Manipulation of Soft Cloths using DenseTact Optical Tactile Sensors

Ankush Dhawan
Stanford University
450 Jane Stanford Way
ankushd@stanford.edu

Sunny Singh
Stanford University
450 Jane Stanford Way
sanjot@stanford.edu

Abstract

This paper presents a method for robotic cloth layer classification using a custom two-finger gripper equipped with DenseTact 2.0 sensors. The gripper performs a rubbing motion to collect optical flow and net wrench data, processed by a transformer-based neural network. A comprehensive dataset of 300 trials was also collected and made open-source available along with this paper.

Our experiments evaluated different model architectures, showing that a transformer model with optimizer state resets achieved the highest accuracy of 78.7%. Including net wrench data did not significantly enhance performance, highlighting the effectiveness of optical flow features in this task. Code for this project is available in this GitHub repository. The raw video and wrench data as well as pre-processed data is available in this Google Drive Folder.

1. Introduction

In recent years, the advancement of robotics has extended beyond traditional industrial settings to encompass tasks that require interaction with flexible and deformable materials, such as cloth manipulation. The ability for robots to effectively interact with fabrics holds significant promise in various domains, including domestic chores like folding laundry and assistive tasks in elderly care [16]. Moreover, in manufacturing, robots capable of handling cloths can revolutionize everyday processes.

However, despite the progress in robotics, cloth manipulation remains a challenging frontier. Fabrics exhibit complex dynamics, high degrees of freedom, and severe self-occlusions when folded or crumpled, posing significant obstacles for traditional robotic manipulation methods. Conventional approaches often rely on electrical signal processing or mechanical system identification [8], which may struggle to adapt to the nuanced and variable nature of cloth interactions.

To address these challenges, this research project focuses on leveraging cutting-edge technology at the intersection of hardware and software. Specifically, we aim to develop a neural network-based solution to determine the number of layers of cloth between two DenseTact sensors mounted on the fingers of a custom robot gripper. DenseTact sensors, pioneered in Prof. Monroe Kennedy's lab, have the unique ability to capture the subtleties of cloth interactions by utilizing optical tactile sensing to detect shear forces and optical flows based on visual gel deformations.

In summary, this research endeavor aims to push the boundaries of robotic manipulation capabilities by tackling the intricate task of cloth manipulation using advanced sensing technology and deep learning methodologies. Through this interdisciplinary effort, we aspire to not only advance the field of robotics but also contribute to real-world applications such as elderly care and manufacturing.

1.1. Problem Statement

This paper addresses the integrated task of identifying the number of layers of cloth between two robotic fingertips by using optical tactile sensor input. The components of the problem statement for this task are as follows:

- **Design of a Robotic Gripper:** The soft, deformable cloth must be grasped by a robotic gripper to allow for classification. We assume that the gripper interacts with the cloth in a constrained environment, where the cloth sits on a flat table, and the gripper can approach the cloth from any angle. To accomplish this, a novel two-finger gripper using small motors and DenseTact sensors was designed, as explained in 3.1. This gripper is capable of 2-axis motion and can perform a rubbing motion between its fingers to allow for cloth layer classification.
- **Data Collection:** For this task, the data must be manually collected using the new gripper design. For this study, labeled classes for 0 layers (no cloth between the fingers), 1 layer, and 2 layers of cloth between the class were collected. This is further explained in 4.3
- **Layer Classification:** The cameras in the DenseTact fingers will record the deformation in gel surface of the sensors over the time period of the rubbing motion. To classify the layers of cloth between the fingers, the optical flow data and the force data from the rubbing motion will be inputted into a neural network (coded in PyTorch [12]), which will output a label corresponding to the number of layers of cloth.

2. Related Work

Building upon the principles outlined in seminal papers such as "DenseTact: Optical Tactile Sensor for Dense Shape Reconstruction" and "DenseTact 2.0: Optical Tactile Sensor for Shape and Force Reconstruction," we propose to utilize RGB videos of visual gel deformations from DenseTact sensors as training data for our neural network model [4, 6].

In the realm of image classification, ResNet has consistently demonstrated high accuracy by leveraging residual connections to address the vanishing gradient problem, enabling the training of very deep networks [9, 11]. Despite this, recent advancements have shown that transformers, which utilize self-attention mechanisms, can surpass traditional CNNs in various computer vision tasks [15, 10]. An innovative technique that further enhances the performance of transformers is the periodic resetting of the optimizer during training, which helps in maintaining training stability and improving convergence [1, 2].

While existing research, such as the work cited in "Learning to Singulate Layers of Cloth using Tactile Feedback," has explored cloth manipulation using alternative sensor modalities like magnetic ReSkin sensors [14], our approach distinguishes itself by focusing on optical tactile sensing. Additionally, unlike previous attempts that may have struggled with cloth layers beyond a single layer, we aim to develop a robust solution capable of handling multiple layers of cloth, thus broadening the applicability of robotic cloth manipulation.

3. Hardware Setup

The hardware developed for this research project is a two-finger robotic gripper. The components of the gripper communicate via ROS2 to perform robotic cloth layer classification.

3.1. Two-Finger Gripper Design

To enable the robotic manipulation of cloths, a custom two-finger gripper was designed (as shown in Figure 1. For each finger, two lightweight and compact Dynamixel XL330-M288-T motors are used to actuate the joints of the fingers. To emulate a finger pad, a DenseTact 2.0 sensor is attached on each opposing finger. An OpenRB-150 Arduino compatible embedded controller is used to control and actuate the motors in a rubbing motion. Custom 3D printed mounts were designed to integrate the motors and DenseTact sensors together. A single computer ran the central code for controlling the robotic system that also provided power to the

motor control board as well as the DenseTact cameras and LEDs.

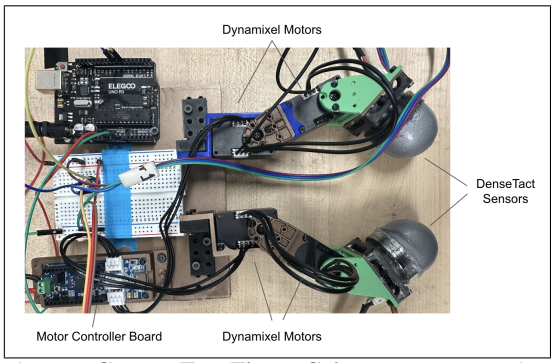

Figure 1. **Custom Two-Finger Gripper**: A custom gripper integrating the DenseTact sensors with Dynamixel motors and 3D printed parts was designed to perform a rubbing motion between two DenseTact sensors together to gather optical flow from a video sequence.

3.2. Densetact 2.0

The DenseTact 2.0 is attached atop the fingers of the gripper, which provide a soft, deformable gel medium and integrated camera and RBG LEDs for optical flow and force sensing. The DenseTact 2.0 was chosen over other designs such as the DenseTact 1.0 [4] and the DenseTact Mini [5] because of its compact design, yet high resolution and hemispherical shape. The DenseTacts can be seen mounted in Figure 1.

The DenseTact 2.0 sensors used for this task include a randomized pattern. This pattern that was stamped on the sensors during fabrication and allows for trackable features during the cloth manipulation. Since many cloths are themselves very smooth with few distinguishable features, having the pattern allows for collecting feature-rich video data where the optical flow (expanded on in 4.1 between frames is particularly informative due to the pattern's movement. Figure 2 shows an image of the camera stream of a the DenseTact 2.0 that depicts the pattern on the gel surface.

In addition to the pattern that helps with detecting the gel deformations, calibrating the sensors following the method outlined in [6] allow for force sensing based on depth image-based point cloud

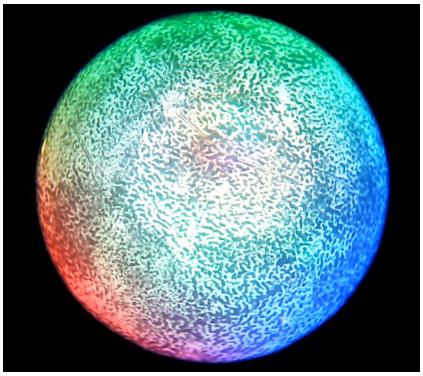

Figure 2. **Patterned DenseTact 2.0**: The DenseTact is patterned to provide rich trackable features for when cloths are rubbed against each other. When the DenseTact's soft gel is pressed, these features move and can be visually and programatically tracked.

generation. Combining the data from both the optical flow and forces during a recorded rubbing motion with either 0, 1, or 2 layers of cloth between the fingers allow for classifying the number of layers of cloth. This calibration allows for the net 6-axis wrench estimation, as explained in 4.2.

4. Methods

For this research, two outputs from the DenseTact were collected to be used as inputs to the network for training: optical flow, and 6-axis wrench.

4.1. Optical Flow

As shown in Figure 2, using a randomized pattern allows for a feature rich image even when the objects that are being manipulated are rather smooth. This enables optical flow to be an effective method of classifying the number of layers of cloth between the two fingers. For this method, the Farnebäck dense optical flow method [7] was used to estimate the motion between two frames. Using this method outputs a matrix, where each element (x, y) is a 2D motion vector (u, v) that indicates the displacement of the points between the consecutive frames. Each vector (u, v) at the every position in the matrix describes how much the pixel at (x, y) has moved in the x-direction (u) and the y-direction (v). To visualize this, we can plot the optical flow in

a quiver plot, where the magnitude and direction of the arrows at each pixel indicate the displacement.

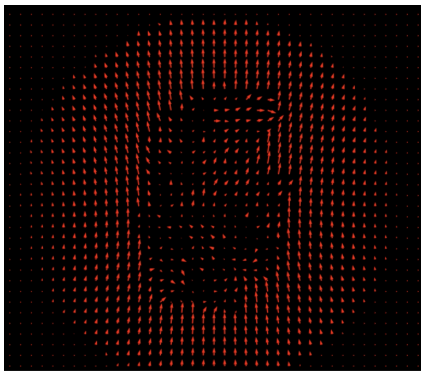

Figure 3. **Dense Optical Flow Quiver Plot**: The above images are subsampled by 32 to provide visual clarity in the plots. The image shows the quivers when the fingers are rubbed against each other. There are high magnitude vectors at the edges of the press, while there are 0 vectors at the center and edges. The angle of the vectors indicates the direction of the rub.

4.2. 6-Axis Wrench

In addition to optical flow data, net force and torque data is potentially informative of the interactions between the DenseTact sensors and the cloth. Calibrating DenseTact sensors following the method in [6] allows for real-time 6-axis wrench estimation ($F_x, F_y, F_z, \tau_x, \tau_y, \tau_x$) for a given RGB image frame. Once calibrated, feeding an RGB image from the DenseTact in a forward pass through the calibrated network returns the wrench estimation. This data was collected per image frame and recorded for every trial in the dataset.

4.3. Dataset Collection

For conducting this task, a custom dataset was collected to provide data to train the classifier on. The dataset for this research comprises RGB videos of visual gel deformations captured by the DenseTact sensors, as well as 6-axis wrench data consisting of force and torque data. This data was recorded using a rubbing motion between the two fingers of the gripper. Real-time video streams of the DenseTact were collected along with the 6-axis wrench data. This provides a comprehensive representation of the tactile interactions between the sensors and various cloth configurations. Ground truth labels indicating the number of cloth layers in each video were noted during data collection to facilitate model training and evaluation. Overall, a total of 100 labeled trials for each cloth layer class were collected, giving a total of 300 examples across the entire dataset. Each video trial was recorded at 10Hz, and sent for pre-processing before training. This data is made available in this Google Drive Folder for open-source use.

4.4. Data Pre-Processing

In order to provide meaningful data to the network, the optical flow data was sub-sampled in frequency so that the magnitudes and directions of the flow vectors were sufficiently large. This is necessary because optical flow data is recorded per consequent frames, and the camera runs at 10 FPS, meaning that between two frames at that high of a frequency, there will not be enough of an image difference for the optical flow calculation to provide a meaningful result. After some experimentation with the frequency and visual inspection of the vectors, optical flow calculations were performed at 2Hz since that provided a high enough frequency where the flows accurately represented the rubbing motion, but also was a low enough frequency where the magnitudes of the flow were large enough to be useful in a network. To the same effect, the optical flow measurements were average pooled by 6 (a window of 36 pixels). After pooling, the dimension of the optical flow inputs was 128 by 170 by 2, indicating the image height, image width, and the number of channels (magnitude and direction). To ensure consistency between the inputs, the net 6-axis wrench data was also collected at 2Hz.

4.5. Network Architecture

The network architecture proposed in Figure 4 is designed to classify layers of cloth using a combination of optical flow and force data. At a high level, the proposed architecture uses feature extractors for the optical flow data and the wrench data, and the following transformer takes in these fea-

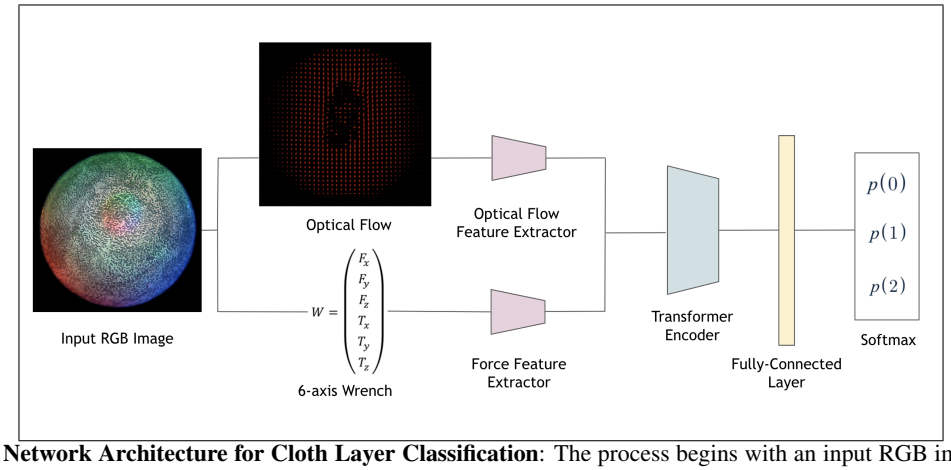

Figure 4. **Network Architecture for Cloth Layer Classification**: The process begins with an input RGB image, from which optical flow data and 6-axis wrench data are simultaneously extracted. The extracted features are then combined and fed into a transformer encoder, which integrates the spatiotemporal and tactile information. The final classification is performed using a fully-connected layer followed by a softmax layer, outputting the probabilities for 0, 1, or 2 layers of cloth.

tures as well as their temporal representation to classify the inputs as one of the three classes, noted by the highest probability after the softmax function. For this work, cross entropy loss and an Adam optimizer were used since they have shown to be effective in similar tasks [13]. The pre-processing steps to generate the optical flow and wrench data are described in detail in Section 4.4

4.5.1 Optical Flow Feature Extraction

The optical flow data, derived from the input RGB images, undergoes feature extraction to capture essential motion patterns that signify cloth layers. This step is crucial because the optical flow provides rich spatiotemporal information about the surface interactions over time. The optical flow feature extractor was designed using either a CNN or a ResNet backbone. Using a CNN backbone over a ResNet backbone provided computational efficiency since the model is smaller, but a ResNet backbone may help to mitigate vanishing or exploding gradients, allowing for a deeper network to be trained [11]. The affect of this design choice is described in detail in 5.2.

4.5.2 Force Feature Extraction

At the same time as optical flow data collection, features of the 6-axis wrench data, which measure net forces and torques in three dimensions, are also extracted. This tactile information helps in understanding the physical properties of the cloth, such as its stiffness or thickness, which are not easily discernible from visual data alone. Adding this feature can help to distinguish between classes that look more visually similar. The force feature extractor designed here was simply designed with fully connected layers since the input space per frame was small (only 6 values per frame) compared to the optical flow data, and hence further encoding was not deemed necessary.

4.5.3 Transformer Encoder

The extracted features from both optical flow and force data are then fed into a transformer encoder. The transformer is well-suited for this task because it can effectively capture long-range dependencies and contextual information between different parts of the input data. By doing so, it integrates the spatiotemporal and tactile features into a coherent representation, allowing the network to understand

the image feature time-dependence when a rubbing motion is conducted. The transformer used here was designed with a 64 dimensional input/output embedding to balance feature representation and efficiency, 8 attention heads for the multi-headed attention mechanism to allow the model to attend to many different parts of the input sequence simultaneously, 3 encoder layers to provide enough depth to capture complex patterns, and a 2048 dimensional feedforward network to ensure that the model has enough capacity to learn complex transformations [2].

4.5.4 Classification

The integrated features from the transformer are passed through a fully-connected layer, which condenses the information into a form suitable for classification. Finally, a softmax layer outputs the probabilities of the input belonging to each class (0, 1, or 2 layers of cloth). The highest value among the softmax outputs (corresponding to the highest probability class) was chosen as the predicted class during evaluation.

4.5.5 Architecture Considerations

This architecture is particularly effective for the cloth layer classification problem due to its ability to integrate multiple modalities of data (optical flow and wrench) and leverage the strengths of transformer models in capturing complex relationships within the data. Optical flow provides detailed motion information, which is crucial for understanding the dynamics of cloth interaction. Meanwhile, wrench data offers complementary tactile insights that enhance the model's ability to differentiate between subtle variations in cloth layers.

Using a transformer encoder is advantageous because it excels in handling sequences and learning contextual dependencies, which are critical when working with time-series data like optical flow. This approach is more effective than alternatives that might only use CNNs or single-modal data, as it combines the strengths of visual and tactile sensing to improve classification accuracy.

5. Experiments

5.1. Encoder Validation

Before designing the full transformer network classifier, an encoder-decoder reconstruction study was completed to determine if the optical flow feature extractor encoder captures the important latent features in each image. Iterating through a number of different architectures, an architecture of four consecutive convolutional layers followed by ReLU activations was able to achieve near-perfect reconstruction, as shown in Figure 5. This study validates that this encoder choice is effectively represents the latent features of the input images, and hence this architecture was used for building the transformer [10].

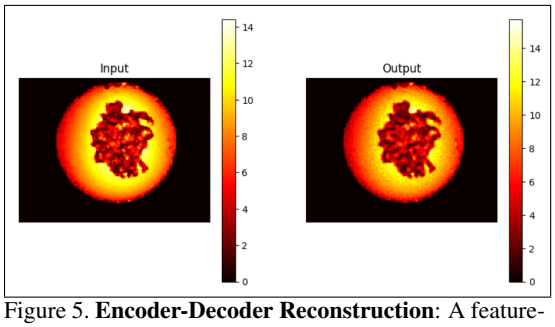

Figure 5. **Encoder-Decoder Reconstruction**: A feature-rich image heatmap is nearly perfectly reconstructed after passing through an encoder-decoder architecture. The input image is passed through the encoder-decoder architecture and is reconstructed in the output image.

To further validate that the encoder was working well, a sanity check study on just differentiating two classes was conducted. For this, all the trials corresponding to the 1 layer class were removed from the dataset. Then, the encoder was given a 75-25 split was done to divide the dataset into train and test, and the encoder's parameters were trained on the training set. When evaluated on the test set, the encoder achieves 100% accuracy when classifying between just two classes. The T-SNE plot [3] in Figure 6 shows that there are two distinct sets between the two classes, meaning that it should be very easy to draw a decision boundary between the classes. From this study, we validate that the encoder design is effective and can possibly extend to

Classifier	Inputs	Architecture Backbone	Optimizer State Reset (Every 50 Epochs)	Epochs Trained	Test Accuracy (%)
Naive CNN	Optical Flow	CNN	No	500	61.3
Transformer	Optical Flow	CNN	No	250	42.7
Transformer	Optical Flow	CNN	Yes	250	75.2
Transformer	Optical Flow	CNN	Yes	500	78.7
Transformer	Optical Flow	ResNet	Yes	250	73.3
Transformer	Optical Flow + Net Wrench	CNN	Yes	500	78.7

Table 1. Comparison of Classifiers with Various Inputs and Architectures

work on the desired three class problem.

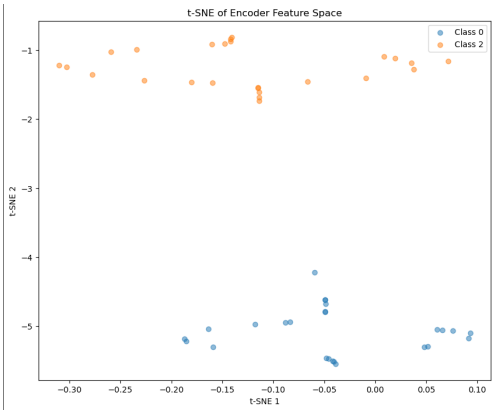

Figure 6. **Encoder Validation T-SNE Plot - 2 Classes**: This T-SNE Plot of the latent vector output from the optical flow feature extractor shows distinct sets between the two classes.

5.2. Ablation Study

To build on top of the encoder design from 5.1, an ablation study was conducted to evaluate various approaches to the classification problem. Our approaches focus on exploring a wide set of design choices while balancing model complexity. The results of this study are displayed compactly in 1, and are further elaborated on here.

5.2.1 Naive CNN Model

The naive CNN model was used to test how the raw optical flow feature extractor performed on the 3 class classification problem. As explained above in 5.1, this CNN model was designed with four consecutive convolutional layers with ReLU activations. With a 75-25 train test split, this model achieved an accuracy of 61.3% on the test set.

5.2.2 Transformer Model

Using the full architecture proposed in Figure 4 without the wrench data input resulted in unstable training, noted by Figure 7. When evaluated on the test set, this model achieved an underperforming 42.7% accuracy because of the instability during training.

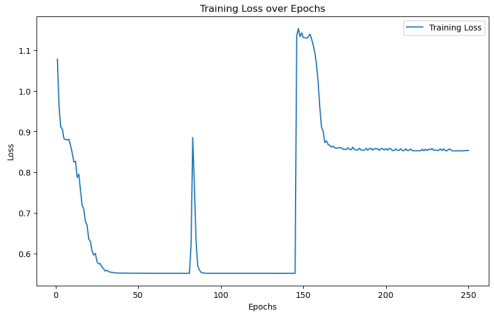

Figure 7. **Transformer Loss without Optimizer Reset:** Initial results from the transformer encoder architure show unstable training from the loss function.

To improve training stability, optimizer state resetting every 50 epochs was used to reset the state of the Adam optimizer [1]. This resulted in more stable training, as shown in Figure 8. This model achieved 75.2% when trained for 250 epochs, and when trained for 500 epochs, the accuracy improved to 78.7%. Figure 9 shows the 3 class T-SNE plot for this model, where it is evident that the latent vector spaces between the 2 layer class and the other classes is easily distinguishable. However, distinguishing between the 0 layer class and the 1 layer class is not possible, making it very difficult to draw a decision boundary between all the classes.

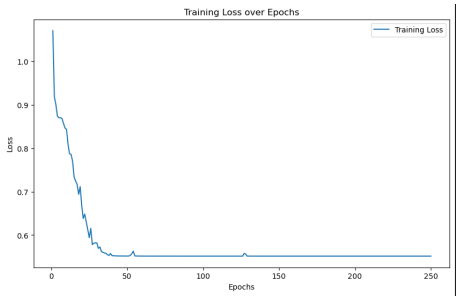

Figure 8. **Transformer with Optimizer Reset:** Using optimizer reset improves the smoothness of the loss.

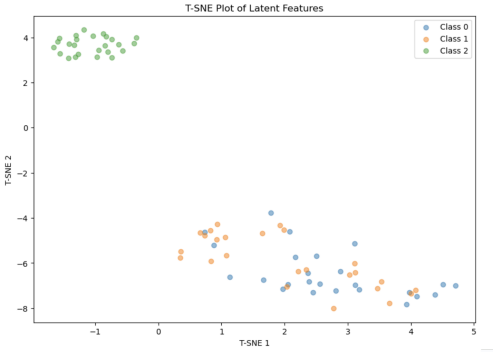

Figure 9. **Transformer T-SNE - 3 Classes:** Feature spaces shows that the 0 and 1 classes are difficult to distinguish, while the 2 layer class is easily distinguishable.

5.2.3 ResNet Optical Flow Feature Extractor

The optical flow feature extractor backbone was also tested with ResNets instead of CNNs. Using the ResNet backbone increased training time because of the model complexity, limiting training to 250 epochs. This resulted in 73.3% accuracy on the test set, indicating that the ResNet backbone did not have a benefit over the CNN backbone even though the model is more complex, highlights the trade-off between model complexity and training feasibility.

5.2.4 Multi-Input Model

After completing comprehensive studies on optical flow, we tested adding net wrench data to the model. The full architecture in Figure 4 was used, incorporating both optical flow and net wrench inputs. T-SNE plots (Figure 9) showed nearly indistinguishable features between the 0 and 1 layer classes. Despite this, the model's accuracy remained at 78.7%, indicating that the additional wrench data did not improve performance compared to using only optical flow data.

6. Conclusions and Future Extensions

From the experimental results presented in the previous sections, several key results and conclusions can be drawn. First, optimizer state resets help to stabilize training and increase the performance of transformer models for this classification task. Second, transformer models, when properly tuned, outperform naive CNN models, demonstrating their potential for complex classification tasks involving optical flow data and time dependence. Third, the inclusion of additional data modalities (Net Wrench) does not always guarantee improved performance, emphasizing the need for careful feature selection. Finally, the models designed in this study are capable of distinguishing the 0 layer class from the 2 layer class, but are not capable of distinguishing the 0 layer class from the 1 layer class.

While a classification accuracy of nearly 80% is promising, there are a few potential extension areas possible for improving this result. First, experimenting with different types of rubbing motions may help to improve feature differences. As depicted in the T-SNE plots, the feature spaces for the 0 layer class and the 1 layer class are very similar, where a different rubbing motion may help to improve differentiating the feature spaces. One such method would be to incorporate a circular rubbing motion rather than the linear rubbing motion, as that could provide more feature rich data when comparing the 0 layer class to the 1 layer class.

In addition, exploring how these models extend to a larger output space, such as for classifying 3 and 4 layers of cloth is also potentially interesting and useful. To the same effect, using multiple different types of cloths to test the model's generalizability to other cloth types is also an important next step.

7. Acknowledgements

Singh designed and trained the initial CNN model and Dhawan trained the Transformer model along with implementing the optimizer reset and pre-processing both the optical flow and net force data. Singh wrote the Introduction, Experiments, and Conclusion sections of the paper while Dhawan wrote the Hardware Setup, Method, and Future Work sections. Hardware was sourced from Prof. Monroe Kennedy's, the Assistive Robotics and Manipulation Lab. GPU's from the ARMLab were also valuable in accelerating the model training and testing.

References

[1] K. Asadi, R. Fakoor, and S. Sabach. Resetting the optimizer in deep rl: An empirical study. *Advances in Neural Information Processing Systems*, 36, 2024.

[2] Y. Bazi, L. Bashmal, M. M. A. Rahhal, R. A. Dayil, and N. A. Ajlan. Vision transformers for remote sensing image classification. *Remote Sensing*, 13(3):516, 2021.

[3] T. T. Cai and R. Ma. Theoretical foundations of t-sne for visualizing high-dimensional clustered data, 2022.

[4] W. K. Do and M. K. I. au2. Densetact: Optical tactile sensor for dense shape reconstruction, 2022.

[5] W. K. Do, A. K. Dhawan, M. Kitzmann, and M. K. I. au2. Densetact-mini: An optical tactile sensor for grasping multi-scale objects from flat surfaces, 2023.

[6] W. K. Do, B. Jurewicz, and M. K. I. au2. Densetact 2.0: Optical tactile sensor for shape and force reconstruction, 2023.

[7] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In J. Bigun and T. Gustavsson, editors, *Image Analysis*, pages 363–370, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[8] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[10] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022.

[11] S. Mohapatra, N. Abhishek, D. Bardhan, A. A. Ghosh, and S. Mohanty. Comparison of mobilenet and resnet cnn architectures in the cnn-based skin cancer classifier model. *Machine Learning for Healthcare Applications*, pages 169–186, 2021.

[12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[13] U. Ruby and V. Yendapalli. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10), 2020.

[14] S. Tirumala, T. Weng, D. Seita, O. Kroemer, Z. Temel, and D. Held. Learning to singulate layers of cloth using tactile feedback, 2022.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[16] Y. Wang, Z. Sun, Z. Erickson, and D. Held. One policy to dress them all: Learning to dress people with diverse poses and garments, 2023.