Estimating Forces from Vision-based Tactile Sensor Images

Jiaxiang Fu
Stanford University
jiaxiang@stanford.edu

Jeremie Gabor
Stanford University
jeremieg@stanford.edu

Julia Di*
Stanford University
juliadi@stanford.edu

Abstract

While deep learning has been employed in the context of vision-based haptic sensors, discussion on the influence of model design, hyperparameters, and transfer learning is limited. We perform our exploration by studying the task of predicting a 3 dimensional force vector from a haptic sensor image. We use data previously collected and generously made available for future study by [5].

In our study, we find that the ResNet-18 model, originally used by the owners of the dataset, performed best compared to the ResNet-50 model and ViT. With proper hyperparameter tuning, feature selection and transfer learning, our best model was able to beat the replicated baseline by a comfortable margin.

Across all model architectures, transfer learning is particularly beneficial even though the sensor images differ substantially from typical classification imagery. Ablation studies demonstrate that weakened inputs (e.g. lower resolution, distorted aspect ratio) have visible negative impact on model performance, and number of transformer layers in ViT models has significant influence on their performance.

1. Introduction

Advances in computer vision have made vision-based haptic sensors a promising approach over non-vision-based techniques. Amongst many other tasks, one of the use cases for these haptic sensors is to perform contact force estimation [5]. This is crucial for the development of robotics because the inability to estimate forces accurately is one of the reasons why robots are not yet as dexterous as humans [4]. Some sensors have been designed with this in mind and provide physical patterns on their inner layers for the camera to track changes from an untouched baseline [9]. We aim to explore the problem of pure vision-based force estimation without the presence of these physical patterns to aid the sensor. Specifically, we use deep learning approaches without any hand-crafted feature detectors. This area has been explored in the past [5] but discussion about task performance across model architectures, different training techniques, etc., is not available to our best knowledge.

Typically, a vision-based tactile sensor consists of at least one camera, a neural network model, and other supporting structures such as sensor skeletons and lighting [4, 5, 13]. The neural network model consumes the images taken by the camera as input, and outputs the estimated contact forces. In this study, we developed and trained our models on the dataset provided by [5]. Specifically, the input to our models is a single image, or for some model variants, derivations/transformations of multiple images, and the main output is the 3 dimensional components of the contact force vector applied to the tactile sensor. The images are collected from the Insight haptic sensor which "combines photometric stereo and structured light to detect the deformation of a full three-dimensional cone-shaped surface" [5]. The three dimensional contact vector is "predicted relative to the surface in normal direction F_n and two shear directions F_{s1} and F_{s2}".

2. Related Work

With the rise of standard image sensors, many new vision-based tactile sensors have recently emerged, and learning-based models are often introduced together with the new sensor designs [4, 5, 13] for force estimation and other related tasks. Particularly, Gelsight (2017) [13] used a CNN adapted from VGG-16 [10] pretrained on ImageNet to estimate contact forces. The only adaptation they made was replacing the last fully-connected layer. DIGIT (2020) [4] trained an autoencoder where both the encoder and decoder are based on ResNet-18 [6]. They then trained another neural network dynamics model using the representations produced by the encoder for robot manipulation. Further, Kakani (2021) [7] applied transfer learning on a VGG-16 adapted network for force estimation. In an apparent deviation from common practices, they only used pretrained weights on the first convolutional block (conv-batchnorm-relu) and the first fully-connected block (fc-batchnorm-relu-dropout) (they have an additional fully-connected layer with

*External Mentor, not a CS231n student

custom dimensions). However, the authors did not provide specific analysis on this choice and its impact on performance. Last but not least, Insight (2022) [5] used an adapted ResNet-18 to estimate contact forces. Interestingly, the authors did not use any pretrained models and still achieved state-of-the-art accuracy.

More broadly, many recent models have been developed for computer vision tasks and are shown to improve state-of-the-art performances. We are particularly interested in models that are proven to perform well across different domains, as our problem domain is unique and distinct from most of the typical computer vision tasks. One such example is ResNet [6], which has proven to achieve state-of-the-art performances in image classification, object detection, object localization as well as image segmentation when it was invented. It has already been well recognized in the field of vision sensors and force estimation [4,5]. The Transformer architecture [12] is another breakthrough in deep learning. In particular, the Vision Transformer (ViT) dropped all convolutional layers in favor of "a pure transformer applied directly to sequences of image patches" [3]. Even though ViT was initially applied to image classification only, subsequent follow-up studies have shown good results by applying ViT to other tasks. For example, Caron et al. in 2021 [1] used a self-supervised learning method that the authors named DINO to train ViT and showed its capability of image segmentation. MaxViT in 2022 [11] demonstrated favorable performances on object detection, visual aesthetic assessment as well as a strong generative modeling capability, on top of achieving state-of-the-art performance on image classification. As such, we believe both ResNet and ViT can be promising backbone models for the force estimation problem.

3. Dataset

We base our work on the dataset provided by [5], which contains 187,358 samples of images with corresponding force measurements. We used a 150k, 10k, 27k split across our training, validation and testing sets respectively. The dataset was collected by using a testing rig mounted with an indenter to probe the Insight sensor [5]. A force-torque sensor was used to measure the force vector applied to the indenter to act as ground truth for the supervised learning problem [5].

The indenter was programmed to apply forces between 0 and 2 newtons while traversing the contact area of the sensor. Figure 1 shows the distribution of the applied forces. We note that some of the measured forces extend beyond 2N which can explained due to mechanical errors when collecting the dataset.

The sample images obtained by the sensor are 308 x 410 jpg images. In addition to the per sample images, the dataset also included baseline images to be used as references. These include a skeleton reference image - an image of the sensor skeleton without the elastomer layer, and an idle image - an undeformed image of the sensor when it is not in contact with anything. Figure 2 displays a sample of each as well as a sample image when a force is exerted onto the sensor and its difference when compared to the reference image.

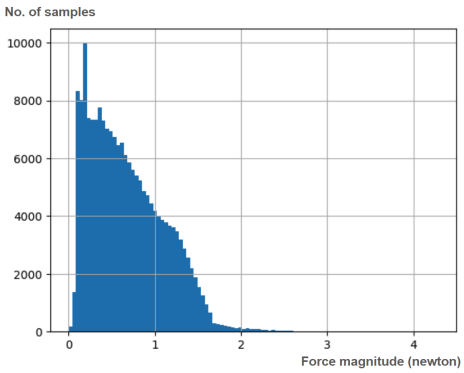

Figure 1. Distribution of force magnitude.

While normalizing data is often recommended to aid the training stability of deep learning models, we avoided any typical normalization techniques (e.g. subtracting by the mean of the batch, dividing by the standard deviation of the batch) to avoid losing predictive information for the force estimation. As our primary goal was to predict the three dimensional force applied to the sensor, maintaining unique information about each sample was important to accurately predict the magnitude of each sample force. Instead, as was done in [5], the only preprocessing we performed was to subtract the idle image from each sample image. We also experimented without this preprocessing and discuss the results in Section 5.

4. Methods

4.1. Baseline

Our baseline model is provided by [5], which is a CNN adapted from ResNet-18 where the global average pooling layer and fully-connected layer at the end were replaced by a max pooling layer and a new fully-connected layer respectively. Notably, although the authors provided their complete model architecture, they did not provide the learned weights of their final model. Further, some details in the training procedures are also missing. Most prominently, the authors used a custom train/validate/test split but did not provide details. As a result, our baseline model is our best attempt of replicating the model by [5] and is an imperfect replication.

The baseline model takes a 6-channel 2D input and produces a 6-channel 1D output. The input is a concatena-

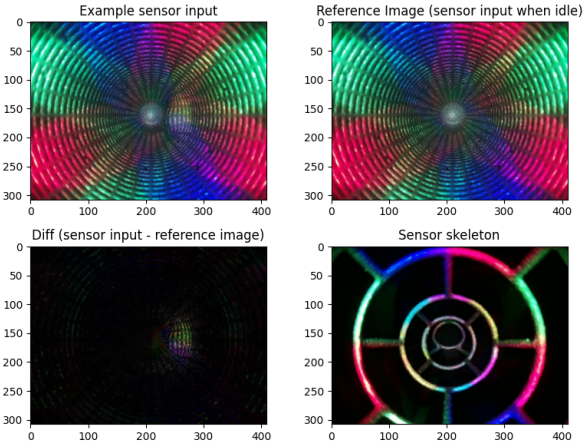

Figure 2. Sample sensor input, idle undeformed sensor image, difference image (i.e. sensor input - idle image), and sensor skeleton image.

tion of two images, with the first being the difference between sensor readings and the idle image, and the second being a static image of the sensor skeleton (See Section 3 for details). Figure 2 gives an example of each. The 6-channel output represents the 3D coordinates of the contact point and the 3 components of the contact force $(P_x, P_y, P_z, F_n, F_{s1}, F_{s2})$. As we focus on estimating the force magnitude and direction only, the contact point coordinates (P_x, P_y, P_z) are therefore not our primary interest and only used as auxiliary tasks to help with model optimization.

We adapted the starter code provided by [5] to replicate this baseline model. The code can be downloaded from the main page for the paper [5] in the official website of the Nature journal. As mentioned, the starter code includes the model implementation, data loading, and model training. However, details about train/validate/test split, and evaluation metrics are missing, and we had to implement these among other minor adjustments for the code to work. The baseline model and all subsequent models we implemented are written in PyTorch [8].

4.2. Our Approach

4.2.1 Baseline Replication & Tuning

Once we achieved reasonably similar results on the baseline model to those described in [5], we experimented with hyperparameter tuning. First, the baseline model scales the labels (i.e. the force components) up by k = 1024 times in the training process. The authors in [5] did not explain why this technique was employed. However, it is clear that removing this scaling factor while keeping everything else the same would lead to much worse results (See results in Section 5.2). We attribute this performance improvement to the fact that the scaling factor can affect the effective learning rate, among other things. Given the choice of loss function MSE, scaling up the labels lead to a much larger scale of losses. While the gradients at each step are scaled up by a factor of k, the loss values are scaled up by k^2 times. Considering the optimizer used (Adam) adjusts for the second moment, this makes the effective learning rate much smaller than the actual learning rate used. The exact effect, however, is difficult to reason about due to the interaction of this change with regularization weight, parameter initialization scale, and optimizer strategies. Due to the unusual and unexplained nature of this change, we consider it a hack introduced by the authors to help with model optimization. To make our models more general and consistent when we experiment different model architecture (e.g. larger models, attention-based models), we decided to remove this hack, and instead search for more appropriate learning rates.

4.2.2 Larger Models & Attention-based Models

It is a well known fact in the deep learning domain that with proper care, larger models and more complicated models often lead to better performance. Our baseline model is a ResNet-18 adapted to fit the input and output dimensions, a relatively small model compared to many state-of-the-art computer vision models. We therefore experimented with larger models and attention-based models as the backbone. Specifically, we used modified versions of ResNet-50 [6] and Vision Transformer [3]. For ResNet-50, we retained the exact same adaptations as the baseline ResNet-18 except for the hacks described above. ResNet models are characterized by their residual connections which are introduced to allow gradients to flow backwards more smoothly through the newtork. More specifically, the residual-connections create a gradient path around the residual blocks allowing the gradient to be backpropagated without passing through multiple activation functions. This helps to avoid typical challenges encountered while training deep-learning models such as exploding or vanishing gradients. For attention based models, we experimented with ViT as introduced in [3]. ViT follows a transformer encoder architecture and is pre-trained on the ImageNet-21k dataset. The 224 x 224 input images are split into patches, transformed to a linear embedding space, and added to a position embedding. Of note, the nature of our supervised learning task does not lend itself directly to the typical advantages introduced by transformer based models. For example, transformer based models are often praised for their ability to model long term relationships (temporally and spatially) as well as their scalability. Given the modest size of our input data and the fact that areas of interest in our images are very spatially concentrated, we may not benefit as much from these advantages. That being said, trans-

former models performance across a wide range of tasks make exploring them in our problem well worth-while.

4.2.3 Transfer Learning

Pre-trained weights are publicly available for all of the backbone models we used: ResNet-18, ResNet-50 and ViT. However, there are two challenges to transferring these pre-trained weights to our model. First, our problem domain differs significantly from the domain where the pre-trained models are trained. Typically, the publicly available pre-trained models are trained with ordinary daily-life images in an image classification task. Our model however is expected to perform regression tasks on a specific type of tactile sensor images. The important features and representations useful for our task may be completely different. Second, the number of input and output channels for our model is different from these models. As a result, the first and last layer of the backbone models have to be dropped. Importantly, the first layer usually contains low-level features that all subsequent layers are based on. Losing the first layer can be significantly detrimental to model performance.

To tackle the second challenge, we tested trimming the model down by removing some inputs that we think are less important. Specifically, we dropped the sensor skeleton image from the model input (See Section 4.1 for details about model input). The trimmed model takes input of the same format as typical images which is 3 channels rather than the 6 channels required by our baseline model. With this trimming, we are able to keep the pre-trained weights for the first layer. We ran experiments for the trimmed model both with and without transfer learning to test the effect of trimming and transfer learning on the model performance.

4.2.4 Ablation Study

Inspired by [2], we conduct two ablation studies to understand how much the various aspects of our models influence the model performance. The first study focuses on weakening model inputs such as lowering image resolutions, and the second study explores the effect of varying transformer model complexity.

Ablation study of input features. We conduct ablation study to explore three different aspects of the input features, namely image resolution, aspect ratio, and input channels. (1) The original images in the dataset have a resolution of 308 x 410. To test the effect of resolution, we resized the images with anti-aliasing down to 50% in both dimensions from 308 x 410 to 154 x 205. Then, we resized them back to the original resolution of 308 x 410 with anti-aliasing in order to keep the model architecture the same. Note that this process does not reconstruct the original images, but rather results in an interpolation of the lower resolution images. The whole process causes the inputs to lose useful information. (2) To test the effect of distorted aspect ratio, we resized the images to 224 x 224 with anti-aliasing. We chose this particular dimension because our ViT backbone takes 224 x 224 images only, so we can reuse this analysis to breakdown the source of performance gain/loss when comparing models. (3) To test the effect of the additional input required by the baseline model, we ran experiments with less input channels. As described in Section 4.1, our baseline model takes 2 concatenated images as input, the second of which is a static images of the sensor skeletons. We ran an experiment with the skeleton image removed.

Model	LR	MAE Mag.		MAE Dir.	
		Train	Eval	Train	Eval
Baseline	1e-3	-	0.054	-	9.46
RN-18-no-hacks	1e-3	0.22	0.22	23.4	25.2
RN-18-no-hacks	1e-5	0.054	0.057	9.35	9.62
RN-18-no-hacks	2e-5	**0.052**	**0.055**	**7.88**	**7.91**
RN-18-no-hacks	5e-5	0.059	0.059	8.77	8.76
RN-18-no-hacks	1e-4	0.071	0.067	10.52	10.08

Table 1. **Evaluation results for baseline model and its variants**

Ablation study of ViT layers The original model described in [3] is composed of an embedding head which converts the image patches to linear embeddings, 12 ViT layers, and a linear classifier. Each ViT layer contains a self-attention model with a linear layer. Chi et al. [2] found that increasing the number of attention heads to the ViT layers, and thus increasing model capacity, led to better performance. Inspired by this result, we wanted to understand the impact of model capacity on performance for force estimation tasks. We experimented with 4 separate ViT models, each with a different number of ViT layers.

5. Results & Analyses

5.1. Evaluation Metrics

Similar to [5], we evaluate our model primarily on force estimation accuracy, defined as the mean absolute error (MAE) of the force magnitude and direction measured in Newton (N) and degrees (°) respectively.

$$MAE_{magnitude} = mean(abs(\|F_{true}\| - \|F_{pred}\|))$$

$$MAE_{direction} = mean(abs(\arccos(\frac{\langle F_{true}\ ,\ F_{pred}\rangle}{\|F_{true}\| \times \|F_{pred}\|})))$$

5.2. Baseline & Tuning

The evaluation results of our baseline model and its variants are summarized in Table 1. It is clear that removing the hack introduced by the authors while keeping the original learning rate does not perform well. As hypothesized,

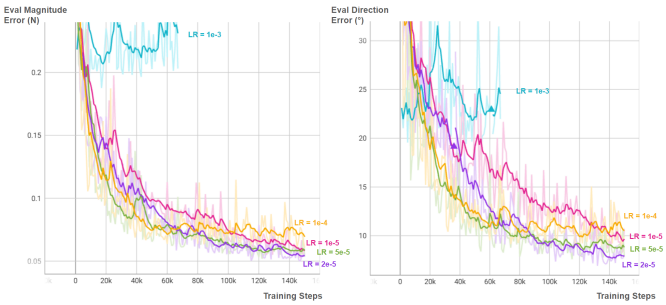

Figure 3. **Evaluation curves of ResNet-18 models with different learning rates.** Learning rate of 2e-5 yields the best performance. Models don't converge in time with lower learning rate (e.g. 1e-5) and converge to a higher error with higher learning rates (e.g. 5e-5, 1e-4) and usually results in higher variance.

lowering the learning rate improves the model performance with the hack removed. We found that a learning rate of 2e-5 performs the best. Figure 3 shows the learning curves of these models. Clearly, with LR=1e-5 the model is still yet to converge, and with higher learning rates the model converges to a higher error and has larger variance.

With appropriate learning rate, we were able to obtain comparable performance as compared to the baseline even without the hack introduced by the authors. In fact, our model achieved much better force magnitude error than the baseline.

5.3. Larger Models & Attention-based Models

We summarize the results of larger models and attention-based models in Table 2. Note that all experiments here were performed without the k=1024 scaling of the output force values. Moving to the larger ResetNet-50 model, we immediately observed that the 1e-3 learning rate employed for the baseline model was not going to converge. Larger models offer stronger model capabilities while requiring the training process to navigate a more complex landscape. This emphasizes the need for an appropriately scaled learning rate. We found reasonable performance with lr=2e-5 where we achieved 0.063N MAE for the magnitude and 8.37° MAE for the angle, the latter of which is better than the baseline.

For the attention-based models, we were unable to achieve similar performance to the ResNet models. First, we note that we used the modified ViT model with 4 ViT layers for this experiment as indicated by the ViT-4 model name. We tested with 1e-5 and 3e-5 learing rates and observed similar results for both mae mag. and mae ang., though slightly better for 1e-5. Given these preliminary results, the ResNet based models appear to be better suited for our desired task. That being said, as discussed in Section 6, we believe this area is one where further investigation would be worth-while. We trained these attention-based models

Model	LR	MAE Mag.		MAE Dir.	
		Train	Eval	Train	Eval
Baseline	1e-3	-	0.054	-	9.46
ResNet-18	2e-5	**0.052**	**0.055**	**7.88**	**7.91**
ResNet-50	1e-3	0.21	0.19	24.0	24.1
ResNet-50	2e-5	0.057	0.063	8.74	8.37
ViT-4	1e-5	0.13	0.13	16.8	15.3
ViT-4	3e-5	0.15	0.15	18.5	17.1

Table 2. **Evaluation results Larger Models & Attention-based Models**

Model	Pretrain	MAE Mag.		MAE Dir.	
		Train	Eval	Train	Eval
Baseline	No	-	0.054	-	9.46
RN-18	No	0.052	0.055	7.88	7.91
RN-18	Yes	0.052	0.053	7.52	7.66
RN-18-trimmed	No	0.049	0.060	8.02	9.32
RN-18-trimmed	Yes	**0.045**	**0.048**	**6.85**	**7.30**
RN-50	No	0.057	0.063	8.74	8.37
RN-50	Yes	0.055	0.053	8.12	7.82

Table 3. **Evaluation results for different models with and without transfer learning**

for the same number of epochs as the ResNet models and while the loss was decreasing at a slower rate, it still appeared to be going down. In summary, our limited experimentation shows that given the same amount of epochs, the ResNet based models were able to achieve better performance on this regression task than the attention-based models.

5.4. Transfer Learning

As shown in Table 3, across all the different model types we experimented, initializing models with pre-trained weights consistently yields better performance. Even though the input domain we deal with is vastly different from the domain that these pre-trained models are trained on, transfer learning still helps.

Notably, after trimming the sensor skeleton image input, the model performance dropped compared to ResNet-18 with original input. However, as described in Section 4.2.3, the input trimming allowed us to reuse the pre-trained weights for the first layer. As a result, the trimmed model outperformed the ResNet-18 with original input when transfer learning is enabled for both. In fact, the former is our overall best model and outperformed all other models and configurations we experimented.

5.5. Model Visualization

To understand what our models have learnt, we performed network visualizations on some of the models we

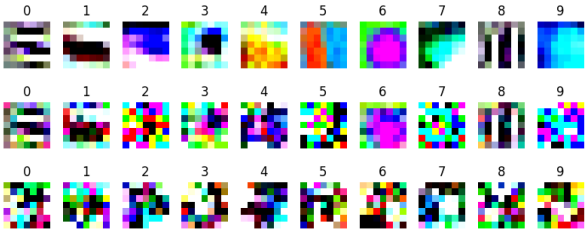

Figure 4. **Visualization of the first 10 convolutional filters in the first layer of select models.** The first row is the ResNet-18 model pre-trained on ImageNet-1k without any finetuning. The second row is our finetuned model using the same pre-trained weights but further finetuned on our dataset. The third row is a model trained on our dataset from scratch without any pre-trained weights. Visualizations of all 64 filters are available in the Appendix. 7.1

have trained. Specifically, we plotted the weights of the first convolutional layer of selected models, and generated saliency maps for our overall best model.

Figure 4 shows the learnt convolutional filters in the first layer of (1) the pre-trained model, (2) model trained with pre-trained weights, and (3) model trained without pre-trained weights, respectively. While the pre-trained model has primarily learnt edges and color blobs, the model trained from random weight initialization primarily learnt amorphous mix of colors. This further confirms that our earlier statement that our problem domain differs significantly from pre-trained models typically trained on ImageNet. Further, it is difficult for humans to interpret the amorphous-mix-of-color filters learnt from scratch. This is understandable as it would be challenging for humans to describe concretely what kind of low level features to look for to determine the force[1]. Interestingly, the model trained with pre-trained weights retains some of the pre-trained features almost intact (e.g. Filter 0, 6, 8) while completely scratching the others (e.g. Filter 2, 4, 5). This shows that even though the two domains are vastly different, some of the features can still be reusable and they do help improve performance.

Figure 5 displays the saliency maps of our best model (RN-18-trimmed with transfer learning) as well as the input diff image for reference. We have included saliency maps for model checkpoints both at an early stage of training and after the model is fully converged. At the early stage of training, the model's "attention" is spread out to the entire input diff image. After the model is fully trained, the model learned to focus on specific areas of the image. In a perhaps counterintuitive way, the model does not focus on where the input signal is the strongest, but rather chooses to focus on the areas around it. Indeed, changing the force applied to the sensor, magnitude or direction, has the most impact on

[1]The size of deformation is one important feature obvious to humans but that's not something that fixed-sized conv filters can learn

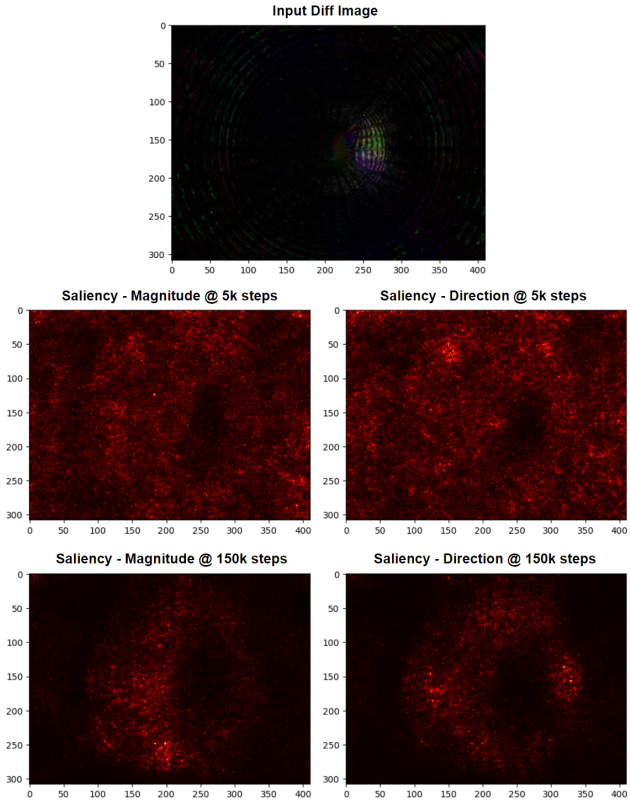

Figure 5. **Saliency map of our best model**. The top row is the input diff image (i.e. sensor input - idle image) for reference. The second row are saliency maps with respect to the force magnitude prediction and the force direction error, taken at an early stage of training (5k steps). The bottom row are saliency maps taken when the model has fully converged (150k steps).

the areas around the point of contact. For example, if we increase the force magnitude, the surrounding areas would transition from completely black to non-zero inputs. The logical converse is also true. If the meaningful areas in the input diff image expand (i.e. surrounding areas change from zero to non-zero), we'd expect the force to have changed. Notably, this phenomenon is different from models trained for typical computer vision tasks such as image classification, where the saliency map usually overlaps with the object of interest. This observation may potentially be useful for designing unique model architecture specifically for force estimation in tactile sensors. We discuss this further in the Future Work section (Section 6).

5.6. Ablation Study

Ablation study of input features For this set of ablation study, we use our best ResNet-18 model without transfer learning as the reference model, and weaken the model input in the 3 ways described in Section 4.2.4. Results in Table 4 show that the eval performance degrades in all 3

Model	MAE Mag.		MAE Dir.	
	Train	Eval	Train	Eval
Reference RN-18	**0.052**	**0.055**	7.88	**7.91**
Lower resolution	0.059	0.063	8.32	9.23
Distorted aspect ratio	0.053	0.060	**7.60**	9.25
No skeleton image	0.049	0.060	8.02	9.32

Table 4. **Ablation study of input features.** Eval performance degrades, and train-eval performance gap widens considerably with weakened input signals

Model	# Trainable Parameters	MAE Mag.		MAE Dir.	
		Train	Eval	Train	Eval
ViT-1	7.8M	0.187	0.182	19.7	19.1
ViT-2	14.9M	0.132	0.133	19.2	17.1
ViT-4	29.1M	0.127	0.127	**16.8**	**15.4**
ViT-8	57.4M	**0.124**	**0.120**	17.1	15.5

Table 5. **Ablation study of ViT layers.** Train versus Eval gap is minimal suggesting further training is necessary. Larger models achieved lower MAEs given same amount of epochs.

scenarios, which highlights the importance of high quality sensor images. Interestingly, the train-eval performance gap widens considerably across all 3 scenarios. The training error even decreased in some cases. This can be explained by the fact that weakened input signals made it harder for the model to generalize well, while it is still easy to overfit due to overparameterization. This suggests that regularization might become more crucial with lower quality sensor images.

Ablation study of ViT layers For this experiment, we aimed to understand the impact of varying the number of ViT layers in the attention based model. Given the limited time and limited resources for our project, we decided to train all four models using the same training strategy: learning rate of 1e-5 and 64 epochs. Results are summarized in Table 5. Firstly, we note that the difference between training and evaluation errors are very small, the evaluation error even falling below the training error in some cases. We attribute this behavior to the lack of training time and would expect the training error to fall below the evaluation error given more epochs. Secondly, as model size increased, we observed that, given the same training time, the larger models were generally able to achieve better performance, though the performance gains were incrementally smaller. Given these results, it appears that the larger ViT models would provide performance gains but once passed a certain threshold of learnable parameters, the gains would be marginal.

6. Conclusion & Future Work

Sun et al stated that they were able to achieve "a force magnitude accuracy of around 0.03N and a force direction accuracy of five degrees" [5]. With our implementation of their model, we were able to achieve 0.054N magnitude accuracy and 9.5° force direction accuracy. We attribute the difference between the original authors' accuracy and ours to three main sources:

- **Lossy image compression**: The publicly shared dataset contains JPG images while their code suggests they used PNG type images. The image quality can be highly influential as highlighted by our ablation study in Section 5.6.
- **Train/validate/test split**: The authors used a custom train/validate/test split but did not provide the relevant details
- **Other details in training**: The authors did not provide full details or complete working code. Our implementation is our best attempt at replicating their results, which is an imperfect replication.

Through searching for appropriate learning rate, experimenting with pre-trained weights, and removing input features found to be insubstantial in our ablation study, we arrived at our overall best performing model which achieved 0.048N magnitude accuracy and 7.3° directional accuracy. We also experimented with larger models (ResNet-50) and attention-based models (ViT) as backbones, but did not find any configurations that outperform the ResNet-18 backbone. Transfer learning helps improve model performance across all different models and configurations we tried, even though the pre-trained weights were trained using completely different domains for different computer vision tasks.

One potential area for future work is to experiment original and innovative model architectures or feature extractors designed specifically for force estimation. As demonstrated in Section 5.5, our problem domain is distinct from the typical computer vision problems that popular models are developed for. As visualized in Figure 4, the learnt convolutional filters of our model differ significantly from the pre-trained ResNet-18, and the saliency map in Figure 5 shows that our model behaves differently from a typical image classification model. Therefore, it may be worthwhile to design unique model architecture or feature extractors to address these discrepancies. One concrete idea is to replace the final pooling layer with 1x1 conv layer. Unlike image classification, where the location of the object has little to no significance, our problem is not translation invariant. The location of the contact point does affect the interpretation of a given patch of input signal. For example, the same image patch at a different location can mean

a different force direction. The final pooling layer reduces the spatial dimensions to 6x6 which can cause the model to lose accurate location information. Compared to pooling layers, a 1x1 convolution can achieve the same purpose of reducing output size while keeping the spatial information. Therefore, we believe this may potentially be helpful for the model.

As discussed in Section 5.3, our exploration using ViT models was limited given project timeline and resources. More in depth hyperparameter search and simply more training time should easily yield better performance on the ViT models we reported above. Use of the ViT model also required resizing the image to 224 x 224 pixels. Exploring different interpolation techniques for the resizing or even other preprocessing steps could help retain more important information in the image before passing it to the model.

7. Appendices

7.1. Appendix: Full visualization of all filter in the first layer of selected models

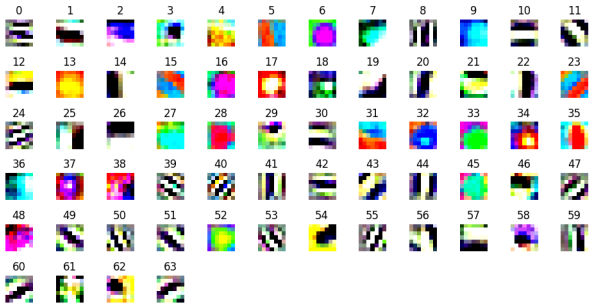

Figure 6. **Visualization of all 64 filters in the first layer of pre-trained ResNet-18**

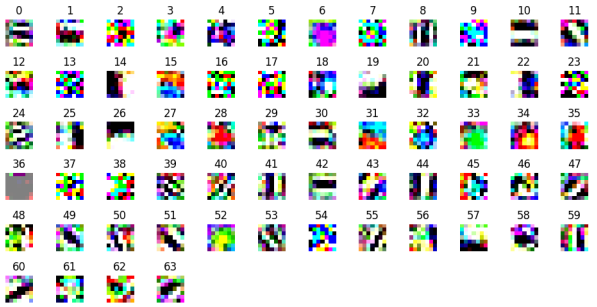

Figure 7. **Visualization of all 64 filters in the first layer of our finetuned ResNet-18 model**

Figure 8. **Visualization of all 64 filters in the first layer of our ResNet-18 model trained from scratch**

8. Contributions & Acknowledgements

8.1. Contributions

The individual contribution from each project team member is itemized as below.

Jiaxiang Fu

- Brainstorming and discussing project ideas
- Setting up the initial project starter code for baseline replication and metrics calculation

- Experiments on ResNet-18 and ResNet-50, hyperparameter tuning and transfer learning
- Ablation study on input features
- Model visualization: conv filters and saliency maps

Jeremie Gabor

- Brainstorming and discussing project ideas
- General tooling for model training and visualization
- Experiments with custom attention based models
- ViT model implementation and experiments
- Ablation study on ViT layers

8.2. Acknowledgements

The authors thank Julia Di who is the project mentor for their helpful feedback and guidance on this project. Julia provided the initial project ideas, project background, and timely feedback on project directions.

The authors would also like to thank Hao Li who is a CS231n TA for their advisory on this project. Hao provided insights on ResNet vs ViT based on their experience in robotics.

8.3. External Resources

- Made use of code and data found here which is provided by [5].
- Made use of code from this GitHub repo for network visualization.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, October 2021. 2

[2] Lu Chi, Zehuan Yuan, Yadong Mu, and Changhu Wang. Non-local neural networks with grouped bilinear attentional transforms. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11804–11813, 2020. 4

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3, 4

[4] M. Lambeta et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. 1, 2

[5] Sun H. Kuchenbecker K.J. Martius G. A soft thumb-sized vision-based sensor with accurate all-round force perception. *Nat Mach Intell*, 4:135–145, 2022. 1, 2, 3, 4, 7, 9

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 2, 3

[7] Vijay Kakani, Xuenan Cui, Mingjie Ma, and Hakil Kim. Vision-based tactile sensor mechanism for the estimation of contact position and force distribution using deep learning. *Sensors*, 21(5):1920, 2021. 1

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 3

[9] Benjamin Ward-Cherrier Nicholas Pestell Luke Cramphorn Benjamin Winstone Maria Elena Giannaccini Jonathan Rossiter and Nathan F. Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft Robotics*, 5(2):216–217, 2018. 1

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[11] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. 2022. 2

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[13] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. 1