

Characterization of turbulent flows in tokamaks

Julien Hu, Matthieu Masouyé and Sébastien Ollquist
Department of Computer Science, EPFL, Switzerland

Abstract—A tokamak is an object in the form of a torus in which we make plasma turn at very high speeds in order to generate energy. The goal is then to push gas into the tokamak while plasma is turning and collect images of it in order to detect the regions where turbulence is the highest. This whole process is called Gas Puff Imaging (GPI) and the concerned specific regions of turbulence are called blobs. We will here describe what we have implemented in order to help identifying the direction of the velocity of the particles in a selected turbulent region.

I. INTRODUCTION

Gas Puff Imaging (GPI) is a technique that involves pushing gas in a tokamak in order to study the turbulence present at the edge of magnetically confined plasmas. It uses a puff of neutral gas so we can increase the local light emission level in order to improve optical imaging of the space-time structure of the edge plasma turbulence.

The primary goal of this project is to understand and implement machine learning methods that can help us detect the shear layer, that is, the separation at which the direction of the flow of particles changes. The idea is to implement a Convolutional Neural Network on the data set in order to easily distinguish the different flows [1].

II. DATA IMPORTATION AND MANIPULATION

We were first given a matlab file with all necessary data to use for the project. For the purpose, we have created a python script that reads the data from the .mat file and creates a numpy array from each part of the given file. Then, these arrays are written into four separate csv files, one for each category. There are four relevant data sets:

- 1) *r_arr* a 10 by 12 matrix of radiuses.
- 2) *z_arr* the corresponding array of the z-axis values.
- 3) *t_window* the array of times at which a particular value was measured (the frequency is $2MHz$ so there is one experiment every $500ns$ approximately.)
- 4) *brt_arr* the 3d-array ($10 \times 12 \times 200000$) of each individual experiment.

III. SYNTHETIC DATA GENERATION

IV. FINDING THE CORRECT ARCHITECTURE

This part essentially contains the research we have done in order to choose the correct architecture for the project. The model we have decided to use is based on ResNet, which is a classic neural network that helps solving computer vision tasks [2]. As we have to analyze a sequence of images, this model is the perfect solution for us.

V. TRAINING THE MODEL ON SYNTHETIC DATA

VI. TRAINING THE MODEL ON THE REAL DATA

VII. CONCLUSION

REFERENCES

- [1] M. J. Park and M. Sacchi, "Automatic velocity analysis using convolutional neural network and transfer learning," 2019. [Online]. Available: https://www.researchgate.net/publication/336339562_Automatic_velocity_analysis_using_convolutional_neural_network_and_transfer_learning
- [2] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" *arXiv preprint*, vol. arXiv:1711.09577, 2017.