CS230

Learning-Based Visuo-Tactile Tendon Perception

Andrew Pasco
Department of Mechanical Engineering
Stanford University
apasco@stanford.edu

Jorge Garcia
Department of Mechanical Engineering
Stanford University
jalgarci@stanford.edu

Abstract

We apply learning-based visuo-tactile depth reconstruction for tendon perception using a vision-based tactile sensor. Our goal is to predict a dense depth field of sub-surface tendon geometry from RGB tactile images acquired during palpation. We train a MobileNetV3 for classification and a U-Net for segmentation, with test accuracy and IoU of 0.958 and 0.196. For depth reconstruction, we use a U-Net with a frozen DenseNet161 encoder, achieving a test MAE of 4.18mm . Because of model selection considerations, this pipeline could be utilized in real-time for image annotation of detected tendons.

1 Introduction

In medical and physical therapy contexts, clinicians rely on force and deformation patterns felt through touch to interpret subsurface anatomical structures (1). Variations in stiffness or local geometry changes are conveyed through tactile sensing. However, tasks that require both high force application and high sensitivity, like deep-tissue palpation or acupressure, can increase the risk of strain-related injury for clinicians, especially to extremities like the thumb (2). Therefore, introducing a method that enables geometry interpretation under high-force application remains essential for acupressure therapies.

Existing Vision-Based Tactile Sensors (VBTS) typically consist of a contact module, an illumination module, and a camera module (3). Deformations in the dome, along with the sensor's proprioceptive signals, allow for reconstructing objects at high fidelity for dexterous manipulation (4; 5). However, current systems are designed for relatively low-force interactions and cannot endure the higher loads required for acupressure diagnostics. The Biomimetics and Dexterous Manipulation Lab (BDML) is developing a new high-force visuotactile sensor for sub-tissue feature identification. We collaborate with the lab to develop an algorithm capable of detecting and reconstructing this geometry.

This project investigates learning-based tactile perception for reconstructing contact geometry from vision-based tactile images. Our objective is to train a deep-learning architecture capable of decoding deformation and color cues from tactile input to segment and reconstruct high-resolution 3D surfaces. By extending this framework toward biomedical use, we aim to explore whether visuotactile signals can recover anatomical surface features through touch alone.

2 Related work

In vision-based tactile sensing, deep convolutional networks (often encoder–decoder models like U-Net or networks with ResNet/DenseNet backbones) are widely used to recover 3D contact geometry from a single tactile image. These models predict either a depth map of the contact surface or a map of surface normals, depending on the approach.

CS230: Deep Learning, Fall 2025, Stanford University, CA.

Traditional methods involve lookup tables that map color intensity to surface normals. For instance, in GelSight, it is mentioned that earlier iterations involved using a lookup table. However, they now use a small multi-layer perceptron (MLP) with 3 hidden layers (6). The learning approach is not dependent on the indentation position, yielding higher reliability than a look-up table.

For reconstruction, PneuGelSight system uses a dual-branch network (combining global shape and local color features) and a small MLP decoder to regress local surface normals, trained with a simple MSE loss (4). Their method involves knowledge of the sensor point cloud at every point in time, which allows them to compute the manipulation surface normals.

Some other methods involve predicting depth. For instance, DenseTact uses a DenseNet-161 encoder with skip-connections in a decoder to produce high-resolution depth maps of the touched surface in real time (5). They use simulation and ray-casting to assign depth values to image pixels.

3 Dataset and Features

Phantoms are models that mimic biological tissue and its properties (7). The phantom material is cast from smooth silicone, and the tendons are 3D printed from PLA. We built a dedicated data collection pipeline using a Flexiv Rizon robot and the BDML sensor. We collected camera frames at a frequency of 30 fps and at a resolution of 1920×1080 (see Figure 1).

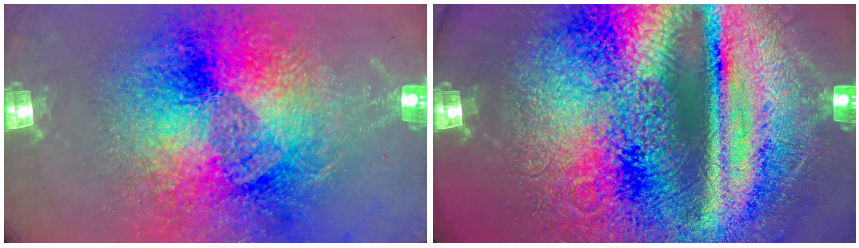

Figure 1: Example of Raw Non-Contact and Contact Frames

From there, we created a labeling pipeline using know STLs and camera intrinsics/extrinsics to label the collected images for training and validation. The simulation was created to streamline the data acquisition process. The full pipeline can be found in our GitHub.

4 Methods

4.1 Detection

Because the model pipeline is intended to be used in a “live” deployment, object detection used a separate model which only fed downstream when a tendon was detected. Initial experimentation transfer learned from the "IMAGENET1K_V1" weights for the `torchvision.models.resnet18` model (8; 9). Layers 3 and 4 were unfrozen, and a binary classification head was added (final 256-hidden unit FC layer connected to single output with Sigmoid activation). Leaving the initial layers of this large stacked CNN with residual connections frozen could help retain the knowledge of a general image classification task while fine-tuning the later layers would yield better results for this specific 1/0 detection task.

After initial experimentation, there was a desire to reduce the inference time by using a smaller model. To achieve this goal, a smaller network was transfer-learned from the "IMAGENET1K_V1" weights for `torchvision.models.mobilenetv3_small` (10; 9). This architecture utilizes depth-wise separable convolutions and requires less computation. Weights of “blocks” within the network except the last four were frozen, and a similar binary classification head as before was appended for this specific task. Additionally, images were downsized from a resolution of 1920x1080 to 320x240.

4.2 Segmentation

After determining which images corresponded to contact, the next step was to segment/mask the portion of the image containing the imprint of the tendon. Given image-mask pairs from the automatic

labeling pipeline, a small CNN inspired by U-Net (11) was trained for pixel-wise classification of tendon/background. This network consists of four "encoding" blocks (2D convolve, ReLU, max pool) which increase channels while decreasing image size, a bottleneck with another 2D convolve, and four "decoding" blocks (2D convolve, ReLU, interpolate, channel-wise concatenate with save from encoder side) which decrease channels while expanding back to the original image dimensions. The concatenation with the save from the same step of the encoder side intends to re-capture local details that were lost through max pools.

Because of the data collection setup, the tendon was oriented the same way in all of the training data. This does not fully represent the true data distribution, so data augmentation strategies were utilized. The size of the dataset was quadrupled, with the three extra copies rotated (seeded) randomly within intervals which evenly divided 360°. Additionally, to protect against deteriorated performance on the rotated copies, a 160x160 center-crop was taken which ensures the central sensor dome, but not any possible background fill after rotation, is included in the examples given to the model.

4.3 Direction Prediction

One simple "endpoint" for this pipeline is identifying the detected region and the orientation of that region in the field-of-view (FoV). Post-processing steps included identifying only the largest contiguous mask (the model would sometimes output additional, spurious, smaller masks in the image), manually overriding outputs where this largest mask was on the outer edges (this often corresponded to detection of the bright LED in the FoV), and computing a direction via principle components analysis (PCA). In a 2D image, the first principle component of a region of pixels corresponds to the "major axis." For a tendon with rectangular or ellipsoid shape, this is the direction.

4.4 Reconstruction

Prior approaches to tactile depth estimation, such as DenseTact (5), reconstruct geometry by predicting surface normals and applying raycasting. However, this requires a forward model of the gel's deformation to simulate light transport – a significant modeling burden that is out of the scope of the class. We instead supervise the network directly with depth maps generated by our labeling pipeline, allowing the model to learn the RGB-to-depth mapping end-to-end without requiring a physics-based deformation model.

The reconstruction involves a small CNN inspired by U-Net (11). The encoder layer uses DenseNet161 weights trained on ImageNet (12) The first step was to have a baseline for the validity of the architecture. Since it is common to use MSE and L1 loss, we wanted to compare both to see which yielded a better result. For this experiment, we used the frozen DenseNet161 encoder + default 1×1 prediction head. After, we decided to try a custom head with a frozen encoder plus a small multi-layer regression head for richer geometric prediction. However, this comes with the tradeoff of a larger model and longer inference times.

5 Experiments/Results/Discussion

5.1 Direction Prediction Pipeline

All models were trained with an 80/20 training/validation split due to the relatively low number of examples. For detection, 2708 samples were used, with 2400 positive examples and 308 negative examples. Validation samples were randomly selected from the same videos used for training and held out. For segmentation, only the 2400 positive examples were used, with augmentation increasing this to 7200. The test set consisted of an additional 784 positive frames from other videos captured with the same data collection setup. Additionally, all images input to both the detection and segmentation models are normalized with statistics computed from the training examples.

For the initial ResNet, training and validation accuracies of 0.998 and 0.992, respectively, were achieved after 3 epochs of mini-batch training with a batch size of 8 and a learning rate of 0.0001. Average inference time was 85ms on an Intel Corei7 CPU. The ResNet was not evaluated on the test set to prevent biasing the choice of model architecture. For the MobileNetV3, training and validation accuracies of 0.999 and 0.996, respectively, were achieved after 5 epochs of mini-batch training with

a batch size of 8 and a learning rate of 0.0001. Average inference time was 11ms on an Intel Corei7 CPU, and accuracy of 0.958 was achieved on the test set.

For the segmentation U-Net, the metric of interest was "intersection over union," the number of pixels in the intersection of the predicted mask and the ground truth mask over the number of pixels in the union of the sets. Values of 0.917 and 0.879 were achieved for train and validation, respectively, after 26 epochs of mini-batch training with a batch size of 16 and a learning rate of .001. Average inference time was 69ms on an Intel Corei7 CPU, and IoU of 0.196 was achieved on the test set. Overall inference time for the pipeline when utilizing the MobileNetV3 for detection is thus 80ms, allowing processing at 12.5Hz, a reasonable frame-rate for real-time use.

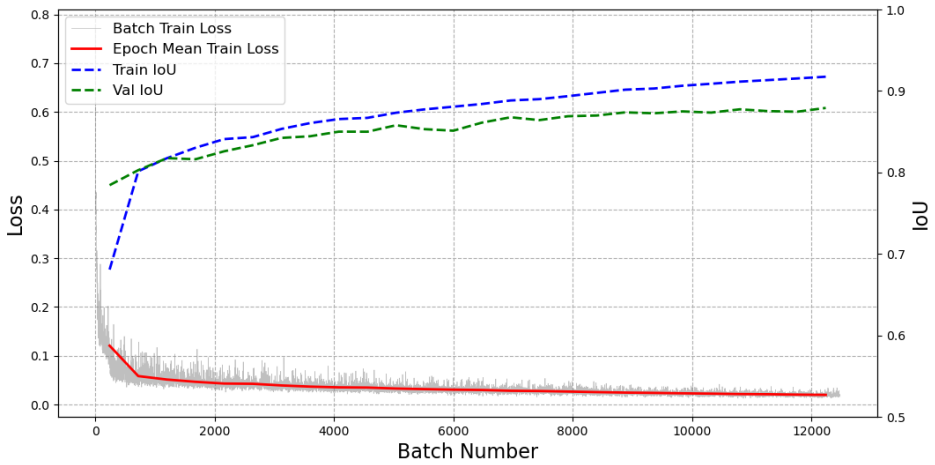

Figure 2: Training and IoU curves for mini-batched U-Net training.

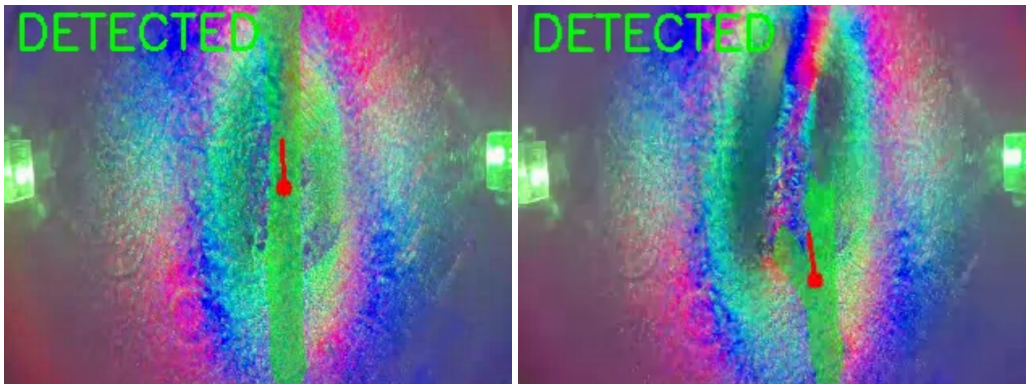

Figure 3: Examples of qualitatively "good" (left) and "bad" (right) segmentation results for frames from the test set. The predicted tendon is masked in green, and the obtained direction is shown in red. The curved tendon can be seen in the "bad" result.

There is one primary reason that the model performed poorly on the test set. We realized after model training that the holdout test set of videos had a curved tendon, while those in the training and validation sets had only straight tendons. This test data thus came from a different distribution, so generalization out of distribution was limited (though not a complete failure, as some frames still had satisfactory results). With more time, additional data could have been collected on both straight and curved tendons so the overall distribution would be more balanced. We wanted the test set to be comprised of entirely separate videos instead of just holding out frames from a general set of videos, but the accidental holding-out of this critical variant yielded poor test results. It should be noted that the detection module still performed reasonably well on this out-of-distribution data since the model likely learned to detect any disturbance of the sensor membrane.

5.2 Reconstruction Pipeline

All models were trained using an 80/20 training/validation split due to the limited number of examples. We used a total of 3,402 samples with size 256x256 for faster training. We used a batch size of 8 and learning rate of 0.0001. Early experiments revealed that depth values in their original units (ranging 0.012–0.018m) caused vanishing gradients. We addressed this by normalizing depths by multiplying by 1000.

To establish a baseline, we compared L1 and MSE loss using an unfrozen DenseNet161 (13) encoder with a 1×1 prediction head. L1 significantly outperformed MSE, achieving a validation MAE of 0.059 compared to 0.377 for MSE. This aligns with prior work showing L1 produces sharper depth boundaries and is more robust to outliers than MSE. We then compared two decoder configurations with the frozen encoder: the default 1×1 prediction head, and a custom multi-layer head with two 3×3 convolutional layers and ReLU activations. The simpler default head achieved a lower validation MAE (0.050 vs 0.057), while the custom head achieved marginally higher SSIM (0.969 vs 0.960). We attribute this to the custom head's additional capacity leading to slight overfitting. Given the lower MAE and simpler architecture, we selected the frozen encoder with default head for final evaluation. Our test consisted on an independent run with 100 frames. We obtained an MAE of 4.18mm and a SSIM of 0.9411. The higher MAE could be attributed to data being out of distribution from train and validation sets. However, results show in Figure 4 show good reconstruction for tendon that can possibly be enhanced with post-processing.

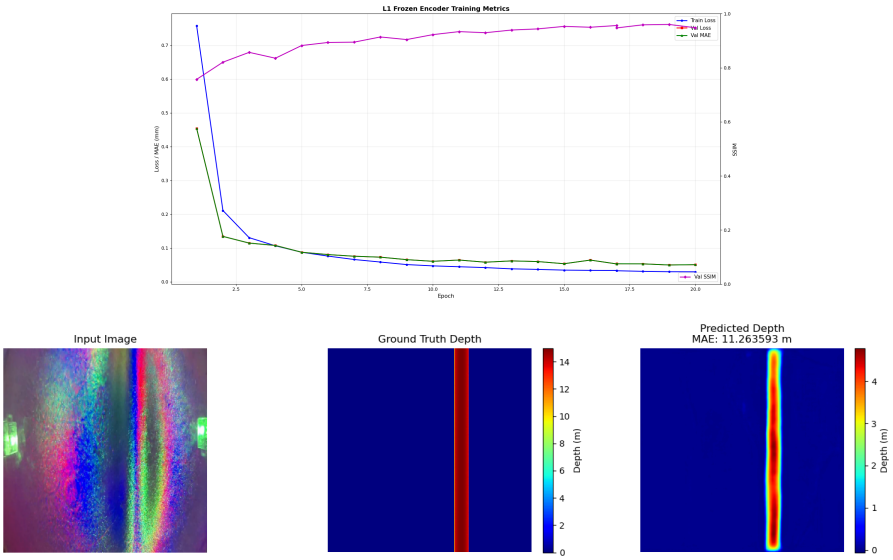

Figure 4: Training-Validation Curves and Test Result.

6 Conclusion/Future Work

We were able to train and implement a sequential computer vision pipeline capable of inferencing in real time on video from a vision-based tactile sensor. Moreover, we developed a reconstruction pipeline that can be used to infer the depth. With additional time, it would have also been interesting to use the segmented mask as input to improve the reconstruction quality.

In the future, we will investigate using multi-modal sensing. Force is a potential feature to estimate how deep the sensor goes into the silicone. Force sensing could also help to reinforce the detector output and indicate the need to execute the downstream inference.

7 Contributions

Please find our Github at this link.

7.1 Jorge Garcia

Created data collection pipeline. Implemented, trained and assessed the reconstruction prediction pipeline.

7.2 Andrew Pasco

Completed initial manual data labeling for initial model assessment before robust data collection pipeline was constructed. Implemented, trained, and assessed the detection/segmentation/direction prediction pipeline.

References

[1] A. Chan, A. Kawazoe, N. Kim, R. Fenton Friesen, T. Ferris, F. Quek, and M. Hipwell, "Characterization of medical neck palpation to inform design of haptic palpation sensors," *Sensors*, vol. 25, p. 2159, 03 2025.

[2] P. Gorce and J. Jacquier-Bret, "A systematic review of work related musculoskeletal disorders among physical therapists and physiotherapists," vol. 38, pp. 350–367.

[3] S. Zhang, Z. Chen, Y. Gao, W. Wan, J. Shan, H. Xue, F. Sun, Y. Yang, and B. Fang, "Hardware technology of vision-based tactile sensor: A review," *IEEE Sensors Journal*, vol. 22, no. 22, pp. 21410–21427, 2022.

[4] R. Zhang, U. Yoo, Y. Li, A. Argawal, and W. Yuan, "Pneugelsight: Soft robotic vision-based proprioception and tactile sensing," *The International Journal of Robotics Research*, p. 02783649251378153, 2025.

[5] W. K. Do and M. Kennedy, "Densetact: Optical tactile sensor for dense shape reconstruction," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 6188–6194, IEEE, 2022.

[6] S. Wang, Y. She, B. Romero, and E. Adelson, "Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6468–6475, IEEE, 2021.

[7] M. Wegner, E. Gargioni, and D. Krause, "Classification of phantoms for medical imaging," *Procedia CIRP*, vol. 119, pp. 1140–1145, 2023.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[9] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, and S. Chintala, "Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation," in *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*, ACM, 2024.

[10] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," 2019.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," 2015.

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.