UNIVERSITY OF BIRMINGHAM

THESIS

SUBMITTED TO THE UNIVERSITY OF BIRMINGHAM FOR THE DEGREE OF PHD

Vision-based Monitoring System for High Quality TIG Welding

Daniel Bacioiu

supervised by

Dr. Mayorkinos PAPAELIAS

School of Metallurgy and Materials, University of Birmingham

Mr. Geoff MELTON

TWI Ltd.

August 11, 2019

Acknowledgements

Firstly, I would like to express my sincere appreciation to my supervisors, Dr. Mayorkinos Papaelias and Mr. Geoff Melton for their support during my entire Ph.D program, for their guidance and patience as well as the knowledge shared. Their support stretches from day one of the program to the day I submitted this thesis and their advice helped and shaped my thinking. I could not have imaged reaching this stage without their support.

Besides my advisors, I would like to thank project leaders and technicians from the Arc Welding and Engineering (AWE) department at TWI. They helped and advised me during the time in the engineering hall. Without the resources provided to me by TWI it would not have been possible to carry out my research.

Sincere thanks go to my colleagues Rob Shaw and Amina Salman for their editorial advice. My research papers read and explain much better my research thanks to their input.

Last but not least, I would like to thank my partner Vanessa Cardoso for supporting me spiritually throughout the program's ups and downs. I would also like to express my utmost gratitude to my family Georgeta Bacioiu, Dumitru Bacioiu and Sergiu Bacioiu for their continuous, relentless and unconditional support throughout my entire life.

Publications list

- Daniel Bacioiu, Geoff Melton, Mayorkinos Papaelias, Rob Shaw. Automated defect classification of Aluminium 5083 TIG welding using HDR camera and neural networks. In *Journal of Manufacturing Processes*, 2019

- Daniel Bacioiu, Geoff Melton, Mayorkinos Papaelias, Rob Shaw. Automated defect classification of SS304 TIG welding process using visible spectrum camera and machine learning. In *NDT & E International*, 2019

- Daniel Bacioiu, Geoff Melton, Mayorkinos Papaelias, Rob Shaw. On-line weld defect classification of SS304 TIG joining process using machine learning. *2018 NSIRC Annual Conference, At Granta Park, Cambridge, UK*

- Daniel Bacioiu, Geoff Melton, Mayorkinos Papaelias, Rob Shaw. Initial steps towards an automated system for monitoring TIG welding process, *First World Congress on Condition Monitoring - WCCM 2017, At London, UK*

- Daniel Bacioiu, Geoff Melton, Mayorkinos Papaelias, Rob Shaw. TIG process online monitoring with Artificial Neural Networks. *2017 NSIRC Annual Conference, At Granta Park, Cambridge, UK*

Contents

List of Figures

List of Tables

Synopsis

The current study evaluates an automatic system for real-time arc welding quality assessment and defect detection. The system research focuses on the identification of defects that may arise during the welding process by analysing the occurrence of any changes in the visible spectrum of the weld pool and the surrounding area. Currently, the state-of-the-art is very simplistic, involving an operator observing the process continuously. The operator assessment is subjective, and the criteria of acceptance based solely on operator observations can change over time due to the fatigue leading to incorrect classification.

Variations in the weld pool are the initial result of the chosen welding parameters and torch position and at the same time the very first indication of the resulting weld quality.

The system investigated in this research study consists of a camera used to record the welding process and a processing unit which analyse the frames giving an indication of the quality expected.

The categorisation is achieved by employing artificial neural networks and correlating the weld pool appearance with the resulting quality. Six categories denote the resulting quality of a weld for stainless steel and aluminium. The models use images to learn the correlation between the aspect of the weld pool and the surrounding area and the state of the weld as denoted by the six categories, similar to a welder categorisation. Therefore the models learn the probability distribution of images' aspect over the categories considered.

Chapter 1

Introduction

1.1 General background

In the current technological landscape, computer vision for monitoring industrial processes and quality control have advanced into more fields, leveraging the computational power of new hardware and freeing labour resources. Several industrial sectors and various applications have benefited from the advances in computer vision to remove the need for human labour to perform simple and repetitive work and even potentially dangerous and hazardous tasks, speeding up production and improving manufacturing quality.

Welding is one particular area where technological innovation and automated computer vision have seen limited advances in recent years. Until now, the hardware processing speed has been the main limitation in establishing more accurate monitoring systems which make use of high resolution images, high frame rates, or employing complex image interpretation algorithms. Most of the research in quality control for arc welds focuses on post-weld non-destructive testing (NDT). Although post-inspection is useful for detecting defects, once they are identified, usually the weld needs repairing, which is expensive in terms of time and financial resources. More accurate real-time identification is required to drive the cost down and decrease the time required to complete the welding process.

Computing hardware and processing have improved rapidly and become commonplace while the cost significantly reduced. The economies from hardware contribute profoundly to making the cost of a real-time welding monitoring system more justifiable. Moving the assessment from post-weld to on-line, i.e. at the time when the component is being welded, improves not only the time of detection but also the amount of work required to correct the defect.

New technologies and techniques for image processing have emerged in recent years [1], improving the accuracy of feature extraction and even allowing automatic feature extraction, made possible by machine learning. The use of customised hardware together with advanced image processing techniques allow the analysis of large amounts of data within very short periods. In particular, the use of the Graphics Processing Unit (GPU) accelerated image filtering, through the application of the same command over the entire matrix of pixels simultaneously. Parallelising an elementary, time-consuming but straightforward operation, brought a reduction in the amount of time needed for image processing. On the software side, there has been a leap in terms of vectorisation, coalescing operations into vectors and matrices in order to take advantage of the parallel architecture of the hardware.

The approach undertaken in current research was made possible primarily by the advances in hardware and software, but the problem of weld quality assessment requires more than processing hardware and software.

Environmental aspects are essential in monitoring the arc welding process. The arc emits strong ultraviolet (UV) light and electromagnetic waves while the electronic equipment, sensor and processing unit are susceptible to electromagnetic fluctuations. Adequate electrical insulation is required to prevent damaging the equipment or distorting the image acquisition. Humidity, vibrations and dust are additional factors which need to be taken into account when designing the system. Space represents a constraint in some applications, such as narrow gap welding [2] [3] where hardware miniaturisation plays a critical role. Images acquisition in an industrial environment with strong and local illumination adds further to the challenges the envisioned system needs to overcome.

The current study aims at researching a system, combining hardware and software for identifying welding defects using imaging sensors in real-time, during welding, similar to an experienced welder. The research focused on several aspects in tackling this problem:

- the **welding problem**: the identification of a suitable arc welding process that provides stability and luminosity suitable to an imaging sensor

- the **imaging sensor problem**: the choice of a sensor that can overcome the excessive luminosity contrast produced by the arc process revealing details from the weld pool and surrounding area

- the **processing problem**: the identification of a processing paradigm, a combination of processing hardware and software with the potential of real time evaluation of the welding condition, although the model deployment step is not considered.

1.2 Contributions and Outline

The content of this dissertation details the approach for linking the welding conditions to a categorisation representative of the welding state.

Chapter 2 builds the background and literature review for the three problems addressed. Section 2.1 looks into different welding processes with a particular focus on TIG welding, defects and material. Section 2.2 takes a look at the current image sensors available on the market, highlighting the desideratum and how different commercially available sensors compare with the ideal sensor. Sections 2.3 and 2.4 shift the attention to the processing paradigms, describing in details the state of the art in the image processing as well as mathematical background for the paradigm chosen for this study, machine learning.

Chapter 3 details the system design choices and the system's general layout starting with the selection of imaging sensor (Sections 3.1 and 3.2). Section 3.3 provides details of the welding process application on the two types of material used stainless steel 304 (SS304) and aluminium alloy 5083 (Al5083), while Section 3.4 shows the images aspect and dataset statistics. The second half of the chapter, Sections 3.5, 3.6, 3.7 and 3.8 details the machine learning parameters choices influencing optimisation, the optimiser selected and the evaluation metrics employed in this study.

Chapters 4 to 6 showcase the overall system performance when applied to the SS304 (Chapter 4), Al 5083 (Chapter 5), and how the knowledge is transferred from Al 5083 to SS304 via neural architecture search (NAS) (Chapter 6). The chapters analysis move gradually from the high-level performance metrics to the granular models quality assessment, comparing fully connected neural networks (FCN), convolutional neural networks (CNN) and neural networks typologies generated by NAS.

Chapter 7 summarises the progress made during this research project, highlighting the advances in the image quality, data acquisition methodology and neural networks models design and optimisation. The last thesis paragraphs gives a broad vision on the future directions for the current research study.

Chapter 2

Background and Literature review

This chapter provides the background and literature review of the main aspects involved in assessing the welding quality. The chapter reviews the welding processes taken into consideration for the quality assessment, describes the physical phenomenon underpinning each process and highlights the specific advantages. Extensive coverage describes the TIG welding process due to its broad applicability and adoption. Included is a non-exhaustive list of welding defects and causes while the material description completes the welding section.

Data acquisition is an integral part of the study and one aspect differentiating the current study from previous research. The literature dedicated to the weld monitoring varies broadly on the type of data used in assessing welds. In the previous studies the types of parameter acquired during the arc welding process include the welding current, arc voltage, welding speed and pass number [4]. Such parameters are easy to record and do not require a significant amount of storage and processing power. Different welding processes require a different set of parameters, laser welding, for example, has different characteristics and the information acquired include the laser power, welding speed and focal diameter [5].

The current study takes a different approach in terms of the type of signal acquired and the acquisition time. Visual signal is a rich source of information revealing the state of the weld pool in real time, carrying at the same time the very first indication of defect formation. Imaging sensors for welding applications are rare, and they combine in many cases with additional illumination to bridge the gap between dark background and excessively bright arc light. The alternative to camera plus laser illumination is a different sensor that absorbs the light non-linearly and employing post-processing, generating an image with a high dynamic range (HDR). The study looks at different alternatives for acquiring the visual signal, highlighting the advantages and disadvantages.

The study covers the weld image analysis extensively, reviewing the state of the art for non-destructive testing (NDT) for different signals before delving into the image processing techniques applied to the weld images. Recent developments in machine learning paradigm and advances in hardware processing power placed the paradigm at the forefront of image classifications. The review explains the artificial intelligence (or machine learning) background, and in particular, neural network architectures applied on image processing with the focus on welding as well as studies and applications where the paradigm is applied.

2.1 Welding

2.1.1 Welding Processes

Welding arc physics

The welding arc occurs when a potential difference exists between the electrode and a workpiece. The electrons are transported from the electrode to the workpiece via a medium composed of ionised gas called plasma [6]. Plasma is a state of matter composed of free electrons and ions. The plasma generally forms when atoms lose an electron resulting in a positively charged particle called ion. The heat introduced by the arc keeps the plasma temperature high facilitating atoms collision, which in turns produces free electrons and ions. The charged particles are accelerated in the electric field transferring the energy to the workpiece. The shielding gas displaces the local environment atmosphere and replaces it with gases used during welding, usually pure argon, helium, hydrogen, nitrogen, oxygen, or a combination of two or three. The primary purpose for the shielding gas is the isolation from the local atmosphere, but at the same time, the choice influences the arc ignition, heat transfer, metal transfer mode, penetration and bead shape [7] [8]. The shielding gas also influences the temperature required to maintain the matter's plasma state. Helium has a higher ionisation energy threshold than argon, requiring higher temperatures, while at the same time produces a higher voltage drop and transfers more heat input to the weld pool. The multi-atomic gasses as CO_2 initially dissociate into the composing atoms before losing electrons and becoming plasma. The split into C and O requires energy, usually taken from hotter part of the arc, and loses energy through reassociation in the colder parts of the arc. The whole process of dissociation and reassociation leads to increase of arc's thermal conductivity.

In real settings, the welding involves the addition of extra material, in the form of feeding wire, to the weld pool generated from melting the component. The different interaction between arc, wire and component describes the characteristics, advantages and disadvantages for every welding process described below.

MIG/MAG welding

Metal Inert Gas (MIG) and Metal Active Gas (MAG) welding are processes where the arc strikes between a consumable wire and the workpiece as in Figure 2.1. The difference between inert and active arises from the shielding gas used: argon and helium being inert gases and CO_2 an active gas [9]. The process claims the largest share in welding processes of choice in most industrial countries due to several important features:

- low heat input - capable to weld plate of 0.5mm with minimum distortion
- high deposition rate - improving productivity
- applicable on a wide range of materials: steel of most grades, aluminium, copper, nickel, etc.
- able to penetrate through coated materials - Zn-coated steel
- welding position versatility

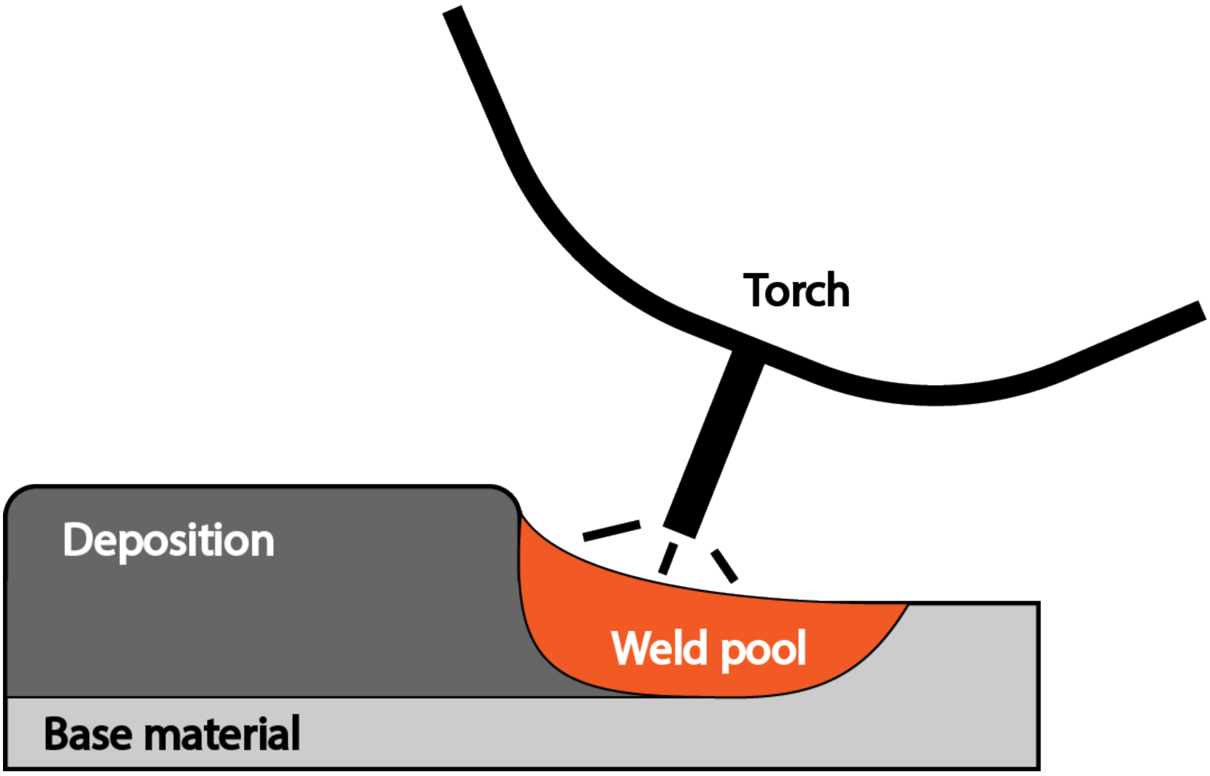


Figure 2.1: MIG welding process schematics

Depending on the values of the current and voltage, MIG welding is divided into three main transfer modes:

- short circuiting (dip transfer) - when low heat input required. In this transfer mode, the arc melts the tip of the wire which makes contact with the weld pool creating a short circuit. The strong electromagnetic forces pinch the molten material connecting the wire with the base material separating the two. The arc reignites, and a new cycle of arc-dip-pinch occurs.
- globular transfer - the electrode (wire) does not necessarily touch the base material, but the molten wire detaches and falls. The droplets dimensions are usually larger than feeding wire diameter.
- spray transfer - is similar to globular transfer except for the droplets dimension which are smaller than the feeding wire diameter. It requires higher voltage and current, and no short-circuit. Due to high heat input, spray transfer is applied on the base material of approximatively 5mm or above.

Plasma welding

Plasma welding torch, Figure 2.2, is composed of inner plasma gas and outer shielding gas. The arc strikes between a non-consumable electrode placed inside the inner tube and the base material. The plasma arc welding advantages are:

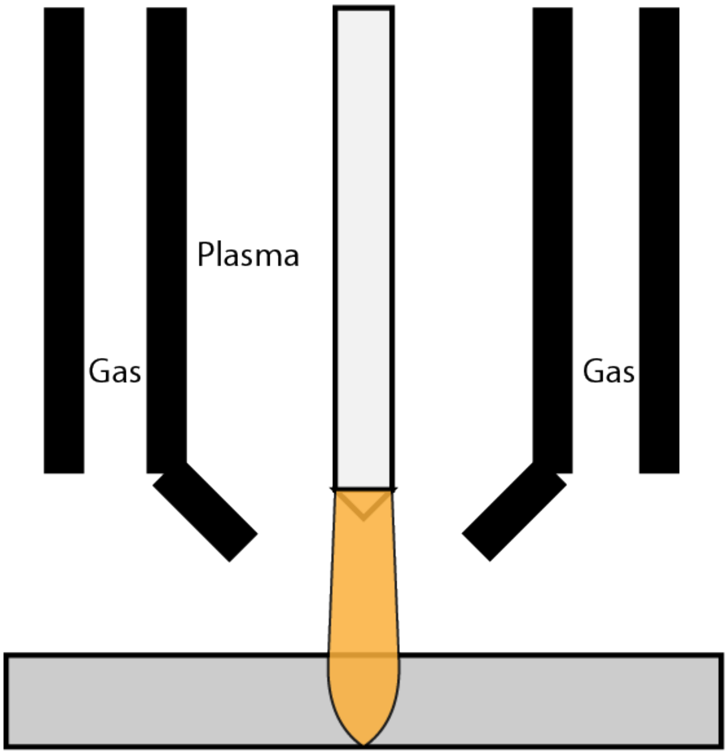


Figure 2.2: Plasma welding process schematics

- The concentration of plasma arc is higher than tungsten inert gas (TIG), therefore more insensitive to length variations
- higher welding speed - up to 5 times higher
- smaller heat affected zone (HAZ) and distortions
- low weld convexity
- capable of delivering keyhole welding and melt-in welds with the same equipment

Submerged arc welding

Submerged arc welding strikes an arc under a layer of powder flux covering the base material in the surrounding of the weld pool. The flux shields the molten metal from the reaction with the atmosphere, melting and forming a removable layer of slug. The submerged arc welding advantages are as follows:

- uniform, good quality and visually appealing welds
- the flux protects against arc light
- high productivity due to deposition rate
- deep penetration capability

Resistance welding

Resistance welding uses the property of the materials to resist when an electric current passes through in exchange of heat emission. The pieces are initially pressed together until physical contact, then using precise squeezing force an electric current passes between electrodes generating heat at the interface between the workpieces causing melt and fusion. The advantages of resistance welding are:

- processing time is short
- no consumable needed (shielding gas, feeding wire, etc.)
- no noise or harmful gases produced
- small heat affected zone and residual stress

Friction welding

In friction welding, the pieces are physically rubbed against each other producing heat and mix metallurgically. The surfaces, although visually smooth, have asperities at the microscopic level. The asperities resist the motion deforming plastically and elastically generating the heat required to bring the materials into a viscous state. There are several known types of friction welding processes:

- friction stir welding (FSW)
- friction stir spot welding (FSSW)
- linear friction welding (LFW)
- rotary friction welding (RFW)

Laser welding

Laser welding uses light to transfer the energy to the base metal for melting and fusing the material. The unique process characteristics are the flexibility of light transport from the source to the desired location using mirrors and glass fibres and the energy concentration delivered. The lenses concentrate the ray into a few tenths of a millimetre on the surface of the material or immediately under the surface melting the material instantaneously and even vaporising it, producing a keyhole. Due to the small area affected, the material solidifies very quickly after the light beam moves, producing a very small heat affected zone and minimal distortion. The process is twice as fast as plasma welding and up to eight times faster than TIG welding. The source of the light is commonly composed of CO_2 or Nd:YAG (dopant neodymium (Nd) in a transparent rod of yttrium aluminium garnet (YAG)), latter producing a 1.06μm wavelength and is used for thinner material while the former produces 10.6μm wavelength and it is suitable for thicker components.

Electron-Beam (EB) welding

EB welding uses electrons to join the workpieces together. The thin column of electrons, i.e. electron beam, channels from the source to the metal via a vacuum to prevent the electron absorption by the air. The beam energy concentration is even higher than laser welding while the energy conversion from electrical input to beam output is very high. The vacuum, although a drawback for scaling the process to large components, provides the benefit of assuring no interaction between the atmosphere and the welded part is taking place. The advantages of EB welding are:

- capable to weld up to 250mm components, as well as very thin material
- the welding speed is much higher than TIG welding, although the process efficiency is hampered by the vacuum requirement
- weld without risk of atmospheric contamination
- low heat input therefore low residual stresses
- capable of welding difficult material

2.1.2 TIG Welding Process

Tungsten Inert Gas (TIG), Figure 2.3, is a welding process invented in 1940. It works on the principle of heat transfer from the tungsten electrode to the workpiece. The polarity of the process influences the resulting welding quality with steel requiring straight polarity (negative electrode, positive workpiece) for optimum penetration and minimum tungsten wear while aluminium and magnesium require alternating polarity for oxide layer removal. The isolation of the process from gases naturally occurring in the air is achieved by shielding it with argon or helium. Helium produces deeper penetration although it comes at an extra financial cost. Most of the applications use pure argon or a mixture of helium and argon. The main factors influencing the bead shape and subsequently the quality of the weld are current, welding speed, arc length, shielding gasses, filler metals and electrode tip angle [10].

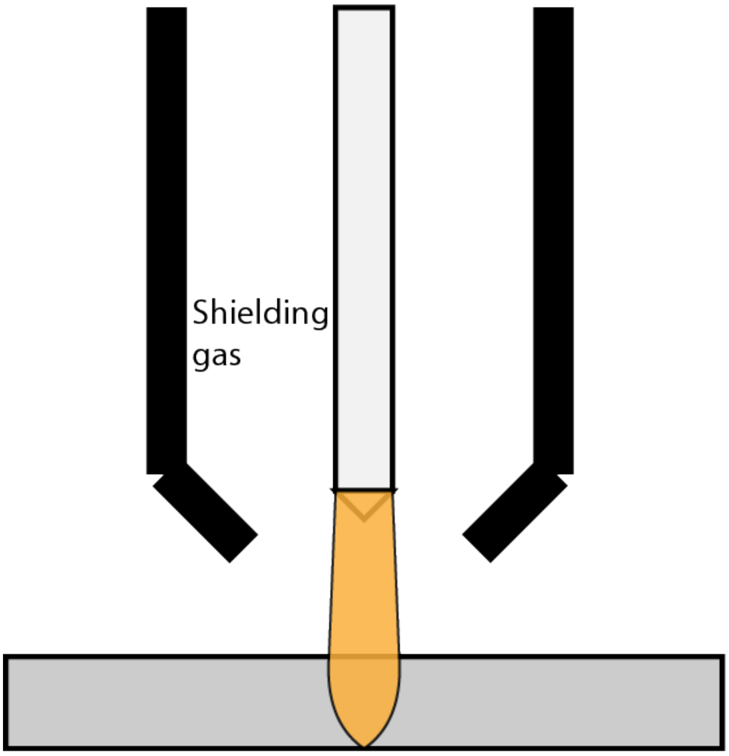


Figure 2.3: TIG welding process schematics

Current

The current is the most influential parameter having a direct impact on bead geometry, quality of the weld and productivity. The current modes utilised are direct current in electrode negative (DCEN) giving rise to a higher degree of penetration and travel speed, while the opposite mode, direct current electrode positive (DCEP) generates more heating and wear of the electrode. Reverse polarity (positive electrode, negative workpiece) is mostly used when welding aluminium or magnesium due to oxide layer cleaning characteristics, although most of the time alternating current is the preferred approach because it provides improved control of the heat transfer and has a cleaning effect.

Travel speed

The welding speed influences the heat transfer to the workpiece. An increase in welding speed correlates to a reduction of the heat input per unit of area and subsequently a decrease in weld cross-section area (penetration depth and weld width decrease). The ratio between depth and width, on the other hand, presents limited dependence on travel speed leading to the conclusion that travel speed is uncorrelated to weld pool formation, it only affects the volume of melted metal.

Arc length

Arc length increase translates to reduced penetration and cross section area due to radiation loss in the arc column. An increase in arc length has to be assisted by an increase in voltage to maintain the stability of the arc.

Shielding gas

Shielding gas provides isolation from the atmosphere which can lead to porosity, scaling, change in chemical composition or weld cracking and affect the stability of the arc. Low ionising gases improves the electric arc ignition while gases with low thermal conductivity promote arc stability. Stainless steel and thin aluminium alloys welding require argon. Helium requires higher ignition and maintenance voltage but has higher heat input transfer, suitable for thick aluminium parts and copper alloys. A small amount of hydrogen (up to 5%) is used for austenitic stainless steel to increase the heat input transfer and consequently the penetration and travel speed.

Electrode tip angle

The electrode in TIG welding is non-consumable, and the tip angle affects weld shape and penetration depth. Small angles (30°-60°) increase penetration depth and arc pressure and have high degradation while larger angles (60°-120°) provides acceptable depth-to-width ratio while decreasing the tip degradation.

2.1.3 Defects

The weld defects are flaws in the weldment that renders the resulting product unsuitable for the intended application. The confusion between imperfection and defects is a common mistake. The application standards and welding procedure qualification standards define the limits and classification of imperfections as defects. The task of navigating between standards could be a daunting one because there are British Standards (BS), European Standards (EN), standards issued by International Standards Organisation (ISO) as well as American Welding Society (AWS) standards and American Society of Mechanical Engineers (ASME) standards. Some of them overlap, some are identical, while others are entirely different. Each body issuing standards tends to categorise them in two main categories: application standards and welding standard. Welding standards are further sub-classified into procedure qualification standard and welder qualification standards. The type of imperfections resulting as a consequence of applying the TIG welding process are numerous with varying degree of severity and are classified as defects only in relation to a standard. This study covers an overview of the main types of defects present in the weld without giving details in respect to the imperfections tolerances, which are standard specific. The following list of imperfections is non-exhaustive and the imperfections analysed in the current research study do not correspond necessarily to workpiece defects, but they can represent anomalies in the welding procedure which will eventually lead to imperfection.

Lack of penetration

Lack of penetration defect is related to the failure of molten material to penetrate the workpiece. TIG process decouples the material deposition and penetration; therefore it is less susceptible to lack of penetration. Usual causes for this type of defects are: thick root face, root gap too small, insufficient heat input, misplaced welds, insufficient power input.

Incomplete fusion

Lack of fusion defects results due to insufficient melting of the previous layer or base material. In order to avoid the lack of fusion, two main parameters have to be correlated: melting rate and welding speed. Varying heat input and welding speed proportionally result in a good weld. Another critical aspect for avoiding lack of fusion is choosing the correct torch angle. A positive torch angle leads to wider, shallower welds, while negative torch angle results in deeper, narrower beads.

In case of I-grooves, the inclination of torch influences the transfer of energy from the torch to the base material, thus resulting in extra penetration in the wall the torch points towards and insufficient penetration in the opposite wall. The presence of oxides or foreign materials that prevents the fusion between weld and base metals also generates incomplete fusion. Poor design of the groove, either the angle or the size could prevent positioning adequately the electrode or block shielding gas leading to incomplete fusion.

Porosity

The presence of foreign gases within the weld denotes another type of defect, pores. Pores can form in two ways: mechanically and metallurgically. Primary sources of contamination are moisture or dirty wire and base materials. Under intense heat, the moisture vapourises and absorbs into the molten metal. The driver for pores formation is the speed of metal solidification (crystallisation speed). As the metal solidifies, the gas solubility reduces, pushing out the gas from the solid material. With the crystallisation speed slow enough, the bubbles have time to surface and escape to the atmosphere. The crystallisation front traps the bubbles in the metal in case it moves too fast.

Undercut

Undercut defects are grooves formed in the workpiece along the weld edge. An incorrect combination of travel speed and arc voltage generates this type of defect. High travel speed, in particular, promotes the formation of the undercut forming a peaked bead due to rapid solidification. The molten metal is drawn from the weld pool edges by the surface tension towards the centre of the weld. The solution for fixing the undercut is a decrease in travel speed or raise in voltage.

Cracking

Cracking could occur within a weld bead, on the top or within the base material. This type of defect occurs due to excessive heat supplied to the material. The material most prone to cracking is aluminium due to the weld pool cooling while the base material is still hot. Steels with a high content of carbon ($\geq$ 0.4 wt%) are also more likely to crack because of their hardness. The imbalance in microstructure (segregation) can also lead to an increased likelihood of cold cracks. Another factor in the formation of cold cracks is the presence of hydrogen, generated by very fast solidification process. The cold crack initiation is due to residual stresses or a combination of residual stresses and load stresses. However, given cold cracking occurs post-welding (sometimes hours later), it may not be suitable for vision system detection.

2.1.4 Material

The current study uses two types of metals for investigating the processing paradigm feasibility on identifying defects. Stainless steel and aluminium components are widely used in industry and have different welding procedures. The difference in aspect and welding produces an excellent example of the paradigm's adaptive capability.

Stainless steel

Humans make use of iron for 7000 years (5000 BC) [11], the testimony being the discovery of iron beads in graves in Egypt and with evidence, people smelted iron in Mesopotamia around the same time. Although initially it was used for ceremonies, around 1300-1200 BC it transitioned towards mass production with the advent of Iron age. Later on, perhaps initially accidentally, the iron was combined with carbon, forming steel. The abundance of iron on earth probably played a role in its discovery. The earth is believed to contain 30% iron, most of it placed in the core.

The most abundant iron ore is **hematite** (Fe_2O_3). Pure iron melts at 1539°C and can be combined with various elements as carbon, chromium, nickel, manganese, molybdenum, vanadium, and silicon to form alloys [12].

The mixture of iron and 0.02% to 2.11% carbon, as mentioned earlier, is called steel. The category of interest in the current study is stainless steel. Stainless steel is a mixture of iron, carbon and at least 15% chromium. Chromium combines with the oxygen covering the workpiece with an oxide layer preventing further corrosion. Nickel is another element improving steel's corrosion resistance. An increase in the amount of carbon increases the strength of the alloy as well as hardness, but on the other hand, it reduces the free chromium content available in the alloy by forming chromium carbide which reduces corrosion resistance properties. Stainless steel performs well in tests measuring strength and ductility making it a good candidate for many applications, although its increased manufacturing processing difficulty. Generally, stainless steel splits into three main categories, as seen in Table 2.1: austenitic, ferritic and martensitic stainless.

- **Austenitic stainless steel** is composed of roughly 18% Cr and 8% Ni, also denoted 18-8. The nickel in the composition has the property of enlarging austenite region in the iron-carbon phase diagram, facilitating the structure of austenite to exist at room temperature. They are the most corrosion resistant of the three categories, very ductile, nonmagnetic and used to fabricate food and chemical processing equipment as well as within applications requiring a high degree of corrosion resistance.

- **Ferritic stainless steel** grades have the composition of 15-20% chromium, very low carbon content and no nickel. Therefore the ferritic phase is present at room temperature. Compared to austenitic stainless, they are magnetic, less ductile and less corrosion resistant, being used mostly for kitchen utensils and jet engine parts.

- **Martensitic stainless steel** is richer in carbon compared to ferritic stainless opening the possibility of strengthening using heat treatment. The percentage of Cr stands at 18% while Ni is non-existent.

Table 2.1: Stainless steel grades composition [12]. In this study grade 304 is used for some experiments.

Chemical Analysis, %							Tensile Strength		
Grade	Fe	Cr	Ni	C	Mn	Other[a]	MPa	lb/in^2	Elongation, %
Austenitic									
302	71	18	8	0.15	2		515	75,000	40
304	69	19	9	0.08	2		515	75,000	40
316	65	17	12	0.08	2	2.5 Mo	515	75,000	40
Ferritic									
405	85	13		0.08	1		415	60,000	20
430	81	17		0.12	1		415	60,000	20
Martensitic									
403	86	12		0.15	1		485	70,000	20
403[b]	86	12		0.15	1		825	120,000	12
440	81	17		0.65	1		725	105,000	20
440[b]	81	17		0.65	1		1790	260,000	5

[a] Element in composition exceeding 1%. All grades contain approximatively 1% silicon and very small amount (<1%) of other elements such as phosphorus and sulfur.
[b] Heat treated.

Since the development of traditional stainless steel in the 1900s, the categories have expanded to include **precipitation hardening stainless steel** and **duplex stainless steel**. Precipitation hardening stainless contains 17% Cr and 7% Ni and other elements as Al, Cu, Ti, Mo, maintains strength and corrosion resistance at higher temperatures and are strengthened by precipitation hardening. The principal application of these alloys is aerospace. Duplex stainless, on the other hand, maintains the equal balance between austenite and ferrite, inheriting austenitic grades' corrosive resistance and improving on stress-corrosion cracking resistance. The duplex stainless application includes long term nuclear waste storing containers and wastewater treatment plants.

Aluminium

Aluminium is a light element, abundant on earth but harder to separate from the ore. The main ore found on earth is **bauxite** which is an impure mixture of Al_2O_3 and $Al(OH)_3$ [12]. It was separated from ore in 1845 by the German physicist Friedrich Wohler after in 1807 the English chemist Humphrey Davy failed. The mass production based on the electrolytic process started in 1888 based on the work of Charles Hall and Paul Heroult. Aluminium forms alloys with copper, magnesium, manganese, silicon, and zinc detailed below. It has excellent thermal and electrical conductivity, and in combination with oxygen, it forms a thin oxide layer protecting the workpiece. Aluminium categorisation based on elements combination are as follows:

- **1XXX - pure aluminium** - it has good appearance and corrosion resistance, excellent weldability and formability, but low strength.
- **2XXX - Al + Cu** - it is strong, machinable, relatively satisfactory corrosion resistance, poor formability and difficult to weld.
- **3XXX - Al + Mn** - formable, resistant to corrosion, weldable and stronger than 1XXX series.
- **4XXX - Al + Si** - formable, relatively satisfactory corrosion resistance, weldable and wear resistant.
- **5XXX - Al + Mg** - strong, formable, excellent corrosion resistance, weldable
- **6XXX - Al + Mg + Si** - strong, formable, good corrosion resistance, weldable
- **7XXX - Al + Zn, Mg & Cu** - very high strength, machinable, relatively satisfactory corrosion resistance, poor weldability
- **8XXX - Al + other**

The heat treatment and work hardening influence most aluminium grades' properties.

2.2 Imaging sensors

The current study approaches the welding condition monitoring from the perspective of visual imaging. The visual imaging involves the utilisation of a sensor for acquiring images of the welding conditions i.e. torch, weld pool and the surrounding solidified metal, in the spectrum similar to the human vision.

It is important to highlight that the light emission in the visible spectrum is not the only signal generated by a welding process. The other signals are the welding process input parameters and the resulting spectral emissions and bead visual characteristic. A non-exhaustive list of recordable signals generated by the welding process are:

- voltage
- current
- wire feed speed
- travel speed
- shielding gas flow rate
- torch position
- acoustic emissions
- thermal emissions
- groove geometry
- bead geometry

The signals list includes preprocess, in-process and post-process information and choosing a signal for analysis depends on the scope definition. Current and voltage, for example, are useful post welding, although they are best analysed during welding when paired with the adaptive control of the welding process. On the other hand, a thermal camera could capture the material cooling rate, post welding, potentially revealing information related to the internal stresses introduced by the welding process.

Related to vision, the welding processes, where an arc is present, emits light in a wide range of frequencies, spanning from UV-C (200nm to 280nm), UV-B (280nm to 315nm), UV-A (315nm to 400nm) to visible radiation (400nm to 780nm) and infrared (IR) radiation (780nm to 1mm) [13]. The amount of the radiation emitted in each band varies, depending on the welding process chosen, i.e. TIG, MIG, the welding current and the welded metal. Generally, there are higher levels of radiations emitted in the UV and near-infrared bands. Due to discrepancies in the level of emissions at different bands, the cameras for welding processes struggle to mitigate the effects of the bright arc and at the same time render an accurate image of the welding emissions.

The approaches to diminishing the excessive light emitted from the arc and welding pool split into three main categories:

- the filtering of specific wavelengths bands. The approach performs global filtering (therefore the entire image or record is affected) or partial filtering for the area of the image where the arc and weld pool is present.
- bridging the gap of luminosity by shining light over the process using a laser diode
- using non-linear light to electrons conversion sensor. The sensor absorption characteristic is logarithmic, i.e. non-linear, having the benefit of reducing the electron emission for a very bright area preventing the sensor from saturating. The technology name is high dynamic range (HDR) because the range of the grayscale representation of luminosity is not fixed and cover higher order of magnitudes that linear range.

This work scopes the selection of off-the-shelf camera systems for recording the welding process.

Cavitar [14] specialises in diode laser illumination. The expansion into imaging sensors follows the route of bridging the luminosity gap by lighting up a laser over the weld pool in order to offset the arc light and illuminate the entire scene. Cavitar takes advantage of the laser's high spectral brightness and careful filtering of thermal light. The technology demonstrates a ”cold” look appearance of the weld pool and surrounding area when applied on the arc welding processes and laser welding. The package of camera, integrated laser illumination and optics are all embedded into a one casing increasing the technology portability and usability.

Oxford lasers [15] employs a similar approach as Cavitar, using a laser to illuminate, except for the high frame rate recording with short pulses of illumination. The laser operates in a narrow bandwidth but with very high intensity. The high-speed camera is behind a filter which excludes all the radiation emitted by the subject under study and allows only the laser frequency to pass through. The approach effectively eliminates all the powerful radiation emitted by the arc, leaving the weld pool visible.

Invisuale [16] proposes an imaging sensor with HDR capability based on the logarithmic response of the sensor to the light.

Intertest [17] produces a camera dedicated to verification of the torch position, wire feed orientation and weld part alignment. The system is an end-to-end solution of camera-monitor using slightly older types of connectors, as PAL, not standard on electronic devices nowadays. Their focus is on accessing remote places using hardware miniaturisation, rather than providing a clean image of the weld pool. The camera's main advantage is a dynamic range of 140dB requiring no additional filter or illumination for in-process weld recording.

The camera of choice for providing the welding images is Xiris XVC-1000. It provides the high dynamic range for offsetting the powerful arc light and at the same time, it is compact, easy to interface with other equipment via Ethernet connection and equipped with an SDK (Standard Development Kit). The power is delivered to the camera through the Ethernet port, the same medium used to send frames back from the camera to the computer. Since the Ethernet port is universal and widely used in electronic devices, it facilitates the integration with other equipment. The SDK provides a platform for delivering the frames to a downstream system to automate processing. An important aspect for a camera is to provide clear images of the weld pool and arc by tempering the powerful arc light and enhancing the weld pool appearance. The feature offsetting the arc light in this instance is the High Dynamic Range (HDR). In this way, the resulting image looks more balanced and closer to human vision. Xiris offers a stand-alone camera, XVC-1000 [18] with the specifications detailed in Table 2.2.

Table 2.2: Xiris XVC-1000 specifications [18]

Image Sensor	2/3" Mono HDR CMOS
Speed/Resolution	Up to 55 FPS at 1280 (H) x 1024 (V) pixels
Dynamic Range	140+ dB
Bit Depth	12 bits
Image Data	Mono 8/16, Bayer 8/16
Shutter Range	1 µs - 53s Exposure

2.3 Non-destructive testing (NDT)

2.3.1 Traditional NDT techniques

The range of methods for identifying defects in welds, or more general anomalies within the structure of a component, includes radiographic inspection, magnetic particle inspection, ultrasonic testing, liquid penetrant inspection, Eddy current testing, acoustic inspection and visual inspection.

Generally, the majority of the testing is performed after welding the component, without the possibility for the process interruption or alteration for the flaw correction. Although the range of testing includes a wide variety of possibilities, every technique exhibits advantages and limitations, specific to the targeted combination of test-component pair. Over the past decades, a number of sensing approaches have been proposed in the literature as a means to monitor, in real-time, the weld pool. Among these approaches are the pool oscillation method [19, 20], ultrasonic testing [21–23], infrared sensing [24, 25] and specular weld pool surface [26, 27].

Radiographic inspection [28] uses short electromagnetic waves to scan the workpieces. The base material absorbs different amount of the radiation in comparison to the holes or material inclusions, creating a greyscale intensity difference on the film placed behind the workpiece. The disadvantages, in this case, are the equipment size, radiation hazard, access requirement to both sides, unsuitability for certain geometries (e.g.tee joint) and skilful operator requirement for the scan interpretation.

Magnetic particle inspection [29–33] is applicable on the ferromagnetic materials for detecting surface and near-surface defects. The method is based on the flux leakage occurring at a flaw, due to the magnetic field leaving the component. Therefore, the subsurface flaws magnetic testing could detect defects located immediately under the surface. The component requires partial or full demagnetisation at the end of the process.

Ultrasonic testing [34–36] is a technique for assessing non-destructively a workpiece using sound waves (ultrasound) at high frequency, typically higher than the human sensitivity range. The flaw identification derives from the speed with which the sound wave travels through material [37], typically 6.3km/s in aluminium and 5.8km/s in stainless steel. The piezoelectric devices generate waves while, at the same time, the same device acquires the reflected waves applying the reverse logic. Piezoelectric devices convert electric energy into mechanical energy by changing their shape as a consequence of electrical excitation. The quartz is an example of naturally occurring material with piezoelectric properties. The ultrasonic technique requires calibration and complex signal interpretation and could only be performed for welds exceeding 6mm thickness. It could reveal surface and subsurface defects and it is suitable for automation. Liquid penetrant inspection [38] identify surface defects by enhancing the contrast between the flaws and the component background. The method requires careful attention to the general surface condition as the application over rough surfaces highlights irrelevant features.

Eddy current testing [38] measures the currents variations introduced into the workpiece by a coil carrying alternative current, placed in the proximity. The method is versatile, easy to automated, applicable on identifying not only the flaws but also the size and material variations.

All the testing methods outline until now are performed after the component reached the final or close to the final stage in the manufacturing process. The oldest of all methods, visual testing [38], is applicable at any stage during the process. Consequently, it has its limitations, identifying only surface flaws and providing approximate quality feedback.

2.3.2 Classical vision approach

Vision techniques for NDT, typically refer to the acquisition and processing of electromagnetic waves stretching from ultraviolet (UV) light's lower bound 10^{-7}m (100nm) to the infrared's (IR) upper bound of 10^{-3}m (1mm). Within this range, the visible spectrum covers the region between 400nm and 780nm [13].

Several authors analysed the plasma arc spectra [39–41]. The lines of the spectrum emitted by the high-temperature plasma arc are recorded and used to determine the correlation between defects and arc stability.

Other research studies reported the use of the specular light for the detection of defects in welds. Specular emission is the reflection of the light from the welding pool. In the literature, approaches branch into the use of the raw specular light emitted from the weld [42] and the reflections of the structured light projected onto the welding weld pool from an external source [43]. In a suite of publications, Liu and Zhang [44–46] combined the specular weld pool surface representation of the TIG welding process with the adaptive neuro-fuzzy inference system (ANFIS) for establishing the correlation between the welding current and speed and the back-side bead width. In a subsequent study [47] the same authors attempted to fuse the human welders' robustness to weld pool variations and the robot's quick response capability for offsetting varying welding currents and input disturbances.

Several other studies record the welding process [48–50]. One study approaches the combination of the laser illumination and video recording for offsetting the powerful arc light [51, 52]. The laser would shine light over the weld pool to bridge the luminosity gap between the very bright arc and very dark surroundings. The alternative to laser illumination is filtering. The additional filter blocks the UV and most of the visible light, the emissions range of the powerful arc light. It allows near-infrared (NIR) light to pass, while the camera is sensitive within NIR bands.

Some other applications involved assessing parameters that are hard to measure in real time. Luo et al. [5] measured the weld quality based on the laser keyhole diameter, penetration depth and inclination angle. When real data are limited, and the simulated data generalise well, manufactured input, i.e. parameters generated by a computer model, represent an alternative [53].

The lines of the spectrum approach involve analysing the optical emission of the thermal plasma. Mirapeix et al. [39] employed the approach extensively by utilising the techniques for electronic temperature estimation and captured spectra correlation with the occurrence of defects. The authors' target was real-time implementation during the manufacturing of a large steam generator for nuclear power plants. The advantages of this approach are:

- the optical spectrum includes emission lines of atoms present in plasma
- the sensor does not interfere with the process
- the CCD-based spectrometer are inexpensive
- the optical fibre is immune to the electromagnetic interference

An important characteristic deduced from the optical spectrum is the electronic temperature, T_e.

Three approaches for the data analysis are considered:

- the spectroscopic analysis based on the background radiation for calculating T_e using the relation between single emission line intensity and the intensity of the adjacent background radiation. Subsequently, the Sequential Floating Forward Selection (SFFS) was applied to reduce the dimensionality (i.e. the number of wavelengths). The choice of optimal selection lines is $a\ priori$, and the monitoring involves measuring emitted light at a particular wavelength relative to surrounding background level.

  The background radiation has a wavelength of maximum emission, and it is stable for a defect-free and constant parameters welding process.
- the second approach involves defect detection through the RMS spectroscopic signal. The method avoids the plasma emission lines by windowing the spectra and considering the plasma root mean square (RMS) spectral intensity within the selected window.
- the third technique is direct defect detection using artificial neural networks, a multilayer feed-forward network with a back-propagation learning algorithm. Principal Component Analysis (PCA) and Sequential Floating Forward Selection (SFFS) achieves the reduction of the amount of spectral information.

The methods validation took place through the welding of a tube-to-tube sheet for assessing the detection and discrimination of defects.

Song et al. [43] used specular light for calculating the three-dimensional geometry of the weld surface. The system projected structured light onto the weld pool and captured the reflected light. The study system viability tests two algorithms: edge-point algorithm (EPA) and one point algorithm (OPA), both assuming the convex shape of the weld pool. In EPA, the edge points are assumed to be at the zero level in z-axis, and EPA-R (row) and EPA-C (column) are used to calculate the slope. In OPA the reference zero level is a single point. The system divides the weld surface into five regions due to the geometry not resembling a circle. Arcs of different radius, deduced empirically, connect the five regions division points. Overall, the OPA performs better due to a smaller error and smoother weld pool surface.

Fujita et al. [3] describe a system for observing a welding process of a thick-walled pipe from different perspectives. The authors identify welding conditions considering the welding current, welding voltage, welding speed, and wire feeding speed. The software assesses the deviation from the standard, preset values, and notifies the welder for exceeding the boundaries. The system monitors the weld pool visually using an Infrared Charged-Coupled Device (IR-CCD) camera, coupled with a neutral density (ND) filter and 1064 nm (infrared) narrow bandpass filter. The software recognises weld pool shape, electrode, wire, and edge shape indicating the correct position and notifies the welder for the detection of an abnormal value. The second CCD cameras, placed behind welding zone, takes a three-dimensional view of the resulting bead surfaces, assessing the level of porosity, undercut and overlap while analysing the colour and texture appearance. The subsystem provides feedback regarding abnormal changes in bead surface. An ultrasonic testing system completes the overall system. A neodymium-doped yttrium aluminium garnet (Nd:YAG) laser (1064nm) is projected onto the material surface, transforming few atomic layers into plasma which in turn produce a volume wave. The wave propagates through the welded region becomes scattered and reflected, revealing defects bigger than 1.6mm in diameter and at a depth of maximum 10mm.

Carrasco et al. [54] explore a method for the detection of flaws using an ensemble of multiple views for discerning flaws from false alarms. The advantage of correlating redundant views as opposed to single view processing lays in the power to filter out the false alarms, generated due to the noise, from real flaws. The whole process is uncalibrated, meaning that it has no prior knowledge of the object structure.

The technique involves two stages: identification and tracking. The proposed method is composed of four steps:

- flaw identification - uses feature extraction for identifying the regions with flaws
- control points extraction - for tracking. The structure's information resulted from filters application (e.g. segmentation, edges extraction, normalisation, smoothing) helps to extract the reference points. The step has two stages:
  - the matching of regions:
    - firstly, the application of Otsu's method
    - secondly, extraction of Flusser-and-Suk moments from each region
    - thirdly, the determination of corresponding regions
  - the matching of control points represents the establishment of pair-points corre-

spondence on the border of a region

- the tracking filters the flaws from false alarms. While a flaw exhibits a spatio-temporal relation in a different view, a false alarm is similar to a random event. The author studied tracking using two and three views.
- the intermediate classifier block is the decision making step. Based on the information collected during the previous steps, the flaws are categories as flaws (F), false alarms (FA) and potential flaws (PF).

Results for two and three-views tracking:

Step	Flaws in sequence	False alarms in sequence	Rate of real flaws (%)	Rate of false alarms (%)
2-Views Track	190	198	100	51
3-Views Track	137	45	100	11.6

Table 2.3: Performance for the uncalibrated tracking [54]

Liao et al. [55] propose a method for defect detection by employing a background subtraction technique. The authors take advantage of the high-frequency characteristic of the pixels that form the background, initially proposed by Kornprobst et al. [56], then it combines the Friedman et al. [57] method of Expectation Maximization (EM), which states that the distribution of values for each pixel are Gaussians mixture model. The algorithm consists of three steps:

(a) Initialise mixture model for each pixel:

- calculate threshold T using Otsu's method, by dividing pixels in two regions G_1 and G_2
- compute the average intensity values u_1 and u_2 for pixels in G_1 and G_2, respectively
- compute the average variance values v_1 and v_2 for pixels in G_1 and G_2, respectively
- define Gaussian models as $G_1(u_1, v_1, k_1)$ for the weld region and $G_2(u_2, v_2, k_2)$ for the background. Parameters u_1 and u_2 are the clusters centres, v_1 and v_2 are the distances and k_1 and k_2 are the frequencies

(b) Classify each pixel based on the mixture model and update the model

(c) Pick cluster centre of the Gaussian model of which frequency is higher

In conclusion, the average methods time for processing one single frame is 25ms, which makes it suitable for online usage.

Another application, discussed by Schwab et al. [58] describes a new method for measuring and classifying spatter using a high-frame-rate camera by applying image processing techniques and tracking the spatter. The camera operates at a sampling rate of 400 frames per second and outputs a 1280×1024 pixels frame of 8-bit depth. A morphological operation applies to the image with a circular 3-pixels filter. Therefore, objects smaller than 3 pixels in radius could not be detected. The author tracks the spatter over several frames and tries to trace the objects trajectories and calculates the $a\ posteriori$ probability for tracks and false detections. The experiment analyses 1200 images for the number of spatter events, average velocity, maximum velocity and size in pixels mapping the pixels dimension to a real-world object.

Liu et al. [59] designed a system, hardware and software, for sensing the top side of a keyhole plasma arc welding. The novel system consists of a CCD camera and a filter split in two part. The different part of the filter attenuates the strong arc light and obtain a more balanced image. The attenuation rates are 6% and 0%. The camera position is perpendicular to the welding direction, on the side, observing the weld pool front, middle and rear sides. The team extracted the weld pool boundary by applying Canny filtering followed by an algorithm for identifying an initial point on the boundary. The authors examined the weld pool width and length variations as permutations of current, voltage and welding speed were applied.

Jiang et al. [60] proposes a sequence of steps for processing the images from a CMOS. The camera delivers images of resolution 1280x1024 pixels, at 40 frames per second, and it embeds a UV chip and narrowband filter of bandwidth 15nm and central wavelength 635nm. The sequence includes target area interception, image inversion, intra-group variance threshold determination, high-level morphological processing, FFT low-pass truncate filter processing and Canny edge detector to extract the edge of the weld pool. The end goal of the study is to extract the weld pool width, by identifying the maximal distance between two melting point boundary perpendicular to the welding direction.

In another study, Zou et al. [28] proposed a method for defect detection in the spiral pipes using radiographic NDT images. The first step consists of filtering and thresholding the image followed by the application of the Prewitt filter for weld edges extraction. A custom designed operator segments the defect by emphasising and smoothing the potential defect region. The employment of Kalman filtering for tracking defect trajectory results in the classification of defect vs no defect. The authors report that the method achieved maximum accuracy if 100%. The authors also compared the Kalman filter approach against another method by Shao et al. [61] based on Hough transforms. The proposed approach provides further resilience against velocity variations with a maximum detection rate of 91.1% as opposed to 66.7% for Hough transform approach.

An alternative approach investigated by Fidali et al. [62] focussed on fusing the visual spectrum and the infrared spectrum into one single image, suitable for downstream assessment. Two cameras provide images, uncooled infrared camera Infratec VarioCam Head and visible light CCD camera ImagingSource DMK21AF04, both delivering 640x480 pixels images. The authors started by studying several techniques for the image registration and the image aggregation, eventually settling for edge orientation maps algorithm and shift-invariant wavelet transform (SIH), respectively. The identification of the suitable conditions for MAG welding involved expanding the study to statistical evaluation of the predefined area of the arc region. The features used for the evaluation include: mean, RMS, variance, standard deviation, contrast and entropy. The topological assessment of welding area used the features: major diagonal, minor diagonal, perimeter and area. The analysis of intensity profiles along lines predefined in the arc, welding area and heat-affected zone used the features: total width, left width and right width. A k-nearest neighbour classifier discriminates between different welding conditions using the features enumerated above, reporting a maximum accuracy of 64%.

The current study analyses the signal received from an imaging sensor. Taking into account that images are grayscale, an image is a signal in two dimensions. Although the signal amplitude (pixels values) is important, the image pixels organisation give rise to features that are as important as the values themselves. The current research refrains from applying filters with the effect of dimensionality reduction over the image (e.g. Canny filter, Sobel filter, Harris operator, high boost filter, Scale-invariant feature transform (SIFT)) allowing the neural network architecture to arrive at the optimal kernels. Although, to the human eye, the image might become clearer after the application of a filter the result is the information removal or alteration of a process-specific, authentic signal. One potential benefit of pre-processing filter application could be the reduction in the processing resource required for each image. Since the aspect of transfer to the production environment is not of primary concern in the current study the pre-processing filtering is not approached. On the premise the input signal is in a grid format, the state of the art research in the field of vision for recognition is machine learning.

2.4 Artificial Intelligence (AI)

The welding process control system using some form of artificial intelligence (AI) started with the introduction of neurofuzzy architecture by Zhang and Kovacevic [63, 64]. The neurofuzzy algorithm was capable of adapting and controlling the non-linear nature of TIG welding by establishing a correlation between the weld pool boundary and the top-side and back-side bead widths.

The field of artificial intelligence experienced unprecedented growth in popularity in recent years due to unparalleled power for adaptation with impressive results in various tasks ranging from image classification as in [65], [66], [67], [68], natural language processing (NLP) as in [69], [70], [71], language modelling through the work from [72], [73], to handwriting recognition in [74] and prediction of chaotic systems [75]. Although the relevant results emerged relatively recently due, primarily, to increase in computational capabilities, the concept of neural networks is not new. The paradigm appeared initially in the 1940s under the name ”cybernetics” in work of McCulloch and Pitts [76] and developed subsequently by Rosenblatt [77] and Widrow et al. [78] in 1950s and 1960s, respectively. Another breakthrough came in the late 1980s from Rumelhart, Hinton and Williams [79] with the invention of backpropagation (optimisation technique) for training neural networks, used to this day almost exclusively. The best known and one of the first application of neural networks came in 1989 from LeCun et al. [80] for automatic digit recognition by connecting the lattice structure locally, similar to Neocognitron by Fukushima [81], nowadays known as a convolutional neural network (CNN). Empirically, it was observed that neural networks trained on one dataset of images could perform synthesis (feature extraction) of any other dataset never used before and furthermore, by training on the new dataset (fine-tuning) the network uses the previously determined internal parameters as the starting point to further enhance the performance.

AI is the overarching umbrella encompassing, depending on the definition, constraints and control theory, computer vision, search and optimisation, reasoning, NLP, machine learning, etc., generally any system that mimics the human intelligence in some form, but with significant limitations, at least for the moment. Machine learning (ML) is only a subset of AI concerned with techniques allowing computers to improve at performing a task with experience. Neural networks are a subset of machine learning, as shown in Figure 2.4, which uses a cascaded lattice of nodes for processing the input. Deep learning/deep neural networks (DNN), the architecture most used, are subsequently a subset of neural networks, being endowed with several hidden layers. Computer vision, on the other hand, is an interdisciplinary field that intersects machine learning due to the application of neural networks on images. In this work, the emphasis falls on machine learning using neural networks as the underlying architectures for classification.

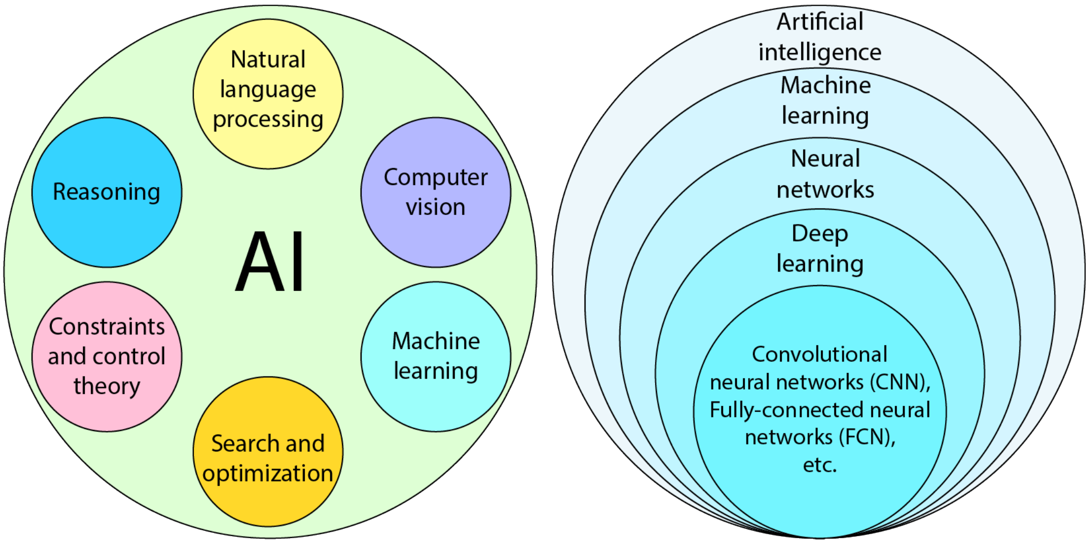

Natural language processing
Reasoning
Computer vision
AI
Constraints and control theory
Machine learning
Search and optimization
Artificial intelligence
Machine learning
Neural networks
Deep learning
Convolutional neural networks (CNN), Fully-connected neural networks (FCN), etc.

Figure 2.4: AI field and denominations

2.4.1 Machine learning

Machine learning paradigm looks to identify a model f from a set of models $\mathcal{F}$ that solve some particular problem. The mapping from input to output could be achieved non-parametrically, for example, the Nearest Neighbour [82] is approximating the label as being the same as the nearest example, or parametrically as in the case of neural networks.

Models in machine learning come in sets (or families) of functions $\mathcal{F} = \{f_\theta | \theta \in \Theta\}$, with $f_\theta(x) = f_{\mathcal{F}}(\theta, x)$ for function $f_{\mathcal{F}} : \Theta \times \mathbb{D} \to \mathbb{T}$ where parameters $\theta \in \Theta$ and input $x \in \mathbb{D}$ maps on a task $\mathbb{T}$. The parameters θ are learnable by following the prior knowledge and structure of $f_{\mathcal{F}}$. Therefore, the family $\mathcal{F}$, from which $f_{\mathcal{F}}$ is part of, directs and limits the kind of solution the model can learn.

Learning is a process of finding f^* from a family of models $\mathcal{F}$ for performing task $\mathbb{T}$, which is identical to finding the optimal parameter value $\theta^* \in \Theta$. The metric for convergence towards the optimal solution is measured by defining some error (or loss) measurement between the models output and the ground truth output. Assuming π being a distribution over $\mathbb{D}$ and $\mathcal{L}$ some define loss, the minimisation is:

$$f^* \leftarrow \arg\min_{f \in \mathcal{F}} \mathbb{E}_{x \sim \pi}[\mathcal{L}(x, f)]$$

where $\mathbb{E}_{x \sim \pi}[\mathcal{L}(x, f)]$ is the expected loss or generalisation error, while $\mathbb{E}_{x \sim \pi}$ is the expectation over x sampled from distribution π. In reality the distribution π is unknown, but the dataset $\mathcal{D} = \{x^{(i)} \sim \pi | 0 < i \leq N\}$ is from π, called the training dataset. In practice, an approach called *Empirical Risk Minimisation* is employed to approximate the function defined as follows:

$$f^*_{ERM} \leftarrow \arg\min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathcal{L}(x, f) \overset{\text{def}}{=} \arg\min_{f \in \mathcal{F}} \mathcal{R}_{EMP}(\mathcal{D}, f)$$

where $\mathcal{R}_{EMP}$ is empirical risk, same as the error on the dataset $\mathcal{D}$ and is similar to maximum likelihood learning in probabilistic models.

2.4.2 Regularization

Given a sufficiently flexible family of models $\mathcal{F}$, one can theoretically choose a function that approximates to perfection the dataset $\mathcal{D}$, totally avoiding generating a mapping between input and output that reflects the distribution π.

To understand and measure the power of the model to represent the distribution (or generalisation error), one has to segregate a part of the distribution π, denoted $\mathcal{D}_{test}$ and called test dataset so that there is no overlap between the two partitions. Empirical risk in this case is defined as $\mathcal{R}_{EMP}(\mathcal{D}_{test}, f)$ and by comparing to the training dataset empirical risk ($\mathcal{R}_{EMP}(\mathcal{D}_{train}, f)$, $\mathcal{D}_{train} \cup \mathcal{D}_{test} = \mathcal{D}$), the model is said to be *over-fitting* or *under-fitting*. *Over-fitting* is the scenario when training error is small while testing error is very large. To overcome this scenario, one has to reduce the search space, $\mathcal{F}$, manually by eliminating the functions that memorise $\mathcal{D}$ without a parametric representation of π or to introduce a regularisation term (or a penalty), restricting the search space for f. The adjusted equation for regularised parameters search is:

$$f^*_{ERM} \leftarrow \arg\min_{f \in \mathcal{F}} \mathcal{R}_{EMP}(\mathcal{D}, f_\theta) + \lambda\Omega(\theta)$$

with $\lambda > 0$ and $\forall \theta \in \Theta : \Omega(\theta) > 0$. $\Omega(\theta)$ is called regularisation term, must be differentiable and its selection is very task specific. λ is the weight of regularisation and drive the minimisation of Ω versus the minimisation or $\mathcal{R}_{EMP}$. At the same time λ is not similar to any other parameter θ which means it cannot be learned, it must be set at the beginning of training and it is put in the category of *hyper-parameters* together with some other parameters discussed later. Due to the fact that *hyper-parameters* are a different set of parameters ($\lambda \notin \Theta$) the *model selection* that incorporate the $tuned$ values is assessed on yet another dataset, independent of $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$, which is called validation dataset $\mathcal{D}_{valid}$ drawn from the same distribution π. Minimising the empirical risk on the dataset $\mathcal{D}_{valid}$ for some parameters θ found on dataset $\mathcal{D}_{train}$ for a fixed choice of hyper-parameters renders the optimal solution for the task. Mathematically, the minimisation is defined as:

$$\lambda^* = \arg\min_{\lambda \in \mathbb{R}} \mathcal{R}_{EMP}\Big(\mathcal{D}_{valid}, \arg\min_{f \in \mathcal{F}} \mathcal{R}_{EMP}(\mathcal{D}_{train}, f_\theta) + \lambda\Omega(\theta)\Big)$$

The search for optimal hyper-parameters (in this example only λ, but in reality there are several) take different routes from manual tuning by intuition of experienced researcher [83] to more systemic approaches involving grid-search, random-search [84], reinforcement learning [85, 86], evolution [87], sequential model-based optimization (SMBO) [88], or gradient-based search [89].

2.4.3 Supervised learning

Depending on the nature of the task one common division for the machine learning is *Supervised learning* and *Unsupervised learning*. Supervised learning rely on the assumption that acquiring image and label pairs $(x, y) \in X \times Y$ is a tractable task. Given the setting, the aim is to identify the true conditional distribution $\pi(y|x)$ from the dataset $\mathcal{D} = \left\{(x^{(i)}, y^{(i)}) \sim \pi(x, y)|0 < i \leq N\right\}$ of identically independently distributed (i.i.d) examples. Allowing a model f_θ to define a parametrised conditional probability density function $p_\theta y|x$ opens the door for leveraging the knowledge from statistics and probability for the identification of optimal set of parameters θ.

Binary classification

Binary classification denotes that cardinality of Y is 2. The model is regarded as Bernoulli distribution mean:

$$p_\theta(y|x) = \begin{cases} 1 - f_\theta(x) & \text{for } y = 0 \\ f_\theta(x) & \text{for } y = 1 \end{cases}$$

The similarity between the conditional Bernoulli distribution and the true $\pi(y|x)$ distribution is the error and it is measured using the *cross-entropy*:

$$\begin{aligned} \mathcal{L}_{CE}((y, x), f_\theta) &= -y \log p_\theta(y|x) - (1 - y) \log p_\theta(1 - y|x) \\ &= -y \log f_\theta(x) - (1 - y) \log(1 - f_\theta(x)) \end{aligned}$$

Multi classification

Multinomial classification refers to the case when there are more than two classes, although mutually exclusive. The approach uses a vector with n entries, one for each class, populated with 0s everywhere except the position, k corresponding to the class, where the vector holds the value 1 (e.g. $[1, 0, 0, \ldots, 0]$ for belonging to class 1, $[0, 1, 0, \ldots, 0]$ for belonging to class 2, etc.). In this case, the formula for approximating the similarity between the distributions is *negative log likelihood*, formulated as:

$$\mathcal{L}_{NLL}((y,x),f_\theta) = \sum_k -y_k \log p_\theta(y_k|x) = \sum_k -y_k \log f_\theta(x)_k$$

where $p_\theta(y_j|x)$ is the probability for sample x to belong to class k and $f_\theta(x)_k$ is the k-th position from the vector $f_\theta(x)$.

2.4.4 Neural Networks

Neural networks is a computing paradigm, a specific set of models, initially introduce in 1958 by Rosenblatt [77] and composed of three types of layers: $input\, layer$, $hidden\, layer$ and $output\, layer$, as in Figure 2.5. The *input layer* denotes $x \in X$, and in the case of this study, it is the images. It has a fixed number of nodes, and each node is connected to every node in the subsequent layer. The *hidden layers* stage is composed of several layers representing latent computations. The last layer is called *output layer* and represents $y \in Y$. Each node is represented by a circle in Figure 2.5 with with detailed computation described in Figure 2.6.

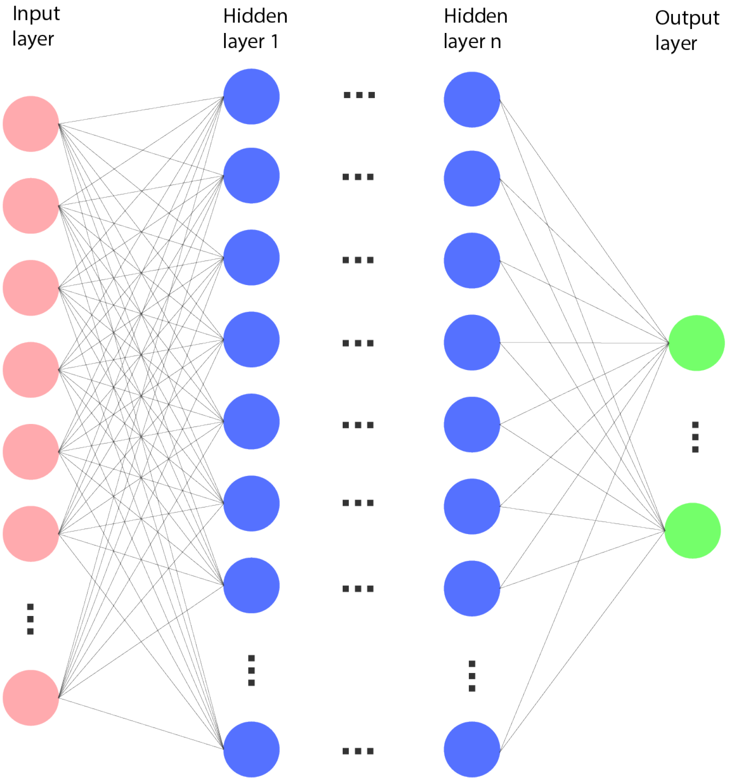


Figure 2.5: Fully Connected Neural Network [90]

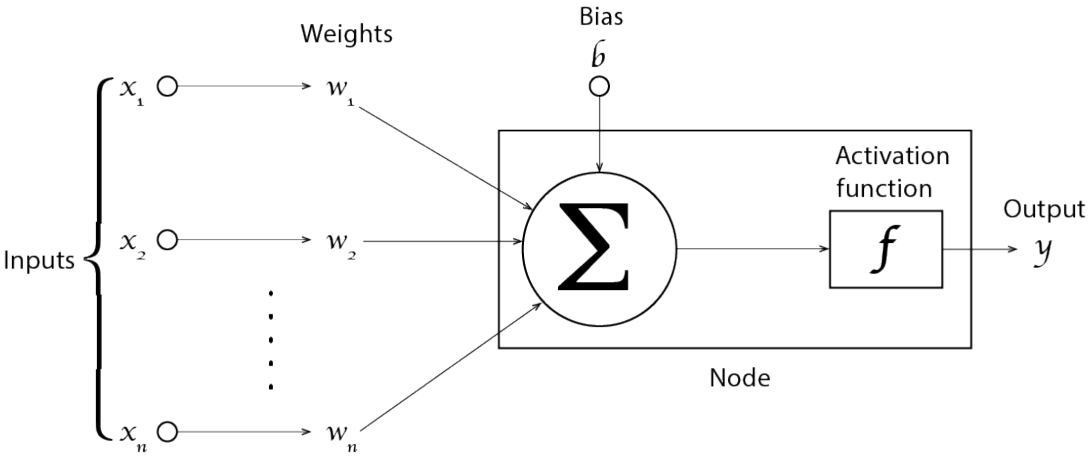


Figure 2.6: Neural network node internal operations [91]

Input layer receives the image (e.g. one node for every pixel) then the values are multiplied by $w^1_{11}, w^1_{12}, \ldots, w^1_{1n}, w^1_{21}, w^1_{22}, \ldots, w^1_{2n}$, all collated into a matrix called $W^{(1)}$ after which the biases b is added to the sum of weighted input as follows:

$$y_k^{(j)} = \sigma\left(b_k^{(j)} + \sum_{i=1}^{n} x_i^{(j-1)} w_i^{(j)}\right)$$

where $y_k^{(j)}$ is the k-th node in the hidden layer j with bias $b_k^{(j)}$, and σ is the a non-linear function (activation function).

The network aims to output y^* that matches the mapping (x, y) defined earlier and the process of minimising the difference between y^* and y is called learning. The network minimises the difference by changing the values of the weights and biases (W, b). (W, b) represents the parameters of the model ($(W, b) \in \theta$). The entire structure of the network is a computational graph with the number of hidden layer and the number of nodes in each hidden layer being task specific. A network with one hidden layer is shallow, and one with more than one hidden layer is deep (hence the name deep learning, or deep neural networks).

The intuition for the neural network is that of a functions composer. The successive layers add their weights and biases constructing a complex function that maps the input to the output. The non-linearity to be composed for every node is given by the activation function σ. Some examples of activation functions are sigmoid, tanh and rectifier function.

$$
\begin{aligned}
sigmoid(a) &= \frac{1}{1 - e^{-a}} \\
\tanh(a) &= \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}} \\
rect(a) &= \begin{cases} a, & a > 0 \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
$$

The output layer in case of multi-class classification uses $softmax$ for transforming the output into a probability distribution.

Convolutional Neural Networks

This study applies the fully connected neural network architecture (FCN) [92–95] and convolutional neural network (CNN) [96] for processing the weld images. FCN and CNN are subsets of the larger ANN processing paradigm [91]. The concept of convolutional neural networks (CNN) came from the work of Hubel and Wiesel [97]. Both architectures have an input layer for receiving the image and an output layer for the classification label. In FCN there is a connection between each input pixel and every node from the subsequent layer. In case of CNN, the region for kernel application is limited to the kernel dimension. One of the first application of CNN [98] was character identification [99].

A representation of convolution dynamics is Figure 2.7, where the input image (green) convolves with the kernel (red) outputting one single value (purple). Each layer has several filters (3×3, 5×5, or generally $n \times n$), also called kernels, convolving with the input image and outputting another set of images called feature maps. To reduce the feature maps dimensionality a step of pooling (sub-sampling) applies to each feature map. The pooling applies on local patches from the feature map summarising the patch. The summarization can take the form of maximal response (selecting the larges value from the patch), also known as max-pooling or averaging the patch, also known as mean-pooling. CNN general layout is shown in Figure 2.8.

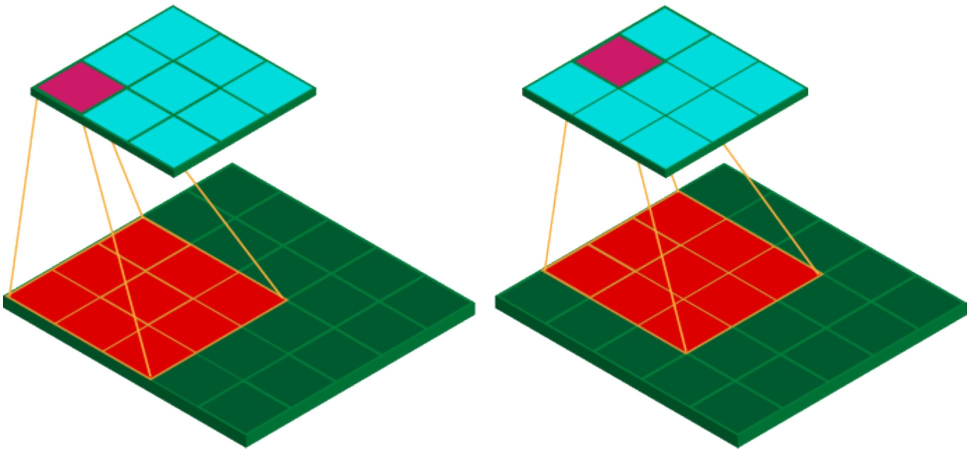


Figure 2.7: Convolution between a feature map of dimension 5×5 (green) and a kernel of dimension 3×3 (red), stride 1 and image padding 0. The result is another feature map of dimension 3×3 (turquoise).

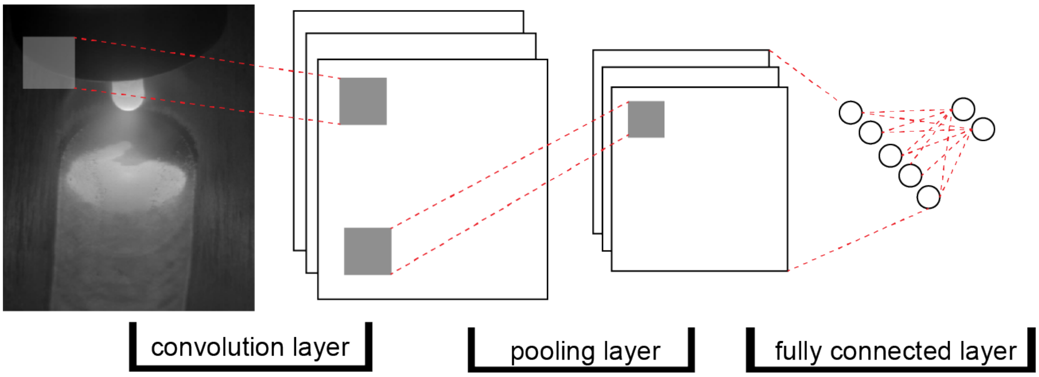


Figure 2.8: Succession of convolutional layers in a CNN

By traversing the input image one pixel at a time (stride 1) horizontally or vertically, another image (feature map) is created (turquoise). The reduction in the number of parameters required to produce the same output is significant compared to FCN. As an example in the current displayed setting of Figure 2.7, there are 3 × 3 = 9 weights producing 9 outputs, while in the case of fully-connected networks (FCN), producing the same output would require 5 × 5 × 9 = 225 weights because the number of parameters grows quadratically with the input image size.

Optimization

The optimisation in the context of the neural network implies finding the optimum set of parameters θ such that the expected cost on a dataset is minimised:

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta)$$

where θ is the set of parameters, and $\mathcal{L}$ is the average of expected loss over the entire dataset and a regularisation penalty. The cardinality of θ in case of a neural network varies widely and could easily be in the order of millions. As such, the approach of guessing the parameters is out of the question. The problem further complicates by the non-convex nature of the multidimensional space described by the millions of parameters.

Gradient Descent

The problem tractability could be improved by imposing constraints on $\mathcal{L}$, such as requiring $\mathcal{L}$ to be differentiable with respect to the parameteres θ. The partial derivates of $\mathcal{L}$ build a map of the parameters' landscape for a specific set θ and taking a small step in the direction opposite to the gradient results in a reliable minimisation of the function $\mathcal{L}$, therefore the expected loss. The method is the Taylor's first order expansion of $\mathcal{L}$ and is given by the Jacobian $\mathbb{J} = \left[\frac{\partial\mathcal{L}}{\partial\theta_1}, \frac{\partial\mathcal{L}}{\partial\theta_2}, \ldots, \frac{\partial\mathcal{L}}{\partial\theta_m}\right]$.

The algorithm name, *gradient descent* (GD), highlights the move down the slope described by the gradients. Another hyper-parameter $\epsilon \in \mathbb{R}_+$ controls the step size. Too big ϵ and the algorithm will diverge, too small ϵ and the algorithm will converge very slowly or not converge.

The increase in the dataset has a direct impact on the convergence speed of GD. A better

Algorithm 1 Gradient Descent Algorithm

```
Initialise the model randomly by $\theta_0$
while not close enough to solution do
    $\Delta\theta \leftarrow 0$
    for all $x \in \mathcal{D}_{train}$ do
        $\Delta\theta \leftarrow \Delta\theta - \left(\nabla\mathcal{L}\left(x, f_{\theta_{[t]}}\right)\right)^T$
    end for
    $\theta_{[t+1]} \leftarrow \theta_{[t]} + \epsilon\Delta\theta$
end while
```

approach is to update the parameters after every example (Stochastic Gradient Descent (SGD)) or after several examples (Mini-batch Stochastic Gradient Descent (MSGD)). The *mini-batch* refers to a small number of samples from the training dataset after which an update of parameters is performed. Iterating over all examples once is denoted a *epoch*. When mini-batch is equal to 1, it is SGD, conversely, when the mini-batch is equal to the cardinality of $\mathcal{D}_{train}$, it is GD, while everything in between is MSGD. Stochastic methods, besides accelerating the learning, points roughly towards the local minima introducing noise, arguably helping to escape narrow minima. The algorithm could be accelerated further by endowing the update step with a **momentum**, moving in the direction that is the most consistent. Let $g = \nabla_\theta \mathcal{L}$. The update step is composed of $v \leftarrow \alpha v + g$ and the applied gradient is $\Delta\theta \leftarrow \epsilon v$. The additional term weighted by α is an exponentially-decaying sum of previous gradient directions, also called the first-moment gradient.

More advanced optimisers seek to change the learning rate ϵ dynamically for every parameter. **Adagrad** [100] is one example of algorithm by calculating initially the running sum of the squared gradients $v \leftarrow v + g \odot g$ and update the gradients as $\Delta\theta \leftarrow -\frac{\epsilon}{\delta+\sqrt{v}} \odot g$, where δ is a very small number (e.g. $1e^{-5}$) for avoiding division by zero. Adagrad accumulates in the denominator the square roots of the sum of square gradients (second-moment or uncentered variance), reducing the learning rate for fast changing parameters. It is also the technique's weak point, as gradients keep changing, the learning rate decreases, driving learning to a halt. Alleviating the effect are **RMSProp** [101] and **Adadelta** [102] algorithms calculating a running mean for the second moment, $v \leftarrow \delta v + (1-\delta) g \odot g$, before updating the gradients. Combining first and second running moments is *Adaptive Moment Estimation* (Adam) [103] algorithm with the advantage of damping large gradients and boosting low gradients.

2.4.5 AI and Welding

AI and welding found intersecting paths involving the welding output signal processing. One approach involves spot welds assessment with the help of computer vision. The study [104] reported a method of image registration based on geometric pattern matching, image segmentation, feature extraction and defect classification using an Artificial Neural Network.

The localisation of the weld spot requires two steps. The first step is edge detection utilising the Canny filtering. Disjoint edges connect into chains then a geometric de-noising method is applied.

The second step is course registration, measuring the similarity between images by calculating the gradient direction of edge points. Four parameters describe the course registration result, i.e. assessment of similarity between images: scale parameter s, i.e. one image can be a zoomed version of another, rotation α, i.e. one image can be a rotated replica of another, translation t_x and t_y, i.e. one image can contain the same object but not at the same place within the image relative to image frame.

The next stage, having localised the weld spot, is the electrode imprint segmentation. Segmentation is a technique for grouping together feature sharing similar characteristics based on a threshold, edge detection and region detection. In this paper the author used the compactness as a feature, with the definition:

$$Compactness = 4\pi\frac{area}{perimeter^2}$$

The next stage, after segmentation, is the feature extraction. The extracted features (see Table. 2.4) of all segmented regions are area, major axis, minor axis, solidity, perimeter, convex area, eccentricity, orientation and compactness.

1	Area	$A = \sum_{i=0}^{n}\sum_{j=0}^{n} a_{i,j}$
2	Major axis	$L = \sum_{i=0}^{n}\sum_{j=0}^{n} p \cdot (a_{i,j} - u)$
3	Minor axis	$e = \sum_{i=0}^{n}\sum_{j=0}^{n} q \cdot (a_{i,j} - u)$
4	Solidity	$S = \sum_{i=0}^{n}\sum_{j=0}^{n} a_{i,j}/A_c$
5	Perimeter	$P = number\ of\ pixels\ forming\ boundary\ of\ object$
6	Convex area	$A_c = \sum_{i=0}^{n}\sum_{j=0}^{n} a_{i,j}$
7	Eccentricity	$E = \sqrt{1 - b^2/a^2}$
8	Orientation	$O = angle\ of\ L$
9	Compactness	$C = 4\pi\frac{area}{perimeter^2}$

Table 2.4: Features formulas [104]

The subsequent stage, classification, uses an artificial neural network (ANN) trained to distinguish between defects. The author reports the classification of six defects without, however, mentioning what type of defects. The reported accuracy of this approach is 98% in classifying the defect correctly and 99% in detecting a defect.

Hou et al. [105] investigated a system for identifying weld defects in x-ray images using a Deep Neural Network architecture. More precisely, the author used three stacked sparse autoencoders (SSAE) for extracting relevant features from images. Subsequently, the classification stage divides the images into two categories, no-defect vs defect. The system processed images by dividing every image into 32x32 blocks, with a stride of 2 pixels between each block, and applying the classification on each block accumulating a score for each pixel within the image. The authors trial a different number of layers, nodes within each layer, and training parameters combinations for finding the best performing configuration achieving, in the end, a 91.84% accuracy.

2.4.6 Neural Architecture Search

The empirical evidence of bigger, deeper and fine-tuned networks exhibiting higher accuracy [106], [107] drives the shift from finding model weights to finding the combination of weights and inter-weight linking topology. The architecture search could be performed manually, requiring human input in defining every single internal aspect of a model: number of layers, layers' type, layers linking, node activation functions, loss metric and optimisation parameters.

The next logical step in improving the current models is to automate, at least partially, the search for architectures. It is finding the optimal number of layers of optimal type, with the right interlinking topology, bearing in mind the number of layers could be, virtually, infinite, with infinite ways of processing the input and an even larger number of inter-node combinations, transforming the problem into an intractable task even for the most powerful supercomputers. It assumes unconstrained search for, ideally, the best model for identifying the difference between defective welds and good welds. By stepping back from the ideal model, the architecture search could break into the definition of the searching space, a procedure to explore the space and accuracy forecasting, described in Figure 2.9.

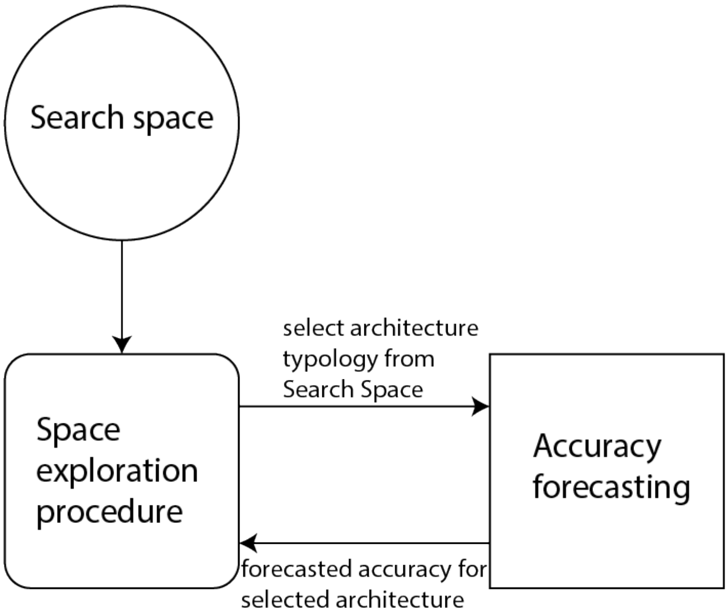

Search space
select architecture typology from Search Space
Space exploration procedure
Accuracy forecasting
forecasted accuracy for selected architecture

Figure 2.9: Neural Architecture Search structure. The forecasting is applied in architecture instances sampled according to the exploration procedure from search space.

The description for each of the steps for the Neural Architecture Search (NAS) is as follows:

- **Search space definition** represents the typological boundaries of the model search space. The search space could incorporate prior knowledge of the task at hand, therefore, diminishing the search space. One example could be searching for all the models of two convolutional layers of at most ten kernels in each of the layers, with layers connected sequentially. Although nothing guarantees the best model satisfies these properties, the search space is tractable. The constraints act as a loose bias instilled by the model designer in much the same way as defining a single architecture rigorously, i.e. 3 layers, 5 kernels/layer, sequential layers interlinking. The looser the bias the larger the search space.

- **Space exploration procedure** defines a method for searching the space for finding the optimal architecture candidate. Ideally, the search would be quick and would avoid sub-optimal candidates at the same time, in a typical exploration-exploitation balance.

- **Accuracy forecasting** is already used in machine learning when searching for networks' hyper-parameters. A fixed architecture uses a training set to adjust parameters and test set to measure the accuracy on unseen data. In this sense, an architecture search used a fraction of training set to forecast the accuracy of training a particular architecture on the whole training set. The smaller the fraction or, the faster the forecasting processing, the faster the exploration becomes.

The typologies search space (or search space definition) covers the type of architectures the model designer allows the exploration. Figure 2.10 shows three types of architectures: chained-structured typology, typology with residual connections and branched architecture. The fundamental linking characteristic is graphical acyclic typology, meaning that there are not loops during processing cascade. Therefore the search space is all graphical acyclic networks of n layers.

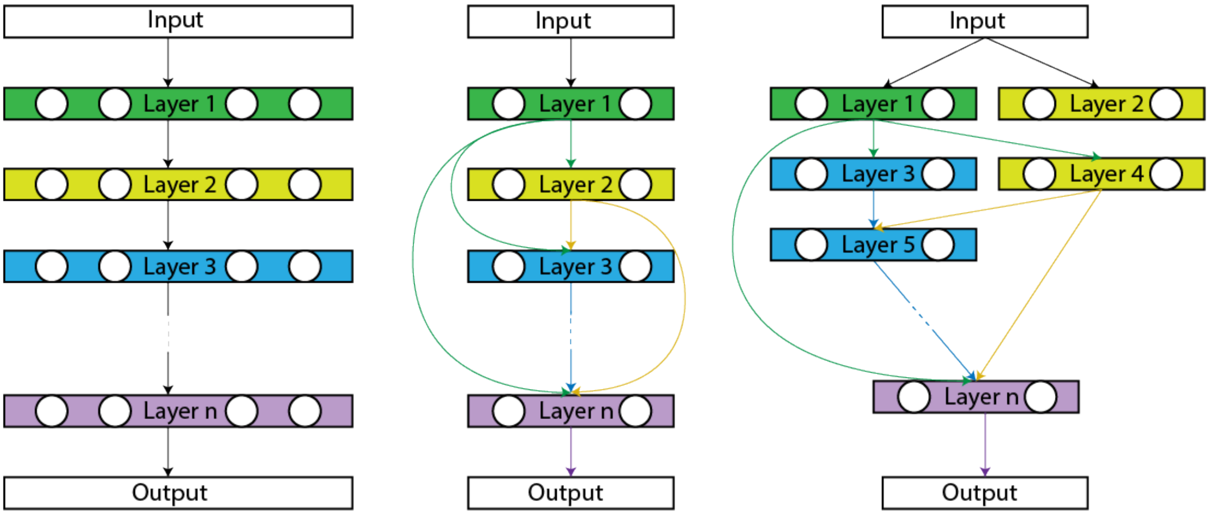


Figure 2.10: Networks typologies. Each rectangle is a layer (e.g. convolution, sampling) and each layer receives input from one or more previous layers forming an acyclic graph.

The typology parametrisation:

1. the number of layers, n, possibly very large [108],
2. the operations performed by each layer: convolution, sampling, etc [109],
3. the parameters defining the operations, as the number of kernels, the kernel size and the kernel stride.

Generally, the input for every layer is a function of the previous layers' output as $f_i(L_{i-1},\ldots,L_0)$. More precisely, for chained architecture the input for layer L_i is the immediately previous layer's output $f_i(L_{i-1},\ldots,L_0) = L_(i-1)$. DenseNets, as in case of Figure 2.10 (centre),

$f_i(L_{i-1}, \ldots, L_0) = concat(L_{i-1}, \ldots, L_0)$, the concatenation of every single output of previous layers.

The exploitation of the repeated patterns observed in many hand-designed architectures ([83], [110], [111]), led Zoph et al. [86] to the introduction of the *cell*, as the building block for architectures. Instead of searching for an architecture with alternating layers of either batch normalisation, convolution and sampling, the proposed idea creates a new entity, the *cell*, composed of such patterns that act as a unity while the whole architecture is a series of such cells. The exploration procedure, in this case, searches for the best cell that placed in the context of a network exhibits the best performance. The author proposed two variants of cells: the *normal cell* which preserves the input size and the *reduction cell* which shrinks the input size, reducing the input dimensions.

The approach [87], [89], [112], [113], [114], [115] severely reduces the search space as the cells have smaller dimension than a network and secondly, the cell typology found can be transferred to other tasks. An example is the cell typology exploration taking place in this study for aluminium alloy 5083 dataset with the found cell applied for training neural network destined to classify Stainless Steel 304 dataset.

It is easy to fall into the exploration of meta-architectures, namely how many cells and what interlinking should take place between cells if the cell typology exploration becomes over-simplistic. Typically, the cells typology should take into account the optimisation of meta-architecture during the process of exploration.

Liu et al. [112] defined stages of composition called hierarchical network composition. The first stage defines what operations a cell could perform (e.g. convolution, sampling, identity, no operation, etc.). The second stage is the linking operations inside a cell, defining which operations feed into which one. The third stage is linking the cells. Zoph et al. [86] chose a sequential model while in another study Cai et al. [115] used the linking designed by Huang et al. [83] in DenseNet.

The space exploration procedure takes several forms including random search, evolutionary techniques, Bayesian optimisation, reinforcement learning (RL) or gradient-based approach. NAS became a branch of machine learning relatively recently with the work of Zoph et al. [85]

on reinforcement learning. The only caveat in their approach is the vast amount of processing power required, approximatively 800 GPUs for three to four weeks. The evolutionary algorithms are older that Zoph et al. work for finding architecture and even for weights as in the work of Angeline et al. [116], Stanley et al. [117], Floreano et al. [118] and Stanley et al. [119]. The reinforcement learning approach for NAS precedes the emergence of the concept of the stacked cell for the neural networks. The work of Zoph et al. [85] used a recurrent neural network (RNN), called controller, to generate the typology (or model hyper-parameters) of another neural network, i.e. kernel height, kernel width, stride height, stride width. The generated neural network is trained and validated, and the loss would represent the reward for RNN. The RNN aims to generate the optimal architecture that would maximise the reward. The generator network is composed of Long-Short Term Memory (LSTM) units producing hyper-parameters for every layer of the generated network. Generating the entire network's hyper-parameters at once means that the search space is vast, every layer's typology being independent. A later work of the same author [86] introduced the concept of cells, with its two variants (normal cell and reduction cell) reducing the search space from generating an entire network to generating a cell and composing the network by stacking cells.

Instead of starting from a random state for the internal parameters, Cai et al. [107] devised a method of applying sequential mutations to a typology, called network morphism, and reused parameters trained during the previous training sessions. The approach could leverage human-designed typology already proven as performant and continues to apply function-preserving mutations to improve its accuracy.

The method of the evolutionary algorithm for NAS used principles imported from biology:

1. a set of trained networks and their performance is present
2. one or more instances from the set are sampled based on some criteria
3. the generation of mutated typologies based in the sampled instances (e.g. adding or removing one layer, altering hyper-parameters as kernel size, stride, etc.)
4. add the newly mutated, trained and tested instances to the set and start again from point 2.

The variations of neuro-evolutionary techniques differ on the method of instances sampling from the set, the generation of mutations and the longevity criteria for set members. The work of Real et al. [87], [120] focuses on the repeated competition between two members of the set, called *tournament selection* [121]. The members are trained and assessed on validation dataset, with the better performing chosen for the next mutation stage while the least performing removed from the set. The selection of mutations applied to the typology is random from a predefined set of mutations:

- alter learning rate
- no change, train for longer time
- reset weights
- insert convolution operation
- remove convolution operation
- alter stride
- alter the number of channels
- insert identity connection
- add skip connection
- remove skip connection

In the first study, Real et al. [120] elected to remove the worst performing from the set while in the second [87], the oldest from the set.

Elsken et al. [114] observed the shortcomings of having an optimisation method based solely on performance criteria, facilitating the model expansion, therefore, resource consumption. The author imposed multi-objective criteria allowing the approximation of Pareto-front. The criteria for optimisation is the predictive performance and the number of parameters. At the same time, to reduce the exploration cost, the author employed an approximate network morphism, similar to Lamarckian inheritance mechanism, where the parent weights are used for the next typologies generation, therefore reducing the training time.

The aspects where RL and evolutionary methods fail in handling NAS, *bayesian optimisation (BO)* provides some answers. Evolutionary approach formulates the strategy of sequential changes to the network elegantly to arrive at the optimal solution, but the networks still have to be trained and evaluated which is an expensive process. RL had relative success with optimisation, although at the cost of complexity, maintaining states and solve credit assignment, due to method's philosophy. RL is not an optimisation process, it is an exploration process. BO, on the other hand, is more expensive with defining the following typology but it pays off due to the emphasis on optimisation, essentially on reducing the number of inefficient typologies. Kandasamy et al. [122] tackles several challenges of BO by defining a distance metric for neural networks and devising an algorithm for traversing the space based on optimal transport. Bergstra et al. [123] used tree-based models to explore the space while Hunter et al. [124] used random forests.

The space explorations procedures described above classify as gradient-free optimisation since the network typology optimisation does not involve calculating any architecture derivative. At the same time, the typologies are discrete instances of the exploration space, trained, evaluated and morphed. Saxena et al. [125] imagined the exploration space as a multi-dimensional *fabric* with the optimal architectures representing a *thread* within the fabric, transforming the space into a continuum. The authors tried to solve the optimal network depth and layers hyper-parameters at the same. Subsequent approaches by Shin et al. [126] focused on finding only the optimal layer hyper-parameters while Ahmed et al. [127] tried to find the optimal connectivity pattern. The simultaneous search for the optimal combination of layer type, layer hyper-parameters, connectivity paths, and layer weight is attempted by Liu et al. [89]. Their approach is similar to Pham et al. [113] method in the sense that the network envisioned as a sub-graph of a larger super-graph. Figure 2.11 shows an example of a super-graphs. In Liu et al. work each super-graph connectivity path is governed by a weight founded on continuous nature of real numbers with only the strongest paths (largest weights) representing instances of trainable networks.

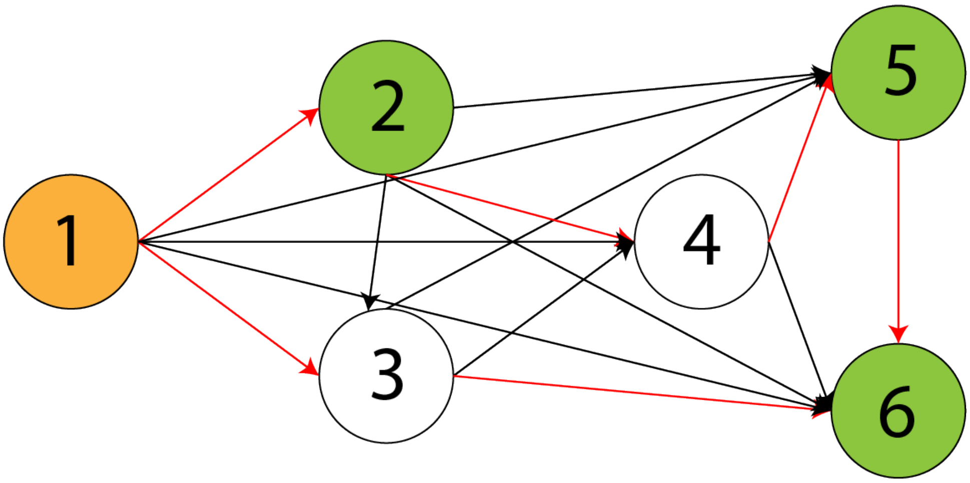


Figure 2.11: Acyclic connectivity paths for a cell with six nodes forming a super-graph. The red lines represent a single possible sub-graph, or a subset of all possible interconnectivity. The orange node is the input, while the green nodes are the output.

The idea is that the gradient-based approach applied to the connectivity weights eventually arrives at the optimal linking typology.

Although defining the search space and exploration procedure is not an easy task, the typologies generated requires training and validation for identifying their performance. The simplest approach for **forecasting the accuracy** of a network is training and validating the network. Several authors [86], [87], [88] and [120] did exactly that, stretching the computational demands to thousands of GPU days. The use of proxy metrics (or lower fidelity performance estimation) achieves the performance estimation with regards to the reduction of computational requirements. The lower fidelities include reducing training set size [128], downsample input data [129], reduce training time [130], or use fewer kernels for every layer as in [86] and [87]. The lower fidelities, although more efficient, tend to introduce bias and underestimate true performance, although as long as the ranking of different networks is maintained the approximation is still relevant. Different recent studies cast doubts on lower fidelities and show a considerable change in ranking when the difference between *fast* forecasting and *slow* forecasting is large [130], with other studies proposing a stepwise increase in fidelity as in the work of Li et al. [131] and Falkner et al. [132].

Approaching the problem of forecasting from another perspective, Domhan et al. [133] proposed learning curve extrapolation for the process acceleration while Swersky et al. [134] combined learning curve information and architecture hyper-parameters.

One-Shot Architecture Search, or the super-graph approach, treats all possible typologies as a subgraph of a larger graph by disabling certain paths as in Liu et al. [89], Pham et al. [113], Savena et al. [125], Brock et al. [135] and Bender et al. [136]. The super-graph is trained once, and all the sub-graphs share the architecture weights making the validation much cheaper. The difference in methods is in the selection of sub-graphs. Pham et al. [113] uses a Recursive Neural Network (RNN) to sample the super-graphs while Liu et al. [89] optimises two sets of weights alternatively, the set of weights related to linking paths and the layers set of weights (the weights that multiply or convolve with the input image). The one-shot NAS limitation is the super-graphs size and typology as it defines the maximum sub-graph size and sub-graph shape. The practicality of such an approach is further limited by the GPU memory, as the super-graph must fit into memory.

The current state of the art presents a gap in using vision sensors and raw image processing for weld defect identification. This study uses images in the visible spectrum and outputs the classification similar to human classification. At the same time, the image dataset proposed in this study is the most diverse and complete for assessing the weld quality.

Chapter 3

Methodology

This chapter delves into the details of the setup used to record the welding process images and the processing algorithm. The details include the general schematics of the system, TIG welding process parameters and information about specific material composition, the images details, dataset composition, neural networks architectures, training and evaluation.

The chapter starts with the general layout of the welding system, the welding equipment and the TIG welding process application on the material. The latter part of this chapter describes, in details, the setup for the image processing workflow based on machine learning from starting with model generation, through hyper-parameter optimisation, training procedure and evaluation metrics.

3.1 Schematics

The guiding principles shaping the initial requirements in terms of hardware and image analysis reduces to the following points:

- system simplicity - the system composition of ideally two part: acquisition and processing. The literature is rich with the combinations of sensors (from unidimensional to multidimensional data generation), lasers, control systems and synthesis subsystems (sometimes requiring synchronisation) working in conjunction. There is nothing wrong with having a complex system, but for exploratory speed, a simpler system provides a better starting base

- high image quality - the sensor requirement to provide clear images, without any auxiliary source of illumination and ideally without (or minimal) filtering
- adaptive analysis - the processing paradigm requirement to adapt to changing luminosity conditions and weld aspect, typical to the welding environment

The research focused on the camera and image processing units and singled out the two main components of the idealised system in Figure 3.1.

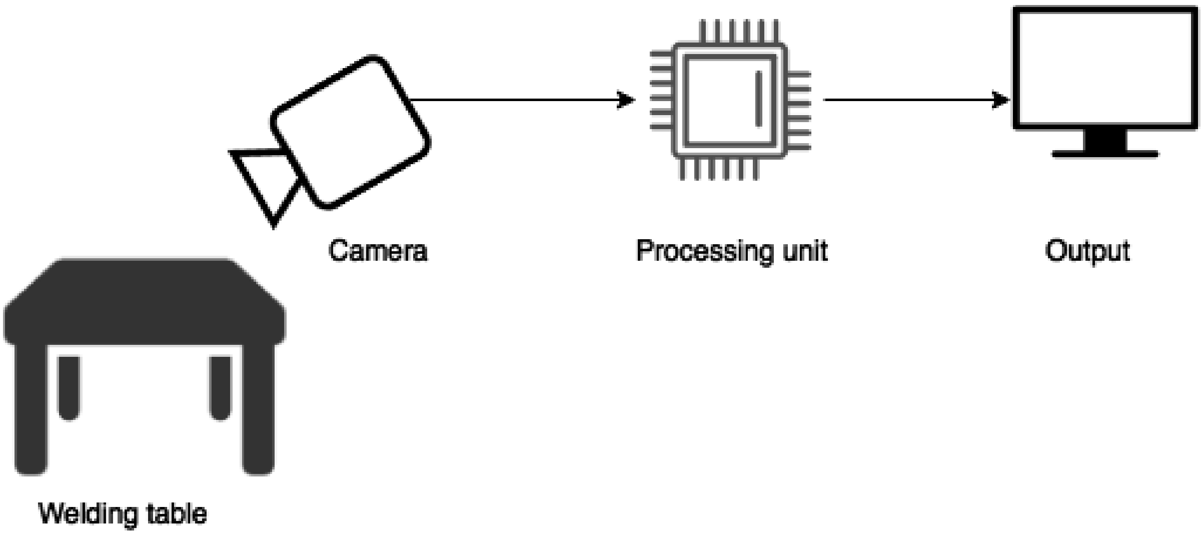


Figure 3.1: Simplified schematic

3.2 Camera

The state-of-the-art in image acquisition dedicated to arc and weld pool monitoring are cameras with high dynamic range. The dynamic range is a property of the sensor sensitivity to the light. The image appears more balanced, the bright areas being darker and dark areas being brighter, relatively to the incoming light. The sensor, exploiting this characteristic, could offset the arc light without any additional filter or laser illumination. The advantages are as follows:

- simpler and cheaper system due to the exclusion of the laser illumination. The laser illumination brings a whole new range of variables including but not limited to position, power and synchronisation
- the retention of information by avoiding the filtering.

Xiris XVC-1000, shown in Figure 3.2, provides the images in the current study.

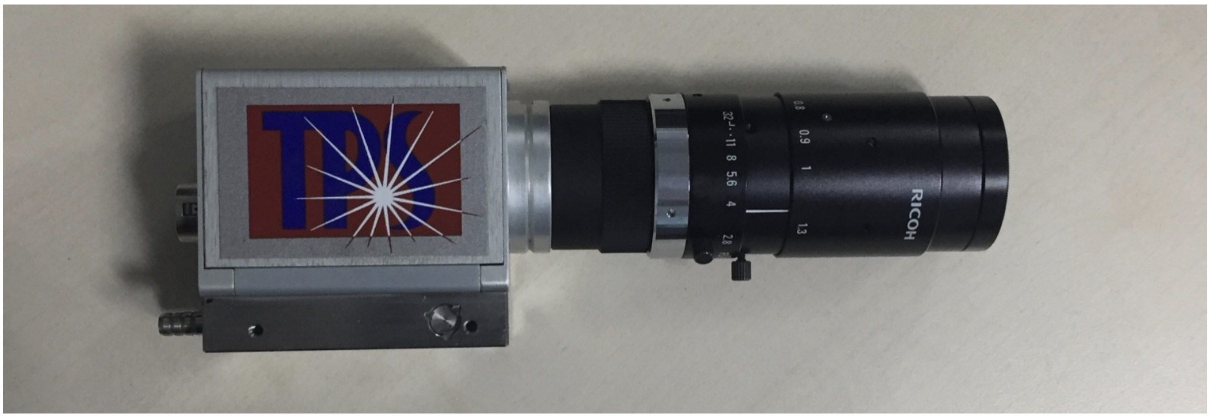


Figure 3.2: Xiris XVC-1000 camera

3.3 Welding process

The approach for defect identification involves utilising a camera trailing the welding process oriented directly towards the weld pool to obtain the real-time images. The camera position, behind the welding direction, provides visibility over the weld pool solidification front, welded solidified material, the arc and the torch position relative to the workpiece. The camera position in front of the welding direction also helps with positioning, but the arc masks the weld pool and solidified material aspect.

This study uses one single camera, mounted on a robot as shown in Figure 3.3.

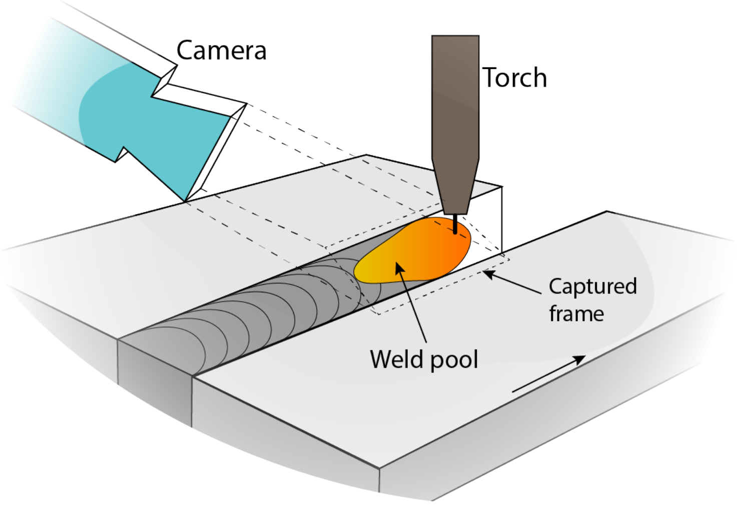
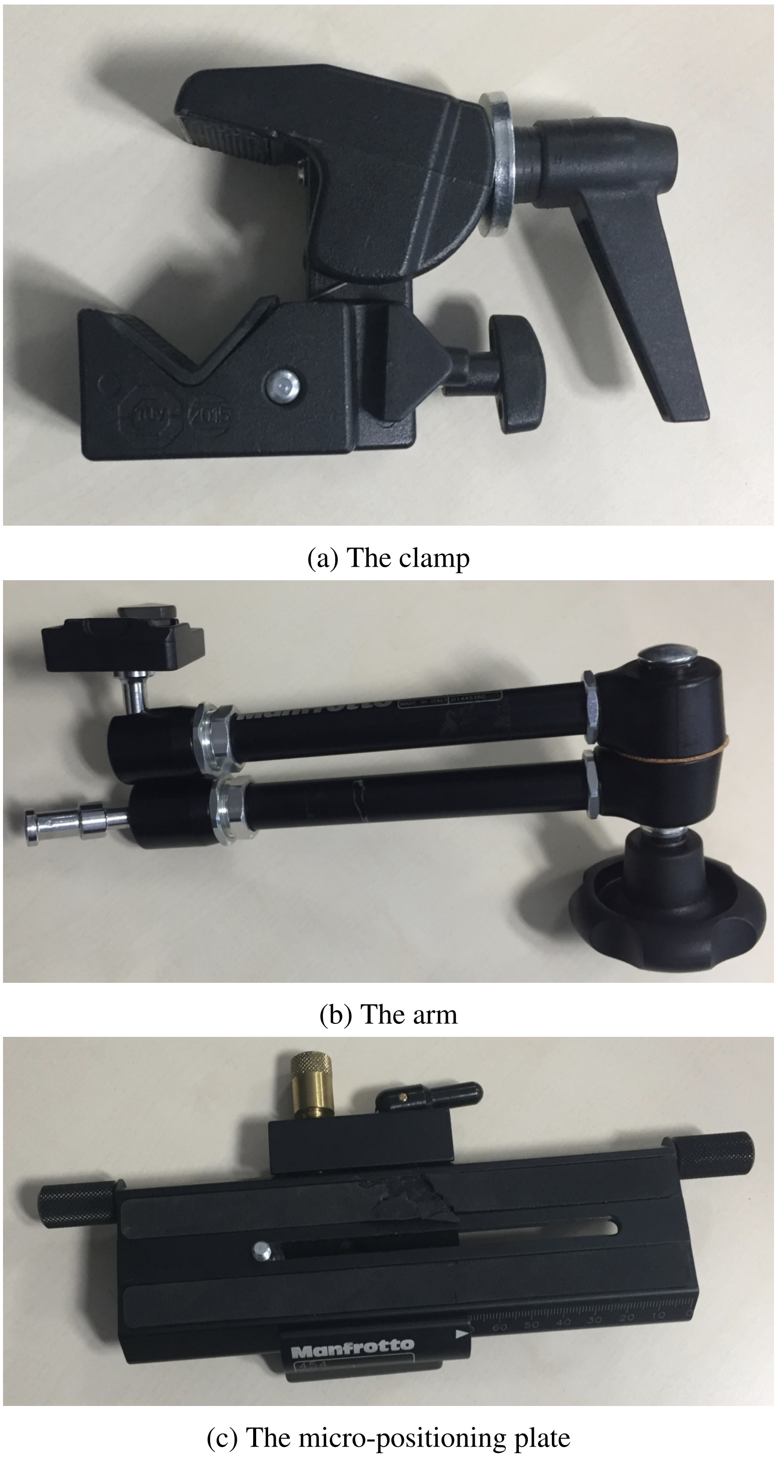


Figure 3.3: Idealised TIG welding process layout.

The clamp, Figure 3.4a, attaches to the robot, the arm, shown in Figure 3.4b, fixes at one end to the clamp and at the other end on the micro-positioning plate, shown in Figure 3.4c, at the other end of the arm. The camera screws into the micro-positioning plate as in Figure 3.5. The role of the micro-positioning plate is to achieve a more granular degree of freedom compared to the arm, due to the small camera field of focus.

The camera connects to the processing unit via an Ethernet cable. The same cable carries the power to the camera and data transfer from the camera to the processing unit. The camera uses a 20mm spacer and 75mm lens.

The current study uses two different widely used materials for paradigm validation. The first material is stainless steel grade 304 (SS304), used for food, dairy and pharmaceutical production equipment due to excellent corrosion resistance property as well as everyday tools as saucepans and tube for good weldability. The second material, aluminium alloy 5083 (Al 5083), is widely used due to its corrosion resistance to the seawater and the industrial chemicals as well as excellent post-weld properties. The material is used typically for shipbuilding, vehicles bodies and pressure vessels.



(a) The clamp

(b) The arm

(c) The micro-positioning plate

Figure 3.4: The clamp, arm and micro-positioning plate

Figure 3.5 shows the camera mounted on a Fanuc ARC Mate 100iB robot.

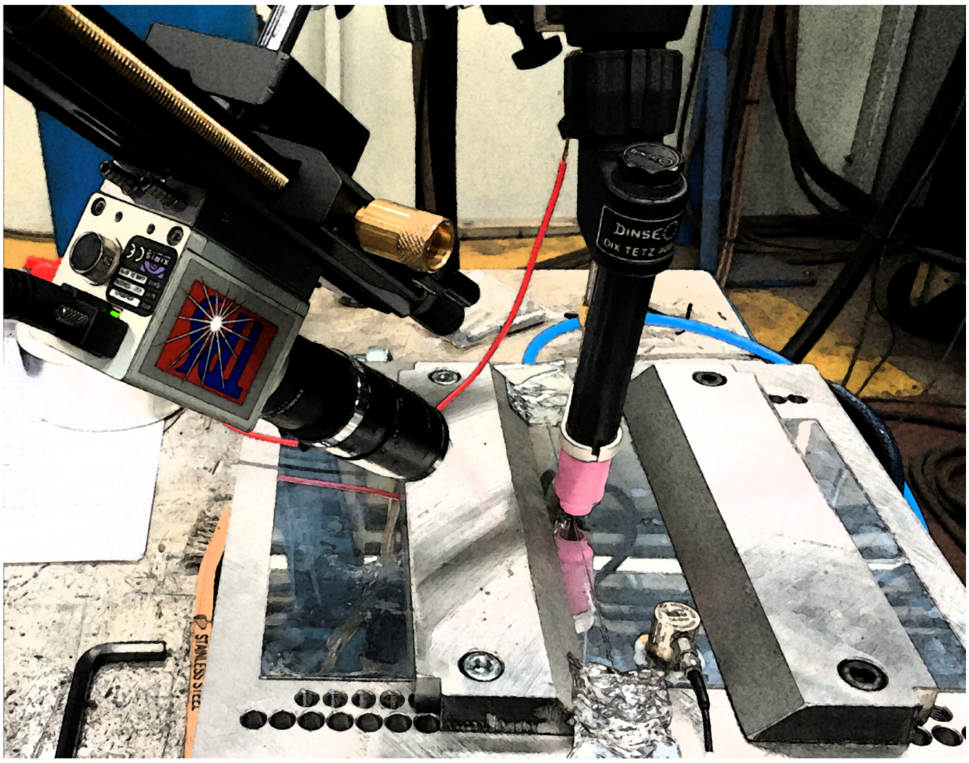


Figure 3.5: Camera and robot setup.

3.3.1 Stainless steel 304

One of the materials used for carrying out the welds was stainless steel grade 304 (SS304) plates of thickness 5mm and 10 mm. Table 3.1 details the stainless steel plates composition for the experimental work. SS304 is a versatile and widely utilised material in a large range of applications due to its good formability and weldability characteristics.

Table 3.1: Stainless Steel 304 composition.

Element	%
Fe	Balance
Cr	18.00-20.00
Ni	8.00-12.00
Mn	2.00 max
C	0.08 max
P	0.045 max
S	0.03 max
Si	0.75 max
N	0.10 max

In general, austenitic stainless steel is considered weldable by conventional fusion techniques although special attention is required towards ensuring the formation of ferrite in the weld deposit, otherwise ”hot cracking” might occur. During all tests carried out the welding process involved was TIG welding.

The trials generated images of six defects starting with ”good weld” and base parameters described in Table 3.2. The other defects emerged as the parameters deviated from base welding parameters as described in Table 3.2. An example is an increase in the heat input, rendering ”burn through”, while a decrease produced a ”lack of fusion” defect. The production of other defects involved increasing the travel speed, introducing contaminant or turning off the supply of shielding gas.

Table 3.2: TIG welding process parameters.

Parameter	Baseline	Deviation from baseline
Gas flow rate (l/min)	30	[10, 15, 35, 40]
Traveling speed (cm/min)	19	[10, 10.5, 16, 23.2, 24.8, 26.4, 33.4, 50]
Voltage (V)	17.2	[12, 22]
Amperage (A)	200	[100, 150, 220, 235, 250, 270, 275, 300]

3.3.2 Aluminium alloy 5083

The second material used in this study is aluminium alloy 5083 (Al5083). The magnesium is the main element alongside aluminium. Table. 3.3 details the plates composition. Aluminium alloy 5083 grade was selected as the work material because it is extensively used in TIG welding. The plates thickness is 2mm, and the groove angle is 90°. Table 3.4 lists the welding parameters

Table 3.3: Aluminium alloy 5083 composition.

Element	%
Al	Balance
Mg	4.00 - 4.90
Mn	0.40 - 1.00
Si	0.40 Typical
Zn	0.25 Typical
Ti	0.15 Typical
Fe	0.40 Typical
Cu	0.10 Typical

used in this work as a standard ”control” to achieve good welding conditions.

Table 3.4: Standard control welding parameters.

Current (A)	90
Travel Speed (cm/min)	35
Voltage (V)	12
Argon flow rate (L/min)	15

Figures 3.6 and 3.7 show the distribution of the input current and the travel speed for each class examined in this study.

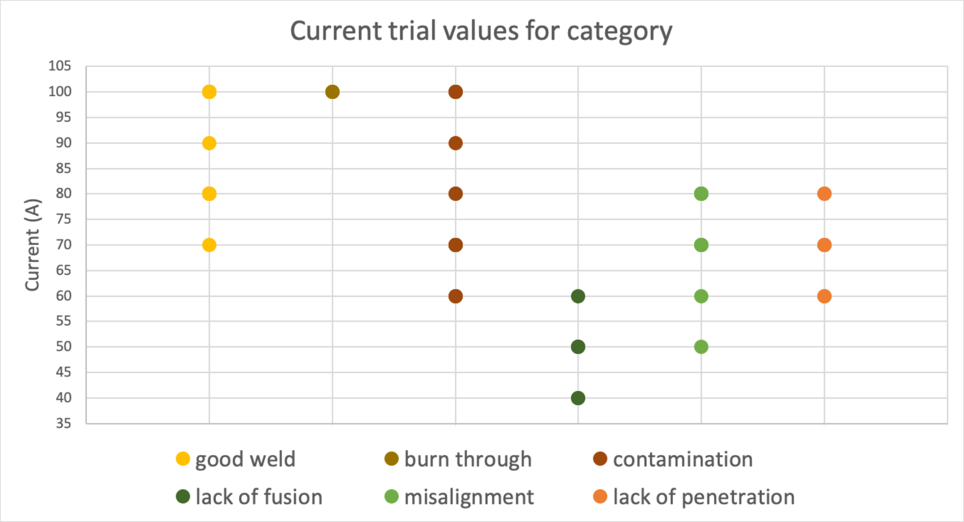


Figure 3.6: Current values for each category

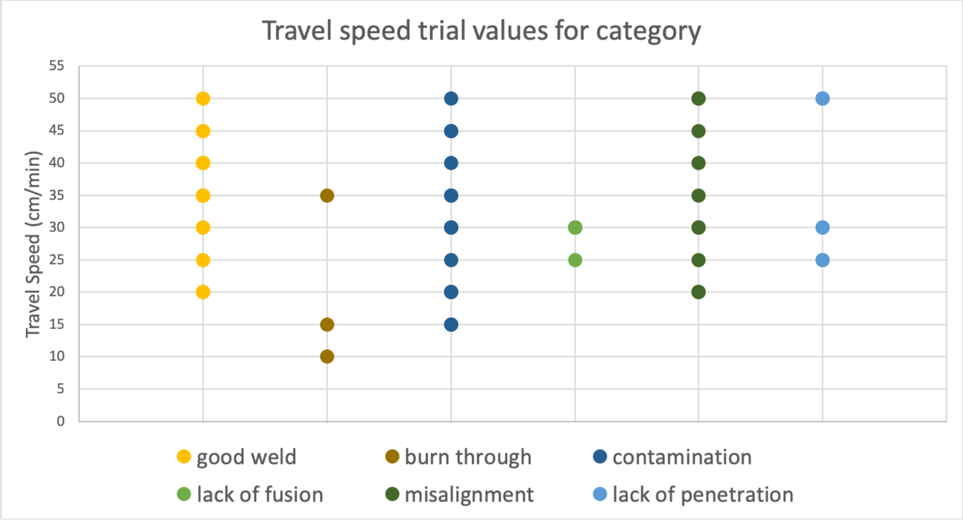


Figure 3.7: Travel speed values for each category

3.4 Images

An experienced welder classified the images in the current research study according to a flaw present and visible during welding. The flaws relate to the aspect of the weld pool (lack of weld pool for ”burn through”), the aspect of the solidified metal in the proximity of the weld pool (”contamination”), as well as the welding procedure (”high travel speed”, ”lack of fusion”, ”misalignment”). Although there is a line between weld defects and welding procedure defects, at the moment of welding the welder does change slightly the procedure to induce the defects development.

The defects taken into consideration for this study are visible and continuous defects. It involved inducing the defects and allowing the process to persist in the flawed state in order to obtain the footages representative of the targeted defect.

The signal processing unit most capable of adapting to the ever-changing conditions, parameters and quality requirements for TIG welding is ANN-based. Neural networks are implemented using Pytorch [137] library, one of the many freely available frameworks for building and executing operations on graph structures.

3.4.1 SS304 dataset

The study produced defective and non-defective welds, representative of the weld conditions. The SS304 dataset consists of 30,008 images [1] originating from 56 welding runs. The number of runs and images differs because the camera records at 55 frames per second, generating desired outcome representations at different stages during the weld. Each run is distributed to either train, validation or test subset. The assurance is necessary to reduce the correlation between subsets and assess the capacity of the networks for defect adaption and representation. Table 3.5 and Table 3.6 describe the dataset composition used for the two scenarios studied. The six-class scenario (5 defects + good weld) is a reduced dataset due to the discrepancy in the number of samples in each class. There still is an imbalance in the dataset, although reduced, by eliminating samples from classes with abundant data.Figure 3.8 shows the images representative of the six classes assessed in this study.

[1] https://www.kaggle.com/danielbacioiu/tig-stainless-steel-304/

Table 3.5: Dataset split between training, validation and test for 2-class test.

	Number of samples		
Category	Train	Validation	Test
good weld	7871	3347	3812
defect	8452	2994	3532
Total	16323	6341	7344

Table 3.6: Dataset split between training, validation and test for 6-class test.

	Number of samples		
Category	Train	Validation	Test
good weld	954	875	769
burn through	977	646	731
contamination	967	339	576
lack of fusion	1005	780	744
lack of shielding gas	196	102	102
high travel speed	630	346	249
Total	4729	3088	3171

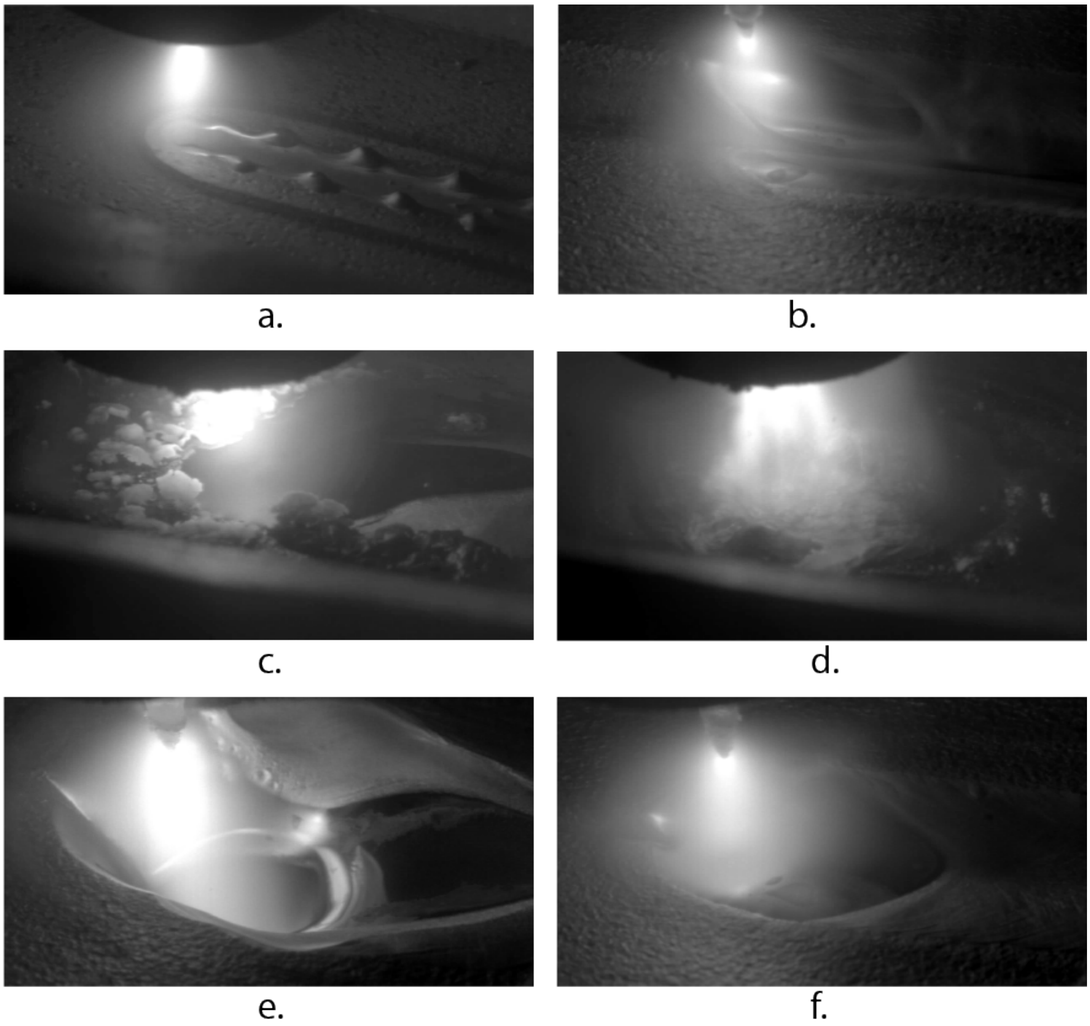


Figure 3.8: Training samples. a) high travel speed, b) no fusion, c) contamination, d) no shielding gas, e) burn through, f) good weld.

SS304 preprocessing

Before processing the image, cropping is applied to the image to remove the area covered by black pixels bringing little information related to the weld quality. An example of the operation is shown in Figure 3.9. Although the camera could deliver frames at a resolution of 1280×1024 pixels, the image dimension reduces to 1280×700 pixels after cropping.

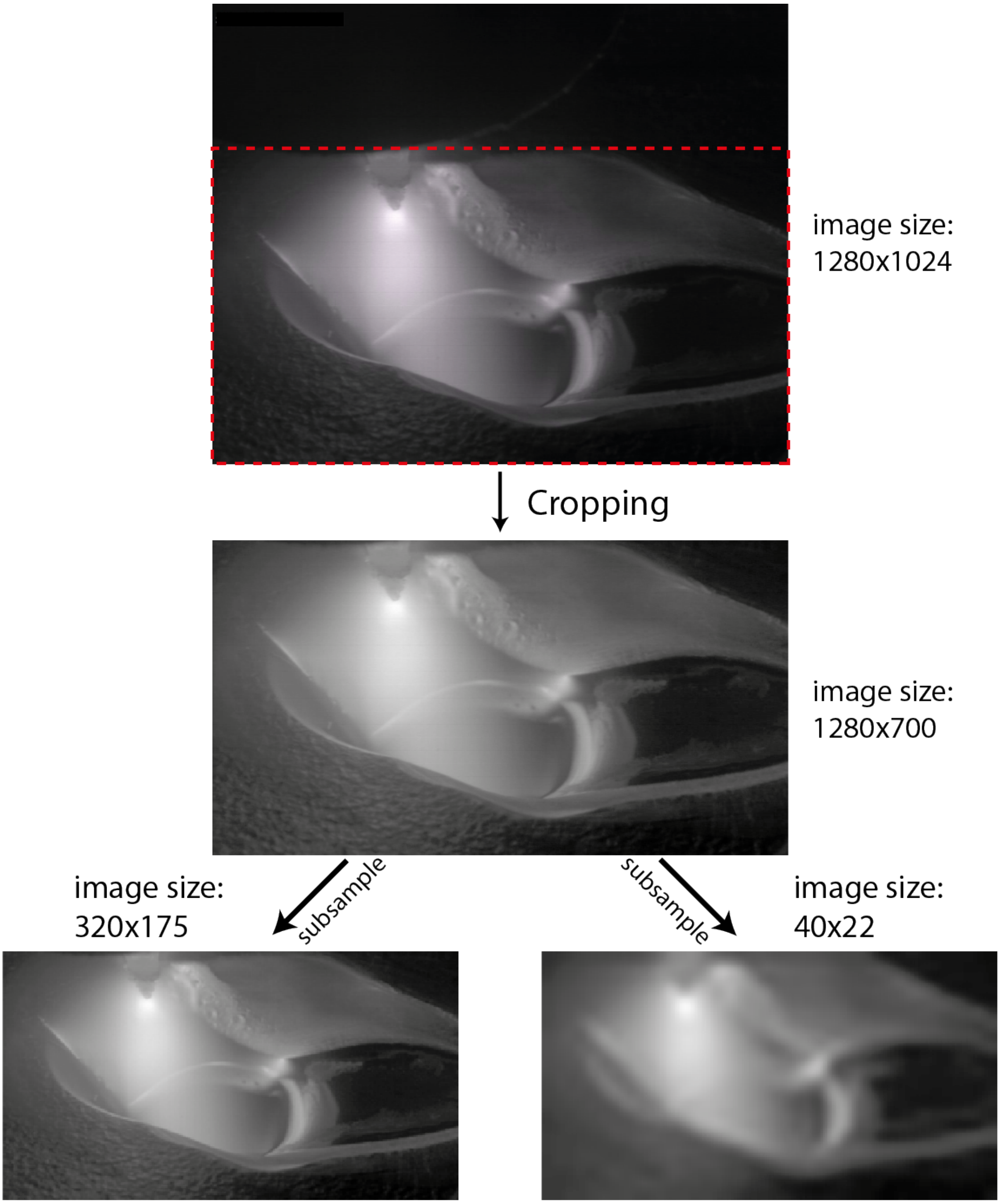


Figure 3.9: Input image preprocessing.

The size at which neural networks processes the image is 175×320 due to processing unit hardware limitation. The neural network models remain small, hardware constraints are relaxed and processing time reduces.

The image subsampling has the potential of affecting the networks' accuracy performance adversely. The hypothesis is tested by subsampling the images from 1280×700 to 40×22 pixels then upsampling to 175×320. The results of the training are compared against the baseline results of training with images subsampled directly to 175×320 pixels.

3.4.2 Al 5083 dataset

This study proposes alongside the SS304 dataset, the Al5083 dataset. Figure 3.10 shows samples extracted from the dataset.

The dataset comprises 60 welding trials, generating images at 55 frames per second, resulting in a large amount of data, relatively quickly. The dataset contains 33254 images [2] of TIG welding of aluminium, divided into six classes, listed in Table. 3.7, by an experienced welder.

Table 3.7: Dataset split between training and test for 6-class test.

Category	Number of samples	
	Train	Test
good weld	8758	2189
burn through	1783	351
contamination	6325	2078
lack of fusion	4028	1007
misalignment	2953	729
lack of penetration	2819	234
Total	26666	6588

The processing architecture used requires the data to split into two main subsets: training and testing, 75% and 25% share, respectively.

[2] https://www.kaggle.com/danielbacioiu/tig-aluminium-5083

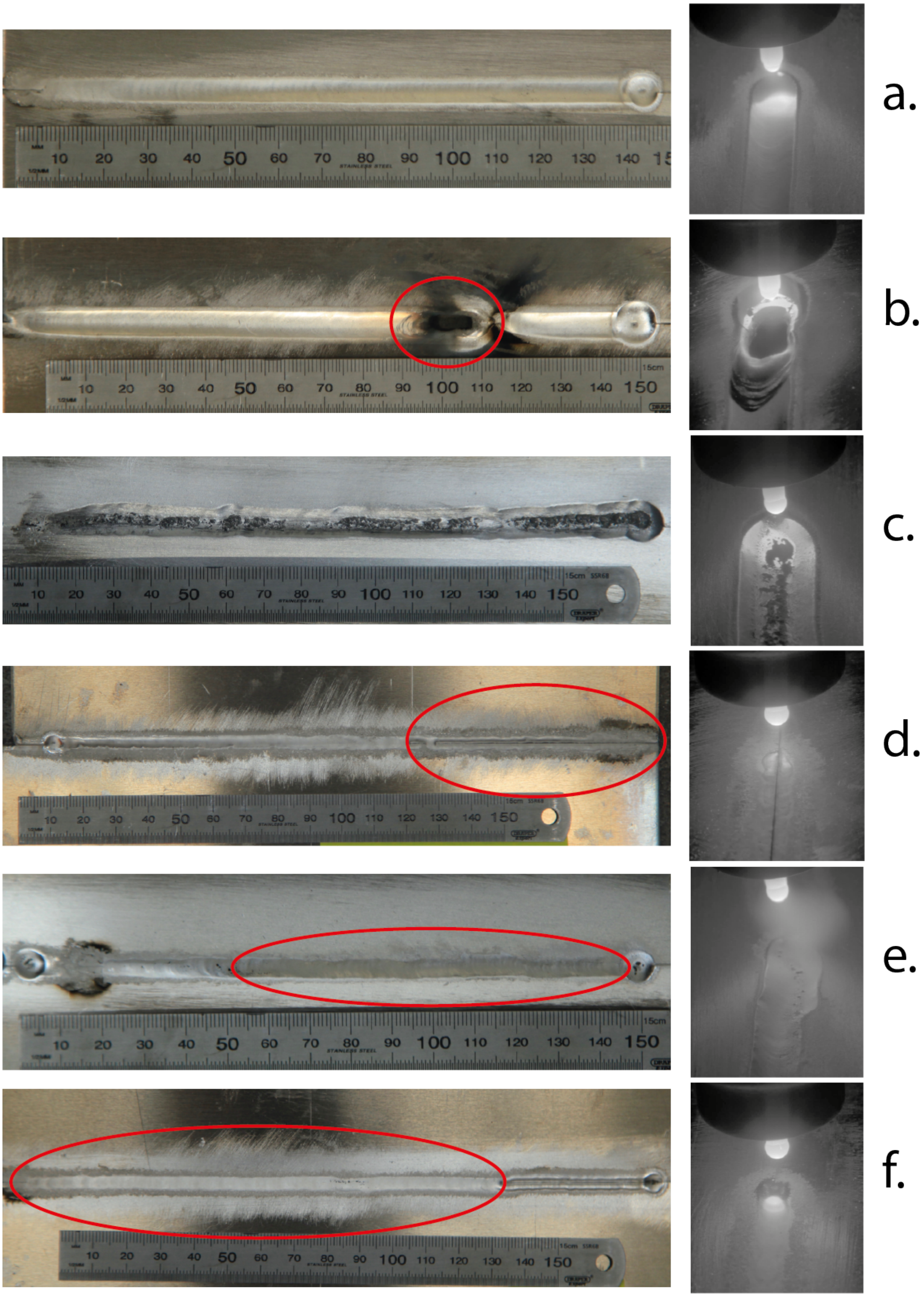


Figure 3.10: Dataset samples of aluminium TIG welding. a) good weld; b) burn through; c) contamination; d) lack of fusion; e) misalignment; f) lack of penetration

Tables 3.7, 3.8, 3.9 describe the data divisions for 6-class, 4-class and 2-class analysis, respectively. The 4-class analysis used an 88%-12% split due to limitations in the generated data. The number of images of certain defects is smaller, therefore for a balanced composition between all four classes, certain images representing good weld, for example, were removed from the dataset.

Table 3.8: Dataset split between training and test for 4-class test.

Category	Number of samples	
	Train	Test
good weld	3763	427
burn through	1783	351
contamination	2918	396
lack of fusion	4182	402
Total	12646	1576

Table 3.9: Dataset split between training and test for 2-class test.

	Number of samples	
Label	Train	Test
good weld	8758	2189
defective	17908	4399
Total	26666	6588

As each of weld generates thousands of images, the problem of correlation between the train and test splits became apparent. In order to reduce the correlation, no weld trial can reside in both, the train and test subsets. Therefore the dataset split based on welding trial not by frame setting.

All 6-class, 4-class and 2-class datasets originate from the same set of welding experiments. Table 3.5 ”defective” category incorporates all the defects from Table 3.6 under one single class.

Al 5083 preprocessing

The camera recorded the images at the resolution 1280×1024 pixels, centred on the weld pool. The images contain a substantial amount of black surrounding the weld pool and the welding arc as seen in Figure 3.11. Therefore, the cropping reduces the original size of 1280×1024 to 800×974. Further to cropping, the images are subsampled, reducing the size to 400×487. The subsampling operation is necessary because of the hardware constraints during the training stage of the networks. The model receiving a higher resolution requires significantly more GPU memory.

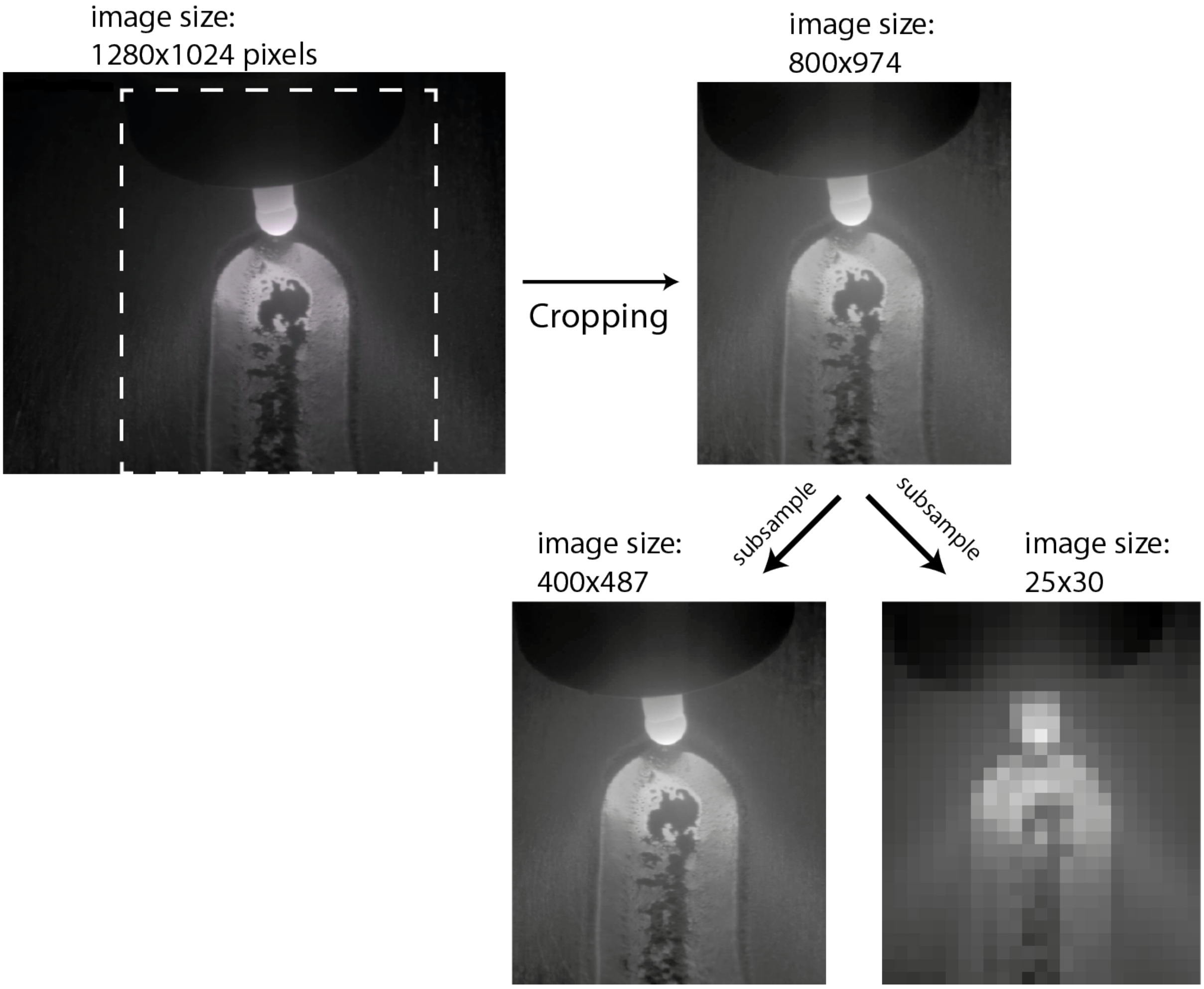


Figure 3.11: Image subsampling

This study performs an ablation analysis on the effect of resolution reduction on the final accuracy. The images of 400×487 pixels are subsampled to 35×30 pixels and unsampled back to 400×487. The subsampling followed by upsampling has two reasons: it maintains the input image size of 400×487 pixels for neural networks and removes details from the baseline image of 400×487. The result is the isolation of fidelity as the only changing parameter across comparison.

3.5 Architecture design

The current study addresses the neural networks feasibility for welding conditions classifications using two types of neural network (NN) typologies: Fully Connected Neural Networks (FCN) described in Subsection 2.4.4 and Convolutional Neural Networks (CNN) described in the same subsection. Internally, the two architectures develop a function that optimises the probability distribution of images aspect over the labels. The design of the neural network typology required decisions with regards to the hyper-parameters defining the network. The hyper-parameters are specific to each type of network. The FCN hyper-parameters are:

- NN internal hyper-parameters:
  - number of input nodes
  - number of hidden layers
  - number of nodes in each hidden layer
  - number of output nodes (given by the number of categories)
  - node's activation function (e.g. ReLU, sigmoid or tanh)
- NN optimisation hyper-parameters:
  - optimization algorithm (e.g. standard gradient descent, Adam, AdaGrad or RMSProp)
  - learning rate
  - decay rates for calculating first moment and second moment estimates
  - distribution for weights initialisation
  - number of epochs for training
  - number of samples for each batch for gradient descent

Conversely, the CNN overlap partially with FCN in terms or hyper-parameters requirements, but they also have their specific addition:

- NN internal hyper-parameters:
  - number of hidden layers

  - number of kernels in each hidden layer
  - kernel size in each hidden layer
  - kernel stride in each hidden layer
  - kernel's receptive field
  - node's activation function
- NN optimisation hyper-parameters:
  - optimization algorithm type
  - learning rate
  - decay rates for calculating first moment and second moment estimates
  - distribution for weights initialisation
  - number of epochs for training
  - number of samples for each batch

The search space, therefore, is not restricted to finding the NN weights only, but the hyper-parameters themselves also define a search space, called the hyper-parameters search space.

3.6 Hyperparameter optimization

The two main hyper-parameters search spaces are model hyper-parameters and training procedure hyper-parameters (optimisation hyper-parameters).

Chapter 4 proposes two architecture without a detailed analysis of the hyper-parameters space exploration providing the proof of concept that NN are suitable for classifying welding conditions of SS304. That is not to say some trial and error did not take place and the hyper-parameters are random. The methodical and structured approach for hyper-parameters search space was not in place when the trials took place. Therefore the reported performance describes the best candidate. In Chapter 5, the full details of the search space is traced and documented together with in-depth analysis of performance variation while Chapter 6 approaches the problem of hyper-parameter optimisation in the same manner as network's internal weights, applying gradient descent for achieving an even more optimal model typology. It is important to mention here that the automatic typology search does not cover the training procedure hyper-parameters, but it is limited only to the NN internal layout (or model hyper-parameters).

Choosing the initial starting point for selecting hyper-parameters follows the research carried out in the machine learning field by Krizhevsky et al. [65] with AlexNet and Simonyan et al. [138] with VGGNet, two landmark architectures. The NN performing well in practice tends to be composed of at least two layers of fully connected layers for FCN or several layers of alternating convolutions and maximum sampling for CNN. As such, a large portion of the networks seen in the current study follows the repetitive nature of stacked layers, some with very few alterations between layers' hyper-parameters. The upper bound for the NN size (number of layers) in the current study limits the models to the amount of GPU memory, which is 4GB. The NN was designed using the available frameworks with higher level abstraction libraries, dedicated to NN research and development as Tensorflow [139] and Pytorch [137]. The libraries provide a simplified interface for building nodes' internal logic, define layers, and flexible methods for linking and stacking the layers together. The frameworks have the advantage of accelerated back-end implementations dedicated to using GPUs for speeding up the matrix multiplications.

3.7 Weights optimization

The optimisation refers to the training of neural networks, the internal weights optimisation. It takes place after all hyper-parameters definition, and crucially after the choice of the optimisation algorithms. The current study uses the Adaptive Moment Estimation (Adam) [103] optimisation algorithm. It is a gradient-based algorithm for computing adaptive learning rates for each weight (neural networks have millions of parameters (weights) in the current study) while keeping track of the gradients first moment (the mean or decaying average of past gradients) and gradients second moment (uncentered variance or decaying average of past squared gradients). Adam intuitive explanation for the navigation path in the parameters' multidimensional landscape is analogous to a ball rolling downhill (towards the minima) with the moments acting as an inertial force resisting any sharp deviation from the course.

The main optimisation hyper-parameter is the learning rate, and it is one of the most important hyper-parameter in the NN definition. The optimal value for learning rate varies with the NN depth (number of layers), batch size (smaller batch requiring smaller learning rate) but there is no specific rule or approach for identifying the best learning rate. The value typically varies between 10^{-1} and 10^{-6}.

The convergence towards the solution uses the loss gradient (g_t) and back-propagates it through the network. The back-propagation alters all the weights for the nodes for minimising the total loss. Successive forward-backwards passes through the network successfully lead in most cases to the solution or very close approximation. Three parameters influence the back-propagation by controlling the steps size at which the overall network converges to the solution: learning rate (η), exponential decay rates for the first moment estimate (β_1) and exponential decay rates for the second moment estimate (β_2). Adaptive Moment Estimation (Adam) [103], the optimisation algorithm of choice, works as follows:

- calculate exponential decay average of past gradients (g_t)

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \tag{3.1}$$

- calculate exponential decay average of past squared gradients

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \tag{3.2}$$

- compute bias corrected first moment estimate

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{3.3}$$

- compute bias corrected second moment estimate

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{3.4}$$

- update weights

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} \tag{3.5}$$

The cross-entropy difference is the loss between the label assigned by the network for a specific example and the correct label for the same example. Mathematically, cross entropy measures the difference between two probability distributions, defined as:

$$H(p, q) = -\sum_{x} p(x) \log q(x)$$

where $p(x)$ is the desired distribution and $q(x)$ is the current distribution.

Suppose there is an example of an image where the label is ”contamination” where all classes of classification are: ”good weld”, ”burn through”, ”contamination”, ”lack of fusion”, ”lack of shielding gas”, ”high travel speed”. The ground truth vector looks as follows:

Table 3.10: Example of ground truth for one single image.

Label	Classification
good weld	0
burn through	0
contamination	1
lack of fusion	0
lack of shielding gas	0
high travel speed	0

The vector for correct labelling (ground truth) populates with zeros everywhere excepting the position corresponding to the correct label, meaning the probability for ”contamination” is 100% and the probability for any other defect is 0%. The NN tries to match the output from the ground truth during training. Suppose at time t, during training, the NN probability distribution looks as follows:

Table 3.11: Example of NN predicted probability distribution for one image.

Label	Classification
good weld	0.09
burn through	0.05
contamination	0.72
lack of fusion	0.04
lack of shielding gas	0.05
high travel speed	0.05

The cross-entropy loss is 0.14, calculated as follows:

$$H = -(0 * \log(0.09) + 0 * \log(0.05) + 1 * \log(0.72) + 0 * \log(0.04) + \\ + 0 * \log(0.05) + 0 * \log(0.05)) = 0.14$$

The cross-entropy plugs into the simplified weights update rule as follow:

$$\theta_i = \theta_i - \eta \frac{\partial}{\partial \theta_i} J(\theta)$$

where $J(\theta)$ is the cross-entropy loss, $H(p, q)$, parametrised by θ and θ_i is one single weight out of millions. Note, the gradient calculation of the (g_t) with respect to each parameter, described above as $\frac{\partial}{\partial \theta_i} J(\theta)$, follows the steps defined in Equations 3.1 to 3.5, but for simplicity and brevity all those steps were compressed into a neat notation.

To summarise, the optimisation algorithm tries to find the NN weights w_i, such that, multiplying the pixels p_i with the weights the output of the last NN layer looks as in Table 3.10. In order to allow the NN the flexibility to reproduce more functions, a non-linear operation needs to be included at the end of each multiplication $w_i \cdot p_i$. The function maps the linear output domain of the multiplication to another non-linear domain. Such functions are the the activation functions in Subsection 2.4.4, *sigmoid*, *tanh*, *ReLu(or rect(a))*. The current study uses *ReLU* as the activation function.

The power of the non-linearity to represent more complex, and the effects of excluding it are as follows:

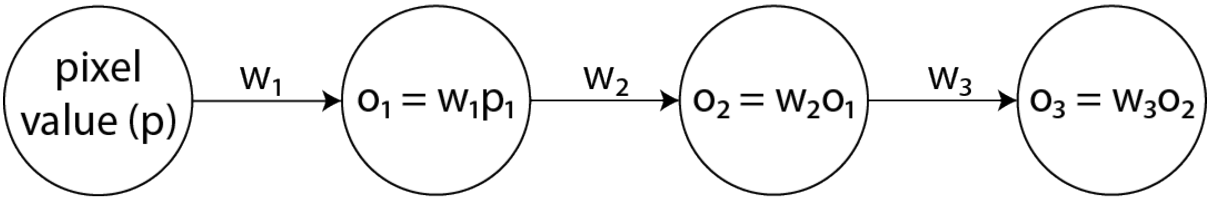


Figure 3.12: Example of a NN internal multiplication without the introduction of non-linear behaviour.

Effectively, the output (o_3) is:

$$\begin{aligned} o_3 &= w_3 o_2 \\ &= w_3(w_2 o_1) \\ &= w_3(w_2(w_1 p)) \\ &= w_3 w_2 w_1 p \end{aligned}$$

Since the multiplication $w_1 w_2 w_3$ is a single number, the entire network is no better than one single node.

The last NN layer (o_3) should resemble (ideally) Table 3.10, where o_3 domain is $\mathbb{R}^K$, where K is the number of output classes. The non-linear activation function sigmoid helps by squashing the output of the node between 0 and 1. On the other hand, the output domain for the non-linear activation function ReLU (the activation function used in the current study) is $[0, \infty)$. The cross-entropy could not be applied directly to any of the activation functions above because they do not output probability distribution, per se the output numbers do not add to 1. To force all output dimensions into the range (0, 1) and their addition to 1, a function called $Softmax$ (or $Normalized\,Exponential\,Function$) applies to the output of the last layer before calculating the cross-entropy loss. Softmax is defined as follows:

$$s(a_i) = \frac{e^{a_i}}{\sum_{k=1}^{K} e^{a_k}} \quad for\ i = 1, \dots, K \tag{3.6}$$

3.8 Evaluation

The model evaluation takes place during the training stage as well as the testing stage (after training). The primary figure for assessing the performance of each NN trained during the study is NN accuracy measured as:

$$accuracy = \frac{correct\ predictions}{total\ samples} \tag{3.7}$$

The accuracy is tracked during training to ensure the NN converges towards a minimum loss (maximum accuracy), in effect validating the hyper-parameters choice for the particular NN instance. After each epoch (one iteration through entire training subset) a validation step is performed (using validation subset) reporting the validation accuracy. The validation stage does not calculate the derivatives, the NN weights are frozen, and loss is not back-propagated through the network. The step is similar to the testing stage but using a different data subset. Another role for the validation subset is to checks the NN overfitting after each epoch.

The overfitting behaviour for NN becomes apparent when the training loss continues decreasing (approaching zero loss and 100% accuracy) while validation accuracy drops. It is equivalent to NN ”memorising” the training dataset without achieving the probability distribution desired. The weights are not allowed to be modified during validation for the network not to apply the same ”memorisation” to validation dataset. The validation data subset is different from testing data subset because the number of epochs is part of hyper-parameters and testing assess the entire combination (only once, at the end) of model internal weights and hyper-parameters.

The validation subset appears only in Chapter 4 due to initial concerns over overfitting and dropped after because the behaviour appears very rarely. Therefore the networks tend to stabilise and not improve the accuracy rather than exhibit a drop in accuracy. The validation subset moves in the training subset.

The accuracy as a metric has the benefit or reducing the NN performance to a single number, allowing easy comparison between NN. At the same time, it hides the imbalances in the dataset (when one type of defect is disproportionally represented) as it is the case in the current study, shown in Tables 3.6 and 3.7, by providing an accuracy misrepresentative of the underlying probability distribution. One could imagine a dataset where 90% of images are ”contamination” while 10% are ”good welds”, while the NN classifies all images as ”contamination” achieving 90% accuracy. Although the accuracy is correct, the model is nowhere near the correct probability distribution.

This work uses several alternative assessment tools as precision, recall and F-Score for overcoming the obstacle of disproportional class representation and pinning down the model quality with a higher degree of confidence. The calculation of precision, recall and F-score are as follows:

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{3.8}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{3.9}$$

$$F - score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3.10}$$

where *true positives*,*false positives* and*false negatives* are extracted from the confusion matrix:

Table 3.12: Confusion matrix explained.

Ground truth	Predicted categories	
	true	**false**
true	true positives	false negatives
false	false positives	true negatives

Per-class metrics, particularly F-score, have the advantage of taking into account the false positives and false negatives. All other metrics relate to the number of true positives while ignoring the true negatives because they convey little information for multi-class classification. Values closer to 1 denotes models with higher capability for discerning defects, offsetting the effects of the skewed dataset and balancing better the trade-off between high accuracy for one class against low accuracy for another.

Chapter 4

Stainless Steel 304 Weld Analysis with Neural Networks

4.1 Introduction

This chapter describes the NN application on categorising images of TIG welding of SS304. It provides the results obtained after the processing of SS304 images acquired during the study. The present study assesses the suitability of a vision-based monitoring technique based on HDR camera and machine learning for the analysis and categorisation of ”good” vs ”defective” welds as well as specific defect identification. The analysis adopts two neural networks architectures, Fully-connected Neural Networks (FCN) and Convolutional Neural Networks (CNN). This study considers two tests. The first test is discriminating between ”good” vs ”defective” weldments (two-class test), while the second case is a six-class test for discriminating between ’good welds’ vs ’burn-through’ vs ’contamination’ vs ’lack of fusion’ vs ’lack of shielding gas’ vs ’high travel speed’. A test used either 10988 for the six-class test or 30008 for the two-class test, with the dataset split described in Subsection 3.4.1, more precisely in Tables 3.5 and 3.6. Each of the two NN architecture trains on each of the scenario targeted generating four sets of hyper-parameters denoted as Fully-con6 (FCN applied on the 6-class scenario, Fully-con2 (FCN applied on the 2-class scenario), Conv6 (CNN applied on the 6-class scenario) and Conv2 (CNN applied on the 2-class scenario). Analysis section develops on the accuracy and model quality the four sets of hyper-parameters.

4.2 Methodology specific to SS304

The 175 × 320 image unrolls into a 56000 vector before being presented to the neural network. The input to the network is a vector obtained by flattening the image (unrolling) as in Figure. 4.1.

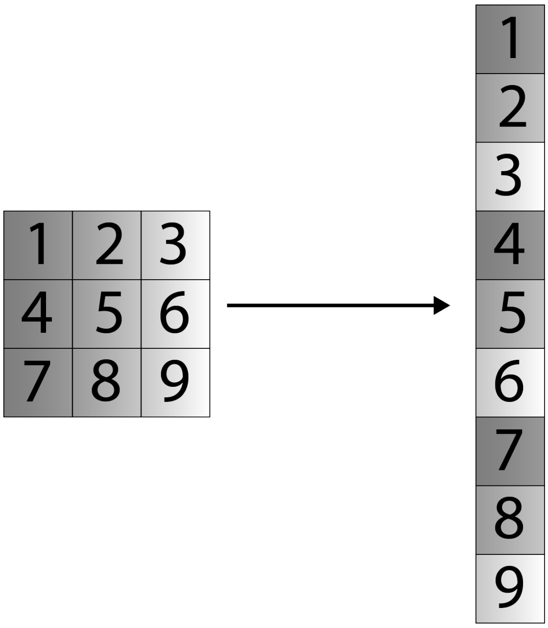


Figure 4.1: Image (3 × 3 grid of pixels) flattening example

Each of 56000 input nodes connects to 64 nodes in the subsequent layer. The second layer connects each of its nodes to each of the 64 nodes from the subsequent (third) layer and so on. Output layer has either two nodes for the good weld vs defective test or six nodes, i.e. 6-class test, indicating which of the classes investigated the image belongs.

4.2.1 Model hyper-parameters

Table 4.1 describes the number of internal nodes in the fully-connected network used in this study.

Table 4.1: Fully-connected Neural Network Architecture considered in current study.

Type	Size
Unrolled image	56000
First fully-connected layer	64
Second fully-connected layer	64
Output layer	{6 or 2}

Typically, the images are compiled in a batch before to the network to speed up the training stage. The test considered in this work use an input of 56000 values (175 × 320=56000), reducing to 64 values in the second layer (hidden layer 1), 64 values in the third layer (hidden layer 2), then two output nodes for the good weld vs defective case and six outputs for the 6-class classification scenario.

Table 4.2 describes the architecture for the convolutional neural networks used in this study.

Table 4.2: Convolutional Neural Network Architecture considered in the current study

Type	Filters	Size
Image	1	175 × 320
First convolutional layer	16	3 × 3 / 2
Maximum pooling		3 × 3 / 2
Second convolutional layer	16	3 × 3 / 2
Maximum pooling		3 × 3 / 2
Third convolutional layer	16	3 × 3 / 2
Maximum pooling		3 × 3 / 2
Fully-connected layer		64
Output layer		{6 or 2 }

In the present study, the architectures investigated consist of three 3×3 convolution layers interlaced with three maximum pooling layers. The maximum pool selects the maximum value from a 3×3 portion sliding across (striding) the image with step two $(/2)$. The stride for convolution layers is also two. Following the third maximum pooling, the resulting $16 \times 1 \times 4$ output is unrolled into a vector and fed into a fully-connected layer of dimension 64, which connects to the final layer formed of either 6 or 2 nodes, depending on the number of categories within training set.

4.2.2 Training hyper-parameters

The optimisation is performed using the Adam algorithm with the hyper-parameters detailed in Table 4.3.

Table 4.3: Optimisation hyper-parameters. η = learning rate, β_1 = exponential decay rate for first moment estimate, β_2 = exponential decay rate for second moment estimate.

Hyper-parameter	Fully-con6	Fully-con2	Conv6	Conv2
η	10^{-5}	10^{-5}	5×10^{-5}	5×10^{-5}
β_1	0.99			
β_2	0.999			
$epochs$	5			
$batch\,size$	10			

4.3 Analysis

The output of the neural network is a decision on which category an individual input image belongs. The metric for measuring the training time is $epoch$ which represents one iteration through the entire training dataset, either 10988 for the six-class test or 30008 for the two-class test.

The current study analyses two architectures for each of the two tests taken into consideration, resulting in four sets of results. The FCN and CNN architectures size is 3.6 million parameters and 5000 parameters, respectively.

Figure. 4.2 shows the training loss for each of the architecture analysed along the training stage.

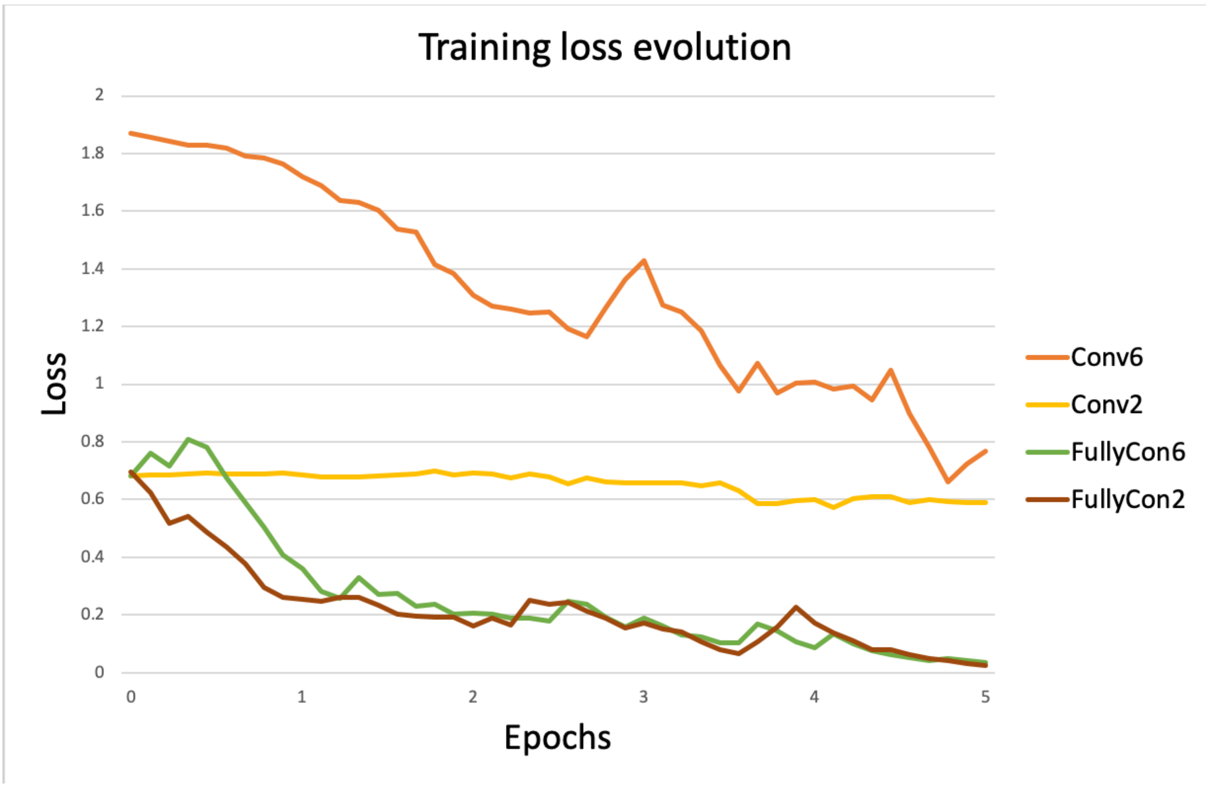


Figure 4.2: Loss evolution during training stage.

The training stage loss evolution suggests the neural networks converge towards a solution. At the same time, the training loss for fully connected architectures is smaller that for convolutional architectures. However, comparing the accuracy after training, FCN architecture builds weaker representations of defects, conclusion arising from studying Table 4.4.

Table 4.4: Testing accuracy

Architecture	Testing accuracy (%)
Fully-con6 Conv6	69 93.4
Fully-con2 Conv2	89.5 75.5

The principal metric for comparing the two architectures in the present study is accuracy. Overall, CNN performs better in terms on pure accuracy compared to FCN when applied to 6-class test, but worst on the 2-class test.

Figure 4.3 shows the accuracy assessed on the validation subset of the models, at different stages during the training stage. All models reach almost maximum performance and stable state very early in the training stage, suggesting the local minima found in parameter space represent well the image categories.

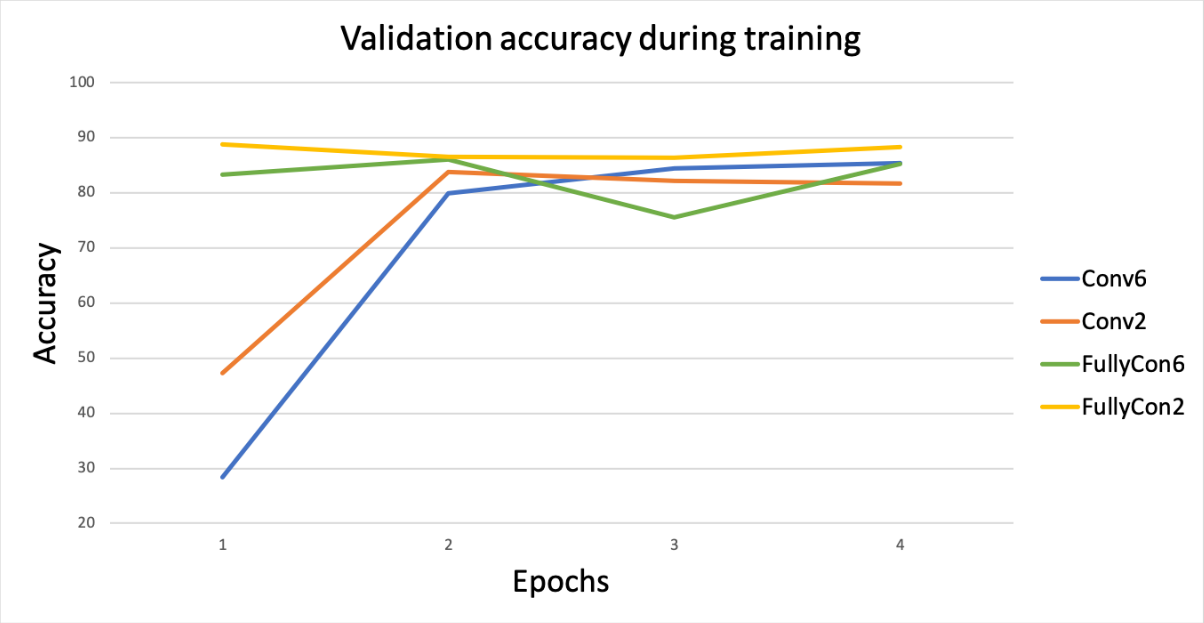


Figure 4.3: Validation accuracy evolution during training stage.

The model accuracy analysis and performance continue with a closer look at the confusion matrices and different per-class metrics. Tables 4.5, 4.6, 4.7, 4.8 are the confusion matrices for Conv6, Fully-con6, Conv2 and Fully-con2, respectively.

Table 4.5: Testing confusion matrix for Conv6 architecture.

	Predicted categories					
Ground truth	good weld	burn through	conta-mination	lack of fusion	lack of shielding gas	high travel speed
good weld	769	0	0	0	0	0
burn through	0	731	0	0	0	0
contamination	16	0	522	0	0	38
lack of fusion	0	26	0	718	0	0
lack of shielding gas	0	1	33	68	0	0
high travel speed	0	0	26	0	0	223

Table 4.6: Testing confusion matrix for Fully-con6 architecture.

	Predicted categories					
Ground truth	good weld	burn through	conta-mination	lack of fusion	lack of shielding gas	high travel speed
good weld	766	0	0	3	0	0
burn through	0	66	0	87	0	578
contamination	0	0	524	0	52	0
lack of fusion	2	0	0	742	0	0
lack of shielding gas	33	1	30	2	36	1
high travel speed	0	0	194	0	0	55

Table 4.7: Testing confusion matrix for Conv2 architecture.

Ground truth	Predicted categories	
	good weld	defect
good weld	3081	731
defect	1071	2461

Table 4.8: Testing confusion matrix for Fully-con2 architecture.

Ground truth	Predicted categories	
	good weld	defect
good weld	3330	17
defect	646	2348

Conv6 outperforms Fully-con6 with the most significant difference observed in the classifying ’burn-though’ and ’lack of shielding gas’ defects. The Fully-con6 model does not recall a consistent part of ’burn-though’ samples while the Conv6 completely misclassifying ’lack of shielding gas’ samples. Taking into account the ’lack of shielding gas’ representation in the dataset, 3.6%, the models weighted the class low during optimisation. Fully-con6 misclassifies ’burn-though’, which could be seen as ’lack of fusion’ or ’high travel speed’ due to the presence of a gap while ’lack of shielding gas’ is a form of ’contamination’.

Tables 4.9 and 4.10 show precision, recall and macro F-score for the models trained and tested on the 6-class and 2-class test, respectively.

Table 4.9: Metrics for 6-class test.

	Conv6			Fully-con6		
Metric	Precision	Recall	Macro F-score	Precision	Recall	Macro F-score
good weld	0.98	1	0.99	0.96	0.99	0.97
burn through	0.96	1	0.98	0.98	0.09	0.17
contamination	0.90	0.90	0.90	0.70	0.90	0.79
lack of fusion	0.91	0.96	0.94	0.89	0.99	0.94
lack of shielding gas	0	0	0	0.40	0.35	0.38
high travel speed	0.85	0.90	0.87	0.09	0.22	0.12
Average	0.77	0.79	0.78	0.67	0.59	0.56

Table 4.10: Metrics for 2-class test.

	Conv2			Fully-con2		
Metric	Precision	Recall	Macro F-score	Precision	Recall	Macro F-score
good weld	0.74	0.81	0.77	0.83	0.99	0.91
defective	0.77	0.69	0.73	0.99	0.78	0.87
Average	0.76	0.75	0.75	0.91	0.89	0.89

The accuracy performance when tested on the 2-class dataset reverses, the convolutional network, Conv2, performs worse than a fully connected network, i.e. Fully-con2.

In case of a dataset where the samples have only one label, as in the current study, the accuracy is the same as micro F-score (not macro F-score). Micro F-score weights each sample (i.e. image), therefore classes with more samples tend to skew the score. Macro F-score weights each class, regardless of the number of samples within the class, therefore offsetting the imbalance. The divergence is visible when applied on the 6-class, and almost identical for the more balanced 2-class test.

4.3.1 Input image resolution impact

Figures 4.4 and 4.5 show the effect of resolution reduction on the accuracy performance for all the neural networks assessed.

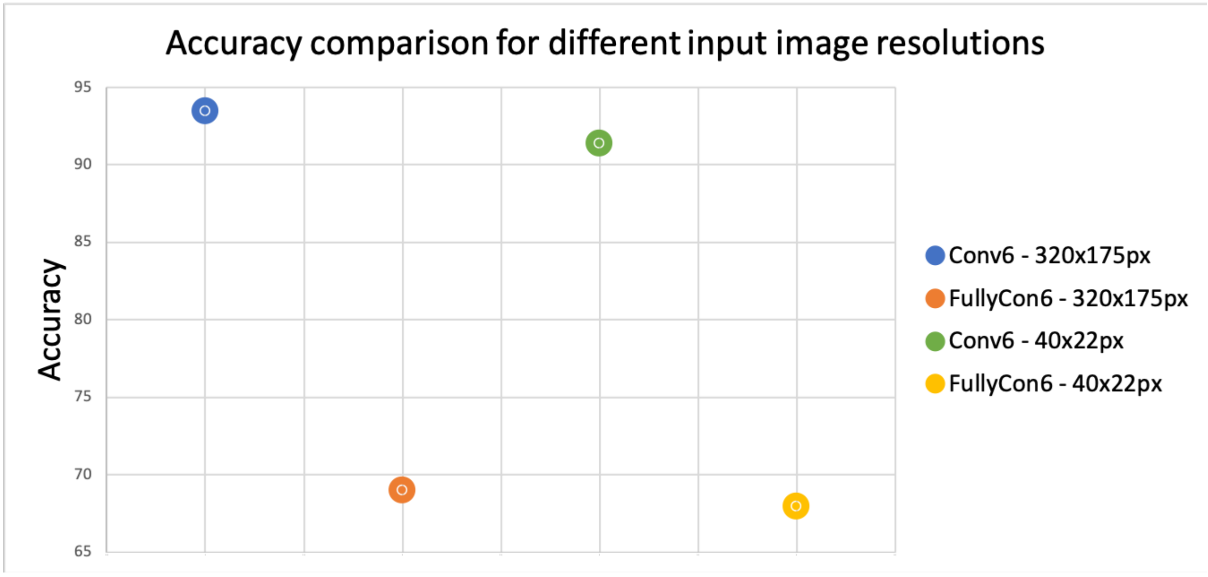


Figure 4.4: Accuracy performance for different fidelities on 6-class test.

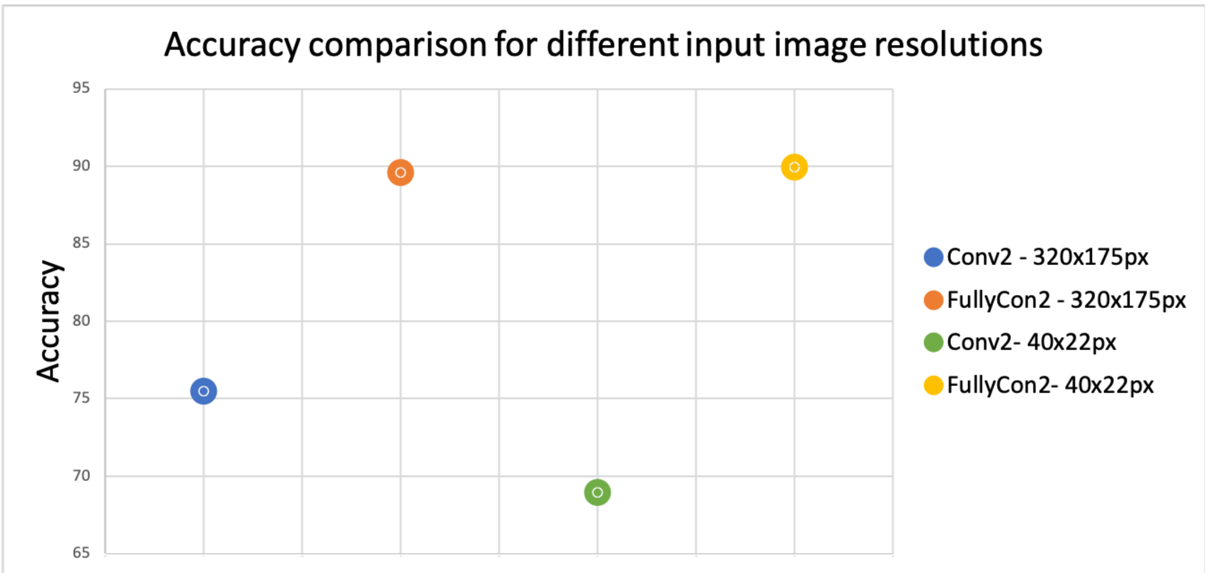


Figure 4.5: Accuracy performance for different fidelities on 2-class test.

The image resolution after cropping is 1280×700. The hardware constraints would not allow assessing the networks at that resolution. Therefore the baseline resolution is 320×175. The assessment of fidelity loss reflected in the networks accuracy performance results from subsampling to 40×22, then upsampling to 320×175. The image losses details of the weld pool and surrounding area and the neural network architectures remain unchanged. The ablation analysis of the effect of fidelity loss shows that convolutional neural networks are affected more than fully connected neural network. It concludes the pixels gradient loss affects the convolutional filters and spatial distribution of features more than the fully connected nodes. The nodes in fully connected networks are agnostic to the value difference between pixels or the features ordering and weight more the values of pixels.

4.4 Conclusion

This chapter presents and discusses the details for constructing a new system for automated monitoring of the SS304 TIG welding using a simple system composed of an HDR camera and a processing unit based on machine learning. The camera successfully filters out the powerful light emitted by the arc while balancing the image and bringing up the details from the weld pool. The processing unit based on machine learning is capable of adapting itself to the process by learning the critical differences between the good and defective welds or recognising specific defects from weldments. The analysis focused on assessing the performance of fully-connected neural networks and convolutional neural networks in classifying weld defects.

CNN has the capability of learning more powerful representations of the defect present and better balance the identification of one defect against misclassification of another. Although superior on accuracy performance, the FCN proves to be more resilient across variations in input fidelity. The present study shows neural networks potential to adapt to industrial requirements contributing to increased productivity, quality and consistency for TIG welding processes.

Chapter 5

Aluminium Alloy 5083 Weld Analysis with Neural Networks

5.1 Introduction

This chapter looks at the results for the application of the neural network paradigm to images of TIG welding of aluminium alloy 5083 (Al5083). The accuracy for classifying the images on the initial trials using SS304 proved that the neural network could distinguish between the different aspects of the welding conditions denoting defects. The data processing in this current chapter uses Al5083 dataset. The dataset acquisition was part of the study, being unique concerning the welding details captured and the range of defects observed. The current chapter aims to study the implications of hyper-parameters (model hyper-parameters and training procedure hyper-parameters) on the accuracy of the overall system for detecting defects. The testing involves training CNN and FCN models of different architectures using a range of learning rates and image resolutions. The chapter analyses the accuracy variation for the changes of the model hyper-parameters, learning rate and resolution, identifying the hyper-parameter that contributes primarily to achieving high accuracy. The chapter includes the range of hyper-parameters where the training of the neural networks rendered model unsuitable for classification. At the end of the chapter, the hyper-parameters achieving the highest accuracy on Al 5083 are trained and tested on the SS304 dataset (dataset from Chapter 4).

5.2 Specific methodology

The analysis covers 12 architectures, six CNN and six FCN. The main parameters defining the architecture variations for CNN are the convolutional kernel size, the number of kernels in each layer and stride, while for FCN the important parameters are number of layers and the number of units in each layer. Tables 5.1 and 5.2 describe the main parameters defining the architecture variations.

Table 5.1: Fully connected neural network architectures

Model reference	Number of layers	Description
7	4	downsize:[400, 487] flatten:[194800] matmul:[194800, 256]-relu matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax
8	5	downsize:[400, 487] max pool:[2, 2]/2 flatten:[48600] matmul:[48600, 256]-relu matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax
9	5	downsize:[400, 487] max pool:[3, 3]/3 flatten:[21546] matmul:[21546, 256]-relu matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax;
10	5	downsize:[400, 487] max pool:[5, 5]/5 flatten:[7760] matmul:[7760, 256]-relu matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax
11	5	downsize:[400, 487] max pool:[10, 10]/10 flatten:[1920] matmul:[1920, 256]-relu matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax
12	5	downsize:[400, 487] max pool:[20, 20]/20 flatten:[480] matmul:[480, 256]-relu matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax

Table 5.2: Convolutional neural network architectures

Model reference	Number of layers	Description
1	12	downsize:[400, 487] conv:[5, 5]x[16]/1-relu, max pool:[5, 5]/3 conv:[5, 5]x[32]/1-relu, max pool:[5, 5]/3 conv:[5, 5]x[64]/1-relu, max pool:[5, 5]/3 conv:[5, 5]x[128]/1-relu, max pool:[5, 5]/3 flatten:[384] matmul:[384, 256]-relu, matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax
2	12	downsize:[400, 487] conv:[5, 5]x[16]/1-relu, max pool:[3, 3]/2 conv:[5, 5]x[32]/1-relu, max pool:[3, 3]/2 conv:[5, 5]x[64]/1-relu, max pool:[3, 3]/2 conv:[5, 5]x[128]/1-relu, max pool:[9, 9]/9 flatten:[2560] matmul:[2560, 256]-relu, matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax
3	10	downsize:[400, 487] conv:[5, 5]x[32]/2-relu, max pool:[3, 3]/2 conv:[5, 5]x[64]/2-relu, max pool:[3, 3]/2 conv:[5, 5]x[128]/2-relu, max pool:[3, 3]/2 conv:[3, 3]x[256]/2-relu flatten:[512] matmul:[512, 256]-relu, matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax;
4	12	downsize:[400, 487] conv:[3, 3]x[16]/1-relu, max pool:[5, 5]/3 conv:[3, 3]x[32]/1-relu, max pool:[5, 5]/3 conv:[3, 3]x[64]/1-relu, max pool:[5, 5]/3 conv:[3, 3]x[128]/1-relu, max pool:[5, 5]/3 flatten:[1024] matmul:[1024, 256]-relu, matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax
5	12	downsize:[400, 487] conv:[3, 3]x[16]/1-relu, max pool:[3, 3]/2 conv:[3, 3]x[32]/1-relu, max pool:[3, 3]/2 conv:[3, 3]x[64]/1-relu, max pool:[3, 3]/2 conv:[3, 3]x[128]/1-relu, max pool:[9, 9]/9 flatten:[3840] matmul:[3840, 256]-relu, matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax
6	11	downsize:[400, 487] conv:[3, 3]x[16]/2-relu, max pool:[3, 3]/2 conv:[3, 3]x[32]/2-relu, max pool:[3, 3]/2 conv:[3, 3]x[64]/2-relu, max pool:[3, 3]/2 conv:[3, 3]x[128]/2-relu flatten:[512] matmul:[512, 256]-relu, matmul:[256, 128]-relu matmul:[128, {6, 4 or 2}]-Softmax

The input image dimension is 400×487. The first layer, ”conv:[5, 5]×[16]/1-relu”, model 1 in Table 5.2, contains 16 kernels with each kernel of dimension 5×5, stride 1, and the activation function ReLU [140]. It produces 16 feature maps of size 396×483 each. 396×483×16 = 3060288 output values with 5×5×16 (weights) + 16 (biases) = 416 parameters. The same number of output values, with a FCN, would require (396×483×16)*(400×487 + 1) = 596,147,162,688 parameters - almost impossible to process.

Following each convolution layer, there is maximum pooling [141] sampling layer. ”max pool:[5, 5]/3” translates to a kernel size 5×5 and stride 3, which samples the largest value in the receptive.

The basic blocks are the 5×5 kernels with stride 1 or 2, 3×3 kernels with stride 1 or 2, the maximum pooling layers of sizes 5×5 with stride 3 and 3×3 with stride 2. The convolution and maximum pooling operations take place at the beginning of the network, having an effect of feature reduction, minimising the input from 194800 (multiplying 400×487) pixels to few thousands. Following the reduction, all the features flatten into a single vector followed by the fully connected layers, ultimately reducing the categorisation to either 6, 4 or 2 labels corresponding to defects described earlier in Tables 3.7, 3.8 and 3.9. The last two hidden layers ”matmul:[384, 256]-relu” and ”matmul:[256, 128]-relu” are the same for all the networks to preserve similarity between different architecture and assess the power of representations built in previous convolutional layers of the CNN.

Convergence

Over the architecture variation, the learning process finds the probability distribution describing the dataset. The probability distribution is parametrised by the weights composing the kernels and fully connected layers in the architectures described earlier. The convergence algorithm used in the current study is called Adaptive Moment Estimation (Adam) [103]. The main influence on the convergence, and the speed of convergence is the learning rate. In the current study the learning rate examined are 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4} and 10^{-5}.The training parameters are described in Table 5.3

Table 5.3: Training parameters.

Learning rate	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4} and 10^{-5}
Number of epochs	5
Batch size	10
First moment estimates decay rate	0.9
Second-moment estimates decay rate	0.999

5.3 Results & Discussion

5.3.1 6-class test

The most challenging test for the current approach is the 6-class classification, with the performance over the range of the architecture and learning rate variations are described in Table 5.4. The architecture differences provide a significant degree of performance difference, particularly between the significant splits, CNN and FCN. The CNN outperforms and builds better representations from the HDR input images compare to the FCN with an accuracy performance gap of 18 percentage points (pp).

Table 5.4: Model accuracy for the learning rate and model for 6-class classification.

Model reference	Learning rates analysed					
	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	model average
1	33.23	33.23	65.94	71.75	71.22	69.64
2	33.23	33.23	55.49	62.34	62.45	60.09
3	33.23	33.23	64.30	62.42	58.26	61.66
4	33.23	33.23	65.83	61.82	56.36	61.34
5	33.23	33.23	64.37	61.38	52.02	59.26
6	33.23	44.14	54.57	40.73	48.09	47.79
7	33.23	33.23	26.05	41.68	39.00	35.57
8	33.23	33.23	40.06	42.02	40.73	40.93
9	33.23	33.23	38.75	46.92	43.14	42.94
10	33.23	35.20	39.62	42.32	44.76	42.23
11	31.54	35.53	45.22	40.70	42.96	42.96
12	33.23	44.64	40.66	47.50	46.84	45.00
average	33.09	35.44	50.07	51.80	50.48	50.78

Table 5.4, on last column (dark grey background), presents the average of the three light grey columns, 10^{-3}, 10^{-4} and 10^{-5} and it measures the model's stability across a range of values for learning rate. The average omits the first two columns, i.e. 10^{-1} and 10^{-2}, because

the networks were unsuccessful in representing the underlying dataset probability distribution for those learning rates. The 10^{-1} and 10^{-2} learning rates were too high for the architecture's internal parameters to converge to a state representative of the dataset's probability distribution. Such parameters are in great numbers since everything larger than 10^{-2} renders unusable models.

Figures 5.1 shows the same set of models as Table 5.4, displayed graphically.

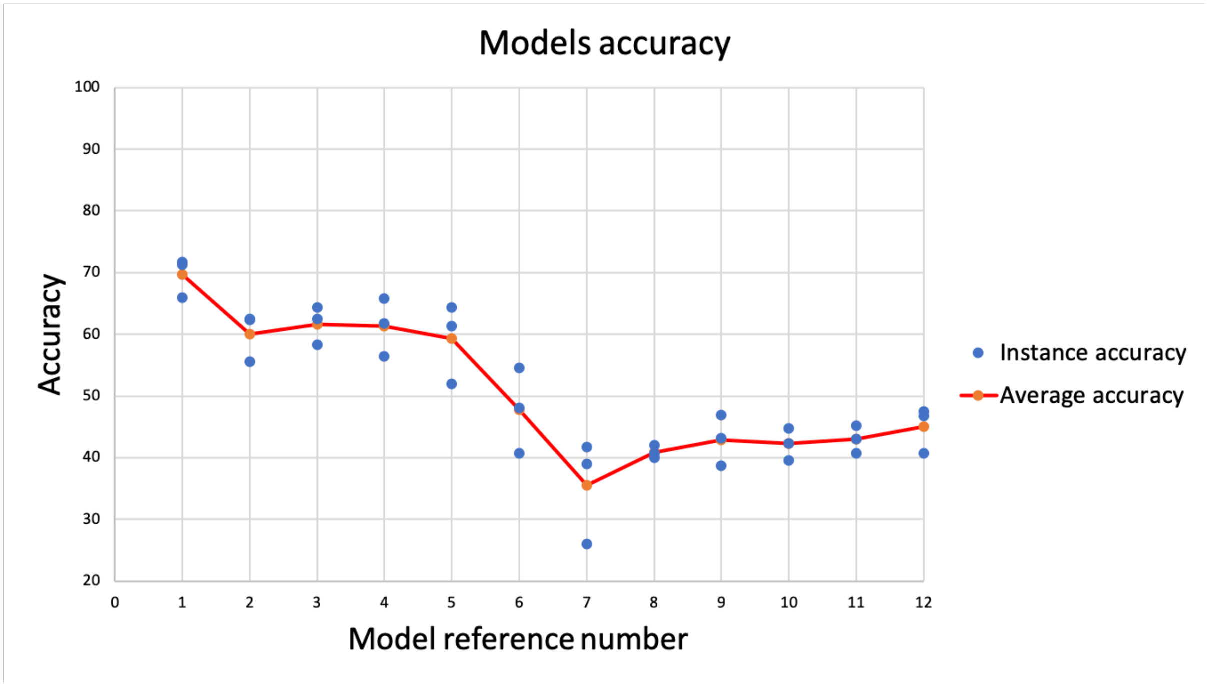


Figure 5.1: Average accuracy for models trained with the learning rates 10^{-3}, 10^{-4} and 10^{-5} for 6-class classification.

Figure 5.2 clusters the models based on the number of layers, highlighting the importance of the increase is networks depth.

The limit in this case is the processing power available, which for this study, was 4GB Nvidia GeForce GTX 980, able to accommodate a model of up to 12 layers. The pattern is also found in FCN, architectures with 5 hidden layers exhibiting an accuracy advantage over architectures with 4 hidden layers.

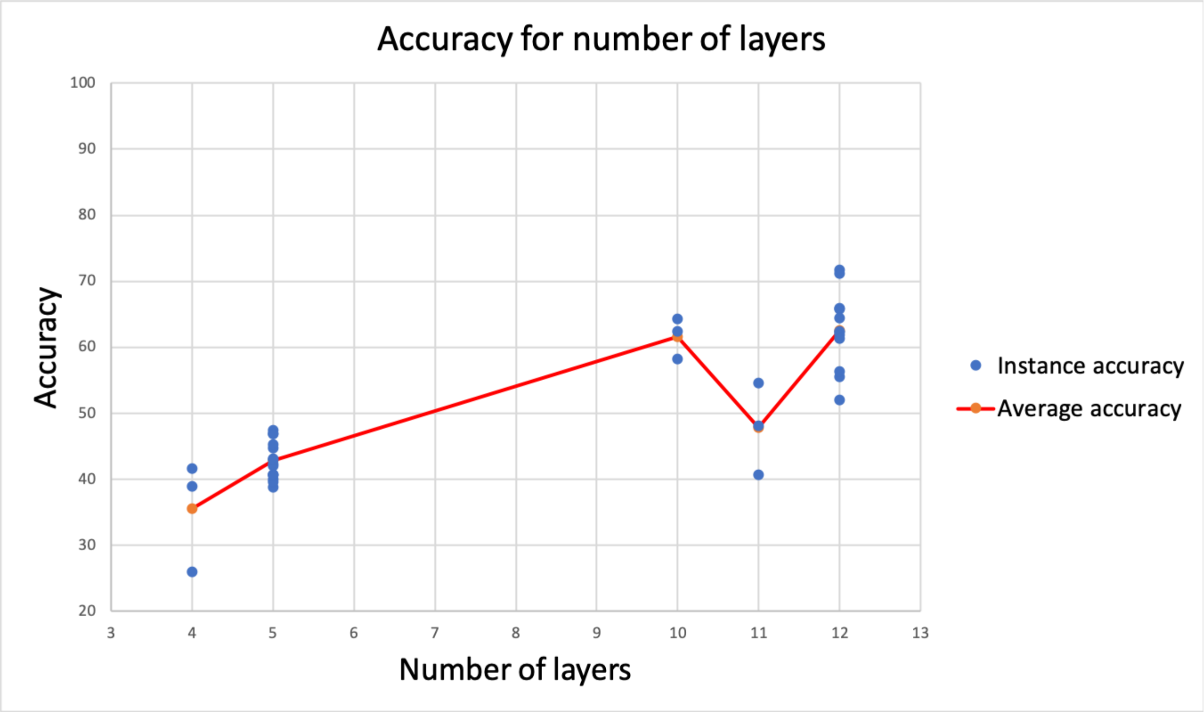


Figure 5.2: Average accuracy as a function of the number of layers for 6-class classification.

Figure 5.3 tracks the accuracy performance when training with different learning rates, 10^{-3}, 10^{-4} and 10^{-5}. The accuracy remains relatively equal within the range, leading to the conclusion that the learning rate determines if the model converges or not without providing a significant advantage towards end accuracy.

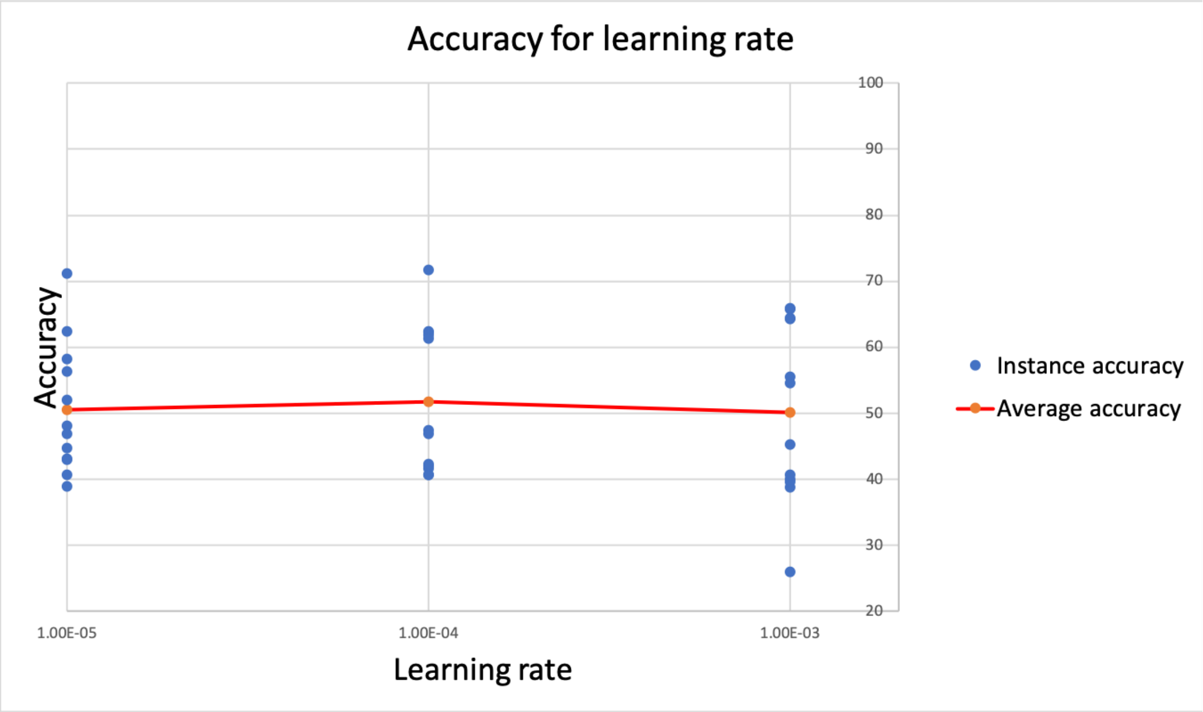


Figure 5.3: Average accuracy as a function of the learning rate for 6-class classification.

Table 5.5 and Table 5.6 is the confusion matrix and the per-class metrics for model number 1, the model exhibiting the best accuracy performance of 71.75%.

Table 5.5: Model reference 1 instance confusion matrix

	Predicted categories					
Ground truth	good weld	burn through	contamination	lack of fusion	misalignment	lack of penetration
good weld	2057	113	0	1	16	2
burn through	0	330	21	0	0	0
contamination	114	658	1041	0	265	0
lack of fusion	0	0	216	788	0	3
misalignment	43	0	8	399	279	0
lack of penetration	0	0	2	0	0	232

Table 5.6: Precision, recall and macro F-score metrics for Model reference 1 instance.

	Precision	Recall	Macro F-score
good weld	0.929	0.940	0.934
burn through	0.300	0.940	0.455
contamination	0.808	0.501	0.619
lack of fusion	0.663	0.783	0.718
misalignment	0.498	0.383	0.433
lack of penetration	0.979	0.991	0.985
average	0.696	0.756	0.691

The aim of the study is performing hyper-parameters' (e.g. learning rate) ablation analysis on values leading to a successful model. A successful model is one that achieves an accuracy exceeding 40% since the models tend to classify everything as the same class. In the current case, that class is ”good weld”.

5.3.2 4-class test

Table 5.7 shows the performance results for training the the same architecture on classifying 4 types of welds.

Table 5.7: Model accuracy for the learning rate and model for 4-class classification.

Model reference	Learning rates analysed					
	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	model average
1	25.51	25.51	68.91	89.66	86.87	81.81
2	25.51	25.51	71.57	73.29	75.13	73.33
3	27.09	25.51	74.05	74.87	74.81	74.58
4	25.51	25.51	89.66	75.32	71.83	78.93
5	25.51	27.09	74.68	81.47	74.18	76.78
6	25.51	73.54	77.03	75.19	70.18	74.13
7	25.51	27.09	69.16	63.32	62.06	64.85
8	27.09	27.09	68.78	55.58	73.10	65.82
9	25.51	56.47	71.38	57.17	61.99	63.52
10	25.51	78.17	68.46	64.91	64.09	65.82
11	27.09	73.92	57.99	70.88	74.62	67.83
12	25.51	65.67	74.75	70.37	72.53	72.55
average	25.90	44.26	72.20	71.00	71.78	71.66

The average performance difference between CNN and FCN shrinks to 9.5%. Figures 5.4 and 5.5 show the average accuracy for each class and average accuracy variation for model size increase.

The natural inclination is to set larger learning rates since the architectures are able to converge to a solution faster, requiring less training. In this regard, the FCN has a wider operating window than CNN, being able to converge even with values as high as 10^{-2}. That being said, almost all architectures examined, have peak performance in the range $10^{-3} - 10^{-5}$. Performance as a function of learning rate degrades sharply for values adjacent to 10^{-3} (namely 10^{-2}) for CNNs as well as FCN by 42pp and 13pp, respectively.

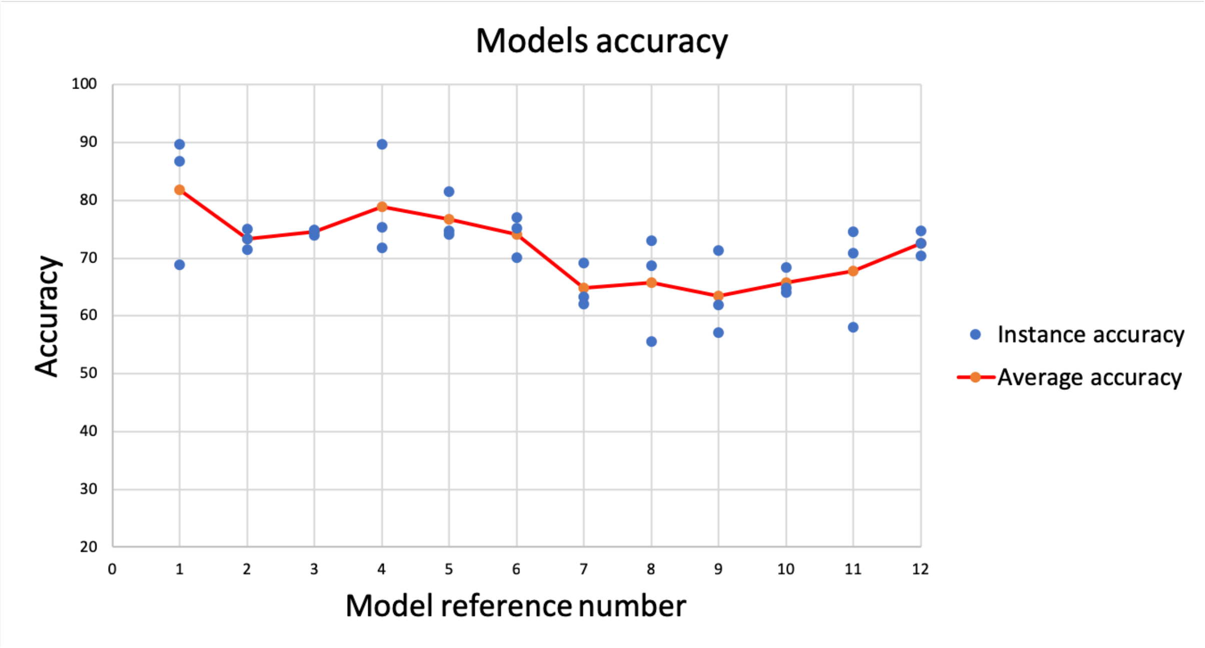


Figure 5.4: Average accuracy for models trained with the learning rates 10^{-3}, 10^{-4} and 10^{-5} for 4-class classification.

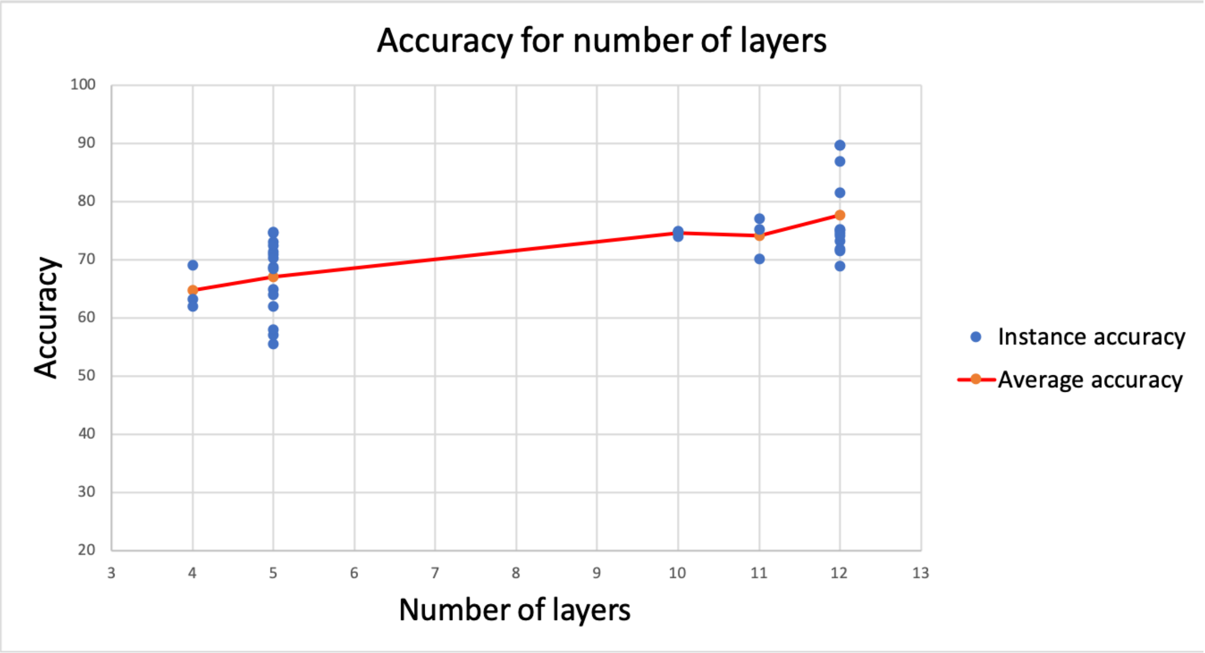


Figure 5.5: Average accuracy as a function of the number of layers for 4-class classification.

Figure 5.6, showing the accuracy performance against learning rate, highlight a similar pattern as Figure 5.5. There is not a significant difference as long as the neural network converges.

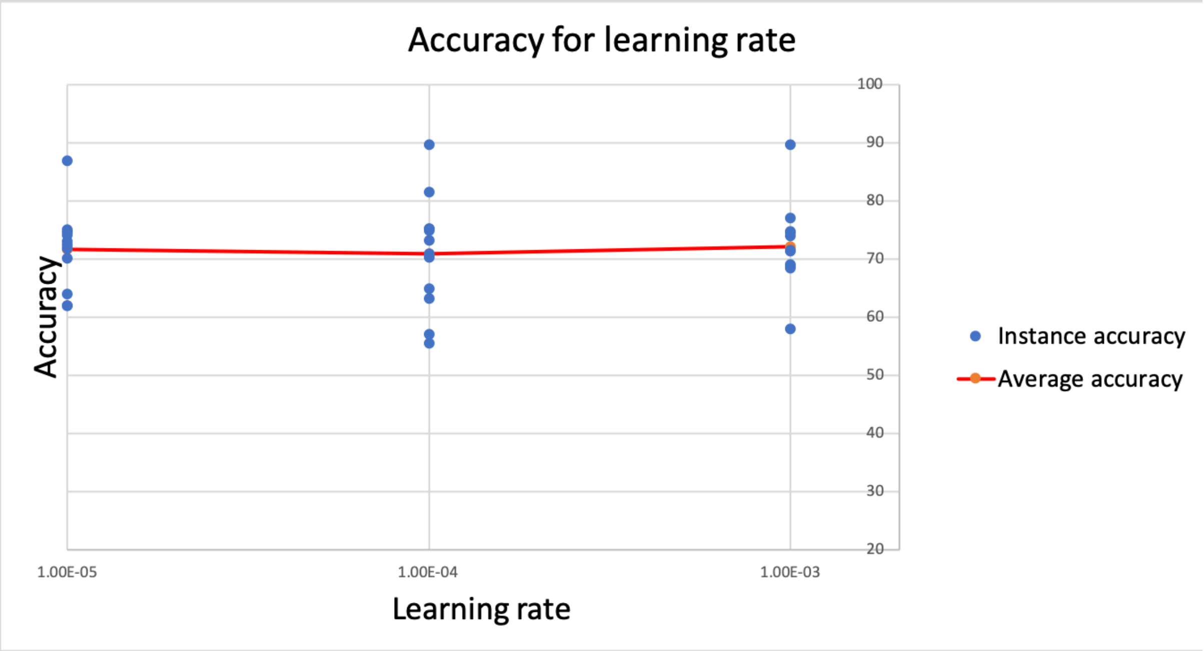


Figure 5.6: Average accuracy as a function of the learning rate for 4-class classification.

5.3.3 2-class test

In the case of 4-class and 2-class classification the accuracy is calculated in Tables 5.7 and 5.8 with the difference that the successful model is define by achieving an accuracy of at least 26% and 67%, respectively. All architectures applied on the 2-class test achieve an accuracy of 67% when the random classification probability is 50% is due to a slight imbalance in test dataset (2189 good weld samples and 4399 defective). By categorising all images as defective, the accuracy is 4399/(2189+4399) = 0.67. Therefore no representation of data was achieved.

Table 5.8: Model accuracy for the learning rate and model for 2-class classification.

Model reference	Learning rates analysed					
	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	model average
1	66.77	66.77	93.91	87.05	83.24	88.07
2	33.23	66.77	95.57	89.40	81.79	88.92
3	66.77	77.47	83.15	80.10	77.14	80.13
4	66.77	66.77	85.82	90.15	80.86	85.61
5	66.77	66.77	91.01	86.08	92.68	89.93
6	66.77	64.28	81.38	87.99	78.79	82.72
7	66.77	66.77	66.77	69.72	77.69	71.39
8	66.77	66.77	65.80	75.96	69.72	70.49
9	66.77	66.77	71.62	74.74	71.83	72.73
10	66.77	66.77	71.92	70.60	68.23	70.25
11	66.77	61.70	70.58	72.09	67.06	69.91
12	66.77	72.01	73.89	71.93	65.70	70.51
average	63.98	67.47	79.29	79.65	76.23	78.39

The results in Figure 5.4 and 5.5 show smaller performance gap between CNN and FCN with a difference of 7.5pp in favour of CNN. Analysing samples labelled incorrectly it is observed that most images of ”contamination” the samples are misclassified, in almost all cases, indicating the defect is not represented sufficiently well in training dataset.

The 2-class performance mirrors the pattern observed in the 4-class test except for overall accuracy numbers. Since the classification involves two classes, the entire graphs shift up, reflecting the simpler categorisation task.

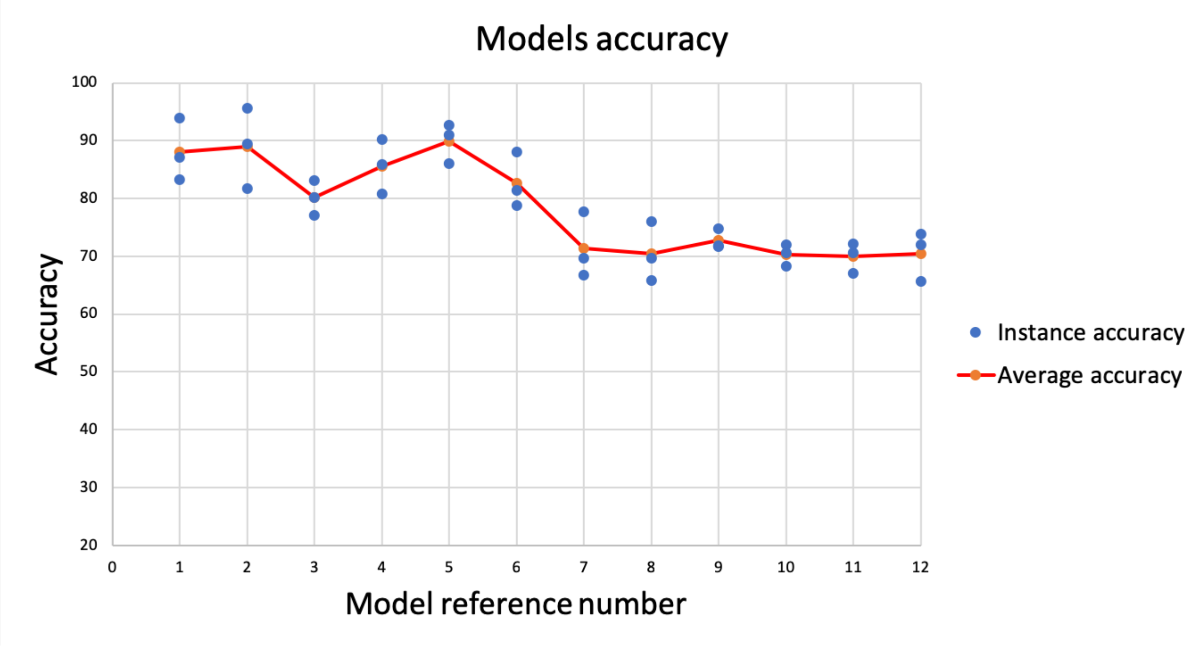


Figure 5.7: Average accuracy for models trained with the learning rates 10^{-3}, 10^{-4} and 10^{-5} for 2-class classification.

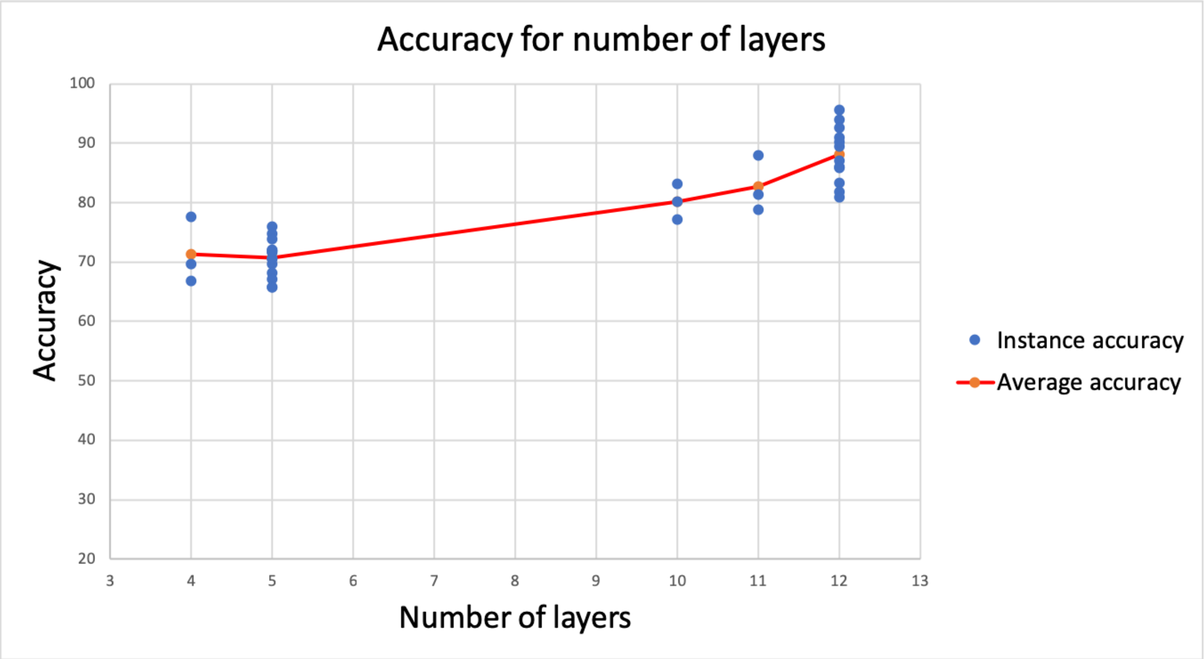


Figure 5.8: Average accuracy as a function of the number of layers for 2-class classification.

Figure 5.7 confirms the conclusion for the equivalent graphs for 6-class test and 4-class test.

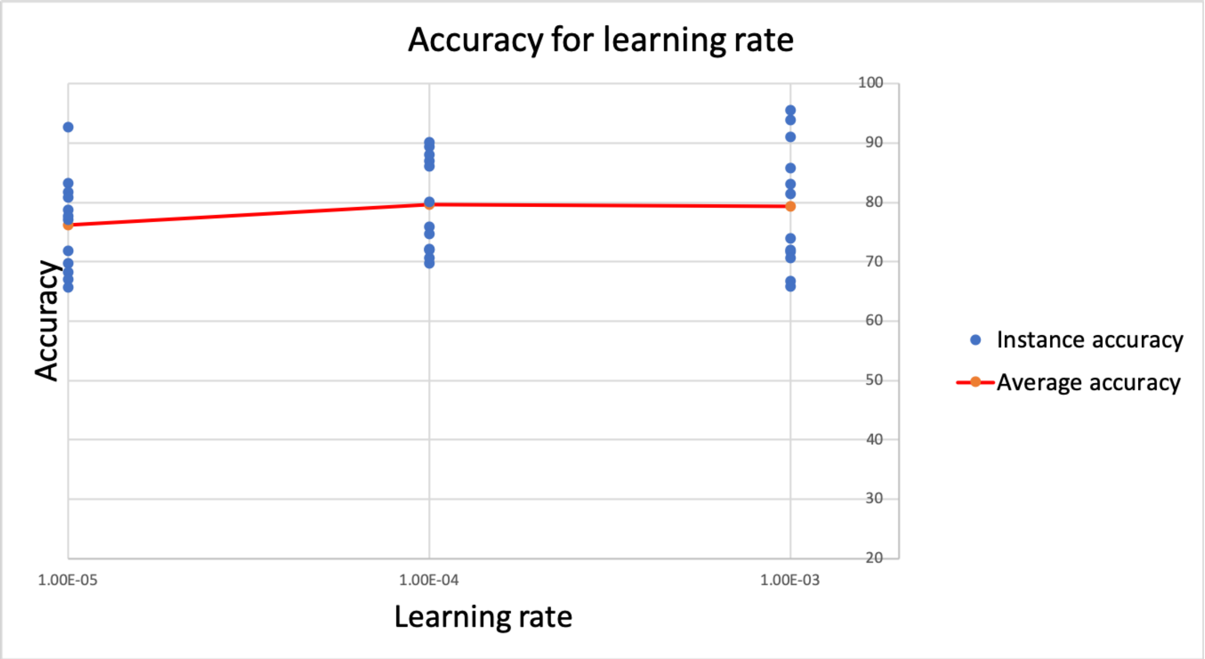


Figure 5.9: Average accuracy as a function of the learning rate for 2-class classification.

5.3.4 Training stability

The training duration is 5 epochs for all the neural network architectures. Figure 5.10 shows the loss evolution during training stage, significant of the convergence towards a solution. In this case the loss graph represents an instance of model reference 4, but the same trend could be observed for all the models.

Figure 5.11 shows the test accuracy at different points during training, notably at every epoch's end.

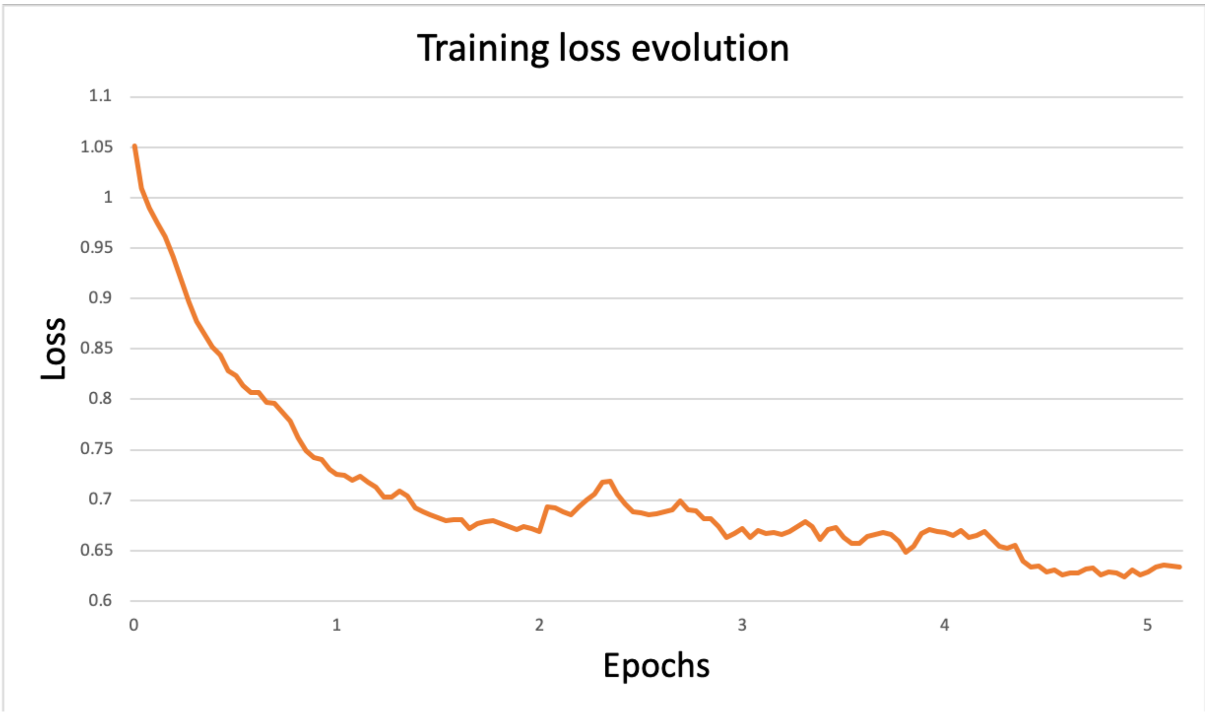


Figure 5.10: Training loss evolution.

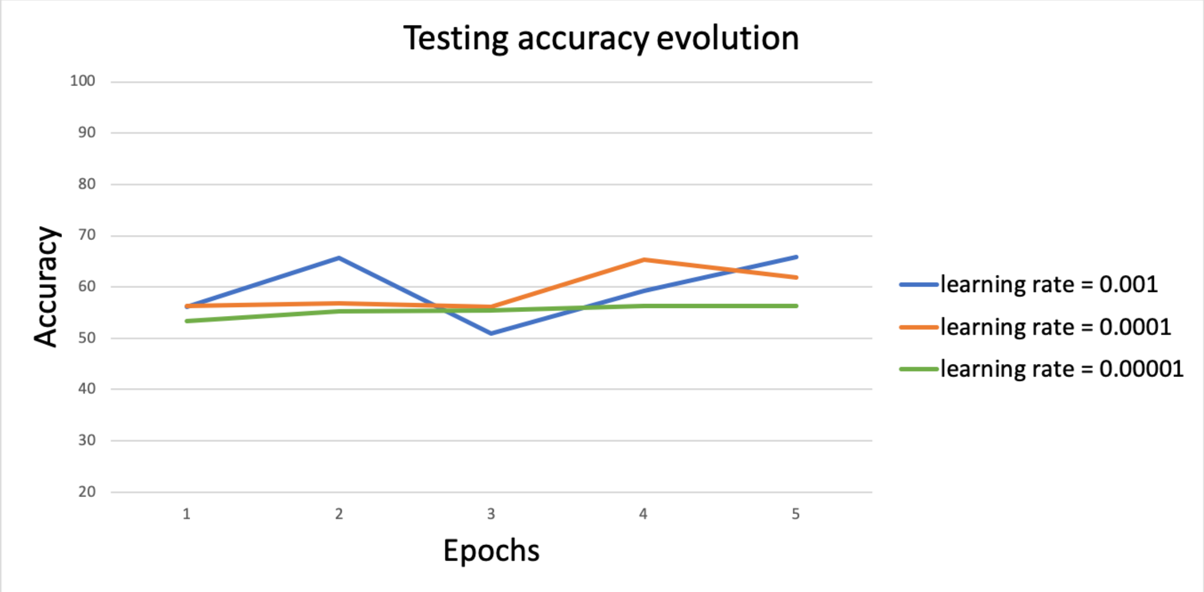


Figure 5.11: Test accuracy evolution.

Figures 5.10 and 5.11 highlight the models' rapid converge during the first epoch, while during the next 4 epochs, the models either maintain the same accuracy or at best improves it marginally.

5.3.5 Subsampling influence

This study analysis the impact of the resolution reduction on the final accuracy performance. The result are presented in Figures 5.12, 5.13, 5.14, for 6-class, 4-class and 2-class, respectively. Models trained using the images subsampled to 25×30 pixels then upsampled to 400×487 are compared against models trained with the images subsampled to 400×487 pixels. The fidelity reduction impacts severally the CNN architectures, particularly when the problem difficulty increases and the distinction between classes became harder, as in the 6-class test.

The FCNs show small decrease in accuracy performance, therefore concluding the architectures are agnostic to inaccurate pixel values and less sensitive to the gradient between pixels values.

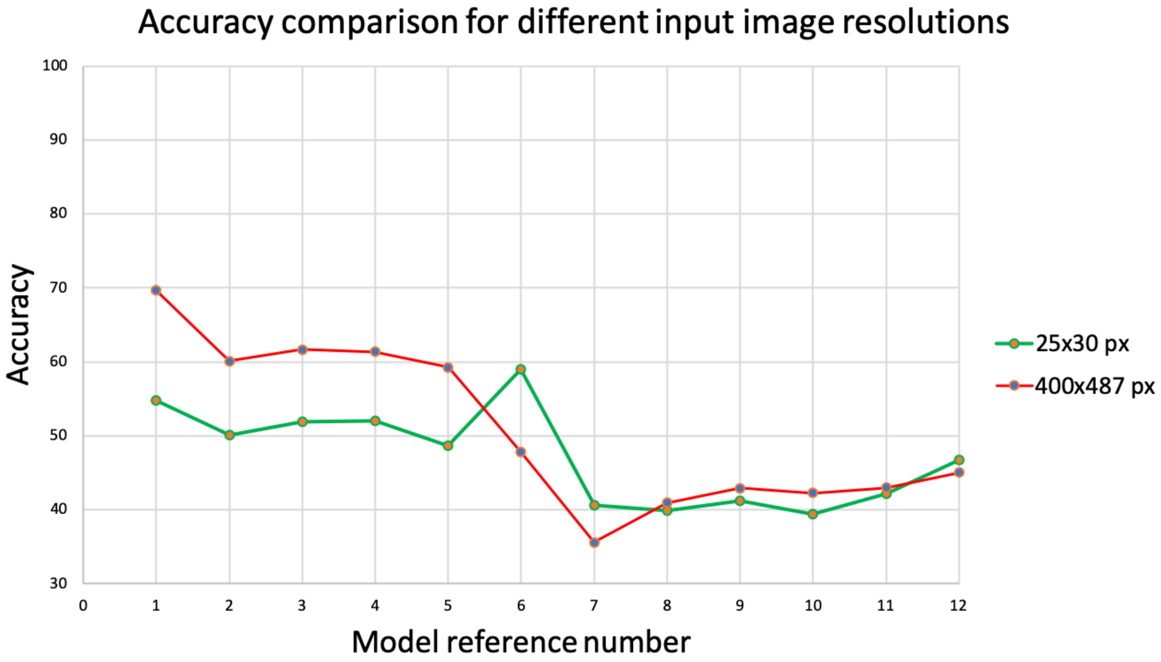


Figure 5.12: Subsampling accuracy impact for 6-class defect.

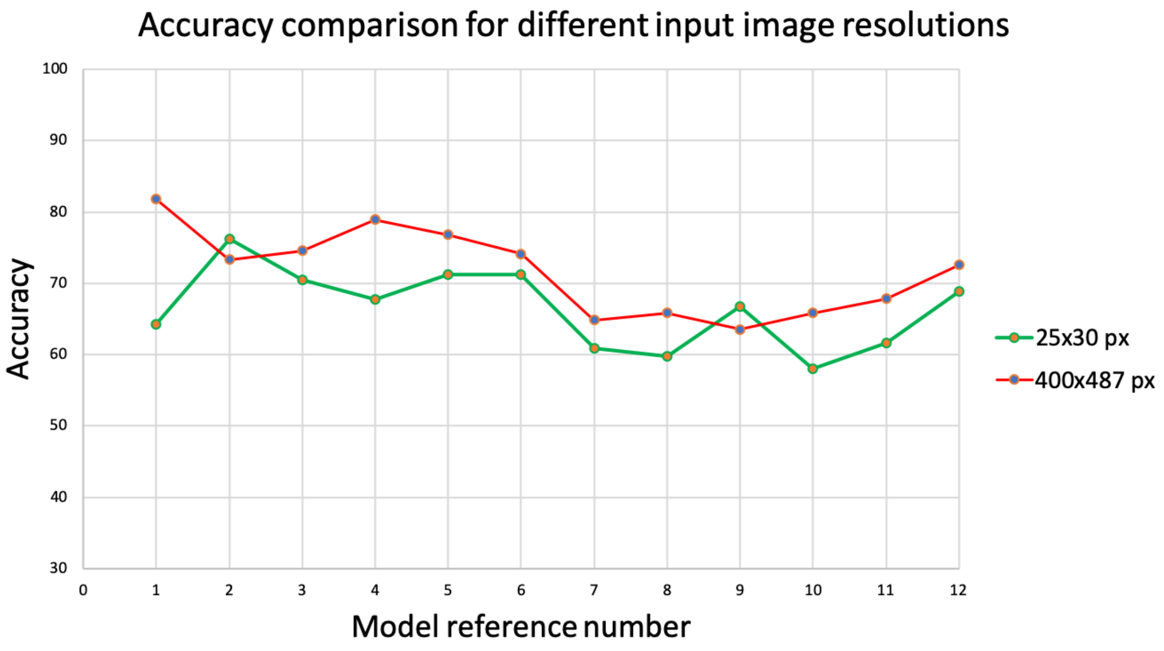


Figure 5.13: Subsampling accuracy impact for 4-class defect.

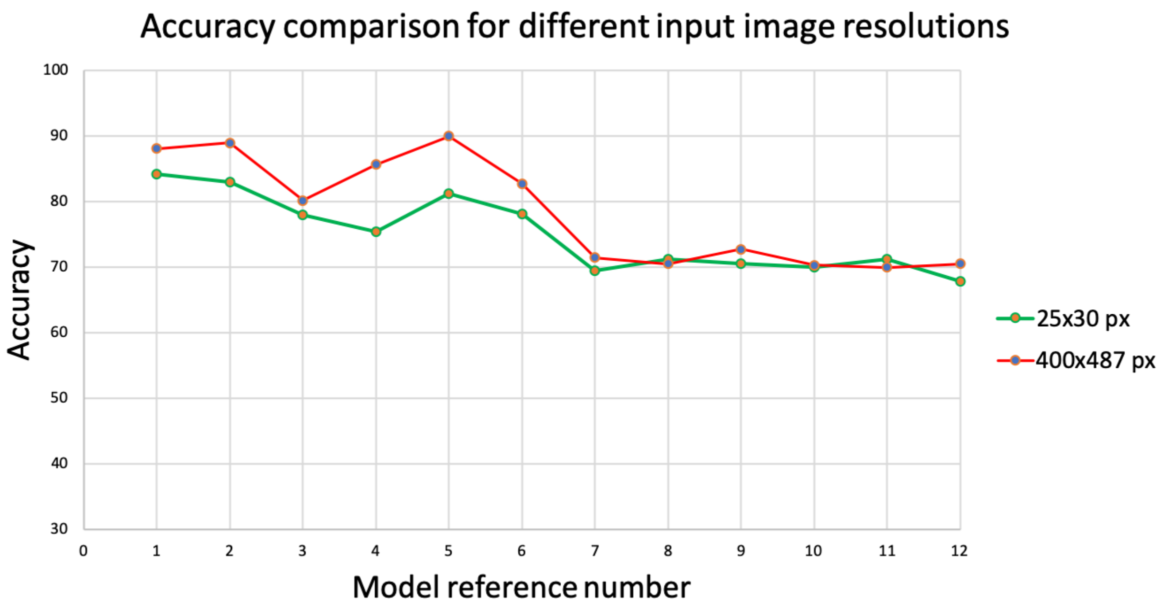


Figure 5.14: Subsampling accuracy impact for 2-class defect.

5.3.6 Model improvement

Figure 5.15 shows the difference in accuracy performance achieved for categorising six classes (good weld + 5 defects) for SS304 (Chapter 4 dataset). The accuracy difference relates to the hyper-parameter optimisation, effectively finding models that perform better on the welding images. The Conv6 model in Chapter 4 has 5190 parameters while the model number 1 in this chapter has 434438. The main problem with the big model applied to ”simpler” problems is the overfitting. In this case, the short training (generally called early stopping) duration ensures the model does not overfit and still achieves high accuracy.

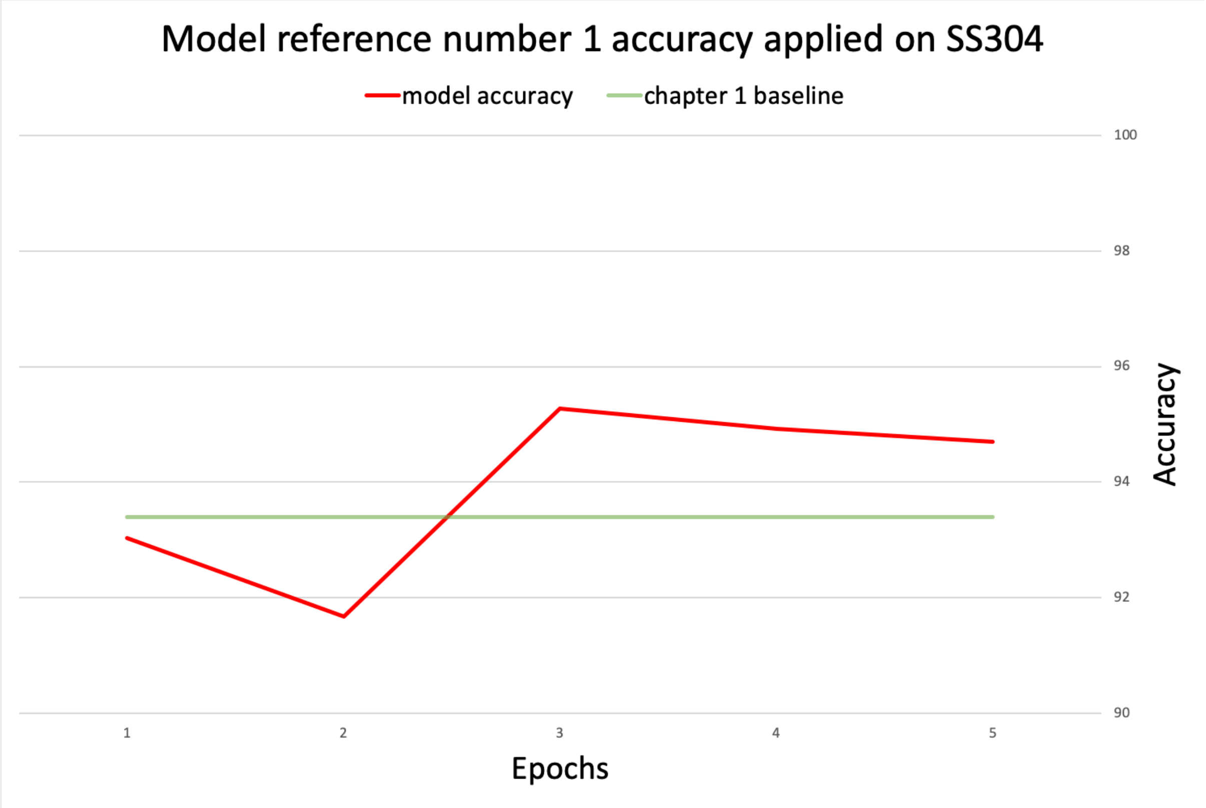


Figure 5.15: The comparison between model number 1 and the Conv6 model defined in Table 4.2. The Conv6 accuracy is the final accuracy extrapolated over the entire epochs range for reference.

The model number 1 builds a better representation of the underlying probability distribution, aspect deduced from studying per-class metrics, as recall, precision and macro-F-score.

Table 5.9: model reference number 1 confusion matrix for SS304.

Ground truth	Predicted categories					
	good weld	burn through	conta-mination	lack of fusion	lack of shielding gas	high travel speed
good weld	769	0	0	0	0	0
burn through	0	730	0	1	0	0
contamination	0	0	576	0	0	0
lack of fusion	0	64	0	586	94	0
lack of shielding gas	0	0	7	2	93	0
high travel speed	0	0	0	0	0	249

Table 5.10: Precision, recall and macro F-score metrics for model reference number 1 applied on SS304

Class	Precision	Recall	Macro F-score
good weld	1.000	1.000	1.000
burn through	0.919	0.999	0.957
contamination	0.988	1.000	0.994
lack of fusion	0.995	0.788	0.879
lack of shielding gas	0.497	0.912	0.644
high travel speed	1.000	1.000	1.000
average	0.900	0.950	0.912
average Conv6 for comparison as of Table 4.9	0.77	0.79	0.78

5.3.7 Radiographs

This study utilises for processing solely the images produced by the imaging of the welding process and categorised by an experienced welder. The radiographs analysis show the existence of pores along the weld line in all the samples inspected using radiography without any effects produced by the pores at the surface of the weld during welding as in Figures 5.16, 5.17 and 5.18. The observation points to the conclusion that the small pores become trapped into the material, having insufficient time to surface, stressing the need for a procedure that ensures adequate resilience towards certain types of defects, as pores formation. At the same time, it uncovers one of the system limitations namely the inability to detect some types of defects that would impact the component usability.

The system under investigation is unable to adequately record, therefore unable to detect small pores forming and trapped into the material. The pores are only one type of defects the system is unable to detect. Generally, any defect that produces no visible difference, as trapped pores, slight insufficient penetration or slight misalignment is very hard to identify.

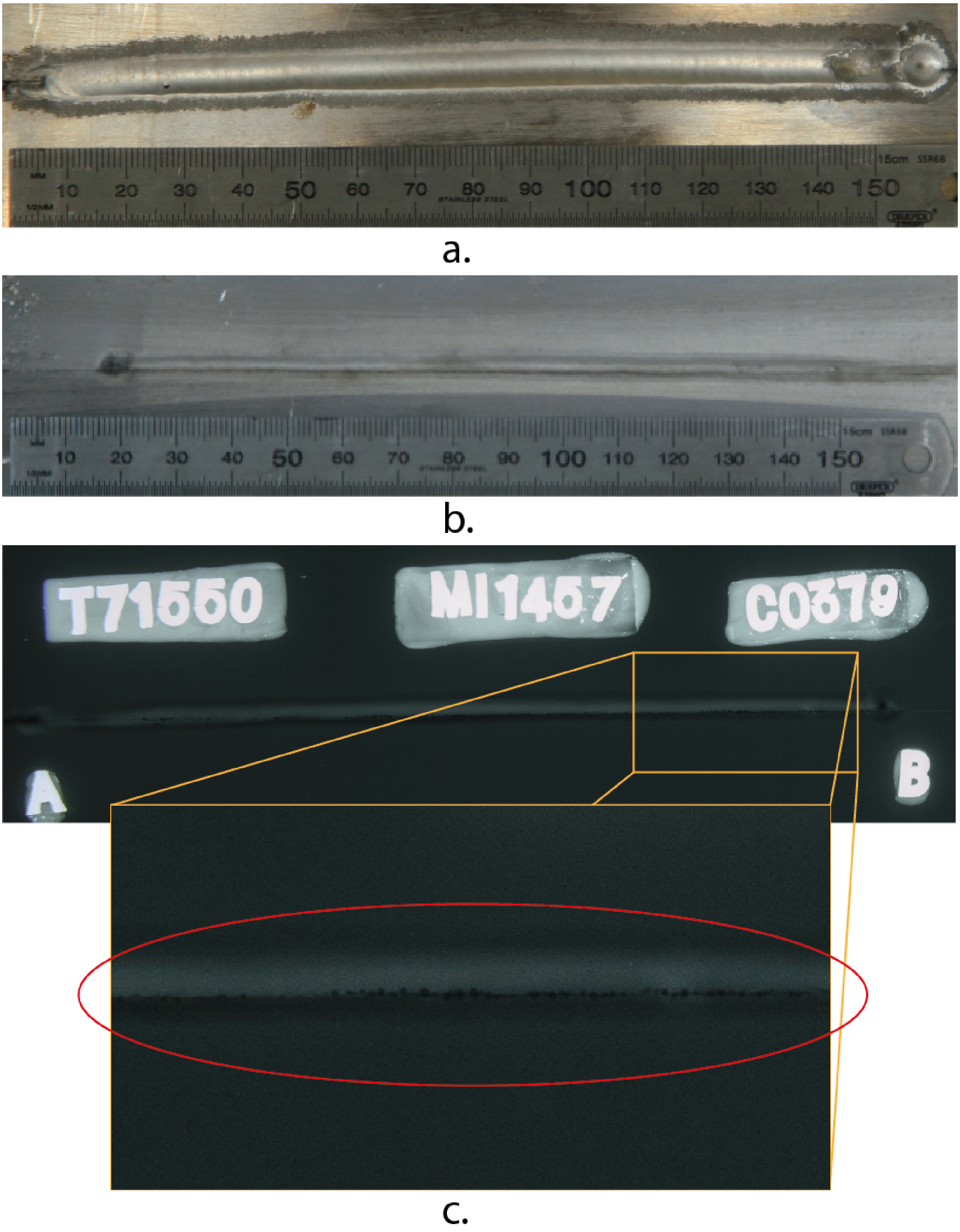


Figure 5.16: a) The front of the workpiece after welding, b) the back of the workpiece after welding, c) the workpiece radiography. The recorded weld generating the workpiece is categorised as ”good weld”.

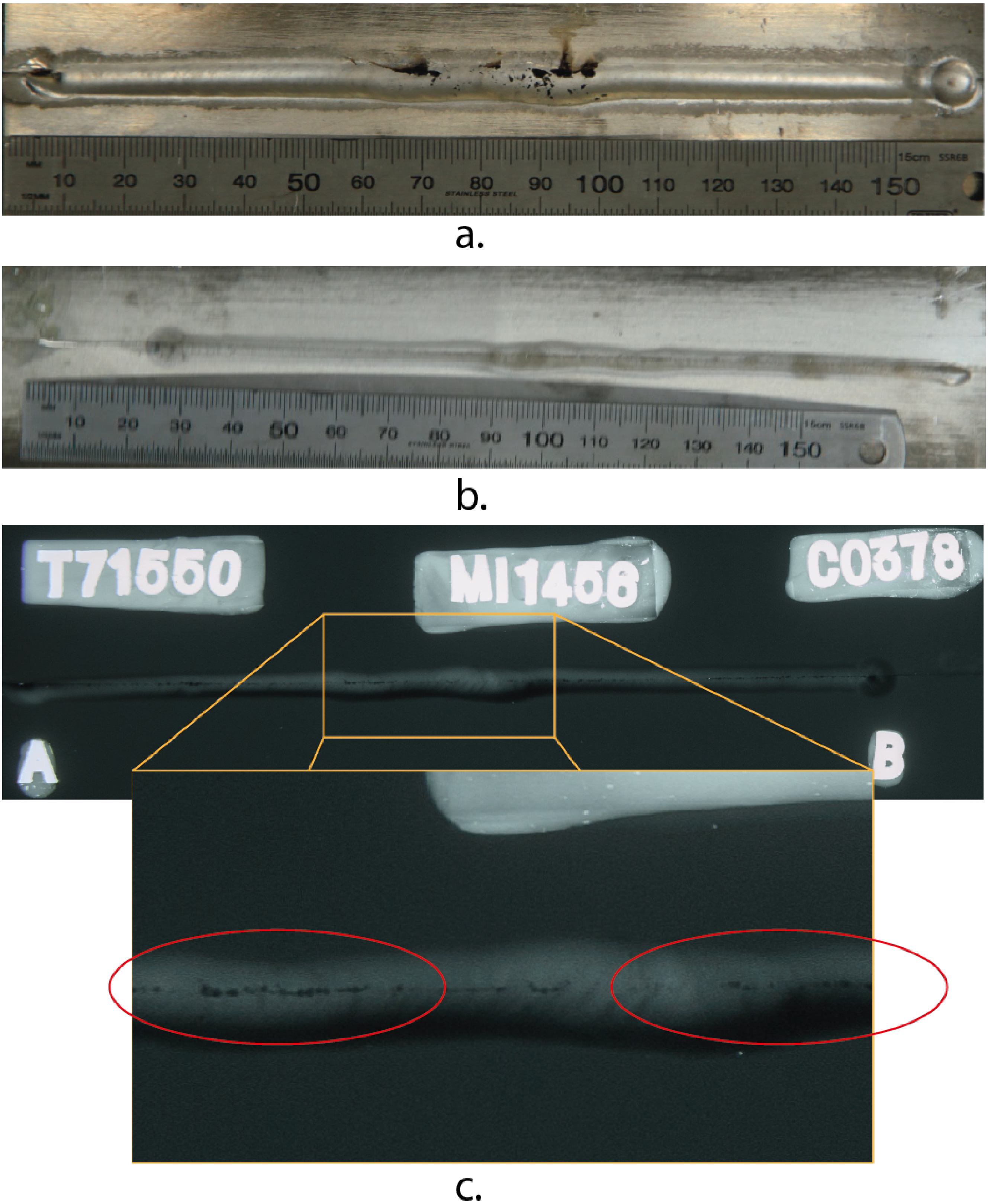


Figure 5.17: a) The front of the workpiece after welding, b) the back of the workpiece after welding, c) the workpiece radiography. The recorded weld generating the workpiece is categorised as ”contamination”.

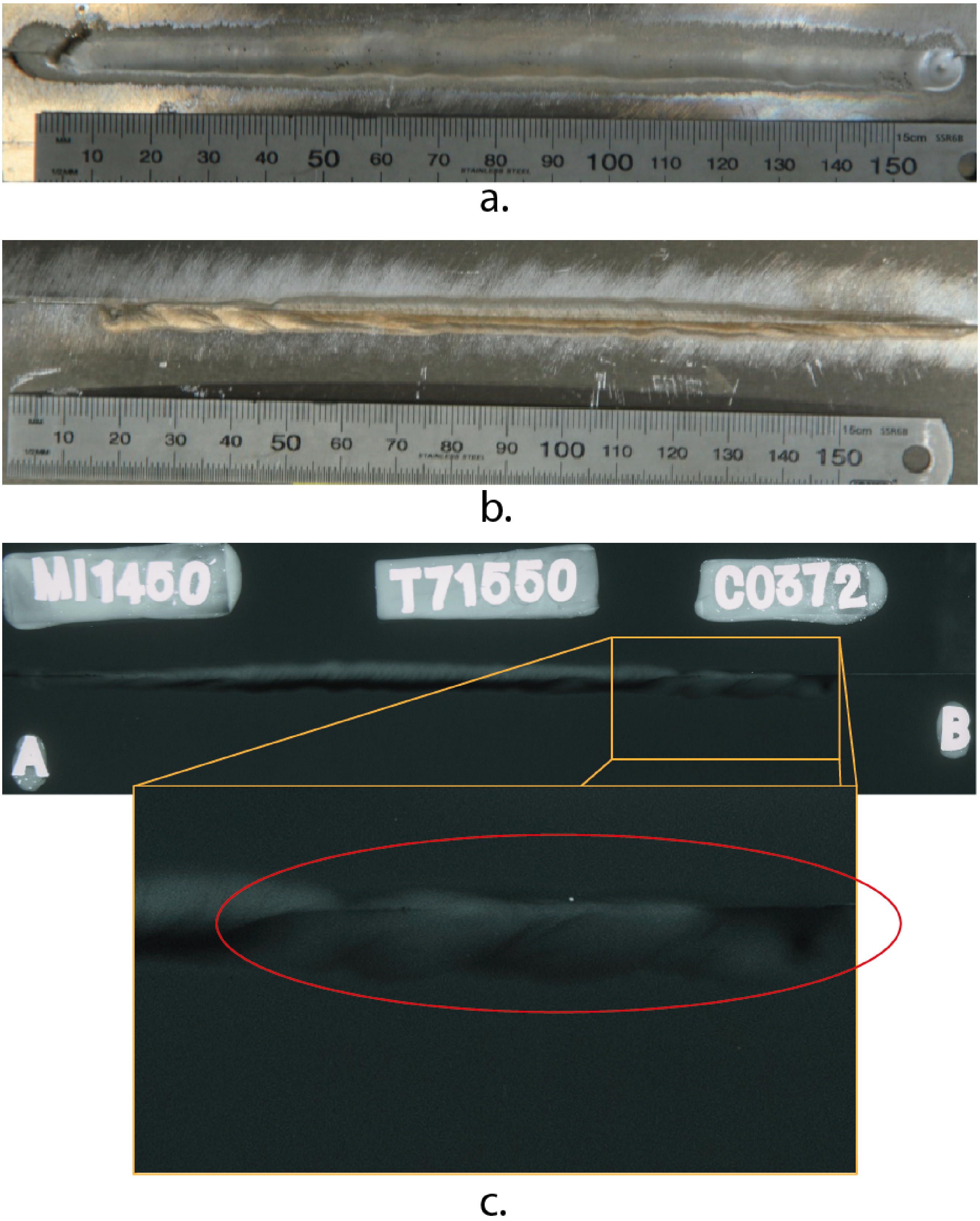


Figure 5.18: a) The front of the workpiece after welding, b) the back of the workpiece after welding, c) the workpiece radiography. The recorded weld generating the workpiece is categorised as ”good weld”.

5.4 Conclusion

The analysis involved the construction of models based on CNN and FCN, varying internal architecture and hyper-parameters influencing the convergence of internal parameters for representing dataset's probability distribution. The models were trained using 6-class, 4-class and 2-class tests with top accuracies of 71%, 89% and 95%, respectively. Furthermore, this study performed neural networks' robustness examination over a set of problems (6-class, 4-class and 2-class), highlighting the parameters influencing the model performance, concluding the architecture is the most important aspect given the learning rate is adequately chosen. The study also cover, the critical analysis of the accuracy performace impact linked to the input images fidelity reduction.

The system required the generation of a new TIG welding dataset representing good welds as well as different types of common defects. This study contributes with 33254 images covering five welding defects.

Chapter 6

Learned knowledge transfer

6.1 Introduction

The previous chapters explored the FCN and CNN of which the network designer defined typologies. The number of layers, the type of the layers (kernels), the kernel hyper-parameters (number of filters, filter size and filter stride) and the optimiser choice, were all selected before the training stage. The optimiser repeatedly adjusts each layer's weights during the training stage to find the optimal set of values that would map the input (images) to output (labels). The previous chapter generated several network typologies for training and evaluation representing a small number of samples with regards to the search space.

6.2 Specific methodology

The repetitive nature of similar layer types in the networks designed manually inspired the search space definition and space exploration procedure in the current chapter. The aim is to find a *cell*, exemplified in Figure 6.1(d), that composed in a sequential fashion, as in Figure 2.10(left), forms a neural network. In this way, the search space shrinks from finding the optimal network to finding the optimal cell that placed in a network is capable of mapping the input to the output.

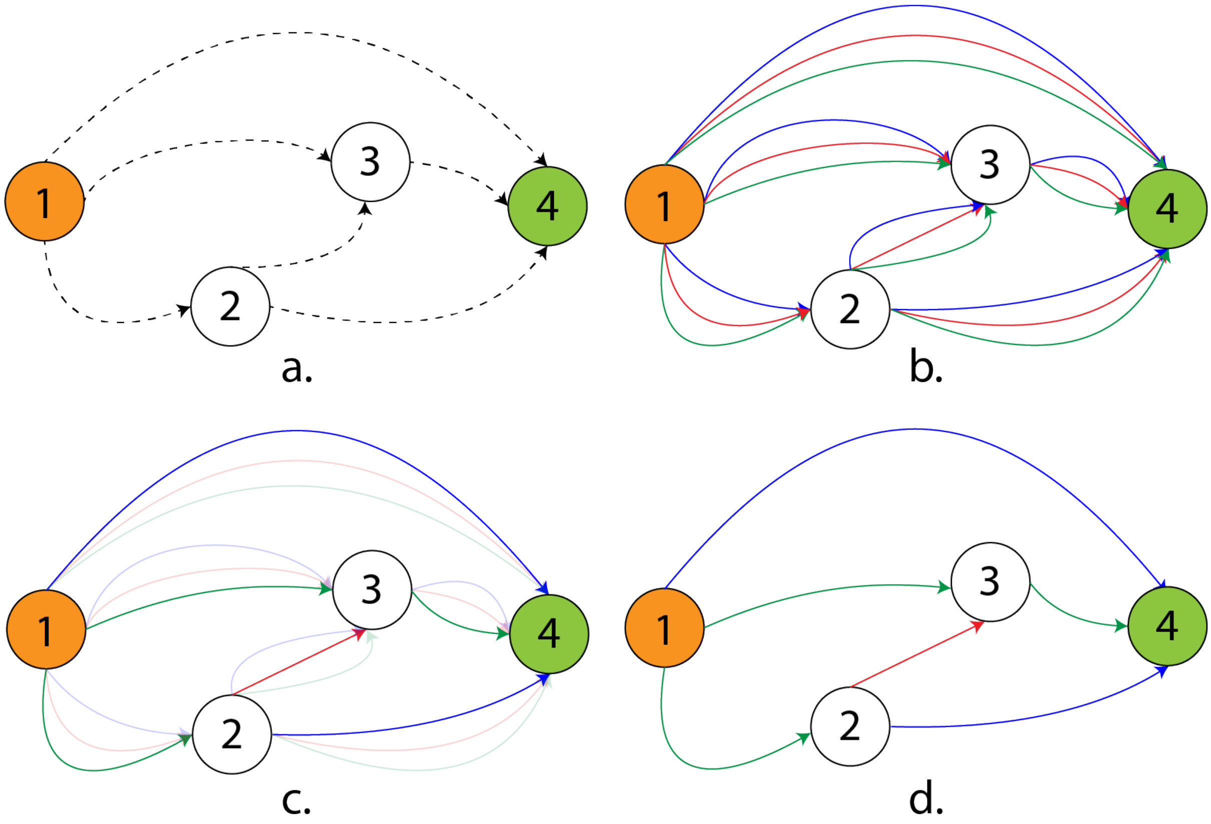


Figure 6.1: Cell example composed of four nodes. The node labelled ”1” is the input, while the node labelled ”4” is the the output. (a) The goal is to link the nodes with operations (operations are on edges) in the optimal fashion. (b) All possible operations are placed on the edges. (c) The edges are governed by weights significant of connection strength. (d) Only the strongest connections are retained, forming a network instance.

The previous chapters limited the training to finding the network layer's weights, while in this chapter there are two sets of weights: cell weights (kernel or layer weights, the search from previous chapters) and the cell edges (linking) weights. An instance of a cell is the sub-graph with only the largest edge weights retained as in Figure 6.1(d). The aim is to link the nodes in Figure 6.1(a) somehow, in the most optimal way, in order for the networks (composed of many sequentially linked cells) to exhibit the best performance. Figure 6.1(b) shows placing three types of operations on each edge (an example of operations could be, but not limited to, convolution, sampling and no operation). Figure 6.1(c) highlights the strongest connections as a result of optimisation process while Figure 6.1(d) represents an instance of a cell typology that would be placed in a network, with the new network trained and tested on a separate subset of the data.

The search space definition includes two aspects: the definition of the operations, i.e. the link types between the nodes, and the cell linking pattern, i.e. the number of inputs and outputs for every node.

In this study the set of operations placed on the edges are:

- 3×3 separable convolution
- 5×5 separable convolution
- 7×7 separable convolution
- 3×3 dilated separable convolution
- 5×5 dilated separable convolution
- 7×1 then 1×7 convolutions
- 3×3 average pooling
- 3×3 maximum pooling
- identity connection (no processing)
- zero (no connection)

The linking pattern is as follows:

- each cell has two inputs and one output
- each node in a cell receives two inputs from other nodes from the same cell
- the output of a cell is the concatenation of the last two nodes from the same cell

Figure 6.2 shows the general typology of the cell. Each dotted arrow is one of the ten operations enumerated above. Therefore, the NAS aims for selecting the appropriate operation for each connection, given the linking pattern.

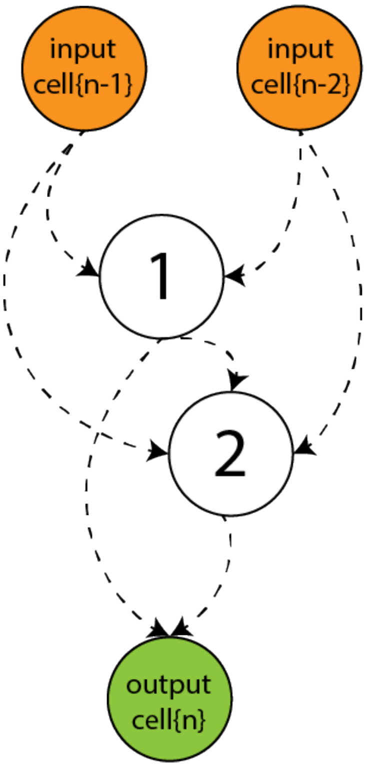


Figure 6.2: General cell pattern. The inputs to the cell are the two outputs generated by the two previous cells. The cell nodes are connected acyclically receiving inputs from any two previous nodes, generating one output. The cell output is the concatenation of node labelled '1' with the node labelled '2'.

NAS explores two types of cell, the *normal* cell and the *reduction* cell. Both cell types share the same pattern but the operations placed on each connection between nodes differ. The reduction cell uses stride 2 for convolutions, therefore shrinking the subsequent cells input dimensionality.

In the previous chapters, the process of finding a new model involved training the model on the training dataset and evaluating it on the test dataset. In the first chapter, the validation checks the overfitting behaviour, but primarily there were two stages of training and testing. In the case of NAS, the training has two part: finding a performant cells typology (using a reduced dataset), then after arriving at the optimal cells typology, construct a new network, using the cells discovered, and train on the entire dataset for accuracy (find the optimal network weights, as in the previous chapters).

The first stage involves building a network out of a cell, with each cell's internal nodes using all types of operations at once. Each operation has a weight, i.e. a scalar, as follows [89]:

$$node_{(k)} = \sum_{i<k} op_{(i,k)} x_{(i)}$$

$$\overline{op}_{(i,k)}(x) = \sum_{op \in \mathcal{O}} \frac{e^{\tau_{op,(i,k)}}}{\sum_{op\prime \in \mathcal{O}} e^{\tau_{op\prime,(i,k)}}} op(x)$$

where op($\cdot$) is an operation applied on the input x, $\tau_{i,k}$ is a vector of dimension $\mathcal{O}$ and $\overline{op}_{(i,k)}$ is the mixing operation representing the architecture typology in a continuous domain. The first stage aims to jointly learn (find) the set of continuous variable $\tau = \tau_{(i,k)}$ and network weights θ for all the mixed operations using gradient descent. A discrete instance of the cell is obtained by retaining the two highest values from τ, equivalent to retaining the strongest connections in the cell.

The first training stage is finding the optimal typology using training and validation datasets. The network tries to find the optimal τ^* by minimising the validation loss $\mathcal{L}_{val}(\theta^*, \tau*)$, while the architecture weights (θ^*) minimises the training loss $\mathcal{L}_{train}(\theta, \tau^*)$. More elegantly the problem is formulated as [89]:

$$\min_{\tau} \mathcal{L}_{val}(\theta^*(\tau), \tau)$$

$$s.t. \qquad \theta^*(\tau) = argmin_{\theta} \mathcal{L}_{train}(\theta, \tau)$$

It is a two-step process described by the following algorithm:

Algorithm 2 Differentiable architecture search [89]

```
1: Given a mixed operation $\overline{op}_{(i,k)}$ for each edge $(i, k)$
2: while not optimal do
3:     Apply $\nabla_{\theta} \mathcal{L}_{train}(\theta, \tau)$ on the weights $\theta$
4:     Apply $\nabla_{\tau} \mathcal{L}_{val}(\theta - \xi \nabla_{\theta} \mathcal{L}_{train}(\theta, \tau), \tau)$ on the weights $\tau$
5: end while
6: Retain only the edge with the highest weights $\overline{op}_{(i,k)} = argmax_{o \in \mathcal{O}} \tau_{o,(i,k)}$
```

At every step t, using the typology τ_{t-1}, the weights θ_t are obtained by minimising $\mathcal{L}_{train}(\theta_{t-1}, \tau_{t-1})$. Then keeping the weights θ_t fixed, the typology weights update by minimising validation loss with respect to each weight as follows [89]:

$$\mathcal{L}_{val}(\theta_t - \xi \nabla_\theta \mathcal{L}_{train}(\theta_t, \tau_{t-1}), \tau_{t-1})$$

The first optimisation stage produces two cells, normal and reduction. The found cells typology represents the basis of a new network, with a fixed typology and the weights θ reset. The new network follows the same training procedure as the previous chapters. It represents the second stage of training. The network depth (number of layers) differ from the first stage depending on processing requirements. A network is a sequence of cells. Therefore it can scale up and down.

In the current study, the cell has two inputs, two intermediate nodes and one output, as in Figure 6.2. The output is the concatenation of the two intermediate nodes' output. Every cell receives the output of the two previous cells. The reduction cell is in the middle of the network, location varying depending on the number of layers.

During the **first stage of training** the hyper-parameters are as follows:

- the image size is 100×100 pixels, down from 400×487 in previous chapter
- the number of channels for the first cell is 24
- the number of layers for the network is 8 (therefore 8 cells)
- the training duration is 10 epochs
- the optimiser for the weights θ is Momentum Standard Gradient Descent (SGD), with the learning rate starting at 10^{-3} and decreasing up to 10^{-4} during training, the momentum is 0.9 and the weight decay is 10^{-2}
- the optimiser for typology weights τ is Adaptive Moment Estimation (Adam), with the learning rate 10^{-3}, the first and second moment estimates are $\beta = (0.5, 0.999)$ and the weight decay 10^{-3}

The **second stage of training** used the following hyper-parameters:

- the image size is 200×200 pixels for the Al 5083 images and 320×175px for the SS304 images
- the number of channels for the first cell is 16
- the number of layers for the network is either 2, 6 or 10 (therefore 2, 6 or 10 sequentially

linked cells)

- the training duration is 100 epochs
- the optimiser for the weights θ is Momentum Standard Gradient Descent (SGD), with the learning rate starting at 10^{-2} and decreasing up to 2.5×10^{-6} during training, the momentum is 0.9 and the weight decay is 10^{-3}

The 4GB memory of the Nvidia GeForce GTX 980 GPU represents the limitation for the number of channels in the first cell and the network depth, i.e. the number of layers. The implementation uses Pytorch [137] library.

The data split for training, validation and testing subsets for Al5083 is described in Table 6.1:

Table 6.1: Dataset split between training, validation and test for 6-class Al 5083

Category	Number of samples		
	Train	Validation	Test
good weld	4025	4733	2189
burn through	1728	55	351
contamination	3892	2433	2078
lack of fusion	1262	2766	1007
misalignment	1872	1081	729
lack of penetration	1780	1039	234
Total	14559	12107	6588

The Al 5083 dataset has a validation subset, and SS304, Table 6.2, does not have one in this chapter, because the optimisation for establishing the optimal linking pattern and operations type uses only Al 5083 dataset. The cell found is then transferred to a network that was never trained on, the SS304 dataset, to study how the typologies could transfer from one dataset to another. Therefore, SS304 dataset is useful only for the second stage of the training and for the testing. The Al 5083 dataset is the same used in the Chapter 5, with the exception for the first stage of training, where the training subset splits into two ”equal” part: training and validation. The new training subset optimises the layers weights, while the validation subset, optimises the linking pattern and operations type.

Table 6.2: Dataset split between training and test for 6-categories SS304 material

Category	Number of samples	
	Train	Test
good weld	1910	540
burn through	977	731
contamination	1613	960
lack of fusion	5036	1490
lack of shielding gas	196	102
high travel speed	630	249
Total	10362	5072

6.3 Results & Discussion

6.3.1 Cells found

Two candidates have been generated, shown graphically in Figure 6.3 after performing the first stage of training for the identification of an optimal cell typology. Figure 6.3(a) is the normal cell, preserving the image dimension (or input dimension more generally) while Figure 6.3(b) is the reduction cell which reduces the input dimension.

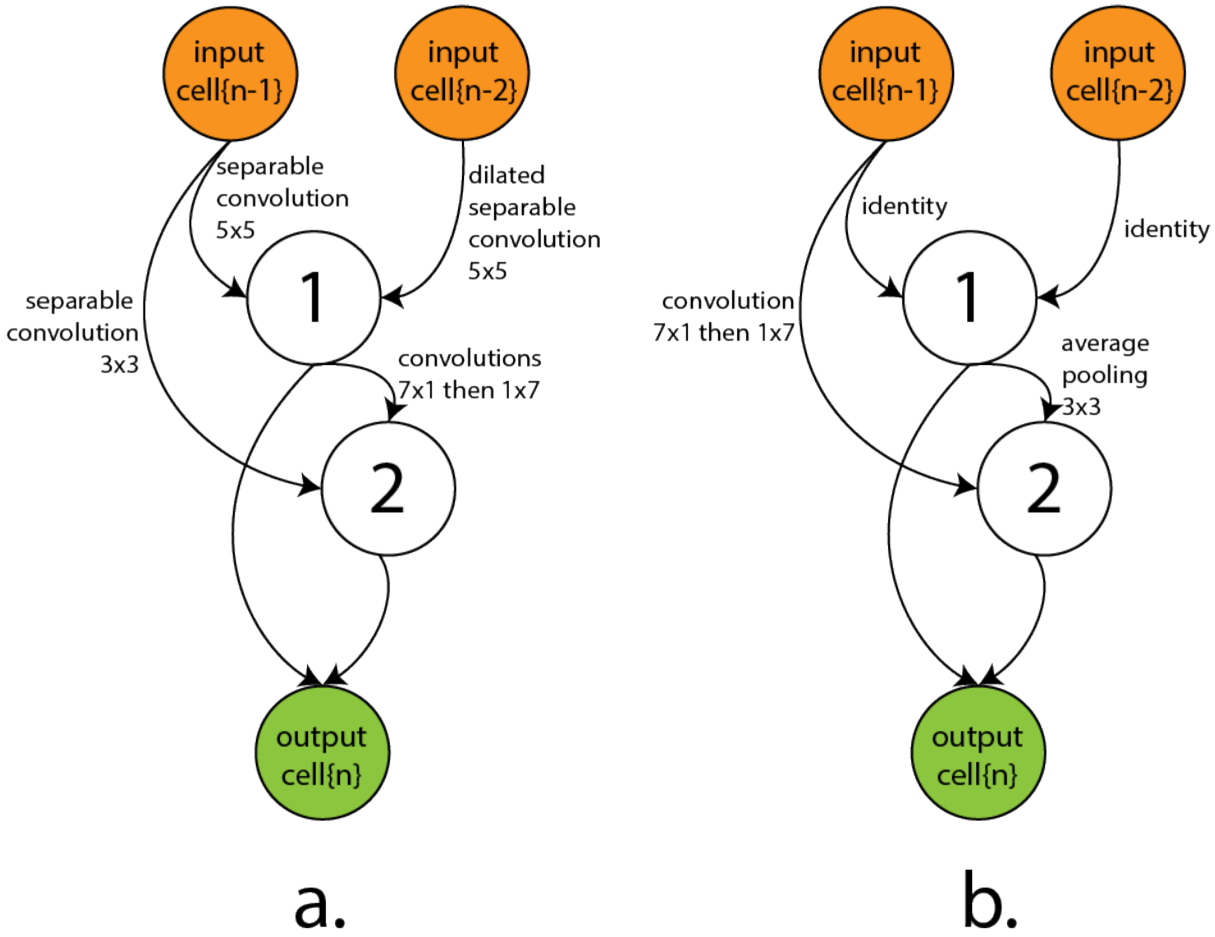


Figure 6.3: Cells found after performing the first stage of training on Al 5083 dataset. a) normal cell, b) reduction cell

The second stage of training tracks the accuracy of three network architectures composed with the cells found. The three architectures differ in the number of internal layers as described in Table 6.3:

Table 6.3: The three types of network size examined in current chapter.

Layer number	Internal cell type distribution for each network size		
	2-layer	6-layer	10-layer
layer 1	normal	normal	normal
layer 2	reduction	normal	normal
layer 3	-	normal	normal
layer 4	-	reduction	normal
layer 5	-	normal	normal
layer 6	-	normal	reduction
layer 7	-	-	normal
layer 8	-	-	normal
layer 9	-	-	normal
layer 10	-	-	normal

During the process of training the learning rate was not fixed at 10^{-2} but it decreased following a cosine annealing schedule given by formula:

$$\eta_{cur} = \frac{1}{2}\eta_{init}(1 + \cos(\frac{T_{cur}}{T_{max}}\pi))$$

where η_{cur} is the current learning rate, taking into account the initial learning rate $\eta_{init} = 10^{-2}$, the current epoch number T_{cur} and the maximum number of epochs $T_{max} = 100$. Figure 6.4 shows the evolution of learning rate graphically, for each epoch during the second stage of training.

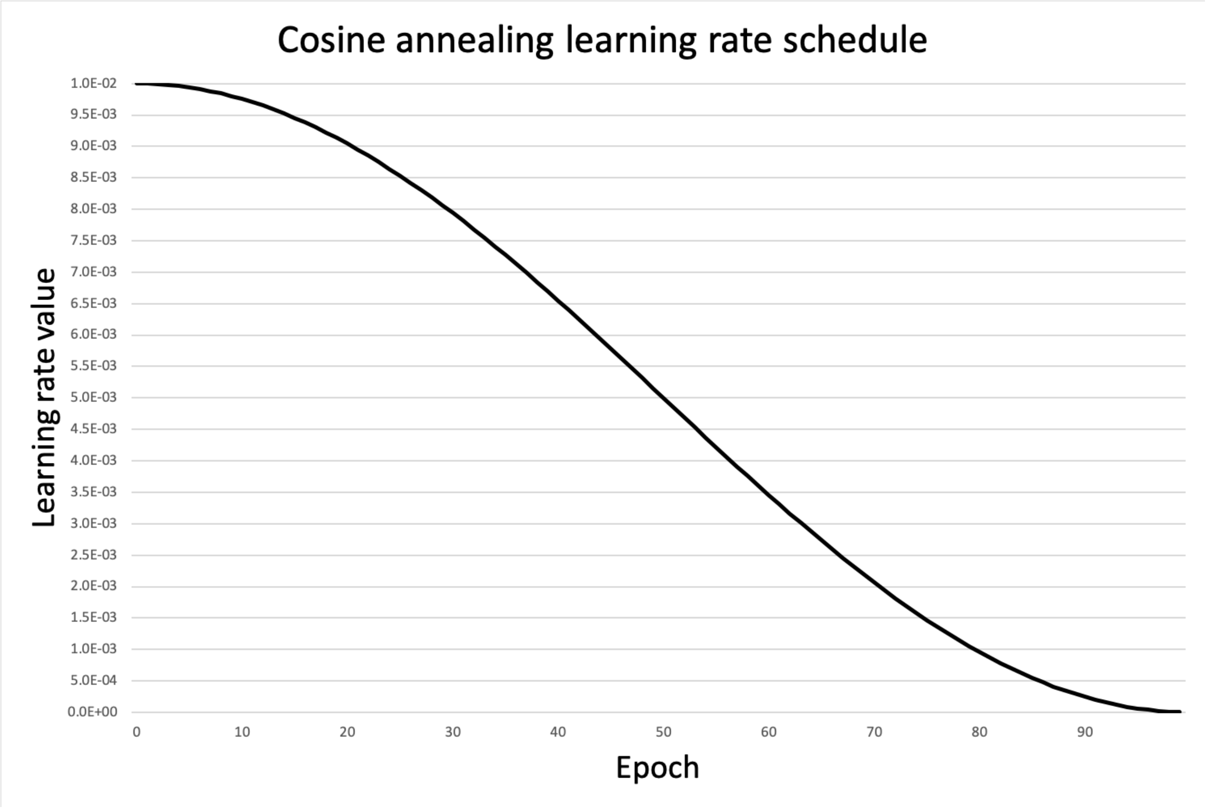


Figure 6.4: Cosine annealing learning rate schedule across all 100 epochs of second training stage

6.3.2 Al 5083 performance

Figures 6.5, 6.6 and 6.7 show the evolution of the accuracy measured after each epoch on the Al 5083 testing subset for tracking the evolution of the accuracy performance.

In Figures 6.5, 6.6, 6.7 the dotted yellow line is the exact accuracy measured after each epoch. The blue line is a weighted average of the previous values given by:

$$a_t^{(avg)} = (1-w)\sum_{i=0}^{t} w^{t-i} a_i$$

where $a_t^{(avg)}$ is the average weighted accuracy figure (a point on the blue line) at time t, $w = 0.8$ is a weight chosen arbitrarily influencing the smoothness of blue line and $a_t^{(inst)}$ is the instantaneous accuracy as calculated after processing the testing subset. The straight red line represents the accuracy obtained at the end of the training for model number 1 in Chapter 5, placed as a reference when compared to the current model's accuracy.

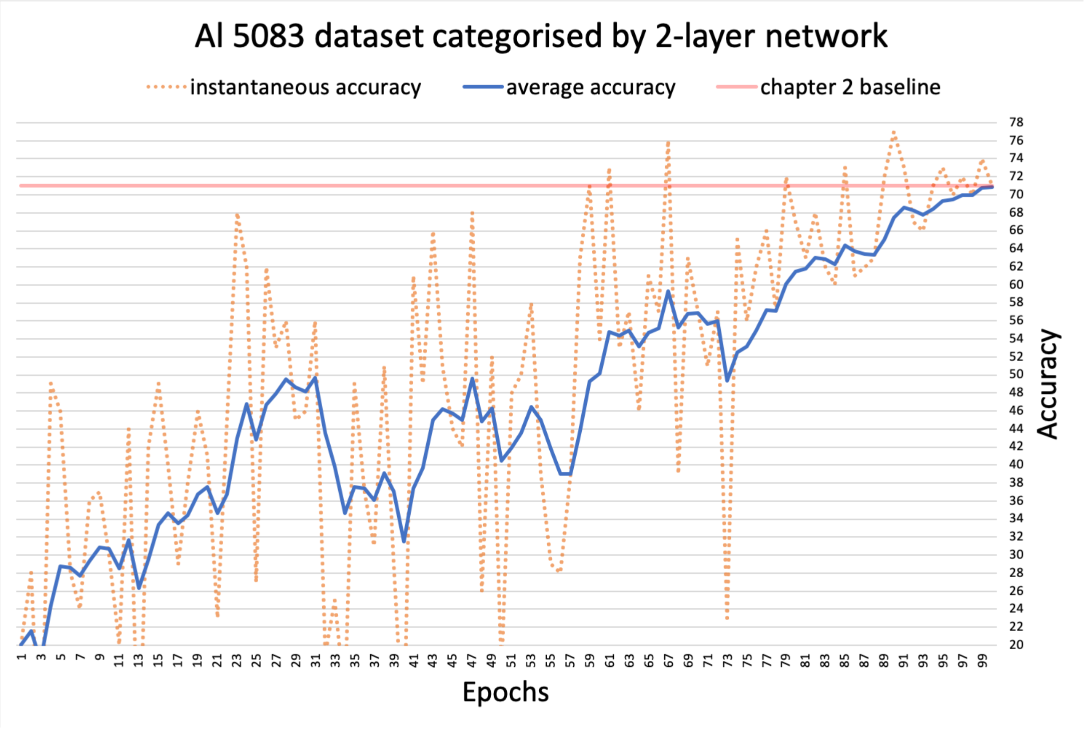


Figure 6.5: The testing accuracy evolution for Al 5083 dataset across 100 epochs of the second training stage as compared to Model 1 (ch. 5).

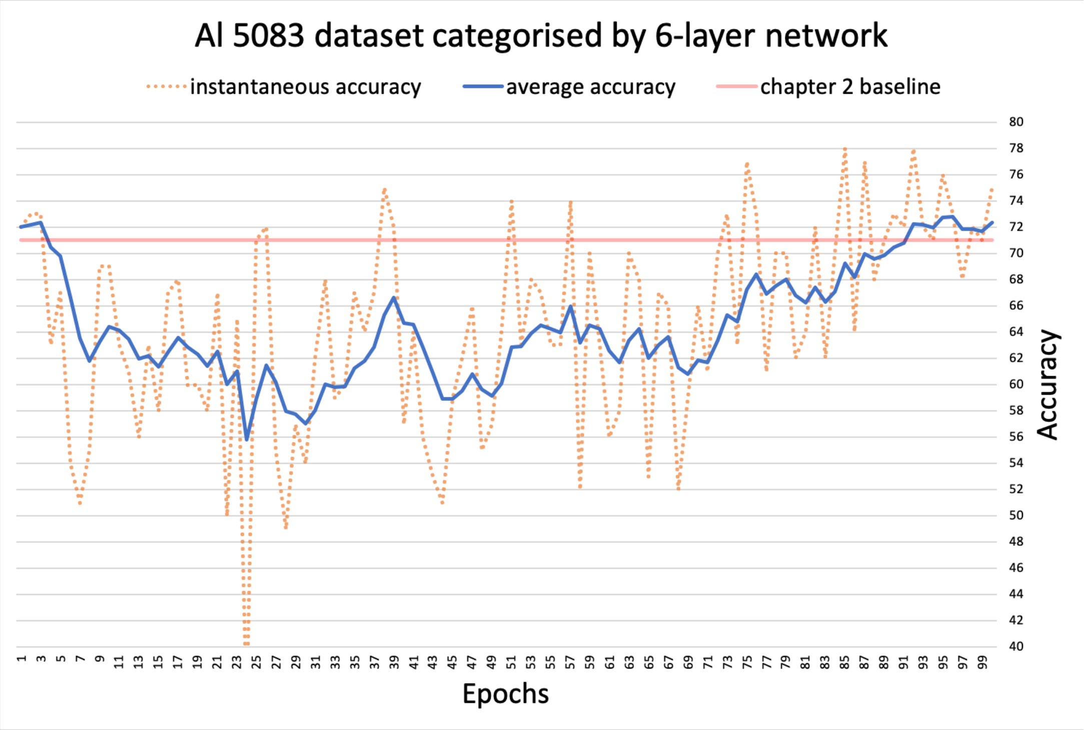


Figure 6.6: The testing accuracy evolution for Al 5083 dataset across 100 epochs of the second training stage as compared to Model 1 (ch. 5).

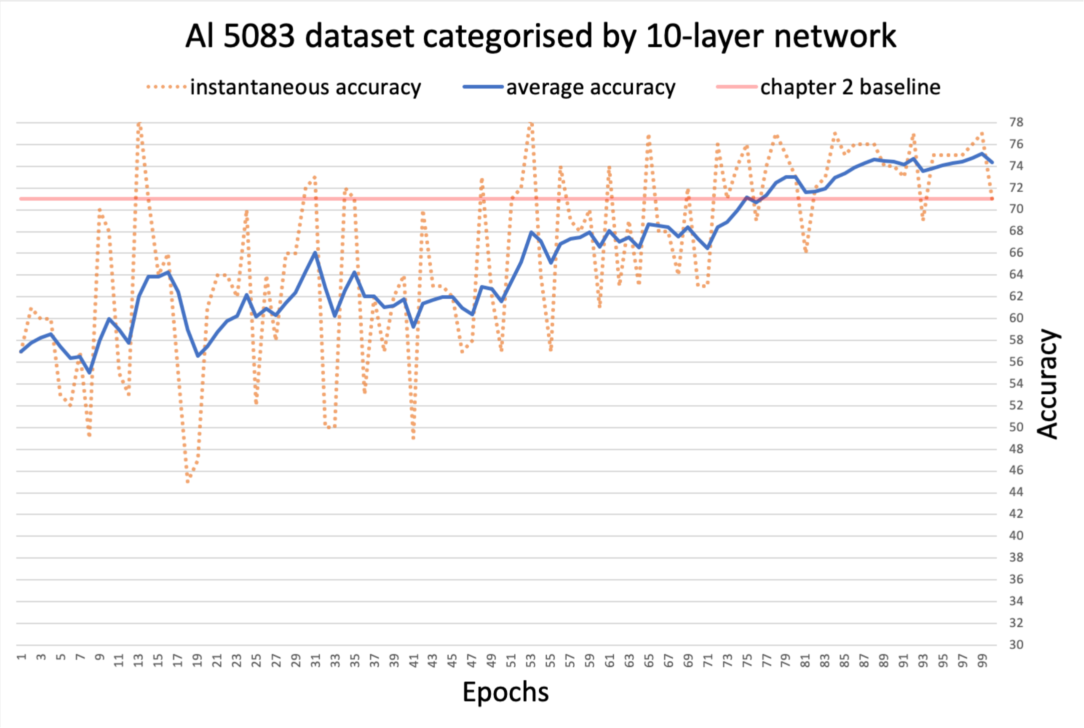


Figure 6.7: The testing accuracy evolution for Al 5083 dataset across 100 epochs of the second training stage as compared to Model 1 (ch. 5).

The networks achieved a more stable state only towards the end of training when learning rate decreased sufficiently for allowing the weights to settle on optimal values in the multi-dimensional feature space.

All three models studied, i.e. 2-layer, 6-layer and 10-layer, matches or outperforms the best model accuracy found in Chapter 5 (when the models hyper-parameters are defined by hand). The conclusion is that the first stage of training did find cells that are not very sensitive to the network depth. The difference in the accuracy between Model1 (Chapter 5) and the 10-layer model tested in the current chapter on Al 5083 is not very large standing at three percentage points (pp) (71% vs 74%), but the more striking difference is the number of parameters contained by each model. Table 6.4 shows the relative size between model 1 from Chapter 5 and all three model from this chapter.

Table 6.4: Number of model parameters for the models applied on the Al 5083 dataset.

Model	Number of parameters	Ratio
Model1 (ch. 5)	401670	1
10-layer model (Fig. 6.7)	305420	0.76
6-layer model (Fig. 6.6)	239148	0.60
2-layer model (Fig. 6.5)	173644	0.43

The 2-layer network matches Model 1 on accuracy using only 43% of the number of parameters, emphasising the model power for representing the welding images and the categories probability distribution. The accuracy difference is slightly larger, 1pp, when using a bigger model, i.e. 6-layer model, and even larger with a 10-layer model.

Although in terms of the accuracy, all the architectures from this chapter applied on Al 5083 dataset outperformed or matched the Model1, there is an area where Model1 performs better than the best performing 10-layer network, shown in Table 6.6. The precision, recall and F1-Score designed to encapsulate more detailed understanding of per-class performance show the Model1 as making fewer mistakes per class, although overall (over the entire dataset) it performs worst. The models in this chapter have a slight bias of categorising better the images belonging to the more numerous classes in the dataset. More balanced categories through recording more defective welds or eliminating some examples from numerous classes could rectify the behaviour. Removing images form dataset is the least preferred option since it leads to a reduction in the dataset and subsequently the model generalisation capability. Tables 6.5 and 6.6 show the confusion matrix and per-class metrics for the 10-layer model.

Table 6.5: The confusion matrix for 10-layer model applied on Al 5083 dataset.

Ground truth	Predicted categories					
	good weld	burn through	contamination	lack of fusion	misalignment	lack of penetration
good weld	2140	48	0	0	1	0
burn through	0	314	37	0	0	0
contamination	55	99	1689	0	235	0
lack of fusion	0	0	55	590	0	362
misalignment	173	0	0	357	157	42
lack of penetration	41	0	0	0	3	190

Table 6.6: Precision, Recall and F1-score metrics 10-layer model applied on Al 5083 dataset.

	Precision	Recall	F1-score
good weld	0.888	0.978	0.931
burn through	0.681	0.895	0.773
contamination	0.948	0.813	0.875
lack of fusion	0.623	0.586	0.604
misalignment	0.396	0.215	0.279
lack of penetration	0.652	0.812	0.459
average	0.643	0.716	0.654
average Model1 for comparison as of Table 5.6	0.696	0.756	0.691

6.3.3 SS304 performance

The Figures 6.8, 6.9 and 6.10 show the accuracy evolution during training for the 2-layer, 6-layer and 10-layer using the SS304 datasets and models identical to previously examined models applied on the Al 5083 dataset. The difference here is the extra model (Conv6) used for comparison. Figure 6.8, 6.9 and 6.10 show a green line representing the final accuracy obtained by Conv6 model in the Chapter 4 displayed as reference.

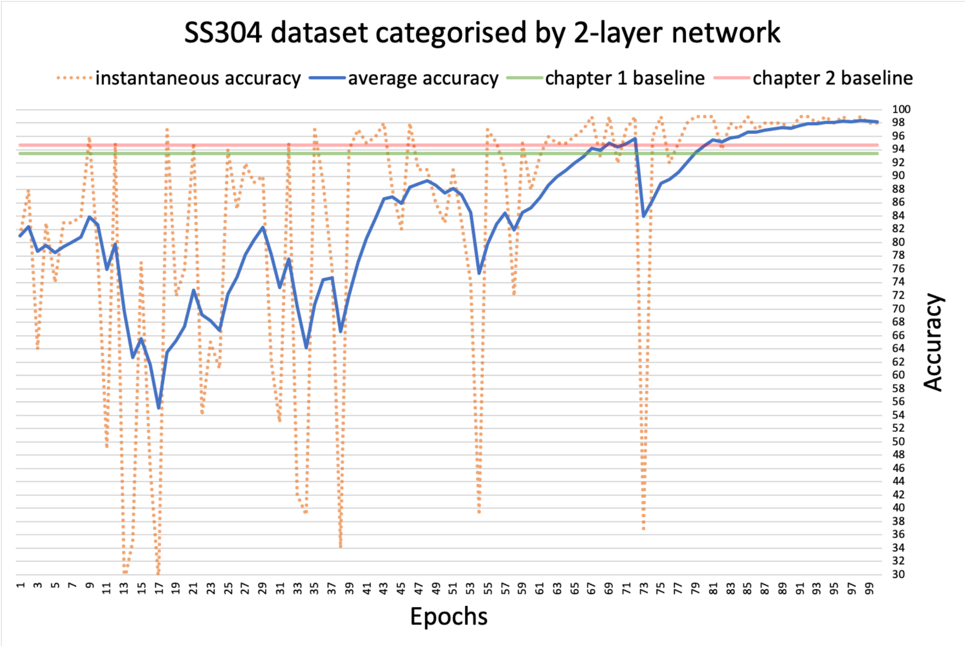


Figure 6.8: The testing accuracy evolution for the SS304 dataset across 100 epochs of the second stage training as compared to Model1 (ch. 5) - red line - and Conv6 (ch. 4) - green line -

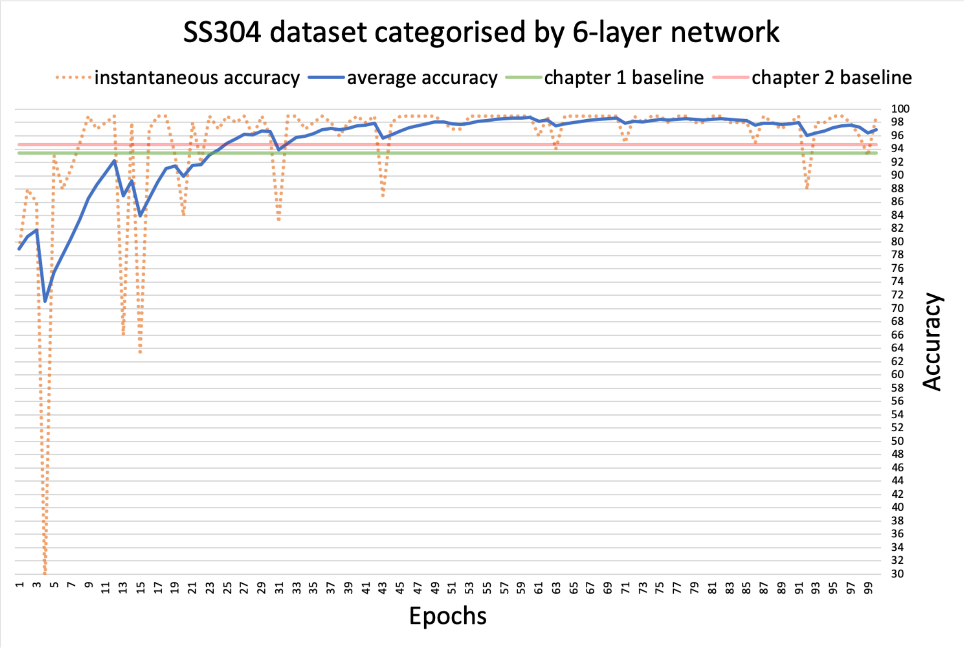


Figure 6.9: The testing accuracy evolution for the SS304 dataset across 100 epochs of the second stage training as compared to Model1 (ch. 5) - red line - and Conv6 (ch. 4) - green line -

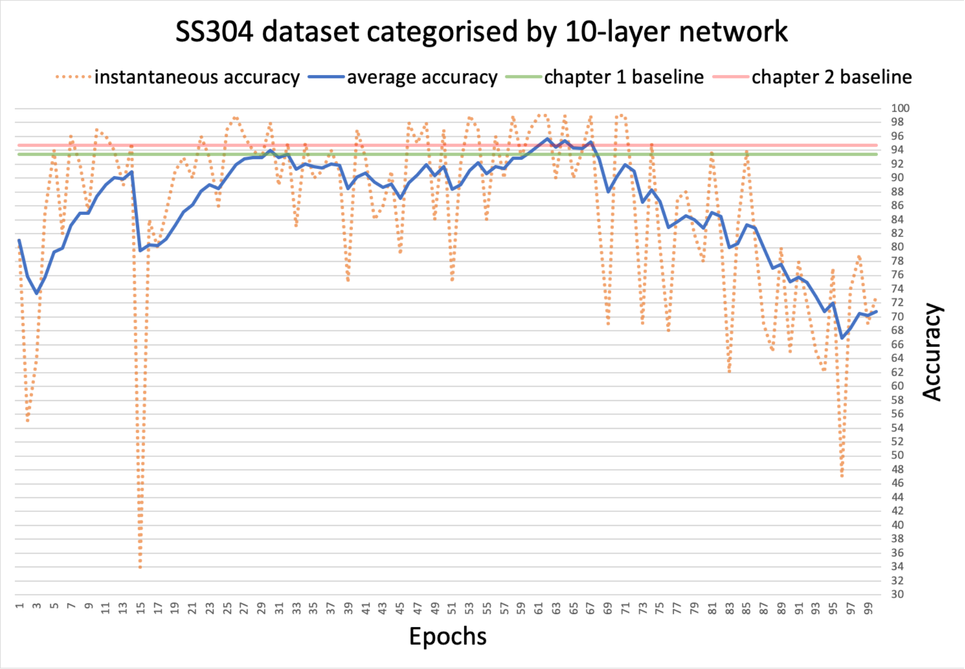


Figure 6.10: The testing accuracy evolution for the SS304 dataset across 100 epochs of the second stage training as compared to Model1 (ch. 5) - red line - and Conv6 (ch. 4) - green line -

The model sizes vary widely, the smallest, Conv6, being 77 times smaller than the largest, Model1, with the models analysed in this chapter in between these two extremes. Table 6.7

show the relative sizes of the 2-layer, 6-layer and 10-layer models relative to previously studied models.

Table 6.7: Number of model parameters for the models applied on SS304 dataset.

Model	Number of parameters	Ratio compared to Model1	Ratio compared to Conv6
Conv6 (ch. 4)	5190	0.013	1
Model1 (ch. 5)	401670	1	77.4
10-layer model (Fig. 6.7)	336908	0.76	64.9
6-layer model (Fig. 6.6)	270636	0.60	52.1
2-layer model (Fig. 6.5)	205132	0.43	39.5

The SS304 dataset is a simpler dataset for the neural networks to categorise, there is not as much variation within each category, and not as much similarity between categories, with every single model assessed displaying higher accuracy although the number of categories is the same (6 classes).

Tables 6.8 and 6.9 show the confusion matrix and per-class metrics for the 2-layer model applied on the SS304 dataset. The model exhibits improved performance in terms of accuracy, as well as per-class metrics for classes with less representation as ”lack of shielding gas”. The model achieves an accuracy of 98.2%.

Table 6.8: The confusion matrix for 2-layer model applied on SS304 dataset.

Ground truth	Predicted categories					
	good weld	burn through	contamination	lack of fusion	lack of shielding gas	high travel speed
good weld	1511	0	0	29	0	0
burn through	0	710	0	21	0	0
contamination	0	0	945	0	0	15
lack of fusion	0	0	0	1490	0	0
lack of shielding gas	0	0	0	1	101	0
high travel speed	0	0	0	0	0	249

Table 6.9: Precision, Recall and F1-score metrics 2-layer model applied on SS304 dataset.

Class	Precision	Recall	F1-score
good weld	1.000	0.981	0.990
burn through	1.000	0.971	0.985
contamination	1.000	0.984	0.992
lack of fusion	0.967	1.000	0.983
lack of shielding gas	1.000	0.990	0.995
high travel speed	0.943	1.000	0.971
average	0.985	0.988	0.986
average Model2 for comparison as of Table 5.10	0.900	0.950	0.912
average Conv6 for comparison as of Table 4.9	0.77	0.79	0.78

The networks composition in this subsection uses the cells found during the same first stage of training which used only the Al 5083 dataset. By omitting the SS304 dataset from the first stage of training, the intention is to demonstrate the cell potential for categorising welding images of different material highlighting the cell **transferability** capability.

The accuracy obtained with the optimised architectures outperform the previously hand designer networks by a margin of 6pp. The precision, recall and F1-score also improved showing a robust categorisation capability.

An outlier is the 10-layer network of which performance improves then degrades during the training. The overfitting behaviour (training accuracy increases while testing accuracy decreases) results from the network depth. The 10-layer network has the power to represent the probability distribution with an increased degree of freedom emergent from higher order functions able to develop.

At the same time, the 6-layer network strikes the right balance between the internal functions complexity and the correct representation of probability distribution, converging fast towards the high accuracy and maintaining approximatively the same level of accuracy across the entire second stage of the training process.

6.4 Conclusion

This chapter makes one step further towards the aim of identifying defects during the welding process by treating the model hyper-parameters as a continuum where an optimiser could be applied to navigate the multi-dimensional landscape. The study framed this optimisation problem by constraining the model to find two cells, normal cell and reduction cell, that placed in the context of a network categorises welding images with high accuracy.

The study found a set of two cells, built three models (2-layer, 6-layer and 10-layer model), tracked the test accuracy and analysed the performance of the three networks when applied on the Al 5083 and SS304 welding images. The chapter shows the comparison between the three models and the previously hand-designer models discovered in the previous chapters.

The system achieved state of the art accuracy of 74% in classifying six welding process defects exceeding the previous best accuracy of 71% on the Al 5083 dataset. On the SS304 dataset, the 2-layer model not only improves on the overall accuracy and the model quality but delivered the accuracy performance with only 43% of the number of parameters, as compared to the previous chapter.

Another aim of the current chapter was to set the training procedure for the models' **transferability** capability examination. The cells found using the Al 5083 dataset have the potential of achieving high performance when applied on SS304 dataset, speeding up the process of finding a better model for assessing Al 5083 and SS304 welding and potentially for many more materials considered weldable.

Chapter 7

Conclusions and future work

The current work studies the welding defects identification using a visible spectrum camera and applying the neural network paradigm for processing the images. The system involves acquiring images with an HDR camera, pre-process the image, then use the neural networks architectures for mapping between the aspect of a defect and the name of the defect.

This study starts with the camera selection. The sensor requirement was the capability for recording in the visible spectrum because the assessment involved the images categorisation similar to a human welder. The problem with this approach is the powerful arc light, which obscures the weld pool. An imaging sensor capable of countering the arc light and capable of reproducing images of the weld pool and surrounding area by using a high dynamic range (HDR) capability led to the selection of Xiris XVC-1000. This study generated two datasets of high quality, high dynamic range images of diverse welding conditions. It considers the acquisition of high-quality images as a contributing factor towards the end goal of identifying weld defects.

The welding process defects and problems considered in this study, occurring commonly during TIG welding, are burn through when the sheets are thin, contamination when workpieces are impure or lack of shielding gas, to name only a few of the problems.

The most substantial part of the current study focuses on processing the images of the TIG welding process of SS304 and Al 5083 recorded with the HDR camera.

The datasets cover six classes representatives the welding conditions observed for SS304 and

Al 5083. The six classes for the SS304 are a good weld, burn through, contamination, lack of fusion, lack of shielding gas and high travel speed. The Al 5083 dataset contains images of welds representative of a good weld, burn through, contamination, lack of fusion, misalignment and lack of penetration. In total the SS304 dataset contains 30,008 images of 56 welding runs and the Al 5083 dataset counts 33,254 images of 60 welding runs.

The previous studies involved hand-designed features and filters for extracting information from an image (signal) followed by hard-coded rules for establishing the correlation between welding states and category or welding states and actions. Unlike those studies, the approach in the current work is the adoption of a paradigm able to build an internal representation of the welding state and map it to a category.

The processing paradigm identified during the literature review for the ability to process the highly-dimensional mapping between an image's pixels and the category is machine learning, more precisely neural network involving supervised learning.

The study used images of SS304 welding for testing the initial idea. The test involved two settings: the discrimination between good welds vs defective welds and the discrimination between all six classes within the dataset. The results for the classification between the good welding conditions and the defective state achieved an accuracy of 89.5% and distinguishing between the good welding conditions and five other defective conditions achieved an accuracy of 93%. The test involved the use of the convolutional neural networks and the fully connected neural networks, assessing the accuracy of both settings and the influence of the images quality. The results of the neural network paradigm application on SS304 fed into the second chapter, where the scope of finding the optimal architecture broadened. The second chapter described the use of neural networks on the Al 5083 dataset. The work involved the model ranking based on the error analysis, the hyper-parameter ablation, i.e. which hyper-parameter is more important, the robustness check for problems with different difficulties, i.e. a different number of classes, and the influence of image fidelity reduction.

The analysis covered 12 neural network models, five learning rates, three problem difficulties, i.e. 2-class, 4-class and 6-class problem, and two image resolutions. The convolutional neural networks built better representations than the fully connected neural networks and achieved an accuracy performance of 71%, 89% and 95% on 6-class, 4-class and 2-class problem, respectively.

The conclusion emerging from the analysis applied to the Al 5083 dataset is the model hyper-parameters are the most significant differentiator towards better accuracy. An architecture optimisation process, therefore, has the potential to exhibit better accuracy performance. A method for optimising the neural architecture search (NAS) was implemented, improving the accuracy for identifying the correct welding conditions to 96% on SS304 and from 74% on Al 5083. At the same time, the architecture found through the optimisation procedure proves the potential for transferring between different material datasets and outperforming the hand-designed architectures.

The system built during this study and the processing approach involved proved the capability of assessing the welding conditions of TIG welding similarly to an experienced welder. The gap between the experience welder accuracy and versatility for welding process defects identification and the system studied here is still significant, but the difference is smaller compared to the beginning of this study.

7.1 Future work

The paths for taking the study further split into several possible ways forward.

The first way forward is the expansion to a different welding process. The step involves the sensor capabilities reassessment and adjustment.

The second approach is the dataset expansion for encompassing several other types of defects not included in the current study, as well as TIG welding of materials other that SS304 or Al 5083 or different workpiece geometry.

A third technique is the use of hybrid input for training the neural networks. A hybrid input could be composed of an image and welding parameters or two images in a different spectrum, or two images at different viewing angle, or an image and an acoustic signal or any other combination.

A fourth way forward is the generation of synthetic images of defects, therefore enhancing and expanding the dataset available for a system like the one presented here for learning a wider variety of classification with greater accuracy.

A fifth approach could look into the mapping between the neural network input and output signal. In the current study, the mapping was between an image and the assigned label denoting the state. The jump between image and output could be larger if the output is the welding parameters, or smaller if the output is the weld area. Another path would be using the current approach to output not only the categorisation, but a combination of categorisation and few welding parameters changes to bring a defective welding state towards a good welding state.

A sixth approach could be the use of unsupervised learning for training the model. It would alleviate the need for labelled examples of good and defective welds by allowing the model to cluster images with a similar aspect within one group.

Bibliography

[1] A. Nieto, D. López Vilarino, and V. Brea Snchez. Towards the optimal hardware architecture for computer vision. *Machine Vision - Applications and Systems*, 2012.

[2] J-P. Barthoux. Narrow gap welding of heavy wall thickness materials in nuclear and fossil fuel industries. 2008.

[3] Y. Fujita, T. Ogawa, S. Asai, S. Yamamoto, T. Ohdake, and M. Ochiai. Development of a welding monitoring system for in-process quality control of thick walled pipe. *Welding in the World*, 56(11-12):15–25, 2012.

[4] I-S. Kim, J-S. Son, S-H. Lee, and P.K.D.V. Yarlagadda. Optimal design of neural networks for control in robotic arc welding. *Robotics and Computer-Integrated Manufacturing*, 20(1):57 – 63, 2004.

[5] M. Luo and Y.C. Shin. Estimation of keyhole geometry and prediction of welding defects during laser welding based on a vision system and a radial basis function neural network. *The International Journal of Advanced Manufacturing Technology*, 81(1-4):263–276, 2015.

[6] K. Weman. 4 - arc welding: an overview. In K. Weman, editor, *Welding Processes Handbook (Second Edition)*, Woodhead Publishing Series in Welding and Other Joining Technologies, pages 31 – 50. Woodhead Publishing, second edition edition, 2012.

[7] M. Onsoeien, D.L. Olson, S. Liu, and R Peters. Effect of hydrogen in an argon gtaw shielding gas: Arc characteristics and bead morphology. *Welding Journal - WELD J*, 74, 1995.

[8] Z.H. Rao, S.M. Liao, and H.L. Tsai. Effects of shielding gas compositions on arc plasma and metal transfer in gas metal arc welding. *Journal of Applied Physics*, 107(4):044902, 2010.

[9] K. Weman. 18 - welding residual stress and distortion. In K. Weman, editor, *Welding Processes Handbook (Second Edition)*, Woodhead Publishing Series in Welding and Other Joining Technologies, pages 185 – 189. Woodhead Publishing, second edition edition, 2012.

[10] J.N. Pires, A. Loureiro, and G. Bölmsjo. *Welding robots: technology, system issues and application*. Springer Science & Business Media, 2006.

[11] H. Maryon, R.M. Organ, O.W. Ellis, R.M. Brick, R. Sneyers, E.E. Herzfeld, and F.K. Naumann. Early near eastern steel swords. *American Journal of Archaeology*, 65(2):173–184, 1961.

[12] M.P. Groover. *Fundamentals of Modern Manufacturing: Materials, Processes, and Systems*. John Wiley & Sons, 2010.

[13] J.D. Jackson. *Classical Electrodynamics, 3rd Edition*. 1998.

[14] Cavitar. http://www.cavitar.com/. Accessed: 2019-03-27.

[15] Oxford lasers. https://www.oxfordlasers.com/. Accessed: 2019-03-27.

[16] Invisuale. https://www.invisuale.com/. Accessed: 2019-03-27.

[17] Intertest. http://www.intertest.com/. Accessed: 2019-03-27.

[18] Xiris. http://xiris.com. Accessed: 2019-03-27.

[19] R.J. Renwick and R.W. Richardson. Experimental investigation of GTA weld pool oscillations. *Welding Journal*, 62(2):29 – 35, 1983.

[20] Y.H. Xiao and G. den Ouden. A study of GTA weld pool oscillations. *Welding Journal*, 69(8):298 – 293, 1990.

[21] L.A. Lott. Ultrasonic detection of molten/solid interfaces in weld pools. *Material Evaluation*, 42:337 – 341, 1983.

[22] D.E. Hardt and J.M. Katz. Ultrasonic measurement of weld penetration. *Welding Journal*, 63(9):273 – 281, 1984.

[23] N.M. Carlson and J.A. Johnson. Ultrasonic sensing of weld pool penetration. *Welding Journal*, 67(11):239 – 246, 1988.

[24] W. Chen and B.A. Chin. Monitoring joint penetration using infrared sensing techniques. *Welding Journal*, 69(4):181 – 185, 1990.

[25] P. Ghanty, M. Vasudevan, D.P. Mukherjee, N.R. Pal, N. Chandrasekhar, V. Maduraimuthu, A.K. Bhaduri, P. Barat, and B. Raj. Artificial neural network approach for estimating weld bead width and depth of penetration from infrared thermal image of weld pool. *Sci. Technol. Weld. Join.*, 13(4):395 – 401, 2008.

[26] W. Zhang, Y. Liu, X. Wang, and Y. Zhang. Characterization of three-dimensional weld pool surface in gas tungsten arc welding. *Welding journal*, 91, 07 2012.

[27] W. Zhang, X. Wang, and Y. Zhang. Analytical real-time measurement of a three-dimensional weld pool surface. *Measurement Science and Technology*, 24:5011–, 11 2013.

[28] Y. Zou, D. Du, B. Chang, L. Ji, and J. Pan. Automatic weld defect detection method based on kalman filtering for real-time radiographic inspection of spiral pipe. 2016.

[29] H.M. Kim, Y.W. Rho, H.R. Yoo, S.H. Cho, D.K. Kim, S.J. S. J. Koo, and G.S. Park. A study on the measurement of axial cracks in the magnetic flux leakage ndt system. In *2012 IEEE International Conference on Automation Science and Engineering (CASE 2012)(CASE)*, volume 00, pages 624–629, 2013.

[30] F. Jian, Z. Jun-Feng, L. Sen-Xiang, W. Hong-Yang, and M. Rui-Ze. Three-axis magnetic flux leakage in-line inspection simulation based on finite-element analysis. *Chinese Physics B*, 22(1):018103, 2013.

[31] M.M. Tehranchi, M. Ranjbaran, and H. Eftekhari. Double core giant magneto-impedance sensors for the inspection of magnetic flux leakage from metal surface cracks. *Sensors and Actuators A: Physical*, 170(1):55 – 61, 2011.

[32] M.M. Tehranchi, S.M. Hamidi, H. Eftekhari, M. Karbaschi, and M. Ranjbaran. The inspection of magnetic flux leakage from metal surface cracks by magneto-optical sensors. *Sensors and Actuators A: Physical*, 172(2):365 – 368, 2011.

[33] B. Vijay and C. Lynann. Residual magnetic flux leakage: A possible tool for studying pipeline defects. *Journal of Nondestructive Evaluation*, 22(4):117–125, 2003.

[34] B.W. Drinkwater and P.D. Wilcox. Ultrasonic arrays for non-destructive evaluation: A review. *NDT & E International*, 39(7):525 – 541, 2006.

[35] A. El Kouche and H.S. Hassanein. Ultrasonic non-destructive testing (ndt) using wireless sensor networks. *Procedia Computer Science*, 10:136 – 143, 2012. ANT 2012 and MobiWIS 2012.

[36] M. Akhnak, O. Martinez, L.G. Ullate, and F. Montero de Espinosa. 64 elements two-dimensional piezoelectric array for 3d imaging. *Ultrasonics*, 40(1):139 – 143, 2002.

[37] J.L. Rose. *Ultrasonic Guided Waves in Solid Media*. Cambridge University Press, 2014.

[38] R. Singh. Chapter 5 - penetrant testing. In R. Singh, editor, *Applied Welding Engineering*, pages 283 – 291. Butterworth-Heinemann, Boston, 2012.

[39] J. Mirapeix, P.B. Garca-Allende, A. Cobo, O.M. Conde, and J.M. Lpez-Higuera. Real-time arc-welding defect detection and classification with principal component analysis and artificial neural networks. *{NDT} & E International*, 40(4):315 – 323, 2007.

[40] D. Bebiano and S.C.A Alfaro. A weld defects detection system based on a spectrometer. *Sensors*, 9(4):2851–2861, 2009.

[41] A. Cobo, A. Álvarez, D. Solana, J.M. Mirapeix, J.M. López-Higuera, O.M. Conde, and P.B. G. *Optical Methods for On-line Quality Assurance of Welding Processes in Nuclear Steam Generators*. INTECH Open Access Publisher, 2011.

[42] F. Timm, S. Klement, T. Martinetz, and E. Barth. Welding inspection using novel specularity features and a one-class svm. In *VISAPP*, 2009.

[43] H.S. Song and Y.M. Zhang. Three-dimensional reconstruction of specular surface for a gas tungsten arc weld pool. *Measurement Science and Technology*, 18(12):3751, 2007.

[44] Y. Liu and Y. Zhang. Control of 3d weld pool surface. *Control Engineering Practice*, 21(11):1469 – 1480, 2013. Advanced Software Engineering in Industrial Automation (INCOM09).

[45] Y.K. Liu and Y.M. Zhang. Model-based predictive control of weld penetration in gas tungsten arc welding. *IEEE Transactions on Control Systems Technology*, 22(3):955–966, May 2014.

[46] Y. Liu and Y. Zhang. Iterative local anfis-based human welder intelligence modeling and control in pipe gtaw process: A data-driven approach. *IEEE/ASME Transactions on Mechatronics*, 20(3):1079–1088, June 2015.

[47] Y. Liu and Zhang Y. Fusing machine algorithm with welder intelligence for adaptive welding robots. *Journal of Manufacturing Processes*, 27:18 – 25, 2017.

[48] S.K. Lee and Na S.J. A study on automatic seam tracking in pulsed laser edge welding by using a vision sensor without an auxiliary light source. *Journal of Manufacturing Systems*, 21(4):302 – 315, 2002.

[49] W. Jamrozik and M. Fidali. Evaluation of the suitability of ir and tv image aggregation algorithms for the purposes of welding process assessment. 2012.

[50] M.S. Węglowski. Utilization of the arc light emission emitted during tig welding to monitoring this process. 2007.

[51] W. Lucas, D. Bertaso, G. Melton, J. Smith, and C. Balfour. Real-time vision-based control of weld pool size. *Welding International*, 26(4):243–250, 2012.

[52] J.S. Smith and C. Balfour. Realtime topface vision based control of weld pool size. *Industrial Robot: An International Journal*, 32(4):334–340, 2005.

[53] T.Y. Lim, M.M. Ratnam, and M.A. Khalid. Automatic classification of weld defects using simulated data and an mlp neural network. *Insight-Non-Destructive Testing and Condition Monitoring*, 49(3):154–159, 2007.

[54] M. Carrasco and D. Mery. Automatic multiple view inspection using geometrical tracking and feature analysis in aluminum wheels. *Machine Vision and Applications*, 22(1):157–170, 2011.

[55] Z. Liao and J. Sun. Image segmentation in weld defect detection based on modified background subtraction. In *Image and Signal Processing (CISP), 2013 6th International Congress on*, volume 2, pages 610–615, 2013.

[56] P. Kornprobst, R. Deriche, and G. Aubert. Image sequence analysis via partial differential equations. *Journal of Mathematical Imaging and Vision*, 11(1):5–26, 1999.

[57] N. Friedman and S.J. Russell. Image segmentation in video sequences: A probabilistic approach. *CoRR*, abs/1302.1539, 2013.

[58] G. Schwab, J.P.H. Steele, and T.L. Vincent. Vision-based spatter classification for contaminant detection. *Welding Journal*, 88(6):121–30, 2009.

[59] X.F. Liu, C.S. Wu, C.B. Jia, and G.K. Zhang. Visual sensing of the weld pool geometry from the topside view in keyhole plasma arc welding. *Journal of Manufacturing Processes*, 26(Complete):74–83, 2017.

[60] C. Jiang, F. Zhang, and Z. Wang. Image processing of aluminum alloy weld pool for robotic vppaw based on visual sensing. *IEEE Access*, 5:21567–21573, 2017.

[61] J. Shao, D. Du, B. Chang, and H. Shi. Automatic weld defect detection based on potential defect tracking in real-time radiographic image sequence. *NDT & E International*, 46:14 – 21, 2012.

[62] M. Fidali and W. Jamrozik. Diagnostic method of welding process based on fused infrared and vision images. *Infrared Physics & Technology*, 61:241 – 253, 2013.

[63] Y.M. Zhang, L. Li, and R. Kovacevic. Neurofuzzy model based control of weld fusion zone geometry. In *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, volume 4, pages 2483–2487 vol.4, June 1997.

[64] R. Kovacevic and Y.M. Zhang. Neurofuzzy model-based weld fusion state estimation. *IEEE Control Systems Magazine*, 17(2):30–42, April 1997.

[65] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[66] M.D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[67] I.J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V.D. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *CoRR*, abs/1312.6082, 2013.

[68] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A.C. Courville, and Y. Bengio. Maxout networks. *CoRR*, abs/1302.4389, 2013.

[69] G. Dahl, M. Ranzato, A. Mohamed, and G.E Hinton. Phone recognition with the mean-covariance restricted boltzmann machine. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 469–477. Curran Associates, Inc., 2010.

[70] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*, 2012.

[71] A. Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.

[72] R. Pascanu, C. Gülçehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. *CoRR*, abs/1312.6026, 2013.

[73] T. Mikolov, I. Sutskever, A. Deoras, H. Le, S. Kombrink, and J. Cernocký. Subword language modeling with neural networks. 2011.

[74] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):855–868, 2009.

[75] H. Jaeger and H. Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication, 2004.

[76] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[77] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[78] B. Widrow and M.E. Hoff. Adaptive switching circuits. In *1960 IRE WESCON Convention Record, Part 4*, pages 96–104, New York, 1960. Institute of Radio Engineers, Institute of Radio Engineers.

[79] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[80] Y. LeCun, B.E. Boser, J.E. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, and L.D. Jackel. Handwritten digit recognition with a back-propagation network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan-Kaufmann, 1990.

[81] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[82] C.M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, 2007.

[83] G. Huang, Z. Liu, and K.Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[84] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[85] B. Zoph and Q.V. Le. Neural architecture search with reinforcement learning. *CoRR*, abs/1611.01578, 2016.

[86] B. Zoph, V. Vasudevan, J. Shlens, and Q.V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017.

[87] E. Real, A Aggarwal, Y. Huang, and Q.V. Le. Regularized evolution for image classifier architecture search. *CoRR*, abs/1802.01548, 2018.

[88] C. Liu, B. Zoph, J. Shlens, W. Hua, L. Li, L. Fei-Fei, A.L. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. *CoRR*, abs/1712.00559, 2017.

[89] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. *ArXiv e-prints*, 2018.

[90] F. Rosenblatt. *Principles of Neurodynamics*. Spartan Books, 1959.

[91] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[92] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 153–160. MIT Press, 2007.

[93] R. Salakhutdinov and G. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2

of *Proceedings of Machine Learning Research*, pages 412–419, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.

[94] M. Ranzato, C. Poultney, S Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1137–1144. MIT Press, 2007.

[95] D.C. Ciresan, U. Meier, L.M. Gambardella, and J. Schmidhuber. Deep big simple neural nets excel on handwritten digit recognition. *CoRR*, abs/1003.0358, 2010.

[96] C. Szegedy, W. Liu, Y. Jia, P Sermanet, S.E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[97] D.H. Hubel and T.N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London)*, 195:215–243, 1968.

[98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[99] Y. LeCun, L. Bottou, and Y. Bengio. Reading checks with multilayer graph transformer networks. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 151–154 vol.1, Apr 1997.

[100] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

[101] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[102] M.D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[103] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[104] Y. Ou and L. Yueping. Quality evaluation and automatic classification in resistance spot welding by analyzing the weld image on metal bands by computer vision. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(5):301–314, 2015.

[105] W. Hou, Y. Wei, J. Guo, Y. Jin, and C. Zhu. Automatic detection of welding defects using deep neural network. In *Journal of Physics Conference Series*, volume 933 of *Journal of Physics Conference Series*, page 012006, 2018.

[106] B. Baker, O. Gupta, N. Naik, and R. Raskar. Designing neural network architectures using reinforcement learning. *CoRR*, abs/1611.02167, 2016.

[107] H. Cai, T. Chen, W. Zhang, Y. Yu, and J. Wang. Reinforcement learning for architecture search by network transformation. *CoRR*, abs/1707.04873, 2017.

[108] H. Mendoza, A. Klein, M. Feurer, J.T. Springenberg, and F. Hutter. Towards automatically-tuned neural networks. In *Proceedings of the Workshop on Automatic Machine Learning*, volume 64 of *Proceedings of Machine Learning Research*, pages 58–65, New York, New York, USA, 2016. PMLR.

[109] M. Suganuma, S. Shirakawa, and T. Nagao. A genetic programming approach to designing convolutional neural network architectures. *CoRR*, abs/1704.00764, 2017.

[110] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[111] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[112] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu. Hierarchical representations for efficient architecture search. *CoRR*, abs/1711.00436, 2017.

[113] H. Pham, M.Y. Guan, B. Zoph, Q.V. Le, and J. Dean. Efficient neural architecture search via parameter sharing. *CoRR*, abs/1802.03268, 2018.

[114] T. Elsken, J. Hendrik Metzen, and F. Hutter. Efficient multi-objective neural architecture search via lamarckian evolution. *ArXiv e-prints*, 2018.

[115] H. Cai, J. Yang, W. Zhang, S. Han, and Y. Yu. Path-level network transformation for efficient architecture search. *ArXiv e-prints*, 2018.

[116] P.J. Angeline, G.M. Saunders, and J.B. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(1):54–65, 1994.

[117] K.O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[118] F. Dario, D. Peter, and M. Claudio. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, 2008.

[119] K.O. Stanley, D.B. DÁmbrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

[120] E. Real, S. Moore, A. Selle, S. Saxena, Y.L. Suematsu, Q.V. Le, and A. Kurakin. Large-scale evolution of image classifiers. *CoRR*, abs/1703.01041, 2017.

[121] D.E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. volume 1 of *Foundations of Genetic Algorithms*, pages 69 – 93. Elsevier, 1991.

[122] K. Kandasamy, W. Neiswanger, J. Schneider, B. Póczos, and E. Xing. Neural architecture search with bayesian optimisation and optimal transport. *CoRR*, abs/1802.07191, 2018.

[123] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 2546–2554, USA, 2011. Curran Associates Inc.

[124] F. Hutter, H.H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, LION'05, pages 507–523, Berlin, Heidelberg, 2011. Springer-Verlag.

[125] S. Saxena and J. Verbeek. Convolutional neural fabrics. *CoRR*, abs/1606.02492, 2016.

[126] R. Shin, C. Packer, and D. Song. Differentiable neural network architecture search, 2018.

[127] K. Ahmed and L. Torresani. Connectivity learning in multi-branch networks. *CoRR*, abs/1709.09582, 2017.

[128] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter. Fast bayesian optimization of machine learning hyperparameters on large datasets. *CoRR*, abs/1605.07079, 2016.

[129] P. Chrabaszcz, I. Loshchilov, and F. Hutter. A downsampled variant of imagenet as an alternative to the CIFAR datasets. *CoRR*, abs/1707.08819, 2017.

[130] A. Zela, A. Klein, S. Falkner, and F. Hutter. Towards automated deep learning: Efficient joint neural architecture and hyperparameter search. *ArXiv e-prints*, 2018.

[131] L. Li, K.G. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Efficient hyperparameter optimization and infinitely many armed bandits. *CoRR*, abs/1603.06560, 2016.

[132] S. Falkner, A. Klein, and F. Hutter. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. *ArXiv e-prints*, 2018.

[133] T. Domhan, J.T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 3460–3468. AAAI Press, 2015.

[134] K. Swersky, J. Snoek, and R. Prescott Adams. Freeze-thaw bayesian optimization. *ArXiv e-prints*, 2014.

[135] A. Brock, T. Lim, J.M. Ritchie, and N. Weston. SMASH: one-shot model architecture search through hypernetworks. *CoRR*, abs/1708.05344, 2017.

[136] G. Bender, P. Kindermans, B. Zoph, V. Vasudevan, and Q. Le. Understanding and simplifying one-shot architecture search. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 550–559, Stockholmsmssan, Stockholm Sweden, 2018. PMLR.

[137] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[138] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[139] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[140] V. Nair and G.E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[141] B. Graham. Fractional max-pooling. *CoRR*, abs/1412.6071, 2014.