

IMPERIAL

IMPERIAL COLLEGE LONDON

DEPARTMENT OF PHYSICS

Machine Learning Guided Diagnostics for Laser-Driven Ion Beams

CID:

02112158

Supervisor:

Dr. Nicholas Dover

Project Partners' CIDs:

02021222

01857821

Assessor:

Dr. Oliver Ettlinger

BSc Project

Word Count: 5819

May 6, 2024

Abstract

Computational methods were used to develop a Monte Carlo simulation of laser-driven ion beams. The beams were simulated to pass through a 3x3 grid of aluminium filters of various thicknesses before depositing their remaining energies in a scintillator. A simulated 12-bit charge-coupled device (CCD) captured the resulting spatial energy profile as an image. Three key parameters were used in the generation of the beams: The maximum energy of a proton in the beam, E_{\max} , the temperature of the proton spectra, T_p , and the number of protons in the beam, N_0 . Synthetic datasets consisting of 100,000 labelled images were generated to provide sufficient training data. This data enabled a neural network (NN) to predict the three beam parameters from an input image. A Bayesian optimisation algorithm was used to efficiently explore different configurations of filter thicknesses and the architecture of the NN to provide the most accurate predictions of the beam parameters. 116 Bayesian optimisation trials were carried out to refine the filter thicknesses and 360 trials were carried out to optimise the architecture of the NN. The optimisation reduced the NN's prediction error rate (mean-squared error loss) from 0.0050 in an unoptimised configuration to 0.0027, significantly improving accuracy. This performance translated to a mean relative absolute error in predicting the three beam parameters E_{\max} , T_p , N_0 of 7%, 13% and 4% respectively. The optimised NN's prediction time on a specific hardware configuration equated 7.3 ms with a standard deviation of 0.8 ms. This performance highlights the strong potential of NNs to provide rapid yet accurate feedback in laser-driven ion beam systems operating at frequencies greater than 100 Hz.

Declaration of Work Undertaken

In this project, which commenced at the start of year 3 - term 2, Aran enhanced the project's computational infrastructure by rewriting the synthetic data generation code for increased efficiency and modularity. He also introduced parallel computing for faster data processing and implemented Bayesian optimization with Optuna to fine-tune the aluminium filter thicknesses and the NN architecture. He orchestrated this process on both the Imperial High-Performance Computing (HPC) resources and on Amazon Web Services (AWS) to circumvent memory limitations.

Adam contributed his prior machine learning experience to the development of the convolutional NN in Keras, integrating lookup tables to accelerate the synthetic data generation and derived the transformation method to sample from the proton energy probability density function. He also created a streamlined version of the NN, taking fewer inputs, in an attempt to speed up inference time while maintaining accuracy, although this was not used when taking data for final results.

Hans focussed on the accurate representation of the physical phenomena by working on the initial computational implementation of the synthetic data generation. He did this by fitting splines to the NIST dataset relating proton energy and metal thickness to energy deposited and created an initial functional version of the data generation workflow. Towards the end of the project, he worked on implementing the contribution of electrons to the laser-driven beam. However, this was ultimately unused due to time and practicality constraints.

Contents

1	Introduction	3
2	Theory	4
2.1	Mechanism of Proton Beam Generation and Energy Distribution in Laser-Driven Ion Accelerators	4
2.2	Proton Beam Diagnostics Using a Multi-Filter Configuration and CCD Image Capture . .	4
2.3	Neural Networks and Bayesian Optimisation	5
3	Method	7
3.1	High-Level Overview	7
3.2	Synthetic Data Generation	7
3.2.1	Motivations for Synthetic Data Generation	7
3.2.2	Image Generation Process	8
3.3	Machine Learning	9
3.4	Bayesian Optimisation	9
3.4.1	Optimising Filter Thicknesses	9
3.4.2	Optimising Neural Network Architecture	10
4	Results	11
4.1	Synthetic Data Generation	11
4.2	Neural Network and Bayesian Optimisation Performance	13
4.3	Optimal Configuration Metrics	14
4.4	Discussion of Results	15
5	Improvements and Future Work	16
5.1	More Realistic Data Generation With Geant4	16
5.2	Ground Truth Comparison	16
6	Conclusion	17
	Bibliography	20

Chapter 1

Introduction

Recent advances in the ability to generate high-power lasers of magnitudes greater than the PW scale ($>10^{15}$ W) have given rise to new types of particle accelerators referred to as laser-driven ion accelerators [1]. These typically operate through a process called target normal sheath acceleration (TNSA). This involves, firing high-power lasers at a metal target which induces an electric field across the metal. Protons are accelerated by this field and released from the surface of the metal. The accelerated protons travel in a group known as a laser-driven ion beam [2]. A substantial advantage of the laser-driven ion accelerator in comparison to conventional types of accelerators, such as cyclotrons and synchrotrons, is their ability to be made much more compact and cheaper [3]. These advantages make such accelerators more accessible to academic institutions that might have been deterred by the high costs and logistical challenges of conventional accelerators. Laser-driven ion accelerators also exhibit significant potential for use in proton beam therapy, an alternative treatment option to radiotherapy [4]. Proton beam therapy offers notable advantages compared to radiotherapy such as achieving the same efficacy in treating cancer while damaging less of the surrounding tissue [5]. The current disadvantage is that proton beam therapy is typically more costly with a larger and more complex infrastructure as a result of using conventional particle accelerators (such as cyclotrons and synchrotrons) [6]. Laser-driven ion accelerators have the potential to address both of these limitations making this treatment option more accessible to the wider population.

Although laser-driven ion accelerators hold significant potential, there are still challenges facing their widespread adoption. These include damage to the target metal necessitating frequent target replacement, difficulty achieving continuous proton beams due to the discrete laser pulses used in acceleration, and ensuring predictable and consistent beam parameters [7]. Conventional types of particle accelerators utilise electric and magnetic fields to produce beams of predictable parameters. In contrast, the theoretical complexity of laser-matter interactions makes it challenging to accurately predict the characteristics of the resulting laser-driven ion beam given specified input parameters [8]. Rapid diagnostics coupled with computational techniques aim to address this challenge. To achieve predictable proton beam parameters, a methodical approach can be employed. Initially, a laser-driven proton beam is generated using specific, known laser input parameters. Subsequently, diagnostic tools measure the proton beam's characteristics, and computational algorithms analyse these data to determine the beam parameters. This feedback informs adjustments to the laser settings. An optimisation algorithm iteratively refines these settings through multiple trials, progressively improving the alignment of the proton beam parameters with the desired outcomes.

Achieving rapid beam diagnostics to facilitate the on-the-fly honing of beam parameters poses a significant challenge. Traditional methods for analysing proton beams such as film media analysis are typically labour-intensive and time-consuming [9]. This motivates the use of automated techniques to accelerate the diagnostics of proton beams. This report details an approach involving machine learning models to rapidly predict proton beam parameters [9]. Specifically, a neural network (NN) was trained on synthetic data to predict beam parameters given an input proton beam image. Bayesian optimisation was then employed to refine both the filter thicknesses used in the synthetic data generation and the NN's architecture to improve the NN's prediction accuracy.

Chapter 2

Theory

2.1 Mechanism of Proton Beam Generation and Energy Distribution in Laser-Driven Ion Accelerators

Laser-driven ion accelerators involve firing high-power laser beams at a metal target. The incident laser accelerates electrons to the opposite side of the metal, forming an electron sheath. The electron deficiency on the incident side of the metal coupled with the electron sheath on the opposite side induces an electric field across the metal, parallel to the direction of propagation of the incident laser. This field then accelerates protons on the surface of the metal, resulting in a proton beam [2]. The resulting beam contains a distribution of protons with varying energies, described by the relationship,

$$\frac{dN}{dE} \propto e^{-\frac{E}{T_p}}, \quad (2.1)$$

where N is the number of protons at a certain energy, E is the energy of the proton, and T_p is the temperature of the proton spectra, dictating the pitch of the distribution [10]. Normalising equation 2.1, taking into account the cut-off associated with the maximum proton energy, E_{max} , yields the probability density function (PDF),

$$p(E; E_{max}, T_p) = \frac{e^{-\frac{E}{T_p}}}{T_p(1 - e^{-\frac{E_{max}}{T_p}})}. \quad (2.2)$$

The third key proton beam parameter, which does not feature in equation 2.2, is the number of protons contained in the beam, N_0 .

2.2 Proton Beam Diagnostics Using a Multi-Filter Configuration and CCD Image Capture

In the diagnostics setup, proton beams are directed to pass through a 3x3 grid of aluminium filters, known as PROBIES [9]. Each filter in the grid varies in thickness. This configuration is designed to capture more comprehensive information about the proton beam's properties. Unlike a single filter, which might either completely block or allow all protons from a beam to pass, the varied thicknesses in the grid create a range of interactions. This variety in data allows the machine learning model to gain a deeper understanding of the beam's characteristics, such as its intensity and distribution, enhancing the accuracy of the diagnostics. Each proton deposits a certain amount of its energy in an aluminium filter as a function of its initial energy and the thickness of the filter. Protons of low energy may deposit all of their energy as they pass through the filter and so do not pass out the opposite side. Protons in the beam with sufficient remaining energy to pass out to the other side will then pass through a scintillator, converting a portion of their remaining energy to photons. Some protons with high energy will pass out the other side without depositing all their energy in the scintillator. The protons produced are collected by a 12-bit CCD situated behind the

scintillator. The CCD captures information on the proton beam's spatial energy profile, which a machine-learning model can then process to determine the parameters of the beam. The image generation process via proton beams passing through a 3x3 filter setup is depicted in figure 2.1 and a snippet of a real image captured with a 2x2 filter setup is displayed in figure 2.2.

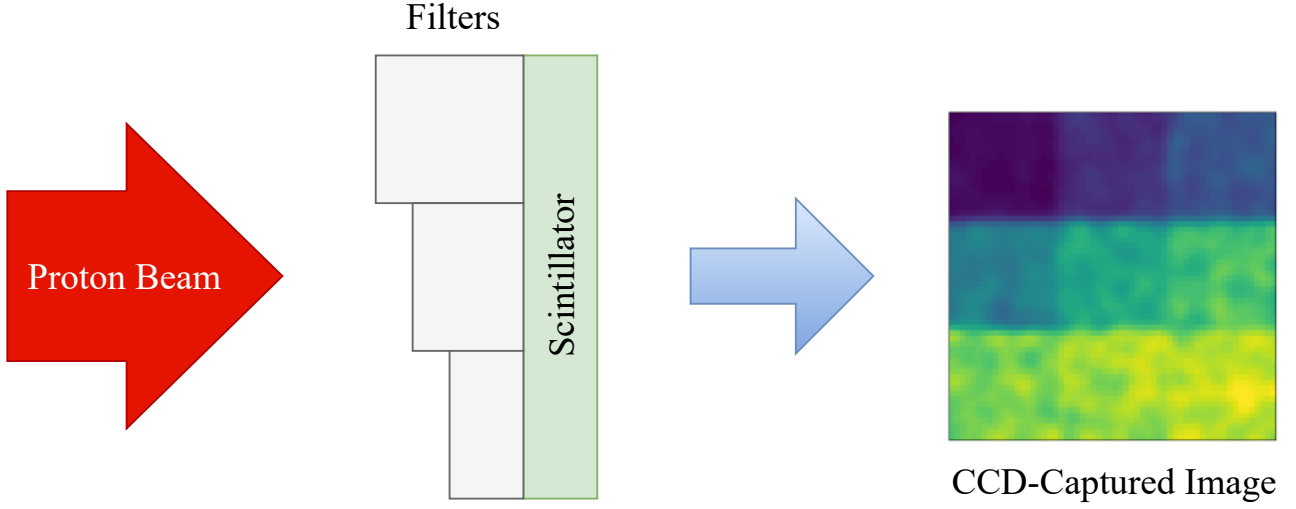


Figure 2.1: Overview of the physical process involving a proton beam passing through a 3x3 grid of aluminium filters, scintillator, and finally being captured by a CCD to produce an image. A side view of the 3x3 filter setup is featured.

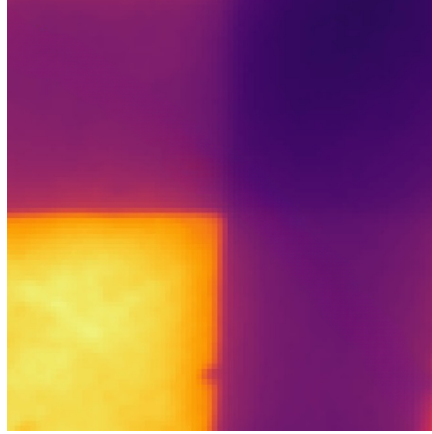


Figure 2.2: A real experimental image generated by a proton beam passing through a 2x2 filter setup (Image courtesy of Dr. Nicholas Dover, 2024).

2.3 Neural Networks and Bayesian Optimisation

Neural networks, a type of machine learning model, are inspired by biological neural networks within animal brains. They consist of layers of nodes or "neurons" that are connected to other layers via weights and biases. A consecutive set of matrix operations controls the flow of data from input to output [11]. In this project, NNs were trained on synthetic CCD-captured beam images labelled with E_{max} , T_p , and N_0 . The weights and biases of the NN were optimized using a method called gradient descent, which is executed across several epochs. An epoch refers to one complete pass of the entire dataset through the training process. This process aimed to minimise the error of the NN which represents the difference between the NN's predicted parameters and the true parameters (labels) of the image. Ensuring the accuracy and generalisability of an NN hinges on supplying a diverse set of high-quality, labelled images, alongside selecting the ideal hyperparameters, which include the NN architecture and training settings.

Bayesian optimisation is a powerful strategy for improving the performance of NN models, particularly in situations where evaluating the performance of the NN is computationally expensive. Bayesian optimisation uses Bayesian probability theory to efficiently search for optimal NN hyperparameters. The algorithm constructs two models that it uses to search the hyperparameters space. A surrogate model is used to approximate the objective function that maps the NN hyperparameters to its resulting performance. An acquisition function, that relies on the surrogate model, is then used to determine the next set of hyperparameters to evaluate the objective function, balancing exploration and exploitation. Both the surrogate model and acquisition function are updated based on the results of all previous trials. In addition to optimising the hyperparameters of the NN, Bayesian optimisation was also used to optimise the filter thicknesses of the 3x3 grid to determine the configuration that allowed the NN to predict the beam parameters with the most accuracy [12].

Chapter 3

Method

3.1 High-Level Overview

The overarching goal of the project was to train an NN to accurately predict the three proton beam parameters E_{max} , T_p and N_0 from images captured by a CCD. The method taken to achieve this involved three main components: Synthetic data generation, machine learning, and Bayesian optimisation. The synthetic data generation involved modelling the distribution of energies in a proton beam according to equation 2.2 by uniformly sampling the three parameters from a range. Then the passing of protons through the 3x3 aluminium filters and the depositing of their energies in the scintillator was simulated. This resulted in a synthetic image labelled with the parameters E_{max} , T_p , and N_0 that were used for its generation. An NN was then trained off 75,000 labelled synthetic images with varying beam parameter values to learn to predict the true beam parameters based on an input image. Figure 3.1 demonstrates the process from generating a synthetic image based on true beam parameters to the NN attempting to predict these parameters. The first Bayesian optimisation step consisted of optimising the filter thicknesses of the 3x3 grid used in the synthetic image generation process. The optimisation was carried out to reduce the prediction error that the trained NN made. Bayesian optimisation was again used to determine the most optimal NN architecture.

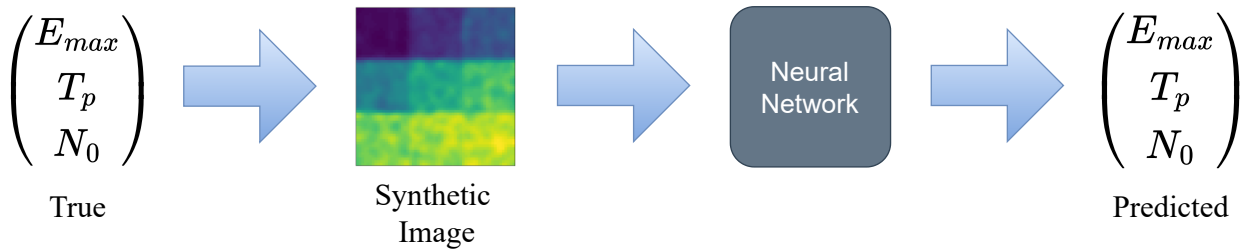


Figure 3.1: A flowchart depicting the workflow of generating a synthetic image and an NN predicting the beam parameters given the image as an input. Synthetic images are generated based on the input parameters, E_{max} , T_p , and N_0 . A hypothetical ‘perfect’ NN would output predicted parameters identical to the true parameters used in its generation for all input images.

3.2 Synthetic Data Generation

3.2.1 Motivations for Synthetic Data Generation

The implementation of synthetic data generation rather than using real labelled images for training the NN was primarily motivated by two reasons. The first is the insufficient quantity of real, high-quality, labelled images. Training an NN to make accurate predictions often requires a large dataset of images. The

time-consuming nature of obtaining a large dataset of high-quality labelled images makes this approach impractical, necessitating the use of computational simulations as an alternative. The second advantage of synthetic data generation is that it facilitates the rapid and systematic variation of physical attributes in the generation of images. This is particularly useful for optimising the aluminium filter configuration which would be impractically time-consuming and expensive to carry out through physical experiments.

3.2.2 Image Generation Process

The first step was to uniformly sample the beam parameters E_{max} , T_p , and N_0 between specified ranges. The parameter ranges were chosen based on the profiles of beams that could hypothetically be generated in a real experiment and were chosen to be $E_{max} \in [0.1 \text{ MeV}, 5.0 \text{ MeV}]$, $T_p \in [0.05 \text{ MeV}, 2.00 \text{ MeV}]$, and $N_0 \in [10^7, 10^{10}]$. The values for E_{max} and N_0 were then used to determine the PDF according to equation 2.2. A set of 10^5 proton energies was sampled from this distribution using the inverse transform method. The inverse cumulative distribution function used in the transformation method, $F^{-1}(x)$, is given by

$$F^{-1}(x; E_{max}, T_p) = -T_p \ln \left[1 - x \left\{ 1 - e^{-\frac{E_{max}}{T_p}} \right\} \right], \quad (3.1)$$

where $x \sim \mathcal{U}(0, 1)$. After the set of 10^5 proton energies had been generated, the maximum proton energy in the set was identified and E_{max} was reset to this value. The purpose of this approach was to ensure that the E_{max} label was reflective of the maximum proton in energy in the beam rather than the theoretical distribution that underlies it. In particular, proton beams generated with high E_{max} and low T_p often had the maximum sampled proton energy being significantly smaller than the PDF's E_{max} which would have resulted in the NN performing very poorly in these cases.

The set of 10^5 proton energies were each randomly allocated with equal probability to one of nine filters in the 3x3 grid. Data relating the initial energy of the proton and the thickness of the aluminium filter was given by the NIST database [13]. A 2D cubic spline was used to interpolate the NIST database. This resulted in a continuous function that approximated the total energy deposited by a single proton given its initial energy and the thickness of aluminium it passed through. The spline was created between filter thicknesses of $100 \mu\text{m}$ and $0.1 \mu\text{m}$. These bounds were selected based on heuristic considerations: The upper limit ensures that the thickest filter can effectively block the most energetic proton produced in the simulations. Conversely, the lower limit allows the least energetic protons to pass through the thinnest filter. Using the python module `cProfile` identified the execution of this cubic spline function as a significant performance bottleneck of the synthetic image generation, taking approximately 40% of the time. To address this, a lookup table was created from the spline that discretised the filter thicknesses to nine uniformly spaced filter thicknesses per decade between $100 \mu\text{m}$ and $0.1 \mu\text{m}$ resulting in 27 possible filter thicknesses. This resulted in a significant reduction in the time taken to compute the energy deposited in a filter.

Deducting the energy deposited in the filters for each proton energy gave a list of all the remaining proton energies before entering the scintillator. A 1D cubic spline was generated from data in the NIST database relating the initial proton energy to the energy deposited in a vinyl toluene scintillator [13]. A 1D spline was used instead of a 2D spline since a constant scintillator thickness of $2.5 \mu\text{m}$ was used for all simulations. Again, for performance reasons, the spline was converted to a lookup table that mapped the initial proton energy to its energy deposited in the scintillator. The lookup table computed the energy deposited in the scintillator from the remaining proton energies. This resulted in nine lists of proton energies deposited in the scintillator with each list corresponding to the filter that those protons passed through. These energies were then randomly allocated to pixels corresponding to the filters that they were associated with (all the filters were modelled as equally sized squares located in front of a CCD containing 60×60 pixels). If multiple protons fell on the same pixel their energies deposited would be summed. This gave the 2D synthetic image before accounting for N_0 and the simulation of experimental noise.

N_0 was accounted for by scaling the image by a factor of $\frac{N_0}{10^5}$. To account for noise that would be present in real images, a Gaussian blur was applied to each of the filters individually and then to the whole image. After generating a set of images with varying beam parameters, the maximum pixel value

from the complete set of images was identified. This was then used to linearly calibrate the pixel values to between 0 and 4095, corresponding to a 12-bit CCD that could be used in real experiments.

3.3 Machine Learning

The type of NN used to predict beam parameters was the convolutional neural network (CNN). CNNs excel at making predictions from image inputs as they contain convolutional and pooling layers which are tailored to identifying spatial patterns in data [14]. The mean-squared error loss, MSE , was minimised during the CNN training process and is given by

$$MSE = \frac{1}{N} \sum_{i=1}^N \left[(E_{max,i} - \hat{E}_{max,i})^2 + (T_{p,i} - \hat{T}_{p,i})^2 + (N_{0,i} - \hat{N}_{0,i})^2 \right], \quad (3.2)$$

where N is the number of images in the batch, $E_{max,i}$, $T_{p,i}$, and $N_{0,i}$, denote the true beam parameters of the i^{th} image while $\hat{E}_{max,i}$, $\hat{T}_{p,i}$, and $\hat{N}_{0,i}$ denote the CNN's predictions. The python module `Keras` (a high-level wrapper for `TensorFlow`) was used for the implementation of the CNN algorithms [15].

Before training the CNN, the images and labels were normalised to between 0 and 1 for uniformity, using min-max scaling. Normalising the labels was particularly important as the N_0 range was of a much greater order of magnitude compared to the E_{max} and T_p ranges. This meant that if the parameters were left in an unnormalised state, the CNN optimisation process would optimise the parameters with a greater emphasis on minimising the MSE of N_0 , neglecting the other parameters. Before applying min-max scaling to N_0 , its values were transformed as $\log_{10}(N_0)$, to ensure that the normalised N_0 values were more evenly spread between 0 and 1, rather than being concentrated at the high-end.

The performance of the CNN was tested on images and labels that were not used in the training process (validation dataset). Using a validation dataset meant that the ability of the CNN to generalise to new data was tested making it possible to identify if the CNN was learning patterns exclusively present in the training dataset (overfitting). 100,000 synthetic images were provided for each CNN training process, with 75,000 allocated as training data and 25,000 as validation data. To prevent the CNN overfitting, an early stopping callback was introduced so that the MSE on a validation dataset was monitored after each epoch. If the MSE did not improve for 5 consecutive epochs, the training process would stop and the parameters of the CNN would be restored to their most optimal configuration.

3.4 Bayesian Optimisation

3.4.1 Optimising Filter Thicknesses

The Bayesian optimisation algorithms were implemented through the high-level Python package, `Optuna` [16]. `Optuna` was initially employed to optimise the filter thicknesses used in the synthetic data generation while the CNN was held constant. The n^{th} filter's thickness, t_n , was sampled from the distribution

$$t_n = ab^{n-1}, \quad (3.3)$$

where n is the filter number and a , b are parameters to be optimised with $a \in [0.1 \mu\text{m}, 90 \mu\text{m}]$ and $b \in [0, 2.4]$. The a range was chosen to span the whole range of the possible filters while the b range was chosen such that its maximum value would result in a distribution that spanned to the maximum filter thickness given the minimum value of a . The exponential form with two parameters was used to ensure that filter thicknesses of a wide range of magnitudes were tested to help differentiate between protons of varying energies. The t_n values were each mapped to their nearest of the 27 discrete filter options before generating the synthetic images. The optimisation of only two parameters defining the set of filter thicknesses was motivated by the infeasibility of optimising each filter thickness separately due to the time-consuming nature of the synthetic data generation. To further address this, only 5,000 images were generated (with 1250 allocated as validation data) during each Bayesian optimisation iteration rather than

the typical 100,000 used when training the final CNN. This decision was made assuming that the best filter thickness distribution for the CNN to learn off 3,750 images was the same as for 75,000 images. Once optimised filter thicknesses were determined for 5,000 images, a 100,000 dataset of images was generated with these thicknesses and used to determine the CNN performance on this larger quantity of training data.

3.4.2 Optimising Neural Network Architecture

The architecture of the CNN was also optimised on a 100,000 dataset of synthetic images generated with the optimised filter thicknesses. The CNN hyperparameters optimised in this process are displayed in table 3.1. The CNN optimisation focused on architectural hyperparameters. This targeted approach prevented the search space from becoming excessively large, which would dilute the optimisation efforts and significantly increase the computational load and time. Architectural hyperparameters were prioritised because, unlike other hyperparameters such as those specific to training, they have a direct impact on the model’s prediction time.

CNN Hyperparameter	Search Space	CNN Hyperparameter	Search Space
No. Convolutional / Pooling Layers	{1,2,3}	Pooling Kernel Size in a Layer	{2,3}
No. Weights in a Convolutional Layer	{1,2,3, ...,128}	No. Dense Layers	{1,2,3}
Convolutional Kernal Size in a Layer	{1,3,5}	No. Weights in a Dense Layer	{1,2,3, ...,128}

Table 3.1: Details the optimisation search space for each of the CNN’s architectural hyperparameters. The Bayesian optimisation algorithm did not trial every possible hyperparameter combination but rather tested combinations that were restricted to the defined search space.

Chapter 4

Results

4.1 Synthetic Data Generation

Overall, the synthetic data generation process produced synthetic images that were similar in appearance to real, experimentally obtained images (such as in figure 2.2). The initial stage of the synthetic data generation process successfully sampled proton energies from the theoretical distribution (equation 2.2). For $E_{max} = 3.00$ MeV and $T_p = 2.00$ MeV with 10^5 proton samples, a histogram of the number of computationally sampled protons against energy is presented in figure 4.1. The proton energy histogram bins appear to conform closely to the shape of the theoretical distribution with a sharp cut-off at E_{max} , demonstrating the robust performance of the proton energy sampling.

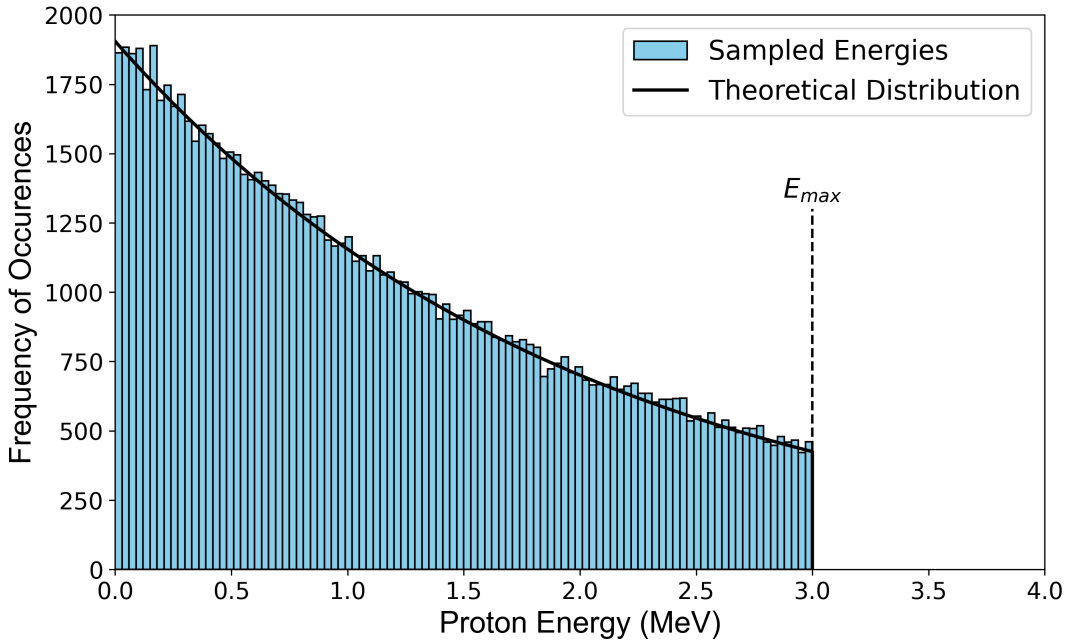


Figure 4.1: Histogram of the frequency of occurrences against the proton energy in MeV for a simulated proton beam with $E_{max} = 3.00$ MeV and $T_p = 2.00$ MeV. The agreement of the histogram bars with the theoretical PDF distribution described by equation 2.2 illustrates the accuracy of the inverse transformation method described by equation 3.1.

Example images have been generated with the filter thickness configuration presented in figure 4.2. Figure 4.3 reflects the contrast between an image with a relatively low E_{max} and high N_0 and an image with a high E_{max} and low N_0 . As anticipated, the high E_{max} image has protons depositing energy in the scintillator through a full range of filter thicknesses in contrast to the low E_{max} image where the most energetic protons are only able to pass through the two thinnest filters visibly. The impact of the variation in N_0 is also demonstrated, with the high N_0 image containing pixel values greater than 1750 while the low N_0 image has its highest pixel value at just over 80. Figure 4.4 demonstrated the contrasting images

produced when varying T_p while the other parameters are held approximately constant. The high T_p image shows more protons passing through greater filter thicknesses in contrast to the low T_p images where an insufficient number of protons are able to penetrate the higher filter thicknesses to be visible. This makes sense in relation to equation 2.2 as higher values of T_p would result in a shallower pitch of the PDF and so a greater quantity of protons produced closer to E_{max} .

90 μm	40 μm	20 μm
9 μm	4 μm	2 μm
1 μm	0.5 μm	0.2 μm

Figure 4.2: Layout of the filter thicknesses used in the generation of the synthetic image examples seen in figure 4.3 and figure 4.4.

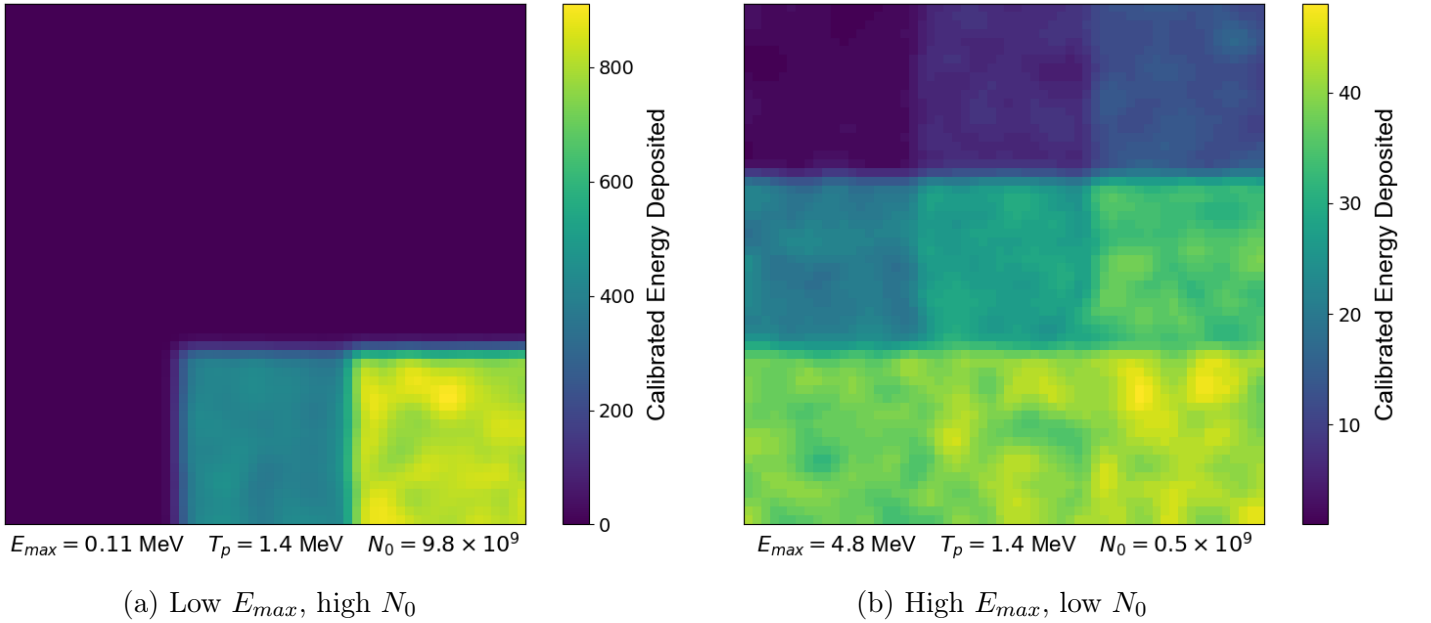


Figure 4.3: The impact of the E_{max} and N_0 parameters are illustrated through the comparison of two synthetic images. The first image is generated with a low E_{max} , and high N_0 while the second is generated with a higher E_{max} but lower N_0 . T_p is approximately equal in the two images.

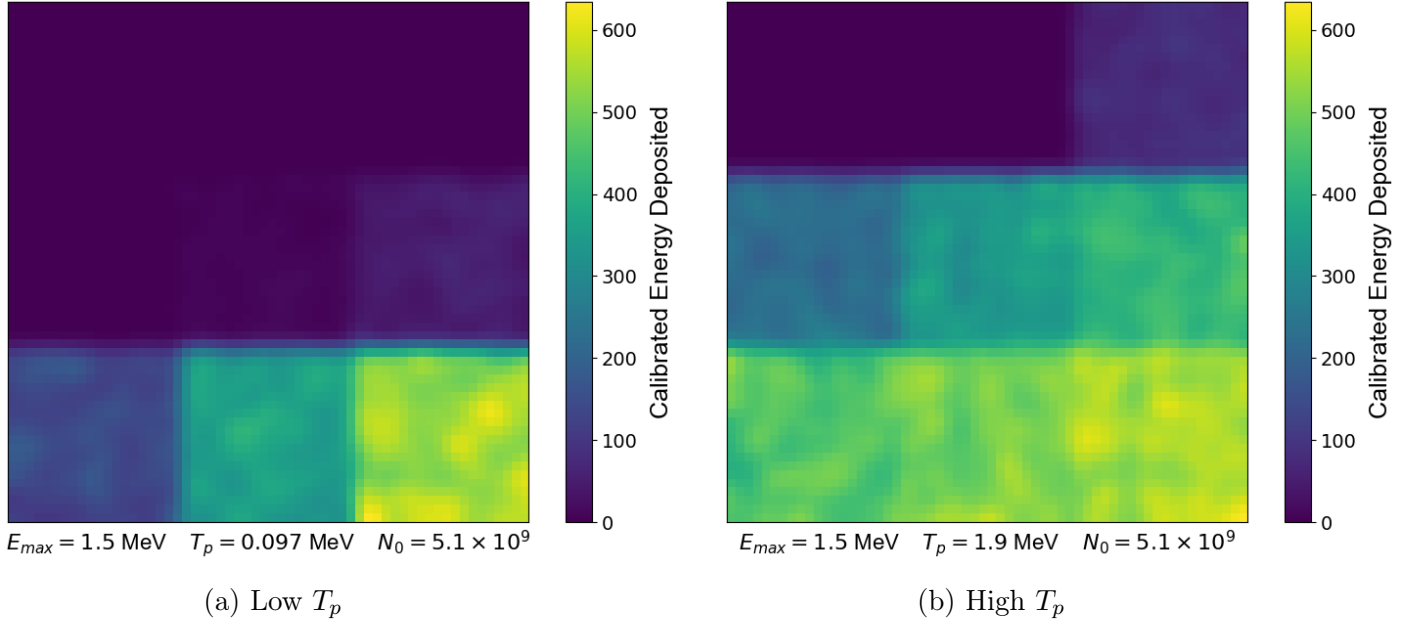


Figure 4.4: The contribution of the T_p parameter is illustrated by comparing two synthetic images. One image is generated with a low T_p while the other is generated with a higher T_p . E_{max} and N_0 are approximately equal in both images.

4.2 Neural Network and Bayesian Optimisation Performance

After 116 Bayesian optimisation trials optimising filter thicknesses and 360 trials optimising the CNN architecture, figure 4.5 summarises the MSE performance of the resulting CNN compared to generating data with randomly sampled filters and an unoptimised CNN architecture. The CNN trained on the optimised filters performed 40% better compared to training on data generated with randomly sampled filters from the 27 discrete thickness options. A more incremental 10% further improvement in MSE performance was exhibited when optimising the CNN architecture in addition to training on data generated with optimised filter thicknesses. The optimal a and b parameters selected by the optimisation in equation 3.3 were $a = 88.5 \mu\text{m}$ and $b = 0.472$. Figure 4.6 displays the resulting set of discrete filter thicknesses. These thicknesses range from $0.2 \mu\text{m}$ to $90 \mu\text{m}$ with the complete set of optimised thicknesses presented in figure 4.2 and their spatial layout displayed in figure 4.2. This distribution spanned almost the full range of filter thickness options ($0.1 \mu\text{m}$ to $90 \mu\text{m}$).

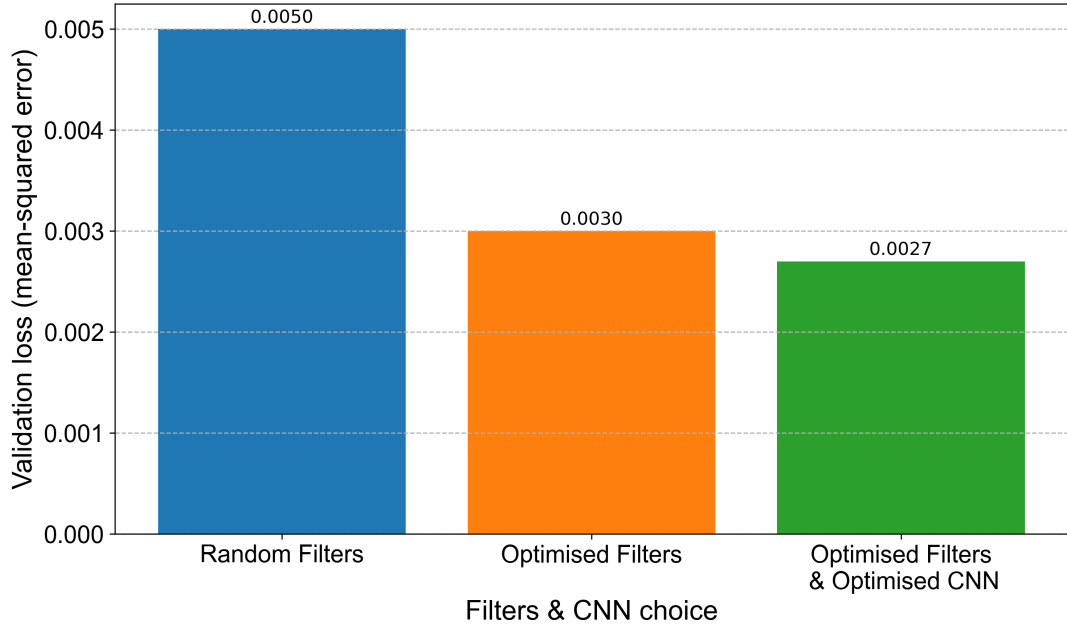


Figure 4.5: Bar chart summarising the performance of the CNN, in terms of MSE , on a validation dataset containing 25,000 synthetic images for three approaches: Using random filters and an unoptimised CNN architecture, optimised filters and an unoptimised CNN architecture, and the use of both optimised filters and an optimised CNN architecture.

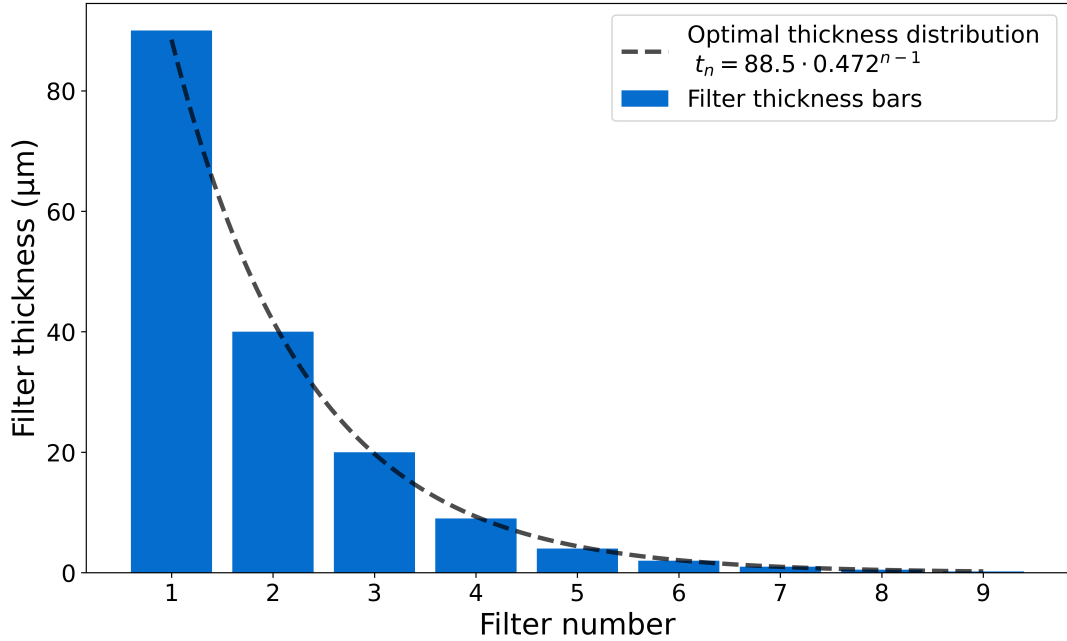


Figure 4.6: Bar chart displaying the optimised filter thicknesses chosen by the Bayesian optimisation algorithm after 116 trials. The dotted line represents equation 3.3, with the optimised a and b parameters, from which the filter thicknesses were sampled.

4.3 Optimal Configuration Metrics

The mean relative absolute error (MRAE) was used to compare the performance of the CNN in predicting the three beam parameters. The obtained MRAE values were 7% for E_{max} , 13% for T_p , and 4% for N_0 . Figure 4.7 demonstrates how the relative error varies with the three beam parameters. The relative error of N_0 appeared most consistent throughout the full range, contributing to having the lowest MRAE of the three parameters. In contrast, T_p was greatly underpredicted at low values leading to the poorest performance. The relative error of E_{max} was notched with an increasing periodicity, potentially a result

of having nine discrete filters (as opposed to sloped filters). When testing on a project member’s laptop, the optimised CNN had a mean prediction time of 7.3 ms with a standard deviation of 0.8 ms. This is promising in the context of 100 Hz lasers as it supports the possibility of achieving beam diagnostics faster than the rate of these laser pulses.

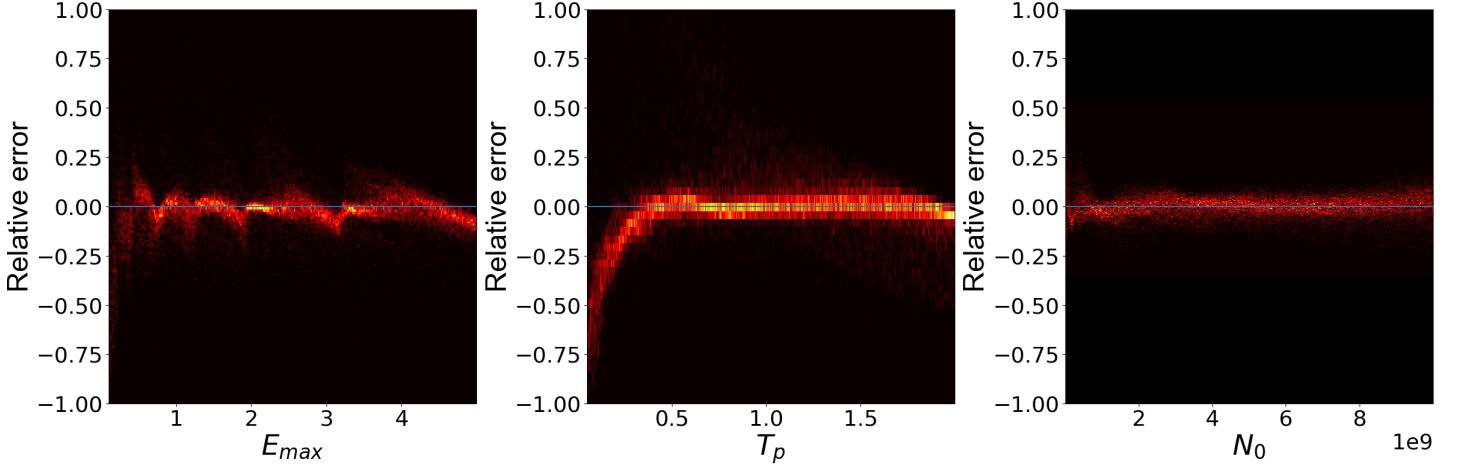


Figure 4.7: 2D histogram of the relative errors of the most performant CNN’s beam parameter predictions on a validation dataset of 25,000 images. Each beam parameter relative error is displayed as a function of the beam parameter itself.

4.4 Discussion of Results

There was a significant accuracy improvement in the CNN predictions after the Bayesian optimisation algorithm selected filter thicknesses in comparison to randomly selected filters. This underscores the benefit of employing Bayesian optimisation to not only optimise CNN hyperparameters but also the physical attributes used in the synthetic image generation. This can help inform the selection of filter thicknesses used in real implementations that require rapid beam diagnostics. The optimisation of the CNN architecture producing more incremental improvements in comparison to the filter thicknesses further highlights the relative importance of the choice of filter thicknesses while suggesting that there could be more scope for improvement in optimising this area. Only the CNN architectural hyperparameters were optimised while other non-architecture hyperparameters could potentially yield more significant performance improvements. These include training-specific hyperparameters such as learning rate, choice of optimiser and regularisation parameters.

The optimised filter thicknesses spanned almost the full range of possible filter thicknesses provided for optimisation. This implies that the ‘true’ optimised filter thicknesses could fall outside this range. Furthermore, using equation 3.3 to sample filter thicknesses was primarily driven by heuristic reasoning and the necessity of decreasing the optimisation search space. A more general approach would be to optimise the value of the smallest filter thickness with the differences between filter thicknesses, though this would result in a search space of nine parameters rather than two. To address the notched appearance of the E_{max} relative errors along with the poor performance at low T_p , a greater number of filters could be used in the synthetic image generation. This may increase the frequency of notches in the E_{max} errors so each notch would have less runway to deviate before resetting. Having consecutive filters of closer thicknesses could help differentiate between beams with lower values of T_p as these correspond to having a lower spread of proton energies with fewer close to E_{max} .

Chapter 5

Improvements and Future Work

5.1 More Realistic Data Generation With Geant4

The current synthetic image generation process is guided by well-established laser-driven ion beam principles such as those described by equation 2.2 and reputable NIST databases to determine the energy deposited by protons in the aluminium filters and scintillator. Although a Gaussian blur was applied to account for experimental noise, other aspects of the physical process were not implemented, such as the coulomb scattering of protons passing through solids and the proton beam's angular distribution [9]. Geant4, a software toolkit for simulating the passage of particles through matter could be used to inform the custom synthetic image generation process [17]. Geant4's capabilities provide the tools to simulate the passage of proton beams through aluminium filters and scintillators. This can be used to create highly realistic synthetic images which can be compared against the custom implementation. The Geant4 simulations can guide the creation of tailored synthetic image generation, optimising the balance between realism and computational efficiency.

5.2 Ground Truth Comparison

Even if the CNN can perform with near-perfect accuracy predicting beam parameters on a synthetic validation dataset, this does not guarantee accurate predictions on images generated from real experiments. There could still be unaccounted factors that cause differences in how models perform on synthetic versus real images. To address this, a high-quality set of labelled real images should be obtained to evaluate the CNN's performance. This can be achieved through existing but more time-consuming methods involving the use of radiochromic film (RCF) detector stacks to reconstruct the proton energy distribution [18]. This approach could be run in parallel with the 3x3 aluminium filters method to obtain real images that can be accurately labelled using the conventional RCF method. This would provide a set of real, labelled images for the CNN's performance to be validated against.

Chapter 6

Conclusion

To address the increasing demand for rapid laser-driven ion beam diagnostics an approach using CNNs to predict three key proton beam parameters, E_{max} , T_p , and N_0 , has been tested. Synthetic image generation was employed to simulate the passage of laser-driven proton beams through aluminium filters of varying thicknesses and a scintillator. This approach was used to generate a large set of 100,000 images for the CNN to be trained on and validated. In conjunction with this, Bayesian optimisation was used to determine optimised filter thicknesses in the synthetic image generation and to optimise the CNN architecture. This approach resulted in a CNN that, in terms of mean-squared error, performed 40% better with optimised filter thicknesses compared to using random filters with the same CNN. A more incremental 10% further improvement was noted when optimising the architecture of the CNN while using the optimised filters. This result highlights the strong potential of Bayesian optimisation in optimising aspects of the physical setup rather than solely the common use case of optimising neural network hyperparameters. The most optimal CNN was able to predict with MRAE values that were 7% for E_{max} , 13% for T_p , and 4% for N_0 . When tested on a specific hardware configuration, the CNN achieved a mean prediction of 7.3 ms with a standard deviation of 0.8 ms. This result is particularly promising as it is within the time required for use in cutting-edge 100 Hz laser systems. Further steps necessary before employing this diagnostic approach in real experiments were outlined. This included the use of Geant4 to guide more accurate synthetic image generation and approaches to obtain a set of real, labelled images to test the CNN's performance against.

Acknowledgments

I would like to extend my gratitude to Dr. Nicholas Dover for his exceptional guidance throughout the whole course of the project. His support has helped spark my strong interest in areas such as machine learning and high-performance computing which I am now keen to pursue in my future career. I would also like to thank Dr. Mark Scott, for taking time out of his schedule to discuss the implementation of Geant4 in our project. I also thank my project partners who have devoted significant time and effort to this project resulting in a productive yet enjoyable experience. I hope this collaboration paves the way for future work together, whether in continuing this project or other endeavours. Finally, I acknowledge the computational resources and support provided by the Imperial College Research Computing Service (<http://doi.org/10.14469/hpc/2232>).

Bibliography

- [1] J. Schreiber, P. R. Bolton, and K. Parodi. Invited Review Article: “Hands-on” laser-driven ion acceleration: A primer for laser-driven source development and potential applications. *Review of Scientific Instruments*, 87(7):071101, 07 2016.
- [2] SC Wilks, AB Langdon, TE Cowan, M Roth, M Singh, S Hatchett, MH Key, D Pennington, A MacKinnon, and RA Snavely. Energetic proton generation in ultra-intense laser–solid interactions. *Physics of plasmas*, 8(2):542–549, 2001.
- [3] J Badziak. Laser-driven ion acceleration: methods, challenges and prospects. In *Journal of Physics: Conference Series*, volume 959, page 012001. IOP Publishing, 2018.
- [4] U Masood, M Bussmann, TE Cowan, W Enghardt, L Karsch, F Kroll, U Schramm, and J Pawelke. A compact solution for ion beam therapy with laser accelerated protons. *Applied Physics B*, 117:41–52, 2014.
- [5] Herman Suit, Saveli Goldberg, Andrzej Niemierko, Alexei Trofimov, Judith Adams, Harald Paganetti, George TY Chen, Thomas Bortfeld, Stanley Rosenthal, Jay Loeffler, et al. Proton beams to replace photon beams in radical dose treatments. *Acta Oncologica*, 42(8):800–808, 2003.
- [6] Florian Rudolf Schroeck, Bruce L Jacobs, Sam B Bhayani, Paul L Nguyen, David Penson, and Jim Hu. Cost of new technologies in prostate cancer treatment: systematic review of costs and cost effectiveness of robotic-assisted laparoscopic prostatectomy, intensity-modulated radiotherapy, and proton beam therapy. *European urology*, 72(5):712–735, 2017.
- [7] J Peñas, A Bembibre, D Cortina-Gil, L Martin, A Reija, C Ruiz, M Seimetz, A Alejo, and J Benlliure. A multi-shot target-wheel assembly for high-repetition-rate, laser-driven proton acceleration. *arXiv preprint arXiv:2401.10173*, 2024.
- [8] YX Geng, D Wu, W Yu, ZM Sheng, S Fritzsche, Q Liao, MJ Wu, XH Xu, DY Li, WJ Ma, et al. Proton beams from intense laser-solid interaction: Effects of the target materials. *Matter and Radiation at Extremes*, 5(6), 2020.
- [9] DA Mariscal, BZ Djordjević, ES Grace, R Hollinger, T Ma, GG Scott, H Song, RA Simpson, JJ Rocca, and S Wang. Design of flexible proton beam imaging energy spectrometers (probies). *Plasma Physics and Controlled Fusion*, 63(11):114003, 2021.
- [10] M Roth and M Schollmeier. Ion acceleration-target normal sheath acceleration. *arXiv preprint arXiv:1705.10569*, 2017.
- [11] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.
- [12] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.
- [13] National Institute of Standards and Technology. Pstar: Stopping power and range tables for protons. Available at: <https://physics.nist.gov/PhysRefData/Star/Text/PSTAR.html>, 2024.

- [14] Keiron O’shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.
- [15] François Chollet et al. Keras. <https://keras.io>, 2015.
- [16] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [17] Sea Agostinelli, John Allison, K al Amako, John Apostolakis, H Araujo, Pedro Arce, Makoto Asai, D Axen, Swagato Banerjee, GJN Barrand, et al. Geant4—a simulation toolkit. *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.
- [18] Marius Schollmeier, Matthias Geissel, Adam B Sefkow, and KA Flippo. Improved spectral data unfolding for radiochromic film imaging spectroscopy of laser-accelerated proton beams. *Review of Scientific Instruments*, 85(4), 2014.