UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
Zepler institute

The Use of Machine Learning Techniques for the Optimisation of Experimental Laser Machining Processes

by

Michael David Tom McDonnell

MPhys

ORCiD: 0000-0003-4308-1165

A thesis for the degree of
Doctor of Philosophy

June 2023

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
Zepler institute

Doctor of Philosophy

The Use of Machine Learning Techniques for the Optimisation of Experimental Laser Machining Processes

by Michael David Tom McDonnell

In recent decades, laser machining has transformed manufacturing. Pulsed laser machining, using short and ultra-short pulse durations, has become a standard technique for the fabrication of features on micro and nano scales. At these power and length scales, many interesting non-linear effects occur, making accurate modelling of these processes challenging without introducing simplifications and assumptions. This thesis presents an alternative to traditional modelling techniques, namely using machine learning, in addition to predictive regression, full experimental modelling is conducted.

A broad set of machine learning techniques has been applied to the field of laser machining. This includes both analytical and generative modelling techniques, showing that neural networks can achieve very high levels of accuracy. These techniques are also extended to situations that would be impossible to model, where knowledge of the system is very limited. In these situations, the presented methods were still able to achieve high accuracy and be used for data optimisation tasks.

In addition, generative methods are applied for the prediction of depth profiles from a variety of machining parameters. It is also shown that generative networks could complement the experimental process by reproducing the results found. This presents opportunities both for increased levels of data collection and for easy presentation and investigation of new ideas before committing to a full experimental process.

Contents

List of Figures

List of Tables

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. None of this work has been published before submission

Signed:..... Date:.....

Acknowledgements

My thanks go out to Ben for giving me the opportunity to complete this massive undertaking. Little did I know when meeting you for the first time during my Masters' dissertation thesis that I would be working with you for the next four years. I also want to thank you for the help you have given and the enthusiasm you have had throughout that time.

I would also like to give thanks to fellow group members, past and present, postdoc and PhD students. Working with you all has been a pleasure and some great work has been conducted in that lab. My thanks go especially to Matt, whose encyclopedic automation knowledge helped make many of the experiments possible, and to Dan who really helped me get to grips with both the lab equipment and machine learning.

Even with all of that, none of this would have happened without the help and support of all my friends and family, especially Nikki who has put up with all of the weird hours I would work, coding long into the night, or sitting watching the networks train, hoping that this time the hyperparameters were "just right".

The work in this thesis has also been supported by the Engineering and Physical Sciences and Research Council (EPSRC) grant EP/N03368X/1.

I also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

Definitions and Abbreviations

FS	Femtosecond
NN	Neural Network
CPA	Chirped Pulse Amplification
TTM	Two-Temperature Model
CW	Continuous Wave
TBP	Time Bandwidth Product
GAN	Generative Adversarial Network
MAE	Mean Absolute Error
MSE	Mean Squared Error
ML	Machine Learning
kNN	k-Nearest Neighbour
SVM	Support Vector Machines
SVR	Support Vector Regression
BCE	Binary Cross Entropy
GPU	Graphical Processing Unit
TPU	Tensor Possessing Unit
RQ	Rational Quadratic
RBF	Radial Basis Function
STD	Standard Deviation
DMD	Digital Micro-mirror Devices
SLM	Spatial Light Modulator
WLI	White Light Interferometer
ANN	Artificial Neural Network
ReLU	Rectified Linear Unit
CNN	Convolutional Neural Network
EM	Electromagnetic
− − − − − − − − −	
λ	wavelength
c	speed of light
L	cavity length (2.3), Length (2.3.1)
n	refractive index (2.3.1), number of pixels above the threshold (7.4.2)
τ_{min}	minimum pulse duration

f	focal length
f_{XX}	function
ν	bandwidth
C	heat capacity
T	temperature (3.2.1) , temporal evolution of the pulse (3.2.1)
k	thermal conductivity
g	electron-phonon coupling
S	spatial power distribution (Laser)
R	reflectance of the material
δ	optical skin depth
δ_b	ballistic skin depth
ω	frequency
ω	beam diameter
ω_0	beam waist diameter
t_p	pulse duration
$L_1\ loss$	mean absolute error loss
$L_2\ loss$	mean squared error loss
y	true result
$\hat{y}$	predicted result
L_{CE}	cross-entropy loss
L_{hinge}	hinge loss
w	weighting
x	network input
b	bias
k	kernel size
o	output
p	padding
γ, β	modulation tensors
X	depth profile
W	weighting vector
Z	noise input
G	generator
d	diffraction limit
NA	numerical aperture
r	pixel radius

Chapter 1

Introduction

Laser machining is a non-contact material removal technique achieved through light-matter interactions between a laser and a sample. In general, the incident laser beam heats the target as energy is absorbed, causing melting, ablation and vaporisation of the material. A certain type of laser, pulsed lasers, will be investigated throughout this report with a focus on ultra-short pulse lasers. These are typically lasers with a pulse duration of at most tens of picoseconds, and typically hundreds of femtoseconds. For this reason, ultra-short pulse lasers will be referred to as femtosecond (FS) lasers.

Machine learning is a subsection of artificial intelligence where a computer is trained to be able to complete a specific task without explicit programming. This can be done by providing the computer data, a way to create similar data, and a way to quantify its success in the creation process. This report will focus on the use of supervised learning where explicit inputs and outputs are provided, with the output of the algorithm compared directly to the desired output.

The objective of this project is to develop a methodology that will lead to an improvement of laser machining processes. The hypothesis is that this will be achieved by utilising machine learning, with an emphasis on generative neural networks. These improvements will focus on increasing the efficiency of material removal, decreasing the minimum possible feature size, and making the outcome of laser machining more closely resemble the desired outcome.

1.1 Motivation

Laser machining is important for industry, from the fast cutting of continuous wave lasers to the high precision offered by FS lasers. However, laser machining, especially for ultra-short pulses, is an extremely complex process. Modelling and simulation of

the process are important to understand how to optimise the parameters used, otherwise a guided but systematic search is often required. This is time-consuming and requires expertise in the field and familiarity with the system. Existing modelling approaches, based on fundamental physics, do not scale up to useful sizes, as the number of interactions takes too much computer processing. Rather than starting with the underlying physics, the data generated from machining can be used directly to model and predict the entire process.

1.2 Prior Art

As the laser machining process can be complex to model many research groups have started to investigate how machine learning techniques can be used to assist the experimental and development processes. All throughout industry and academia these techniques have been used to enhance processes that are hard to simulate or predict.

Within laser machining, artificial neural networks have been used extensively since the early 2000s to perform task such as predicting machining parameters Casalino et al. (2017); Campanelli et al. (2013); Yousef et al. (2003); Jeng et al. (2000); Casalino and Ludovico (2002) and predicting the outcome of the machining process Yousef et al. (2003); Zhang et al. (2005); Cheng and Lin (2000); Chang and Na (2001).

While some machine learning techniques have been employed for a long time, an area of massive growth is the use of deep learning for many aspects of industry in general. One of the biggest improvements allowed through deep learning is the introduction of 2-dimensional data into the prediction process, a key step in tasks such as the real-time monitoring of beam aberrations Mills et al. (2019a) and preventing over machining Xie et al. (2019).

Deep learning techniques can also be used outside of monitoring tasks, allowing for highly detailed predictions such as predicting hardness distribution for heat-treated steel when machined with a 2kW laser Oh and Ki (2019). Outside of laser machining of metals, neural networks have been used across industries such as being used to control optical tweezers Praeger et al. (2021).

While deep learning has seen a lot of activity, generative modelling is a much smaller field. Examples of where it has been used within the laser industry include predicting the appearance of dough when browned with a CO_2 laser Chen et al. (2019) and prediction of microstructure formation that occurs during sintering of alumina Tang et al. (2021). Within the category of laser machining initial work has been conducted on the visualisation of machining with fibre lasers Courtier et al. (2021b,a) along with

predicting the results of machining with shaped pulses Heath et al. (2018b); Mills et al. (2018).

As the work completed in this thesis covers a range of topics, a separate literature review has been conducted in each chapter where relevant to ensure that the concepts discussed are understood prior to reading. For an in-depth overview of the use of machine learning across the laser machining industry, consult Mills and Grant-Jacob (2021).

1.3 Thesis Outline

This report covers a range of both experimental and computational techniques which will be discussed in detail in the relevant chapters.

An overview of the fundamental physics of lasers and their uses is given in Chapter 2, where both theory and equipment are discussed. Along with this, analytical methods (i.e. from fundamental physical equations) to simulate the effects of laser-material interactions are explored and the associated challenges presented.

The other major focus of the report is machine learning and its application to experimental research. An introduction to this area is covered in Chapter 3 where several techniques for a range of applications are explored. Along with mathematical and component descriptions of the techniques, there is a particular focus on the types of networks that are employed later in the report.

Neural networks have been used since 1993 Tóth et al. (1993) to predict the results of laser machining and optimise laser parameters. This idea is extended in Chapter 4 to include multiple parameters over a wide range along with experiment modelling. A selection of analytical and generative techniques are explored and their relative strength is presented. This chapter presents work completed in McDonnell et al. (2021a).

Chapter 5 explores two highly related pieces of work. The first represents a significant advance on previous work aimed at predicting the effects of shaped laser pulses using generative neural networks. This chapter focuses on extending the prediction abilities to a sequence of up to 3 pulses (from a known initial condition) while maintaining accuracy. This network is then used to simulate an experimental environment used to train a separate network designed to perform the inverse transformation. This network implements a novel training strategy as well as presents the possibilities of simulated experimental environments. This chapter presents work completed in McDonnell et al. (2020) and McDonnell et al. (2021b).

In many situations, teams will work together, both in academia and in industry, and not all participants will have full access to the data collection processes. This problem is investigated in Chapter 6 where the experiment is treated as a black box and details of the data are kept hidden. Along with presenting that networks can be trained without any prior information about the data and still perform well, optimisation tasks aligned with typical practices are explored.

The final chapter of the thesis, Chapter 7, summarises the findings from all of the preceding experimental chapters. In addition to this, key areas for progress are identified and the potential next steps are presented.

Chapter 2

Pulsed Laser Machining

This chapter will discuss the use of, and some of the physics behind, lasers and laser machining. This will include some underlying theory describing the different effects seen with different types of lasers and explaining why they might be used. As there is a large focus in this thesis on machining with ultra-short pulses, methods that are involved in generating them, such as mode-locking and chirped pulse amplification (CPA), will be explored.

2.1 The use of Lasers in Machining

Lasers are becoming an increasingly popular option for material processing, and the laser industry was valued at $17.48 billion in 2021 with expectations of its continued growth noa. Lasers are used throughout a huge range of industries, ranging from those that need massive power and high throughput, such as welding and cutting Cao et al. (2003, 2006); Choudhury and Shirley (2010); Schuocker (1989), to those which involve much finer control, such as surface texturing Arnaldo et al. (2018); Riveiro et al. (2018); Voevodin and Zabinski (2006); Ryk et al. (2002). Lasers that produce a constant beam are referred to as Continuous Wave (CW) lasers and are often used when the process requires a large amount of total energy. Some of the most common forms of lasers used for this are gas lasers, such as CO_2 lasers, many operating at powers of up to 15 kW Nath et al. (2005). Another type of laser that is often used in these high-energy applications is fibre lasers, capable of powers of up to 100 kW Shcherbakov et al. (2013). CW lasers are used for these processes as they can reach much higher average powers than pulsed, and especially short-pulse, lasers. Lasers are used in industry as they offer a non-contact, therefore little to no wear, machining option that provides consistent results while only removing a minimal amount of material. Despite this CW lasers do have limitations, they are unable to match the

cutting depth of some other techniques, such as plasma cutting, and do not allow for the precision allowed by pulsed lasers.

$$P_{peak} = \frac{P_{avg}}{t * f_{rep}} \tag{2.1}$$

While a lot of industrial-aimed research work has revolved around increasing the power available from lasers, applications that require precise laser control are often needed. One type of laser where power is no longer a major limiting factor is an ultra-short pulse laser. With ultra-short pulse lasers, the material can be removed precisely and on very small scales assuming that the correct machining parameters are used. Ultra-short pulse lasers are often defined as pulsed lasers, where the pulse duration is at most tens of picoseconds Paschotta. Most systems have even shorter pulses, with a lot of commercial systems along with the ones used in the group, having pulse durations of hundreds of femtoseconds. Therefore, from this point onwards in this document, ultra-short pulse lasers with pulse durations on this scale will be referred to as femtosecond (FS) lasers. The average power from solid-state FS lasers is often only rated at 10s of watts and the most powerful fibre lasers have reached up to 1 kW while CW systems can reach 10s of kWs. Even though the average power is much lower, their short pulse duration allows them to achieve very high peak powers. To calculate the peak power available from a pulsed laser, Eq. 2.1 can be used. An FS laser with average power 16 W, pulse duration 250 fs, and repetition rate of 100 kHz will reach a peak power of 640 MW. This is much higher than the peak power that is available from CW lasers, allowing FS lasers to machine structures at scales and on materials not possible by other methods Heath et al. (2017); Gattass and Mazur (2008).

The two types of lasers discussed are designed to operate under different power and energy regimes. The CW lasers are aimed at the deposition of very large amounts of energy to be able to machine as much material as possible. This machining can come in the form of removal via vaporisation or melting & blowing or thermal shocking due to the extreme temperatures involved. This large amount of energy being transferred to the sample means that the material surrounding the machined area will have large heat-affected zones that will impact the properties of the sample. Even though the peak power can be much higher in pulsed lasers, the total amount of energy that the sample receives is far lower, especially when not using the full repetition rate of the laser.

2.2 Short Pulsed Lasers

Originally thought to happen instantaneously due to an inability to measure at the time scales involved, electrons in the sample respond rapidly (on the order of 100 as

Hassan et al. (2016)) to the laser's electric field, absorbing its energy. In normal conditions each electron absorbs the energy from a single photon, becoming excited to a higher band if the incoming photon had sufficient energy. A photon with sufficiently high energy could excite an electron beyond the ground state, allowing it to become detached from its original atom independent of the intensity of such photons in an effect known as the photoelectric effect. The fact that electrons can only be emitted once the incident light had reached a certain frequency was part of the evidence that light could also act as particles as well as a wave. While the electron excitation speed is still fast relative to the shortest pulsed lasers that have been made, the relaxation time for these electrons, the time taken for them to return to their original energy state is much longer, around 100 fs to 1 ps von der Linde et al. (1997) by emitting phonons, quasi-particles carried by vibrations in the crystal lattice. With the advances in laser technology, this is now very comparable with some of the shorter pulse lengths achieved in industrial lasers. Even when the durations of the pulses are larger than the relaxation time, the high peak energies and small beam sizes involved lead to extremely high intensities, and therefore densities, of photons. When there are this many photons in a small space and a short amount of time, some of the photons do not interact with electrons at rest, but instead with ones that have already become excited to a virtual excitement level. This allows for band gaps larger than the energy of the individual photons to be breached. This effect is known as multi-photon absorption and is a key player in the use of ultrashort pulse lasers. The ability of an electron to absorb more than one photon allows the electron to be exited in a situation where it would not normally be possible due to the individual photons being of lower energy than the smallest band gaps. This in turn allows lasers that have high enough intensities to be able to machine materials that would normally be transparent to the wavelength of the laser.

2.2.1 Nanosecond Regime

If the duration of laser irradiance is much greater than the electron relaxation time, then the electrons and the atomic lattice approach a state of dynamic thermal equilibrium. This condition is the case for continuous wave or longer pulse lasers. This effect can even be seen when moving into the nanosecond regime with lasers that would be considered as having a short pulse duration. Nanosecond lasers at typical powers used in industry may create the multiphoton absorption effect described above but the majority of the energy in the pulse with be deposited into the sample. As the electrons excited by the incoming pulse relax some of the stored energy is passed to the surrounding atomic latice, with the rest being reemitted in the form of photons. In the case of a nanosecond laser, the duration of the pulse is far longer than the combined excitation and relaxation time of the electrons, each electron can become excited multiple times, each time transferring a portion of its energy into the

surrounding material. Due to this, and especially true at lower intensities the major removal mechanism in nanosecond lasers is melting. Despite this, some electrons will experience multiple photon interactions before relaxation and so will become ejected from the surface. This will lead to some of the material becoming ionised and violently ejected. This rapid expansion of the material, combined with the melting of the material from heating causes both high removal rates along with unclean machining. Holes made with lasers with nanosecond length pulses are often surrounded by a crown of raised material, formed of solidified molten slag, and exhibit large amounts of debris.

2.2.2 Femtosecond Regime

When an electron is emitted from the sample, the energy that it absorbed is also lost from it. This means that even though large amounts of energy are deposited into the sample in very short time frames from a femtosecond laser, there is less thermal damage than can be found with lasers that have lower peak energy but a longer pulse. This escaped electron also leads to the final sample being ionised as it has lost a negatively charged particle. This ionised material is then quickly repelled by the surrounding material, causing it to be vaporised and expelled as plasma. This expelling of material further acts to dissipate the energy of the original photons, reducing the temperature increase experienced by the sample. If the peak power of the laser is high enough, and therefore has a higher intensity at the same spot size, large amounts of material will be ionised, vaporising it and expelling it in a mixture of plasma and other species in a process called ablation. This is one of the main advantages of using an ultra-short pulsed laser.

The lack of heat transfer arising from femtosecond pulses gives rise to another feature of their use. In CW lasers, and those with pulse durations of at least nanoseconds the heat build-up will be significant. This heat transference can be limited to avoid thermal damage and melting to the sample, in these situations it will also be true that little machining will occur. Even when intensities are increased for nanosecond lasers to the point that ablation will occur, the length of the pulse will mean that thermal equilibrium will be reached more rapidly and extensive heat damage be present. In contrast to this, when machining with a femtosecond laser at low intensities, there will be very limited impact on the sample. There will be some heat transfer from the relaxation of electrons, but the sample will be far from thermal equilibrium, especially if the repetition rate of the laser is kept low. This leads to the existence of the ablation threshold for materials, below which the intensity of laser will be insufficient to induce multiphoton absorption, and as such the ionisation of material required for ablation will not take place.

When using a laser to precisely machine a material via ablation, it is important to use a peak power that is only slightly above the ablation threshold of the material. This acts to improve the quality of the machined feature since too high a power would likely lead to a crown of molten material around the feature. It will also reduce the ionisation of the surrounding medium to a minimum. As opposed to the other two effects discussed (melting and ionisation of the sample), ionisation of the surrounding medium does not contribute to material removal. In contrast, this effect is detrimental to the laser machining process and should be avoided. This effect occurs when the intensity of the beam within the focus causes multiphoton interactions with the surrounding medium. This leads to scattering of the beam and decreases the fluence on the surface, reducing the resulting machining quality.

2.3 Generating Ultra-short Pulses

Reaching this point in the creation of pulsed lasers has been a long process. In this section, some of the techniques that are used to produce short pulses will be discussed. Along with this, some of the theories that underpin the generation of these pulses will be explored.

2.3.1 Mode Locking

$$\lambda_i = \frac{2L}{i} \qquad \omega_i = \frac{i \cdot c}{2nL} \tag{2.2}$$

One of the major components of a laser is the cavity which comprises a series of mirrors designed to reflect light and set up a series of standing waves. Due to the formation of these standing waves, the laser cavity can also be referred to as a laser resonator. Within the laser resonator, many longitudinal modes will form, with the number and size dependent on the cavity length of the laser. Each standing wave will have a possible wavelength and frequency found by applying Eq. 2.2 where L is the cavity length and n is the refractive index of the cavity. This equation can then be used to calculate the frequency separation between adjacent modes as shown in Eq. 2.3.

$$\Delta\omega = \omega_{i+1} - \omega_i = \frac{c}{2nL} \tag{2.3}$$

On their own, the existence of standing waves does not contribute to the production of a laser pulse, however, when superimposed they can combine to form a beat. This is achieved by having a fixed phase relationship between the different modes, the more correlated they are the stronger and more defined the beat will be. The effect of superposition with both fixed and random phase relationships can be seen in Fig. 2.1

where each sine wave represents a standing wave frozen in time within the cavity. While these waves all oscillate around zero, they have been shifted in this figure to provide more clarity on what is occurring. The top two lines (orange and blue) represent different possible modes, while the other two (red and green) each represent a superposition of many modes. The lower line (green) shows the effect of superposition when the modes have a constant phase relationship, while in the other (red) the phase relationship is random.

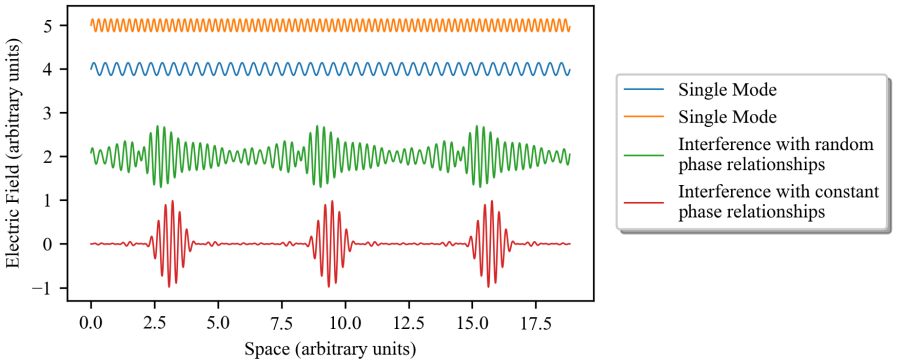


FIGURE 2.1: The effect of phase relationship on laser pulse train. The top two waves represent a single frequency each. The next wave represents a combination of 20 frequencies with a random p[hase relationship. The bottom wave represents the interference pattern from 20 frequencies with a constant phase relationship.].

As can be seen in Fig. 2.1 the existence of any single standing wave will never cause a beat to occur, to form a pulse the light must consist of multiple wavelengths, as without it there can be no constructive interference of multiple modes. The maximum number of modes that can contribute to lasing within each laser is determined by combining the mode spacing, the frequency dependency of the gain material used, and the laser threshold. The breadth of possible frequencies found in the resultant output is known as the bandwidth of the laser.

Even though many modes may be present within the laser cavity,m not all will be able to contribute to the final produced pulse. Only those frequencies that are able to achieve a net round trip gain after passing the amplification medium will contribute to lasing and this effect can be seen in Fig. 2.2. The minimum achievable pulse duration for a given bandwidth is determined by the Time-Bandwidth Product (TBP) which itself is dependent on the temporal shape of the wave. The TBP is calculated using a Fourier transform of the spectral components, and for a Gaussian wave has a value of 0.441. The value of the TBP can then be combined with the bandwidth to give a minimum value for the pulse duration as shown in Eq. 2.4 where τ_{min} is the minimum pulse duration and ν is the bandwidth in hertz. As can be seen, a laser with a higher bandwidth will have a lower minimum pulse duration, agreeing with the example shown in Fig. 2.1.

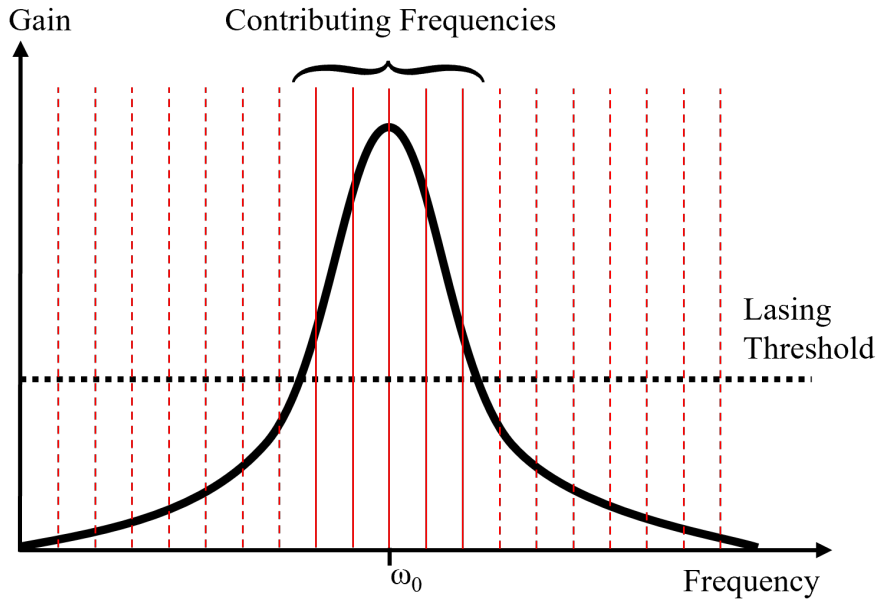


FIGURE 2.2: Demonstrating how individual frequency modes with intensities high enough to induce lasing can contribute to pulse production. Those where the gain is above the lasing threshold will contribute to lasing while those outside this region will not.

$$\tau_{min} = \frac{TBP}{\nu} \tag{2.4}$$

In order to ensure that the constant phase relationship can be achieved and maintained a technique called mode-locking is used. Mode-locking can either be caused by an active or passive method, depending on the type of laser and the pulse durations involved. Active mode-locking is generally used with pulses of picosecond or longer durations and involves modulating the resonator loss in order to control the leaking of radiation. The modulators often take the form of an electro-optic or acusto-optic modulator, which works by adjusting the optical losses of the modulator at a prescribed frequency, such that any pulse that arrives while the losses are low will be favoured. On each pass, each part of the pulse that does not arrive at the minima loss position will be attenuated, further acting to shorten the pulse. This effect has to work in opposition to effects which act to lengthen the pulse, such as chromatic dispersion, and can produce pulses with length in the order of picoseconds.

In order to produce pulses of sub picosecond durations, an alternative form of mode-locking, passive mode-locking, must be used. Passive mode-locking can either be achieved with a saturable absorber or via making use of the Kerr lensing effect. In both cases, the aim is to achieve a variable loss characteristic within the modulator dependent on the incoming optical intensity, decreasing with higher intensities. This produces the same effect as manually adjusting the loss of the modulator, but can be achieved much more quickly, and without the requirement for precise controls.

Passive mode-locking acts on a survival of the fittest basis, where those pulses that have the highest phase coherency, and therefore the highest intensity, are promoted while those that are out of phases are not. This effect self propagates and is an automated process, decreasing the complexity of the system as no timing electronics or precise controls are needed.

When using a saturable absorber, its parameters are chosen so that it can balance with the gain medium used. This is done by choosing a saturable absorber with a loss normally greater than the energy increase through the gain medium. This means that in normal operation, any light simply passing through both would eventually be dissipated. However, when a spontaneously generated pulse reaches the saturable absorber the higher intensity of the pulse reduces the loss of the absorber, allowing the pulse to receive a net gain during the cavity round trip from the gain medium. This configuration has the added benefit of any unwanted radiation being eliminated due to the loss from the saturable absorber exceeding the increase from the gain medium. While the process of changing dissipation characteristics is fast, it is not instantaneous. This reaction time also acts to shorten the pulse duration as the leading edge is slowed more, counteracting some of the effects of dispersion from the gain medium. Along with the time taken for the loss to reduce, there will also be a relaxation time where the loss increases again after a period of high intensity.

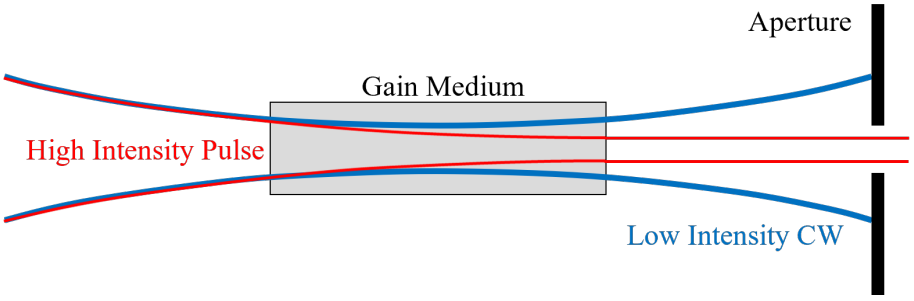


FIGURE 2.3: Kerr lens mode-locking using the Kerr effect from the gain medium to focus thehigh-intensityy radiation and an aperture to clip and remove background energy.

Another method of passive mode-locking is achieved with the use of the Kerr effect as described in Appendix B, and an example setup can be seen in Fig. 2.3. When the incident beam has a Gaussian profile it will tend to self-focus, loss reduction being greatest when the light itself has a large intensity, at the centre, when passing through the gain medium. This means that the main pulse of interest will become more focused than any background radiation. This can then be combined with a physical aperture that cuts off more of the less focused, low-intensity light than the more focused pulse. This leads to a similar effect as found with the use of a saturable absorber and promotes the amplification of the pulse. One of the major differences between using a saturable absorber and a system that takes advantage of the Kerr effect is the response time of the focusing material. The Kerr effect occurs on a much

shorter time frame than the response from a saturable absorber. While this does allow for the potential of a much shorter pulse, the inability to slow the front of the wave means that Kerr lens mode-locking lasers are often not self-starting. Instead, these lasers require a physical perturbation, such as nudging a mirror, to be added manually before the mode-locking process will begin.

2.3.2 Chirped Pulse Amplification

One of the key features of femtosecond lasers is the high peak powers they can produce. While this is useful to their ability to remove material on a target sample, the high intensities can also cause issues within the laser. The gain material was often a major limiting factor in the system when it comes to the peak power of the laser as it would become damaged at these very high intensities. To avoid this, a way to reduce the peak power through the gain material, while not affecting the overall peak power available to the system, is required. In ultra-short pulse lasers, this is achieved by varying the duration of the pulse through various steps. The duration of the pulse can be increased when passing through the gain material, decreasing the chance of damage, and then finally reduce the duration again once the desired power has been achieved. An example of this can be seen in Fig. 2.4 where the pulse is stretched before passing through the gain medium before being compressed again afterwards.

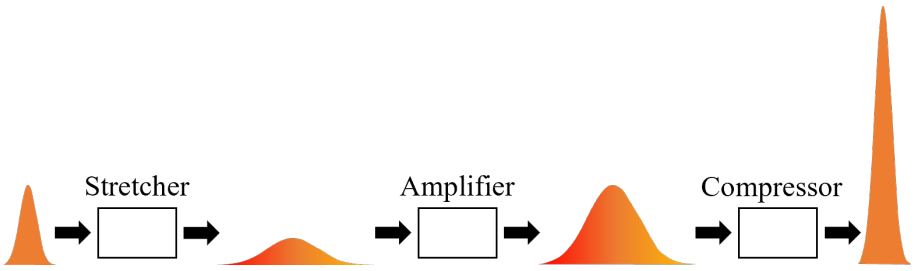


FIGURE 2.4: Pulse stretching and amplification. The laser pulse is stretched by temporally separating the frequencies before passing through the amplifier. Once the pulse has been amplified sufficiently the frequencies are recombined and the pulse is compressed.

In lasers that operate in the femtosecond regime, a common technique to achieve this shortening and lengthening is to use chirped pulse amplification (CPA). A chirped pulse is one where the frequency of the wave varies throughout the pulse temporally and can be used interchangeably with the term sweep signal. A chirped pulse can either be up-chirped, where the frequency increases throughout the pulse, or down-chirped, where the frequency decreases over the pulse in time. The effect of pulse chirping is similar to that of group velocity dispersion found in dispersive media such as glass with a wavelength-dependent refractive index. In this effect, the longer wavelengths will tend to be slowed more than the shorter wavelength parts of the pulse. The idea of chirped amplification was first proposed to solve a very

different problem, overcoming the power limitations of radar systems Strickland and Mourou (1985) and was recognised with the award of the 2018 Nobel prize for physics.

CPA is a method of stretching out the pulse, while still allowing it to be compressed with little loss. The two major components of CPA are the stretcher and compressor which allow the chirping to take place. As discussed before, creating a pulsed laser is dependent on the existence of different wavelengths in the pulse. This also holds for CPA, where there must be different wavelengths present to allow for parts of the wave to be slowed by differing amounts. In a grating-based system, this is achieved via the use of diffraction to angularly disperse the various wavelengths within the pulse. Once dispersed, each wavelength will travel along a unique path, each with a different path length. By delaying components of the pulse with different wavelengths by differing amounts, it is possible to stretch the overall duration of the pulse envelope.

$$2d \ \sin\theta = n\lambda \tag{2.5}$$

The stretcher and especially the compressor are often formed using diffractive gratings, rather than dispersive media, due to the same concerns that affect the gain material such as damage from high intensities and beam stretching. When chirping, the beam will pass through several passes of stretching and amplifying, before finally going through similar passes in the compressor. By utilising a method of multiple passes through the compressor, fine control can be achieved over the final duration of the pulse, leaving all other characteristics unchanged as the amplification process has already been undertaken, although a pulse that is not fully compressed will exhibit a temporal frequency sweep. As the total intensity must be kept low to avoid damage to the gain medium, dispersive media can be used within the stretcher. According to Bragg's law (Eq. 2.5), when electromagnetic (EM) waves are incident upon a grating it is the wavelength that determines the angle through which the beam is deflected.

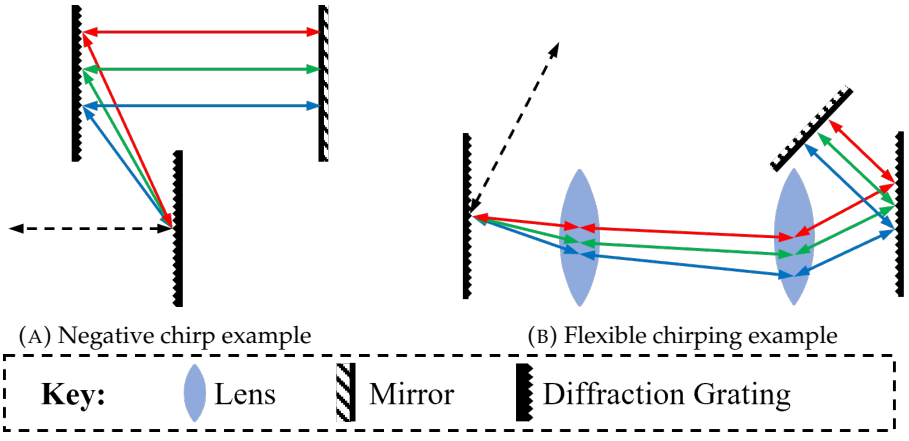


FIGURE 2.5: Chirping device examples.

One typical design for a chirping device is a pair of gratings set to pass the beam path in a Z shape with a mirror to retro-reflect the beam and return it to a collimated state as shown in Fig. 2.5a. This effect can also be achieved with a single grating and additional mirrors, reflecting the beam of the single grating 4 times. This is harder to set up though and loses a very simple method of tuning the amount of chirping. This is accomplished by changing the separation between the gratings. As this process can only cause negative chirping it is almost always used as the compressor, requiring another setup to be used for the stretcher.

One example of a more complex setup is shown in Fig. 2.5b uses two gratings and a mirror at the end, where the angles of the gratings are mirrored about their centre. An additional two lenses are required, positioned between the gratings. This is a far more adaptive setup and can be used to either positively or negatively chirp the pulse; a distance (L) of less than the focal length of the lens (f) between the lenses and their respective gratings will give a positive chirp. If L is greater than f a negative chirp will be applied, if L is equal to f then no chirp is applied.

The main limitation of this setup, and why it is generally not used for compression, is the inclusion of the optical elements. These suffer the same issues as the gain medium and have the potential to be damaged by the high peak intensities found in the final pulse. Therefore, the two-grating setup is used for the compressor when these intensities are likely to be found. However, this is not an issue before the amplification of the pulse when it is stretched. This can only lower the peak intensity passing through the material, as can be seen in Fig. 2.4.

2.4 Modelling Pulsed Laser Machining

To predict the effects of laser ablation is a complex subject. In this section, a method called the two-temperature model will be explored. Although there is only one approach discussed here, there are a wealth of other methods which are beyond the scope of this report to cover. This section is purely intended to highlight some of the complexities involved with attempting to fully model the machining process that takes place. The focus here is on pulses on the order of 100s of femtoseconds as that is the range of pulse length used directly in the experiments discussed later in this thesis. For further information, the book '3D Laser Microfabrication' Misawa and Juodkazis (2010) should be consulted.

2.4.1 Two-Temperature Model

One of the most common techniques for simulating laser machining on these time scales is the Two-Temperature Model (TTM) which has been often used to describe the

effects of laser irradiation Zhang et al. (2015); Jiang and Tsai (2004); Fang et al. (2010); Dong et al. (2019). Similarly to previous discussions, in the TTM, the interaction between a laser pulse and the target material is understood to occur in the form of electron excitation via photon absorption. Heat is then transferred to the lattice through electron-phonon collisions. Due to the large difference between the electron and lattice vibration's momentum, the interactions between these two systems take place on the order of a few picoseconds. When a laser pulse is much shorter than the thermal relaxation time, the lattice is not able to reach thermal equilibrium with the electron gas and so the excited electrons will be at a much higher temperature than the lattice. This means that the electrons and the lattice can be considered as two connected systems, and these can be represented as a pair of coupled differential equations. Despite the two systems each being at different temperatures, each individual is assumed to have a well-defined local temperature and be at a state of dynamic equilibrium. The equations in this section represent a simplification of the TTM referred to as the parabolic TTM and are taken from Zhang et al. (2015).

$$C_e \cdot \frac{(\partial T_e)}{\partial t} = -\Delta \cdot (-k_e \Delta T_e) - G(T_e - T_l) + Q \tag{2.6}$$

$$C_l \cdot \frac{(\partial T_l)}{\partial t} = G(T_e - T_l) \tag{2.7}$$

These differential equations can be solved using the finite difference method, taking small steps in time. In the equations C represents the heat capacity, T is the temperature, k is the thermal conductivity, and G is the electron-phonon coupling. The subscript e and l relate to the electron and lattice respectively. While the electron-phonon coupling factor does depend on the electron temperature, it was shown by Fang et. al. Fang et al. (2010) that for metals such as aluminium it remained approximately constant at temperatures below 1000 K. The heat capacity of the electrons is also dependent on the temperature of the electron lattice, with an approximately linear relationship. The thermal conductivity of the electrons is instead dependent on both the lattice and electron temperatures, being proportional to that of the electrons, and inversely proportional to the temperature of the lattice. It can also be seen that the equivalent heat transfer term from the lattice heat equation has been ignored. This is due to the heat transfer in the lattice occurring on a timescale at least two orders of magnitude slower than that of the electrons in the model.

$$Q(x, y, z, t) = S(x, y, z) \cdot T(t) \tag{2.8}$$

$$S(x, y, z) = \frac{1 - R}{\delta + \delta_b} \cdot F \cdot \frac{w_0^2}{w^2(z)} \times \exp\left(\frac{z - z_s}{\delta + \delta_b} - \frac{2(x - x_0)^2 + 2(y - y_0)^2}{w^2(z)}\right) \tag{2.9}$$

$$T(t) = \frac{1}{t_p}\sqrt{\frac{4\ln 2}{\pi}}\exp\left(-4\ln 2\left(\frac{t-2t_p}{t_p}\right)\right) \tag{2.10}$$

The properties S and T represent the spatial power distribution of the laser and the temporal evolution of the pulse. In the equations, R is the reflectance of the material, δ and δ_b represent the optical and ballistic skin depth, w and w_0 represent the beam diameter and beam waist, and t_p is the pulse duration. In this example, the pulse is assumed to be a focused pulse with a Gaussian profile both spatially and temporally. Material is assumed to be removed when the lattice temperature reaches 90% of the critical temperature as described in reference Zhang et al. (2015). When applied over multiple pulses, the relative times that the pulses are interacting with the sample are very short. To resolve this it is often optimal to implement a variable time step that has a high resolution for the duration of the pulse. When the laser intensity is removed the Q term is zero and the calculations become simpler. The advantage of maintaining calculation during this time gap is that it allows for the sample to retain heat build-up in the lattice that can influence the results of future machining.

$$w(z) = w_0\sqrt{1+\frac{z^2}{Z_r^2}} \tag{2.11}$$

$$Z_r = n\pi\frac{w_0^2}{\lambda} \tag{2.12}$$

One major limitation of the implementation of the two-temperature model shown here is that the material removed with each pulse is not taken into consideration. Improvements have been made over some previous methods by taking into account the focus of the laser. This means that various focal positions could be considered as well as ensuring that machining does not continue for an arbitrary distance. Despite this, there is still no calculation of the impact of the pulse not being absorbed by the material it is calculated to have passed through. This means, that while a single pulse may be accurate, subsequent pulses will become less accurate due to the additional predicted attenuation from material that is no longer present. There are methods to avoid this and an example of one such method will be discussed in the next section. Even with this taken into account, there are still many factors that are hard or almost impossible to include in whichever model is used. Calculations of these methods are highly dependent on a good knowledge of the material being machined and are often designed for static systems that assume the sample is perfectly normal to the beam, although some works have tried to solve this Dasallas and Garcia (2018).

Beyond simple environmental challenges, many assumptions are still needed, and system peculiarities are not taken into account such as non-uniformity within the

beam and aberrations caused by interactions with optical elements. Further to this, there are other methods that can be used to simulate the material removal and improve the accuracy of the model including ray tracing Otto et al. (2012) and deformable meshes Dong et al. (2019).

2.5 Conclusions

The generation and underlying theory behind the generation of short and ultra-short pulses is a complex topic that has seen much research. Despite this, it has been a very beneficial area of research due to the potential lasers operating in these regimes offer. The peak powers found in these lasers mean that areas of machining are possible that would either be unfeasible with CW lasers, or would cause extreme surplus damage to the sample in question. Even though nanosecond laser machining produces less clean results than found with a femtosecond laser, they still offer more precise machining than many other potions.

The modelling of how these pulses interact with a material is a very complex subject and goes far beyond what was presented here. Despite this, there was already a high level of complexity present while still containing many assumptions and simplifications.

Additionally, the techniques explored in this chapter were entirely focused on the femtosecond regime and are not very applicable to others without major modification. While the basis of the TTM is still sound, the model as presented takes advantage of the very clean machining that such short pulses achieve, with low thermal overspill and most machining taking place via a single mechanism. Moving up to pulses with durations on the order of nanoseconds would require many more factors to be included in the calculations, such as the heat-affected zones, melting, and changing conditions during the pulse itself. Despite that this simple model does not fully capture the intricacies of femtosecond machining, with phenomena such as laser-induced periodic structures not explored. While this has been far from a comprehensive look into the field of computational modelling of laser machining, it serves to provide a reference point into the relative complexities of the modelling processes to be compared to the alternative techniques presented throughout the rest of the thesis.

Chapter 3

Machine Learning

Machine learning is an area of computing where the traditional approach of coding a solution is eschewed in favour of algorithms that are designed to self-optimise. This allows for very complex problems to be solved without the need for a designed solution that will only work in that specific situation. Machine learning is a very broad term that covers a lot of different techniques, each one applicable in different situations and lots of research has gone into both making the techniques more powerful and more efficient.

The field of machine learning has grown exponentially over the past few years, with many factors owing to its growth and success. Three of the most important aspects to the success of machine learning are the techniques used, the computational power available, and the amount of data that can be used to train the networks. As machine learning is such a broad topic, a full investigation of the field is beyond the scope of this thesis and more information can be found in deep learning by Goodfellow et al. Goodfellow et al. (2016). Rather than trying to cover the entire field, in this chapter, all types of machine learning encountered within the thesis will be described. There will be a large focus on neural networks to match their use throughout the experimental chapters of the thesis. Alongside descriptions of the techniques themselves, there will also be discussions on the importance of the data used within machine learning as well as the computational requirements and restrictions involved.

3.1 Data

As machine learning is a fundamentally data-driven exercise, the data used is of vital importance, in the quality, quantity, and how it is applied. When training a machine learning process, the data should be separated into at least 2 sets, the training set and the validation set. The training set is the set used while the parameters are updated

during the training phase. This set can indicate the performance of the model, and indeed it is what is used to update the model. However, an improvement on the training set does not necessarily mean an improvement in the performance of the model as it can lead to a phenomenon called memorisation or over-fitting. This describes when the model can give very good results on the data it is trained with, but increasingly poor results on unseen data. To get around this issue, the second set of data should be held back for periodic assessment of the network, and this is the validation set. This dataset should not include any data from the training set, to assess the model's ability to generalise. A third set of data, the test data, can be held back to assess the performance of the final network by the developers. When a stakeholder is involved they may hold back an additional set of data that is unseen by the development team. This allows them to test the performance of the network with data that the network cannot have been biased towards.

3.1.1 Data Quantity

One of the simultaneously simplest and most difficult methods for increasing the performance of a model, and reducing the chance of over-fitting, is to increase the amount of data used in training. For a given number of training steps using the entire dataset, the length of training will be approximately proportional to the amount of data used, increasing the length of training, although, with more data, fewer steps may be required. While it is generally the case that more data leads to better performance, this is not always the case, and the time required, and ability to collect the data may play a large role in how much data is used. The impact of the size of the dataset is also dependent on the task at hand, and the number of trainable parameters. As the computational difficulty or data entanglement increases, the amount of data required will also tend to increase

One well-known dataset often used in teaching the use of neural networks is the Iris Flower dataset Dheeru and Graff (2017), which became a standardised test point for a machine learning methodology referred to as support vector machines. This dataset represents a classification problem with 4 input variables and 3 different types of flowers to assign a sample to. As there are only a few inputs and outputs the dataset can be quite small, using only 150 data points, while still allowing high performance from networks. Machine learning techniques have come a long way, but examples of more difficult tasks include the classification of images from the MNIST Deng (2012) and CIFAR-10 Krizhevsky (2009) datasets. The classification tasks performed here not only include a greater number of outputs but the number of inputs is increased greatly since the images themselves are used as the basis to classify from. These two datasets contain 60,000 and 50,000 training examples each, although, with modern techniques,

extremely high levels of accuracy can be achieved, with state-of-the-art networks attaining an error of 0.17% Zhao et al. (2019) on the MNIST dataset.

3.1.2 Data Quality

While there may be many machine learning techniques that would be able to complete the tasks set before them, none can do so without data. Even more than this, every network will only ever be as good as the data used to train it. One of the major issues with machine learning techniques is that they are largely black box techniques, especially as the size and complexity of the networks increase. Being a black box technique means that there is very little ability to understand how the network is reaching the results it does, and is therefore an empirical rather than theoretical process. This means that a poorly designed experiment and/or dataset can have a large impact on the validity of the results, and there can be underlying issues that cannot be gleaned from just looking at the results.

One famous case of misinterpretation of the results from a neural network comes from a paper designed to predict if a person was a criminal or not. This network was based on assessing their facial features Hashemi and Hall (2020) and the validity of the results is highly questionable. A detailed investigation into this case was produced by Bowyer et al. Bowyer et al. (2020) where they discussed the effects of the data on the results. In essence, the data for images of 'criminals' were taken from a dataset of mugshots, while non-criminal images were taken from a variety of other sources. Other than the major difference in mugshots or not, there were several other discrepancies in the images, including the format (PNG vs JPEG), greyscale conversion process (natural vs converted) and original medium (print vs digital).

When trying to create a dataset for machine learning, there is a fine balance to achieve between including everything you would like to train the network on and removing human bias that can lead to artificially positive results. This is true for the training data and especially the validation & testing data, as that is largely what is used to assess the performance of the network, especially in generative networks. Throughout the experiments described within this thesis, careful consideration has been given to how best to meet both the condition of a full data spread, as well as trying to remove the human bias. Even in cases where there are handcrafted examples in the validation dataset, other examples also exist that are generated with the same ideals as the training set.

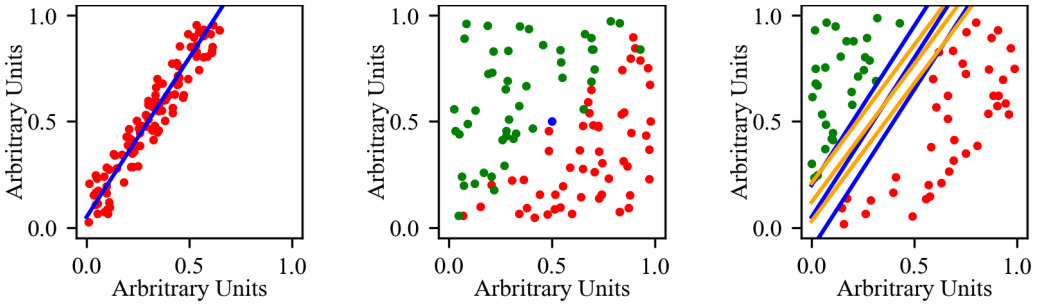
3.1.3 Data Relationships

There are two types of connections that can be used to describe sets in machine learning, the mapping between them, and the combinations, or pairings used for training networks.

In a general sense, a machine learning task can be thought of as a process to convert data from one domain to another. In a mathematical sense, a domain is the complete set of possible values that could replace a variable, for example, in machine learning, it can refer to all possible inputs into the network. Relationships between domains for each dataset come in different forms including one-to-one, one-to-many, and many-to-one. In the example of neural networks, there will usually be two datasets in different domains, one domain containing the input data from the network, and the other containing the output data. In a one-to-one mapping, each input data item has a corresponding output data item. An example of this is the multiplication of two prime numbers. Each pair of primes will produce a single number, and there is no other way to reach that number with any two other integer pairs that do not include one and itself. Taking the product of any two numbers that include a non-prime number is an example of a many-to-one mapping. Each pair of numbers will produce exactly one answer, but multiple such pairs will produce that same answer. The one-to-many case is the same as the many-to-one case but reversed, such as trying to find factor pairs of a number.

Each of the cases described above offers differing levels of complexity, both in the calculation and the verification, and each case will be unique with not even each direction in a one-to-one being identical. There are also issues in the form of the answer required, whether you want a single response or a range including as many answers as possible. This fact is one of the key failings of generative networks called mode collapse, where only very small variations on a single result are produced, even from very different initial random seeds.

The other type of relationship between datasets is in how the data is paired. Pairing does not explicitly imply exactly 2 sets of data, but rather how the inputs to the machine learning process are mapped to the outputs. When data is paired for training and validation, each input into the network has a corresponding output that it can be compared to. This leads to the biggest difference between paired and unpaired training being the losses used to quantify the effectiveness of the network. When data is explicitly paired, the network can be classified as conditional, and great use can be made out of comparative losses such as the Mean Absolute Error (MAE) or Mean Squared Error (MSE). This is a very useful foundation to build on as there is a very clear distinction between what the networks get correct and incorrect, leading to strong derivatives with which to update the parameters of the network.



(A) Linear Regression (B) k Nearest Neighbour (C) State Vector Machine

FIGURE 3.1: Various machine learning techniques. The Linear Regression method is being used to fit a line of best fit to data, the k Nearest Neighbour is being used to assign a single item to a category, and the State Vector Machine is being used to define the separation between two categories.

Unpaired data can come in a few different forms the first being where there the dataset only contains data representing desired outputs from the network. Despite the description of transforming data from one domain to another, this does not mean that all data is explicitly collected. In many cases, part or all of the input to the network will consist of a latent space, where the network itself learns how it should map outputs to this input space. A famous example of this is StyleGAN Karras et al. (2019) where faces were generated utilising latent space. This latent space could then be explored, with features in the outputs being mapped to the latent space. By doing this the final results from the network could be manipulated to show desired characteristics. The use of latent space also allows the possibility of complex transformations based on vector arithmetic. An example of this is taking the latent space that would produce a man with glasses, subtracting a latent space for a man, and adding that for a woman. The final result from this transformation will provide a woman wearing glasses Radford et al. (2016). Some examples Chen et al. (2016) take this even further with tricks being used to convert the latent space into a more usable set of parameters for control over features.

3.2 Machine learning approaches

Machine learning is a broad term that refers to a subsection of artificial intelligence. Artificial intelligence generally refers to the concept of computers acting in a way perceived to be smart or intelligent. Machine Learning, however, is the use of techniques for computers to complete a task without explicit programming of a solution. ML encompasses several techniques such as k-nearest neighbour support vector machines, linear regression, and neural networks.

3.2.1 Regression

One of the simplest Machine Learning (ML) techniques is called regression, where the task is for the computer to fit a line to a dataset to be able to predict missing results, either within the current limits or to extrapolate beyond. The simplest form of regression is a linear regression with one independent and one dependent variable. In this situation the algorithm tries to fit a line with the equation $y = mx + c$ to the data, shown in Fig. 3.1a, calculating a loss after each iteration to measure the success. To calculate this loss the distance of the line from each point is taken by some method, and the total loss is the combination of all distances. This is a very simple process requiring little computational power, even when increasing the complexity to fit lines with more parameters and more variables.

3.2.2 K-Nearest Neighbour

The k-Nearest Neighbour (kNN) technique was a method of machine learning and was first proposed in 1951 Fix and Hodges (1951) and is often used in classification. The premise of the kNN algorithm is that each data point is represented by a vector, with similar data points being contained within an area of the vector space. An initial set of labelled data is required to be able to classify future objects. When a new object is to be classified, the distance from its position in the vector space to all other, labelled, data points must be calculated. The distances are ordered and the highest classification representation of the closest k points is used to determine the classification of the new object. With large datasets and high dimensional space, kNNs become very computationally expensive due to the necessity to calculate the distance to each point for every classification. In the example shown in Fig. 3.1b, the value of k chosen was three when trying to classify the blue point. Out of the three closest points, one is green and two are red, hence the point is classified as red. This method can also be used with regression when there is a high correlation between the mapping variables and a desired unknown variable. Here the value for the new data point can be determined via an average of the same k points discussed earlier.

3.2.3 Support Vector Machines

Support Vector Machines (SVMs) represent a leap in complexity and combined the idea of both kNN and linear regression first proposed in 1995 Cortes and Vapnik (1995). In SVMs data is mapped to a high dimensional feature space and the hyper-plane that optimally separates two distributions is sought. This process allows for a very effective classification method for data from one of two possible classes. When more than two classes are involved multiple SVM networks are required and

the result is given by a combination of the results. Fig. 3.1c shows a two-class dataset where two boundaries have been fitted to the data. The minimum distance between the central blue line and the two datasets is larger than that for the central orange line and therefore would be the preferred fitting line. Support vector regression (SVR) works in a very similar way to SVMs where data is mapped to a higher dimensional space to be able to fit a straight line. The support vectors in this case represent a margin of acceptable error where loss is low, and the aim is to fit the line such that the most possible points are within this margin. An additional tolerance parameter can be included to account for outliers and make the model less prone to over-fitting.

3.2.4 Neural Networks

Neural networks (NNs) are an area of machine learning inspired by biological neural networks. One of the biggest developments in neural networks was the application of backpropagation algorithms in the 1970s Werbos (1975). Backpropagation is the method used to find the gradients of the overall loss function using automatic differentiation. To use backpropagation both a loss function and an optimiser are required. The loss function is used to measure the success of the network predictions and is very dependent on the both structure of the data, and the nature of the prediction. When performing a regression task, losses such as the mean absolute error (L1 loss) or mean squared error (L2 loss) are used. These two losses are shown in Eq. 3.1 and Eq. 3.2 with y representing the true result and $\hat{y}$ representing the predicted result from the neural network.

$$L_1 = \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3.1}$$

$$L_2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3.2}$$

Often L2 loss is preferable due to its stability compared to L1 loss, however, L2 loss will be more affected by outliers due to the error being squared. In contrast, for classification tasks, categorical losses such as categorical cross-entropy are used. Categorical cross-entropy requires that the true values take the form of a one-hot encoded vector, where every value is 0 except one that is 1, representing the category as shown in Eq. 3.3.

$$\begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

The categorical cross-entropy loss is then defined as:

$$L_{CE} = \sum_{i=1}^{n} -y_i \log(f_{sm}(\hat{y})_i) \tag{3.4}$$

where $f(\hat{y})_i$ is the softmax activation function:

$$f_{sm}(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \tag{3.5}$$

With the addition of the one-hot encoding on the true values, all values of y_i that are not the correct category will be 0, leading to a reduction of the sum in Eq. 3.4. When more than one label might be used, in an image with two animals, for example, a different approach is used. Rather than a softmax activation function, the sigmoid activation function (Eq. 3.6) is used instead. This is combined with a variation on the cross-entropy loss, where the loss for each category is calculated individually, called binary cross-entropy (BCE), shown in Eq. 3.7.

$$f_{sigmoid}(s)_i = \frac{1}{1 + e^{-s_i}} \tag{3.6}$$

$$L_{BCE} = \sum_{i=1}^{n} -(y_i \log(f_{sigmoid}(\hat{y})_i + (1 - y_i) \log(f_{sigmoid}(1 - \hat{y})_i) \tag{3.7}$$

Using BCE means that the prediction is compared to the true vector for each of the categories. If the category exists in the true vector, $y_i = 1$, then the first term is used, the reverse being true for categories which are not present, $y_i = 0$. The BCE loss is also used when there are only two possible classes.

Hinge Loss

$$L_{hinge} = \sum_{i=1}^{n} 1 - y_i \cdot f_{\tanh}(\hat{y})_i, \;\; y \in \{-1, 1\} \tag{3.8}$$

$$f_{\tanh}(s)_i = \frac{e^{s_i} - e^{-s_i}}{e^{s_i} + e^{-s_i}} \tag{3.9}$$

This means that when the output from the network shares a sign with the true value, the loss will be small, to a minimum of 0 when they have the same value. The advantage of a hinge loss is that it is very computationally simple and is traditionally used in SVMs, although it sees use in other applications.

After the loss is calculated the process of backpropagation is then used by the optimiser chosen, such as stochastic gradient descent, to update the calculation

parameters used in the network. The loss function itself is used to compare the outputs from the network to the desired result and this is used by the optimiser.

3.3 Neural Network Deep Dive

One of the areas of machine learning that has seen a lot of growth in recent years is that of neural networks. These networks are very adaptable and can be found in many forms depending on the task to be solved. This section will look into the history of machine learning, how the techniques have been developed over time, and later at some of the modern advances maximising the performance available from them.

3.3.1 Artificial Neural Networks

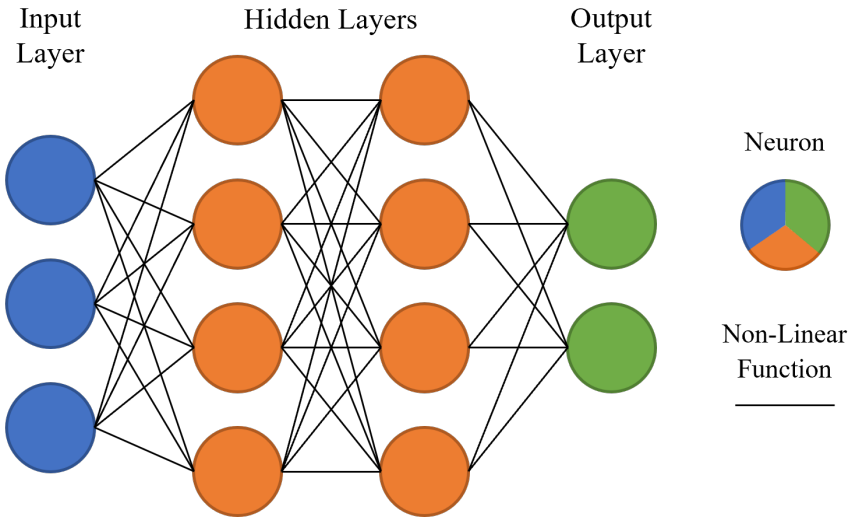


FIGURE 3.2: The Structure of an Artificial Neural Network. The network consists of a series of inter-connected nodes arranged in layers, with an input, output, and some number of intermediary layers.

An artificial neural network (ANN) is formed of a series of calculation nodes, called neurons, grouped into layers. The initial layer is formed of the raw input to the network, the final layer is the output, and the intermediary layers are called the hidden layers. The inputs and outputs are vectors, with the neurons in each hidden acting as a non-linear function acting on all outputs from the previous layer, referred to as a fully connected network, as shown in Fig Fig. 3.2. In a fully connected ANN with 1D layers, each layer is referred to as a Dense layer, where each neuron is defined by a set of weightings and a bias. Each layer is therefore made up of an $n \times m$ matrix of weights and n biases, where m and n are the number of neurons in the previous and current layers respectively. Within the neurons, a weighted (w_i) summation is

performed across all inputs (x) before the final addition of the bias (b_i), as shown in Eq. 3.10 where i denotes the neuron of interest.

$$s_i = x \cdot w_i + b_i \tag{3.10}$$

Each layer will also often contain a non-linear activation function that is used on all outputs of the layer. Non-linear activation functions are used to ensure that the entire network itself isn't just a large combination of linear functions. Commonly used activation functions include Rectified Linear Unit (ReLU) (Eq. 3.11) and sigmoid (Eq. 3.6), with ReLU being used in hidden layers and sigmoid for binary outputs.

$$f_{ReLU}(s)_i = \begin{cases} s_i, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

3.3.2 Convolutional Neural Networks

One area where traditional artificial neural networks are less effective is when working with images, as a lot of information can simultaneously be location independent, while also caring about what is around it. It is often not the individual pixels themselves that are important, but how they relate to the ones closely around them, often with less importance the further away you travel from the original pixel. The pixel's overall position in the image can also play an important part in the desired outcome, although this is not always the case, it will depend on the situation in question. As a solution to this problem, a convolutional network is used, consisting of one or more convolutional layers in combination with other types.

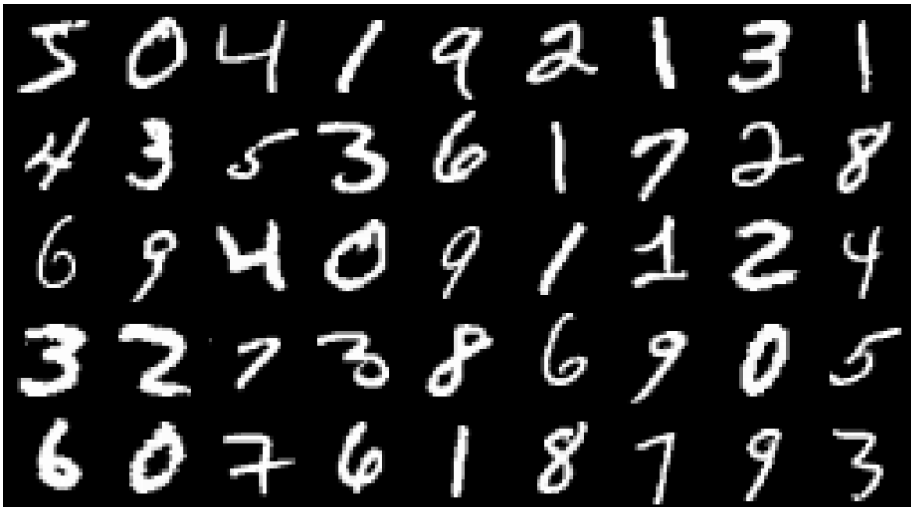


FIGURE 3.3: Example hand-drawn digits that form the MNIST dataset.

Working with images generally requires large amounts of data, with the complexity growing with increasing image size and channel numbers. With this in mind, a lot of networks and ideas are tested on one or more standard datasets including the MNIST dataset, CIFAR-10 and CIFAR-100, and the Celeb HQ dataset. To demonstrate the uses of CNNs, the MNIST dataset will be used. This dataset consists of 10,000 grey scale 28×28 pixel images of handwritten digits, labelled with the digit represented within, examples of which can be seen in Fig. 3.3.

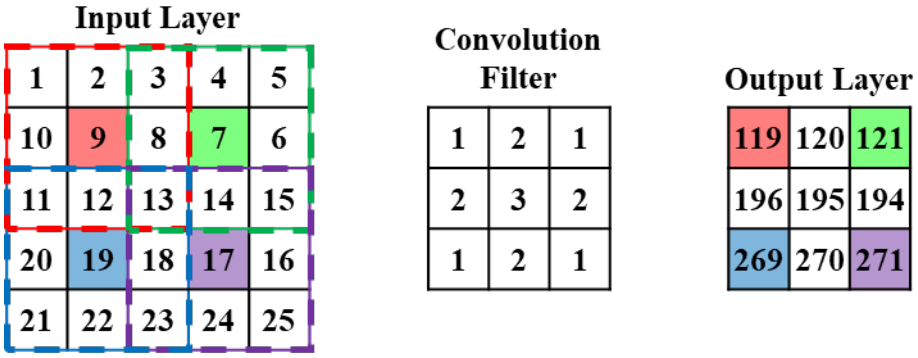
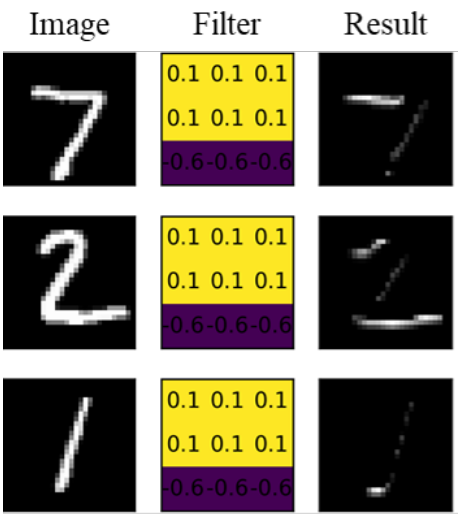


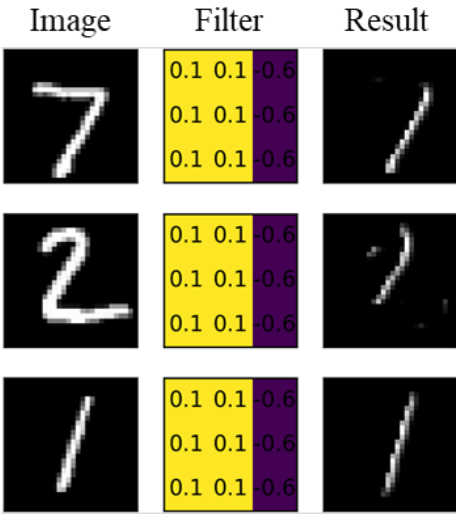
FIGURE 3.4: Example output from a Convolution layer with a kernel size of 3. Each coloured square in the right grid corresponds to the cell and area marked out by the same colour in the left grid. The marked area is multiplied elementwise with the cells in the convolution filter and then summed to give the final value.

In a convolutional neural network (CNN), the input is often not a vector but rather a 2D or 3D array, interpreted as a tensor. A simple example of this situation is a colour image which would be a stack of three 2D arrays representing the RGB channels of the image. Although a CNN is different to an ANN, they both still follow the same principle of input, hidden layers, then output. The biggest difference is the inclusion of the convolutional layers that act directly on the tensors. Rather than being a single function at each node, a CNN uses a kernel, a small 2D array that scans across the entire image, as seen in Fig. 3.4. In this way, features can be identified no matter where they are in the image and pixel proximity is also taken into account rather than treating each pixel independently.

Within each convolutional layer, there will be several kernels that are all sensitive to different things, allowing for many features to be examined. As an example, a kernel may find vertical or horizontal features, with kernels designed for both being sown in Fig. 3.5. In Fig. 3.5a the kernel is mostly negative on the right-hand side and symmetrical vertically. That means that if all pixels have similar values, the overall convolved value will also be negative. If, on the other hand, there is a decreasing gradient to the right the convolved value will tend toward positive, depending on the gradient. This can then be combined with a ReLU activation function (Eq. 3.11) to have pixel values of 0 except on the right-hand edge of features. The same is then true for Fig. 3.5b where descending gradients in the vertical direction mean that the lower edges of the features were captured.



(A) Kernel for capturing horizontal features



(B) Kernel for capturing vertical features

FIGURE 3.5: The effects of kernels designed for different purposes on MNIST images. The kernels that capture horizontal details show uniformity along the rows and disparity in the columns, the opposite being true for those that capture verticaL details. The horizontal and vertical kernels capture the lower and right edges of the images respectively.

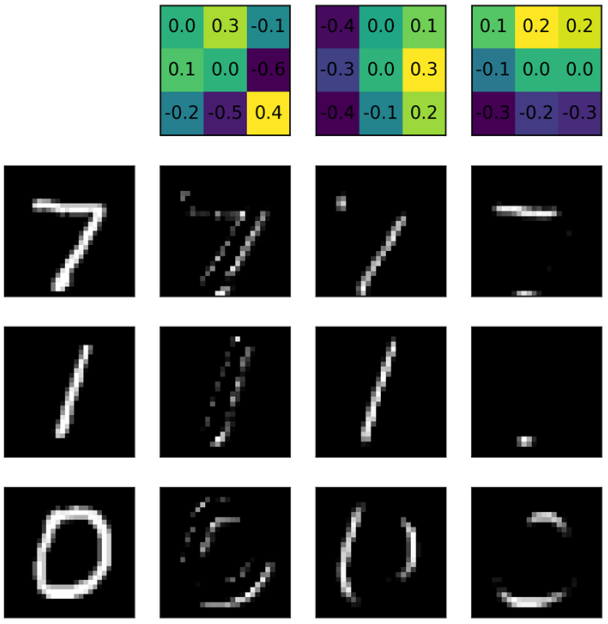


FIGURE 3.6: Examples of convolutional kernels in a trained network and the information they pick up from some example images.

The power of neural networks is that the user does not have to define the features that should be isolated, but rather the training process will favour those kernels that identify useful features for the specified task. As can be seen in Fig. 3.6 kernels that identify horizontal and vertical features have indeed developed throughout training. Along with this other kernels such as those which find the outside edge of objects in the image can develop. Typically, CNNs will have many kernels within each

Convolution layer and may contain many Convolution layers. Taking a very simplistic look using the kernels described above, if, after the convolutions, the image only contains vertical features and no horizontal features it would suggest that the image contains a 'one'. This is a very simple example with only three kernels, but in a full network containing a far greater number of layers and kernels many more complex features could be identified.

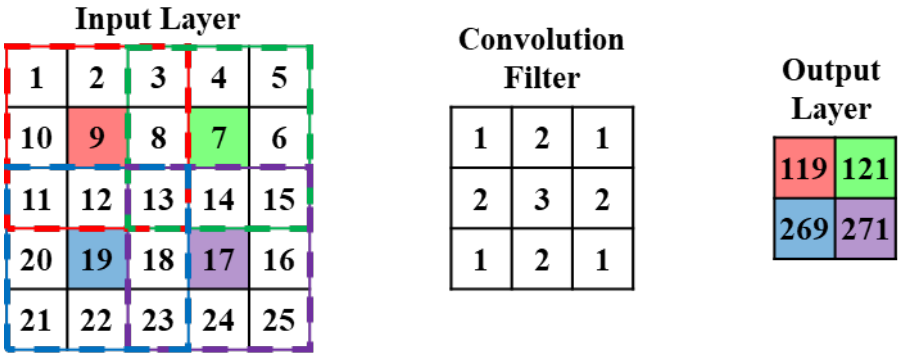


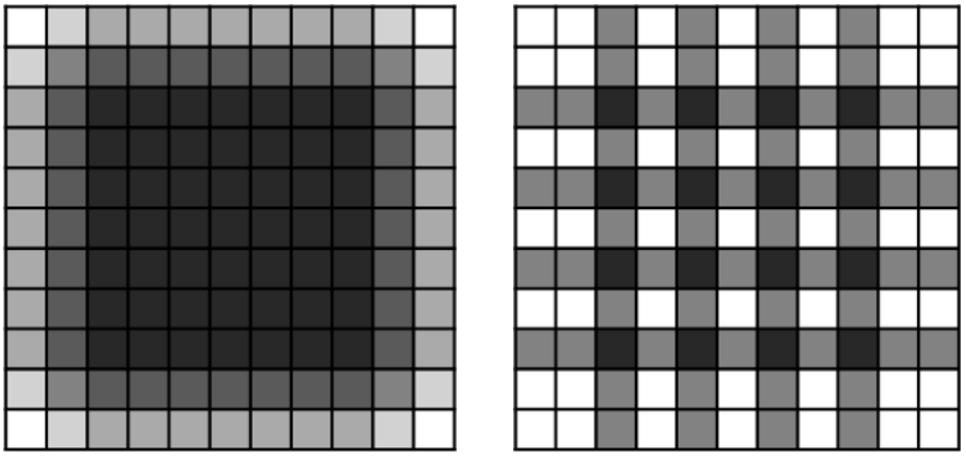
FIGURE 3.7: Example output from a Convolution layer with a kernel size of 3 and a stride of 2. Only the highlighted cels in the input layer are captured by strided convolution.

In addition to the use of multiple kernels, scaling is used throughout the networks to look at features of different sizes. There are two common ways of performing the scaling operations, the first of which is the use of strided convolutions. In a strided convolution not every position in the input is evaluated, but instead every n^{th} position is evaluated as shown in Fig. 3.7.

$$o = \left\lfloor \frac{i - k + p}{s} \right\rfloor + 1 \tag{3.12}$$

Using strided convolutions leads to a reduction in the size of the input layer as shown calculated in Eq. 3.12. The final output size o depends on the input size i, the padding p, the kernel size k, and the stride of the convolutions s. This process has the advantage of reducing the total number of steps due to the resizing and convolution happening in a single step.

One disadvantage of strided convolutions is that certain pixels can appear more often in the calculations than others which can lead to the network giving that pixel higher importance than others. When using a stride of one, all pixels that are at least the kernel size away from the edge of the layer will have equal importance transferred onto the next layer. In strided convolutions, the values of both the stride and the kernel size play a major part in the final effect. If the kernel size is not divisible by the stride size, there will be a periodicity introduced into the pixel influence, becoming multiplied across a 2D grid. This effect can be seen in Fig. 3.8, where Fig. 3.8a shows the effective influence of each pixel when using a stride of 1 and a kernel size of 3.



(A) Stride = 1 (B) Stride = 2

FIGURE 3.8: The effect of convolution stride on pixel weighting. The number of times each cell is used in a calculation is represented by the darkness of the cell, with darker cells having been used more often. Using a stride of one gives an almost uniform distribution with the outer cells seen less often but the inner cells being consistent. When using a stride of 2, there is a distinct grid pattern where central cells are only seen once and corner cells are seen most.

While those pixels at the edge have a low influence, those towards the centre have equal importance. This can be combined with padding which can both improve this, by having the padded pixels be the least influential, and has the added benefit of maintaining the 2D size of the layers. In contrast to this, in Fig. 3.8b it can be seen that when a stride of 2 is combined with a kernel size of 3, there is a highly periodic relation to the influence of each pixel. This can lead to some areas having far more influence on the final result than would normally be expected.

A second common method to downscale is the use of pooling, where a value is calculated from a subset of the pixels. Two of the most common pooling methods are max and average pooling. In max pooling the maximum value of each $n \times n$ region is evaluated, where $\frac{1}{n}$ is the scaling factor of the pooling layer. While this process does have a low computational cost, it does introduce the issue of some pixels having far greater influence than others. While the weights of the Convolution layers can counteract this, it is not always the case. In average pooling, the mean of each $n \times n$ region is taken with a stride of n. This process can be thought of as a convolution with a kernel size n, a stride of n, a bias of 0 and fixed weights of $\frac{1}{n^2}$ at each position. The advantage of this is that it is quick compared to a convolution in training as no updates to the layer are required.

3.4 Generative Networks

Along with the classification and object detection tasks CNNs have found great application in the space of generative networks. Often these networks aim to produce 2D image outputs based on either an image-based or vector-based input. Examples of tasks that generative networks have been designed for include image super resolution Watson (2020); Ledig et al. (2017), audio and voice generation Huang et al. (2022); Gao et al. (2018); Liu et al. (2020); Shahriar (2022), and even medical applications such as protein mapping and drug discovery Cheng et al. (2021); Bian and Xie (2021); Strokach and Kim (2022); Repecka et al. (2021); Anand and Huang (2018).

Generative networks are often formed from an initial latent space which passes through Dense layers before being reshaped into a 3D tensor. Once in this form, convolutions can be used alongside various up-scaling methods until the layers have the correct size.

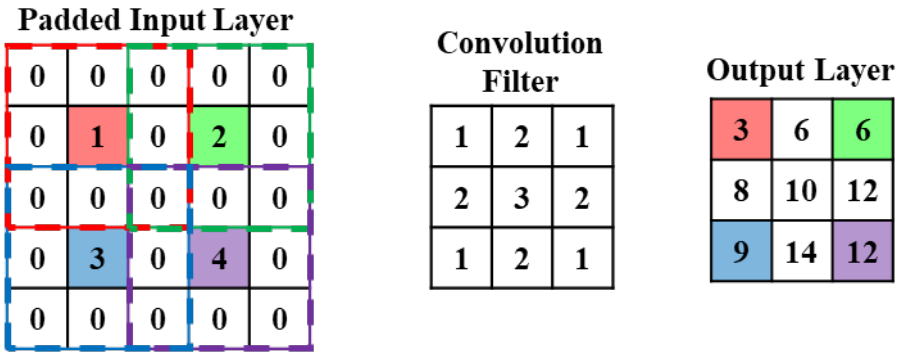


FIGURE 3.9: Example output from a transposed Convolution layer with a kernel size of 3 and a stride of 2. The original 2x2 input is padded with 0 values cels placed around and between all of the cells.

Similarly to down-scaling, up-scaling can be included in the convolutions via transposed convolutions. To perform a transposed convolution, the input is first padded with zeros such that the original data is separated by the stride of the convolution. The convolution itself applies with a stride of one and this padding then means that there are more convolution steps, and thus increases the size of the output.

$$o = s \cdot (i - 1) + k + p \tag{3.13}$$

When using a transposed convolution, the final size of the layer is given by Eq. 3.13 where i and o represent the input and output dimensions, k is the kernel size, and p is the padding.

While transposed convolutions can be used, many networks have opted to use alternative methods, often found in traditional image manipulation. Networks that

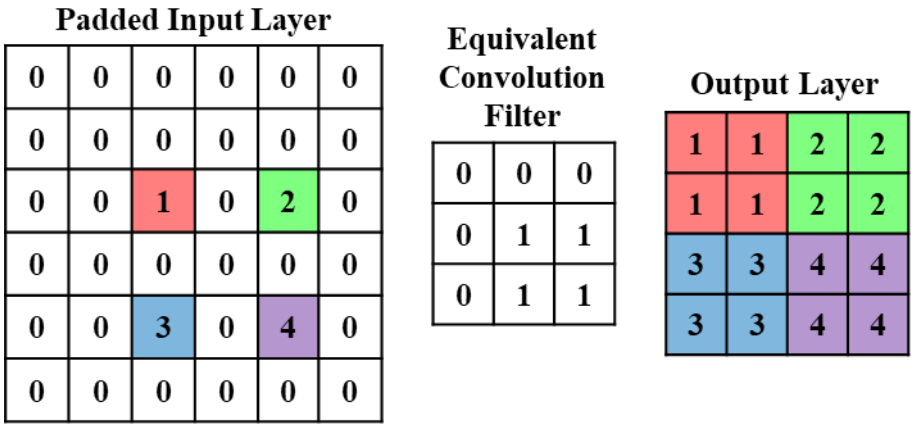


FIGURE 3.10: Example output from a nearest neighbour upscaling layer. The example shown here is the equivalent convolutional filter that can produce the same result.

are designed around increasing resolutions in steps of scaling factor 2 may opt to use methods such as a nearest neighbour or linear interpolation. Similarly to how average pooling can be thought of as a strided convolution, so too can various up-scaling methods. An example of how a nearest neighbour up-scaling method with a factor of 2 can be formulated as shown in Fig. 3.10 where the given kernel can be used as the basis for a transposed convolution with padding 1 and stride of 2. Again, this method is quicker than a traditional transposed convolution for up-scaling as no weights or biases need to be updated. Despite this, the transposed convolution can play a role in the calculation of the generated output and so additional Convolution layers may be needed to replace those lost.

In a very simplistic sense, a generative network can be thought of as performing a regression for each pixel in the output image, and so similar losses can be used, with L1 (Eq. 3.1) and L2 (Eq. 3.2) losses being common. One difference to many regression tasks is a fixed output space where each pixel will have a fixed range, often between 0 and 1, representing the range of 0-255 used to represent RGB images. This fixed range means that two of the most common activation functions for the final layer of the network are the sigmoid and tanh functions, with the former producing numbers in the range of 0-1 naturally.

3.4.1 Generative Adversarial Networks

In 2014, Dr. Ian Goodfellow released a paper on a technique called generative adversarial networks (GANs) Goodfellow et al. (2014) as a proposed improvement on the autoencoder structure. GANs are an effective method for transforming the style of one image into another (e.g. turning photos into labelled photos), or even simply randomly generating a style of image (e.g. images of human faces). GANs are composed of two different networks, where one is the generator and the other is the

discriminator. The generator often takes the form of a CNN, following a very similar idea to that of autoencoders. The discriminator bears more similarity to a convolutional classification network with a single output in the range of 0-1.

The relationship between the two networks can be thought of as similar to that between a counterfeiter (the generator) trying to create fake currency, and the police (the discriminator) trying to detect the fakes. Both start not knowing anything about what real or fake currency is but are trained simultaneously. The counterfeiter aims to make fake currency that will fool the police while the police want to be able to detect fakes with 100 % certainty. The end goal of the training process is to achieve a Nash equilibrium where neither network can improve its ability to beat the other.

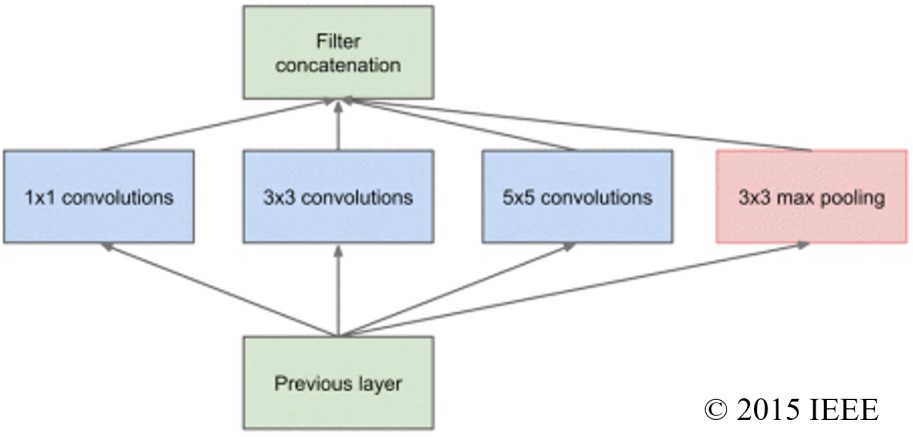
GANs have shown promise in the field of image generation and a lot of work has been done with them since, further improving their quality. This ranges from both supervised to unsupervised learning. Supervised learning refers to situations where there are direct translation pairs and so the desired result of the GAN is known exactly; these are often known as conditional GANs. Unsupervised learning, however, uses unlabelled data to translate from one domain to another. An example of this is transforming pictures of horses into zebras Zhu et al. (2017). This is useful for tasks such as stylising images where exact pairs do not exist. While this is powerful, networks trained on data pairs will often perform better and so should be used where possible.

3.4.2 Reference Networks

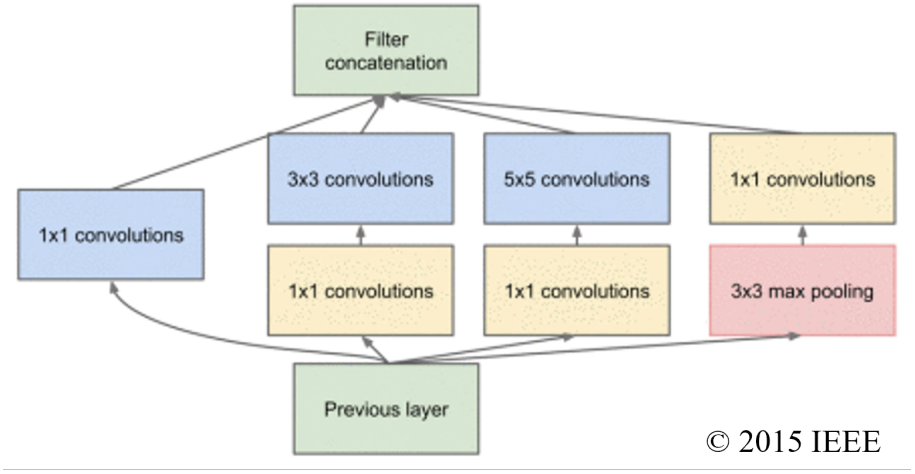
With neural networks and GANs being such an active field of research, many companies and research teams are working to continuously optimise their performance and design novel architectures. These are often large teams working with powerful hardware to be able to fully test and iterate through designs. Due to this high volume of development, the decision was taken to utilise and adjust existing architectures in order to maximise the time that could be spent on the experimental process, while allowing rapid adaption to the field and continued use of cutting-edge techniques.

3.4.3 Inception

First competing in 2014 with GoogLeNet Szegedy et al. (2015), Google has competed in the ImageNet Large-Scale Visual Recognition Challenge Russakovsky et al. (2015) many times, winning in 2014, and coming second in 2015. After the first entry, Google went on to develop a series of similar networks, all taking some form of the Inception nomina Szegedy et al. (2015); Ioffe and Szegedy (2015); Szegedy et al. (2016, 2017).



(A) Inception module



(B) Inception module with dimensionality reduction

FIGURE 3.11: The structure of the first iteration of Inception blocks from Szegedy et al. (2015).

One of the key features of the inception series was a creative exploration of to better use of multiple convolutions to reduce the sizes of the networks used, while not restricting the network to use a single size of convolution kernel. Initially, with GoogLeNet (Inception-V1) they proposed the use of multiple scales of convolutions, using 1x1, 3x3, 5x5 convolutions, and a 3x3 Max Pooling layer at each level as can be seen in Fig. 3.11. This allowed the network to determine the ideal size of the kernel at each point in the process and apply a higher weighting to it. This process did provide two complications, it greatly increased the size of the network, as well as increasing the computational costs. To avoid this each of the 3x3, and 5x5 Convolution layers were preceded by an additional 1x1 Convolution layer, with the pooling layer followed by one. These 1x1 convolutions were used to reduce the dimensionality of the network with cheap operations, while still being able to be trained. This idea was extended even further in later versions of the inception architecture, with each of the larger (NxN where N>1) convolutions being broken into sequential Nx1 and 1xN convolutions. While this did add to the number of operations, it was more efficient for

all sizes, with the smaller 3x3 convolutions being ∼2.6x as computationally expensive as the 3x1 convolutions.

Potentially even more than experimenting with interesting Convolution layers, the biggest impact from the inception series was found in BN-Inception-V2 Ioffe and Szegedy (2015). In this network, Ioffe et al. proposed the use of batch normalisation to avoid the vanishing gradient problem in deep neural networks. As networks became deeper and deeper, there were possibilities that the magnitudes of deviation within later layers decreased over time, especially when using saturating non-linear functions. A consequence of this is that not only will it self-propagate, but it will also lead to a reduction in the gradients fed through the network. In an attempt to find a solution to this, Ioffe et al. introduced normalisation into the training process.

By this point, almost all networks were utilising the idea of mini-batches, where the network computes gradients on several samples simultaneously. Using mini-batches has benefits in both smoothing out training due to gradients being averaged, and allowing quicker computation due to parallelisation of the operations. The concept of Batch Normalisation proposed to adjust the distribution of layers prior to the non-linear functions in order to avoid saturation. Batch normalisation takes the form of two operations, firstly adjusting the mini-batch such that it has a mean of 0 and a standard deviation of 1 before extending that with a weight and bias learned per layer, referred to as gamma and beta. While this was the first major implementation of normalisation, the techniques introduced have been used throughout neural networks since. Beyond batch normalisation, several other normalisation techniques have been used, including instance and layer normalisation, each one suited to different use cases.

3.4.4 U-Net

Beyond the conceptualisation of the GAN, one of the biggest developments in the space of image generation was the U-net structure Ronneberger et al. (2015) shown in Fig. 3.12 that can be used for image-to-image translation. In this network, the generator used a down and up scaling path to first reduce the input image down to a size of 28x28 pixels, before gradually increasing the resolution up to the desired 388x388 pixels. This style of network was commonly used in GANs and autoencoders designed for image translation as it allows for filters of constant size to see different portions of the image. At large sizes, each filter only sees a very small portion of the image and so picks out fine details, such as strands of hair. Once the image has passed through the network to the constriction point, each filter looks at a larger portion of the image and so will pick out broader features, such as areas of land and sky. This effect could be achieved by using large filters, but that would not ensure that the large features would be the focus (the filter may consist of mostly zeros) and would also be

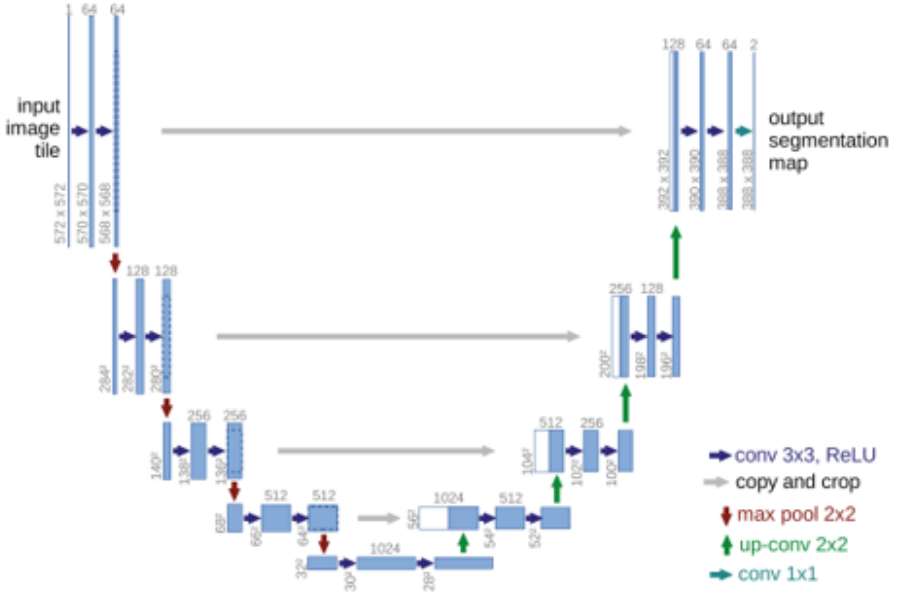


FIGURE 3.12: The structure of the U-Net network, reprinted from Ronneberger et al. (2015) Copyright (2015) by permission from Springer Nature.

far more computationally expensive. When these operations have been completed the image is again up-scaled to the final resolution.

The downscaling section had been traditionally required as only using the low-resolution scales would lead to a large loss of information that can be captured by the network at higher resolutions. Despite this, there is still information lost as the network scales, which is what the U-net is trying to solve. The main distinguishing feature of this structure is the skip connections that are used to tie layers together from the down and up-scaling paths which have the same resolution. These connections allow for the high-level information learned early in the network descaling path to be utilised by the later parts of the network. The network was originally designed for the segmentation of cell images, but the technique has seen wide use throughout machine learning, especially in other image-to-image translation applications.

3.4.5 Progressively Growing GAN

While GANs can produce visually accurate and sharp images, they generally have greater difficulty the higher the resolution that is used. This is generally true due to the discriminator having an easier job of differentiating real and generated images at higher resolutions Odena et al. (2017). This problem is exacerbated by generative tasks that are not directly similar image-to-image translation tasks due to initial results being wildly different from the ground truth. In 2018 Karras et al. Karras et al. (2018) proposed a solution to this issue by breaking the training into various stages. The idea behind the network was designed to be adaptable to many situations, with the key

component being an up-scaling decoder that started from a low resolution and would eventually output at the desired high resolution. The structure of the network preceding the first decoding layer is not specified and can be adjusted depending on the input type. This meant that the network could implement Dense layers when using a latent space or vector input, but could be adapted to use the U-Net structure for image-to-image translation purposeses.

In order to simulate the use of a reference image and allow the network to learn the large, coarse features, the output from the network was initially taken from a very small resolution layer, going down to as low as 4x4 pixels. Once this has been trained for a time, an additional set of layers can be added to upscale the network to 8x8 pixels. When this is added, none of the preceding parameters in the network are changed and it remains trainable if required by the network. This process is then repeated until the desired final resolution has been reached. When implemented correctly, training a progressively growing GAN, especially for situations where there is no initial reference image, can be quicker than a traditional GAN due to many training steps being conducted without the expensive larger layers.

3.4.6 ResNet

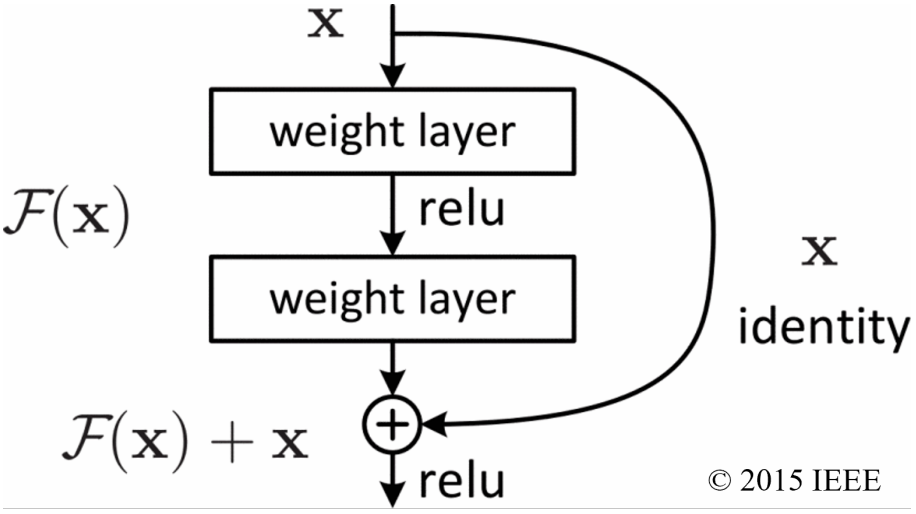


FIGURE 3.13: The structure of the ResBlock, the fundamental building block of a ResNet, from He et al. (2016).

Since the early days of neural networks, one of the most common techniques used to improve the accuracy of the networks was to increase their size. This increased size leads to an increased capability for the networks to infer from the data and pick out import information at all different scales. Despite this, deep networks are not without issue, and in early 2015 it was often found that errors would saturate and then worsen, similarly to over-fitting, but would also occur on the training data. The network called ResNet He et al. (2016) was developed in an attempt to maintain the

possible benefits of very deep networks, while also working within the limitations of current frameworks to reduce this degradation. The proposed solution was to introduce shortcut, or skip, connections between certain layers in the network, as shown in Fig. 3.13.

A ResBlock, which is the key component of a ResNet, is a combination of multiple standard Convolution layers, with the skip connections tying the input to the first, to the output from the last. In the initial implementation, the skip connection was a 1x1 Convolution layer, used to match the dimensions of the final Convolution layer in the block. This allowed the skip connection to be summed with the output while also allowing for a change in the number of filters used throughout the network. While this is one implementation of the ResBlock, the base idea is incredibly powerful and has been utilised in a range of networks. Another way to combine the two paths is to use a concatenation, allowing for the pure input to be used throughout the network, although this can lead to a higher computational cost.

3.4.7 Pix2Pix

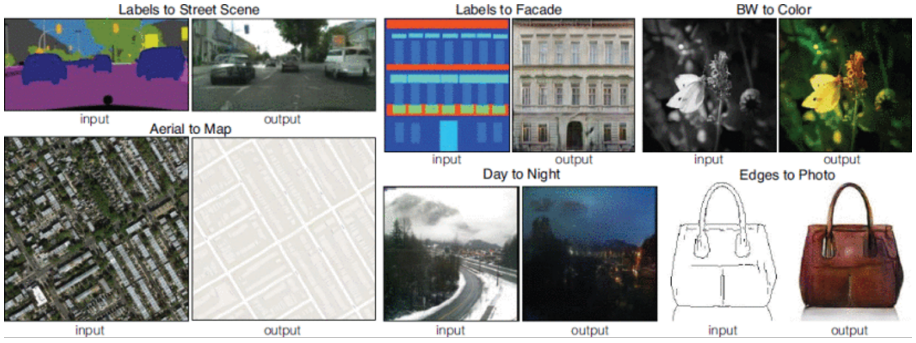


FIGURE 3.14: Examples produced by the Pix2Pix network from Isola et al. (2017).

While there have been many advances in GANs, in the field of image-to-image translation, the work by Isola et al. Isola et al. (2017) on the Pix2Pix network acts as a foundation block. Many networks are designed with a specific task in mind, highly tuned to that specific purpose. While this leads to some very powerful and accurate networks, they can be difficult to adapt to other uses, and so lots of work must be dedicated to this. Pix2Pix was created with the idea of finding out how to make a flexible network that could be used and adapted to many tasks, specifically in the field of image-to-image translation, with several examples of different applications shown in Fig. 3.14. The Pix2Pix network makes use of the U-Net structure (see Section 3.4.4) and combines a comparative loss with a PatchGAN discriminator. The PatchGAN discriminator is designed to assess several patches of an image to determine whether they are real or generated examples. It works by applying down-sampling layers to

the initial image until each of the pixels represents an area of the desired size on the initial image. This is then judged in the standard way to calculate the loss. The Pix2Pix network investigated (and allows the use of) PatchGAN discriminators of various sizes to determine what caused results to be good for different applications.

Beyond being a powerful and flexible network in its own right, the adaptability of the network means that has formed a basis for development since its presentation. The ability to use the network with many datasets also promoted its inclusion on the TensorFlow website, being used to induce novices to the world of image-to-image translation.

3.5 Deep learning and Computational Power

Deep learning has been a well-known term to describe aspects of ML since it was popularised by Hinton in 2006 Hinton et al. (2006), although had been around from as early as 1986 when used by Rina Dechter Dechter (1986) and is used to describe an array of methods. Despite this being the first use of the term, the techniques now described by that term had been in use long before that, not long after the first implementations of neural networks. In general deep learning is commonly defined by two axioms: the use of multiple non-linear layers, and the breaking down of information into progressively more abstract information Deng and Yu (2014). The use of deep learning is often a task that requires a lot of both computational power and data to produce the best results. Along with advances in the techniques used, the growing power and proliferation of Graphical Processing Units (GPUs), which are ideally suited to many similar operations on matrices, has greatly accelerated the growth of this area. Alongside traditional GPUs, manufacturers have been improving their capabilities with the inclusion of technology such as tensor cores, or even discreet Tensor Processing Units (TPUs) specifically designed to be efficient at the typical tensor calculations performed in deep learning. While machine learning techniques can be designed to run on a huge range of devices, from microcontrollers to supercomputers, generative and deep learning models are generally designed to work on higher-end devices, often requiring a minimum of a GPU. The work conducted in this thesis has used a setup on the low end of power for generative networks, utilising either an Nvidia Quadro P6000 with 24GB of RAM and a newer Nvidia Geforce RTX 3060Ti with only 6GB of ram, but utilising a newer architecture including tensor cores that allowed for further speed and memory optimisations.

3.6 Conclusion

Machine learning as a whole is a huge field that has a long history and is at the forefront of research across many different fields. Various approaches were presented here that have been used in the laser machining industry to provide a baseline of understanding of the topic.

Beyond the high-level overview, the importance of data and computational power was also discussed. The amount of computing resources being dedicated to machine learning tasks is growing. It was also shown how more and more hardware is being developed to assist with machine learning tasks, even at the consumer level in the case of Nvidia GPUs.

As well as an overview of the topic, greater attention was paid to the technique called neural networks. It was shown that while they may be designed for different purposes, and act on different types of data, how they work remains similar. This is especially true when moving to deep learning where convolutional and generative networks are built upon a small number of vital blocks that are then combined to produce very complex networks.

Chapter 4

Using Neural Networks to Optimise Laser Machining Parameters

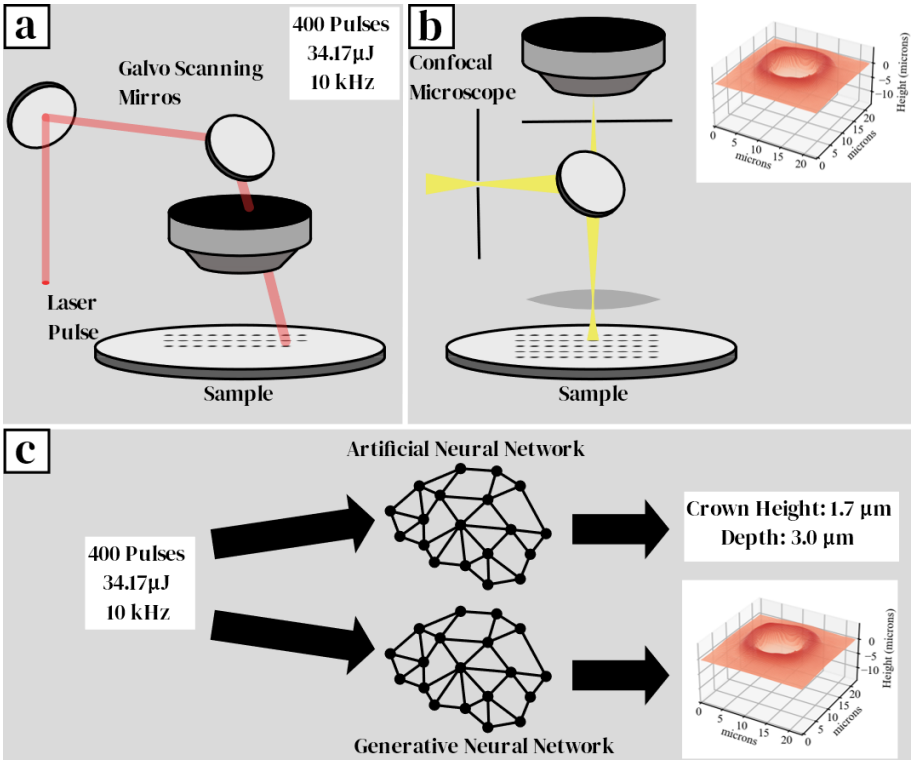


FIGURE 4.1: The experimental setup used to predict properties of machined dimples. First, a laser is directed with a pair of galvo scanning mirrors before being passed through a 100 m m lens to focus it at the surface of the iron sample. The sample is then measured using a confocal microscope to calculate a 3d height map. Neural networks were then used to model the machining process.

In this chapter, the application of neural networks for the identification of the optimal laser parameters for machining microscale blind holes (referred to here as dimples) is demonstrated. The structure of this experiment is shown in Fig. 4.1. where the dimples are first machined before the depth profiles were captured. This information

will then be used with a variety of machine-learning techniques to predict certain properties of these dimples. The performance criteria for this work were that each dimple crater must be deeper than 4 µm and that the raised rim around the outer edge of the dimples is minimised. The raised rim is caused by the redeposition of molten material and is referred to as the crown. Here, it is shown that machine learning can allow the discovery of the optimal laser parameters that enable the fabrication of anti-friction surfaces using nanosecond pulses. For this chapter, the experimental data was collected by technical staff at Oxford Lasers and results have been presented in McDonnell et al. (2021b).

4.1 Motivation

One of the factors that affect the friction found between two objects is the amount of surface-to-surface between them. Therefore, in applications that require contact of moving parts a way to reduce friction can decrease energy consumption, reduce wear, and prevent excess heat build-up. While common methods of friction reduction include coating or lubrication these are not always available. One alternative that has been used in a variety of industries is to machine an array of dimples onto the surface of a part to reduce the surface area of contact Qu et al. (2014); Mezzapesa et al. (2013); Scaraggi et al. (2014); Sakai et al. (2007).

These machining tasks are often performed using lasers with pulse durations on the order of femtoseconds. These lasers are chosen partly for the lack of heat damage caused to the part, maintaining its desired properties, such as strength. Another reason for choosing femtosecond laser machining is due to the clean results they can produce. Due to the lack of melting involved, they often display little displaced material that is common to other methods. This melted material can either form a ridge around the machined area or be deposited across the surface, both of which increase the roughness and so counteract the reduction in friction.

Despite this, many laser machining tasks that are currently carried out on femtosecond systems could, in principle, be achieved with nanosecond lasers. Nanosecond lasers do offer several potential advantages over using femtosecond based ones. The base cost per performance is one of the areas in which nanosecond laser can beat femtosecond alternatives. While it is not a direct comparison, in general, a nanosecond laser with the same pulse energy and repetition will be cheaper than an equivalent femtosecond laser. In addition to this, as discussed in Chapter 2, the longer pulses allow a higher proportion of the energy in the pulse to be deposited in the sample. This allows nanosecond lasers of an equal power to have far higher material removal rates than femtosecond lasers, although at the cost of reduced machining quality Neuenschwander et al. (2013); Ren et al. (2005). These two effects combine to

make nanosecond laser an appealing option when companies are constrained both by the cost of equipment and the time taken to mache a part, as higher throughput can lead to greatly increased profits.

Due to the complex relationships between laser machining parameters (pulse energy, repetition rate, number of pulses Cheng et al. (2009); Grant-Jacob et al. (2014); Lorbeer et al. (2017)) finding a combination of these parameters that will achieve the desired results can be a difficult process. While the throughput of nanosecond lasers is high, trying to attain a strong understanding of how all of these parameters and more interact would require a long investigation and is an area where dedicated machining companies excel. Despite this, there will always be occasions where a new process or material is introduced that behaves unexpectedly. In these situations, it may be imperative to gain a good understanding in a very short period of time, as business may depend on beating the competition to a solution, or a guarantee that one can be found. This approach also introduces the risk that human bias will influence the results in an attempt to reduce the time taken to find a solution and so vital areas of the data space may be missed.

4.2 Prior Art

When finding a solution to a machining task, the ideal solution is often the starting point and the optimisation process tries to reach that goal. The other method is to start at the solution and predict what can be done to reach it. This was the approach taken by Yousef et. al. Yousef et al. (2003) where a network was designed that would take the desired depth and diameter value, and return the pulse energy that would meet this requirement. A second network was also used that, using the result of the first, would then predict the stability of the result, an often desirable attribute in a commercial setting. This approach worked well as duplicate results for the combined depth and diameter were unlikely to be achieved by changing a single parameter. Other investigations into the use of ANN to predict the results of laser machining have also looked at using a single variable parameter Casalino et al. (2017); Campanelli et al. (2013), focusing on making a network as effective at these tasks as possible.

Alongside ANNs, there are many other machine learning techniques that can be used. Methods such as fuzzy expert systems and genetic algorithm-based approaches were compared to other modelling techniques by Parandoush and Hossain Parandoush and Hossain (2014). Here it was found that the machine learning options were able to predict the results of machining well and could complement other theoretical and analytical methods that have been employed traditionally. Another investigation into the effectiveness of different machine learning techniques was carried out by Kim et. al. Kim et al. (2018). It was found that, while not the most robust method, ANNs

provided some of the best results in the field. Another promising alternative was SVMs that approached the performance of the ANNS. While not directly in the field of laser machining the conclusions can still be used to direct the investigations conducted here.

Previous work Arnaldo et al. (2018) has shown that the nanosecond lasers can be used to achieve the results required in surface texturing and that improvements to the process could still be made. This will be combined with the findings of previous machine learning investigations to expand the capabilities in two directions, introducing the ability to simultaneously predict and explore multiple output parameters, and experimental re-creation with the use of generative networks.

4.3 Equipment Setup

While the experimental setup and design were performed by Oxford Lasers, details will be included here in order to provide a greater overview of the process. This experiment was based on the previous tribological work conducted by Oxford Lasers Arnaldo et al. (2018) and this influenced the choice of both laser and target material. The laser itself was a Ytterbium fibre laser with a central frequency of 1060 nm, in the near infra-red range, with an $M^2 < 1.8$. While the laser had a selectable pulse duration, the shortest value of 0.17 ns was chosen to best approximate the quality of ultra-short pulse machining. To machine the sample a 100 mm focal length lens was used to focus the beam down to a diameter of 36 µm measured using the D2 method Liu (1982) such that $I(w) = I_{max}/e^2$. The sample itself was a cylindrical section of grey cast iron DIN GG20 (ATSM A48 n.30) with 20% max ferrite phase, cut into 30° segments. In order to machine the full array of dimples, the samples were mounted on a 5-axis stage, which was combined with control over the optical axis via the use of galvanic scanning mirrors. Once machining was complete, a confocal microscope was used to measure the depth profile of all dimples.

4.4 Experimental Data

The goal of this experiment was twofold, being able to predict certain values from the experimental data, as well as recreating the experimental outputs. The data itself was paired data, with inputs of the laser machining parameters used, and outputs of a 2d array of values, representing the height at each position in microns. Each dimple consisted of 3 main features shown in Fig. 4.2: the surrounding (un-machined) surface, the raised crown of material caused by the redeposition and melting, and the crater itself. To calculate the values for the dimple depth and crown height, the 1^{st} and 99^{th} percentile of the data were taken respectively.

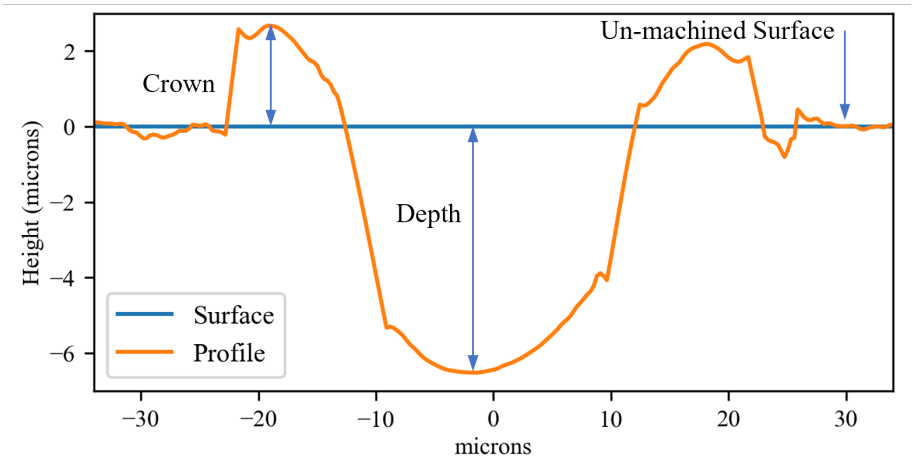


FIGURE 4.2: Cross-section through the centre of a machined dimple. The blue line at 0 represents the level of the material if it had not been machined. The peaks found near the edge of the view are the area of the raised crown, and the value taken is the 95th percentile of the data for each dimple. The depth was calculated using the 5th percentile of the same data.

Part of the process was for us to fully explore the possibilities of neural networks to perform the desired tasks, and as such, all of the data was preprocessed to ensure it was well suited to training a neural network. Initially, the raw data was provided in the form of arrays of 2d values, representing the height in microns, with a size of 301×301 pixels square. This data was captured in an automated fashion using a confocal microscope to measure the surface profile of the material. Within each of these arrays, the height data for a single dimple was contained, although the data was not centred. While this issue could be avoided, for the purpose of generating dimple profiles, it was desirable for them all to be centred in the final arrays to provide fewer parameters for the network to learn, and better utilise comparative losses.

Some data was discarded at this stage due to the dimples being too close to the edge of the provided arrays, in some cases, clipping the edge. These cases were discovered by analysing the values at the edges of the arrays and highlighting the ones where the values deviated significantly from 0. These arrays were then reviewed on a case-by-case basis to determine whether they would be within tolerance. Once all valid data had been selected, the arrays were then 0 padded to a size of 400×400 pixels. The images were centred by recursively cycling the images until the centroid of their absolute values was positioned at the centre of the new size array. The initial 0 padding allowed easy determination of the edge of the initial image, and therefore the maximum size that all images could be cropped to before including non-contiguous data. Once this maximum was calculated, the arrays were cropped to a size of 244×244 pixels, including none of the 0s used to pad the images in the final crops.

The base dataset itself consisted of 884 height profiles, machined using 170 combinations of laser parameters. Due to the high variation within the machined dimples, each combination was machined between 4 and 6 times to both capture some

	Minimum Value	Maximum Value
Pulse Energy (μJ)	5	50
Number of Pulses	50	400
Repetition Rate (kHZ)	10	1200

TABLE 4.1: The range of parameters used to machine dimples in the training and validation datasets.

variance, and provide duplicate data in the case that some were not suitable for use. This could be caused by several issues, including the aforementioned positioning of the dimple in the height data, along with machining defects such as gas pockets in the surface leading to very uneven machining.

Each of the sets of 4-6 samples were machined using a unique combination of the variable parameters. The three parameters that were adjusted were: repetition rate, laser pulse energy, and the number of pulses used to machine the dimples as shown in Table Table 4.1. The repetition rate of the laser was adjusted in the range of 10 to 1200 kHz, which was the maximum possible with the laser. Each dimple was machined with between 50 and 400 pulses, each with an energy of between 5 and 50 µJ

4.5 Network Architecture

A major focus for the investigation of this work was the determination of the efficiency and effectiveness of a variety of machine learning techniques, with a further exploration of the most promising options. The techniques used were: a Gaussian Process regression, a Support Vector Machine, an Artificial Neural Network, and a Generative Adversarial Network. Some advantages offered by the Gaussian Process and SVM are that they are very quick and easy to implement, being used with the python scikit learn library. Each of these methods was tested using different kernels to find which would perform the best.

To optimise the machining parameters, two different neural network architectures were investigated. The first of these was a simple fully connected ANN that took in the three input variables to output both the crown height and depth. This allowed for the optimisation of inputs based purely on the output conditions. The second method chosen was a GAN used to recreate the full experimental results. The GAN was used, due to its capability to transform numbers into arrays, such as laser parameters into a predicted surface.

The model used for the ANN was also relatively simple, consisting of a set of sequential layers as shown in Fig Fig. 4.3. After taking the input, three identical sets of layers were used, consisting of a dense layer, a dropout layer, and a batch normalisation layer. In each of these three blocks, the Dense layers were followed by a

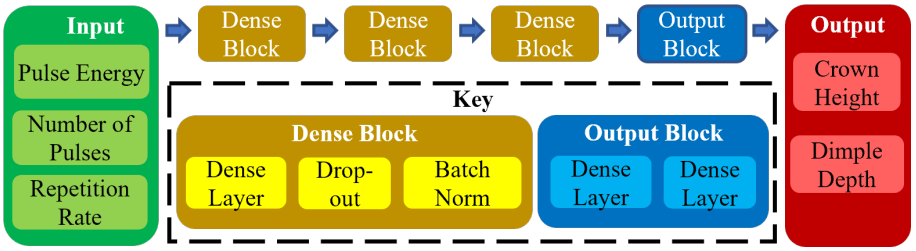


FIGURE 4.3: Block schematic of the Artificial Neural Network, from McDonnell et al. (2021a).

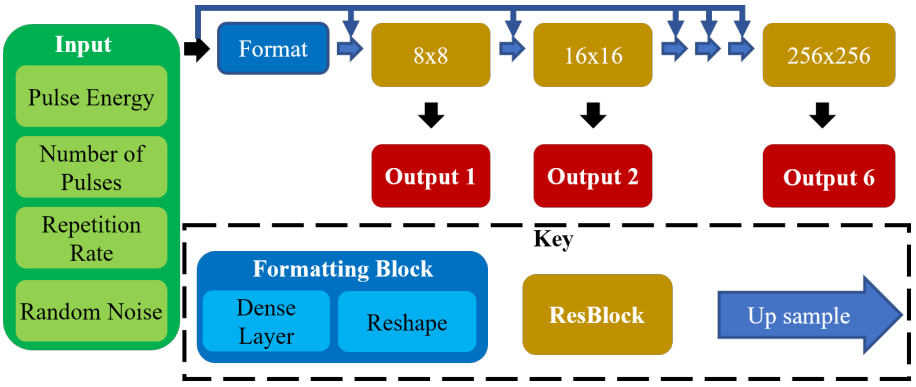


FIGURE 4.4: Block schematic of the Generative Adversarial Network generator, from McDonnell et al. (2021a).

ReLU activation layer. The dropout layer was used during training, where the value was set to 50% retention, being put to 100% retention during the interrogation of both training and validation data. Dropout is a technique used to reduce the risk of overfitting by not using the activation from all neurons during training, but a random subset of them. Following the final of the three blocks were two further dense layers, the first of which followed the same pattern as the ones used earlier in the network, with the same number of neurons and a ReLU activation. The final dense layer only had 2 neurons and no activation function, each one representing a different output, the dimple depth and height of the raised crown of material. The final layer was designed without a final activation layer to allow for it to be used for regression purposes. The network itself was trained using an Adam optimiser using a learning rate of 1e-5

In contrast to the previously discussed methods, the GAN used had a much more complicated structure. This owed to both the requirement for 2 separate networks and the designed task, profile generation as opposed to regression. The generator was based on a truncated encoder-decoder network, only utilising the up-scaling decoder path as there was no initial image to encode. As there was no initial image and the amount of data was low, it was decided to use an implementation of the Progressively Growing GAN (as discussed in Section 3.4.5). The input into the network was very similar to that for the ANN, but also included a noise vector representing latent space, allowing for variation in the final results as seen in the experimental data. Following the input was a formatting block that consisted of a Dense layer to interpret the input

Method	Variant	Error (%)
Base Data		13.0
Gaussian Process	Constant	62.2
	Rational Quadratic	**13.1**
	Exponential Sine Squared	25.7
	Dot Product	38.9
	Matérn	13.3
	Radial Basis Function	26.8
SVM	Linear	38.5
	Polynomial	19.5
	Radial Basis Function	**13.9**
ANN	8 Neurons	44.6
	16 Neurons	29.2
	32 Neurons	20.6
	64 Neurons	16.9
	128 Neurons	16.6
	256 Neurons	**12.8**
	512 Neurons	13.3
GAN		14.9

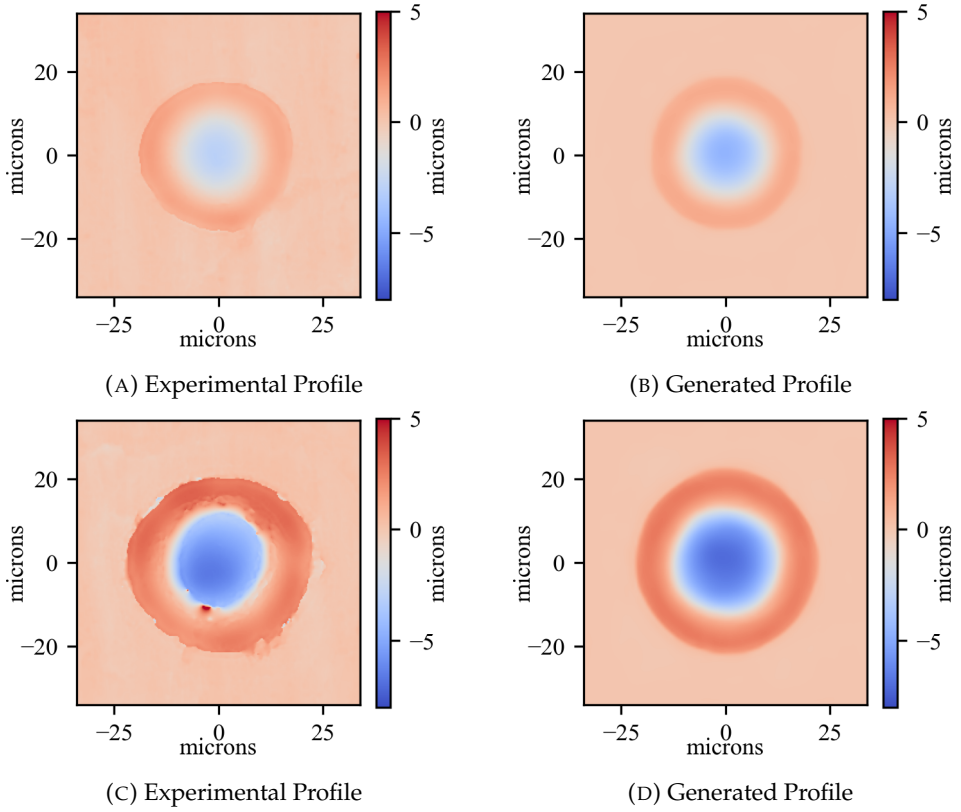
TABLE 4.2: Percentage error in predicted crown heights for various methods.

and a reshaping layer to allow that to be used in convolutional layers. After formatting, the bulk of the network was formed of a series of ResBlocks containing 2 convolutional layers with ReLU activation layers after each.

After each of the ResBlocks was a 1×1 convolution, that would act as the output and, except for the block at the desired output size, a bilinear up-scaling layer to increase the resolution. The network structure for each desired resolution was identical except for the number of ResBlock and up-sampling combinations. With the layers that are shared between desired resolutions, the shapes are identical, allowing for the weights from any to be loaded into another for continuous learning. While the outputs from lower resolutions were not used for training, they were maintained as their cost was very low and allowed the continual monitoring of network performance at various scales.

4.6 Analysing Performance

In order to determine the machine methodology to focus on, an initial test run was performed, using a number of different setups for each of the methodologies discussed in Section 4.5, the results being presented in Table 4.2. Since the initial publication of McDonnell et al. (2021b) further investigation was performed on a number of the cases, with optimisations made. For the Gaussian Process, an alpha value of 1×10^{-3} was found to produce stable results across all kernels. This was



(A) Experimental Profile

(B) Generated Profile

(C) Experimental Profile

(D) Generated Profile

FIGURE 4.5: Examples of experimental and GAN generated height maps of dimples. Figs. 4.5a and 4.5b represent dimples machined using 100 pulses at a pulse energy of 12.42 µJ and a repetition rate of 1200 kHz. Figs. 4.5c and 4.5d represent dimples machined using 100 pulses at a pulse energy of 38.67 µJ and a repetition rate of 600 kHz.

combined with an allowance of 10 iterations, after which further gains were not noticeable. The set of parameters was used with 5 different kernels, with the strongest kernels being the Rational Quadratic (RQ) kernel, and the newly tested Matérn kernel, reaching an error 0f 13.1% and 13.3% respectively, much better than the previously presented best of 32.0% found using the RQ kernel. Similar gains were seen when using the SVMs, with the best performing kernel being the Radial Basis function (RBF) at 13.9%, with the Polynomial kernel close behind at 19.5% using 2 degrees of freedom. Even with the improvements found in the intervening time, the highest performing ANN Still outperforms the Gaussian Process with RQ kernel and SVM with RBF kernel by a small margin without having had any intervening changes.

While it is designed for a different purpose, and not as optimised for this specific task as the previously discussed methods, the GAN could still be tested similarly. Rather than the numerical outputs of the former methods, the GAN provided a full 3d profile, from which the crown height and dimple depth could be calculated in the same manner as the original, experimental data. A visual comparison between experimental and GAN generated profiles is shown in Fig. 4.5. Both Figs. 4.5a and 4.5c are taken from the experimental data in the training dataset. Fig. 4.5a was machined using 100 pulses at a pulse energy of 12.42 µJ and a repetition rate of 1200 kHz while

Fig. 4.5c was machined using 100 pulses at a pulse energy of 38.67 µJ and a repetition rate of 600 kHz. The other two profiles (Figs. 4.5b and 4.5d) are GAN generated profiles, with Fig. 4.5b using the same laser properties as Fig. 4.5a and Fig. 4.5d using the laser parameter from Fig. 4.5c. While there are differences between the GAN and the experimental profile, there is a high correlation in all three of crown height, depth, and dimple diameter between the experimental data and the predictions. Some level of difference would be expected due to the variance within the experiment, even between experimental examples using the same parameter, as evidenced by the 13% deviation within the data. This demonstrates that while the network is designed for the generation of visual data, this can be used to calculate the same data as a numerical network with a similar degree of accuracy while providing far more information.

Despite the similarities, there are key differences that cannot be passed off as simple variations. The biggest factor that differentiates the GAN and experimental is the smoothness of the profiles. It is possible that adjustments to loss weighting, either fixed before training or variable throughout, could resolve some of this. The reason for this is that smoothness in the profiles, analogous to blurriness in images, is a common limitation in autoencoders that comes from the comparative loss, with the GAN discriminator model losses tend to lead to sharper images. While the comparative loss can greatly expedite training, especially in the early stages, at later stages of fine-tuning, loss weightings could be adjusted to put a greater focus on the GAN loss. Even with the visual discrepancies, the GAN was tested using by calculating the dimple depths and crown heights using the same method that was used to calculate the same values on the original experimental data. When tested on the same validation combinations used for the other methods, and calculating the crown height directly from the results of the GAN, the network had an error of 14.9%. While this is the worst out of the optimal setups for each method, it still beats many of the other setups used for all methods, including the ANN with reduced numbers of neurons per layer. This shows that for this exact task, the GAN is not the network of choice, although that is expected for regression problems. The GAN does, however, offer flexibility and visual output that could be very valuable, and is unique out of the methods tested.

As can be seen in Table Table 4.2 and Fig. 4.6 the size of the layers in the ANN was adjusted in an attempt to optimise the process. The layers being adjusted were the three Dense layers, with the number of neurons being consistent across the three within each individual test. The network was tested with layers containing only 8 neurons each, up to a maximum of 512 neurons per layer. As expected, when the number of neurons was low the performance of the network was also low, due to the limited capacity of the network. Additionally, with low numbers of neurons, the initialisation states of the layers play a big factor in the performance with the average percentage error over 10 runs being 39.7% with a standard deviation of 5.0%. Very

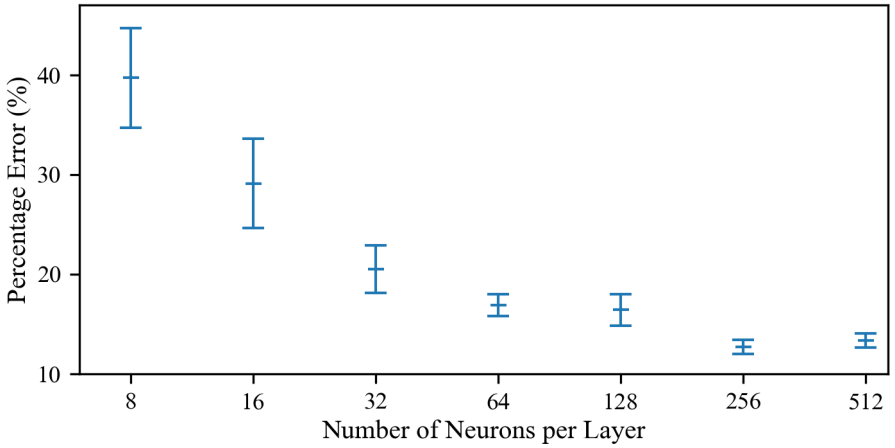


FIGURE 4.6: The effect of varying network complexity on the possible performance of the network shows that increasing the number of neurons in each layer of the network generally increased the performance of the network to calculate the hight of the raised crown of material around each dimple. Modified from McDonnell et al. (2021a).

quickly the performance of the ANN improves, with the average percentage error dropping to just 16.9% when the number of neurons per layer was increased to 64. At this stage, the initialisation of the network became less important, with a standard deviation of only 2.4%. Beyond this the gains are small, peaking at 12.7% when 256 neurons are used for each layer.

The general trend up to the 256 neuron point was for higher numbers of neurons to reduce the error. The only exception to this is the test with 128 neurons, which also had a much larger deviation than the surrounding data. The large variation indicates that the initialisation state can still provide a factor at these scales, with only 10 runs not sufficient to gain a full statistical description of the results. Having stated this, there is an interesting data point at 512 neurons per layer, which has both a higher error than that found at 256 neurons, at 13.3%, as well as having a slightly smaller deviation. While this does go against the previous trend and was still tested using a small statistical sample, interesting observations can be drawn from this. While increased network size can lead to increased performance, the higher number of trainable parameters meant that memorising issues could arise where the network 'remembers' the answers. This presented itself in the form of over-fitting, where the loss on the training data decreased over that of the network with 256 neurons, while the validation loss was higher. Alongside this, the training time for the network with 512 Neurons was higher than that of the others, further making it a less suitable candidate for further investigation than the other options of 256 neurons per layer. Due to its slightly better performance and similarity to other paths of investigation, it was decided that the ANN should be used to further study the impact the dataset and training parameters would have on network performance.

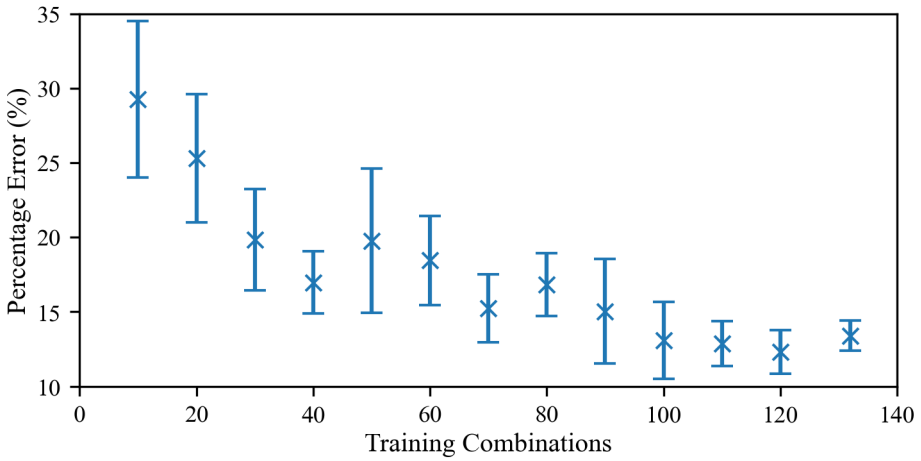


FIGURE 4.7: The effect of varying dataset size on the possible performance of the network shows that the performance of the trained network is loosely coupled to the amount of data used to train it. modified from McDonnell et al. (2021a).

4.7 Artificial Neural Network Investigations

Beyond just providing predictions of machining outputs, the machine learning techniques discussed can be used as a tool to solve a number of problems. This section will investigate how to optimise the data collection process as well as looking into how the network can be used to explore the experimental domain.

4.7.1 Optimising the Number of Training Combinations

While, ultimately, the accuracy of the network is likely to be high in the priorities of an end user, there are many other factors that come into play in the design, training, and use of neural networks. One major factor that makes machine learning techniques appealing is the time-saving measure that can be gained from reduced requirements for experimental processes. This effect can be amplified by reducing the amount of initial data needed to train the network in the first place. Many factors can play into this, with the main two for a single data collection run being the set-up time and per sample time. Each of these is calculated for each step in the processes, for example, laser machining and height map measurement. If the set-up time and network training time are low compared to the per sample time then the incentive is pushed towards being able to continuously update the network. In contrast, if the set-up time and especially training time is high then erring on the side of more data collection would be the ideal choice. One of the more complicated situations is where the training time is much lower and the data collection portion is time-consuming. In this situation knowing how much, and what, data to collect is vital to maximising the efficiency afforded by machine learning techniques.

In an effort to explore the ideal amount of data, the ANN was trained on various subsets of the entire training dataset. While this is not an overarching experiment and cannot be applied universally it can provide insight and a data point for future investigation. To do this, the network was tested using between 10 and 130 training combinations, randomly sampled from the full set, in increments of ten combinations, with results presented in Fig. 4.7. Additionally, the full dataset is included to provide a reference for the network performance at lower quantities of training data. In each case, 10 subsets of the full dataset were randomly chosen to train the network, with the error bars on the graph representing the standard deviation within these results.

While the results do show that with lower numbers of training combinations, there is a decrease in the accuracy of the network. Even with only 10 training combinations used the network was able to perform better than some alternative methods discussed previously, and even beat the same network when using a low number of neurons. Errors quickly drop to an average of below 20% with an initial training set of 30 combinations. This value further improves to 15% with just 70 training combinations, just over half of the total data collected. While this value is not quite as good as the full dataset, it is within 3% of the internal error of the data and may well be deemed suitable. To reach the full potential of the network it appeared that only 100 combinations were required, meaning that the data provided contained unnecessary data points, increasing the time to collect data and train the network.

Within all of these tests, as the data was randomly selected, the more combinations that were included, the more likely it was that the full range of the input domain would be covered. This means that these results could likely be improved to ensure that only interpolation was used rather than extrapolation. This does tie in with the requirements for a machine learning dataset, where a wide range of combinations should be collected. As discussed, while ultimate accuracy is usually desired, in this case, for example, it may have been beneficial to undertake investigatory multiple stages. Even with the huge advances in neural networks and machine learning capabilities, they cannot be guaranteed, and all results should be experimentally verified when it comes to novel combinations. To reduce the time taken for investigation, fewer initial samples could be taken, with a slightly higher error, followed by a more specialised investigation into areas of interest indicated by the network. For example, if it was known that the optimal combination used a low pulse energy then the parameter combinations chosen could focus on that area. This would improve the accuracy in the vicinity of focus but may lower the overall accuracy and may lead to better combinations being missed. This example shows how machine learning can be combined with traditional experimental processes to enhance the capabilities of both.

	Repetition Rate (kHz)	Pulse Energy (μJ)	Number of Pulses	Dimple Depth (μm)	Crown Height (μm)
Prediction by ANN	1200	7.62	364	4	1.26
Best from data (1^{st})	1200	7.58	400	4.04	1.29
Best from data (2^{nd})	1200	10.1	400	5.25	1.62
Best from data (3^{rd})	1000	8.91	200	4.19	1.63
Best from data (4^{th})	1200	12.4	400	4.44	1.74

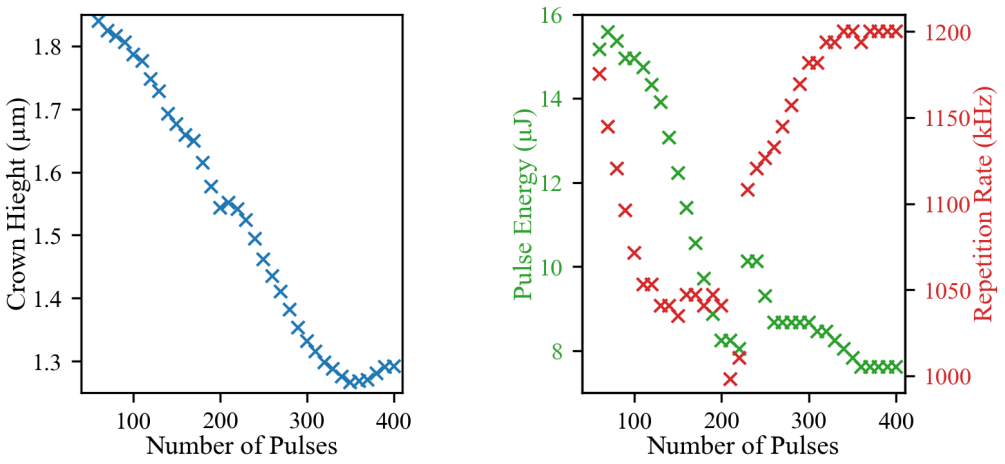
TABLE 4.3: Optimisation results from the network testing.

4.7.2 Finding Optimal Parameters

One of the most useful tasks to perform with the neural network is the determination of optimal parameters. Without the use of machine learning tasks, or other databases/algorithms there are two major approaches that can be taken. The first of these is taking a high-level sweep of the data, as exemplified in the dataset provided by Oxford lasers. This dataset can then form the basis of a database of data for current and future use. Once this has been performed, traditional optimisation techniques can be used to find a better solution. The issue with this is that the iterations required for optimisation will each require a full experimental setup which becomes unfeasible if any of the experimental collection or the set-up is time intensive. To combat this data will normally be collected in batches, but it can still be time-consuming, and overly large batch sizes can add time if unnecessary data is collected. In addition to this, a condition will be set to be good enough rather than finding a true optimum in the form of value caps or floors, or tolerances on fixed targets. While the tolerance issue will be true for any empirical method, the issue is the magnitude of the allowed error.

The amount of data that can be collected, and the speed of doing so is where the machine learning approach can offer a large advantage. When interrogating the ANN, receiving a prediction took 0.77 ms, without any need for set-up time or multiple stages of machining and measurement. As an example, performing a scan over the entire data space with a granularity of 50 steps per parameter would lead to 50^3, or 125,000 combinations to measure. Assuming a generous case of each experimental measurement taking 1 second total for machining and measurement, and no set-up time, the entire run would take more than 1.5 days. In contrast, running the entire set of combinations on the neural network took less than 100 seconds. This is combined with the lack of set-up time to allow the best use of optimisation techniques possible, all in an automated manner.

While many black-box optimisation techniques exist and can be used, that is not the focus of this investigation and so will not be explored here. The idea behind a black-box process is that it is independent of the process used to produce the data and so to assess the performance of the machine learning algorithm as it compares to an experimental technique, any optimisation search could be used. Here a simple



(A) The effect of the number of pulses on the minimum crown height

(B) The required parameters to meet the minimum crown height

FIGURE 4.8: Investigation into optimising machining speed showing a decrease in the minimum achievable crown height while maintaining the desired depth of 4 µm and increasing the number of pulses. This is achieved by reducing the pulse energy and varying the repetition rate. Modified from McDonnell et al. (2021a).

parameter sweep is used, testing each data point at an evenly spaced 50 points with the range of each of the variables using the ANN described previously. this parameter sweep was chosen as it does not rely on any aspects of the underlying data and will always contain a set number of steps. The best result that was found from the ANN is shown in Table Table 4.3 alongside the four best results from amongst all data in the training and validation datasets. The proposed optimisation task was to minimise the height of the raised crown of material while maintaining a dimple depth of at least 4 µm. The prediction from the ANN has managed to produce a solution that gives a smaller crown height than any of the data in the experimental set when keeping to the depth condition.

In each of the combinations, both experimental and network generated, the repetition rate was kept near the maximum of the allowed range, 1200 kHz. The number of pulses was also mainly kept high, with all of the experimental data except the 3rd best from the dataset. While this example appears to be much lower, at 200, it was the next lower step in the number of pulses tested in the original data collection. The prediction by the neural network drops the number of pulses slightly, down to 364 pulses, slightly below the maximum seen value. In the best combinations from the dataset, the pulse energy was kept low with the above-discussed parameters, the same being true with the prediction from the ANN. While these results were not experimentally verified, they did match the expectations of the technical staff at Oxford Lasers.

The fact that the prediction from the ANN does not utilise the maximum number of pulses brings up an interesting point for investigation. Finding an acceptable trade-off between crown height and the number of pulses used can act to minimise the time taken to machine a feature. This problem is investigated in Fig. 4.8a where the

minimum achievable crown height has been plotted against a range of numbers of pulses while maintaining a dimple depth of 4 µm. As expected from the results shown in Table 4.3, the lowest crown heights were found at the highest of the tested numbers of pulses. There is a peculiarity in that at the very high numbers, there is a turning point with a minimum value found at 360 pulses as shown in the table. While this could be a true effect, it goes against the general trend of decreasing crown height with increasing numbers of pulses. In addition to this, the technical staff at Oxford Lasers suggested that at higher pulse numbers than those tested, better results might be found.

To investigate the effect further, the corresponding pulse energy and repetition rate were plotted in Fig. 4.8b for each data point in Fig. 4.8a. This graph shows a possible explanation for the up tick in crown heights. Around the turning point of 360 pulses, both the repetition rate and pulse energy stabilise. This stabilisation occurs at the highest and lowest tested values respectively, with the repetition rate being the maximum possible with the laser. This capping of values would have put a limitation on the optimisation possible for reducing the crown height. A clear example of this is the inverse relationship between the number of pulses and the energy per pulse to balance the amount of energy deposited into the sample. Once the minimum tested energy is reached, the deposited energy cannot be offset and just increases, raising the crown height and the dimple depth.

If, however, the results are incorrect, it might highlight a limitation in the provided data. Rather than true, continuous, random values, each of the machining parameters had different individual steps they could be chosen from. The pulse energy was the most diverse, although this was due to the way the parameter selection occurred. Rather than being directly chosen, the pulse energy was chosen as a percentage of the maximum and then measured. The maximum energy, however, was not a fixed value, but rather changed with the repetition rate. This created a lot of granularity within the pulse energy parameter, although without a lot of cross-comparison. The repetition rate and the number of pulses were both fully independent variables, with the repetition rate having selections of 10, 100, 400, 600, 800, 1000, and 1200 Hz. The number of pulses had fewer steps with either of these, with steps appearing in powers of 2 having 4 data points of 50, 100, 200, and 400 pulses. This leads to very large gaps between the data points, especially between 200 and 400, which could be a cause of the uptick in the crown height after 360 pulses.

4.8 Conclusions

This examination of an experiment using a neural is a perfect example of how machine learning techniques can be used alongside traditional experimental

processes. In this example, the initial data collected by Oxford Lasers allowed for initial training of the network. This network was then interrogated to quickly test the results of machining a much larger array of different laser parameter combinations. This test revealed a turning point in the data that did not match previous expectations. The next step in this process could be to perform a more focused experimental search, investing the areas of higher numbers of pulses, high repetition rates, and low pulse energies. This data could then be fed back into the neural network as training data and additional training epochs performed. Generative techniques were also shown to provide a full experimental recreation. While initial results were not perfect they could be used for similar tasks to the analytical methods, trading a small amount of accuracy for the ability to visualise the results.

Chapter 5

Simulating Shaped Pulse Laser Machining via Neural Networks

As discussed in Chapter 2 modelling of femtosecond laser machining is a large topic that involves many difficult calculations and steps, even for simple Gaussian pulses. Scaling these models to three-dimensions with multiple incident pulses, would require a very complex model or a large number of assumptions and simplifications. These effects are especially true when using a spatial light modulator and the beam profile is poorly defined at its incidence to the surface.

In this chapter, pulses with computer-controlled, spatial intensity profiles are used to machine an electroless nickel-coated sample. The overall process for the investigation is shown in Fig. 5.1, with the first steps being the pattern generation before the laser machining (Fig. 5.1a). Once all patterns had been machined, the resultant depth profiles were measured using a white light interferometer (Fig. 5.1b), completing the data collection portion. This data was then used to train two generative neural networks (Fig. 5.1c), one performing the transformation in each direction. In this chapter, the details of the networks will be presented, along with the outputs they produce. These results will be investigated to answer the question of whether the neural network can be used to simulate the experimental process.

The work discussed in this chapter has been published in McDonnell et al. (2020, 2021a)

5.1 Motivation

While simulating the result of a single pulse is challenging, these techniques become even more complex when the sample will be machined with successive pulses in a single location on the sample. The simplest cause of this difficulty is the fact that the

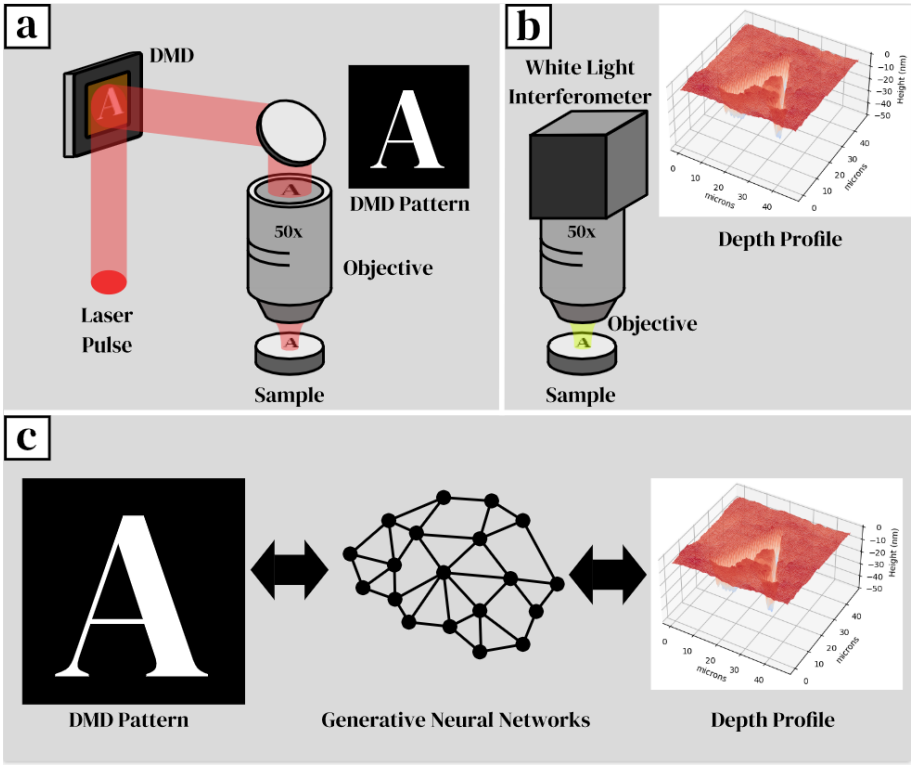


FIGURE 5.1: The experimental setup used with digital micro-mirror based experiments. The initial experimental data was machined using a digital micromirror device to shape pulses. The resultant depth profiles were then measured using white light interferometry. Modified from McDonnell et al. (2021b).

sample can no longer be assumed to be flat before any pulse except the first. Along with the obvious example of slopes in the surface, the interaction with the pulse will have also changed the properties of the material of the sample itself. While The effects will be less than with other types of laser machining, femtosecond laser will still introduce a heat effected zone that will grow over time as more pulses are used. This will change the material removal mechanisms as the incubation effect increases the temperature and leads to a higher proportion of melting. The importance of including these effects in a simulation are shown in Fig. 5.2. In this figure, the simple pixel removal method uses that same spatial pattern as the experimental profile, but assumes a fixed depth of material removal wherever the laser is transmitted.

On top of any difficulty in modelling the interaction mentioned above, the two domains do not share a one-to-one relationship. When machining with a laser there are a number of random or semi-random properties that can influence the results including: power variation of the laser, beam cross-section inhomogeneities, and imperfections and roughness on the surface of the target material. This means that using the same pattern sequence to machine in multiple locations across the sample may result in slightly different depth profiles. To a greater degree, the same is true in reverse, owing to phenomena such as the diffraction limit Hecht (1987), where the same depth profile could be caused by many pattern sequences. This becomes

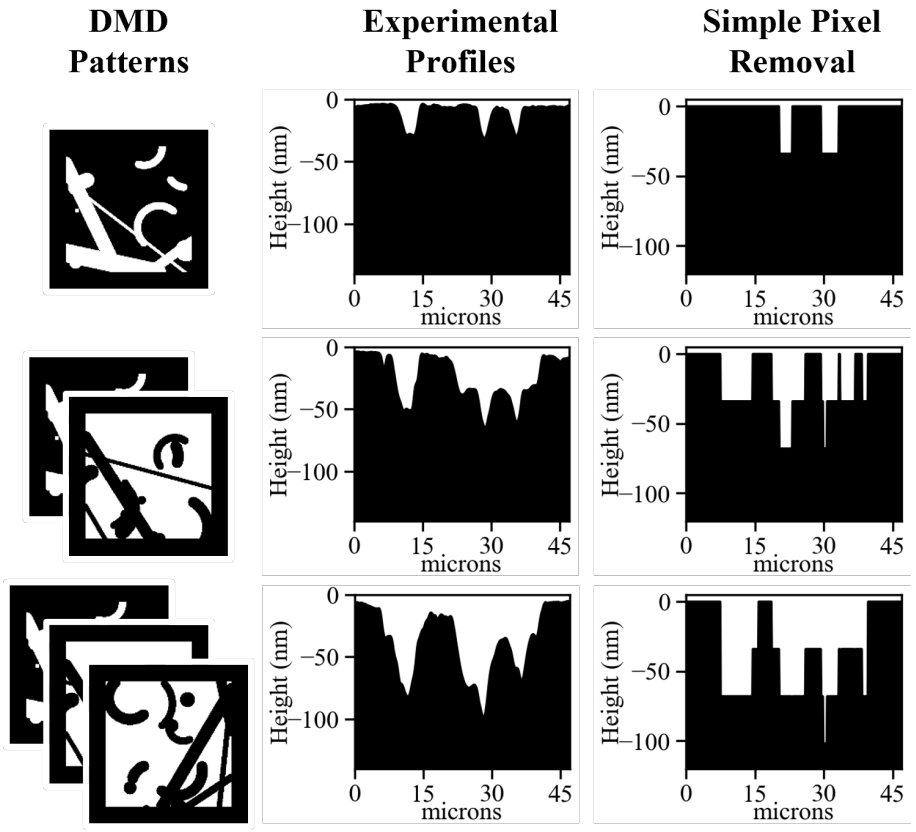


FIGURE 5.2: The central column shows a cross-section of the depth profile produced by machining with the corresponding sequence of DMD patterns in the first column. To demonstrate the complex interaction the final column represents the cumulative value that would be machined assuming the laser machined a fixed depth with a perfectly square profile.

especially true when more than one pattern is used, as the sequencing can lead to further possibilities, such as when there are two unconnected machined areas with a large (compared to the diffraction limit) area between them. These factors result in the initial sequence of patterns to the final, machined, depth profile having an approximately many-to-one relationship. This one-to-many relationship means that traditional comparative techniques can cause overfitting and a lack of variation that would be expected. The comparative losses can also lead to results that are correctly being labelled as wrong and so must be used with care.

The primary motivation for the work presented in this chapter is to demonstrate the effectiveness of neural networks, and GANs in particular for use in modelling laser machining. Novel neural network techniques are demonstrated to best suit the particular domain relationships.

5.2 Prior Art

One of the biggest advances that the group has made is in the use of Digital Micro-mirror Devices (DMDs). This work was initially started by Dr Ben Mills with simple pattern ablation tests demonstrating the abilities of the DMD Mills et al. (2013a). In this paper, it was shown that a DMD is a much cheaper alternative to a transmissive spatial light modulator (SLM). It has also been shown that the use of a DMD along with multiple exposures allows for sub-diffraction limit machining Heath et al. (2017). This approach has been shown to work on a number of materials and the use of a femtosecond laser allows for machining of materials that would otherwise be transparent at the wavelength of the laser, such as diamond Mills et al. (2014). This was made possible due to the use of multiphoton absorption as described in Section 2.2. Use of this effect has also been extended to multiphoton polymerisation to create complex patterns in a photoresist Mills et al. (2013b) useful for engraving and etching processes.

Along with the novel use of DMDs there has been an increasing utilisation of ML techniques and NNs, in particular, which are well suited to modelling the complex interactions involved. Machine learning has been widely used within laser machining (as discussed in Chapter 3) and this includes lots of work in combination with DMDs.

Work started with a demonstration of the ability of a GAN to be able to reproduce the surface of a machined sample, first in the form of scanning electron microscope images Mills et al. (2019b), and later full 3D depth profiles Heath et al. (2018b). These experiments involved exposing an electroless nickel-coated sample to a single femtosecond pulse, spatially shaped with a DMD. An electro9less nickel coating uses a chemical deposition process rather than a chemical on which leads to a high surface uniformity. The resultant patterns were analysed and then used, along with the initial DMD pattern, as the training input pairs for the networks. Once trained the networks could be used to predict the surface of the material after being exposed to a pulse with an arbitrary (binary) spatial profile, including those not seen while training. Throughout this process, the algorithm is never explicitly programmed with a theoretical understanding of the problem, but rather gathered all information empirically. This is a benefit when trying to model systems as complex as the ones seen here as no simplifications or assumptions were required.

5.3 Equipment Set-up

The experimental setup shown in Fig. 5.3 is the one used in most experiments within the group. In this setup patterned laser pulses are projected onto a sample with a digital micro-mirror device (DMD) acting as a spatial light modulator (SLM).

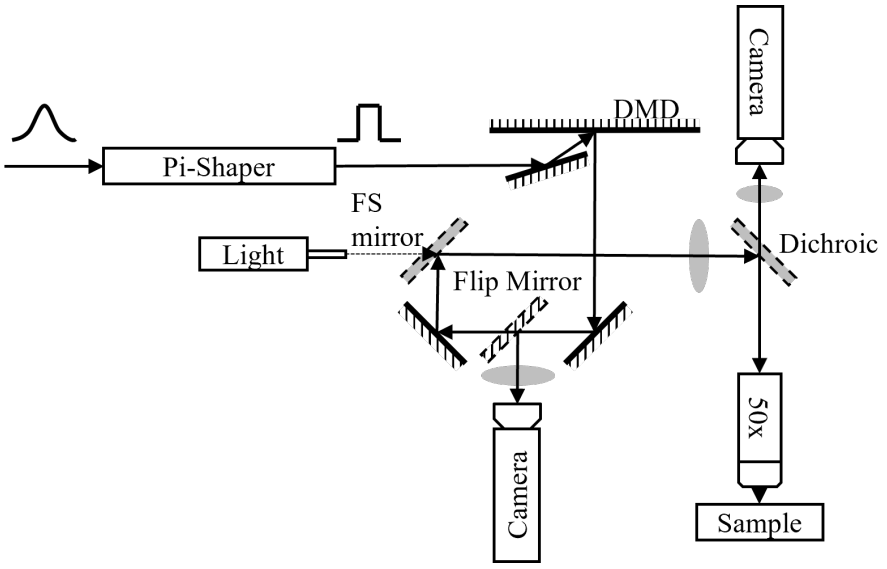


FIGURE 5.3: The experimental setup used with DMD experiments.

The laser used in the setup shown is a Ti:sapphire laser with a central output of 800 nm and pulse duration of 150 fs, seeded by an 800 nm, 80 MHz Ti:sapphire oscillator.

Immediately after exiting the laser the ideal beam has a Gaussian spatial intensity profile. This is not suitable for producing detailed structures from the DMD as a flat intensity profile is required. To rectify this, the beam is first passed through a Pi Shaper (π Shaper 6_6 — AdlOptica), which acts to transform the beam from a Gaussian spatial intensity profile to a top hat spatial intensity profile Laskin and Laskin (2011). After the Pi Shaper is the DMD, which acts as a blazed diffraction grating as well as its primary function as an SLM. From this grating, only a single diffracted order is captured, the one that is perpendicular to the surface. Additionally, if the beam is incident on the DMD from the correct angle, then this order carries the highest intensity. Selecting the central order minimises beam distortion, ensuring, for example, that circular patterns defined on the DMD do not produce elliptical beams. The selected order is then reflected by a series of 45° mirrors and directed towards the objective.

Before the objective, the beam passes through a lens in order to re-collimate the beam and to counteract some of the spatial dispersion caused by diffracting from the DMD. This is needed to preserve the wide bandwidth of the laser and allow short pulses. A dichroic mirror then directs the beam through the objective and onto the sample. The dichroic mirror was used so that the surface of the sample could be imaged from above while in position to be machined. The sample surface can be illuminated via a co-linear white light source that is positioned behind the multi-layer femtosecond mirror (which is transparent to visible wavelengths). Alternatively, there is a ring light attached to the objective to provide a dark-field view.

Samples are mounted on a 3-axis stage allowing movement in the horizontal plane for position and vertically to adjust the focus. The DMD can be set to display a simple shape, such as a circle or square, and then the laser used in a traditional way, by moving stages to track the beam across the sample. Alternatively, more complex patterns can be machined quickly by using the DMD to display discrete patterned areas. If a pattern is needed that is larger than the projected DMD area, then multiple pulses and displayed patterns can be used to create a single stitched pattern.

The flip mirror and camera allow the beam on the DMD surface to be imaged directly. This helps with Pi Shaper alignment, allowing the minimising of aberrations, and helps to ensure that the beam is centred on the DMD area.

To calculate the depth profiles of the samples, a Zygo Zeescope interferometric optical profiler, accurate to below 1 nm for depth measurements, was used after all desired positions had been machined.

5.3.1 Digital Micro-mirror Device

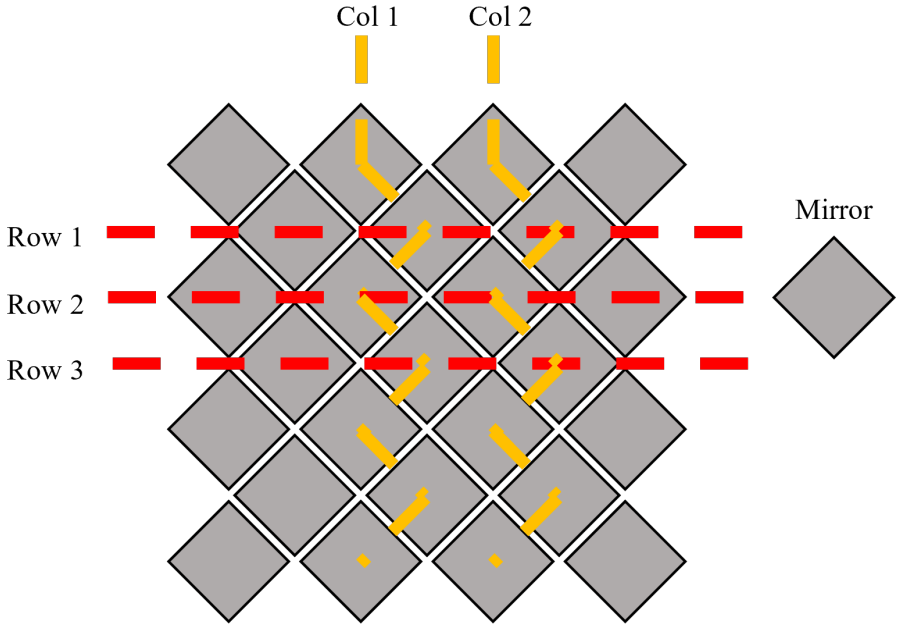


FIGURE 5.4: Diagram of the DLP-3000 DMD with a diamond grid pixel pattern.

A lot of the work conducted with the FS laser in the group uses a DMD. A DMD consists of an array of mirrors that can be used to reflect light in a structured way. The DMD used in all experiments in this report is a DLP3000, which has mirrors arranged in a diamond pattern as seen in Fig. 5.4. They are often used in projectors where they reflect light from LEDs through a lens and onto a surface. The use of them in a laser machining setup is slightly different for several reasons. The most significant difference is that it is not possible to use the DMD to control the exposure of each pixel

with a continuous value, only as a binary mask. Despite this, it is possible to simulate a greyscale value when the projected pixel sizer is smaller than the diffraction limit of the machining process with the use of static dithering. In normal operation, exposure is controlled by rapidly rotating the mirror, equivalent to rapidly turning an LED on and off, however, this will not work for FS pulses as the pulse duration is shorter than the oscillation period of the mirrors.

The DMD can be used for several tasks, most simply changing a spot diameter by displaying a circle of any size. In more complicated cases, it can be used to project complex shapes, difficult to achieve via other methods in a single exposure. Smooth intensity gradients can even be achieved by using a chequered pattern of active pixels Heath et al. (2018a). This offers the opportunity to study unique interactions at the cost of being harder to model.

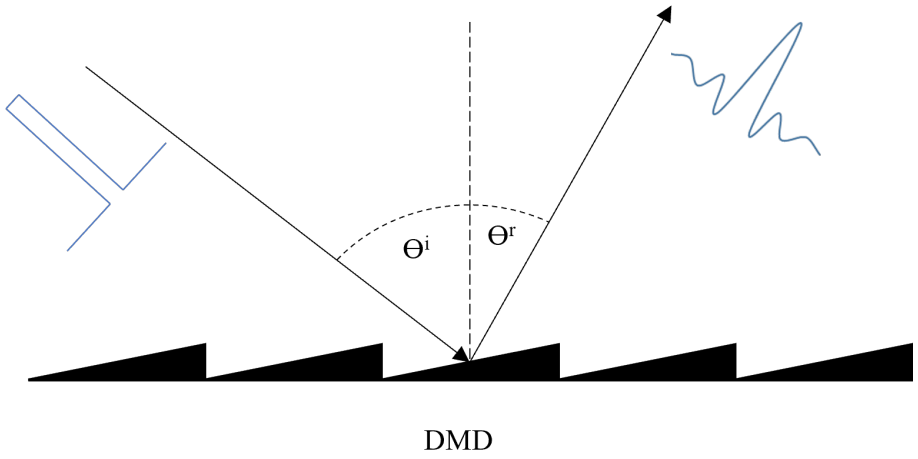


FIGURE 5.5: A demonstration of the light interaction with the mirrors in the DMD. Due tho their fine pitch they act like a diffraction grating rather than a pure reflective surface.

One of the biggest problems in modelling the light interactions on the surface is trying to model the beam properties after the DMD. Immediately before the DMD, the beam is well-defined as a uniform top hat beam. After the DMD, however, the beam has changed significantly with diffraction, as the DMD itself is a grating as illustrated in Fig. 5.5. The other effect is that when passing through lenses that are small compared to the angular spread of colours within the beam, clipping may occur. This means that high-frequency spatial information, along with the extremes of the spectral bandwidth, will be lost, further reducing the resolution of the pattern projected to the surface. All these factors combine to make modelling of the machining process almost impossible, although the focus of the work here is looking at the effect of multiple pulses rather than that specifically of the DMD.

5.3.2 Pi Shaper

As stated previously, to be used effectively, the incident beam onto the DMD should have a top-hat spatial intensity profile. To convert the Gaussian profile from the laser, a Pi Shaper is used which enables a pulse of a specified diameter to be converted with very little loss. The Pi Shaper works with the use of two paired specially shaped lenses as shown in Fig. 5.6. The first lens acts to diverge the beam in the centre and converge the beam at the extremities. This flattens the distribution with an opposing lens collimating the beam. The beam path throughout the Pi Shaper is kept constant to ensure there is no tilt on the pulse.

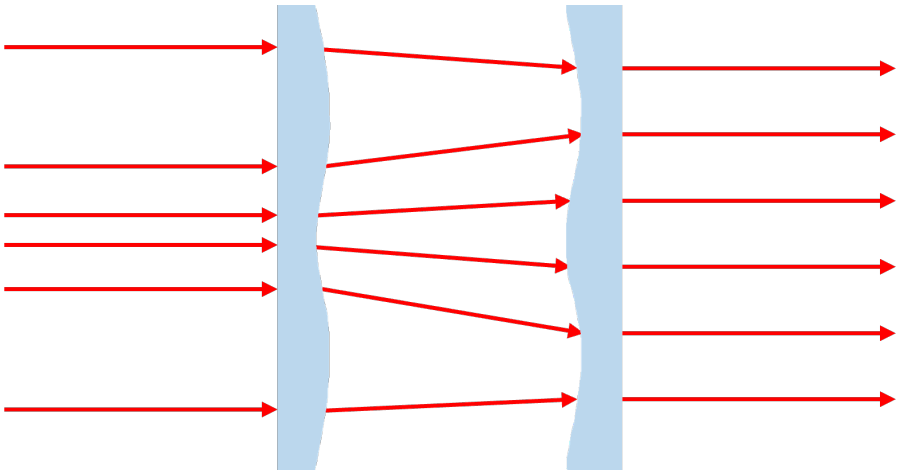


FIGURE 5.6: The Pi Shaper works by refracting some of the high-intensity radiation at the centre of the pulse towards the edge of the pulse as well as directing some of the low-intensity skirt inwards. Due to the fixed shape of the lens shape the Pi Shaper was only designed to work with an ideal 6 m m Gaussian beam.

5.4 Experimental Data

The extension from 1 pulse to 3 pulses was chosen for a few reasons. One was to provide enough permutations ($2^3 = 8$) in order to demonstrate the effectiveness of this technique for predicting the effect of the order of multiple laser pulses during laser machining. The reason that the limit was chosen as three rather than higher was to keep the data easily understandable by humans and as a stepping stone to prove the concept that multiple pulses (i.e. an unlimited number of pulses) could in principle be simulated via a neural network.

The dataset used for the experiment consisted of 484 positions at which between 1 and 3 machined patterns were machined into the sample surface. These patterns were semi-randomly generated patterns consisting of lines, circles, and arcs with an overall predetermined fill density. In a third of cases, the areas to be machined are inverted and so the patterns represent the areas not to be machined. Examples of these patterns

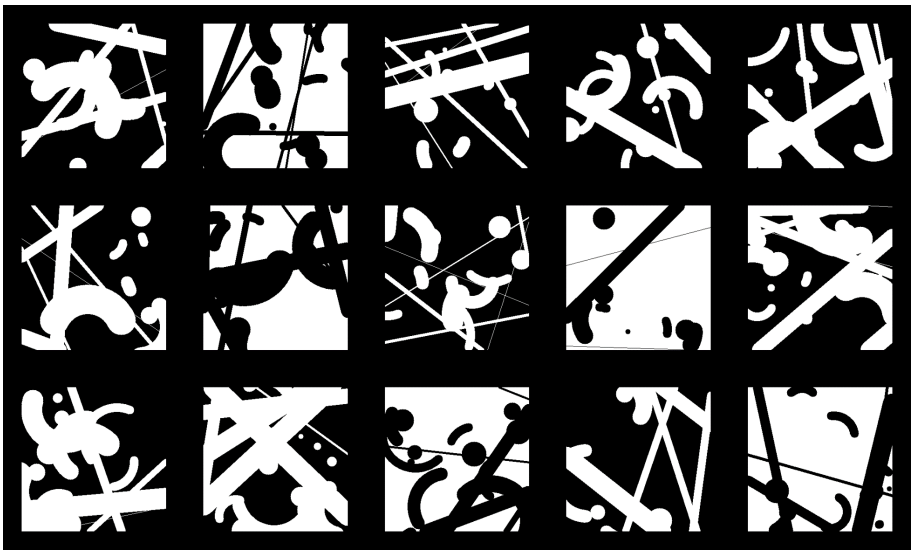


FIGURE 5.7: Example generated DMD patterns using a range of sizes of lines, arcs and cirles. In a third of cases, the image was inverted in the machining region.

can be seen in Fig. 5.7 where the white pixels represent 'on' pixels where the material should be machined.

Each pixel in the depth profiles shown corresponds to a theoretical size of 92 nm on the sample. The sizes of the features ranged both above and below the diffraction limit of the setup when the patterns were imaged onto the sample. This variety of input shapes was designed to allow the neural network to learn the laser machining process without simply remembering what the output of each pattern should be. Each overall structure produced via up to 3 pulses was around 30 µm across. For the training of the NN, a total of 404 input-output pairs were recorded and used.

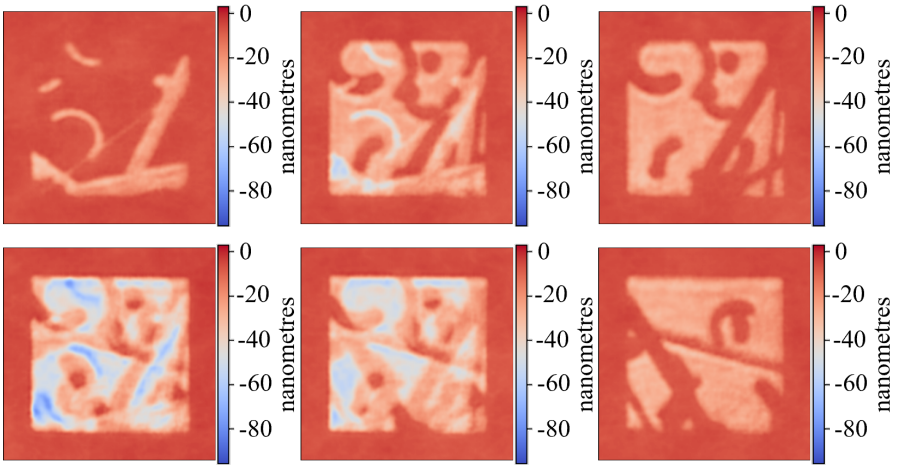


FIGURE 5.8: Example depth profiles from the experimental process.

Once the laser machining step had been completed and the height maps had been measured using WLI, the data was available in the form of 2 dimensional arrays of float values. Pictoral representations of a sample of the final height profiles can be

seen in Fig. 5.8. It is worth noting that the general depth across the machined area is consistent with a greater depth of machining in the upper left corner of the images shown in Fig. 5.8. This is a real effect in the laser used where the intensity incident on the surface was not uniform. This effect justifies one of the decisions made when training the network, not to include any rotational augmentation. While rotating the data pairs in increments of 90 degrees would provide effectively 4 times as much data and reduce the risk of overfitting, the network can lose some specifics of the system.

5.5 Neural Network Architecture

To complete the planned experiments covered in this chapter, a number of neural networks were created and combined in a way to maximise their performance. The first case of interest was the generation of depth profiles given a sequence of DMD patterns. To allow for simplicity and to reduce the size of the dataset required, sequences of between 1 and 3 DMD patterns were chosen. The use of a maximum of three sequences in the pattern allowed for easy compatibility with existing neural networks. It also gave the ability to easily display information in the form of RGB images, with each colour channel representing a unique DMD pattern as shown in Fig. 5.9. Due to the equipment set up it was only possible to capture the 3D surface profile after the machining had taken place and so fixed sequences were used. With the ability to capture the surface before and after each laser pulse, the network could be extended to be able to calculate an arbitrary number of pulses, each with a unique spacial profile.

As all laser parameters, excluding the spatial intensity profile, were kept consistent between each pulse, only two sets of information were needed for training. The first of these was the sequence of DMD patterns, formatted as a 512×512×3 array, examples of which can be seen in Fig. 5.9. The first two dimensions formed the binary masks representing each pattern displayed on the DMD while machining, with the third dimension representing the ordering of the up to three patterns used, with the first at index 0. The depth profile was represented by a 512×512, 2D array with each value in the array representing the depth in nanometres, compared to the un-machined area, of the corresponding position on the sample.

Due to the ability to collect one-to-one paired data, and the existence of random features in the output, such as debris and variations in the laser intensity profile, it was decided to use a GAN for these tasks. As discussed in Chapter 3, GANs allow for the creation of realistic images that are indistinguishable from the experimental data. The GAN would also be capable of learning features of the beam without having to be explicitly stated, such as an uneven intensity distribution across the pulse.

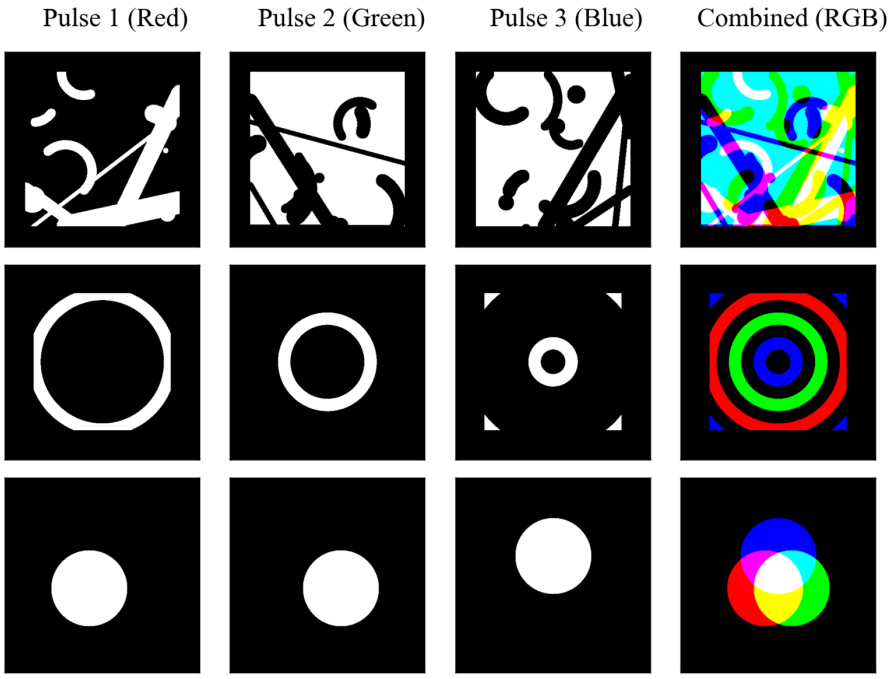


FIGURE 5.9: RGB representation of individual three-pulse sequences. The final column shows a combination of the previous columns using the colour shown in the header as their channel. As the images were provided to the network as 512×512×3 arrays the RGB representation is an accurate view of what the network uses.

5.5.1 Generating Depth Profiles

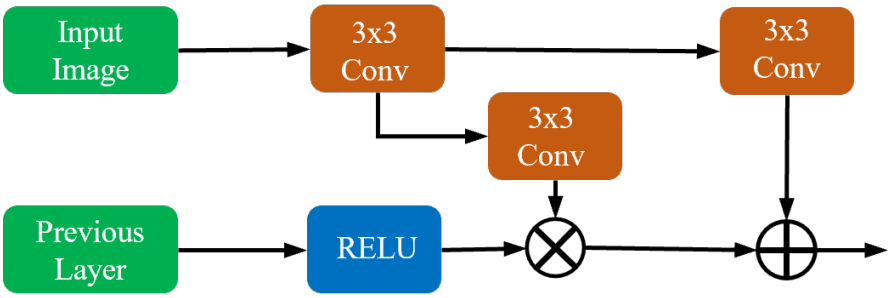
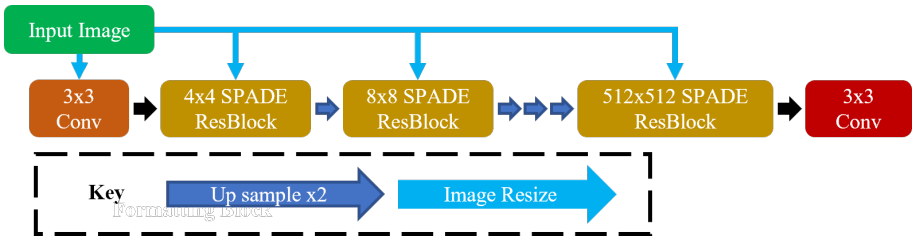


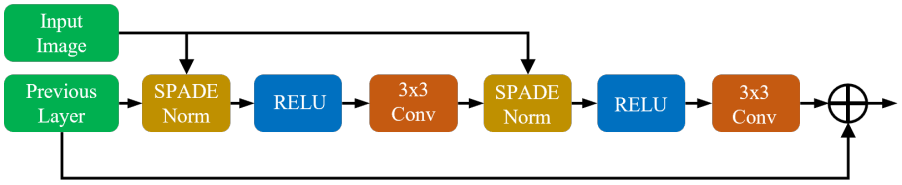
FIGURE 5.10: A block diagram of a SPADE Normalisation Block combining a revised version of the initial input image with the previous layers of the network.

The network used to generate depth profiles from DMD patterns was based on a development of the pix2pixHD network Wang et al. (2018), called SPADE Park et al. (2019). This network was specifically developed to convert flat, semantic images, into photo realistic images. Semantic images consist of flat (unshaded) blocks of colour, where each colour has a specified meaning. For example an area of light green could represent grass, and light blue could represent the sky. This shares a lot of similarities with the representation of the sequences of DMD patterns, as shown in Fig. 5.9. In these representations, each of the primary colours relates to the 'on' pixels for each pattern. One of the major developments of this network was the incorporation of the

SPADE normalisation block, the structure of which is shown in Fig. 5.10. The SPADE normalisation block replaces other normalisation techniques, such as instance normalisation, and is designed to better preserve the semantic information. The SPADE normalisation block, takes two inputs, one is from the preceding layer in the network, and another is an appropriately resized version of the initial image input of the network. The image is first convolved onto an embedding space before two further independent convolutions produce two modulation tensors, labelled γ and β. These two tensors are then multiplied and summed element wise respectively to the batch normalised output from the previous layer.



(A) Network structure



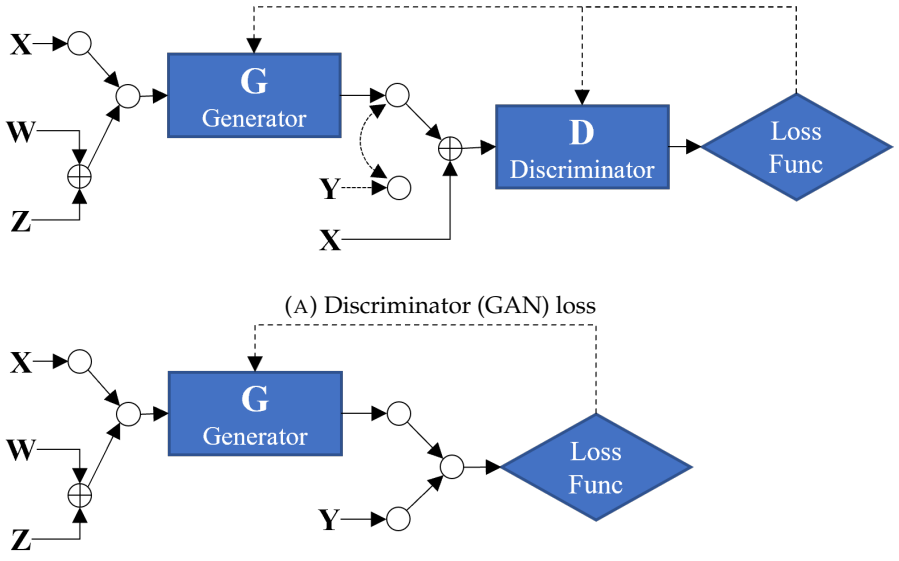
(B) SPADE ResBlock

FIGURE 5.11: Network Block Diagram showing both the generator and the discriminator parts of the GAN.

The generator uses an up-scaling generator, eschewing the down-sampling path of a traditional U-net structure Ronneberger et al. (2015). This allows for the network to contain fewer parameters and therefore be quicker to train. The full structure is shown Fig. 5.11a the method of building from the initial input, all the way to the final output is shown. One of the frequently occurring components in the network is the SPADE ResBlock (Fig. 5.11b). This performs the same task as the initial ResBlocks He et al. (2016) but has been updated to incorporate the SPADE normalisation block Park et al. (2019). Each block is formed of a repeated set of three layers: a SPADE normalisation block, a ReLU activation, and a convolution with a kernel size of 3. The initial SPADE normalisation block takes an input of the output from the layer preceding the block, as well as an appropriately resized version of the network input image. The second uses that same resized image as the first, but also uses the output from the proceeding convolutional layer. As in the original ResBlock, the output after the final layer of the block is summed with the output of the layer preceding the block.

The first layer of the network is a Convolution layer with a kernel size of 3, which takes in a single input of the initial image resized to 4×4 pixels. This is then followed by a sequence of a SPADE ResBlock and a 2x up-sampling layer using nearest

neighbour interpolation, increasing the working size to 8×8 pixels. This is then repeated until the desired size of 512×512 has been achieved. After the last up-sampling layer there is a final SPADE ResBlock followed by a Convolution layer with a kernel size of 3 and 1 filter to form the output image.



(A) Discriminator (GAN) loss

(B) Comparative (GAN) loss

FIGURE 5.12: Network loss configurations. X and W are network inputs, Y is the experimental output target, and Z is noise. Modified from McDonnell et al. (2021b).

In order to train the network the Adam optimiser was used with a beta_1 of 0.5, a beta_2 of 0.999, and a learning rate of 2×10^{-4}. As this network was trained on paired data, it could be trained based on two of losses shown in Fig. 5.12, using the traditional GAN loss (Fig. 5.12a) and a comparative loss (Fig. 5.12b). In the figure, The letters related to different inputs, and in the case of this network, X is the depth profile to be used to generate the sequence of DMD patterns. The other inputs are W, the weighting vector used for pulse ordering, and Z, the noise input used to provide variance. Where multiple inputs are used there are two methods used, they are concatenated in situations with a plus, and treated as separate variables where there is a blank circle. In some cases one of two inputs are used, as in Fig. 5.12a, and this is indicated by a dashed and curved line between the two possibilities. The generator (**G**) in the images all refers to the same network but has been included in each of the configurations for the sake of clarity.

The first of the losses described is the traditional GAN loss, using the discriminator to determine the quality of the generator. Using this loss can generally not provide a learning cut-off point as both networks are continuously learning, even though the generator will improve. The ideal case is that the two networks will balance each other and should maintain an even ability to generate and recognise images, therefore the loss will not show a downward trend. The other loss used was the direct comparative loss, and in this case, the specific method used was the mean absolute

error. This again was used to provide a loss to train the network with, and to give a better overall idea of how the training was progressing throughout. While an exact cut-off was not used training was stopped after 300 epochs, once the losses had stabilised, and the depth profiles produced were with an MAE of less than 3 nm.

5.5.2 Generating Sequences of DMD patterns

A slightly different approach was taken for the network used to produce DMD patterns. This generator was again based on the SPADE network, formed of an up-scaling convolutional network comprised of SPADE ResBlocks and $\times 2$ nearest neighbour up-scaling layers. The network started at a resolution of 4 pixels by 4 pixels, going up to the final resolution of 512 pixels by 512 pixels. In an addition to the previously described network, a mapping input was also used, comprised of the weighting vector and random noise. The noise portion of the input was included to allow for the network to perform the one-to-many transformation, as it allowed for different results from the same depth profile and weighting pairs. The weighting was used to describe the desired pulse make-up of the final sequence of DMD patterns, allowing control over the pulse breakdown. The weighting this way could only be used as an input to the network rather than an output, and this was decided as a way to investigate the amount of control possible over the network, rather than to use it as a direct optimisation tool.

The mapping input was included in the network with a series of 4 Dense layers followed by a reshape layer to form a 4×4×128 layer. This output was then resized and used at each SPADE ResBlock in the network, being concatenated with the input. The first SPADE ResBlock used 1024 filers, reducing with each scaling value, reducing down to 32 filters by the final block operating at 512 pixels by 512 pixels. After the final SPADE ResBlock there was a single Convolution layer with 3 layers that formed the output of the network and produced the sequence of DMD patterns.

The discriminator used was a variation on a patch GAN discriminator, taking inputs of both the weighting vector and the sequence of DMD patterns. Through a series of strided convolutions, the sequence of DMD patterns are downsized gradually from original size down to 32 pixels by 32 pixels, with the final convolution only having a single filter. The output from this layer was then flattened to form a vector of size 1024. Concurrently the weighting vector passes through a Dense layer to then be concatenated with the flattened DMD sequence vector. This combined vector is then passed through a further two Dense layers, the final one having a single filter to produce the output. This output was a prediction on whether the input pair corresponded to a real, experimental pair, or to one created by the generator.

One of the major differences with the GAN to produce the sequences of DMD patterns was the structure of the losses chosen. Due to the losses available to them, in particular comparative losses such as MAE, GANs that are trained on paired data are often easier to train than those where the data is not directly paired. As discussed in Chapter 3 one method to get around this was the CycleGAN, where the advantage of comparative errors can be used. While traditional paired data techniques worked for the GAN discussed in Section 5.5.1, they were found to be ineffective in this situation. This was likely due to the highly many-to-one nature of the data, and comparative losses being ineffective. To counter this, a partial CycleGAN was used, with only the DMD generation GAN utilising this technique. This was chosen as the depth profile generation GAN could be proven to be effective without this, and that meant that it could be used in this process, as well as reducing training time by only having to train one further network rather than both operating under the CycleGAN simultaneously.

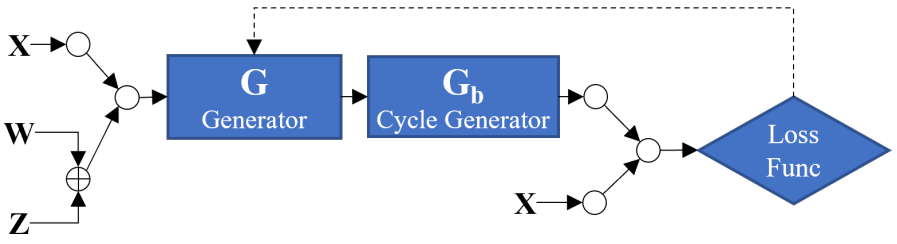


FIGURE 5.13: Cycle consistent loss. X and W are network inputs, Y is the experimental output target, and Z is noise. Modified from McDonnell et al. (2021b).

The network used to produce the sequences of DMD patterns was trained via the use of three different loss functions simultaneously. These three losses include the two shown in Fig. 5.12, as well as a new one shown in Fig. 5.13 utilising all three of: the traditional GAN loss (Fig. 5.12a), a comparative loss (Fig. 5.12b), and cycle consistent loss (Fig. 5.13). The first two losses were used in the same way as the previous network, although with less emphasis on the comparative loss due to the difference in aims between the two. The main addition here was the cycle consistent loss, which utilised the depth profile prediction network as the secondary network shown in the figure. While in a traditional cycle consistent setting, both networks would be trained simultaneously, in this situation this was not required as the secondary generator had already been trained and tested. While the comparative loss had less importance later on in training, early on it was heavily weighted to promote initial pixel accuracy, before later refinement. The traditional, comparative, and cycle losses had a normalised weighting at a ratio of 1:10:1 respectively initially, dropping to 5:1:5 by the end of training. This meant that the comparative loss contributed to the training of the network far more initially, dropping to a fifth of the other two by the end.

5.6 Analysing Network Performance — Generating Depth Profiles

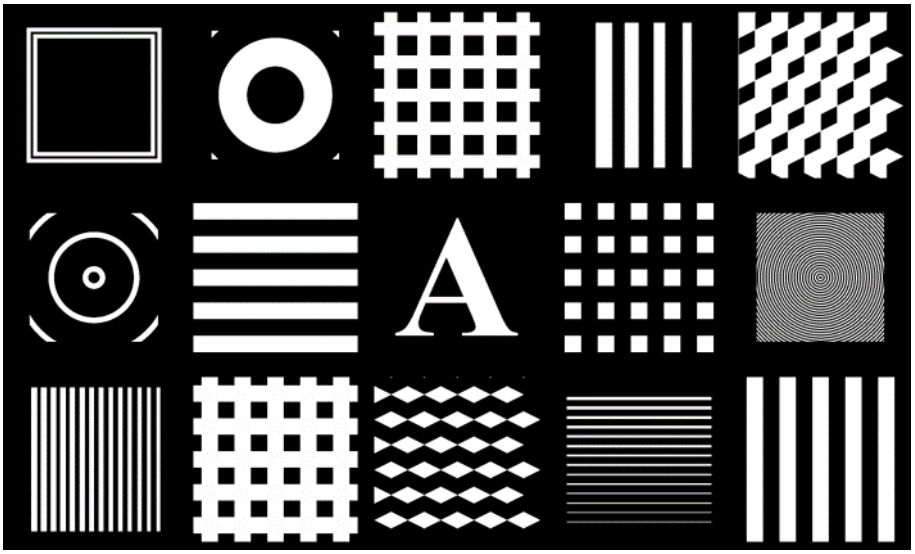


FIGURE 5.14: Examples of patterns designed to investigate the limits of the network while also displaying features that could be easily judged by humans.

As always in machine learning, a validation dataset was kept separate from the main training data, and used to verify the performance of the network. Alongside the randomly generated images of the style shown in Fig. 5.7, input data with user-defined structure was also included in the validation set, and can be seen in Fig. 5.14. These were included to try to interrogate the network and see what it had learned with human-understandable data, such as the letters. This also allowed specific situations to be investigated such as the effect of ordering on different structures, such as looking at the grid structures.

Once trained the network was able to generate predictions for the depth profiles produced by sequences of DMD patterns, examples of which can be seen in Fig. 5.15. The two experimental and generated profiles, while not identical, show good agreement, and the differences can be attributed to random noise within the experiment that the network could not be expected to match perfectly. Examples of things that contribute to the noise include any power fluctuations within the laser, both peak power and intensity distribution, as well as any defects in the sample. This can be seen especially in the unmachined area beyond the extent of the laser pulse, where the random features on the surface of the sample exist. A more accurate result could have been achieved by capturing the depth profile before each pulse, however, at the time of conducting the experiment, this was not feasible. The reason for the difficulty was due to the fact that the machining setup, and the white light interferometer were not in the same setup, but were in different buildings. Another option would have been to use a camera image, but this would not contain all depth

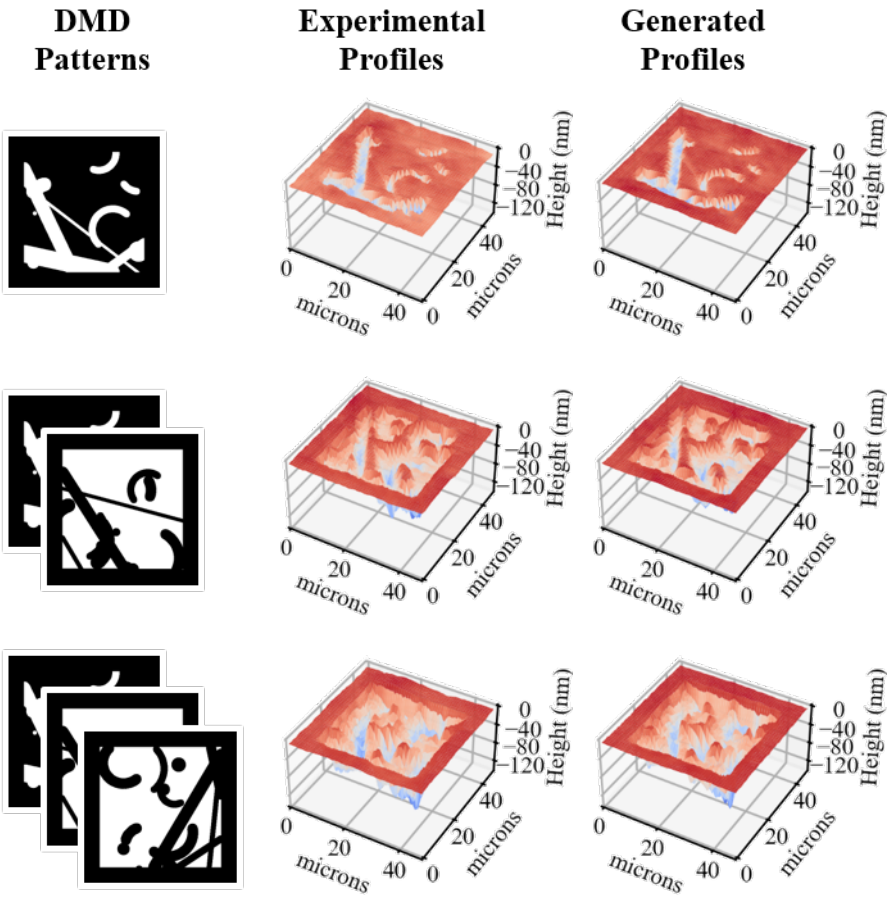


FIGURE 5.15: A comparison of experimental and generated depth profiles over a series of three shaped pulses, each building on the previous.

information and with the setup, it was not possible to capture images of sufficient sharpness and clarity.

To better compare the experimental and generated profiles, a cross-section of each of the depth profiles is shown in Fig. 5.16. While each of the matching profiles were not exact matches, this is not expected due to the variations in the machining process such as laser power fluctuations and sample imperfections. Despite this, the results were far superior to the simple removal shown in Fig. 5.2, where a pixel-perfect removal method was shown. The overall softening of features has been captured well with many small rounded peaks present and similar depths across the profile. The mean absolute errors of the generated profiles, in this case, were 3.5 nm, 4.3 nm, and 4.5 nm, for the one, two, and three pulse case respectively.

5.6.1 Pulse Ordering

When performing a laser machining task, there is generally a goal to be reached for the resulting surface profile, and one factor that can affect this is the shape and order

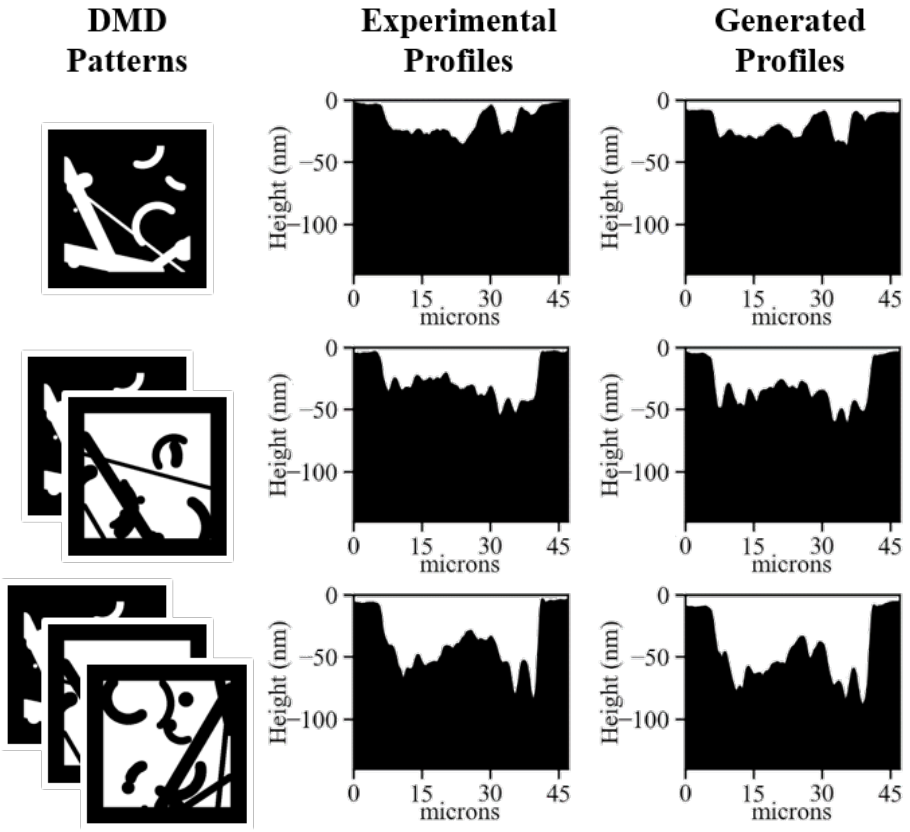
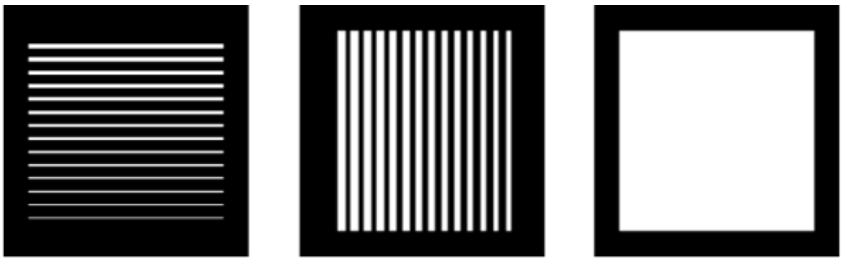


FIGURE 5.16: A comparison of depth profile cross-sections over a series of three shaped pulses. The generated profiles show similarities to the experimental profile in the softness of the machining and the comuliliative effects of machining in a single location.

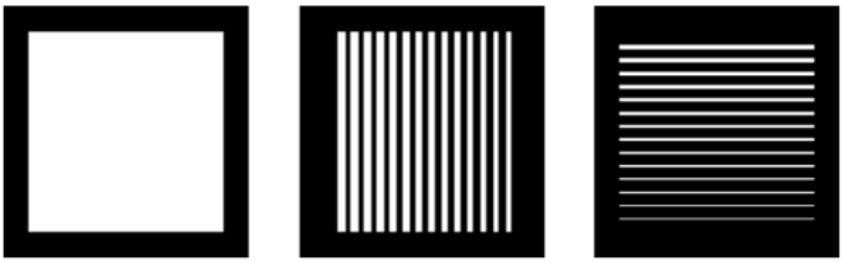
of the pulses. As an example, when trying to machine clean, sharp features in an overall recess, would it be beneficial to first machine the detailed area first, or would the results be better when machining the overall depression initially? This situation was explored using various overlapping grid structures solid squares, examples of which are shown in Fig. 5.17. Grids of varying sizes were used to test this, with Figs. 5.17a and 5.17b showing the extreme fine example and Figs. 5.17c and 5.17d represent the sequences with the largest features and spacings. The sequences where the solid square is listed as the final pulse (as in Figs. 5.17a and 5.17c) are referred to as grid first sequences, while ones where the squares are machined first, are referred to as grid last sequences.

$$\iint_{\xi} \left[\frac{d^2 g(x,y)}{dx^2} + \frac{d^2 g(x,y)}{dy^2} \right]^2 dxdy \tag{5.1}$$

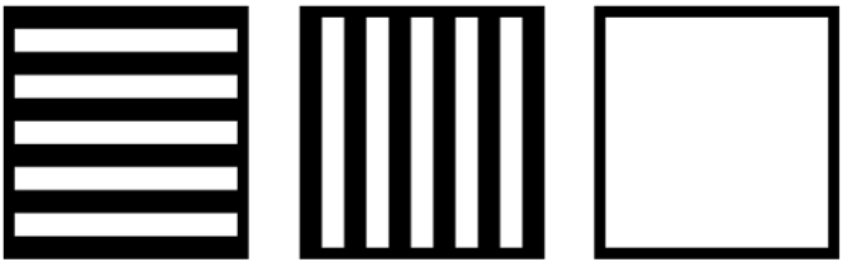
In order to quantify this effect, a way to measure the difference was required. The method chosen for this was measuring the sharpness of the depth profile, with the crisper depth profiles having a higher sharpness. This sharpness was calculated using



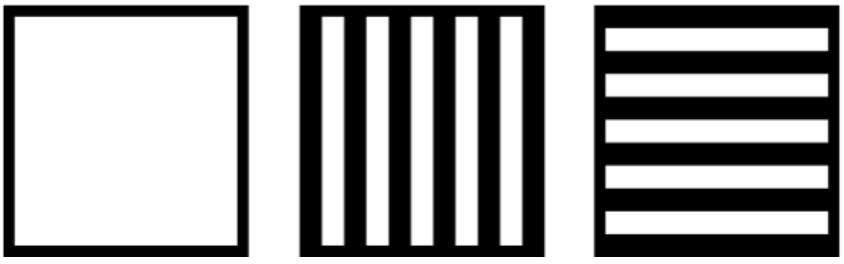
(A) DMD pattern with a fine grid machined before a large square



(B) DMD pattern with a fine grid machined after a large square



(C) DMD pattern with a coarse grid machined before a large square

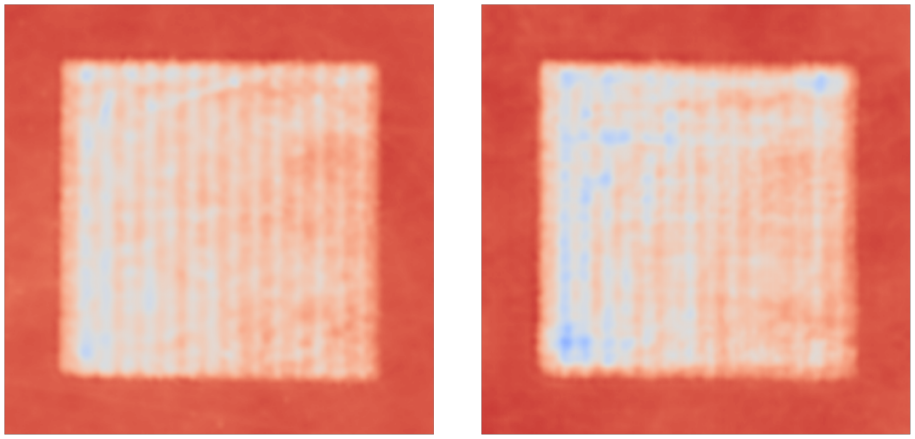


(D) DMD pattern with a coarse grid machined after a large square

FIGURE 5.17: Sequences of DMD patterns used to test pulse ordering with fine and course grid li9nes machined before and after a large square.

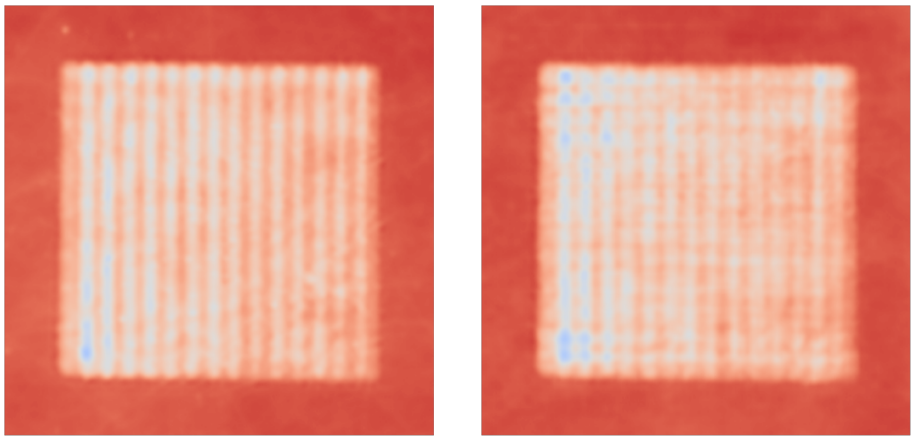
the Laplacian energy Wee and Paramesran (2007) of the depth profile array. To calculate the energy Eq. 5.1 is used, with $g(x, y)$ representing the depth profile array.

The results from machining the fine grids shown in Fig. 5.17 are presented in Fig. 5.18 alongside the predictions from the neural network. The profiles shown in Fig. 5.18a were produced using the grid first sequence of the fine grid patterns shown in Fig. 5.17a and Fig. 5.18b was produced using the grid last sequence of the same scale.



Experimental Depth Profile Generated Depth Profile

(A) Machined with the grids first



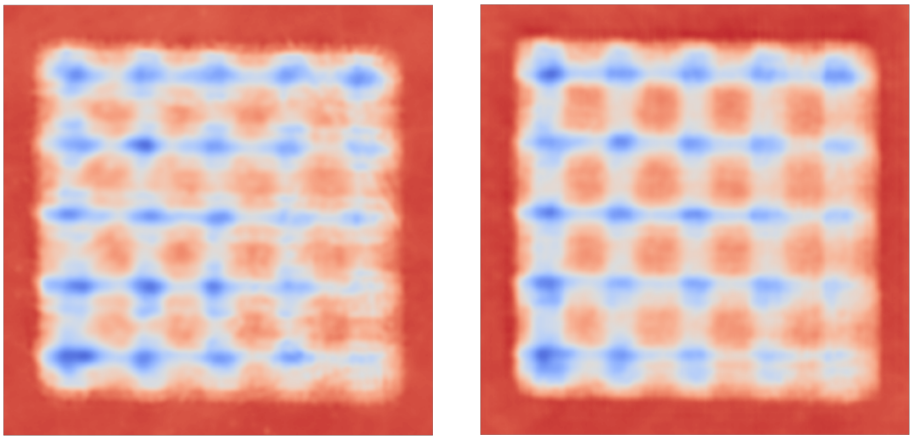
Experimental Depth Profile Generated Depth Profile

(B) Machined with the grids last

FIGURE 5.18: Depth profiles produced using the sequences of DMDs shown in Fig. 5.17a and Fig. 5.17b respectively. When the grids were machined first the details are softer than when they were machined after the square.

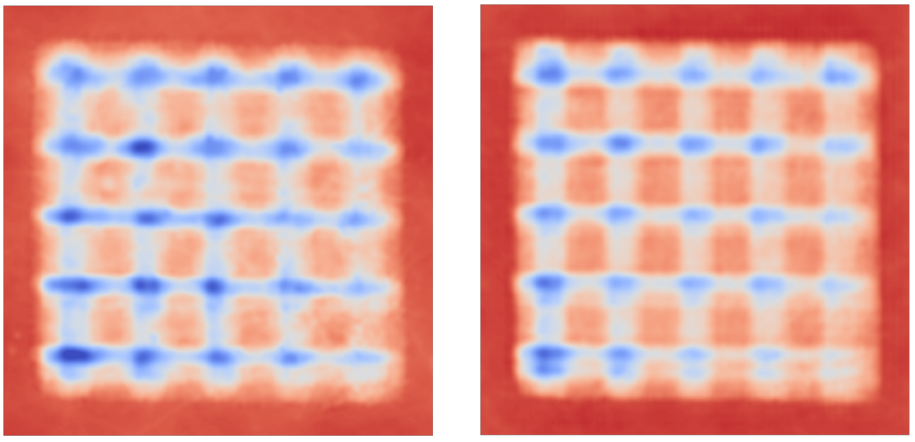
When examining the experimental depth profile from the two profiles, it is clear to see that the features produced by the grid last sequence remain sharper than those produced by the grid first sequence. This can be seen in particular along the vertical ridges which are all clearly visible in the grid last profile, but become difficult to make out in the grids first profile. The same is true for the profiles generated by the neural network, although to a lesser extreme in both cases, indicating that there is still room for improvement within the network. In all cases it can be seen that there is an area of higher machining (greater depth) in the lower left corner, which is present to a greater extent in the profiles where the final machining step was the solid square.

The same effect can be seen When the grid size is larger, with both thicker features and larger gaps as demonstrated in Fig. 5.19. Both the experimental and the generated profile in Fig. 5.19b, where the grid has been machined last, are sharper and better



Experimental Depth Profile Generated Depth Profile

(A) Machined with the grids first



Experimental Depth Profile Generated Depth Profile

(B) Machined with the grids last

FIGURE 5.19: Depth profiles produced using the sequences of DMDs shown in Fig. 5.17c and Fig. 5.17d respectively. Again, when the grids were machined first the details are softer than when they were machined after the square.

defined than when a solid feature is machined afterwards. This is as expected, however, in both examples, the grids predicted by the neural network are more well-defined in most areas than the predicted depth. This contrasts with the examples shown in Fig. 5.18 where the features in the generated depth profile are slightly softer than those in the experimental profile.

Using Eq. 5.1 the energy can be calculated for all depth profiles shown in Figs. 5.18 and 5.19 and the results are presented in Table 5.1. Looking at the fine grids in Fig. 5.19b, when the grids are machined after the large feature, the energy of the experimental depth profile was calculated to be 8.89×10^{-4} with a standard deviation of 0.298×10^{-4} compared while the energy of the generated profiles was calculated as 11.08×10^{-4}. Taking the case of machining the grids before the square, the mean energy of machined patterns was 7.27×10^{-4} with a standard deviation of

	Sharpness (a.u.)	
	Experimental Profile	**Generated Profile**
Grid First	7.27×10^{-4}	9.05×10^{-4}
Grid Last	8.89×10^{-4}	11.08×10^{-4}

(A) Fine Features

	Sharpness (a.u.)	
	Experimental Profile	**Generated Profile**
Grid First	13.0×10^{-4}	10.2×10^{-4}
Grid Last	14.5×10^{-4}	11.1×10^{-4}

(B) Coarse Features

TABLE 5.1: The sharpness metrics for experimentally measured and generated depth profiles

0.154×10^{-4}. This compares to the value of 9.05×10^{-4} when calculating the energy of the generated profile for the same case.

Taking the alternate case of the coarse features, as presented in Table 5.1b, The mean Laplacian of the experimental profile when the square was machined first was calculated as 14.5×10^{-4} with a standard deviation of 0.218×10^{-4}, higher than the value of 11.1×10^{-4} for the generated profile. Similarly, the energy of the experimental profile for the square machined last case of 13.0×10^{-4}, standard deviation of 0.142×10^{-4} is also higher than that of the generated profile which was calculated as 10.2×10^{-4}.

The sharpness values for each of these conditions roughly relate to how well-defined the machined edges are. This suggests that the edges are sharper when the large solid square is machined before the grating structure than when it is machined after. This agrees with the idea that features will be softened by re-machining the same area, hence reducing edge steepness. The other most notable feature of the data is that the sharpness in the generated data is less varied than the experimental data.

While The GAN does correctly predict that machining solid features will soften any previous features, the difference between the two grids is very small. Part of this is due to artefacts in the profiles that are not easily visible. The artefacting only appears when taking the sharpness of the profile, with a noticeable difference between the experimental and generated profile. These artefacting issues are likely due to the kernel size influence on pixels discussed in Chapter 3 where there are periodic effects. Despite this, the trend is the same between the experimental and generated profiles and this suggests that the neural network is capable of determining the effect of ordering of pulse patterns.

5.6.2 Multiple Pulses and the Diffraction Limit

Beyond simple rapid prototyping, one novel use of neural networks is an investigation of the physics involved in the experiment. One aspect that can be investigated is the diffraction limit of the laser machining setup for a single exposure. This is a good value to look at, as an approximate value can be calculated very easily using the Abbé diffraction limit (Eq. 5.2).

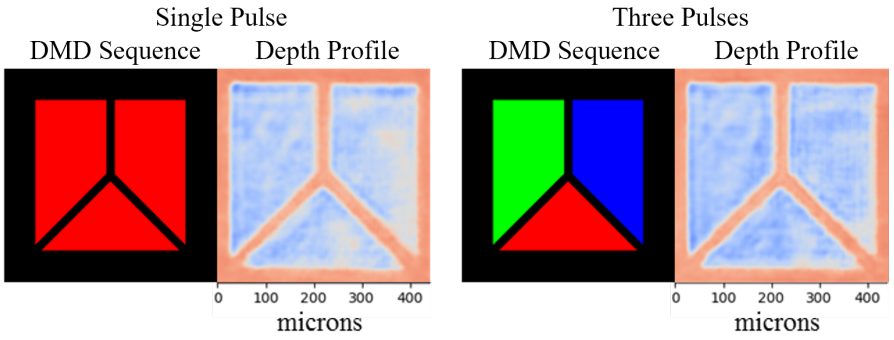
Previously experiments have also been conducted using this setup which have investigated both the diffraction limit of the system and the effect that multiple exposures would have Heath et al. (2017). This experiment uses the same 50x objective and was machined on a matching electroless nickel sample. In the work it was found that by using multiple exposures they could produce features at a factor of 2.7× smaller than the diffraction limit of the system. The predictions from the neural network could therefore be compared to the results presented in this paper.

$$d = \frac{\lambda}{2NA} \tag{5.2}$$

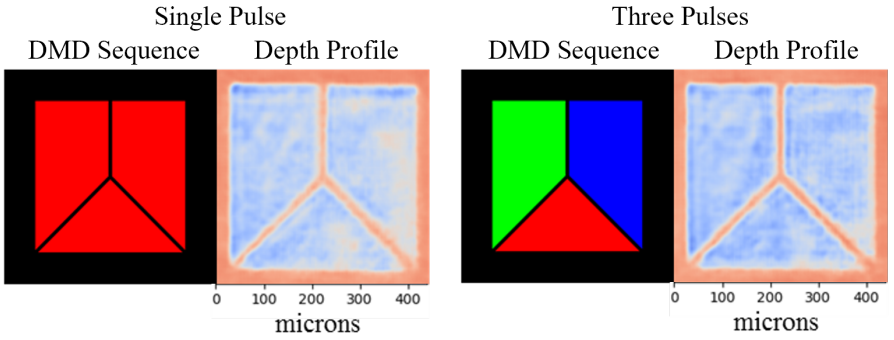
The single pulse diffraction limit (d) for the setup used for this experiment was calculated as 952 nm, using the wavelength of the laser ($\lambda = 800$ nm) and the numerical aperture of the objective ($NA = 0.42$). Features with a size smaller than this were achieved with the use of multiple overlapping pulses, and a similar effect can be simulated with two pulses machining separate, but nearby, areas and looking at the non-machined area between them.

To investigate how well the neural network was able to capture the effects of the diffraction limits, a series of patterns were used with the network to simulate their corresponding depth profiles. In each pattern there was a solid region machined across the image with three spokes from the centre left unmachined. The effect of the diffraction limit was tested by varying the width of each of these spokes, from 1 pixel all the way up to 15, corresponding to a range of 92 nm to 1380 nm. Each of these separations were tested under two conditions, firstly each was simulated using a single pulse, and secondly, each of the three regions was machined with separate pulses, each region was still only machined with a single pulse. Examples of these can be seen in Fig. 5.20 where the sequences using separations of 92 nm, 828 nm, 1104 nm are shown, along with the resultant generated profiles. As discussed previously the colouring method for the sequences of DMD patterns is the same as that described in Fig. 5.9.

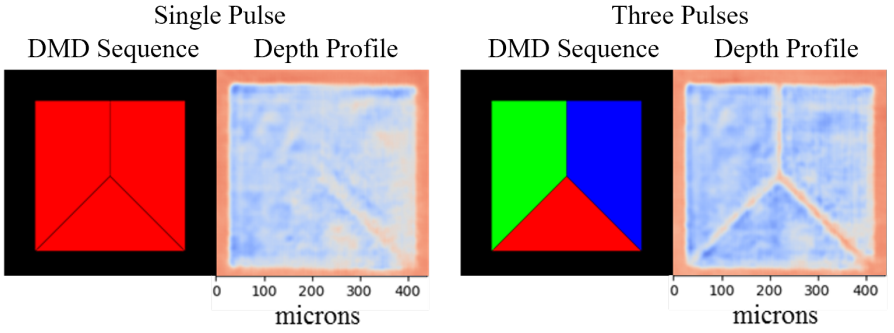
Looking at the sequence shown in Fig. 5.20a the separation is well above the calculated diffraction limit. As the un-machined feature is large, there should be very little difference between the cases where there is a single pulse, and where the



(A) Machined with a spatial separation of 1200 nm



(B) Machined with a spatial separation of 828 nm



(C) Machined with a spatial separation of 92 nm

FIGURE 5.20: Demonstrating the ability of the network to differentiate between the effects of a single pulse and multiple pulses. In the images with only red regions, the machining was conducted in a single pulse, while in the multicoloured sequence, each primary colour was machined using a separate pulse. Modified from McDonnell et al. (2020).

machining is separated. As can be seen, this is the case, with the only differences found within the machined areas.

When reducing the separation between machined areas to less than the diffraction limit, it would be expected to see a difference between the two cases. This is explored in Fig. 5.20b where the separation is 828 nm, lower than the diffraction limit of 952 nm. It can be seen here that for the multiple pulse case, the ridge between the two machined areas is narrower than in Fig. 5.20a, however, the height of the ridge is the same. In the case of machining being completed in a single pule, the same is not true,

and it can be seen that the ridge between the two machined areas is lower, indicated by the lighter colour.

The situation changes again when machining far below the diffraction, and can be seen in Fig. 5.20cc where the separation was reduced to 92 nm. In both cases it was predicted that the ridge would be heavily affected. When the machining took place in a single pulse, the ridge for the upper and lower left spokes completely disappeared. In contrast, when the multiple pulse situation, the spokes have not completely disappeared, but do show a greatly reduced height.

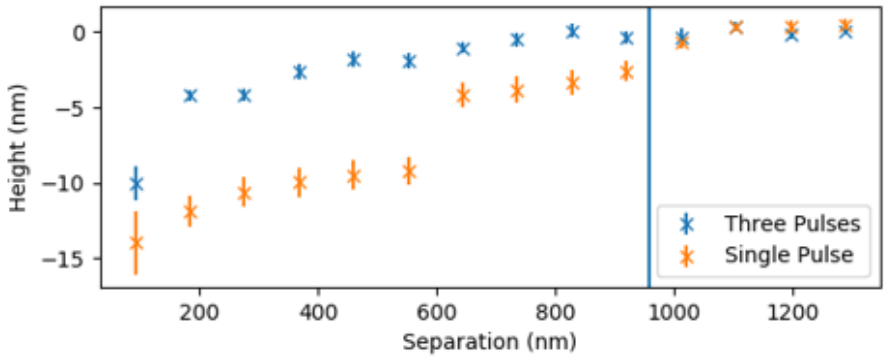


FIGURE 5.21: The effect of separation on the height of remaining non-machined area for both single and multiple pulse cases. The vertical line represents the diffraction limit of the system, from McDonnell et al. (2020).

While a trend can be seen in Fig. 5.20, a full analysis of all tested sequences can be seen in Fig. 5.21. In this figure, the height of the ridges of non-machined material has been plotted along with the diffraction limit of the system. Above the limit of 952 nm, there is no difference between the cases where the machining was done in a single or in multiple pulses. In both cases, while the width of the ridge decreases, the height of the material stays constant and matches that of the surrounding material, indicated by a depth of 0. Going below the diffraction limit, Heath et al. Heath et al. (2018b) found that features were still visible, even down to a quarter of the diffraction limit, although the maximum height of the feature was reduced. The reduction in feature height was predicted by the GAN, with the height reduced beyond the diffraction limit indicated by the vertical blue line. In contrast, the GAN predicts that when neighbouring regions are machined in different pulses the height remained constant until around half of the diffraction limit before a small drop. The height of the un-machined ridge for three pulses remains roughly equivalent to that of the single pulse case machined at twice the separation. This shows that the network has learned that the single pulse diffraction limit can be beaten with the use of multiple pulses.

5.7 Data Preparation for generating sequences of DMD Patterns

As discussed previously, the one-to-many relationship between the depth profiles and the sequences of DMD patterns used to create them, makes this transformation more complex. As such it would be expected that a method of improving network performance would be to simply collect more data. As an alternative, techniques could be implemented to reduce the amount of data required by the network while still maintaining comparable performance. This is not an insurmountable task to overcome, and indeed has been an area that has seen a lot of research (Noguchi and Harada (2019); Shaikhina and Khovanova (2017)) due to the sheer amount of data used by some of the state-of-the-art networks. Despite this, many of these approaches are very situational such as highly specific augmentation that relies on a peculiarity of the data, and so not suitable in all situations. Even when the solutions can be applied, they are aimed at reducing the amount of training images down from millions to the order of thousands Karras et al. (2020). At this scale, to collect the experimental data would still involve an unfeasibly large period of data collection, making this an unattractive option.

As techniques to reduce the amount of data would not be completely suitable, this led to another option of artificially creating more data to match the style of the data collected through the experimental process. For use with neural networks, this creation of data often takes the form of data augmentation. Augmentation can take many forms, and can include scaling, rotating and cropping. Due to the nature of the data involved, having multiple scales involved would not be suitable due to the size relationship of machined features, such as the diffraction limit discussed in the previous section. For example, if the images were scaled differently then the machined features in a more zoomed-in region would appear comparatively softer than those in a less zoomed-in area. Similar restrictions are true for cropping, where features outside the view would impact the result. While the beam used was nominally gaussian, through the beam optics an intensity slant was introduced across the profile and introducing any rotation of the images would have obscured this information, leading to less accurate results.

With this in mind, it was decided that while more data should be used to train the network collecting more data was not the most practical option. In addition to the amount of time required to collect the data itself, over the time the previous study had been carried out, the laser became non-operational. This posed the issue of not being able to collect more data that would be equivalent to that used in the previous experiment. The first option present was to set up the beamline on a new laser and repeat the entire experiment including data collection, network training and

validation. All of this would take a lot of time on top of that required for collecting a larger dataset.

With the excessive time required to recreate the experiment, and the limited ability to use data augmentation, an alternative, novel option was chosen. In the previous section, it was established that it was possible to create a generative network that was able to simulate the experiment. Therefore, it was decided that this pretrained and proven network would be used to artificially create an entirely new dataset while maintaining compatibility with the true experimental results. This allowed for the very quick creation of a dataset containing 5000 pairs of DMD pattern sequences and depth profiles, taking about 20 seconds to produce. This provided a far larger dataset than the original 200 data pairs. This was used in place of regular data augmentation which would have the problems described previously. Using the trained network, however, new data could be created that was in effect an interpolation of the original data and contained the same features and distribution while allowing for less chance of overfitting. While the network was proven to be capable, it was decided that this should form the training dataset, while all experimental data would be kept in the validation dataset. This includes all the structured data, such as grids and letters, designed as validation for the initial network, as well as all the random patterns used in both the training and validation datasets.

5.8 Verifying the Validity of the Network

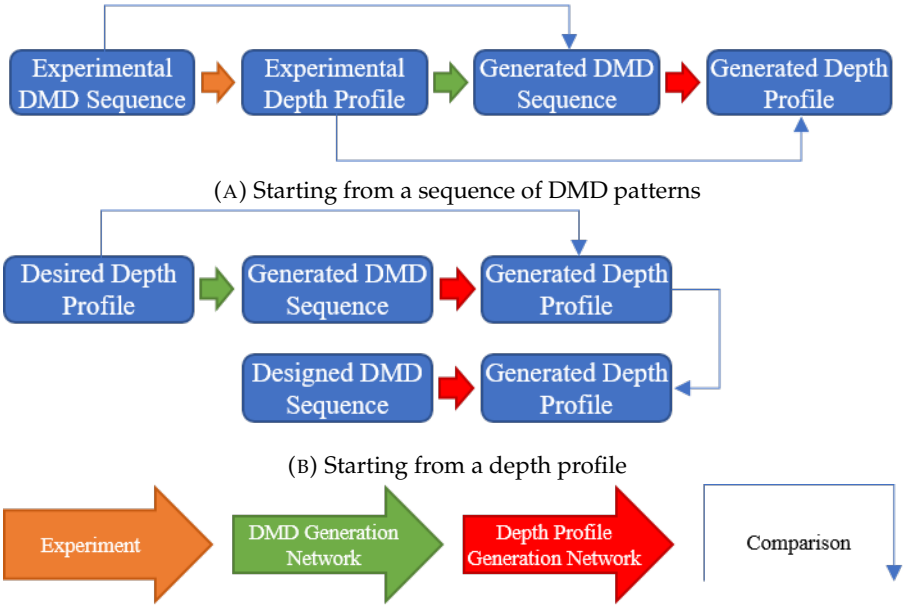


FIGURE 5.22: The process for creating and testing the trained, NN generated, DMD profiles.

Before looking deeply into the capability of the network to generate sequences of DMD patterns, the network was analysed using the experimental data that was not included in the training. As discussed previously, the important metric to assess the capability of the network is how well the generated sequence DMD patterns would lead to the initial depth profile. In Fig. 5.22 an example is shown of the full generation, prediction, and validation process. The first shown step in testing the network is the initial sequence of DMD patterns created from the random combination of lines, arcs, and circles. The next step shown is the measured depth profile captured using the white light interferometer after the laser machining took. This depth profile was then fed into the new neural network to produce a prediction of a sequence of DMD patterns that would have produced that depth profile. Finally, in order to measure the critical performance of the network, the generated sequence of DMD patterns is used along with the previously demonstrated neural network to produce a newly predicted depth profile.

This process allows a comparison both between the sequences of DMD patterns that are the actual output from this network, and the depth profiles that result from those. The sequences of DMD patterns are relatively simple binary masks, albeit in three layers, and are therefore quite easy to comprehend. This allows for very quick inspection by humans to be able to tell how close the generated sequence is to the initial one. Ultimately, while this is usually a good metric for neural network performance, the differences between the DMD patterns could be misleading with regard to the performance of the network. This is especially true when considering the ordering of machined locations. The second comparison, between the resultant depth profiles, represents the true goal of the network, and so will form the basis of all quantitative analysis performed on the network.

Fig. 5.23 shows an example of a sequence of DMD patterns which were generated by the neural network. In this example, a further depth profile has been generated from the sequence of DMD patterns that was itself created using the neural network. This allows for both the sequence produced to be compared to the ground truth, as well as the final result of machining both of them. The aspects of the DMD patterns that can be assessed are the overall, combined shapes of the sequence, where in the pulse order machining took place, and the number of pulses used to machine at each location. The final depth profile that would be machined using the sequence produced is shown alongside the experimental version to allow for easy visual comparison. While the generated DMD sequences could not be tested using the laser, simulations were performed using the neural network.

While all of these aspects are important, and very visually obvious to a human observer, the true determination of success is in the depth profile those sequences would produce, and this is the process that the network is attempting to optimise. This newly generated depth profile has been compared to the experimental one with

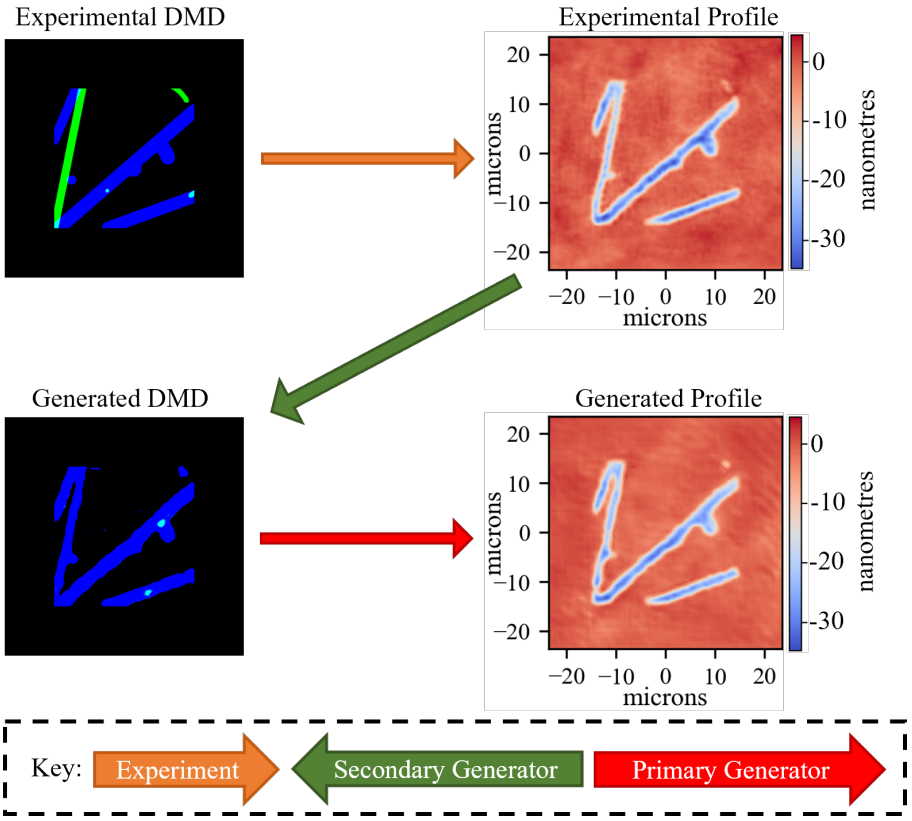
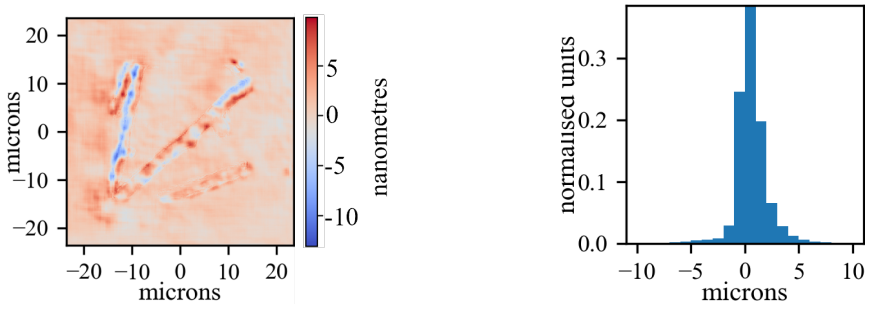


FIGURE 5.23: An example of the full process from initial DMD, Experimental depth profile, generated sequence of DMD patterns, and finally a depth profile.



(A) Pixelwise difference (B) Histogram of differences

FIGURE 5.24: Comparing an experimental profile to one predicted from a generated sequence of DMD patterns. Most of the differences are centred around 0 with almost all being in the range of $\pm$ 5 µm.

both a pixel-wise comparison shown in Fig. 5.24a combined with a histogram of depth differences shown in fig:Shape:Various:Hist. Due to the previously discussed relationship between the sequences of DMD patterns and depth profiles, the final result will never match the original experimental profile, the same being true of even two experimental profiles. While an exact match cannot be attained, the similarity of the result can be assessed using several metrics including the depth at each position, the total overall removed material, and the sharpness of machined and un-machined features.

To get a measure of the error in the network, it was decided to use the error from the generated depth profile when compared to the original corresponding experimental depth profile. To do this the average error was calculated across the entire validation dataset for this network, which included all of the experimental data collected for the previously discussed network. This allowed for a true comparison, while removing some potential for a network bias that would skew the results. When taking a comparison of all of the depth profiles, there was an average error of 1.23 nm, which reduced to only 1.01 nm when only including sequences of DMD patterns only using a single pulse.

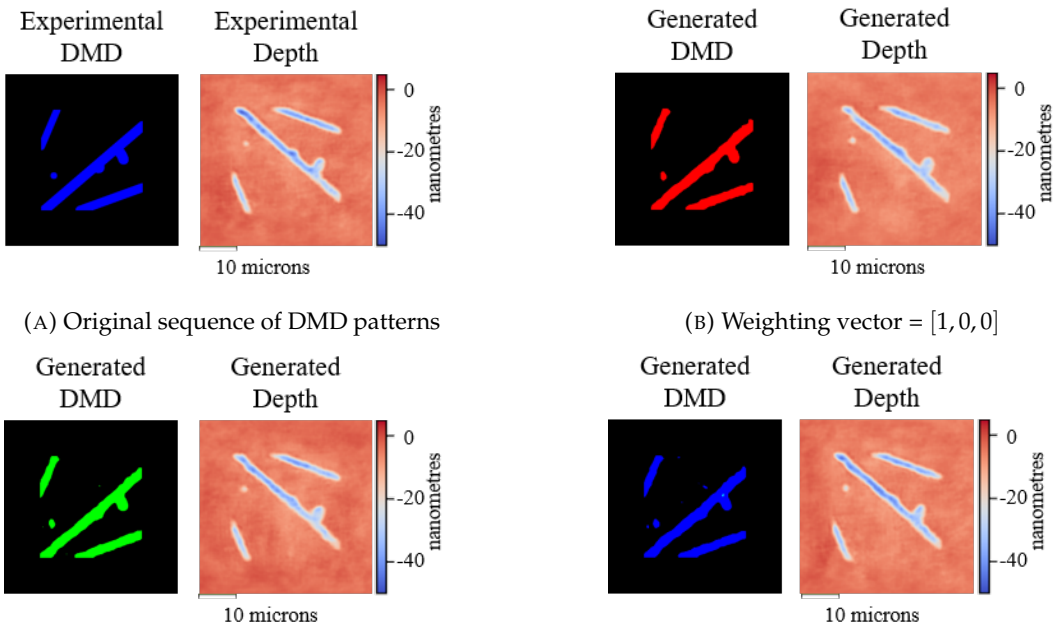
5.9 Examination of the network

5.9.1 Controlling the weighting of pulses

As stated, one of the important features to highlight in this work was the ability to use neural networks to predict situations that were more complex than the use of a single pulse. With the use of up to three pulses in this particular network, the weighting vector discussed in Section 5.5.1 gave a powerful tool to investigate the network. It could be used to determine how the network was able to capture how the choice of when to machine would affect the sequence of DMD patterns required. Each value in the weighting vector represented the proportion of the machining that would take place in the first, second, and third pulses respectively, with the total of all three values summing to one. As an example, if the weighting vector $[1, 0, 0]$ was given, then all machining should ideally take place in the initial pulse. If however a weighting vector of $\left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$ was provided then the machining should be as evenly spread as possible.

The weighting vector acts more as a style guideline to the network rather than a strict rule; if it is not possible to machine the shape found in the way the weighting vector suggests it will be overruled. For example, if an area had a depth typically associated with that found after three pulses in one location, but a weighting vector only allowing a single pulse was given, the network would still use all three pulses. Fig. 5.25 demonstrates this effect, where for each generated profile, a different weighting vector has been given for the same experimental depth profile. The vectors for Fig. 5.25b, c, and d were given as $[1, 0, 0]$, $[0, 1, 0]$, and $[0, 0, 1]$ respectively. For each of the generated sequences it can be seen that almost all of the machining took place in the pulse specified in the weighting vector. In fact, the only sequence that includes "on" pixels in pulses other than the one specified by the weighting is Fig. 5.25d where there is some machining shown in the second pulse.

While this was a very simple example, it could be the case that multiple pulses would be required, and the weighting vector will be more complex. While it may be



(A) Original sequence of DMD patterns

(B) Weighting vector = $[1, 0, 0]$

(C) Weighting vector = $[0, 1, 0]$

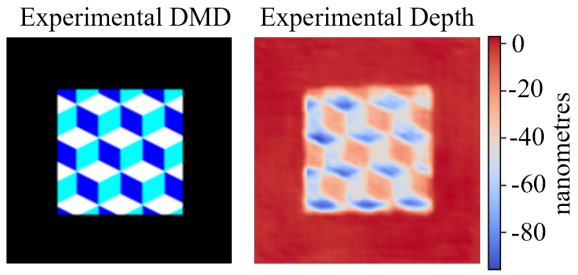
(D) Weighting vector = $[0, 0, 1]$

FIGURE 5.25: Using the weighting input of the network to control the temporal position of white pixels in the sequence of DMD patterns. Modified from McDonnell et al. (2021b).

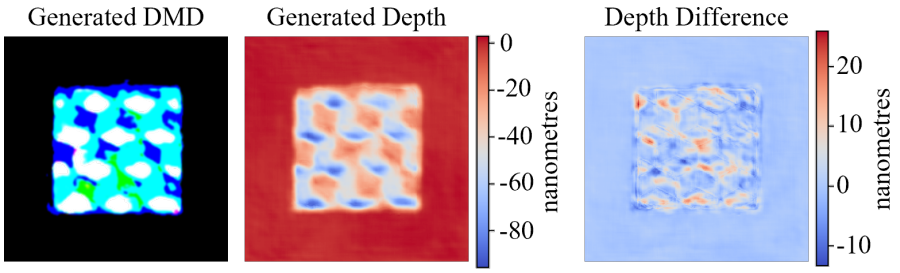
beneficial to reduce the number of pulses used to decrease total machining time, due to the desired structure of the depth profile, this may not be possible. In these cases, the network will incorporate the restrictions imposed by the weighting vector while still producing a viable sequence of DMD patterns. This is explored in Fig. 5.26 where a depth profile that requires three pulses is combined with a variety of weighting vectors to test the resilience of the network.

To provide a baseline for the comparisons, a sequence of DMD patterns was generated using a weighting vector of $[\frac{3}{6}, \frac{2}{6}, \frac{1}{6}]$, which matched the pixel distribution in the original sequence shown in Fig. 5.26a. The generated sequence can be seen in Fig. 5.26b where the colour palette predominantly matches that of the original. Looking at the predicted depth profile that would result from the generated sequence, the overall structure of a cube effect is still present, although there is some distortion present, especially in the central area. To investigate more explicitly, the difference between the predicted and experimental depth profiles is shown on a pixel by pixel bases in the "Depth Difference" image. This agrees with the observation that most of the difference is in the centre of the image, as well as some difference around the edge for a total MAE of 1.37 nm.

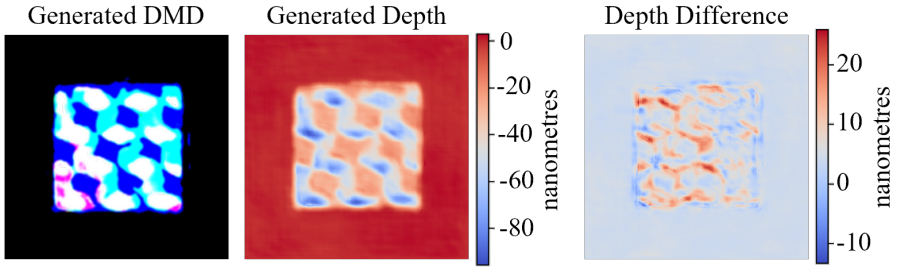
As most of the machining already took place in the third pulse, the next test was conducted using a weighing vector of $[1, 0, 0]$ as shown in Fig. 5.26c. This was chosen as the network would be encouraged to produce a sequence of patterns close to that of the original. Again, the network has managed to produce a sequence of patterns that



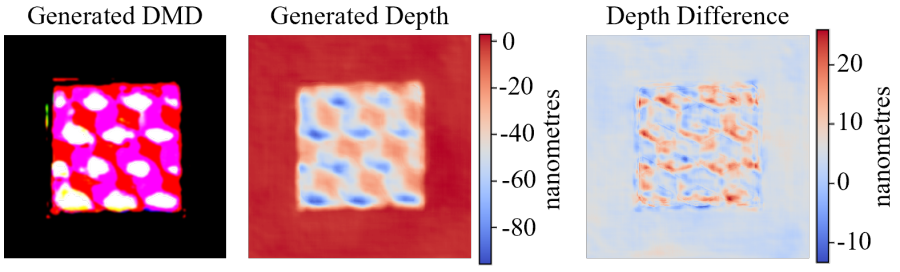
(A) Original sequence of DMD patterns and the resultant depth profile



(B) Sequence of DMD patterns generated with a weighting vector of $[\frac{3}{6}, \frac{2}{6}, \frac{1}{6}]$



(C) Sequence of DMD patterns generated with a weighting vector of $[1, 0, 0]$



(D) Sequence of DMD patterns generated with a weighting vector of $[0, 0, 1]$

FIGURE 5.26: Demonstration of the effect of the weighting vector on a profile requiring three pulses. Modified from McDonnell et al. (2021b).

is visually similar to that requested. To do this the network has correctly predicted that to produce the required depth profile, the full three pulses would be required and so has utilised all three channels. One of the differences to the results shown in Fig. 5.26b is that where only a single pulse is required, only the blue channel is used, which is closer to the original sequence. Visually the generated depth profile is again similar to the previous example, with distortion on a central patch. Despite this, the MAE for the depth profile is actually higher at 1.86 nm.

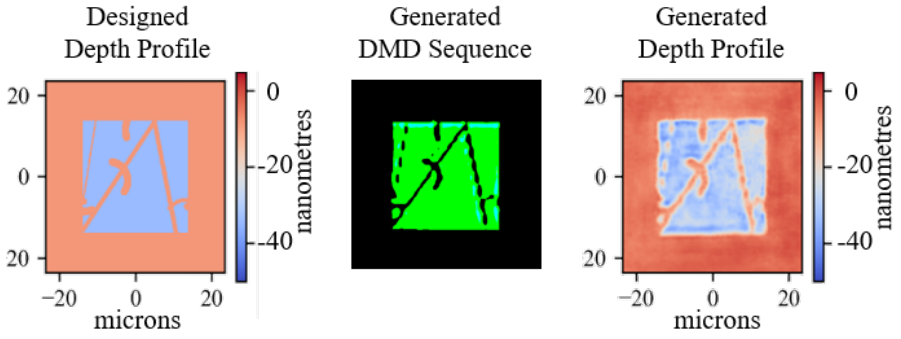
The final test conducted was to use an inverse weighting vector, using $[0, 0, 1]$, to promote the use of pulses in the red channel. The first observation, seen in Fig. 5.26d, is that the sequence of DMD patterns has a very different colour bias to that shown previously. Similarly to Fig. 5.26c, wherever only a single pulse was perceived as necessary, only a single channel was used, in this case, channel 1, matching the request in the weighting vector. Despite this, the next most populated pulse was the blue channel. This appears to be in a direct attempt to match the sharpness characteristics of the initial depth profile and has still managed to maintain an MAE of 1.88 nm.

In the original sequence, the coarse features were machined first, leading to a sharp depth profile as described in Section 5.6.1. This property was matched with Fig. 5.26c where the network was prompted to make the highest use of the initial pulse allowing it to again machine coarse features predominantly in earlier pulses. This paradigm is changed in Fig. 5.26d however, where the network has been instructed to prioritise the third pulse. While this was indeed the case, the network was not given any instructions regarding the first and second pulses, allowing it to determine an ideal order. In an attempt to maintain the sharpness of the depth profile, the network has prioritised the coarsest reaming features in the first pulse, with the finest features in pulse two.

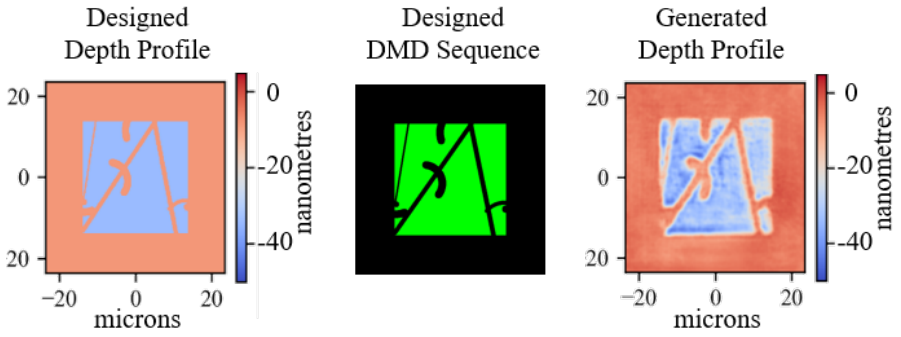
5.9.2 Comparing designed and generated DMD pattern sequences

Apart from simply generating a sequence of DMD patterns from a measure depth profile, one could also be generated from the desired depth profile as shown in Fig. 5.22b. An example of this process is seen in Fig. 5.27 where a simple square-edged depth profile (similar to that shown in Fig. 5.2) has been designed. This profile is then passed into the network using a weighting vector of $[0, 1, 0]$, suggesting the machining all be completed with a single pulse. The resultant sequence from the network was then used with the previously demonstrated network to simulate the depth profile that was produced from it, which can then be compared to the desired depth profile, this process is shown in Fig. 5.27a.

Comparing the final generated depth profile to the initially designed one, there are several similarities as well as a few differences. While both the surface and machined areas of the designed profiles are perfectly flat, this is not true in the generated profile. For the surrounding un-machined material this will be due to the simulation of machining a real surface, and so is unavoidable, the perfectly flat surface will not be possible under these conditions. For the material in a depression, the generated profile is also not perfectly flat. This again matches experimental results and is due to a variety of effects such as the small variations in laser power. Beyond the simple non-flatness, another observation to make is that the neural network has produced a sequence of DMD patterns that contains machining in more than one step. This can be



(A) Network Generated DMD Sequence



(B) Directly converted DMD Sequence

FIGURE 5.27: Generation of sequences of DMD patterns to machine a designed depth profile compared against a naively designed DMD pattern that could be used to produce the desired output.

seen with the coloured band along the edge of some unmachined parts of the depth profile.

To give a comparison to work against, a simple sequence of DMD patterns was created from the final depth profile, before also being used with the depth profile generation network to produce a new depth profile as shown in Fig. 5.27b. Similarly to the case with the network-generated sequence, the final depth profile has differences from the designed profile, most notably in the flat areas. Again this is an artefact that cannot be solved using the current setup and implementation. There are however two major differences between the two sets of DMD patterns, the first being the number of pulses used. As the depth profile has a uniform depth, the converted set of patterns only contains machining in a single pulse.

The second difference present is the shape of the features, with the converted sequence containing much smoother lines, and wider features. One impact of that is the two more vertical straight lines in the depth profile have a cleaner rendering in the depth profile generated from the converted sequence. Despite this, the un-machined ridges are noticeably thicker in the depth profile generated from the converted sequence, than in the one generated from the generated sequence, which shows a

much closer correlation to the base, designed depth profile. This can also be seen in the lower left of the depth profile, where there is a small machined area in the designed depth profile. This still exists in the depth profile in Fig. 5.27a, while on the depth profile in Fig. 5.27b that area is all at the level of the surrounding material. Taking an overall comparison of each generated depth profile against the designed one, the generated sequence of DMD patterns produces a depth profile with an MAE of 4.8 nm. This error is lower than the MAE of 5.1 nm produced when looking at the depth profile generated from the directly converted sequence.

5.10 Conclusions

Here it was shown that generative networks can be used to predict the outcomes of highly complex machining tasks, producing results that match the look and values found in the experimental data. The network designed to predict the results of laser machining was able to predict the effects of non-linear effects such as diffraction limiting and pulse ordering without any specific instruction to do so. The network build to create the sequences of patterns required for machining was able to produce results that would accurately reproduce the desired depth profiles while having never been trained on any experimental data, providing a closed-loop experiment and showing one potential application in the use of laser machining.

Based on the accuracy of the results presented here, it is anticipated that the result of laser machining with a much larger number of pulses could similarly be predicted by a neural network, provided that appropriate training data was provided. The inclusion of additional pulses may also require modifications to the neural network architecture, for example increasing the number of parameters. The additional physical complexity of exposing an already-machined surface strengthens the previous results. Here, it was shown that this approach offers a template for modelling processes whose underlying principles are too complicated to model accurately or that are entirely unknown.

Chapter 6

Using Generative Networks to Simulate Unknown Machining Processes

In Chapter 4 the concept of optimisation was explored, and several machine learning techniques were investigated. Amongst those, the option of using generative networks was discussed briefly, showing its strengths and weaknesses. This Chapter will be exploring processes where generative networks can be used and their strengths best utilised. Previously there have been discussions about the impact of human bias on the results of the ML processes and how this may affect the apparent validity of the results, both positively and negatively. To examine both of these points, an entire experimental process was simulated without knowledge of the system or the parameters used, making use of anonymised and randomised inputs. This demonstrates how Neural networks can be used to find information from closed systems and aid in experimentation, investigation, and understanding. This is an ongoing piece of work with further progress continuing to be made.

For the data used in this Chapter, data was collected by the technical staff at Oxford Lasers, with many of the details intentionally hidden to ensure that any conclusions found are the result of the network rather than human interpretation or expectation.

The experiment consisted of machining vias in a substrate using a set laser which had several parameters that could be tweaked in order to change the results of the machining. During this machining process, the intention was to produce a hole passing completely through the material. Despite the idea to train the network without any human bias, the Oxford Lasers technical staff made the decision to ensure that the bounds of the parameters were limited such that all results should result in a clean via. While this would add a small measure of bias, there were no excluded

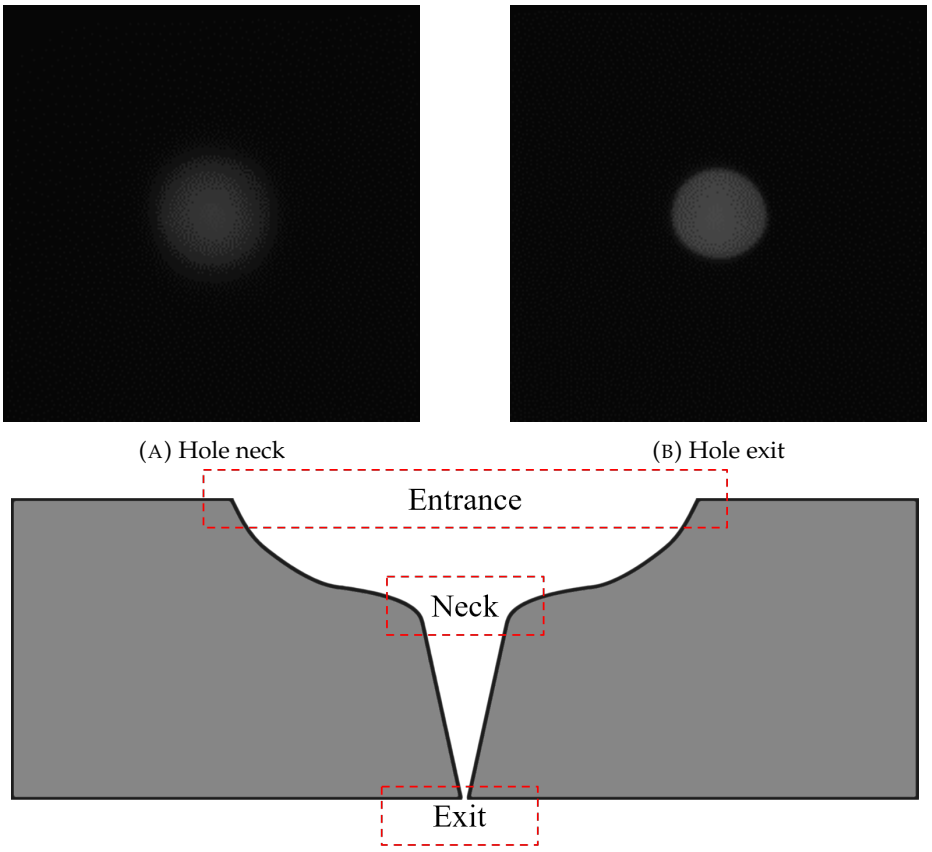
combinations, and each parameter would have an approximately uniform distribution.

6.1 Experimental Data

Within this experiment, the data was formed from strict pairs of vector inputs and image outputs. For the input, there were up to 15 parameters used each being provided in a uniform normalised range from 0 to 1. These parameters represented the laser parameters used to machine the sample at each location. Throughout the process, the number of provided inputs was varied to scale the complexity, with an initial test with 5 parameters conducted before a total of 9 being used in the final experiment presented here. Each of the 5 parameters used in the initial investigation were later used when testing with more parameters in use. The full set of 9 parameters tested were labelled with letters (from a to i) with letters a through to e being the initial 5 parameters that were used in both of the experimental runs. This was a choice made to increase the opportunities for interrogation and comparison of the neural network, especially on the impact of increased numbers of parameters. These variables were all anonymised and their meaning was kept hidden from both us and the neural networks. While the results were normalised, they were still not in an optimal format for training a neural network. To avoid mode collapse or vanishing gradients the variables were all remapped to have a mean of 0 and a standard deviation of 1, matching the initialisation of the weights within the network.

6.1.1 Laser Parameters

The other part of the data used for this investigation was formed of images of the holes machined by the laser, shown in Fig. 6.1. The images were captured on a microscope using a backlight behind the sample from the perspective of the camera. This produced images that were dark, tending to black, except for the location of the hole which was lit with white light. Each data point was supported by two images, one taken from each side of the sample, and in total 2000 combinations were machined. For each set of machining parameters, a set of 20 holes were machined in a grid format, providing repeated samples for variance. These sets were imaged in a single exposure, capturing all of the holes in one large image rather than each hole individually. As the machining was designed to produce through holes, with both sides being of interest, there were two sets of images, one from each side. The part of the hole visible from the side of laser incidence is referred to as the neck of the hole, and examples of these can be seen in Fig. 6.1a, the other side being the exit, shown in Fig. 6.1b.



(A) Hole neck (B) Hole exit

(C) An exaggerated diagram of the areas of interest

FIGURE 6.1: Example output data images pair. The neck image is taken from the side of laser incidence, focused on the inner direction change within the hole, leading to a slightly blurred image. The exit image is taken from the opposite side, focussed on the surface of the material.

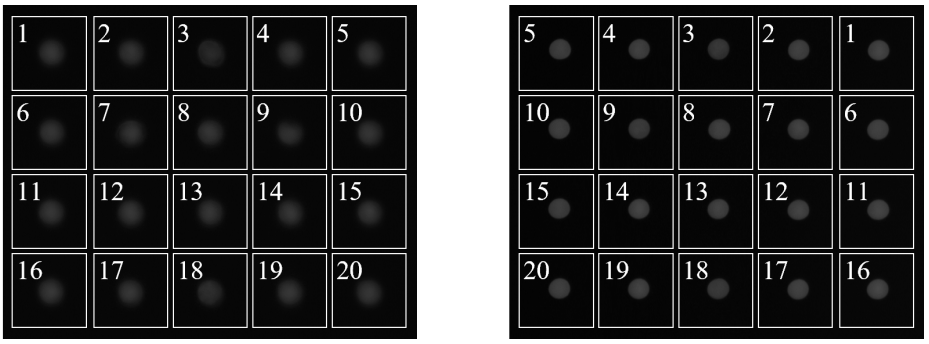
In the initial set of data presented by Oxford Lasers, their standard data processing techniques had been used to create the dataset provided to us. This process involved using a fixed range on the camera, and the default, JPEG, saving format. The fixed range of the data led to two issues that could hinder the performance of the network. The first of these issues was that the chosen range was a highly compressed version of the full dataset available to use in standard image encoding. In greyscale image processing, the overwhelming standard is the uint8 data format, providing data in a range of 0 to 255. The data received from Oxford Lasers only contained a range of 0 to 34, indicating that some amount of pre or post-processing had taken place. The low upper value of the data suggested that the exposure setting on the camera had been set too low or calibrated for too bright a light. These values were likely set on an empty stage where there was the maximum amount of light reaching the sensor. This would normally be a sensible process, except that in this case, the holes used are each small when compared to the field of view, meaning that a lot of light was lost. This reduced upper bound meant that large portions of the information were lost due to the compression of the raw light levels into the uint8 format. This compression was

exaggerated by the small range of values use, making the effect far worse. While this is not just a limitation applied to the neural, but also the standard inference techniques, bad data can lead to a poorly trained network.

The existence of 0 values in the data could indicate that a good lower bound for the exposure was set, although in the data provided it is the prevalence of 0 values that was concerning. The vast majority of all unlit areas were completely black, lacking any evidence of camera noise within those areas. This artefact suggested that a background reduction operation had taken place on the data, although the source of this was unknown. The problem with this was twofold, the first being that with the background reduction, information in the data was lost. The loss of the noise information would likely not have caused any issues and increased the performance of the network. Rather than the loss of the noise itself, the issue with this was that the degree of background reduction was unknown, possibly removing useful information from the data, especially in cases where lots of information was contained within that low-level data.

The second issue with the background reduction was that, without knowing the process used, it couldn't be guaranteed that the operation was performed uniformly. If different values of the background subtraction were used on two similar images, this could lead to the values calculated using the output of the network to be incorrect, in the same way as using the raw data. Beyond this, an uneven background subtraction could easily increase the one-to-many nature of the dataset, increasing the learning difficulty. It is also likely that images that contain large openings would also contain more light bleed through them, illuminating the surrounding area. Given constant camera settings, there should be an equal amount of noise present in all images, however, if an automatic noise reduction algorithm has been used, it will likely be more aggressive in the lower light images as the noise will form a greater portion of the background. If, however, a background reduction has been used, which is more likely due to the solid black region, images with softer lighting will be more affected. When the noise in the image is the main part of the background, the background reduction will mostly just remove the noise from the image. In contrast, when there is a high level of background light, there will be a more aggressive reduction, causing a larger loss of information.

The fact that sets of holes for a single parameter combination were captured in a single image led to several data processing steps. Firstly the overall image location was not perfectly set, meaning that the locations of the holes must be found. This process was conducted by Ben Mills, and followed a similar process as that described in Chapter 4. Rather than using the centre of mass of the image, as described there, a grid of weights was used instead. This was possible since all holes had an equal and consistent spacing between machining and capturing steps. There were two main issues encountered in this step, the first being when there is very little breakthrough in the



(A) Neck image numbering (B) Exit image numbering

FIGURE 6.2: How all of the holes were numbered after saving. These were then used to match up the neck and exit images to use as data pairs when training the network. All images here relate to a single set of laser parameters used and are not representative of the full range found within the data.

machining process. This, combined with the limited dynamic range, led to some holes not being centred in their desired locations. The second issue arose when the image was very misaligned and in the area of interest for each hole, there was an overlap. When this occurred, it was often found that the image would be marked as aligned while having partial sections of multiple holes in the corners. As there were only a small number of cases where the initial alignment was off enough that the holes were centred in the corners, a rough manual alignment was performed initially, before running the full alignment algorithm, fixing the issues.

Once the holes were centred on the grid, the overall image was then cropped to form a set of 20 images, each containing a hole that should be centred. These images were then numbered based on the position of the cropping, with the numbering order specific to each of the neck and exit images. The numbering had to be unique for each as the substrate was flipped between imaging, introducing a plane of symmetry, and the overall numbering for each is shown in Fig. 6.2. The cropped and saved image files use a prefix of the number shown in the figure with the upper left hole in Fig. 6.2a and the upper right hole in Fig. 6.2a both having a suffix of 1 in the file name and both representing different sides of the same machined hole. Once cropped, the various holes in each parameter combination were compared to each other, with the maximum values for each hole showing a common trend across combinations. Rather than having a consistent maximum value across all of the cropped images, it was found that the central holes had a consistently higher brightness than those found at the edge of the large image. This effect was likely caused by a natural difference in the lighting in the imaging rig, with the bulb causing a bright spot in the centre of the image, although could also have been an effect of vignetting in the camera itself. This question was posed to the staff at Oxford lasers, and while a definitive answer was not obtained, the theory of a bulb bright spot was ruled out as the entire sample was mounted on an illuminated jig. While this is a real effect of the image processing, and

would not have a large impact on the calculation of values from the images, it would introduce a new factor for the network to learn, increasing the training time. Rather than include this artefact in the final images, a fit function of the lighting was calculated across the entire dataset and the images were scaled accordingly. This method worked well due to the low value background providing a baseline to scale from. An alternative method would have been to ask for an image of the pure lighting with no sample to create a full profile of the lighting, but this approach was decided against due to the aim of the project to prove the capability of networks to perform a task without large prior knowledge.

While the aim was to stay separated from the data collection process to avoid introducing accidental bias, there were clear improvements that could be made to the data and so there was a decision point about requesting changes to the data capturing methodology. Despite the desire to be uninvolved, it was decided that having better data could also be useful, both for training the neural network and to provide a demonstration point. The other factor at play is that no more data processing was requested but rather less. The three main areas of change requested were the background noise reduction, the image data format, and the use of the dynamic range. While the original data was provided in JPEG format, this is a very lossy compression technique, and even though it results in smaller images, there are many artefacts introduced in the process. In order to remove this, a lossless compression method was requested as this would still cause image sizes to be reduced, but would not introduce artificial items in the images for the network to learn. Another area of change was the background noise reduction process, and again it was requested that nothing be done for this, and all automatic processing be turned off. This allowed us to receive a much purer form of the images that could then be processed in a more appropriate way for the neural network. The final request made was to increase the dynamic range of the supplied images, making better use of the 255 light levels available in the uint8 data format. While the data will always be converted to float values within the network, having a larger initial range means that once the data was normalised there was more granularity and therefore more opportunities for the network to learn features slowly and smoothly.

6.1.2 Analysing the images

With the plan of this experiment being to use a GAN to simulate the experiment came the requirement to set up a method of testing the performance of the network. In the previous work with Oxford Lasers (Chapter 4) the experiment aimed to calculate two properties of the machined dimples in order to perform an optimisation task. These properties could be calculated directly from the experimental data provided, and the same could be done for the GAN-generated images. While a specific task was not

(A) Original (B) After Thresholding (C) After Quartering

FIGURE 6.3: Image processing steps to calculate network performance. The image is first thresholded to provide a clear boundary to the hole to use to calculate the area. The image is then quartered to calculate the variance in the radius of the holes.

provided for this experiment, the same process could be used, with properties of the holes calculated from both the experimental data and any future generated data. The image properties chosen to form the basis of the investigations were the radius and circularity of each hole. An example of the processing steps is shown in Fig. 6.3, with the base image shown in Fig. 6.3a.

In order to ensure that consistency was maintained between all data calculations, a script was set up to perform each of these calculations. This script was written in python to allow for easy integration with the machine learning process and simple comprehension. The first step in both of these processes was to threshold the images using a value of 50% of the maximum value, with values above set to one, and values below set to 0, shown in Fig. 6.3b. This was deemed suitable as the images were clean with no anomalies detected beyond the observed hole. This meant that it would be very likely that any pixels with a final value of one were initially part of the hole.

The next step for both of the calculations was to split each of the images into 4 along both centre lines, creating four quartered circles, each 128×128 pixels, as shown in Fig. 6.3c. The average pixel radius for each of these quadrants was then determined using Eq. 6.1 where n is the sum of all pixels in the sub image, or the number of pixels above the threshold. Note that all values here were calculated in terms of pixels as no units of scale were provided.

$$r = 2 \cdot \sqrt{\frac{n}{\pi}} \tag{6.1}$$

The final step in calculating the radius of the hole was to simply take the mean of the four radii calculated from the image. In a similar manner to this, the circularity, or regularity, was calculated by taking the standard deviation of these same four values. This method does have flaws, such as potentially giving non-circular shapes a good, low, value for the circularity. While this is a detriment, the methods of calculating the radius and circularity were chosen based on several criteria. Firstly, the methods used

had to be quick. Training a GAN can be a long and involved process and these calculations may be required many thousands of times. A second requirement was that the algorithm should be easily describable and recreatable if it was determined that the results found should be confirmed while not having access to the original code. Because of this requirement, it was decided that no opaque algorithms should be used as, even if a standard library was used, the same algorithm in a different library or language may have a subtly different implimentation, making it potentially hard to diagnose differences.

6.2 Network Architecture

As the experiment that is described within this chapter was focused on the creation of images that are both accurate and had high visual fidelity, using a GAN was a good fit for the task. Based on the success of the networks demonstrated in Chapters 2 and 4 a similar structure was chosen for this network with small variations to account for the differences in input and output data structures. While the data does share a lot of similarities in structure to the regression task from Chapter 4 some of the advances in neural network techniques, such as better normalisation techniques, were incorporated from the later network demonstrated in Chapter 5.

The inputs to the network were purely numeric with the laser parameters formed into a vector. As mentioned previously each of the laser parameters had been provided as a normalised value falling between 0 and 1. In addition to this, each of the values were selected randomly from within this range using a normal distribution. While not all the values appeared to be continuous, the generation condition of range and uniformity on the randomness remained true, with no weighting as seen in the number of pulses parameter from Chapter 4. This meant that it was assumed that each of the parameters would have a mean value of 0.5 and a standard deviation of $\frac{1}{\sqrt{12}}$. The data was then scaled accordingly to match the mean of 0 and standard deviation of 1. While this is not strictly necessary it is common practice and allows for easy transference of techniques and pretrained parameters between networks if there were all designed with a standardised input.

One of the key features of the provided dataset was that it was not just implicitly one-to-many, but explicitly so as each of the combinations of parameters had been used to machine a set of 20 holes. To ensure that this condition was maintained, a secondary random noise vector input was used alongside the provided parameters. It was decided to use a vector of size 100 to provide the random element into the network.

While the number of values in the input vector would change throughout the different stages of the experiment, the output of the network would remain constant. The

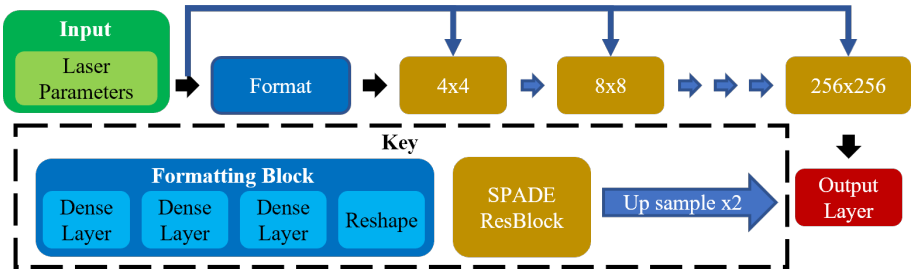


FIGURE 6.4: A block diagram of the generator network in the GAN. The input is first formatted and then put through a series of SPADE ResBlock and upsampling layers, using the original input as a secondary input at each stage.

outputs were the pairs of images extracted from the grids provided by Oxford Lasers, each of the individual holes isolated from the original 20. These images were processed as described previously and concatenated to provide an output tensor of size 256×256×2. As this data represents image data rather than the height data of previously discussed experiments the output format could be more constrained. The original images were in the uint8 data format and so were converted to be floats within the range of 0 to 1 by dividing the values by 255. This conversion allowed for the sigmoid activation function to be used, providing hard limits on the final values to allow for easy conversion back to images. The sigmoid activation function highlighted the importance of ensuring that as much of the image range as possible should be used to avoid areas of very steep gradients found at the extremities of the function.

The final structure of the generator used in this experiment was again based on the SPADE network proposed by Park et. al. Park et al. (2019) with some variations that can be seen in Fig. 6.4. There were two major differences in how this network was structured compared to those used in the original paper, and in previous experiments conducted within the group. One of these differences in the input into the SPADE normalisation layers, which in the original paper was the base image used for translation. Rather than this, the same approach was taken as that described in Section 5.5.2, where the vectorised parameter input was first concatenated with the noise. This larger vector then passed through 3 Dense layers and then reshaped to form a tensor of size 4×4×1024. This input then formed the image portion of the network, being upscaled throughout using the Nearest Neighbour interpolation algorithm as described in Section 3.4. The other input to each of the SPADE ResBlocks was based on the original, combined, vector input passed directly to each of the SPADE ResBlocks.

The other major difference from the previous networks used was in the final layers of the network. The network used for this experiment was designed to output the two images, the neck and the exit. During initial testing it was found that, at most, only one of the images would be gaining visual fidelity at any one time, the other seeming to contain a ghost of the other, and in some cases neither would be clear. To rectify

this, a decision was made to truncate the network before the final layer and instead add a separate output layer for each desired image. Each of these was then treated as a separate network for training purposes, although shared a common base network. This meant that while the training time for each epoch was increased, the memory required for the network was negligibly affected due to the training of final layers updating on a reference of the same base network. Even though each epoch would take twice as long, total training time would not, as the majority of layers were still updated in each pass.

The first of the SPADE ResBlocks acted on the 4×4 output from the mapping layer and has the same structure as that shown in Fig. 5.10. In total there were 5 Spade blocks, with all except the final one being followed by the Nearest Neighbour upscaling blocks. The first and second SPADE ResBlocks used Convolution layers with 1024 filters, with each subsequent reducing the number of filters by a factor of 2, with the final block taking 64 filters. The final step for each of the separate networks was a Convolution layer with a kernel size of 1, 1 filter, and a sigmoid activation function as described earlier. The overall structure of the network remained consistent when testing with different numbers of input parameters. The only change was to adjust the size of the noise input as this allowed for each layer to have a consistently sized input. This was important as it allowed for maximum compatibility when it came to the possibility of using the pre-trained network on new samples before conducting transfer learning.

To complete the GAN structure, a discriminator was required to contribute to the losses and training of the generator. As there were effectively 2 generators, producing two different types of images, two discriminators were required to assess each of the images. Each of the two discriminators used here had the same network structure as each other. The discriminators had two inputs, one being the images, either collected experimentally or created using the generator network. The other input to the network was the vectorised laser parameters used to machine the holes. The image-based input was passed through 4 Convolution layers with a kernel size of 3 and stride of 2 to reduce the size of the layers by 2 each time. After the Convolution layers, the image path was then reshaped to form a vector. The other path was based on the vector input passed through 2 Dense layers, the output of which was then concatenated with the flattened output of the image path before a final Dense layer with a sigmoid activation layer.

To train each of the generators a combination of hinge loss and MSE was used, with a relative weighting ratio of 100:1 and the structure of these losses is shown in Fig. 5.12. These losses were used in combination with an Adam optimiser with a beta_1 of 0.9, a beta_2 of 0.999, and a learning rate of 2×10^{-5}. The discriminator was trained using the same hinge loss and an Adam optimiser with the same parameters as the one used for the generator.

6.3 Network Analysis

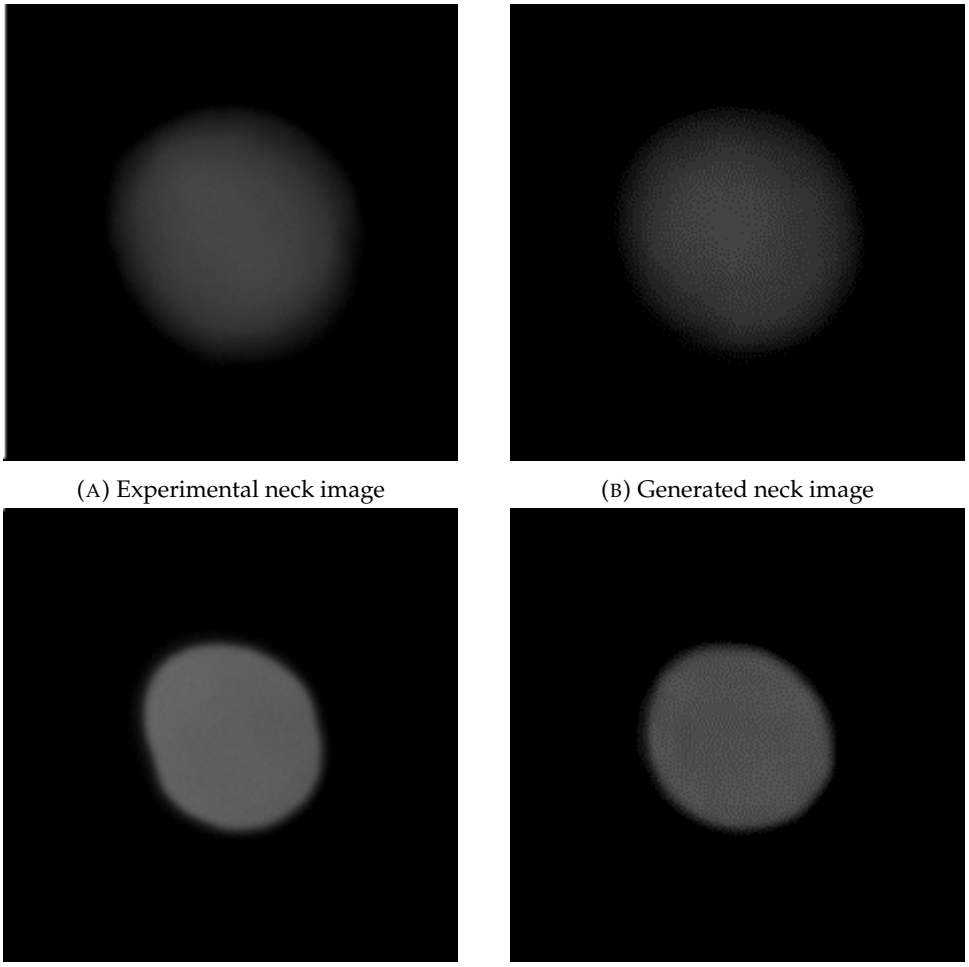
While the original intention was to use the earlier set as a comparison point, the changes in the data collection format meant that this process was not as easy as multiple factors had been changed. Despite this, it still provided a useful reference point to compare training times and quality, as this would be expected to become worse with the increased number of input parameters. There was also significant testing performed on this network in order to demonstrate the possible performance of the network.

6.3.1 Initial Testing: 5 Laser Parameters

As in all previous experiments, there was a separate validation dataset that was kept apart from the data used for training. This data was selected randomly from within the full dataset. While all of the initial machining parameters were selected randomly, there was no way to ensure that there were no systematic influences in the data such as changes in the laser power over time or imperfections in the makeup of the sample. Once trained the two generators described previously were used to recreate both the training and the validation dataset. While the images from the training dataset couldn't be used to fully test the network, they provided a reference point to ensure that the results seen in the validation set are representative of a properly strained network. If there was a large disparity between these it would indicate that overfitting had taken place. While this would be true in all cases, as the data was completely anonymised data high levels of feedback were provided to allow the qualification of the network ability.

After training the GAN for 20 epochs, the network was able to generate clean images that approximated the experimental neck and exit images as shown in Fig. 6.5 without any network artefacts being visible. As can be seen, the network was able to closely match the size and style of both sets of images. The generated neck image in Fig. 6.5b has the characteristic soft edge that is seen in all neck images, such as that in Fig. 6.5a. The exit images in both Figs. 6.5c and 6.5d show a sharp edge around the hole. As well as the sizes being consistent, the generated exit also exhibits some of the asymmetrical nature of the hole that is also present in the experimental example rather than a perfect circle as seen in some of the experimental patterns. The network also captures some of the specle seen in the true images. The network doesn't quite manage to match the full skew of the image, however, and the generated images are generally both dimmer.

As an additional data point, the average of all validation samples was taken for both the experimental images, as well as those generated by the network and the difference between them can be seen in Fig. 6.6. It can be seen that the central points of the image



(A) Experimental neck image

(B) Generated neck image

(C) Experimental exit image

(D) Generated exit image

FIGURE 6.5: Example images from the validation dataset. These holes were produced with normalised laser parameters of 0.21056, 0.356136821, 0.0725, 0.3375, and 0.41.

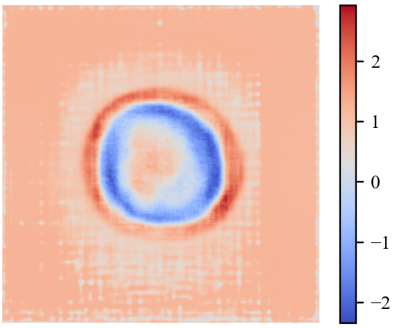


FIGURE 6.6: The difference (subtraction) between the experimental and generated images. On average the background is slightly higher in the experimental images. The central equality indicates the similarity in the maximum brightness. The two turning points indicate that the experimental images had a softer fall off and wider base than the generated ones.

show little difference between the two sets of images. This shows that the network was able to accurately determine the light levels that should be present within the holes. In a similar manner, the average background level is also very similar in the generated images. This is likely due to the effect explored in Section 6.1 where the background of the experimental images was mostly low if not zero. Due to this, there was little information for the network to learn, and as all images experienced the same phenomena in the background, it was an area that could be easily trained. Despite this when taking the difference the small variations are exaggerated that would not be seen in a normal image and it can be seen that there is still evidence present of the network artefacts that are consistent across generated images.

While parts of the images are similar, there is also a band of distinct differences between the images, with an inner band of higher predicted values, and an outer band of lower ones. This suggests a potential cause for the overall difference in predicted radii for the generated images. If the network was simply predicting smaller holes, there would be a single band where the generated images would have a lower average value than the experimental ones. The existence of a band of higher values indicates that the network is not correctly reproducing the falloff in light. Rather than a soft reduction in the values, the images generated by the GAN have a wider solid circle of light surrounded by a sharper gradient. The soft falloff appears to stem from the lack of focus found in the neck images. While the exit images are taken at the bottom surface of the material, providing a clear focal point on which to base the image capture, the neck images are taken with a focal plane that is within the hole.

Having a focal plane that is difficult to determine could cause issues in image capture in two ways, each based on a method of choosing the plane to image. One option is to attempt to focus on the neck of each of the holes in turn. The issue with this is that there is no single feature to provide a reference point for the autofocus algorithm, especially one that is based on maximising sharpness. This method would also have difficulties in calculating the ideal focal distance in the situation of imaging the necks as there were multiple holes imaged in each frame. While the surface should be at a fixed distance, assuming that the sample is flat and properly aligned, the necks of each of the holes may be at different heights. The other option is to keep the focal distance constant, choosing a value that should provide a good image in the majority of situations. The issue with this is that the neck of the hole will not be at a fixed position and so the images will have a different level of blur.

As the network had difficulty with the falloff of light, it suggested that there was an inconsistency in the focusing of the neck images. Whichever focusing method was used, it means that the images themselves would have an internal inconsistency that would be almost, or entirely, independent of the laser parameters chosen. While this is not an ideal set of requirements for the network to learn, two possibilities could have resulted from this. The first is that the network could have predicted an average level

of light falloff and then produced approximately this for each of the images generated. The other possibility would be for the network to predict a random level of focus for each image, with the best case being that this distribution matched the distribution of focus in the original training dataset. In this case, it is evident that the network has defaulted to predicting a good focus in the image. If the network were to be producing a distribution that matched that of the dataset, the average difference in the predicted radii would still be similar, but the average luminosity would show a much closer correlation and the errors would be distributed in both the positive and negative directions.

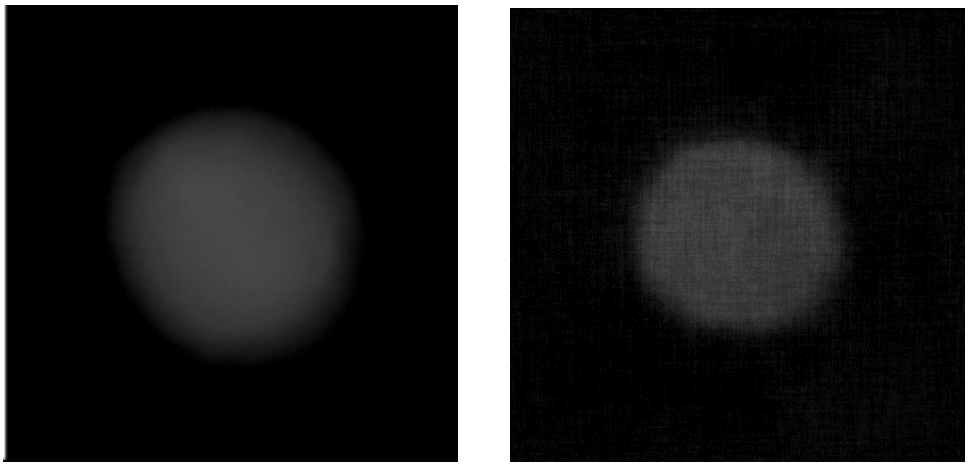
The failure of the network to produce a wide variety of light falloffs represents one of the common failings of generative networks, mode collapse. The network has found a value that can achieve constantly ”not bad” results that produced a lower error than when small deviations to this pattern are applied. One of the main causes of mode collapse is the loss function used, and this was one of the problems that GANs were designed to solve. In the GAN presented here, the comparative loss was given a high weighting when compared to the GAN loss, which in this case was a hinge loss. Mode collapse can be caused by the comparative loss as it can lead the network into producing the exact form averaging seen here. The other contributing factor is the type of comparative loss used, here being an MSE loss. Traditionally in GANs, an MSE loss is preferred as it encourages the network to produce sharper results than when using an MAE loss. This is because the network will suffer higher losses when further from the ground truth due to the error being squared. This works well when the ground truth can be determined from the input directly, but when there is a highly many-to-one relationship and differences from causes beyond the scope of the input, the network may prefer a consistent error as opposed to a mixture of high and low errors. In the particular example here, the mode collapse is not overly detrimental to the performance of the network, but if the network were to predict a consistently out-of-focus image the result would be far worse. While mitigation steps can be put in to avoid this it is ultimately a data quality issue with no obvious solution, especially without direct control over the data collected. For example, the data received could be adjusted manually to cover the full range of the 255 light levels recorded by the camera, but this would not improve the precision of the data and would provide no more information. To train the network, the levels were normalised to the same degree as they would have been with the full range used to avoid any vanishing gradient issues. Some of this could be avoided by simply using a numerical network, but that would have the same limitations discused in Chapter 4 and could not be used to investigate any aspect of the data not captured in the output values.

6.3.2 Varying size of the input dataset

As stated previously, as well as the quality of the data used, another factor that can contribute to the success of the network is the amount of data available to train the network. In Chapter:Optimise it was found that training the ANN was possible with far less than the total amount of data that was provided within the investigation. As data collection can be a major bottleneck in the initial training of a network, it was decided that further investigation on this front would be beneficial and could provide interesting information. In a similar manner to the previous investigation, the amount of input data was varied by taking subsets of the initial training dataset. In order to maintain consistency throughout the testing process, the data used for each task was not fully random. The first step was to separate the initial validation dataset from the data to be used as inputs as this would allow accurate comparisons between all tests. While there would be additional data that was not seen by networks that saw fewer training examples, the randomness of the data in the training set should ensure that a representative sample is shown, and the benefits of direct comparison were deemed to be sufficiently greater than any additional benefit from having an increased number of validation points available.

$$X \subseteq Y \iff |X| \leq |Y| \tag{6.2}$$

In addition to the validation data being kept consistent throughout, an additional decision was taken, that each progressively smaller dataset should simply be a subset of any larger dataset such that Eq. 6.2 is satisfied. This meant that any data pair found in dataset X would be found in dataset Y assuming that there are at least as many data points in Y as there were in X. This was by storing all of the unique combinations in a list before shuffling with a set sort after having removed the validation dataset. This would put the list in repeatably random order, with the ability to change the ordering possible by changing the seed, and allowing for the selection of the training data via a slice of the list. While possible to train the network with any number of parameter combinations, due to the time constraints involved, only 5 sets of data were tested, the original set along with 4 smaller sets. These sets were chosen as percentages of the total training dataset, with values used being: 1%, 3%, 10%, and 30%. While this was not the only way to choose the data for the experiments, it was felt that it would represent the process of data collection and periodic training to assess performance, with data collection runs being conducted after an initial training period. In this situation, the network would have access to all previous data seen by the network in addition to the new data collected by the additional experiments. The values for this progression were chosen based on an approximation of collecting twice again the total amount of data already seen by the network.



(A) 100% of the total data (B) 1% of the total data

FIGURE 6.7: Example neck images generated by GANs after 20 epochs with varying dataset sizes from 100% of the potential date, down to just 1%.

Along with selecting the data to use, a method of determining the allowed training time was required, which in the previous experiment was simply the total number of epochs required for the network to converge. In the case of an ANN, this is possible as in a fully trained network you will reach a plateau point, beyond which no further gains will be made and the risk of overfitting increases. This network was a GAN, however, and therefore convergence cannot be determined by simple stabilisation of the losses therefore an alternative method was required. The first option trialled was to use a fixed number of epochs based on the number required for the originally trained dataset, training all networks for the same number. This proved ineffective as the networks with smaller numbers of images were not trained for long enough to even produce clear images. This effect can be seen in Fig. 6.7 where the GAN trained on the full dataset was able to produce visually sharp and clean images (Fig. 6.7a). In contrast to this, when the dataset used to train the network only contained 1% of the total samples, the results shown in Fig. 6.7b show an image that still has very visible convolution artefacts, caused by the effect shown in Fig. 3.8. Another option would be to provide the network with an equal number of images for training, as is often used to describe the training of GANs. The problem with this option was that the time available made this option prohibitive due to the length of training required on the full dataset, which took a full week to train. This length of training would to two further issues above making the experiment take over a month for a single run and locking up resources from any other task. The first of these issues was that training a network for such a long time on the smaller datasets would lead to high amounts of repetition as well as a high risk of overfitting, with very little possibility of further gains. The long training time is also in antithesis to the aim of this experiment which was to simulate a situation where a small dataset was taken initially to reduce the time taken. If network training time is high and constant then it would likely be beneficial to simply collect more data to maximise the efficiency of training time.

Dataset Size (images)	Number of Epochs	Training Time (hours)	Radius Errors (px)	
			Exit Images	Neck Images
19	140	10	1.46±9.42	2.99±8.54
57	70	16	1.39±11.9	1.47±5.41
190	50	28	0.23±7.95	1.11±4.75
570	24	38	0.63±8.37	1.33±3.99
1900	20	100	-1.0±9.11	2.70±7.53

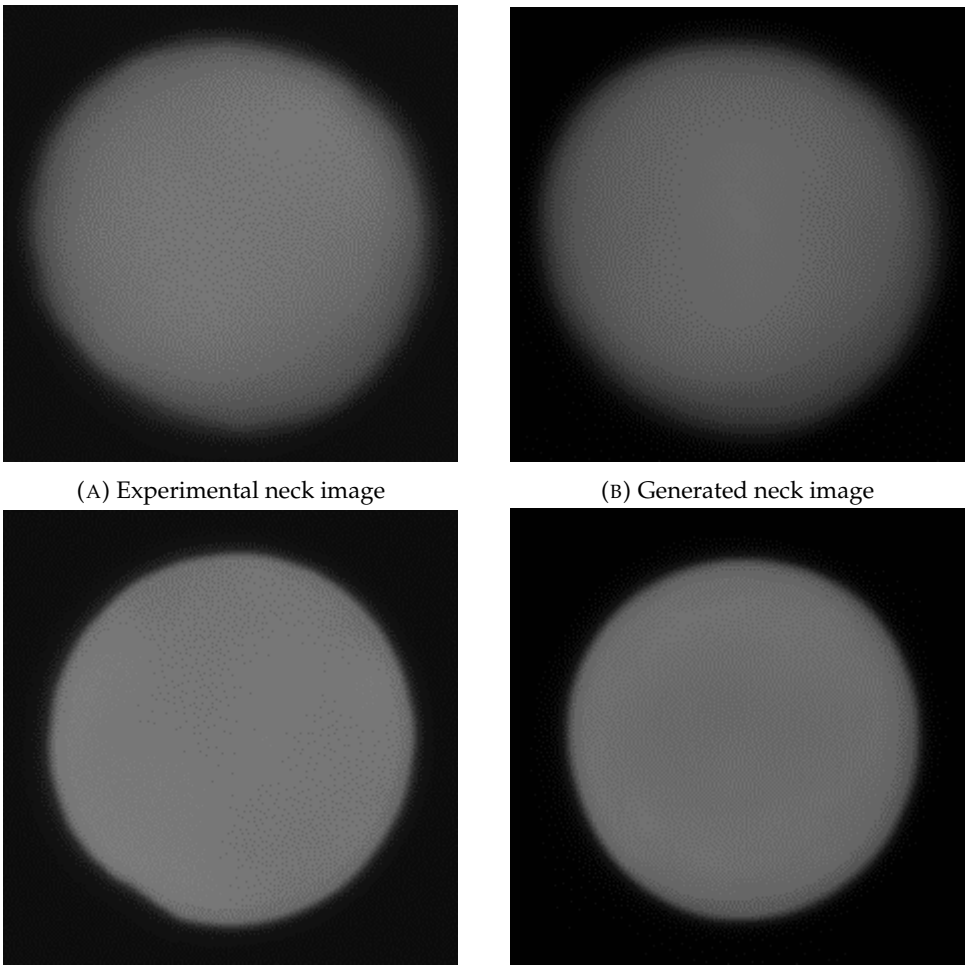
TABLE 6.1: The effect of changing the dataset size on network training and performance.

With the limitations discussed it was decided the training would run without interruption for at least 20 epochs, checking the results at least every 10 epochs. Training would only be stopped once the images produced by the generator had reached a visual fidelity that was hard to distinguish from the experimental samples. This stopping condition led to the networks each being trained for the number of epochs shown in Table 6.1, starting at 140 epochs for the network trained on 19 samples and decreasing to the final 20 found when training on the full dataset. The total time for training each was measured to allow a better comparison of the efficiency of the networks. This was required due to an epoch, one complete set through the full dataset, taking a different length of time for each network. Excluding the network trained on the full dataset, which did not have the same stopping condition as the 4 new ones tested here, the time taken to train the networks closely matches an inverse exponential function with an R^2 value of 0.9998. This shows that while training time did tend to increase with increasing amounts of data, the time to reach visual fidelity did not follow linearly with the amount of data. Other than the amount of time to train the networks, the other factor to consider was the accuracy of the networks in predicting the values chosen for their qualification. This data is again presented in Table 6.1 where it can be seen that there was no general decrease in the errors from the network. In contrast, the error reaches a minimum for the error in the predicted radii for both sets of images with the network trained on 10% of the data where the standard deviation in the errors is also close to being the minimum. This shows that simply waiting for the network to generate images with high visual fidelity is not enough to provide confidence in the performance of the network. In the cases of low amounts of data, the high losses indicate that there is insufficient variation within the training dataset to provide the network with a good ability to generalise. At the other end, when there are high numbers of data points, it might be expected that losses would further decrease. This can be combined with the fact that the networks with more data were also trained for the longest time and so may again be expected to have better performance. Despite this, it is possible that the generators in question did not have sufficient exposure to repeated data to fully train all parameters.

6.3.3 Initial Testing: 9 Laser Parameters

As stated previously, the next step after training with the data produced with 5 laser parameters was to train a network with data that was collected using additional parameters. There were a further 4 parameters used, keeping the original 5 used in the previous experiments to allow for potential data comparison. For this dataset, it was estimated that more training examples would be required due to the increased complexity of the input-to-output mapping. Despite this fewer data points were requested, and this was driven by the previous investigation into the amount of data required for training. It was clear that when using 5 parameters, fewer than 2000 data points were required to achieve acceptable results, and so requesting even more than that would likely just be detrimental to the training process and introduce unnecessary experimental time required. Due to the convenience of data collection timeframes, a similarly sized dataset was provided, with a total of 1500 combinations of laser parameters provided. This was again split into the internal training and validation dataset using the same method as described previously.

Training on this dataset again proved successful with both neck and exit images having high accuracy and visual fidelity as shown in Fig. 6.8. While there are some differences, such as the non-circularity seen in the bottom left of Fig. 6.8c, that cannot be accounted for at an image-to-image comparison level, the general size and shapes of the holes are consistent between the two sets. In addition to this, it can be seen that the network has been able to correctly predict the light falloff seen in the neck image from the experimental dataset, one of the problematic areas from the previous dataset. Using the calculations for the radii on images generated from the validation dataset, the network was able to produce neck images with an MAE of 2.79 pixels, which is larger than the MAE of 1.07 pixels when measured on the training dataset. This indicates that there is still room for improvement in the training of the network, either with further training or with careful monitoring of the validation loss to prevent overfitting. A similar situation is seen with the neck images, with an error of 2.42 pixels for the validation set and 1.04 pixels for the full dataset. When the true value of the error is calculated, rather than the absolute values, the errors on the neck and exit images reduce to 0.78 and 0.83 pixels respectively, showing that while the network does predict images with larger and smaller radii, it is predominantly guessing higher. While the error found in this training run of the 9 parameter data is larger than the best error achieved when the data was only produced using 5 laser parameters, it was better than the results when it was trained on the full dataset. This suggests that while the increased number of parameters may have provided additional difficulties to the network, there is potential for varying the number of training data points to increase the performance of the network.



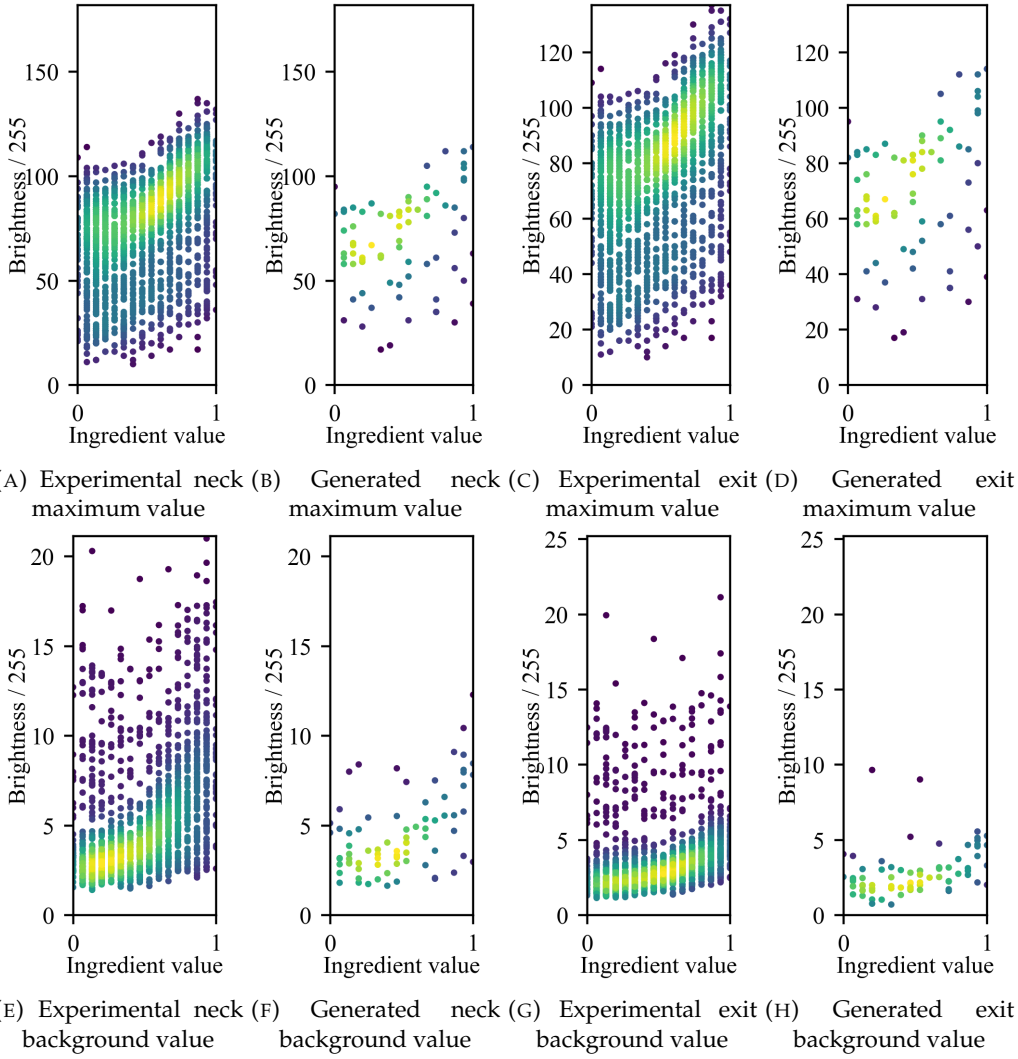
(A) Experimental neck image (B) Generated neck image

(C) Experimental exit image (D) Generated exit image

FIGURE 6.8: Example experimental and generated images from the validation dataset where holes were machined using 9 parameters displaying both neck and exit images. The holes were machined using normalised parameters of 0.899762233, 0.759299781, 0.151282051, 0.987341772, 0.913793103, 0.72, 0.495, 0.266666667, and 0.8.

In addition to the comparisons described above, the maximum and background values were calculated to see how the network had captured some of the more variable information that was less directly tied to the machining parameters used. While not directly linked, it would be expected the fixed exposure of the camera would allow for some coupling to occur, with larger holes both offering less occlusion, and more light bleed to provide higher background and peak values. The first of these was the maximum values from each of the images, also used to observe the effect of the uneven lighting. The second of these values was the background level described earlier. To ensure consistency with the previously calculated values, a truncated mean was taken for each of the images. The low cutoff value was set to 0, and the high cutoff value set the same threshold as used when calculating the radius of the machined holes.

A broad comparison between the experimental and generated images across both



(A) Experimental neck maximum value (B) Generated neck maximum value (C) Experimental exit maximum value (D) Generated exit maximum value

(E) Experimental neck background value (F) Generated neck background value (G) Experimental exit background value (H) Generated exit background value

FIGURE 6.9: A comparison between predicted maximum and background levels from experimental and generated images, grouping the results by the value of the "i" parameter used to machine them. While sparse the generated data (using values found in the training set) shows a good correlation in all cases. The brightness values start low at low values of "i" and grow at an increasing rate.

discussed parameters can be seen in Fig. 6.9. While the data seen for the generated images is slightly harder to picture, as it is far more sparse, consistent trends can be seen in all comparisons. For example, the maximum values in both the experimental and generated images shown in the exit images, in Figs. 6.9c and 6.9d, are both, in general, higher than their counterparts from the neck images seen in Figs. 6.9a and 6.9b. The main area that the generated images are lacking is in the highest values for each of the background and maximum values. For example, the maximum background brightness shown in Fig. 6.9g is close to 25, the comparative highest value from Fig. 6.9h is below 10. While this may be a real effect due to anomalous values not being present in the validation dataset, it is more likely that the network is tending to predict middling values to have a better chance of being close to the truth.

6.4 Conclusions

One of the aims of the experiment described in this chapter was to identify how a neural network might be used in an industrial setting where training may take place over periods of time and with slowly increasing amounts of data. The data found here showed that there was not a simple relationship between the amount of data and the quality of the network, and that simply collecting more data from the start may not be the best option for setting up a neural network. This is caused by two factors, the first being that there may be an inefficiency in the process by both spending more time collecting data than is useful and subsequently causing the network to take longer to train. It also indicates that with a large amount of input data the network may give an artificially low level of confidence due to long training times and comparatively low performance. Training with smaller and progressively growing datasets both naturally matches with the process of experimentation and allows for visibility in the training of a network when compared to the amount of data, indicating when sufficient data may have been collected. One opportunity that should be explored to gain a better comparison to the concept of a growing dataset is the idea of transfer learning. In this situation, the network would be continuously trained on larger sets of data while still maintaining the updated weights from the previous training epochs. Transfer can greatly reduce the training time of large networks when they are used for new tasks, and this would be especially true for the example here. When testing on networks of varying sizes a lot of time was spent on image style, with visual fidelity being the stopping point, and networks trained on smaller datasets took far less time to train than those trained on larger ones. While each epoch would take longer with more data, fewer epochs may be required to reach equal or better results. It would be valuable to investigate how new data should be added to the training dataset as data that has been available since the initial training of the network will have been seen more and so may cause an undesirable bias in the training.

In addition to the testing on data collected with 5 parameters, an additional set of data was taken using an additional 4 parameters. A network was then trained on this data to observe the effect of increasing data complexity on the ability of the network to learn. It was found that not only was the network able to be trained on a similarly sized dataset that could be used for the original network, but the performance also exceeded the network at that point. While it was not possible to test the training of this network with varying amounts of data, a full investigation would be valuable to determine the effect of data complexity on the possible performance of a network trained on it. This would provide valuable information with which to determine the amount of data that should be collected if the experiment was to be attempted with a greater number of variables, up to the 15 available. Again this is a situation in which transfer learning should be examined, using the network trained on the data collected with 5 parameters as a basis for the 9-parameter data.

While there has been some interesting information drawn from this experiment, the most important aspect was the investigation into the process. Every laser machining and machine learning task will be different and so it is impossible to determine an absolute number of parameters that would be ideal. This will depend on a huge number of factors, from the structure of the network, the complexity and type of the data, and even the time taken to perform the experiment and train the networks. Here a problem was presented and some of the possible routes to optimisation were presented, rather than assuming that simply collecting as much data as possible would be the ideal situation. It was shown that it is possible to gradually increase the amount of data to provide good results while showing that a relationship between training time and data volume should be observed as it may provide valuable information.

An additional step that could be taken to improve the quality of the results achieved would be to move away from the single model approach and go with an ensemble of networks designed for different tasks. The dominating feature of the images is the size and brightness of the holes and so many loss functions will focus on those aspects and can lead to more averaging than would be desired. The use of a GAN loss helps to mitigate this but cannot remove entirely. Of more interest perhaps would be to also develop a network with a loss function designed to capture the shape of the speckle seen in the images as that could provide important information about the internal structure of the holes.

Chapter 7

Conclusions and Future Work

Throughout this thesis, an investigation has been performed into how various machine-learning techniques can be applied to the field of laser machining. It was discussed how these methods can be used to both predict the properties of the features that would be machined as well as provide visualisations of these results. In addition to all of the following discussions, there are continuous advancements in the field of generative neural networks and the trade-off these provide should be fully considered. For example, one novel type of network that was not discussed in this report is the diffusion style model that slowly and iteratively creates images. These can produce images with very high fidelity but at a slow speed offering an interesting alternative that would be well worth exploring.

7.1 Optimisation of Laser Machining Parameters

In laser machining tasks data collection and processing can be a lengthy process, requiring preparation time and expensive equipment. To avoid some of the pains involved in collecting and presenting the results of a machining process, a variety of machine learning techniques were investigated. Throughout this investigation 4 different techniques were employed, each having several variations to provide an understanding of the tools available and which could be best applied. The task to be solved was the prediction of the dimple depth and crown height that would result from machining with different laser parameters in a cast iron sample.

It was found that out of the analytical methods investigated, the ANN was able to outperform both an SVM regression technique as well as a Gaussian Process. The ANN used was able to produce an accuracy that matched the internal variation within the data. Along with the accuracy of the method, the network was able to perform predictions far more quickly than could be achieved in an experimental setting.

Beyond simple quick calculation, the network has an instant setup and so would be possible to integrate into another process, such as parameter optimisation and instant verification of prediction.

Other than the analytical approach demonstrated by the ANN, a generative method was also tried, utilising a GAN to simulate the full 3-dimensional profile of the dimples. Despite the extra complexity involved, the GAN was able to produce results that matched the crown height and dimple depth close to the quality of the ANN and better than some of the other analytical attempts. While the GAN would not be suitable for the optimisation tasks discussed it could be used as an additional verification tool to complement the other methods. The results from the GAN also contained far more data, so it would be possible to extract any information from it that could be gained from true experimental samples.

While the results from this experiment were positive, there were still areas that could be improved. The data collected was heavily influenced by human bias as the engineers collected most of the data in regions they thought were of particular interest. This meant that the data used did not provide a range and granularity that could be fully utilised by the network. The GAN, while able to provide accurate estimations for crown height and dimple depth, did not match the visual fidelity of the true values, with the results being smoother than expected. Further investigation would be required to determine the cause and to find a solution.

Now that the use of machine learning has proven to be accurate at predicting the results from the machining tasks at hand a future task would be to increase the range of values seen by the network. Alongside this, a catalogue of networks could be developed to provide solutions to a wide range of problems. The speed of creating these networks ensures that it could coincide with other work that is undertaken, providing a complementary technique.

7.2 Modelling Laser Machining Processes that use Spatial Light Modification

In this chapter initial work was focused on improving the capabilities of neural networks produced within the group to simulate the effect of a shaped pulse on a sample. Through the use of novel GANs, the effect was scaled to use up to three pulses in the machining process. Even with the added complexity of more inputs and more variable output space, the network was able to produce results that closely matched the results from the experiment. The use of multiple pulses also provided an opportunity to investigate whether the network could learn effects that could only be introduced by a subsequent pulse. The first of these was the effect of the ordering of

the pulses, where the network was able to correctly show that fine features machined early on would be smoothed by any coarse machining that took place afterwards. Along with this the network correctly predicted that the diffraction limit of laser machining is mainly a concept of a single pulse and that when machining with multiple pulses, it is possible to produce features that beat this limit.

The second task attempted was to create a network that performed the same operation as the first but in reverse. This was particularly challenging due to two factors. The first was that the translation from depth profile to a sequence of pulses was expected to be far more difficult than the previous task, and had been attempted previously without success. The difficulty arose from the task being highly one-to-many, with the three output patterns having many possibilities that were all correct. The other issue was that it was not possible to either collect more experimental data, which would have helped with the complex problem. The solution to this was two-fold and represented the major advance in the chapter. The first step was the introduction of a hybrid cycle consistent network that used the previously demonstrated network as part of the loss for the new network. This was combined with using the previous network to generate large amounts of data on which to train the new network. Based on this the GAN used to generate depth profiles simulated the full experimental process, providing an artificial feedback loop. Even when tested on previous experimental data, the new network was able to achieve impressive results.

To improve the process described here, a possible next step would be to create a network that was able to take in both a surface profile and a spatial intensity mask. Having this would allow for the network to predict the results from an arbitrary number of pulses as the process could become iterative, predicting a single pulse at a time. This could also be combined with an increased variation within the input, such as including the laser power as an input to the network. This would provide a more robust network that could be used across a wide range of applications. Data collection for this would be one of the challenges to overcome, as for successive machining, the surface profile would need to be measured between each pulse, either in place or moved with a high level of repeatability.

7.3 Simulating Unknown Machining Processes

While careful control over, and creation of, the data used in a machine learning process is a good way to improve the likelihood of achieving good results it is not always possible. This experiment took the example to the extreme, taking a situation where nothing about the data was known apart from what could be seen in the data itself. This provided an excellent opportunity to showcase the raw possible performance of a neural network. Removing knowledge of the data also acted to

remove human bias from the results, which in turn could lead to a better representation of the data space. It was shown that even with no knowledge of, or control over, the data, preparation of the data is still highly important, with poor data still leading to poor results. Not knowing the final purpose of the data made it difficult to develop a suitable process to determine the quality of the network.

As this task was linked to industry one of the key problems investigated was the effect of data quantity and complexity on the results from the network. It was discovered that even with few data points and little augmentation it was possible to produce both visually and numerically accurate results, and show how the improvements scaled with increasing data set size. An important issue was also discovered in the setting of goals. When training a GAN it is hard to define a stopping point without a very clear goal, and when using the visual fidelity of the image as that stopping point, the performance of the network for other tasks was affected.

This is an ongoing experiment with many avenues left to explore. One path is to conduct a more thorough investigation into the data requirements for the network based on machining with 9 laser parameters. While the network was able to produce accurate results, the investigation of the smaller dataset proved that varying the size of the dataset can have a large impact. The next step would be to use these results to decide on a data collection plan for increasing the complexity of the task by introducing increased numbers of parameters, up to the 15 available to use.

Appendix A

Published Work

McDonnell, Michael, David Tom, Grant-Jacob, James, Praeger, Matthew, Eason, R.W. and Mills, Benjamin (2021) *Identification of spatial intensity profiles from femtosecond laser machined depth profiles via neural networks.* Optics Express, 29 (22), 36469-36486. (doi:10.1364/OE.431441).

McDonnell, Michael, David Tom, Grant-Jacob, James, Mills, Benjamin, Eason, R.W., Praeger, Matthew, Karnakis, Dimitris, Arnaldo, Daniel and Pelletier, Etienne (2021) *Machine learning for multi-dimensional optimisation and predictive visualisation of laser machining.* Journal of Intelligent Manufacturing, 32 (5), 1471–1483. (doi:10.1007/s10845-020-01717-4).

McDonnell, Michael, David Tom, Grant-Jacob, James, Xie, Yunhui, Praeger, Matthew, MacKay, Benita, Scout, Eason, R.W. and Mills, Benjamin (2020) *Modelling laser machining of nickel with spatially shaped three pulse sequences using deep learning.* Optics Express, 28 (10), 14627-14637. (doi:10.1364/OE.381421).

McDonnell, Michael, Arnaldo, Daniel, Pelletier, Etienne, Grant-Jacob, James, Praeger, Matthew, Karnakis, Dimitris, Eason, R.W., Mills, Ben (2021) *Using Machine Learning for Prediction and Optimisation in Laser Machining.* 7th International Laser Applications Symposium. 24-25 March 2021

Mills, Benjamin, McDonnell, Michael, David Tom, Heath, Daniel, Xie, Yunhui, Grant-Jacob, James, MacKay, Benita, Scout, Praeger, Matthew and Eason, Robert (2019) *Deep learning for 3D modelling of multiple pulse femtosecond ablation.* In 15th International Conference on Laser Ablation (COLA) 2019.

Xie, Yunhui, Heath, Daniel J, Grant-Jacob, James, MacKay, Benita, Scout, McDonnell, Michael, David Tom, Praeger, Matthew, Eason, Robert and Mills, Benjamin (2019) *Deep learning for the monitoring and process control of femtosecond laser machining.* Journal of Physics: Photonics, 1 (3), 1-10, [3]. (doi:10.1088/2515-7647/ab281a).

Xie, Yunhui, Heath, Daniel, Grant-Jacob, James, MacKay, Benita, Scout, McDonnell, Michael, David Tom, Praeger, Matthew, Eason, Robert and Mills, Benjamin (2019) *Smart lasers for manufacturing of the future.* FEPS IoT Showcase. 05 Apr 2019.

Appendix B

Kerr Effect

The Kerr effect is a result of third order non-linear polarisation of light as is passes through a material. As the first order susceptibility is generally much larger than those of higher orders, at low intensities only the linear interactions need to be considered. However, at greater intensities higher orders become increasingly important. The polarisation (P) can be calculated as shown in Eq. B.1 where χ_n represents the nth order susceptibility, ϵ_0 is the permeability of free space and E is the electric field.

$$P = \epsilon_0(\chi_0 E + \chi_1 E^2 + \cdots) \tag{B.1}$$

This demonstrates that even when $\chi_1 \gg \chi_3$ if the electric field is great enough then the contributions from the third order susceptibility will become relevant. To explore this further the electric field can be considered as a one-dimensional wave of the form show in Eq. B.2 where E is the electric field, the amplitude of the field oscillations is E_0, and ω is the angular frequency of the wave.

$$E = E_0 \cos \omega t \tag{B.2}$$

Specifically investigating the effect of the third order susceptibility, the third term becomes $\chi_2 E_0^3 \cos^3 \omega t$. Using standard trigonometric identities, the cube of the electric field can be rewritten as:

$$E^3 = \frac{E_0^3}{4}(\cos 3\omega t + 3\cos \omega t)$$

Written in this form, the contributions from third order non-linearities can be separated into two terms. The term containing the $\cos \omega t$ represents third harmonic generation, with light being emitted with a wavelength a third of that of the incident

wave. The other term can be rewritten as $\frac{3}{4} \cdot E_0^2 \cdot (E_0 \cos \omega t)$ where E_0^2 is proportional to the intensity (I) of the electric field. This combined with the fact that $E_0 \cos \omega t$ was the definition of the initial wave means that the refractive index of the material becomes intensity dependent and can be rewritten as:

$$n = n_0 + In_2$$

Where n_2 is the contribution from third order non-linearities. This has the effect for a beam with a Gaussian spatial intensity profile, as is typical in many situations involving laser beams, the beam will become self-focussing when passing through a medium with sufficiently large χ_2 This is because the central region of the beam along the optical axis has a higher intensity and as such experiences a larger refractive index. This in turn leads to the higher intensity sections being delayed due to the rest of the wave, the same effect as a lens.

References

Global Laser Processing Market Size Report, 2030. URL https://www.grandviewresearch.com/industry-analysis/laser-processing-market.

Namrata Anand and Possu Huang. Generative modeling for protein structures. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/afa299a4d1d8c52e75dd8a24c3ce534f-Abstract.html.

Daniel Arnaldo, Del Cerro, Etienne Pelletier, Dimitris Karnakis, Alexandre Cunha, and Karyne Juste. Laser surface texturing of grey cast iron for tribological applications in refrigeration hermetic compressors: the effect of processing parameters on ablated crater rim formation. August 2018.

Yuemin Bian and Xiang-Qun Xie. Generative chemistry: drug discovery with deep learning generative models. *Journal of Molecular Modeling*, 27(3):71, February 2021. ISSN 0948-5023. . URL https://doi.org/10.1007/s00894-021-04674-8.

Kevin W. Bowyer, Michael C. King, Walter J. Scheirer, and Kushal Vangara. The "Criminality From Face" Illusion. *IEEE Transactions on Technology and Society*, 1(4): 175–183, December 2020. ISSN 2637-6415. . URL https://ieeexplore.ieee.org/document/9233349/.

S. L. Campanelli, G. Casalino, A. D. Ludovico, and C. Bonserio. An artificial neural network approach for the control of the laser milling process. *The International Journal of Advanced Manufacturing Technology*, 66(9):1777–1784, June 2013. ISSN 1433-3015. . URL https://doi.org/10.1007/s00170-012-4457-9.

X. Cao, W. Wallace, C. Poon, and J.-P. Immarigeon. Research and Progress in Laser Welding of Wrought Aluminum Alloys. I. Laser Welding Processes. *Materials and Manufacturing Processes*, 18(1):1–22, January 2003. ISSN 1042-6914. . URL https://doi.org/10.1081/AMP-120017586.

X. Cao, M. Jahazi, J. P. Immarigeon, and W. Wallace. A review of laser welding techniques for magnesium alloys. *Journal of Materials Processing Technology*, 171(2): 188–204, January 2006. ISSN 0924-0136. . URL https://www.sciencedirect.com/science/article/pii/S092401360500734X.

G Casalino and A D Ludovico. Parameter selection by an artificial neural network for a laser bending process. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 216(11):1517–1520, November 2002. ISSN 0954-4054. . URL https://doi.org/10.1243/095440502320783350.

Giuseppe Casalino, Aurora Maria Losacco, Angela Arnesano, Francesco Facchini, Maurizio Pierangeli, and Cesare Bonserio. Statistical Analysis and Modelling of an Yb: KGW Femtosecond Laser Micro-drilling Process. *Procedia CIRP*, 62:275–280, 2017. ISSN 22128271. . URL https://linkinghub.elsevier.com/retrieve/pii/S2212827117300628.

W. S. Chang and S. J. Na. Prediction of laser-spot-weld shape by numerical analysis and neural network. *Metallurgical and Materials Transactions B*, 32(4):723–731, August 2001. ISSN 1543-1916. . URL https://doi.org/10.1007/s11663-001-0126-3.

Peter Yichen Chen, Jonathan David Blutinger, Yorán Meijers, Changxi Zheng, Eitan Grinspun, and Hod Lipson. Visual modeling of laser-induced dough browning. *Journal of Food Engineering*, 243:9–21, February 2019. ISSN 0260-8774. . URL https://www.sciencedirect.com/science/article/pii/S0260877418303595.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, June 2016. URL http://arxiv.org/abs/1606.03657. arXiv:1606.03657 [cs, stat].

J. Cheng, W. Perrie, S. P. Edwardson, E. Fearon, G. Dearden, and K. G. Watkins. Effects of laser operating parameters on metals micromachining with ultrafast lasers. *Applied Surface Science*, 256(5):1514–1520, December 2009. ISSN 0169-4332. . URL https://www.sciencedirect.com/science/article/pii/S0169433209012926.

P. J Cheng and S. C Lin. Using neural networks to predict bending angle of sheet metal formed by laser. *International Journal of Machine Tools and Manufacture*, 40(8): 1185–1197, June 2000. ISSN 0890-6955. . URL https://www.sciencedirect.com/science/article/pii/S089069559900111X.

Yu Cheng, Yongshun Gong, Yuansheng Liu, Bosheng Song, and Quan Zou. Molecular design in drug discovery: a comprehensive review of deep generative models. *Briefings in Bioinformatics*, 22(6):bbab344, November 2021. ISSN 1477-4054. . URL https://doi.org/10.1093/bib/bbab344.

I. A. Choudhury and S. Shirley. Laser cutting of polymeric materials: An experimental investigation. *Optics & Laser Technology*, 42(3):503–508, April 2010. ISSN 0030-3992. . URL https://www.sciencedirect.com/science/article/pii/S0030399209001972.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, September 1995. ISSN 1573-0565. . URL https://doi.org/10.1007/BF00994018.

Alexander F. Courtier, Michael McDonnell, Matt Praeger, James A. Grant-Jacob, Christophe Codemard, Christophe Codemard, Paul Harrison, Ben Mills, and Michalis Zervas. Modelling of fibre laser cutting via deep learning. *Optics Express*, 29(22):36487–36502, October 2021a. ISSN 1094-4087. . URL https://opg.optica.org/oe/abstract.cfm?uri=oe-29-22-36487.

Alexander F. Courtier, Michael McDonnell, Matt Praeger, James A. Grant-Jacob, Christophe Codemard, Paul Harrison, Ben Mills, and Michalis Zervas. Predictive Visualisation of Fibre Laser Machining via Deep Learning. In *2021 Conference on Lasers and Electro-Optics Europe & European Quantum Electronics Conference (CLEO/Europe-EQEC)*, pages 1–1, June 2021b. .

Lean L. Dasallas and Wilson O. Garcia. Numerical simulation of femtosecond pulsed laser ablation of copper for oblique angle of incidence through two-temperature model. *Materials Research Express*, 5(1):016518, January 2018. ISSN 2053-1591. . URL https://iopscience.iop.org/article/10.1088/2053-1591/aaa4e8/meta.

Rina Dechter. Learning while searching in constraint-satisfaction-problems. In *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*, AAAI'86, pages 178–183, Philadelphia, Pennsylvania, August 1986. AAAI Press.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Li Deng and Dong Yu. Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387, June 2014. ISSN 1932-8346. . URL https://doi.org/10.1561/2000000039.

Dua Dheeru and Casey Graff. UCI Machine Learning Repository, 2017. URL http://archive.ics.uci.edu/ml.

Yiwei Dong, Yiwei Dong, Zongpu Wu, Yancheng You, Chunping Yin, Wenhui Qu, and Xiaoji Li. Numerical simulation of multi-pulsed femtosecond laser ablation: effect of a moving laser focus. *Optical Materials Express*, 9(11):4194–4208, November 2019. ISSN 2159-3930. . URL https://www.osapublishing.org/ome/abstract.cfm?uri=ome-9-11-4194. Publisher: Optical Society of America.

Ranran Fang, Duanming Zhang, Hua Wei, Zhihua Li, Fengxia Yang, and Yihua Gao. Improved two-temperature model and its application in femtosecond laser ablation of metal target. *Laser and Particle Beams*, 28(1):157–164, March 2010. ISSN 1469-803X, 0263-0346. . URL https://www.cambridge.org/core/journals/

laser-and-particle-beams/article/abs/
improved-twotemperature-model-and-its-application-in-femtosecond-laser-ablation-of-metal-
DC418EC75EF0429B199D58E8BFC4C111. Publisher: Cambridge University Press.

Evelyn Fix and Joseph L Hodges. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. Technical report, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

Yang Gao, Rita Singh, and Bhiksha Raj. Voice Impersonation Using Generative Adversarial Networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2506–2510, April 2018. . ISSN: 2379-190X.

Rafael R. Gattass and Eric Mazur. Femtosecond laser micromachining in transparent materials. *Nature Photonics*, 2(4):219–225, April 2008. ISSN 1749-4893. . URL https://www.nature.com/articles/nphoton.2008.47.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts, 2016. ISBN 9780262035613.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680. MIT University, December 2014.

James A. Grant-Jacob, Ben Mills, and Robert W. Eason. Parametric study of the rapid fabrication of glass nanofoam via femtosecond laser irradiation. *Journal of Physics D: Applied Physics*, 47(5):055105, January 2014. ISSN 0022-3727. . URL https://doi.org/10.1088/0022-3727/47/5/055105.

Mahdi Hashemi and Margeret Hall. RETRACTED ARTICLE: Criminal tendency detection from facial images and the gender bias effect. *Journal of Big Data*, 7(1):2, January 2020. ISSN 2196-1115. . URL https://doi.org/10.1186/s40537-019-0282-4.

M. Th Hassan, T. T. Luu, A. Moulet, O. Raskazovskaya, P. Zhokhov, M. Garg, N. Karpowicz, A. M. Zheltikov, V. Pervak, F. Krausz, and E. Goulielmakis. Optical attosecond pulses and tracking the nonlinear response of bound electrons. *Nature*, 530(7588):66–70, February 2016. ISSN 1476-4687. . URL https://www.nature.com/articles/nature16528.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. pages 770–778. IEEE, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Daniel J. Heath, James A. Grant-Jacob, Matthias Feinaeugle, Ben Mills, and Robert W. Eason. Sub-diffraction limit laser ablation via multiple exposures using a digital micromirror device. *Applied Optics*, 56(22):6398–6404, August 2017. ISSN 2155-3165. . URL https://www.osapublishing.org/ao/abstract.cfm?uri=ao-56-22-6398. Publisher: Optical Society of America.

Daniel J. Heath, James A. Grant-Jacob, Robert W. Eason, and Ben Mills. Single-pulse ablation of multi-depth structures via spatially filtered binary intensity masks. *Applied Optics*, 57(8):1904–1909, March 2018a. ISSN 2155-3165. . URL https://www.osapublishing.org/ao/abstract.cfm?uri=ao-57-8-1904. Publisher: Optical Society of America.

Daniel J. Heath, James A. Grant-Jacob, Yunhui Xie, Benita S. Mackay, James A. G. Baker, Robert W. Eason, and Ben Mills. Machine learning for 3D simulated visualization of laser machining. *Optics Express*, 26(17):21574–21584, August 2018b. ISSN 1094-4087. . URL https://www.osapublishing.org/oe/abstract.cfm?uri=oe-26-17-21574.

Jeff Hecht. Optics: Light for a New Age. In *Optics: Light for a New Age*, pages 237–238. Jeff Hecht, April 1987. ISBN 9780684188799.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, July 2006. ISSN 0899-7667. . URL https://doi.org/10.1162/neco.2006.18.7.1527.

Rongjie Huang, Chenye Cui, Feiyang Chen, Yi Ren, Jinglin Liu, Zhou Zhao, Baoxing Huai, and Zhefeng Wang. SingGAN: Generative Adversarial Network For High-Fidelity Singing Voice Generation, August 2022. URL http://arxiv.org/abs/2110.07468. arXiv:2110.07468 [cs, eess].

Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456, Lille, France, July 2015. JMLR.org.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, July 2017. . ISSN: 1063-6919.

Jeng-Ywan Jeng, Tzuoh-Fei Mau, and Shyeu-Ming Leu. Prediction of laser butt joint welding parameters using back propagation and learning vector quantization networks. *Journal of Materials Processing Technology*, 99(1):207–218, March 2000. ISSN 0924-0136. . URL https://www.sciencedirect.com/science/article/pii/S0924013699004240.

Lan Jiang and Hai-Lung Tsai. An improved two-temperature model for metal thin film heating by femtosecond laser pulses. *International Congress on Applications of Lasers & Electro-Optics*, 2004(1):M602, October 2004. . URL https://lia.scitation.org/doi/10.2351/1.5060340. Publisher: Laser Institute of America.

T. Karras, S. Laine, and T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, June 2019. . ISSN: 2575-7075.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, June 2020. . ISSN: 2575-7075.

Dong-Hyeon Kim, Thomas J. Y. Kim, Xinlin Wang, Mincheol Kim, Ying-Jun Quan, Jin Woo Oh, Soo-Hong Min, Hyungjung Kim, Binayak Bhandari, Insoon Yang, and Sung-Hoon Ahn. Smart Machining Process Using Machine Learning: A Review and Perspective on Machining Industry. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 5(4):555–568, August 2018. ISSN 2288-6206, 2198-0810. . URL http://link.springer.com/10.1007/s40684-018-0057-y.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alexander Laskin and Vadim Laskin. ?Shaper ? Refractive Beam Shaping Optics for Advanced Laser Technologies. *Journal of Physics: Conference Series*, 276:012171, March 2011. .

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. pages 4681–4690, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Ledig_Photo-Realistic_Single_Image_CVPR_2017_paper.html.

J. M. Liu. Simple technique for measurements of pulsed Gaussian-beam spot sizes. *Optics Letters*, 7(5):196, May 1982. ISSN 0146-9592, 1539-4794. . URL https://opg.optica.org/abstract.cfm?URI=ol-7-5-196.

Jen-Yu Liu, Yu-Hua Chen, Yin-Cheng Yeh, and Yi-Hsuan Yang. Unconditional Audio Generation with Generative Adversarial Networks and Cycle Regularization. In

Interspeech 2020, pages 1997–2001. ISCA, October 2020. . URL https://www.isca-speech.org/archive/interspeech_2020/liu20i_interspeech.html.

Raoul-Amadeus Lorbeer, Jan Pastow, Michael Sawannia, Peter Klinkenberg, Daniel Johannes Förster, and Hans-Albert Eckel. Power Spectral Density Evaluation of Laser Milled Surfaces. *Materials*, 11(1):50, December 2017. ISSN 1996-1944. . URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5793548/.

M. D. T. McDonnell, J. A. Grant-Jacob, Y. Xie, M. Praeger, B. S. Mackay, R. W. Eason, and B. Mills. Modelling laser machining of nickel with spatially shaped three pulse sequences using deep learning. *Optics Express*, 28(10):14627–14637, May 2020. ISSN 1094-4087. . URL https://www.osapublishing.org/oe/abstract.cfm?uri=oe-28-10-14627.

M. D. T. McDonnell, B. Mills, M. Praeger, J. A. Grant-Jacob, and R. W. Eason. Data for "Identification of spatial intensity profiles from femtosecond laser machined depth profiles via neural networks", 2021a. URL http://dx.doi.org/10.5258/SOTON/XXXXXX.

Michael D. T. McDonnell, Daniel Arnaldo, Etienne Pelletier, James A. Grant-Jacob, Matthew Praeger, Dimitris Karnakis, Robert W. Eason, and Ben Mills. Machine learning for multi-dimensional optimisation and predictive visualisation of laser machining. *Journal of Intelligent Manufacturing*, 32(5):1471–1483, June 2021b. ISSN 0956-5515, 1572-8145. . URL https://link.springer.com/10.1007/s10845-020-01717-4.

Francesco Mezzapesa, Michele Scaraggi, Giuseppe Carbone, Donato Sorgente, Antonio Ancona, and Pietro Lugarà. Varying the Geometry of Laser Surface Microtexturing to Enhance the Frictional Behavior of Lubricated Steel Surfaces. *Physics Procedia*, 41:670–675, May 2013. .

B. Mills, M. Feinaeugle, C. L. Sones, N. Rizvi, and R. W. Eason. Sub-micron-scale femtosecond laser ablation using a digital micromirror device. *Journal of Micromechanics and Microengineering*, 23(3):035005, January 2013a. ISSN 0960-1317. . URL https://doi.org/10.1088/0960-1317/23/3/035005.

B. Mills, D. J. Heath, M. Feinaeugle, J. A. Grant-Jacob, and R. W. Eason. Laser ablation via programmable image projection for submicron dimension machining in diamond. *Journal of Laser Applications*, 26(4):041501, August 2014. ISSN 1042-346X. . URL https://lia.scitation.org/doi/10.2351/1.4893749. Publisher: Laser Institute of America.

Ben Mills, Daniel J. Heath, James A. Grant-Jacob, and Robert W. Eason. Predictive capabilities for laser machining via a neural network. *Optics Express*, 26(13): 17245–17253, June 2018. ISSN 1094-4087. . URL https://www.osapublishing.org/oe/abstract.cfm?uri=oe-26-13-17245.

Ben Mills, Daniel J. Heath, James A. Grant-Jacob, Yunhui Xie, and Robert W. Eason. Image-based monitoring of femtosecond laser machining via a neural network. *Journal of Physics: Photonics*, 1(3):1–10, January 2019a. ISSN 2515-7647. URL https://eprints.soton.ac.uk/423066/.

Benjamin Mills and James A. Grant-Jacob. Lasers that learn: The interface of laser machining and machine learning. *IET Optoelectronics*, pages 1–18, April 2021. ISSN 1751-8776. . URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ote2.12039.

Benjamin Mills, James A. Grant-Jacob, Matthias Feinaeugle, and Robert W. Eason. Single-pulse multiphoton polymerization of complex structures using a digital multimirror device. *Optics Express*, 21(12):14853–14858, June 2013b. ISSN 1094-4087. . URL https://www.osapublishing.org/oe/abstract.cfm?uri=oe-21-12-14853.

Benjamin Mills, Daniel Heath, James Grant-Jacob, Benita MacKay, and Robert Eason. Neural networks for predictive laser machining capabilities. In *Emerging Digital Micromirror Device Based Systems and Applications XI*. SPIE, March 2019b. . URL https://eprints.soton.ac.uk/428811/.

Hiroaki Misawa and Saulius Juodkazis. *3D laser microfabrication: principles and applications*. Wiley-VCH, 2010. ISBN 9783527310555 9783527608461 9783527608409 9781280723384. OCLC: 1039166128.

A. K. Nath, T. Reghu, C. P. Paul, M. O. Ittoop, and P. Bhargava. High-power transverse flow CW CO2 laser for material processing applications. *Optics & Laser Technology*, 37(4):329–335, June 2005. ISSN 0030-3992. . URL https://www.sciencedirect.com/science/article/pii/S0030399204000994.

B. Neuenschwander, B. Jaeggi, and M. Schmid. From fs to Sub-ns: Dependence of the Material Removal Rate on the Pulse Duration for Metals. *Physics Procedia*, 41: 794–801, January 2013. ISSN 1875-3892. . URL https://www.sciencedirect.com/science/article/pii/S1875389213001648.

Atsuhiro Noguchi and Tatsuya Harada. Image Generation From Small Datasets via Batch Statistics Adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2750–2758, Seoul, Korea (South), October 2019. IEEE. ISBN 978-1-72814-803-8. . URL https://ieeexplore.ieee.org/document/9009051/.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2642–2651. PMLR, July 2017. URL https://proceedings.mlr.press/v70/odena17a.html.

Sehyeok Oh and Hyungson Ki. Deep learning model for predicting hardness distribution in laser heat treatment of AISI H13 tool steel. *Applied Thermal*

Engineering, 153:583–595, May 2019. ISSN 1359-4311. . URL https://www.sciencedirect.com/science/article/pii/S1359431118361696.

Andreas Otto, Holger Koch, and Rodrigo Gomez Vazquez. Multiphysical Simulation of Laser Material Processing. *Physics Procedia*, 39:843–852, January 2012. ISSN 1875-3892. . URL https://www.sciencedirect.com/science/article/pii/S1875389212026363.

Pedram Parandoush and Altab Hossain. A review of modeling and simulation of laser beam machining. *International Journal of Machine Tools and Manufacture*, 85:135–145, October 2014. ISSN 0890-6955. . URL https://www.sciencedirect.com/science/article/pii/S0890695514000856.

T. Park, M. Liu, T. Wang, and J. Zhu. Semantic Image Synthesis With Spatially-Adaptive Normalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341, June 2019. . ISSN: 2575-7075.

Dr Rüdiger Paschotta. Ultrafast Lasers. URL https://www.rp-photonics.com/ultrafast_lasers.html.

Matthew Praeger, Yunhui Xie, James Grant-Jacob, R. W. Eason, and Benjamin Mills. Playing optical tweezers with deep reinforcement learning: in virtual, physical and augmented environments. *Machine Learning: Science and Technology*, January 2021. ISSN 2632-2153. URL https://eprints.soton.ac.uk/446998/.

Ningsong Qu, Xiaolei Chen, Hansong Li, and Yongbin Zeng. Electrochemical micromachining of micro-dimple arrays on cylindrical inner surfaces using a dry-film photoresist. *Chinese Journal of Aeronautics*, 27(4):1030–1036, August 2014. ISSN 1000-9361. . URL https://www.sciencedirect.com/science/article/pii/S1000936114000442.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, January 2016. URL http://arxiv.org/abs/1511.06434. arXiv:1511.06434 [cs].

Jun Ren, Michael Kelly, and Lambertus Hesselink. Laser ablation of silicon in water with nanosecond and femtosecond pulses. *Optics Letters*, 30(13):1740–1742, July 2005. ISSN 1539-4794. . URL https://opg.optica.org/ol/abstract.cfm?uri=ol-30-13-1740.

Donatas Repecka, Vykintas Jauniskis, Laurynas Karpus, Elzbieta Rembeza, Irmantas Rokaitis, Jan Zrimec, Simona Poviloniene, Audrius Laurynenas, Sandra Viknander, Wissam Abuajwa, Otto Savolainen, Rolandas Meskys, Martin K. M. Engqvist, and Aleksej Zelezniak. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333, April 2021. ISSN 2522-5839. . URL https://www.nature.com/articles/s42256-021-00310-5.

Antonio Riveiro, Anthony L. B. Maçon, Jesus del Val, Rafael Comesaña, and Juan Pou. Laser Surface Texturing of Polymers for Biomedical Applications. *Frontiers in Physics*, 6, 2018. ISSN 2296-424X. URL https://www.frontiersin.org/articles/10.3389/fphy.2018.00016.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 9783319245744. .

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015. ISSN 0920-5691, 1573-1405. . URL http://link.springer.com/10.1007/s11263-015-0816-y.

G. Ryk, Y. Kligerman, and I. Etsion. Experimental Investigation of Laser Surface Texturing for Reciprocating Automotive Components. *Tribology Transactions*, 45(4): 444–449, January 2002. ISSN 1040-2004. . URL https://doi.org/10.1080/10402000208982572.

Tetsuo Sakai, Nikolay Nedyalkov, and Minoru Obara. Friction characteristics of submicrometre-structured surfaces fabricated by particle-assisted near-field enhancement with femtosecond laser. *Journal of Physics D: Applied Physics*, 40(23): 7485–7491, November 2007. ISSN 0022-3727. . URL https://doi.org/10.1088/0022-3727/40/23/035.

Michele Scaraggi, Francesco P. Mezzapesa, Giuseppe Carbone, Antonio Ancona, Donato Sorgente, and Pietro Mario Lugarà. Minimize friction of lubricated laser-microtextured-surfaces by tuning microholes depth. *Tribology International*, 75: 123–127, July 2014. ISSN 0301-679X. . URL https://www.sciencedirect.com/science/article/pii/S0301679X14001169.

D. Schuocker. Laser Cutting. *Materials and Manufacturing Processes*, 4(3):311–330, January 1989. ISSN 1042-6914. . URL https://doi.org/10.1080/10426918908956297.

Sakib Shahriar. GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network. *Displays*, 73:102237, July 2022. ISSN 0141-9382. . URL https://www.sciencedirect.com/science/article/pii/S0141938222000658.

Torgyn Shaikhina and N. A. Khovanova. Handling limited datasets with neural networks in medical applications : a small-data approach. *Artificial Intelligence in Medicine*, 75:51–63, January 2017. ISSN 0933-3657. . URL http://doi.org/10.1016/j.artmed.2016.12.003. Publisher: Elsevier BV.

E. A. Shcherbakov, V. V. Fomin, A. A. Abramov, A. A. Ferin, D. V. Mochalov, and V. P. Gapontsev. Industrial grade 100 kW power CW fiber laser. In *Advanced Solid-State Lasers Congress (2013), paper ATh4A.2*, page ATh4A.2. Optica Publishing Group, October 2013. . URL https://opg.optica.org/abstract.cfm?uri=ASSL-2013-ATh4A.2.

Donna Strickland and Gerard Mourou. Compression of amplified chirped optical pulses. *Optics Communications*, 56(3):219–221, December 1985. ISSN 0030-4018. . URL https://www.sciencedirect.com/science/article/pii/0030401885901208.

Alexey Strokach and Philip M. Kim. Deep generative modeling for protein design. *Current Opinion in Structural Biology*, 72:226–236, February 2022. ISSN 0959-440X. . URL https://www.sciencedirect.com/science/article/pii/S0959440X21001573.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015. . ISSN: 1063-6919.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, June 2016. . ISSN: 1063-6919.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, February 2017. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806.

Jianan Tang, Xiao Geng, Dongsheng Li, Yunfeng Shi, Jianhua Tong, Hai Xiao, and Fei Peng. Machine learning-based microstructure prediction during laser sintering of alumina. *Scientific Reports*, 11(1):10724, May 2021. ISSN 2045-2322. . URL https://www.nature.com/articles/s41598-021-89816-x.

Gábor J. Tóth, Tamás Szakács, and András Lörincz. Simulation of pulsed laser material processing controlled by an extended self-organizing Kohonen feature map. *Materials Science and Engineering: B*, 18(3):281–288, April 1993. ISSN 0921-5107. . URL https://www.sciencedirect.com/science/article/pii/092151079390144C.

A. A. Voevodin and J. S. Zabinski. Laser surface texturing for adaptive solid lubrication. *Wear*, 261(11):1285–1292, December 2006. ISSN 0043-1648. . URL https://www.sciencedirect.com/science/article/pii/S0043164806001335.

D von der Linde, K Sokolowski-Tinten, and J Bialkowski. Laser–solid interaction in the femtosecond time regime. *Applied Surface Science*, 109-110:1–10, February 1997. ISSN 0169-4332. . URL https://www.sciencedirect.com/science/article/pii/S0169433296006113.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, June 2018. . ISSN: 2575-7075.

Alexander Watson. Deep Learning Techniques for Super-Resolution in Video Games, December 2020. URL http://arxiv.org/abs/2012.09810. arXiv:2012.09810 [cs, eess].

Chong-Yaw Wee and Raveendran Paramesran. Measure of image sharpness using eigenvalues. *Information Sciences*, 177(12):2533–2552, June 2007. ISSN 0020-0255. . URL https://www.sciencedirect.com/science/article/pii/S002002550700014X.

Paul John Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1975.

Yunhui Xie, Daniel J. Heath, James Grant-Jacob, Benita MacKay, Michael McDonnell, Matthew Praeger, Robert Eason, and Benjamin Mills. Deep learning for the monitoring and process control of femtosecond laser machining. *Journal of Physics: Photonics*, 1(3):1–10, June 2019. ISSN 2515-7647. URL https://eprints.soton.ac.uk/432292/.

Basem F. Yousef, George K. Knopf, Evgueni V. Bordatchev, and Suwas K. Nikumb. Neural network modeling and analysis of the material removal process during laser machining. *The International Journal of Advanced Manufacturing Technology*, 22(1): 41–53, September 2003. ISSN 1433-3015. . URL https://doi.org/10.1007/s00170-002-1441-9.

Jinping Zhang, Yuping Chen, Mengning Hu, and Xianfeng Chen. An improved three-dimensional two-temperature model for multi-pulse femtosecond laser ablation of aluminum. *Journal of Applied Physics*, 117(6):063104, February 2015. ISSN 0021-8979. . URL https://aip.scitation.org/doi/10.1063/1.4907990. Publisher: American Institute of Physics.

Yongqiang Zhang, Wuzhu Chen, Xudong Zhang, Yanhua Wu, and Qi Yan. Synthetic evaluation and neural-network prediction of laser cutting quality. In *Lasers in*

Material Processing and Manufacturing II, volume 5629, pages 237–246. SPIE, January 2005. . URL https://www.spiedigitallibrary.org/conference-proceedings-of-spie/5629/0000/Synthetic-evaluation-and-neural-network-prediction-of-laser-cutting-quality/10.1117/12.575008.full.

Zhen Zhao, Ashley Kleinhans, Gursharan Sandhu, Ishan Patel, and K. P. Unnikrishnan. Capsule Networks with Max-Min Normalization, March 2019. URL http://arxiv.org/abs/1903.09662. arXiv:1903.09662 [cs].

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, Venice, October 2017. IEEE. ISBN 978-1-5386-1032-9. . URL http://ieeexplore.ieee.org/document/8237506/.