Traction Force Microscopy using a Hybrid Unet-Axial-Attention model

Chirag Sreenivasamurthy Panchakshari

12/09/2024

# Abstract:

Traction Force Microscopy (TFM) is a powerful technique for measuring cellular forces exerted on substrates [1]. This study presents a novel approach to TFM using a single deformable hydrogel microparticle as a force sensor. We propose a method to predict the magnitude and direction of forces applied to the particle based on before and after images of its deformation.

Our approach utilizes the U-Net architecture from the 2015 seminal paper “Imagenet large scale visual recognition challenge” [2] with Axial Attention [3], a modification of the other seminal 2017 paper, “Attention is all you need” [4]. mechanisms integrated into the skip connections. The U-Net encoder-decoder structure allows for multi-scale feature extraction, while the axial attention modules enhance the network's ability to capture long-range dependencies and refine localization information. By applying axial attention to the skip connection features before appending them to the decoder, we aim to improve the network's capacity to learn the complex relationship between particle deformation and applied forces.

We simulate deformations of a single microparticle under various force conditions to generate a dataset of before and after images. The network is trained to predict force magnitude and direction directly from these image pairs. This method offers potential advantages over traditional TFM techniques, including simplified experimental setup, reduced computational complexity, and the ability to measure forces in three dimensions.

Our study demonstrates the feasibility of using deep learning approaches for force prediction in TFM and opens new avenues for investigating cellular mechanics at high spatial and temporal resolution. This technique could find applications in studying various biological processes, including cell migration and tissue morphogenesis.

# Introduction:

Conventional TFM involves data acquisition of the “stress” and “null” bead images. The next step is calculating the displacement, which can be done using one of two methods, Particle Tracking Velocimetry [5] or Particle Image Velocimetry [6]. From here it is the task of performing Finite Element Analysis and solving the inverse problem to infer the traction forces based on the displacements. The disadvantage of this method is the computational expense of solving the finite element analysis and inverse problem.

With the advent and rise of the Artificial Intelligence, more and more scientific fields are finding ways to incorporate this marvelous black box and advance science and humanity. The Attention mechanism has been cited over 140,000 times to this date and its cousins are still revolutionizing the limits of computation. Its success is mainly attributed to its ability to gather long-range semantic relations in text. Axial Attention is one such variant that takes extends the one-dimensional correlations to two dimensional vectors, such as images.

The term “convolution” first appeared in neural networks in a 1987 paper by Toshiteru et al [7]. Since then, Convolutional Neural Networks (CNN) have been inconsequential to the advancement of image analysis. During training, kernels learn and extract local features but are semantically weak. They cannot capture long range relations beyond the kernel size. To combat this the famous Unet architecture was created, and it revolutionized the industry. By having “skip-connections”, high-level features that would normally be lost in deep neural networks are appended during the decoder phase of the architecture. Here is where the idea of combining both CNNs and Attention was first incepted by us. Using the Unet, high-level features could be preserved, and Attention could be used to infer long-range relations within the feature-maps.

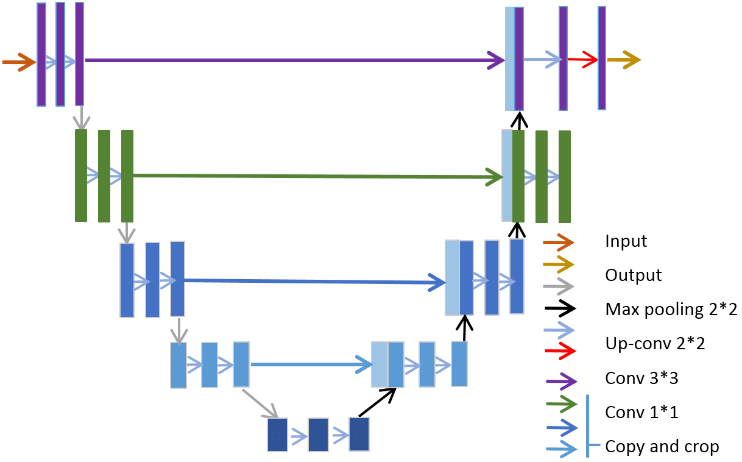
# Materials and Methods:

## Simulated Images:

Dr. Sangyoon Han was gracious enough to provide the MATLAB code to generate the simulated images of size 256x256. Force and size of the particle were varied with ranges as (START, STOP, STEP) of (200, 2000, 200) and (2, 20, 2) respectively. With a simple Python script to invoke the relevant MATLAB functions, we were able to gather the null and stress bead images along with ground truth forces in the X and Y directions.

## Neural Network Architecture:

Let’s start with the Unet. It consists of an encoder and decoder phase forming the iconic “U shape”. Each “step” of the encoder consists of two convolution layers with ReLU activation functions with a Batch Normalization layer in-between followed by a Max Pool layer. The feature maps at each stage are stored. The decoder phase starts with a Up-convolution layer, which performs Transposed Convolution. The stored feature maps act as the skip-connections and are appended to the up-convolved vector before performing the double convolution procedure in the encoder stages.



*Ding, Yi & Chen, Fujuan & Zhao, Yang & Wu, Zhixing & Zhang, Chao & Wu, Dongyuan. (2019). A Stacked Multi-Connection Simple Reducing Net for Brain Tumor Segmentation. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2926448.*

A variation of the Axial Attention mechanism we used is shown below. In our project, we perform the Height-Axis and Width-Axis Attention on the feature map and add both results before appending to the decoder feature map.

A diagram of a graph

Description automatically generated

For our project, we used a channel structure of (64, 64, 128) for the Unet encoder-decoder and 4 heads for the MultiHeadAttention layer. A batch of 8 with random sampling of the first 500 images.

## Loss Function:

One of the most important factors in my opinion is the custom loss function developed for this project. While MSE is a standard loss metric, it is limiting in its primitive nature. To develop a more nuanced loss function better able to steer the gradient descent we incorporated the Deviation of Traction Magnitude (DTM) metric in our calculations.

From here we take a slightly more ***relaxed assumption*** of proportionality and consider only the numerator. We also square the result to avoid negative values and add RMSE to obtain our final loss metric given below.

The final iteration of the loss function that was used was masking all the zero values and only computing the loss for non-zero values.

# Results and Discussion:

## Basic Unet:

The project had a rocky start with the Unet consuming all my machine’s GPU resources and crashing with a channel structure of (64, 128, 256, 512) and batch size of 32. Further iterations showed that the upper limit was a channel structure of (64, 64, 128) with a batch size of 32. Initial results are posted below.

A graph of a train loss

Description automatically generated

As you can see, the initial loss is extremely high and plateaus at ~2000 and this can be seen in the quiver plot on testing on a random image.

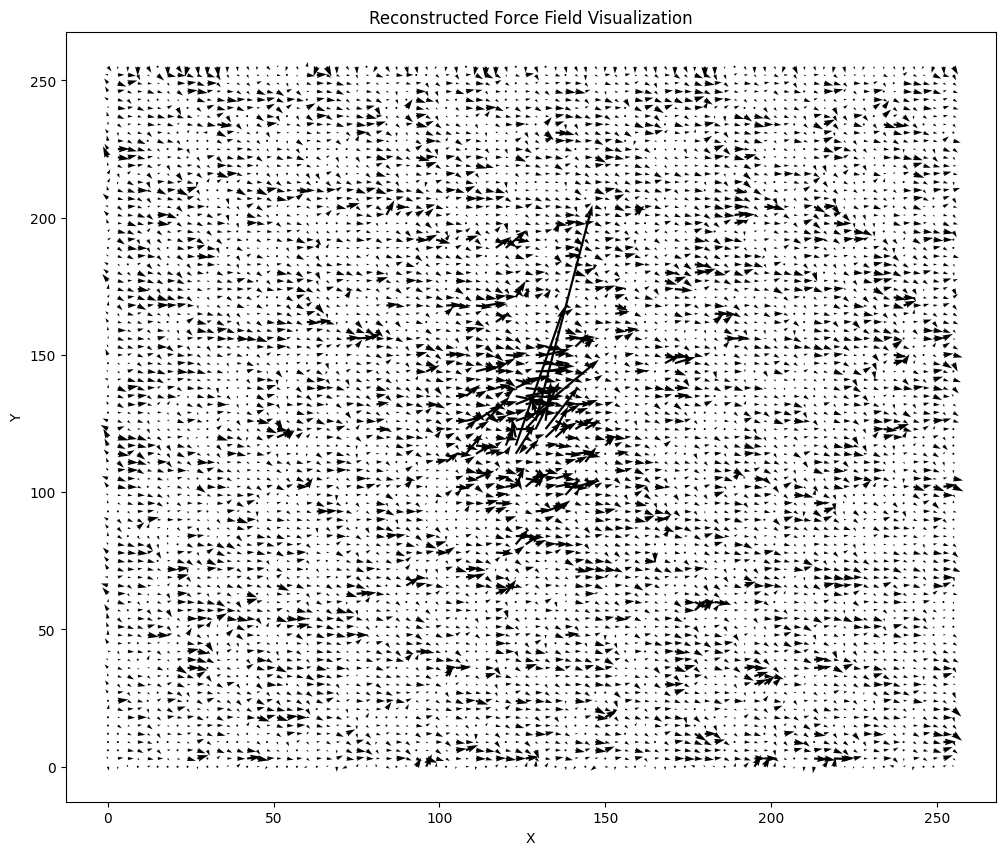
A grid of random dots

Description automatically generated with medium confidence

A graph of a graph showing the number of points in the same graph

Description automatically generated with medium confidence

Further modifications to the epoch number, loss function and batch size also didn’t yield better results, with the best results shown below.



A graph of a graph showing the number of points in the same graph

Description automatically generated with medium confidence

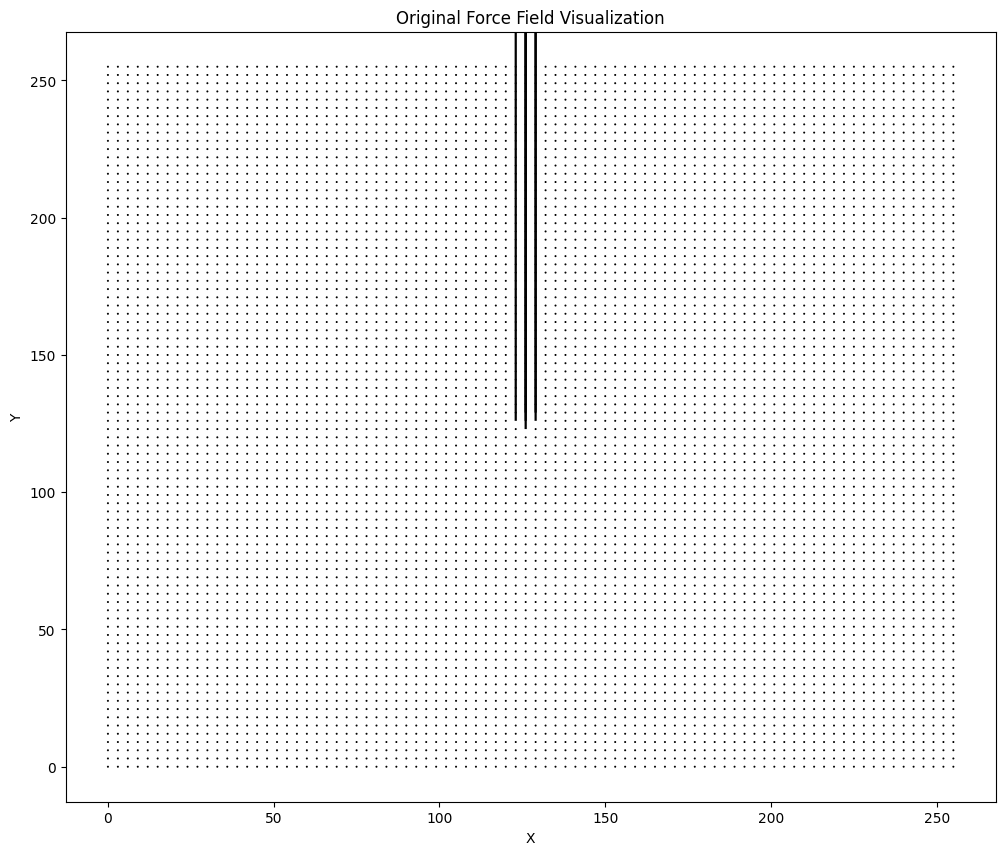
The issue of needing to capture long-range relations were inevitable and turned to Attention to see if it would fare any better.

## Unet + Axial Attention:

I want to deviate a little here and share with you my “Eureka moment”. Granted with GPU access and a potentially promising approach, I was filled with new vigor. This was short lived as the model would throw “fits” because of dimension mismatch when appending the skip-connections after Axial Attention. Furthermore, the model was consuming the entire 32GB of GPU storage with an initial structure of (64, 64, 128), batch size of 32, 8 heads and the first 500 images. After some reduction to a batch size of 8 and 4 heads in the MultiHeadAttention layer, the model would work. With the initial model architecture, I trained the model on 10 images and for 3 epochs and this was the result (I cried a little seeing this).

A diagram of a graph showing the force field visualization

Description automatically generated



Yes, it is a beautiful sight when you haven’t slept for two days worrying about not being able to show anything and a looming project deadline. Anyway, back to the original discussion.

The first run (shown in green) was for 100 epochs, and the model reached a min of 408, a significant reduction from the previous approach. The second run (shown in red) was for 200 epochs, and the model reached a low of 18 and plateaued at the 100th epoch.

A graph showing the growth of a train loss

Description automatically generated

The Y-axis is shown in the log scale to get a better sense of the minor differences with the large range of initial and final loss values. A random image is predicted on this final model.

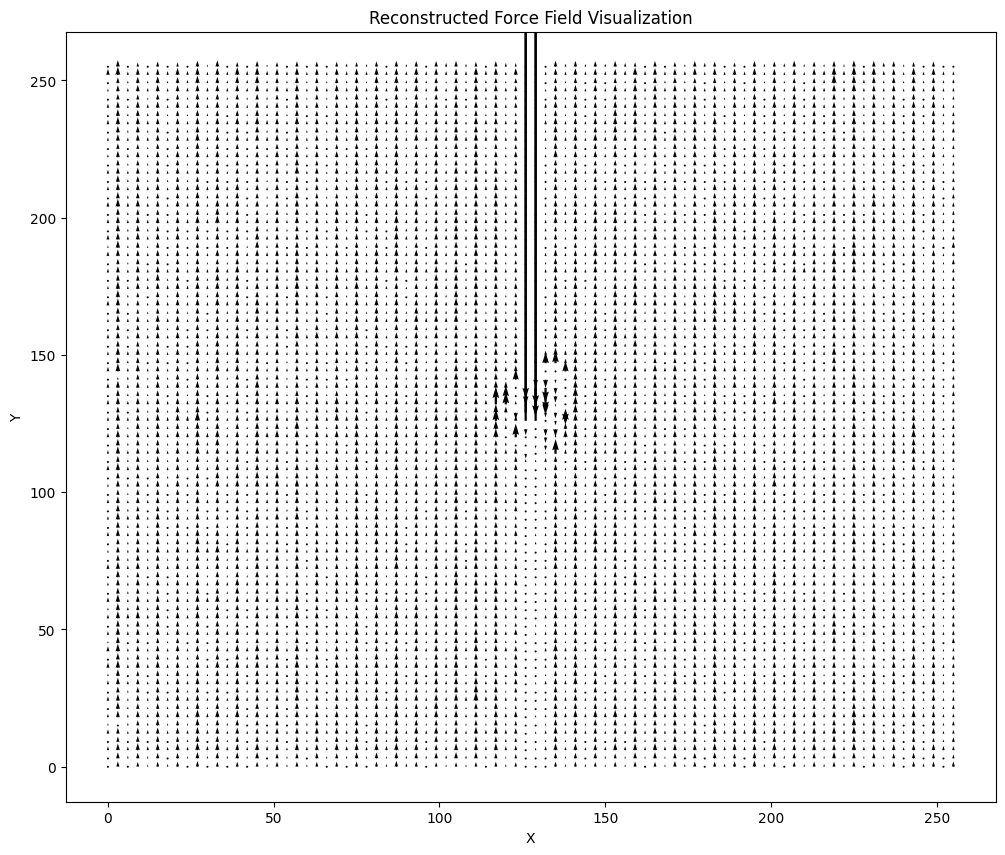
A graph of a graph showing a line graph

Description automatically generated

A graph of a force field

Description automatically generated

Based on visual inference, it is clearly seen that the Attention mechanism can reduce the phantom forces predicted by the Unet alone. Using a higher resolution for the quiver plot and we see that although it performed brilliantly, it isn’t perfect, yet.



A graph of a graph showing the same number of points

Description automatically generated with medium confidence

This is the prediction on a force with small distances. This can be accredited to the fact that the model was not trained on the whole dataset with random sampling but rather on the first 500 images.

## Final Comments:

Although the model outperformed the basic Unet model, it still requires more refinement. Possible improvements include tuning the batch size, increasing the number of heads in the MultiHeadAttention layer, increasing the channel structure of the Unet, using a more robust and larger dataset, modification of the loss function and finally more GPU resources.

# References:

1. Style, R. W., Boltyanskiy, R., German, G. K., Hyland, C., MacMinn, C. W., Mertz, A. F., ... & Dufresne, E. R. (2014). Traction force microscopy in physics and biology. *Soft matter*, *10*(23), 4047-4055.
2. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115, 211-252.
3. Ho, J., Kalchbrenner, N., Weissenborn, D., & Salimans, T. (2019). Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*.
4. Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.
5. Robben, F., Cheng, R. K., Popovich, M. M., & Weinberg, F. J. (1980). Associating particle tracking with laser fringe anemometry. *Journal of Physics E: Scientific Instruments*, *13*(3), 315.
6. Grant, I. (1997). Particle image velocimetry: a review. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *211*(1), 55-76.
7. Atlas, L., Homma, T., & Marks, R. (1987). An artificial neural network for spatio-temporal bipolar patterns: Application to phoneme classification. In *Neural information processing systems*.