimation abides by objectives different than we have seen in other computer vision tasks. This also meant that we had to learn about different methods to optimize our models' performances, which is an area that we still hope to address in the future. Additionally, we learned the importance of normalization and its importance as it was needed to reduce the complexity of the patterns that our models had to learn. Lastly, we were also able to learn about 3D Reconstruction which is a Computer Vision task that relies heavily on depth perception and classification. This helped us understand the bigger picture of what we were working on, thus enabling us to make wiser choices during the defining and training of our models.

References

[1] David Eigen, Christian Puhrsch, and Rob Fergus. "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network". In: 2014. arXiv: 1406.2283 [cs.CV].

[2] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. "Unsupervised Monocular Depth Estimation with Left-Right Consistency". In: 2017. arXiv: 1609.03677v3 [cs.CV].

[3] Iro Laina et al. "Deeper Depth Prediction with Fully Convolutional Residual Networks". In: 2016. arXiv: 1606.00373 [cs.CV].

[4] Fangchang Ma and Sertac Karaman. "Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image". In: 2018. arXiv: 1709.07492v2 [cs.RO].

[5] Shaoxiong Wang et al. "GelSight Wedge: Measuring High-Resolution 3D Contact Geometry with a Compact Robot Finger". In: 2021.