2023

ESTABLISHING THE FOUNDATION TO ROBOTIZE COMPLEX WELDING PROCESSES THROUGH LEARNING FROM HUMAN WELDERS BASED ON DEEP LEARNING TECHNIQUES

Rui Yu
University of Kentucky, rui.yu@uky.edu
Author ORCID Identifier:
https://orcid.org/0000-0002-6410-6088
Digital Object Identifier: https://doi.org/10.13023/etd.2023.410

Right click to open a feedback form in a new tab to let us know how this document benefits you.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

<div align="right">

Rui Yu, Student

YuMing Zhang, Major Professor

Daniel Lau, Director of Graduate Studies

</div>

ESTABLISHING THE FOUNDATION TO ROBOTIZE COMPLEX WELDING PROCESSES THROUGH LEARNING FROM HUMAN WELDERS BASED ON DEEP LEARNING TECHNIQUES

_____

DISSERTATION
_____

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
Rui Yu
Lexington, Kentucky
Director: Dr. YuMing Zhang, Professor of Electrical and Computer Engineering
Lexington, Kentucky
2023

ABSTRACT OF DISSERTATION

ESTABLISHING THE FOUNDATION TO ROBOTIZE COMPLEX WELDING
PROCESSES THROUGH LEARNING FROM HUMAN WELDERS BASED ON DEEP
LEARNING TECHNIQUES

As the demand for customized, efficient, and high-quality production increases, traditional manufacturing processes are transforming into smart manufacturing with the aid of advancements in information technology, such as cyber-physical systems (CPS), the Internet of Things (IoT), big data, and artificial intelligence (AI). The key requirement for integration with these advanced information technologies is to digitize manufacturing processes to enable analysis, control, and interaction with other digitized components. The integration of deep learning algorithm and massive industrial data will be critical components in realizing this process, leading to enhanced manufacturing in the Future of Work at the Human-Technology Frontier (FW-HTF).

This work takes welding manufacturing as the case study to accelerate its transition to intelligent welding by robotize a complex welding process. By integrate process sensing, data visualization, deep learning-based modeling and optimization, a complex welding system is established, with the systematic solution to generalize domain-specific knowledge from experienced human welder. Such system can automatically perform complex welding processes that can only be handled by human in the past.

To enhance the system's tracking capabilities, we trained an image segmentation network to offer precise position information. We incorporated a recurrent neural network structure to analyze dynamic variations during welding. Addressing the challenge of human heterogeneity in data collection, we conducted experiments illustrating that even inaccurate datasets can effectively train deep learning models with zero mean error. Fine-tuning the model with a small portion of accurate data further elevates its performance.

KEYWORDS: Smart Manufacturing, Deep Learning, Weld penetration, Human-robot Interaction, Double-Electrode GMAW Process

Rui Yu

08/22/2023
Date

ESTABLISHING THE FOUNDATION TO ROBOTIZE COMPLEX WELDING
PROCESSES THROUGH LEARNING FROM HUMAN WELDERS BASED ON DEEP
LEARNING TECHNIQUES

By
Rui Yu

<div style="text-align:center">

Dr. Yuming Zhang
Director of Dissertation

Dr. Daniel Lau
Director of Graduate Studies

08/22/2023
Date

</div>

ACKNOWLEDGMENTS

The following dissertation, while an individual work, benefited from the insights and direction of several people. First, I sincerely thank my advisor Dr. Yuming Zhang, for his invaluable guidance, encouragement, and constructive suggestions. In addition, I am grateful to Dr. Michael T. Johnson, Dr Yuan Liao, Dr. Peng Wang, and Dr. Wei Li for their timely and instructive support in my study as my committee members, allowing me to complete this project on schedule. Next, I would like to give my thanks to all of my colleagues in the Welding Research Lab: Chao Li, Qiyue Wang, Yongchao Cheng, Wenhua Jiao and Yue Cao for their help and suggestions for my work.

Finally, I want to give my thanks to my parents for their support and endless love throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1   Background

Welding, as one of the reliable, low-cost, and precise method for material joining, is extensively utilized in the industrial manufacturing such as automobiles/aircrafts, marine, building and construction, etc. Manual welding process involves a variety of factors including materials, equipment, welding techniques, and most importantly, human intelligence decision making. A skilled human welder can control the welding process by adjusting the welding torch to ensure the quality of the welded joint base on the vision and sound information he/she obtained, especially when the unexpected disturbance happens. Those decisions are heavily relying on the welder skills, experience, and expertise, which will significantly impact the efficiency and quality of the welding process. Therefore, manual welding is always the first choice for the complex, critical and high-quality required welding tasks. These human intelligence participations difference the manual welding from the automatic welding which usually specifically designed for massive but trivial production. However, there is a practical need to transition from manual welding processes to intelligent robotic welding for several reasons: (1) the shortage of the welding labor force will reach a deficit of 400,000 workers by 2024 [1]; (2) manually welding requires welder concentrating to the weld pool which will expose the human welder to the toxic gas and fume generated during the welding process; (3) the efficiency of manual labor cannot meet the increasingly growing production demands.

While fully robotic systems have been able to replace human welders for relatively simple welding tasks, they typically have difficulties in successfully accomplishing complex tasks. Furthermore, the development of a dependable and adaptable robotic

control system is both time-consuming and financially demanding. To address this challenge, there is a need to lay the groundwork for automating complex welding tasks effectively. To robotize the complex welding processes that can only be handled by human in the past. it is essential to understand, learn, and generalize human intelligence decision-making process in welding. Understanding how human intelligence sensing the complex welding process, and generalize their understanding based on deep learning models. This generalized understanding can then be used to optimize and digitalize the welding process to make it more efficient. By automating the welding process, manufacturers can reduce the costs and improve the consistency and quality of their products. By digitalizing the welding process, real-time monitoring and analysis of the welding parameters will become possible, leading to better control of the process.

1.2   Intelligent welding

By integration the artificial intelligence (AI) and automation to enhance the efficiency, accuracy, and quality of welding processes. Intelligent welding approach leverages data-driven insights, machine learning algorithms, and robotic systems to revolutionize the way welding tasks are executed. Compare with traditional welding process which heavily rely on human expertise and labor. Intelligent welding aims to overcome these challenges by utilizing AI based algorithms with robotic systems, enabling real-time monitoring, analysis and adaptation during the welding process. The benefits of the intelligent welding can be list as the following:

Enhanced Precision: Compare with human welder, robotic system can provide more stabilized and accurate operation process, this ensures consistent and accurate welds, minimizing defects and rework.

Efficient Data Utilization: These systems can collect and process vast amounts of data from sensors, cameras, and other sources. This data can be used to identify patterns, predict potential issues, and optimize welding settings.

Adaptive Control: Intelligent welding systems can analyze welding parameters and adjust them in real-time for optimal precision. By continuously monitoring and analyzing welding conditions, intelligent welding systems can dynamically adapt to changes in the workpiece, environment, or other variables. This adaptability enhances overall process stability.

Quality Assurance: AI-driven inspection techniques can identify defects, discontinuities, or deviations from desired weld specifications. This proactive approach helps maintain high-quality welds and reduces the need for post-weld inspection.

Skill Enhancement: Intelligent welding systems can assist less experienced welders by providing real-time guidance and suggestions, ensuring consistent weld quality across various skill levels.

With above advantage, intelligent welding is transforming various industries, including manufacturing, construction, and automotive, by improving efficiency, reliability, and cost-effectiveness. As technology continues to evolve, the integration of AI and automation is expected to unlock even more possibilities for enhancing welding processes and achieving higher levels of precision and control.

1.3 Learning from human welder

Welding robot in the manufacturing increase the welding efficiency, enhance product quality, decrease the operation cost and injuries risks. However, the place utilizes the most welding robots is automotive manufacturing where the robot only needs to weld less

complex interior and exterior parts [2]. In current welding environments, a common approach involves human-robot collaboration, where human welders and welding robots collaborate on the same structure or component. In this collaboration, human welders handle tasks demanding adaptive welding skills, while robots take on tasks that require precise physical control and adherence to specific rules. Robots are preprogrammed with process settings such as arc voltage, current, torch movement, and travel speed based on prior information about the weld component's shape and specifications.

Despite these capabilities, the lack of adaptability renders welding robots unsuitable for intricate welding tasks characterized by uncertainties or complex welding processes requiring real-time adjustments based on welding dynamics. The aspiration to equip robots with adaptability is a persistent pursuit in the field of robotic welding [3-9], yet practical success in achieving this goal has been limited.

The concept of "learning from human" involves integrating insights from human cognition, sensory perception, and decision-making into the design and programming of robots. This approach can lead to the creation of robots that can adjust to changing environments, learn from experience, and engage with humans in a more intuitive and natural manner. Take complex welding tasks as an example, it's hard to decide how to perform the welding process with the robot from scratch. Learning from human provide an effective solution by predefining a majority of the welding parameters and control protocols, such parameters and control protocols serve as an initial foundation, which can be further updated as the welding system advances. Importantly, this approach draws on human expertise, endowing it with a level of flexibility beyond conventional methods. As a result, the incorporation of human insights provides a comprehensive and adaptable

4

framework for addressing complex welding requirements. Enable robots to be more adaptable in handling such tasks, thereby achieving a more efficient welding process.

To successfully learn from human and apply these insights to design and improve robotic systems. Key aspects list as follows:

Task observation and analysis: Researchers closely observe and analyze how humans execute tasks. This includes understanding the strategies, decision-making processes, sensory perception employed by humans to accomplish a specific task.

Data Collection: Data is collected from human actions and interactions, often using sensors, cameras, motion capture systems, and other technologies. This data is then used to create models that capture the patterns and dynamics of human behavior during the task.

Algorithm Development/Modeling: Develop algorithm to replicate human behaviors. Involving machine learning techniques such as deep learning or reinforcement learning, to enable robot to learn and adapt from human demonstrations.

Sensor Integration: Robots are equipped with sensors that mimic human sensory modalities, such as vision, touch, and proprioception. These sensors provide robots with the ability to perceive their environment and interact with objects and humans.

Validation and Improvement: The performance of robotic systems is validated through real-world testing and evaluation. Any discrepancies or limitations compared to human performance are addressed through iterative improvements to the algorithms, models, and sensor integration.

In conclusion, learning from human enables robots to bridge the gap between human intelligence and machine capabilities, leading to more advanced and capable robotic systems that can operate effectively in a wide range of real-world scenarios.

1.4 Objectives and Outline

This work aims to establish a scientific foundation for intelligence co-robots complex welding manufacturing in the following two aspects: (1) increasing the robot ability to perform complex weld tasks which can only be done by human in the past; (2) increasing the data analysis ability by integrating deep learning algorithms.

To robotize the complex weld tasks which has never been performed by the robot in the past, the control methodologies need to derive from human practices by closely observe and analyze how humans execute such welding processes. Furthermore, the system should also have the ability to assure the weld quality, which can be done by utilizing the state-of-the-art deep learning method or generalize the expertise from human welders on assessing welding quality.

In this dissertation, the intelligence welding system has been established based on human robot interaction (HRI) and deep learning algorithms and is organized as follows. In the Chapter 1, the purpose and motivation of this research has been introduced, In the Chapter 2, the state-of-the-art double electrode welding and its application in the wire arc additive manufacturing, intelligence welding, deep learning has been reviewed. Chapter 3 developed an image segmentation network which be used to obtain the welding pool boundaries in real-time, which will be used to provide the position information to the bypass torch to follow the main torch in the future. In the Chapter 4, a convolution long short-term memory network has been trained to predict the welding penetration in a better accuracy. Chapter 5 conducted a series of experiments to simulate how a massive but inaccurate dataset could be used to improve the training accuracy of deep learning models. This approach can address the inconsistency in the human action database. Such inaccurate dataset could be easily obtained through the IoT from different operations at different

locations in different time. Chapter 6 perform the DE-GMAW process and developed a

deep learning model classify the operational states/modes of the process. The operation

state can be used to evaluate the DE-GMAW process quality. Chapter 7 demonstrates the

DE-GMAW be executed by a tractor and robot with a stabilized bypass current. In the end,

the Chapter 8 draws the conclusion and presents the future work.

CHAPTER 2. LITERATURE REVIEW

Gas tungsten arc welding (GTAW) represent the penetration process and gas metal arc welding (GMAW) represent filling process, which are the two most common welding techniques been used in manufacturing processes. In both processes, there has an arc established between the welding electrode and the work pieces. With the heat generated by the arc accumulating, the workpieces metal will be melted and forming a liquid weld pool between two based metals, once the metal pool cools, the solid metal will join the two pieces of base metal together. To protect the welding area from atmospheric contamination, a shield gas is provided through the torch. The key difference between GTAW and GMAW lies in the electrode used: GTAW employs a non-consumable electrode, while GMAW involves a consumable wire. In this chapter, the intelligent welding, deep learning, and two electrode combined Double-Electrode GMAW process will be reviewed.

2.1   Double-Electrode GMAW Process

2.1.1   Conventional DE-GMAW process

In a regular GMAW process, as illustrated in Figure 2.1, the current from the power source passes through the wire located at the center of the welding torch and then flows to the workpieces. The heat generated by the current melts the wire, which subsequently deposits into the welded workpiece. This allows for the fusion of the materials and the formation of a strong weld joint. Therefore, the wire melting rate should be matched with the torch travel speed in a specific relationship to ensure that each unit of welding length corresponds to a certain volume of melted wire. As in the reference's [10], if the metal

8

transfer is in spray mode (melting current great than 250A), the melting speed m, melting

current I, wire extension L, and cross-section area of the wire S can be formulated by the

following equation:

$$\dot{m} = 5.1 * \frac{10^{-13}I^2 L}{S} + 2.2 * 10^{-6}I \qquad (1)$$

That means the melting current need to be increased to increase the melting speed.

However, the current goes through the wire will equal to the current which goes through

the workpieces, i.e., base metal:

$$I = I_{bm} \qquad (2)$$

as if increasing the current to expedite the melting of the feed wire inadvertently amplifies

the heat generated on the workpiece, this process will introduce a challenge that the

unnecessary heat generated on the workpieces will increasing the welding pool, leading to

workpiece deformation and the accumulation of residual stress [11]. Furthermore, GMAW

process is widely used in the wire arc additive manufacturing (WAAM) [12], where

precise heat control holds even greater importance within this construction-oriented

procedure. But limit by its process principle, decouple the wire feeding speed and the heat

generated on the workpiece is very difficult.
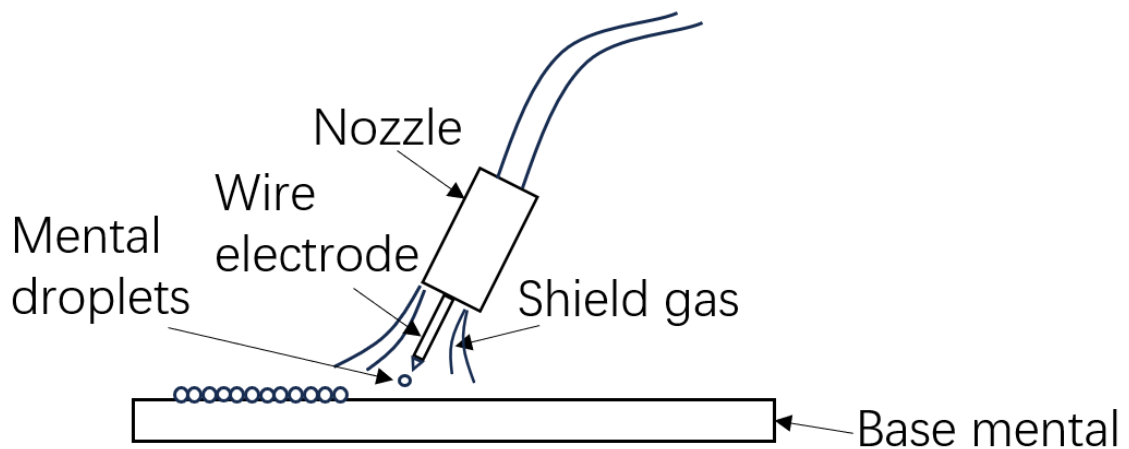
Figure 2.1 A typical GMAW process

To address this challenge, the double-electrode GMAW process has been proposed as a solution to reduce the heat input towards the workpieces during the welding process [13], as shown in Figure 2.2, this approach involves introducing a second torch alongside the main/original GMAW. The second torch position the second electrode in close proximity to the main/original arc created between the first electrode and the workpiece. This relationship allows the conductive ionized gas or plasma of the arc to establish a circuit, which the current can flow from the first electrode to the second. As a result, a bypass arc is formed as:

$$I = I_{bm} + I_{bp} \qquad (3)$$

In contrast to the conventional GMAW process (shown in Figure 2.1), this portion of the current is directed to the workpiece. This energy transfer process improves the fundamental structure of the GMAW system, i.e., the arc established between the wire and workpiece will not 100% flows through the workpiece. Consequently, decreasing the current directly reduce the heat imposed on the workpiece while maintaining the current (first electrode) that melting the wire unchanged [14]. Thus, DE-GMAW process successfully decouples the production speed and heat input.
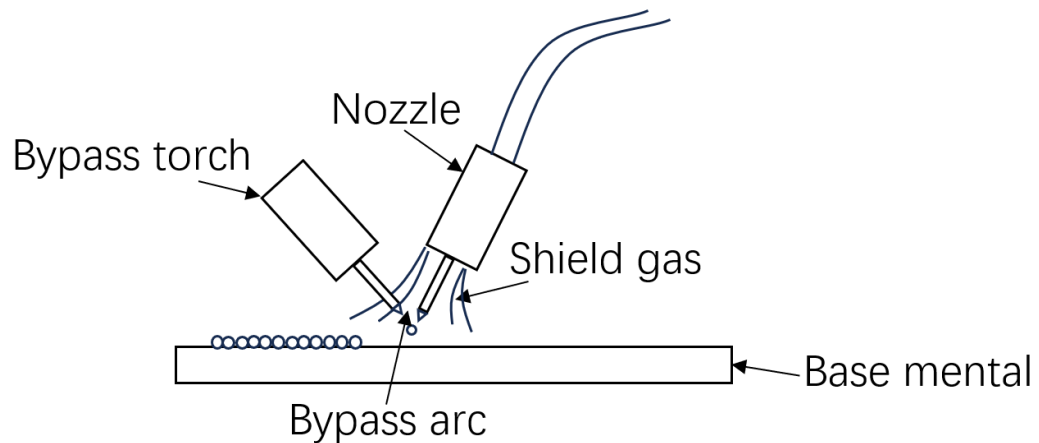
Figure 2.2 A DE-GMAW process

A study [15] use thermal behavior experimental verification that DE-GMAW does not influence weld pool depth (penetration) but can significantly reduce the weld pool area by up to 75% compared to the conventional GMAW process. As shown in Figure 2.3. The reduction in weld pool area, achieved through decreased heat input at a consistent deposition rate, results in decreased heat accumulation and lower cooling temperature. Consequently, this leads to enhanced control over geometric precision and surface quality in welded products. This distinct ability is made feasible by the presence of the bypass arc. Maintaining this crucial bypass arc necessitates the second electrode's tip to remain in close proximity to the main arc, a position that adjusts as the second electrode is fed or melted.
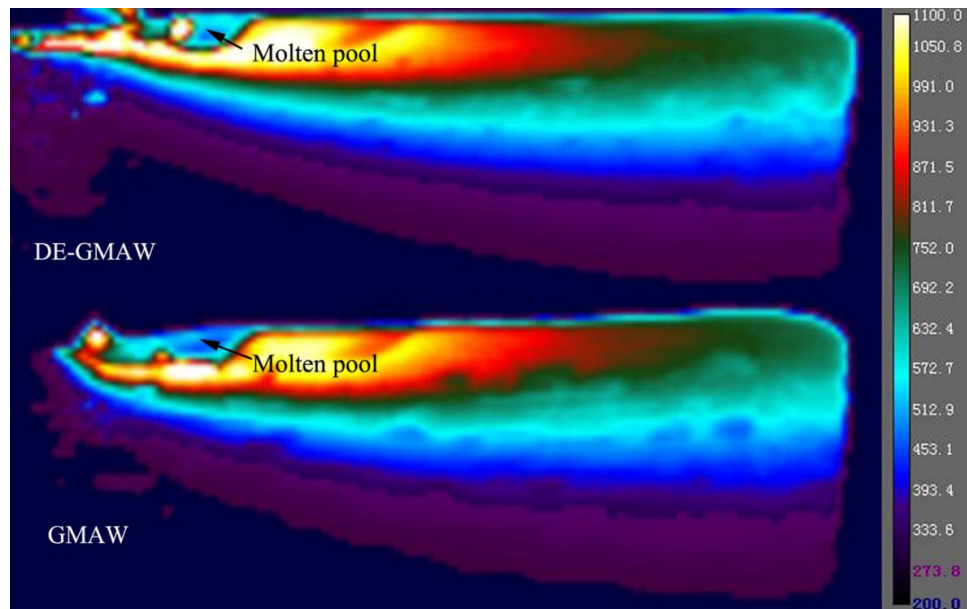
Figure 2.3 Comparison between DE-GMAW and conventional GMAW process [15]

From the above, it is obvious that the fundamental characteristic of the DE-GMAW process is the presence of the bypass arc. The stability of this bypass arc is crucial for the proper functioning of DE-GMAW. Therefore, it is essential to thoroughly investigate and understand the behavior and stability of the bypass arc. Given the unique advantages of

the DE-GMAW process, numerous researchers have explored its potential since its invention at the University of Kentucky. Based on the bypass electrode characteristics, DE-GMAW can be classified into two categories as non-consumable DE-GMAW and consumable DE-GMAW. In its initial validation in 2004 [16], the DE-GMAW process employed a plasma arc welding torch (PAW) to serve as the non-consumable second electrode. The choice of the PAW torch was aimed at facilitating the establishment of the bypass arc, capitalizing on the pilot arc's capability to reliably bridge the main arc and the second electrode. However, due to the advantages of cost and size of the PAW torch, GTAW torch replace the PAW torch in the following research. Given that tungsten electrodes exhibit greater electron emission efficiency compared to workpieces, the majority of the current supplied by the power sources flows through the bypass torch. Consequently, effective control measures are essential to sustain the desired bypass current level. In the case of the DE-GMAW system illustrated in Figure. 2.2. the behavior and stability of the bypass arc are influenced by several parameters [13] as bypass electrode, shield gas for bypass electrode, bypass electrode to feed wire distance, bypass electrode to workpiece distance, angle between bypass electrode to feed wire. Although non-consumable DE-GMAW process reduce the heat input is on the work pieces, the current flow through the bypass electrode is wasted. If replace the bypass electrode with a consumable wire, i.e., replace the GTAW torch with a GMAW torch provide the feed wire [17][18]. The waste can be eliminated while still take the advantage of the DE-GMAW providing.

Among various research studies, the control of this DE-GMAW process has been approached using different methods. Some researchers have focused on modeling the

process as a non-linear dynamic model [19], aiming to achieve control. Others have explored the application of predictive control techniques [20] to effectively manage the process. Throughout these control investigations, both non-consumable tungsten electrodes [13] and consumable GMAW wires [17] have been extensively studied as bypass electrodes, along with the investigation of various shield gases [21].

2.1.2   Robotized DE-GMAW process

However, upon reviewing all the existing studies, it is evident that all the position parameters are predetermined. If the position of the wire tip varies or the materiel to provide the bypass electrode changes, the bypass electrode would have to be re-positioned. The positioning of the bypass electrodes and main electrodes is pre-determined in each experiment means that there has no tolerant or flexibility that can adaptively adjust this process as a skilled human welder. To address this problem, an automatic robot welding system is needed to manipulate the bypass electrode. The two torches need to be handled by two separate operators. Called as leader torch/main torch and follower torch/bypass torch. To replicate the expertise of a skilled welder who utilizes raw images as input to generate control actions as output, the system should be equipped with multiple visual sensors to capture comprehensive visual information encompassing the weld pool area. During the welding operation, the main torch will execute the welding process along the predetermined weld seam path. Concurrently, the bypass torch will collaborate with the main torch, appropriately placing the second electrode in an optimal position to stabilize the bypass arc. Both operators need to dynamically adjust the torches orientation and speed. Furthermore, the crucial welding parameters, including wire feeding speed, voltage, and current, need to be adaptively controlled. Due to varying welding geometries demanding

different melting rates, based on the visual information system received, the welding power source will autonomously fine-tune these three parameters to guarantee that the melting rate aligns with the feed speed. Ideally, we desire to have a stable process where the bypass arc is sustained, However, this can only be realized under the conditions of precisely set parameters and the absence of any variations. In this ideal case, the current at the bypass electrode is primarily determined by its wire feed speed and the voltage setting, so as the current at the main wire. Thus, variations will change the relation of the dynamic relations; for example, an increase in the bypass current will increase the heat generated on the feed wire, resulting in an increased melting speed, Consequently, both torches' movement speeds must be increased to ensure that the wire deposits onto the workpiece are at the required rate per inch.

Upon capturing images, the primary torch must formulate its movement strategy by analyzing information from the perception system. For instance, the geometry characteristics of the workpieces. Meanwhile, the follower torch needs to determine how to follower the main torch, ensuring that the bypass is positioned optimally to sustain a consistent bypass arc. Follow the main torch needs to understand the image context, segment the main torch from the captured image. Position at the optimal place requires domain-specific knowledges on adaptive control rules, which skilled human welder possess.

Then the question becomes how we can segment the main troch from the image and how we can extract these control knowledges from human welders. Addressing this requires training two distinct systems, the first takes the captured image as input and generates the main torch's position as output, while the second takes welding scenarios as

input and produces corresponding actions taken by human welders as output. Training the first system requires raw welding image and associated main torch label. Training the second system requires different welding scenarios and human welders' adaptive operations. By collect those data, we can establish a knowledge base of weld scene-torch manipulation relationships. However, an essential question needs to be considered is the human heterogeneity issue, different welders might exhibit distinct preferences to deal with the identical welding scenario. To establish a standardized knowledge base for co-robotic control, the acquired relationships between welding scenes and torch manipulation must be generalized, i.e., the inaccuracy caused by human need to be addressed when training the system.

The subsequent sections of this chapter will delve into a comprehensive review of the fundamental scientific principles necessary to establish this robotized DE-GMAW system.

2.2   Intelligence Welding

Penetration level is always the most important criterion for assessing arc welding quality, as illustrated in Figure 2.4. when the welding process initiated, an electrical arc is generated between the workpieces and the welding torch, serving as a heat source to melt the base metal and create the weld pool. In certain applications, additional filler metal may also be added to melt with the base metal to enhance mechanical and metallurgical properties. Based on the melting area, the create weld pool will have three stages as (1) incomplete penetration; (2) complete penetration; (3) over penetration. During the stage (1), the melt area is insufficient and the weld pool depth d_ω is smaller than the workpieces thickness t. If the heat input is terminated at this stage, incomplete fusion will occur on

the back side of the joint when the molten pool solidifies. This incomplete fusion can have detrimental effects on the mechanical properties of the joint. During the stage (2), the weld pool continue grows until its backside bead width (BSBW) w_b hits a specific width w_0 which is pre-decided by t. If the heat input terminated at this stage, after solidification, the joint will have an optimal quality. During the stage (3), the size of the weld pool continues to increase beyond the limits that can be supported by surface tension. As a result, the molten metal drop and leaves a hole on the workpieces. This occurrence is considered as a welding failure and is unacceptable.
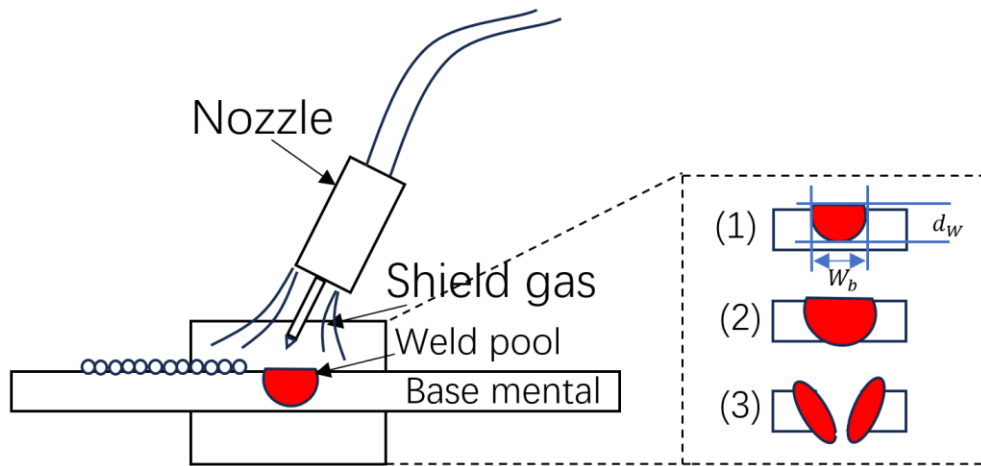
Figure 2.4 Three state during the welding process

Therefore, the real-time monitoring of the backside bead shape width (BSBW) is crucial during the welding process. By continuously tracking the BSBW, it becomes possible to achieve the optimal control point that minimizes the w_b to reduce the risk of burn-through while ensuring that the w_b are greater than the baseline width w_0 to meet the strength requirements. However, due to the limited physical accessibility, it is usually inconvenient to set up a sensor to view the backside of the workpieces directly during the real manufacturing process. Hence, a lot of indirect sensing method has been developed to

estimate the BSBW from the information monitored by the sensor installed at the top side of the weld pool.

The difficulty of measuring the backside bead width from the topside is that the penetration process is occurring underneath the weld pool which is not directly visible [22], the luminous of the arc and the fume from welding further increase the complexity of the welding pool's view [23]. To contest this problem, various methods have been proposed. These include pool oscillation [24-26], which utilizes the dynamic changes in the surface of the weld pool to identify the weld penetration state; infrared camera, which analyzes the heat distribution to assess the progress of the welding process [27], detect defects [28], and control the process [29]; and ultrasonic testing [30], which employs ultrasonic waves to penetrate the liquid weld pool and reflect off the solid metal at the bottom side of the weld pool.

Efficient analysis methods and appropriate pre-processing algorithms are crucial in extracting relevant features from the collected raw information. These extracted features exhibit correlations with the penetration state, thereby aiding in accurate assessment and analysis. Through the Fourier transform, the fluctuation of the weld pool surface/weld pool oscillation caused by arc voltage fluctuation can be analyzed. This fluctuation is closely related to the dimensions and penetration state of the weld pool [31]. With one infrared sensing image, weld pools' shape as width and height are easily to be defined and obtained through some classic edge detection algorithms such as Canny [32], peak temperature and weld pool's area under certain temperature are also readily available, with those features, non-liner model as neural network [33] or adaptive neuro fuzzy inference system (ANFIS) could be established with the reconstruct temperature distribution field [34]. With a series

of infrared sensing image, a cooling curve can be derived over a specific time period, since different welding joint states, such as those with large porosity or incomplete fusion, exhibit distinct cooling rates [35]. This cooling curve could then provide valuable information for identifying welding defect. Ultrasonic-based sensing employs a direct method by measuring the time of flight (TOF) of ultrasonic waves between a generator and a receiver. The TOF exhibits a linear relationship with the depths of the weld pool, allowing for the identification of weld penetration status based on the variations in TOF [36][37]. Arc sensing method analyzing the periodic transformation of the arc voltage during the GTAW process, which is closely related to the penetration state [38].

Passive vision sensing approach view the phenomena occurring on the heating side and then drive what occurs underneath. With appropriate optical filter, a camera system can serve as a human welder and control the weld quality based on the observed view from the weld pool [39]. However, this control quality is heavily related to the image quality the system obtained. To further image the weld pool clearly, studies have been proposed which imaging the weld pool area during specific events such as the arc extinguishing during short-circuiting transfer in gas metal arc welding (GMAW) [40], or when the arc current reaches a low base level resulting in weakened arc brightness [41]. Denoising techniques have also been studied, either filtering raw images [42], or inpainting images from strong process interferences [43]. Active vision sensing employs a laser generator to project a dot matrix or grid on the weld pool [44], the molten weld pool act as a mirror to reflect the project laser to an image plane, the camera has been synchronized with the laser to capture the image on the plane when the laser on the peak power, this peak power can reach tens of kilowatts for an illumination laser which is much brighter that the weld arc.

Consequently, a distinct dot/grid reflection image can be obtained, accurately representing the shape of the weld pool in three-dimensional space. Which leads to a precise predict of weld penetration state. With the development of the camera technology, high dynamic range (HDR) camera can provide views close to human view to machines [45]. As rich information HDR image contains, the conventional handcraft feature extracting approach cannot guarantee the hidden information been utilized effectively, therefore, deep learning models that automatically extract information propose a promising solution for the future study [46-49].

Based on the above analysis process, the obtained states, i.e., penetration level are then utilized for the development of control algorithms, which are then implemented to control the power source providing minimum heat input for full penetration. By optimizing the heat input and ensuring full penetration, the welding process can meet the necessary criteria for achieving desired weld quality and structural integrity.

2.3   Deep Learning

Deep learning is a learning method that has evolved from conventional neural networks, different from the shallow network which usually only have fewer hidden layers. Deep learning models are characterized by having a larger number of hidden layers, which allows them to learn complex patterns and representations from the data. In deep learning, the depth of the network, achieved through stacking multiple hidden layers, on the one hand, is one of the key factors that contribute to its success in handling tasks such as image recognition [50], natural language processing [51], and speech recognition [52]. On the other hand, adding more layers to a neural network can introduce certain challenges and considerations in terms of network training and performance: (1) vanishing or exploding

gradients problem as when the network becomes deeper, the gradients used for updating the network weights can diminish or explode, leading to slow or unstable training [53]; (2) overfitting was more likely to happen when the model has a large number of parameters to be trained [54]; (3) with the increased size of the network model and training data, it requires more computational resources and training time, making them more computationally expensive compared to shallow networks [55]. Fortunately, With the recent advancements in deep learning, many of the aforementioned challenges have been effectively addressed: (1) Using activation functions [56] and employing normalization techniques [57] can help alleviate the vanishing or exploding issue; (2) regularization techniques such as dropout [58], L1 or L2 regularization [59], early stopping, and data augmentation [60][61] can be used to reduce overfitting. In addition, start training the new model by transfer training [62] from an old model or use pre-trained models on large datasets can help improve generalization. (3) specific designed hardware like GPU can provide high-performance computation resources. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are two representative models commonly used for handling grid-shaped data and sequential data, respectively. In the subsequent sections, a brief overview of these models will be provided.

2.3.1   Convolution Neural Networks

The first CNN, proposed by Kunihiko Fukushima, revolutionized the field of computer vision and has since become a fundamental model for processing grid-shaped data. CNNs have been widely applied in various domains, including image recognition, object detection [63], and image segmentation [64]. By leveraging the inherent spatial structure of grid-shaped data, CNNs can effectively capture local patterns and hierarchical

20

features, making them highly suitable for analyzing and understanding visual information. The skeleton of the network architecture is illustrated in Figure. 2.5, taking ResNet [65] as an example. Other network models, such as U-net, share a similar structure to ResNet but may differ in the number of layers and the inclusion of elements like normalization and activation functions. The specific structure always depends on the specific task the model needs to handle.

Input
↓
| 7*7 conv 64 |
↓
| 3*3 conv 64 |
↓
| 3*3 conv 64 |
↓
| 3*3 conv 64 |
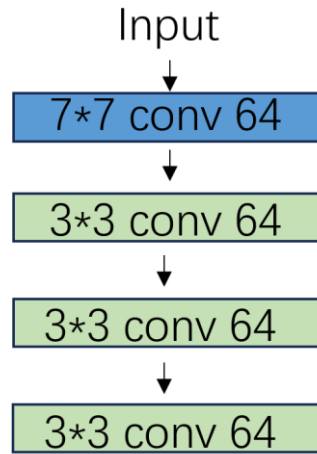
Figure 2.5 A part of ResNet structure

CNNs are composed of three types of layers: convolutional layers, pooling layers, and fully connected layers. Taking images or any grid-like data as input, convolution operations are applied to extract feature maps by convolving the feature maps from the previous layer with small-sized kernels that shift across the input, this process can be represented by the following equation:

$$F_o = \sigma(\Sigma_{i=1}^{n} W_i * F_i + b) \tag{4}$$

Where F_o represent the output feature map, σ denotes the activation function, n is the number of input feature maps, W_i represents the kernel weights for the i-th input feature map, F_i represents the i-th input feature map and b represents the bias term. The convolution operation (*) applies the kernel weights to the corresponding spatial locations

of the input feature map and sums them up, followed by adding the bias term and applying the activation function to obtain the output feature map. In conventional image processing algorithms like edge detection or Gaussian blur, convolution is also involved. However, unlike the convolution process in deep CNNs, the kernels used in these algorithms are fixed and not as adaptable or learnable as in CNNs. Activation function σ applied in order to endow nonlinearity to the network, Common activation function include ReLU, Sigmoid, Tanh, etc. After the convolution process, in most of the CNNs model, it will be followed by a pooling layer, which reduce the spatial dimensionality of the feature maps and capture the most important information while maintaining local invariance:

$$Output[i, j, k] = Pooling(Input[P * i: P * (i + 1), P * j: P * (j + 1), k]) \qquad (5)$$

Where $Output[i, j, k]$ represent the output value at position (i, j, k) in the pooled feature map, (i, j) denote the position in the 2-dimensional matrix and k denote which channel. Pooling layers play a crucial role in reducing the number of trainable parameters and the computational complexity of the model. By aggregating information within local regions, pooling layers effectively downsample the feature maps, thereby decreasing the spatial dimensionality of the data. This reduction in dimensionality not only reduces the computational burden but also increases the robustness of the model by reducing the impact of local information or disturbances on the final decision-making process. In this way, pooling layers help to capture the most salient features while maintaining a certain level of spatial invariance and improving the efficiency and stability of the algorithm.

As the convolution process continues, the feature maps become smaller. Some distant but related features will come closer together as the feature maps shrink. Eventually, they can be associated by a convolutional kernel. Upon extracting features from the convolutional and pooling layers, the fully connected layers play a crucial role in

performing reasoning tasks by establishing relationships between these features and the desired targets. The outputs of the fully connected layers are computed through a matrix product operation as:

$$y = WF + b \tag{6}$$

where the learned weights are multiplied with the input features to produce the final output. y represents the output vector, W denotes the weight matrix, F represents the input vector, b denotes the bias vector. This process allows the network to capture complex relationships and make predictions based on the learned representations. The fully connected layers enable end-to-end learning and enable the network to generate meaningful outputs based on the extracted features from earlier layers. After obtaining the prediction from the last fully connected layer, the model needs to be updated based on the calculate lose:

$$L(\theta) = \frac{1}{N} \sum_{i=0}^{N-1} loss(y_i, o_i) \tag{7}$$

Here, L(θ) represent the average loss function, N is the total number of training samples, y_i is the predicted output for the i-th sample, o_i is the true label or target value for the i-th sample, loss is the loss function that measures the discrepancy between the predicted and true values. By iteratively updating the parameters θ using gradient descent or other optimization algorithms, the model aims to minimize the average loss and improve its prediction performance on the training data. Usually, iterative gradient-descent method called backpropagation is most commonly used optimization algorithm, the first step is to Calculate the gradient of the loss function with respect to the output layer activations:

$$\frac{\delta L}{\delta a} = \frac{\delta loss}{\delta a} \tag{8}$$

Then, propagate the gradients backward through the layers using the chain rule:

$$\frac{\delta L}{\delta z} = \frac{\delta L}{\delta a} * \frac{\delta a}{\delta z} \tag{9}$$

Next, compute the gradients of the loss function with respect to the parameters in each layer:

$$\frac{\delta loss}{\delta W} = \frac{\delta L}{\delta z} * (activation \ of \ previous \ layer) \tag{10}$$

Finally, Update the parameters:

$$W_{new} = W_{old} - \alpha * \left(\frac{\delta loss}{\delta W}\right) \tag{11}$$

Here, α represent learning rate. This backpropagation and parameter updates process will continue for all layers in the network. The specific equations for the gradients depend on the activation functions, loss function, and network architecture used in the CNN. The backpropagation algorithm efficiently computes these gradients, allowing the network to learn and adjust its parameters during the training process. Some variant optimizers have also been developed to accelerate the training processing such as Adagrad [66], Adadelta [67], and Adam [68]. In the practical applications, limited by the memory, only a subset of data is used for each iteration, which is known as a minibatch. The size of the data subset used for gradient computation is referred to as the minibatch size.

2.3.2   Recurrent Neural Networks

Different from CNNs, RNNs are designed to handle sequential or time series data [69]. They are widely used in various tasks such as natural language processing (NLP), speech recognition, and image captioning [70]. One key characteristic of RNNs is their ability to utilize information from previous inputs to influence the processing of current inputs and generate corresponding outputs. This memory feature enables RNNs to effectively capture the temporal dependencies in the data. A fully connected RNNs

typically has three layers as input layer, hidden layer, and output layer. The input layer receives the input for the neural network, processes it, and transfers it to the hidden layer. The hidden layers are interconnected and use a consistent activation function with varying weights. Each hidden neuron retains its own state and utilizes this state to produce its output and update its hidden weights. It passes the output to the output layer and hidden weights to the next hidden neuron. Figure 2.6illustrates a typical RNN structure, where sequential inputs $x_{(1,2,...,t)}$ are processed, and the model maintains distinct hidden states $h_{(1,2,...,t)}$:

$$h_t = f(W_{hh} * h_{t-1} + W_{hx} * x_t + b_h) \tag{12}$$

the hidden state at time step t is obtained from activation f, weight matrix for the recurrent connection of the hidden state W_{hh} and weight matrix for the input connection to the hidden state W_{hx}. Each hidden neuron generates its own output $y_{(1,2,...,t)}$:

$$y_t = g(W_{yh} * h_t + b_y) \tag{13}$$

and updates its hidden weight $w_{(1,2,...,t)}$, forwarding these parameters to the corresponding next position in the sequence.
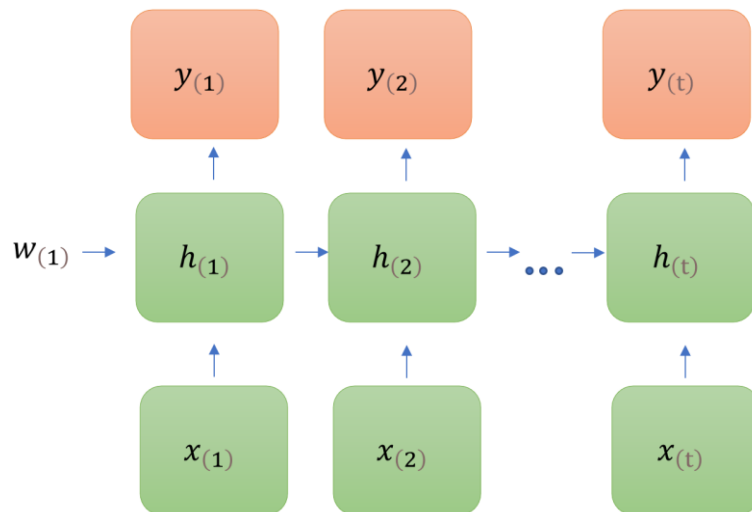
Figure 2.6 A conventional RNN structure

There also has a lot of variation model been developed based on typical RNNs structure. Such as long short-term memory (LSTM) [71]. Difference from traditional RNNs which suffer from the vanishing gradient problem, makes it difficult to capture and propagate information over long sequences. LSTM addresses this issue by introducing a memory cell and several specialized gates that control the flow of information. This enables LSTM to effectively capture and retain information over longer time steps. With its ability to capture long-term dependencies and handle variable-length sequences, LSTM is particularly effective in modeling complex relationships in sequential data. It has been successfully applied in various domains. Gated recurrent unit (GRU) is another variation [72], it simplifies the architecture through combining the forget and input gates into a single "update gate" and merges the memory cell and output gate into a "reset gate." This simplification reduces the number of parameters and computational complexity while still maintaining the ability to capture long-term dependencies. By reducing the parameter in the network, GRU is computationally more efficient compared to RNNs because it has fewer gating mechanisms. This makes GRU particularly useful in scenarios where computational resources are limited.

2.4   Human-Robot Interaction

Welding is a prominent application for industrial robots due to its repetitive nature and requirement for precision. In sectors like automotive, aerospace, and heavy machinery manufacturing, robots are widely deployed for diverse welding processes. This utilization brings benefits such as heightened productivity, enhanced quality, improved safety, and cost savings. Robots' adoption continues to expand with advancements in robotics technology, sensors, and artificial intelligence. This progress enables robots to offer

greater capabilities and cater to a broader range of industries. However, due to the difficulty of developing an artificial intelligence (AI) based control algorithm, most of the welding robots are pre-programmed through on-line teaching, or off-lane programming. Thus, most of the robot can only perform repeated tasks in a production line with strict requirements regarding a consistent working environment.

As a comparison, skilled human welder can perform high quality welding despite of working environment, by precisely adjusting the welding torch movement after perceiving, analyzing and integrating information from welding processes [73]. This is the reason why high requirements welding application are still performed by skill human welder. Moreover, human can adapt the new requirement much quicker and endow them more flexibility to handle the small volume of the welding task with less time and finance costing. However, welding process as a high energy process will produce hazardous fume, gas and arc radiation endanger weld labor's health. Furthermore, compared to welding robots, human welders exhibit poorer performance in precise, repetitive, and prolonged control of welding torch movement. Additionally, there's a pressing global shortage of qualified welders, as highlighted by the American Welding Society (AWS) [1]. Analyzing this, it's evident that both humans and robots possess distinct strengths and weaknesses. Human-robot interaction (HRI) emerges as a potential solution to address this challenge.

Through the HRI framework, three aspects of the application has been proposed:(1) telerobotic welding [74][75]: Compare with human, robot can sustain more extreme environments conditions such as temperature, pressure, vacuum, remotely utilize robot as extensions of human welders. This significantly expands the scope of human welders' work boundaries, enabling unstructured welding tasks to be accomplished in extreme

environments like space, oceans, or chemical leak; (2) manual welding with robot assistance [76]: Human welders collaborate in the same welding process alongside robots. Robotic assistance reduces the burdensome and physically demanding aspects for humans while mitigating sudden and disruptive movements that could affect the welding process. This approach leverages both human intelligence and robotic stability to enhance production efficiency and effectiveness; (3) Welding robots' demonstration-based imitation learning [77][78]: Human welders serve as mentors, demonstrating their skills to robots acting as apprentices. By capturing state-action pairs, a control policy can be developed for welding robots, replicating the expertise of human welders. After robots generalized human welder's expertise, they can finish the welding tasks independently.

For all three aspects of application, the information communication between the human and the robot serve as a critical link. A lot if interface method has been proposed to send human manipulation operation to the robot such as voice [79], gestures [80], haptic gloves [81]. However, these interface does not satisfy human's intuitive operation behavior. The welder needs to receive some training process to learning how to convey their control operation to the robot. Without equipment interfaces, accurately translating human commands to robot actions becomes challenging. For instance, describing precise angle adjustments solely through voice commands is difficult. Additionally, equipment interfaces can impose physical limitations on workers or impact their performance due to design or workload constraints. Ideally, information communication should have no adverse impact on human welders. An optimal solution could involve capturing on-site welder information through multiple sensors, providing comprehensive data for reconstructing human welder operations. However, this reconstruction process

28

necessitates the robot's ability to comprehend and interpret this data effectively. Virtual reality (VR) can be considered as another acceptable solution [82]. By replicate every component involved in the welding process to a digital environment, human welder can wear a VR headset and perform the welding process in the virtual world as real world, their intuitive action can then be obtained and transmit to the robot in the real world, perform the weld tasks as the human do in the virtual one. A system has been established in the University of Kentucky [83], In such VR-based HRI systems, humans teleoperate the robots via the control handles and receive the onsite scenes from the head-mount display (HMD) as the visual feedback.

CHAPTER 3. REAL-TIME RECONITION OF ARC WELD POOL USING IMAGE
SEGMENTATION NETWORK

3.1   Introduction

As previously discussed, the DE-GMAW process requires two torches manipulate
by two separate operators. Called as lead torch and follower torch, the follower torch needs
to monitor the weld process and identify the lead torch's position. However, suffered by
the intense arc radiation/illumination and unpredictable human welder operation, it is
challenging to design a static feature pattern to consistently detect the weld pool
boundary/position under different welding conditions as a conventional method. Existing
studies that extract the complete boundary of the weld pool during automated welding
require prior knowledge of the weld pool position. Moreover, these studies often utilize a
pulsed laser with a higher peak power compared to the arc, enabling easier visualization
of the weld pool despite the presence of the arc. There is a lack of studies in the literature
that employ a passive vision approach to detect the entire weld pool during welding
without the need for stronger external illumination. Deep learning network provide a more
promising solution to such a challenge. The difficulty to apply the deep learning approach
is that training deep learning models requires a large amount of data. In classification
problems, automated methods can be used to calculate the labels for each image. However,
in the case of image segmentation, specifically in the context of detecting the weld pool,
automatic generation of labels is not possible. If it were possible, the problem could have
been solved using conventional image processing techniques. To address this challenge,
we propose the use of a lightweight CNN structure, adapted from U-net [84], for real-time
extraction of the weld pool boundary. This modified model benefits from its small
parameter count, requiring only a few manually labeled weld pool images for training and

validation. Furthermore, the compact model size also meets the requirements for later real-time deployment.

3.2   System and Data Preparation

The experimental setup for the GTAW welding system used in this study is illustrated in Figure 3.1. It comprises a Miller Maxstar 210 power source, a welding torch equipped with a 2.4 mm diameter tungsten electrode, a motor-driven conveyor for workpiece movement, and a pure argon shield gas with 99% purity. To capture the weld pool surface images during the welding process, a high dynamic range (HDR) camera, specifically the Xiris XVC-1000, is positioned at the topside of the workpieces, Figure 3.2 shows the example weld pool images captured from the top side. Both the weld power and motor are controlled by a computer via the PCI-6229 National Instruments data acquisition card. The motor is regulated by the moving speed v and moving time t_1, while the weld power is adjusted through the weld current I and weld time t_2. The HDR camera is controlled by the computer separately.
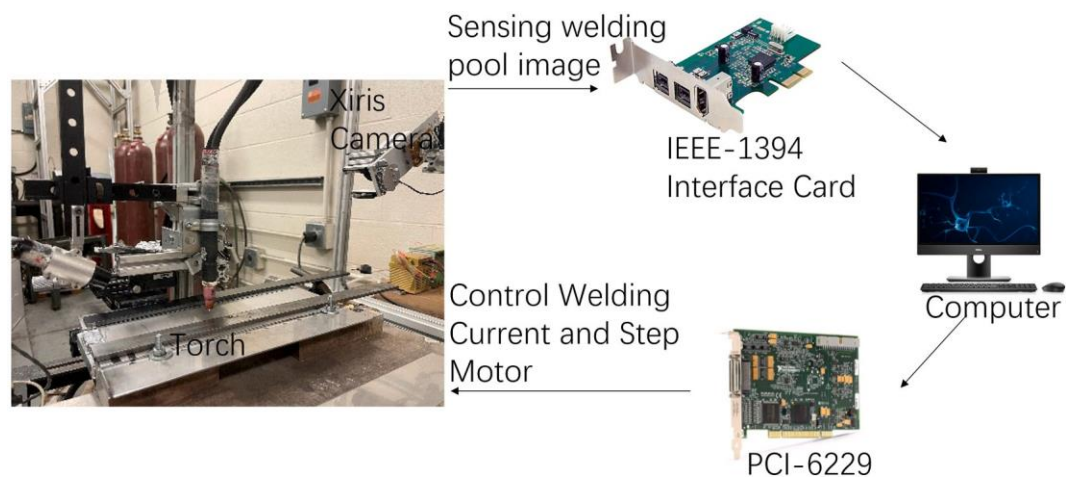
Figure 3.1 Experiment platform
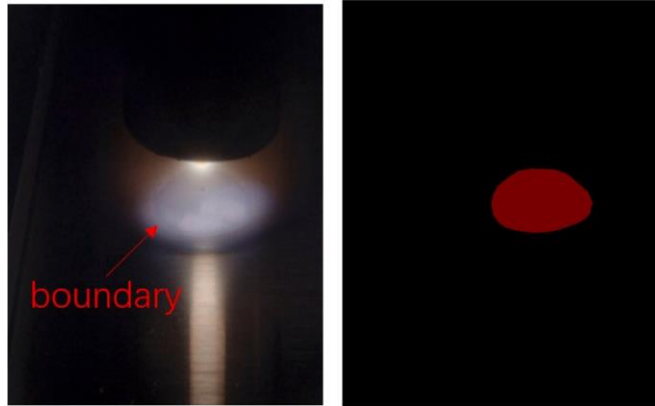
Figure 3.2 Topside image

To simulate different welding and arc conditions, both the welding current and motor speed vary randomly, shown in Figure 3.3, to generate different heat input, The welding current varies randomly between 90 A and 130 A and the workpiece moving speed, also randomly, varies between 1.4 mm/s and 2 mm/s. For one experiment lasted 80 seconds, both welding current and workpieces moving speed alter every 2 seconds, which means 40 combinations will be applied, generate 40 different heat inputs. Thus, based on different heat input, various weld pool boundaries/shapes are generated under various welding conditions. The weld pool, being a liquid puddle, possesses distinct visual characteristics compared to the surrounding solid or solidified metal. However, the identification of the weld pool in images is often hindered by the intense interference caused by the strong arc radiation. The clarity of the weld pool boundary observed in the images is influenced by various welding conditions, especially the welding current. Furthermore, challenges arise from the high temperatures involved and the presence of specular surfaces, which further complicate the identification process.

Figure 3.4 shows images acquired under different welding currents. It is obvious that although human can recognize the boundary under such different circumstances, the changes in brightness and reflection make its challenge to be detected automatically.

<center>(a)   (b)</center>
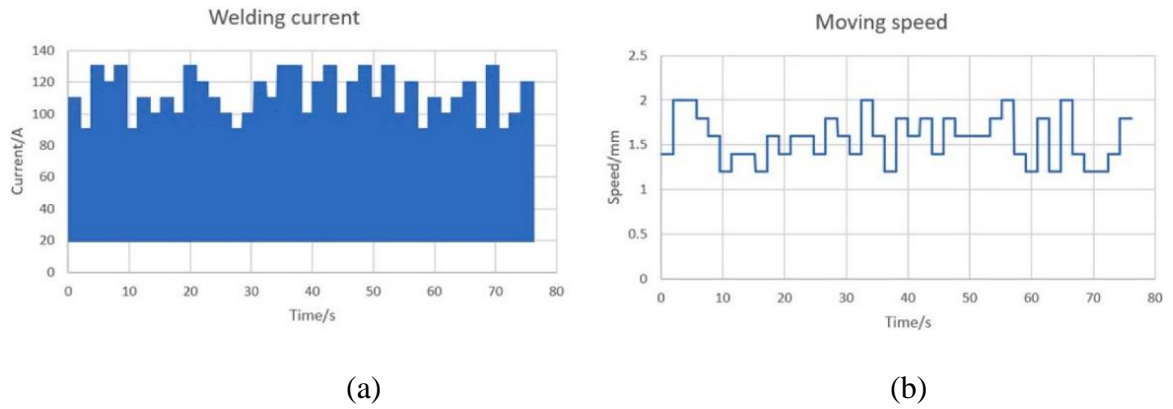
Figure 3.3 (a) Welding current; (b) Platform moving speed
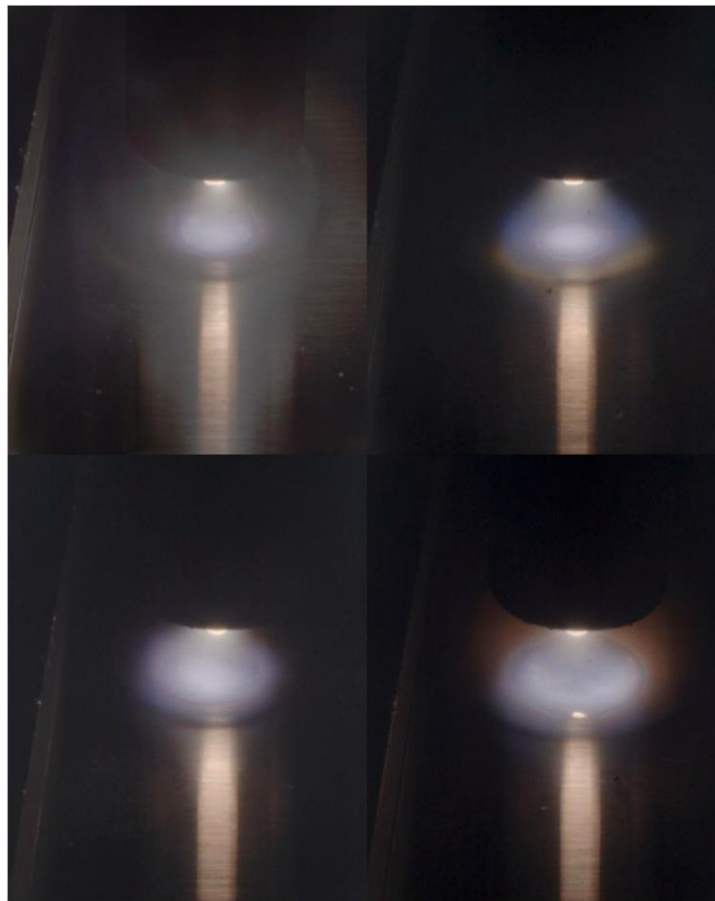
Figure 3.4 Captured image under different welding currents

A total of eight experiments were designed and conducted using GTAW, with the welding parameters detailed in Table 3.1. The parameters related to the HDR camera used to capture the weld pool images, including frame rate, sharpness, gamma, shutter, and gain,

are specified in Table 3.2. As mentioned earlier, by applying various heat inputs, we have

obtained diverse states of the weld pool and different weld pool boundaries, resulting in a

rich and extensive dataset. Training with this dataset will ensure that the results perform

effectively across different welding conditions.

Table 3.1 Welding parameters

Materials	Thickness (mm)	Arc Length (mm)	Gas Flow Rate (L/min)
Stainless Steel 304 L	1.85	4.8	7

Table 3.2 Camera parameters

Sharpness	Gamma	Shutter	Framerate	Gain
3000 (A)	2.5 (A)	5 ms (T) 0.2 ms (B)	60 fps (A)	0 dB (A)

To accurately calibrate a specific weld pool position, an annotation process was

conducted by manually labeling the weld pool over hundreds of the captured images.

Those images were randomly selected from the diverse dataset we obtained above, to

minimize the artificial influence. The labeling results are illustrated in Figure 3.5. The

accuracy of the calibration result can be appreciated from Figure 3.6. Such annotation

process has been made doable through using the LabelMe [85].

With the original image served as input and annotated image serve as output, the

dataset construction is complete, the size of the training set, validation set, and testing set

are 170, 21, and 22 respectively. Benefit from the unique capabilities from the U-net. Such

a small dataset is enough to finish the training process without overfitting occur. The

validation set can help to optimize the hyperparameters and model structure, in the end,

model's efficiency will be evaluated based on the testing set.
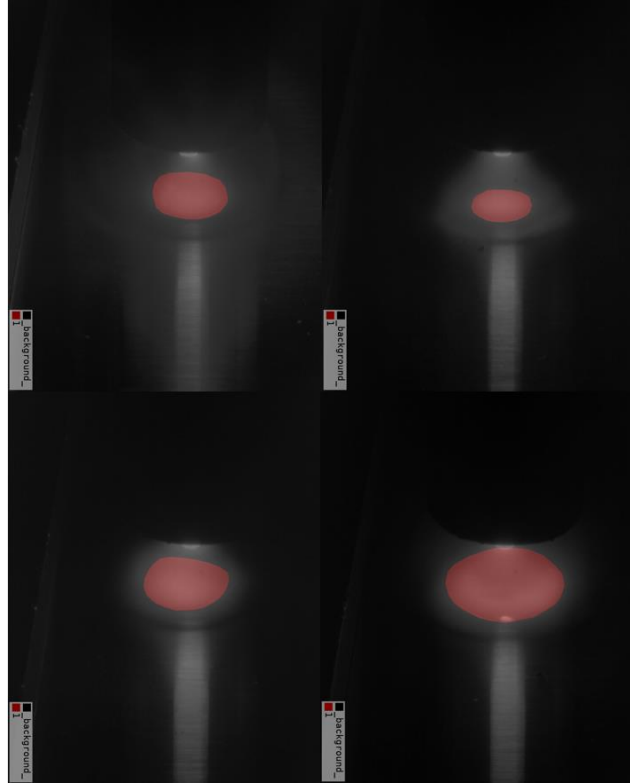
Figure 3.5 The labeled image
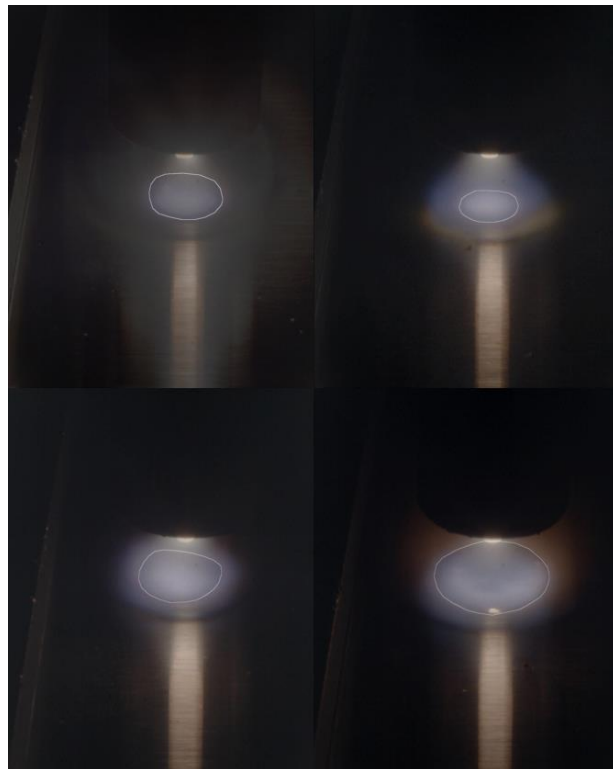
Figure 3.6 Boundary of the labeled weld pool

3.3   Model Structure and Training

3.3.1   Model Structure

The structure of developed convolutional neural network is shown in Figure 3.7. It has modified from a U-net architecture that consists of a contracting path and an expansive path, but with a modification to further reduce the model complexity so that the training is more efficient and actual computation during application is reduced due to two reason: (1) the training data in these specific applications is manually labeled which means it's very limited; (2) when compared to natural images, welding images exhibit similarities in many cases. They generally have a lower density of content and information and adhere to a fixed structure, making the extraction of semantic information relatively straightforward. For example, the weld pool is typically located in the central area of the image (assuming the camera is correctly positioned) and appears as the brightest element in the overall view.
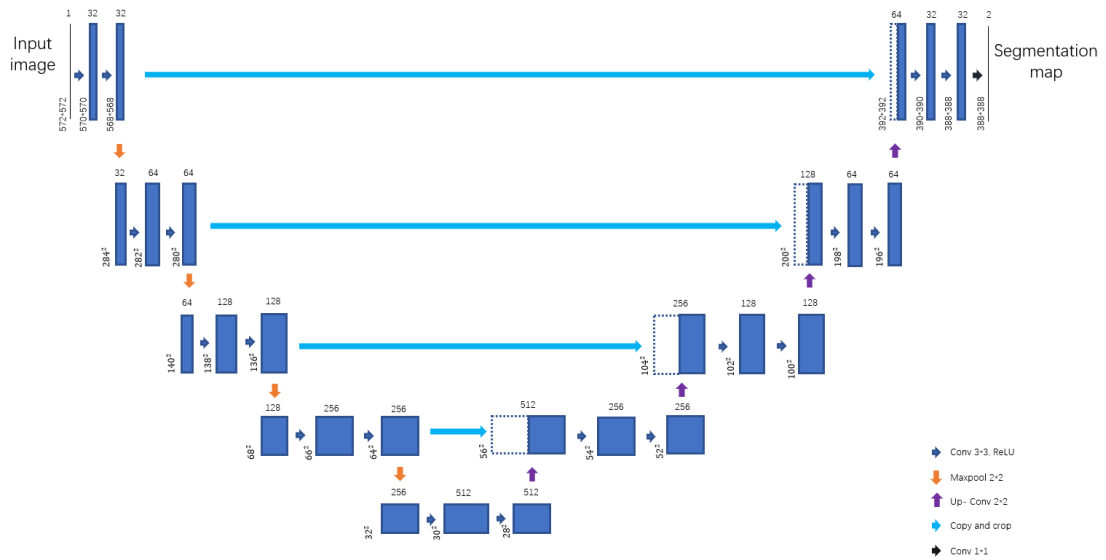
Figure 3.7 CNN architecture

In the contracting path:

$$C_k = conv(C_{k-1}) \; \forall k \in [1, N] \tag{14}$$

36

the image undergoes a 3×3 convolution process, which is repeated twice. Each convolution is followed by a rectified linear unit (ReLU) activation function and a 2×2 max pooling operation with a stride of 2, allowing for down-sampling and doubling the number of channels. In the expansive path:

$$E_k = conv(E_{k-1}, D_k) \ \forall k \in [N-1, 1] \tag{15}$$

each step involves up-sampling the feature map, reducing the number of feature channels by half. This is followed by concatenating:

$$M_k = concat(C_k, E_k) \tag{16}$$

the up-sampled feature map with the corresponding cropped feature map from the contracting path, i.e., M_k represents the merged feature map at the k-th stage, combine the feature map from both paths. Two 3×3 convolutions are then applied:

$$Output = conv(M_n) \tag{17}$$

each followed by a ReLU activation function to generate final output. Overall, the network consists of a total of 23 convolutional layers. The cropping process is necessary to account for the loss of border pixels during each convolution operation.

Compared to other instance segmentation networks like FCN and Deeplab [93], U-Net offers two significant advantages that are particularly beneficial for our task. Firstly, since there is no existing labeled welding database available, we need to manually obtain the necessary labels. This limits the availability of labels, requiring us to consider models with a smaller number of parameters. The U-Net network we used has relatively fewer parameters, after we modified the structure, the model only has 7.55 million compared to a standard FCN network which has around 57 million parameters [93]. Secondly, U-Net employs a unique architecture with four times up-sampling. It consists of both a contracting and expansive path connected through a skip connection. This differs from many other

networks that directly supervise and backpropagate the loss on high-level semantic features. The U-Net's architecture ensures the integration of more low-level features and enables the fusion of features at different scales. With four times up-sampling, the segmentation map can recover more refined information, such as edges, resulting in a multi-scale prediction with deep supervision at different scales.

3.3.2   Training Process

We conducted the training, validation, and testing of the CNN model using a computer equipped with an Intel® Core™ i9-10900K CPU, 32GB of RAM, and an NVIDIA GTX 2080 GPU. The training process was carried out using the PyTorch library and the stochastic gradient descent algorithm in a Python programming environment.

As mentioned earlier, our dataset consists of 213 images, which were divided into three sets: training, validation, and testing. The sizes of these sets were 170, 21, and 22, respectively. To ensure a homogeneous distribution of the dataset, we employed the stratified random sampling method, drawing samples randomly while maintaining representative distributions in each set. The only difference between the sets was their sizes.

We trained the model for 2000 iterations, and the results of accuracy and loss on the training and validation datasets are presented in Figure 3.8. The accuracy quickly increased and then reached a stable level of around 99%, while the loss rapidly decreased and then stabilized at around 1%. These results indicate that the model achieved high accuracy and effectively minimized the loss during the training process.
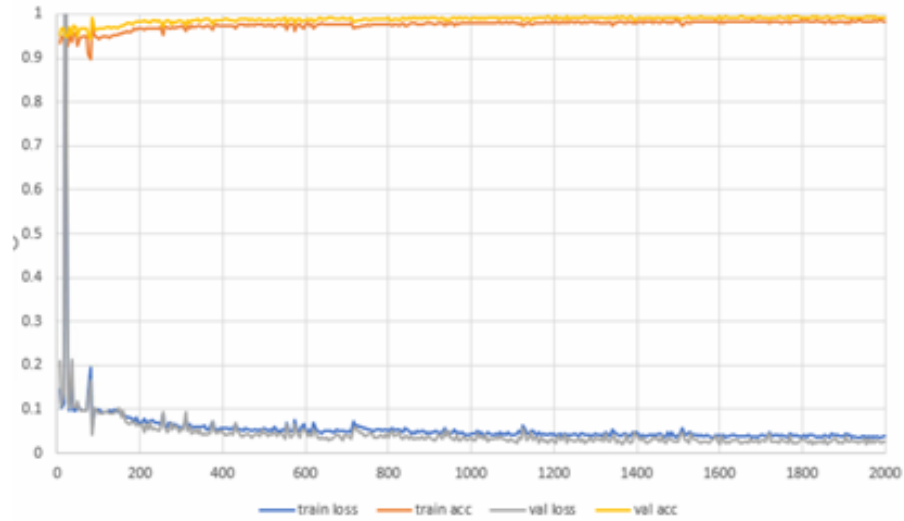
Figure 3.8 The accuracy and loss of the training and validation process

3.4   Result and Summary

After the completion of training, we evaluated the model on the epoch with the minimum validation loss. Figure 3.9 illustrates the input images as left and corresponding output images as middle generated by the model for different welding conditions, including variations in welding currents, welding speeds, weld pool sizes, and weld pool boundary shapes. In the output images, the detected weld pool pixels are represented by dark areas.
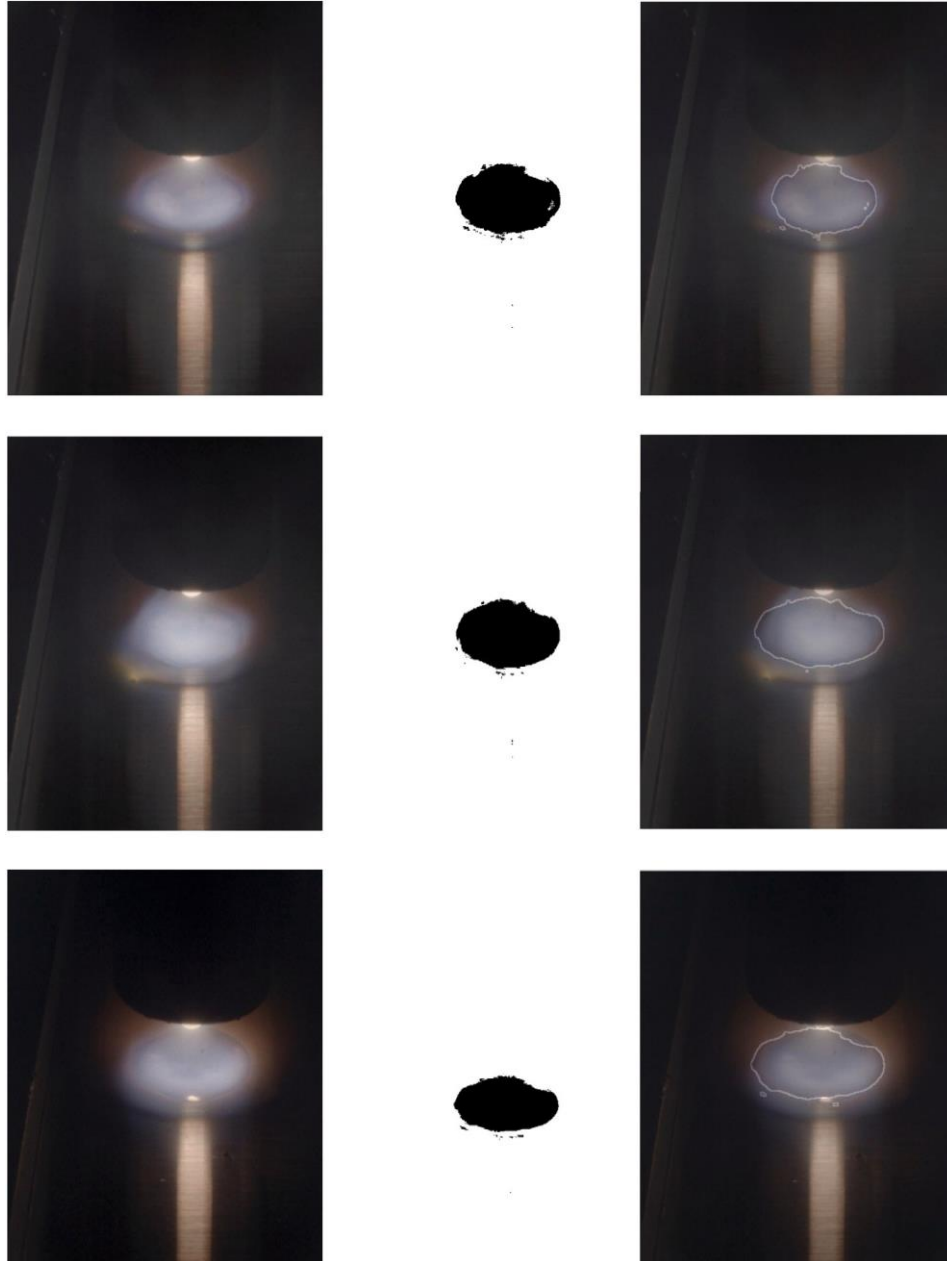
Figure 3.9 Detection results. Left: Input images; Middle: Output images; Right: Illustration of boundary of detected pool

To further analyze the results, we applied a Laplace Gaussian filter to the dark area

in the middle image to enhance the detection of the weld pool boundary/edge. The resulting

boundary/edge detection was then compared with the raw image shown on the right. It is

evident that the trained U-Net CNN model successfully detects the weld pool boundary for all the tested images obtained under various welding conditions/states.

With the hardware mentioned in the training process, the output network model is capable of processing images at a speed of over 10 frames per second. This speed is sufficient for real-time monitoring of a GTAW process, which is relatively slow compared to other welding processes. This high processing speed is especially important for precision welding applications where timely monitoring and control are crucial.

CHAPTER 4. MONITING THE WELD PENETRATION FROM DYNAMIC WELD
POOL IMAGES WITH CNN-LSTM MODEL

4.1   Introduction

As mentioned in the previous chapter, weld joint penetration is a key parameter for

assessing weld integrity and it's quantified by the width of the weld on the backside of the

workpiece, i.e., BSBW, the problem is that it is difficult to monitor the BSBW by installing

the backside sensor directly since its usually inconvenient and has limited physical

accessibility. Therefore, the BSBW needs to be estimated indirectly from the available

information sensed by the topside sensors.

Some studies have been done for this BSBW estimation via various information

source. For instance, temperature distribution [28], weld pool image [86], arc voltage [87]

are all been used by different researchers to estimate the BSBW. Most of these studies

predefine the features need to be extracted from the information source and then fed those

features into a machine learning model such as support vector machine [88] or neural

network model to establish a correlation between the features and the penetration state.

Since the feature is predefined by the artificial process, those method has two inherent

issues that cannot be overcome (1) the predefined artificial features heavily rely on the

personal experience of the researchers, making them unsuitable for real-world

manufacturing applications; (2) the process of defining features is labor-intensive, as new

features need to be predefined for different information sources. This limitation prevents

the utilization of massive data generated by intelligent manufacturing.

Hence, there is a need for a more robust approach that overcomes these issues and

enables accurate estimation of the BSBW in real-manufacturing scenarios. Li proposes to

address this problem by using reflect laser pattern to estimate penetration state during the

welding process through deep learning approach. With CNNs to extract the features from the sensed weld images automatically, this work can be solved in two steps (1) obtaining the input image and corresponding label as much as possible under experiment setup, I.e., setting a backside camera to visualize the BSBW as corresponding label; (2) Training and applying the obtained CNNs model in real-world manufacturing, i.e., only need topside image, can remove the backside camera. Following Li's research, several studies have focused to use CNNs as a major type of deep learning model to extract relevant features automatically from weld pool images to predict the weld penetration through training from experimental data [44]. However, most studies are experimental and mathematical ignoring the basic physical ground if the weld pool images contain adequate raw information.

Weld pool images are typically captured from the surface of the workpiece. However, it is important to note that the weld pool is a three-dimensional entity, and the penetration is determined by the bottom of the weld pool beneath the workpiece. This raises a question about whether a single image of the visible weld pool surface provides enough information to accurately assess what is happening underneath. Upon analysis in the following section, it becomes apparent that relying solely on a single image of the weld pool surface is unlikely to be sufficient, especially when the weld pool is undergoing dynamic changes. In such cases, it may be necessary to consider sequential weld pool images to obtain a more comprehensive understanding of the welding process.

4.2   Principles

Figure 5.1 shows the gas tungsten arc welding (GTAW) process, which is extensively utilized for precise joining applications. Unlike the gas metal arc welding (GMAW)

process, GTAW allows for precise control of welding current, arc power, and arc heat. Additionally, the figure illustrates a fully penetrated weld pool, characterized by the backside width w_b and height h_b of the liquid pool, which serves as a representation of its state.

The backside width w_b is a crucial parameter that determines the state of the weld. It is essential for w_b to be greater than zero $w_b > 0$ because a value of $w_b = 0$ indicates partial penetration, which can result in an explosion under high temperature and pressure conditions. Additionally, considering the seam tracking error $e \geq 0$, which represents the deviation of the weld's symmetrical axis from the interface of the metals being joined (as shown by the dotted line in Figure 4.1(b)), if $\frac{w_b}{2} < e$, the interface may not be fully melted, leading to incomplete penetration. Therefore, ensuring w_b is sufficient and accounting for potential tracking errors are critical factors in achieving complete penetration during welding.
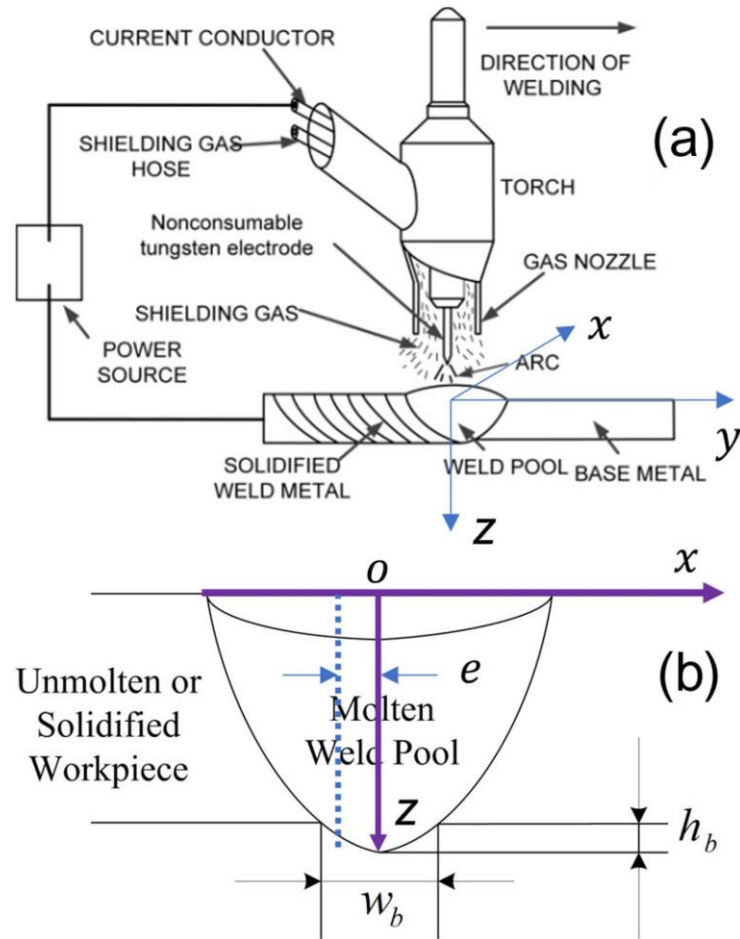
Figure 4.1 GTAW and complete weld penetration. (a) Illustration of GTAW process; (b) Cross section of the workpiece and weld. Axis z is that of the tungsten electrode and the weld is in general symmetrical about it. The dot-line in (b) is the interface of the two metals being joined and e is the seam tracking error

To ensure the desired complete penetration during welding, it is important to determine an appropriate backside bead width w_b denoted as w_b^*. In general, a larger w_b^* provides better assurance of complete penetration. However, increasing w_b also leads to higher heat input, resulting in greater distortion and residual stress. Therefore, w_b^* should be minimized while still ensuring complete penetration. The minimization of w_b^* depends on the specific application, which determines the maximum allowable weld seam tracking error e_{max}. It is necessary to ensure that $2e_{max} \leq w_b$, which translates to $2e_{max} \leq w_b^*$. The achievable accuracy in monitoring and controlling w_b is also a factor. Let ε represent the

45

estimation/control error between the estimated backside bead width w_p and the actual w_b, such that $w_b = w_p + \varepsilon$. Therefore, the condition becomes $2e_{max} + \varepsilon \leq w_p$. If we denote ε_{max} as the maximum allowable estimation/control error $\varepsilon_{max} \geq |\varepsilon|$, then the condition becomes $2e_{max} + \varepsilon_{max} \leq w_b^*$. It is evident that precise monitoring and control of the weld penetration involve minimizing ε_{max}, which minimizes the heat input while ensuring complete penetration, i.e., fully melting the interface throughout the entire thickness direction.
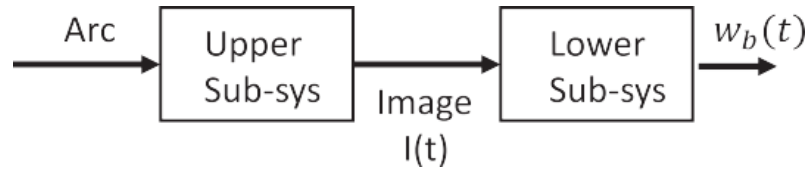
Figure 4.2 Dynamic development of the penetration

During the GTAW process, the workpiece is melted in a sequential manner. Initially, the arc heats the surface of the workpiece, and then the heat is transferred in three dimensions within the workpiece, including along the thickness direction, to deepen the weld pool. To better understand this penetrating process, it can be artificially divided into two subsystems: the upper subsystem and the lower subsystem, as shown in Figure 4.2 the upper subsystem takes the arc $a(t)$ and the upper weld pool (represented by the image $I(t)$) as inputs and outputs. On the other side, the lower subsystem takes the output $I(t)$ as the input and measured bottom surface $w_b(t)$ as output. Due to the complex heat transfer and phenomena occurring in the weld pool, the relationship between $w_b(t)$ and $I(\tau)$, where $\tau \leq t$, can be expressed as $w_b(t) = f(I(\tau))$. Thus, using a single image $I(t)$ to estimate $w_b(t)$ is an approximation. The accuracy of this approximation depends on the magnitude and rapidity of the dynamic changes in $I(\tau)$. To obtain more comprehensive and potentially critical information, we consider using a sequence of images $I(\tau)$, where

46

$t - \Delta t \leq \tau \leq t$ and $\Delta t > 0$. Utilizing this time window provides more raw information for the estimation of $w_b(t)$. However, an excessively large Δt may not be necessary and can increase computational and modeling complexity. To address this, we will employ a CNN-LSTM (Convolutional Neural Network-Long Short-Term Memory) network to correlate $I(\tau)$, where $t - \Delta t \leq \tau \leq t$, with $w_b(t)$, and compare its accuracy with a CNNs that only correlates $I(t)$ with $w_b(t)$. To accomplish this, we will generate pairs of data $[I(k), w_b(k)]$, where k represents discrete-time instances.

4.3 Data generation

The platform introduce in the chapter 3 has been used to perform the designed GTAW welding experiments. The experiments use two cameras while one camera captured images $I_b(k)$ of the backside of the workpiece, which were used to calculate $w_b(k)$, and the other high dynamic range (HDR) camera captured images $I(k)$ of the weld pool. The recordings were synchronized to form pairs of $[I(k), I_b(k)]$. The cameras and welding torch were fixed, while the workpiece was moved using a linear motion system.

In the experiments, the welding current and travel speed were varied every 2 seconds within the ranges of $[80A, 130A]$ and $[1.4mm/s, 2.0mm/s]$, respectively. These variations were random and aimed to create a moving arc on a 1.8 mm thick stainless-steel workpiece. The welding current and travel speed together determined the heat input, which represents the input imposed on the workpiece per unit length of weld. The relationship between the current and heat input is proportional to the arc pressure, which significantly influences the flow of liquid metal in the weld pool. The heat transfer within the pool is proportional to the square of the current I^2, where I represents the current.

47

Welding current and tractor travel speed were varied during the experiments to introduce dynamic evolution of the weld pool into the dataset. Maintaining constant values for both parameters would result in a mostly steady-state weld pool, making it difficult to observe the effect of $I(m < k)$ on $w_b(k)$. In this study, six experiments were conducted, with each experiment running for approximately 80 seconds. Figure 4.3 shows the variations in current and travel speed observed in one experiment. While the values in other experiments are similar, they are not identical as they represent different realizations of the random processes. Figure 4.4 presents a series of $[I(k), I_b(k)]$ pairs captured at a frame rate of 60 frames per second, with one-second intervals between each image. For segmentation, each image I_b is binarized using a specific threshold, dividing it into bright and dark regions. The width of the bright region is then calculated as w_b, considering the calibrated threshold and the pixel-to-millimeter resolution.
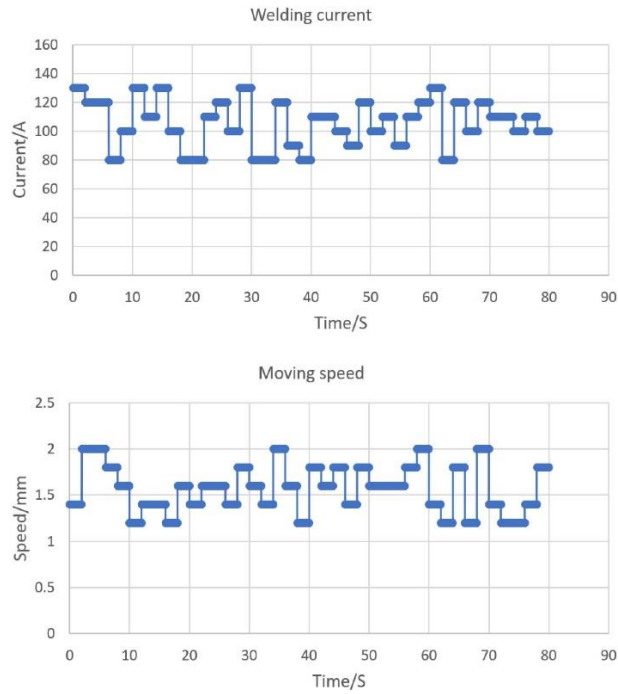
Figure 4.3 Welding current and travel speed from one experiment

Since the images were sampled at a rate of 60 Hz, a total of 28,560 $[I(k), w_b(k)]$ pairs were obtained. To showcase the dynamic changes, Figure 5.5 displays the variations in w_b from two different experiments.
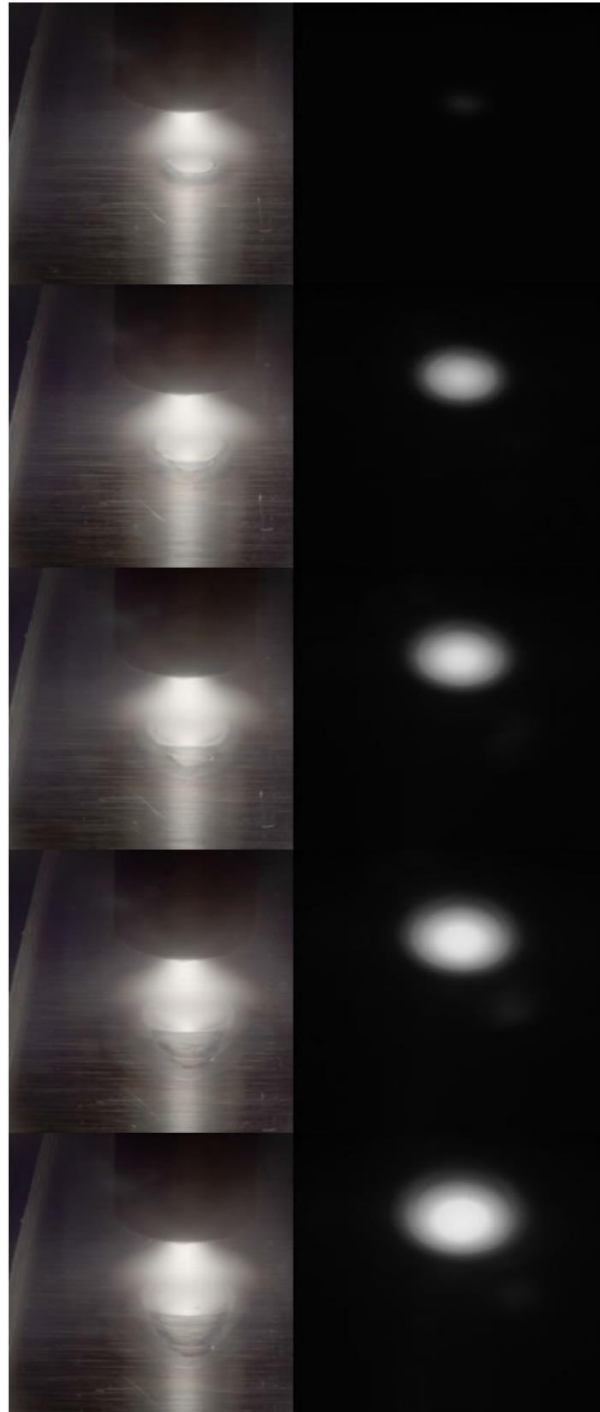
Figure 4.4 Five second sequence of image pairs
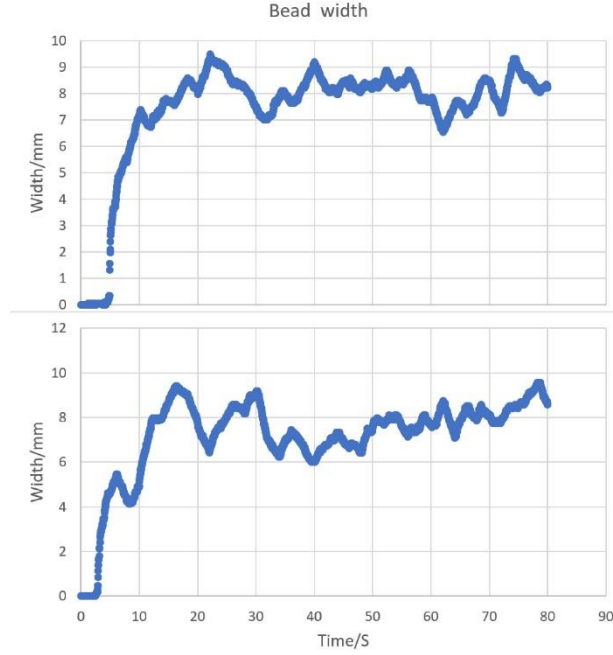
Figure 4.5 Backside bead width variation in two experiments

4.4　Network structure

The architecture of the comparative CNNs is illustrate in Figure 4.6, which predicts $w_b(k)$ based on a single image $I(k)$. The proposed CNN-LSTM architecture, shown in Figure X, predicts $w_b(k)$ using sequential images $I(m)$ $(m \leq k)$. Both architectures have identical CNN components with trainable parameters.

Each input image, with a size of 256×256 pixels, is passed through the CNN via a series of convolution layers followed by pooling layers. This process is repeated four times, with the convolution layer parameters set as (1, 32, 5, 2, 2), (32, 64, 3, 2, 1), (64, 128, 3, 2, 1), and (128, 256, 3, 2, 1) respectively. Batch normalization and ReLU activation are applied between each convolution and pooling layer. After the convolution process, the input image is transformed into a 1×256 feature vector, denoted as $V(k)$, which serves as the input for the subsequent fully connected layer to predict $w_b(k)$.
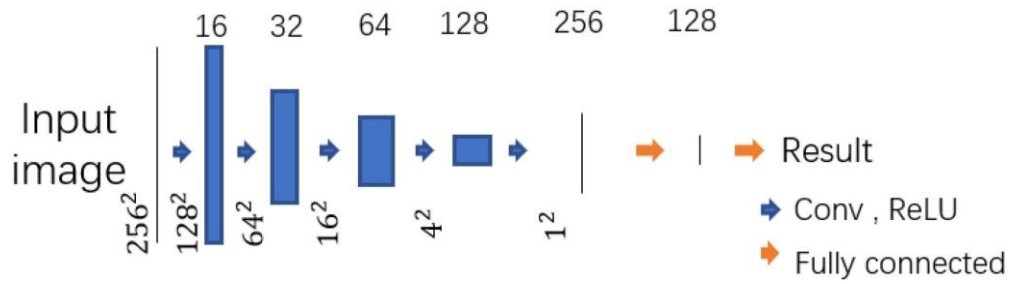
Figure 4.6 CNN architecture for comparative model

LSTM, a distinctive type of artificial recurrent neural network (RNN) architecture [XX] that incorporates feedback connections, has demonstrated superior performance in generating outputs from sequential data. It has outperformed conventional dynamic prediction methods such as hidden Markov models in various evaluations [89]. Based on this, it is reasonable to propose that integrating LSTM with CNN can offer an effective approach to automatically extract essential dynamic and abstract features from sequential weld pool images, enabling the prediction of weld penetration occurring beneath the workpiece.

In the CNN-LSTM model proposed in Figure 4.7, the feature vector $V(k)$ is connected to the LSTM model instead of the fully connected layer. The input size of the LSTM model, therefore, is set to 256. In our study, we initially experimented with a hidden size of 128, a num_layers value of 3, and a sequence length of 8 to assess if we could achieve satisfactory results. Since the hidden state in the RNN incorporates information from previous states, the preceding states can influence the current state. This implies that earlier images, $I(m < k)$, can contribute to predicting the current $w_b(k)$.
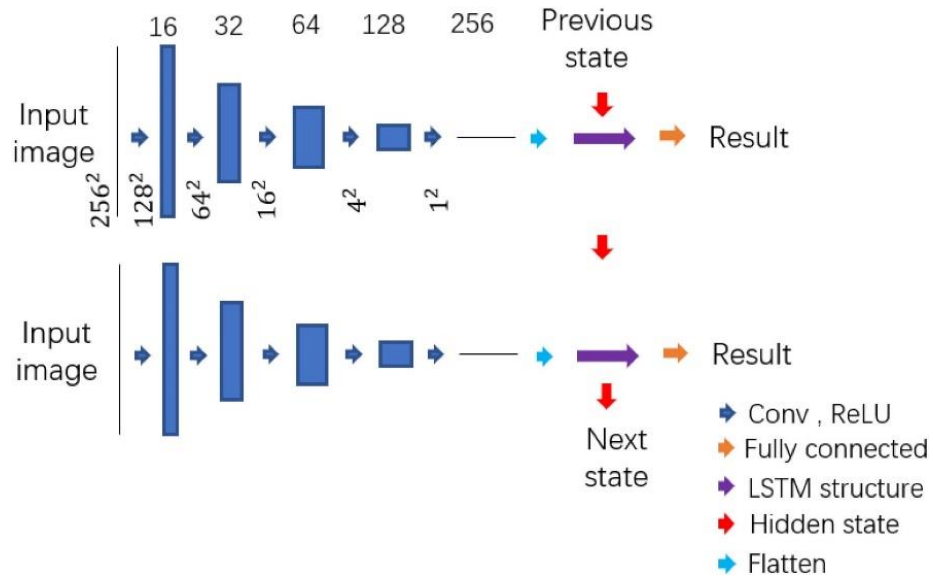
Figure 4.7 Proposed CNN-LSTM model structure

Given that we captured images at a rate of 60 frames per second and set the batch size as 32, each batch encompassed 1.875 images per second. With a sequence length of 8, the CNN-LSTM model utilizes the feature vector V as far back as 4 seconds ago. Therefore, the CNN-LSTM model employs $V(k - j)$ where $j \leq 32 * 7 = 224$ as input, which corresponds to the image series $(k), I(k - 33), \ldots, I(k - 224)$, to predict $w_b(k)$. The chosen 4-second timeframe represents an approximate settling time for the GTAW process, as illustrate in Figure X of reference [XX].

4.5　Training and discussion

We perform the training, validation, and testing process on a NVIDIA GTX 2080 graphic card. Both models were trained iteratively 100 times with SGD optimizer and mean-square loss under Python environment with Pytorch library.

Out of the six experiments conducted, four were allocated for training, while the remaining two were reserved for validation and testing, respectively. All experiments were

52

performed under the same nominal welding conditions; however, due to the random variations in welding parameters, each experiment exhibited unique characteristics. Consequently, there was no requirement for random data selection, and any experiment could be utilized for training, validation, or testing purposes.

The dataset of 28,560 paired images was split into three subsets: training, validation, and testing, with sizes of 19,040, 4,760, and 4,760, respectively. To train the CNN model, the dataset was shuffled to ensure random sampling. For training the CNN-LSTM model, the data was pre-arranged into multiple sequences, each containing 8 consecutive images with a frame difference of 32. For example, one sequence would contain frame 1, frame 33, frame 65, and so on. This arrangement ensured that the images fed into the CNN-LSTM model had time correlation with each other, and each frame followed the previous frame in the sequence after 0.53 seconds.

Figure 4.8 displays the loss results for both the training and validation datasets of both networks. Clearly, the model that incorporates LSTM shows a significant reduction in validation error compared to the CNN model without LSTM, using the mean square error loss (MSE). Specifically, the CNN model achieved a minimum validation error of 0.6 mm^2, while the CNN-LSTM model achieved a significantly lower validation error of 0.34 mm^2, representing a 43 percent reduction. This reduction is crucial for achieving accurate monitoring in the precision joining process targeted in this work.

The network was trained using the mean squared error as the loss function. While an absolute error-based loss might be more logically reasonable, mean squared error-based losses offer analytical gradients, which lead to more efficient solutions. Moreover, absolute error-based losses minimize the mean of the absolute errors, but they may not

address the maximal absolute error ε_{max}. Considering the objective of this study, which focuses on proposing dynamic image series to analyze physical processes and ensure the sufficiency of raw information, the specific choice of network, loss function, and learning algorithm should not impact the main goal of verifying the idea using dynamic images to ensure the adequacy of raw information.
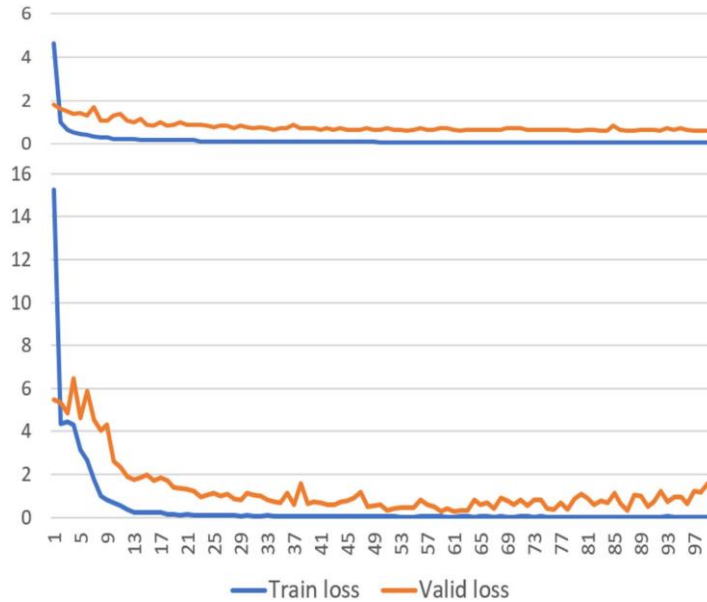
Figure 4.8 CNN training curve (Top) and CNN-LSTM training curve (Bottom)

Both models were selected based on their respective minimum validation error epochs for testing. Figure 4.9 shows the prediction results for the test experiment. The average error for the CNN model was $0.54\ mm$, whereas it was $0.3\ mm$ for the CNN-LSTM model. This demonstrates a 44 percent reduction in the prediction error, which is contributed by adding LSTM to the CNN.

In particular, it is evident that the CNN model struggled in the initial stages of welding, where the weld pool experiences rapid and dynamic changes. However, the CNN-LSTM model exhibited a fundamentally different behavior, successfully tracking the swift increase in weld penetration. This accomplishment is highly challenging, and only a few

studies have reported success in predicting weld penetration under such drastic and dynamic changes. Moreover, throughout the entire range of the experiment, where the dynamics exhibited various levels of randomness, the CNN-LSTM model consistently delivered outstanding performance. The achieved 0.3 mm error is remarkably accurate for a welding process that involves a broad arc heat distribution and numerous factors affecting the heat transfer to the bottom of the workpiece.
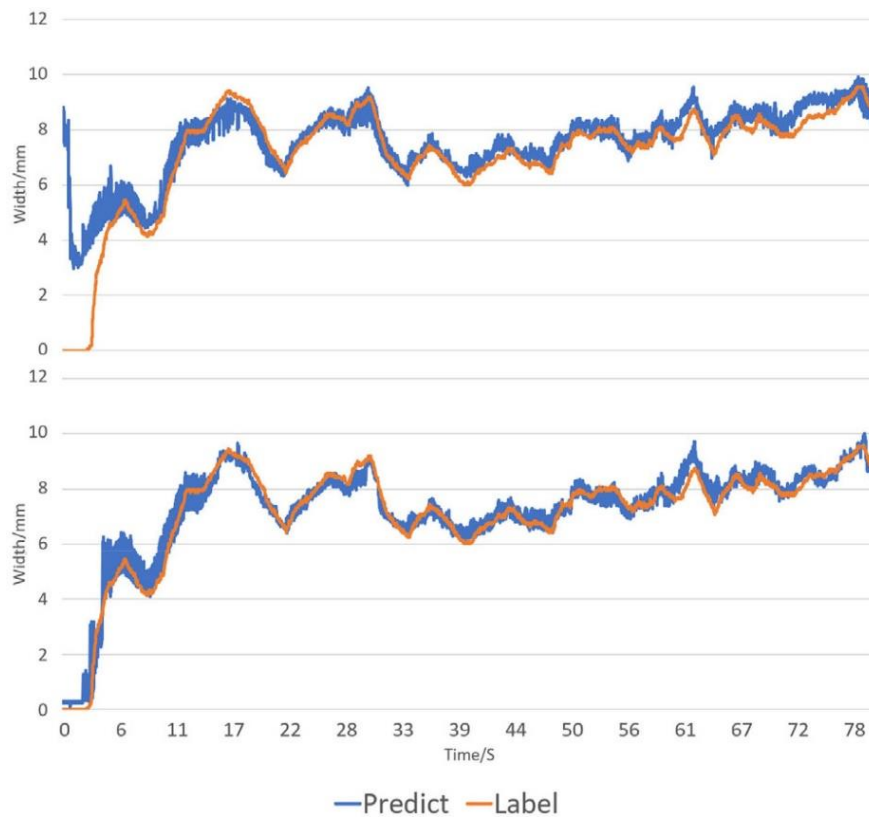
Figure 4.9 CNN (Top) and CNN-LSTM (Bottom) performance on test dataset

We were unable to find many reports that directly compare the actual and predicted weld penetration using a deep learning approach, similar to our results shown in Figure 4.9. However, in Figure 4.10, a previous effort used a CNN model to estimate the growth of weld penetration for a stationary weld pool without relative motion between the arc and workpiece. Figure 4.11 presents the results from other previous efforts that used hand-

crafted features from moving weld pools. In (a), the 3D surface of the weld pool was measured using an innovative laser dot-matrix projection on the mirror-like specular surface, and the 3D surface was characterized by length, width, and a novel key parameter called "convexity" [91]. (a) compares the actual and predicted weld penetration using the 3D weld pool surface characteristic parameters. In (b), a series of weld pool images were used as input to estimate the back-side width of the weld. However, in this approach, the weld pool images did not directly employ deep learning for automatically extracting abstract features. Instead, hand-crafted features were proposed to represent the weld pool boundary, and the resultant features from the most recent 5 seconds were used to estimate the back-side weld width using a neural network. As seen in these previous efforts, their accuracies were significantly lower than what we achieved in this study. Notably, (a) demonstrated that a single image, even from an accurate 3D surface, is insufficient, and (b) illustrated the challenges of manual handcrafting to extract the necessary information. Thus, both deep learning for automatic feature extraction and serial images containing adequate information are crucial, and the CNN-LSTM model serves as a powerful mathematical surrogate for achieving these objectives.
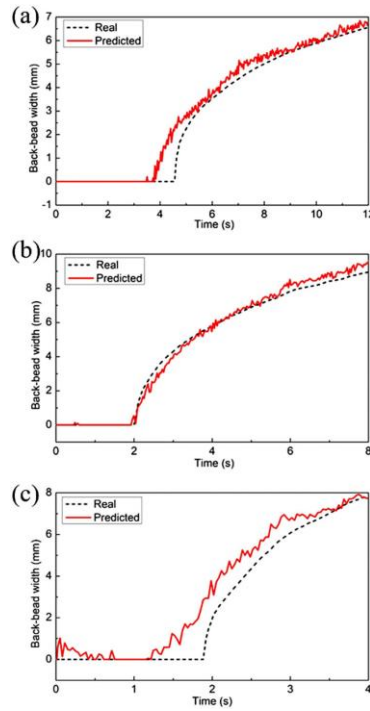
Figure 4.10 Comparative efforts in predicting the weld penetration at different welding current using CNN [90]; (a) 60 A; (b) 83 A; (c) 110A.
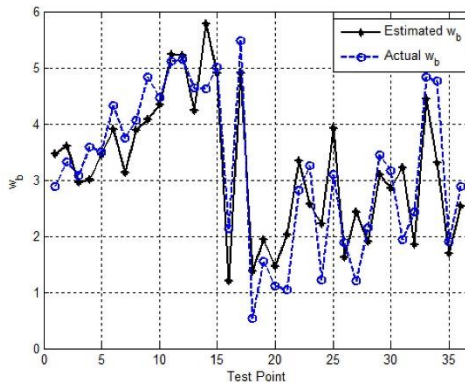
(a)   Prediction using 3D weld pool surface as characterized by width, length, and convexity of the liquid weld pool surface [27]
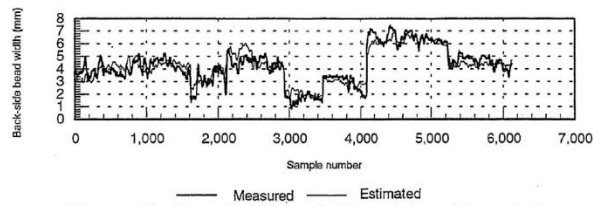
(b)   Prediction using dynamic 2D weld pool boundaries as input of a neural network [2]

Figure 4.11 Comparative efforts in predicting the weld penetration based on hand-crafted features [91]

4.6   Conclusion

This study utilized randomly varying waveforms for welding current and travel speed to generate dynamic weld pools. It revealed that a series of weld pool images taken in the most recent 5 seconds contains sufficient raw information to accurately estimate the back-side bead, indicating the degree of full penetration. However, relying on a single weld pool image at the current time, especially when the weld pool undergoes rapid and dynamic changes, is not sufficient. To address this, a CNN-LSTM model was employed to extract relevant dynamic features from the sequential images, enabling accurate determination of rapidly changing weld penetration. The contributions of the CNN-based deep learning model and the serial images were separately verified through comparative studies. The resulting CNN-LSTM model successfully estimated the back-side weld bead with a remarkable accuracy of $0.3\ mm$, based on serial weld pool images captured by an HDR camera, while the welding parameters changing randomly.

While deep learning networks take weld pool images as input has become a popular and standard approach to monitor the weld penetration in recent literature, there has no study to analysis weather raw information is adequacy or not. This study analyzed the dynamic weld pool evolution that determines the weld penetration. Per analysis, the weld pool images must be serial in order to for the raw information to be adequate to reflect the critical dynamic evolution. This provides a novel innovative thinking and direction to choose the raw information and design the needed models accordingly. It played a decisive role in improving the prediction accuracy when the weld pool is under dynamic adjustment/development as in the beginning of the welding.

To accurately represent the dynamic changes in the weld penetration process, we carefully selected weld pool images from the most recent 4-second period, taking into

account the settling time of the GTAW process. Additionally, we adopted a sampling rate of one image from every 32 images captured at 60 Hz, resulting in a sampling time of 0.53 seconds. This approach ensured that the selected images effectively captured the differences in the weld pool's evolution while keeping the network manageable in size. By incorporating the most recent 8 images, we found that they contained the necessary features to predict the backside bead width accurately. This innovative network structure, designed based on thorough analysis, successfully addressed the weld pool image-based weld penetration prediction challenge for GTAW, which is a widely used welding process critical for applications requiring assured weld penetration.

The effectiveness of the prediction under dynamic adjustment/development of the weld pool is crucial for real-time control of the weld penetration. We expect that this method will be particularly useful when applied to real time control where the weld pool is subject to continuous dynamic adjustment. We will apply the proposed method to obtain the feedback for real time control of weld penetration and compare its accuracy in resultant welds with using feedback from non-serial images-based models.

CHAPTER 5. TRAINING DEEP LEARNING MODEL VIA DATASET WITH INACCURATE LABELS

5.1   Introduction

Since a knowledge base of weld scene-torch manipulation relationships will encompass a wide array of operation actions by human welders, the inherent diversity implies that their actions may not be identical even in similar welding scenarios. However, since welding adheres to certain underlying principles, their actions are anticipated to exhibit a general consistency despite nuanced differences. In essence, these variations can be likened to noise, and with a sufficiently large dataset, the mean error can be effectively treated as 0, i.e., they can be considered as an inaccurate dataset with mean error equal to 0.

As such, this chapter aims to address a fundamental question: Can a deep learning model be effectively trained using inaccurate labels that are easily obtainable? Specifically, we explore whether a deep learning model trained from accurate penetration labels can achieve comparable or even improved accuracy when trained with inaccurate labels. To investigate this question, we draw similarities with the standard Least Squares problem and propose the zero-mean as a condition for labeling inaccuracy. We demonstrate that the challenge of inaccuracy can be overcome by increasing the size of the dataset. Furthermore, we experimentally verify the effectiveness of this approach using weld penetration as a demonstration case. In our experiments, we utilize the filtered welding current from carefully designed experiments as an easily obtainable alternative for the inaccurate labels of weld penetration. It will provide valuable insights into the potential of training deep learning models with imperfect labels while maintaining high accuracy levels.

On the one hand, back side bead width w_b usually measured after welding to obtain the labels to train a network. However, this task may pose challenges for manufacturers who lack the necessary expertise. On the other hand, with the advent of Industry 4.0/5.0 and the Internet of Things (IoT), cloud-based computation services have become preferable, leveraging the vast amounts of data generated from various manufacturers and applications. In this context, manufacturers who are unable to acquire the necessary labels after welding may find their generated data going to waste. Moreover, for those manufacturers capable of setting up additional platforms to generate labels, this process can incur higher costs and may become practically infeasible. Therefore, in the era of Industry 4.0/5.0, it becomes crucial to explore substitutive labels for training networks efficiently and cost-effectively, ensuring that data generated during the manufacturing process is maximally utilized.
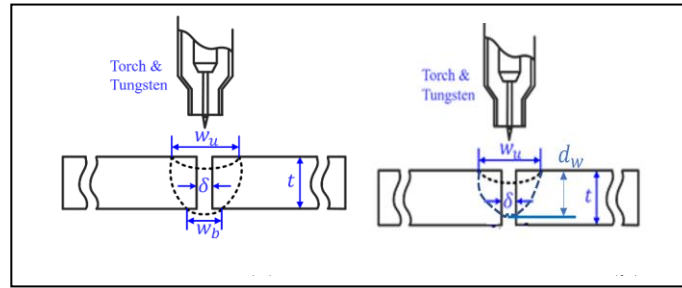
Figure 5.1 Illustration of weld penetration. (a) Complete penetration; (b) Incomplete penetration

Take welding cases as example, shown in Figure 5.1. The problem can be state as the following:

Given: 1) Paired topside sensor data Ξ_k and corresponding accurate label $X_k = x(k)(k = 1, \dots, N)$, where x is either w_b or d_w depending on the type of the penetration (complete or incomplete), forming the dataset (X, Ξ); and 2) A deep learning model $M(\Xi_k)$ trained from (X, Ξ) to predict X_k from Ξ_k with error $e(k) = X_k - M(\Xi_k)$ being quantified as $\sigma_e^2 =$

$(1/N) \sum_{k=1}^{N} e^2(k) = \kappa \geq 0$ where κ measures the accuracy from training a deep learning model by using N accurate labels. We are to 1) Obtain an alternative label set X_L^0 of increased size, with its size $\tilde{N} \gg N$, as an approximation of X_{Large} that is an extended accurate set from X; 2) Train the same network structure M by using the new dataset $[\Xi_L, X_L^0]$, rather than $[\Xi_L, X_{Large}]$, to achieve $\kappa^0 = \sigma_e^2 = (1/\tilde{N}) \sum_{k=1}^{\tilde{N}} e^2(k) \leq \kappa$. The labeling inaccuracy $E = X_L^0 - X_L$ should has zero mean and variance $\sigma_\varepsilon^2 = (1/\tilde{N}) E^T E > 0$. We note that X_{Large} exists but is not considered obtainable while its corresponding Ξ_L and X_L^0 are easily obtained during a manufacturing process.

5.2 Feasibility

Considering a conventional model M with a definite model structure and correlative parameters' vector $\theta \in R^p$:

$$x(k) = M(\varphi(k), \theta) \tag{18}$$

Here, $\varphi(k)$ serve as feature vector from Ξ_k. The model needs to find θ from dataset (X, Φ) where $X_k = x(k), \Phi_k = \varphi(k), k = 1, \ldots, N$. In general, when the labels are both accurate, θ can be solved if $= p$ (rank $(X, \Phi) = p$) and the same θ is obtained if $N > p$.

If inaccurate label X_k^0 are utilized where $X_k^0 = X_k + \varepsilon(k)$ $(k = 1, \ldots, N)$ with labeling error $\sigma_\varepsilon^2 = \left(\frac{1}{N}\right) E^T E > 0 (E_k = \varepsilon(k))$, it will have no "exact solutions" if $N > p$, a unique solution if $N = p$, infinite solution if $N < p$. However, with $k = 1, \ldots, N$ those solutions only ensure:

$$X_k^0 = M(\varphi(k), \theta) \tag{19}$$

Or

$$X_k = M(\varphi(k), \theta) + \varepsilon(k) \tag{20}$$

It cannot satisfy what system desired:

$$X_k = M(\varphi(k), \theta) \tag{21}$$

In the case, calculated θ only provide less ideal prediction result. Where best prediction in this scenario is typically obtained through Least Square Estimates $\hat{\theta}_{LS}$:

$$\sum_{k=1}^{N}(X_k^0 - M(\varphi(k), \hat{\theta}_{LS}))^2 = \min_{\theta \in R^p}\sum_{k=1}^{N}(X_k^0 - M(\varphi(k), \theta)^2 \tag{22}$$

Eq (19) training the model with inaccurate dataset and Eq (22) is the standard Least Squares problem. It is obvious that they share the similarity with each other. As such, the identification of a conventional model using inaccurate labels is exactly the same as its classical Least Squares estimation. Furthermore, the underlying assumption is that the measurements (labels) are subject to inaccuracies with a statistical zero-mean error. Therefore, employing a classical model with inaccurate labels is not a novel concept, and it is not only possible but also a standard practice. Hence, our task is to apply this principle to our specific purpose and implement it accordingly.

To illustrate it in detail, consider a linear model as an example:

$$X_k = \varphi^T(k)\theta \tag{23}$$

Let's illustrate this in detail with a linear model, where $\varphi^T(k) = [\varphi_1(k), \varphi_2(k)]$, and $= [\theta_1 \theta_2]^T$. We assume a known θ to calculate X_k from $\varphi(k)$. For convenience, let's assume that $-1 \leq \varphi_1(k) \leq 1$ and $-1 \leq \varphi_2(k) \leq 1$ ($k = 1, ..., N$) and φ_1 and φ_2 are independent uniformly distributed random numbers. Given θ, we can calculate $X_k's$ ($k = 1, ..., N$), forming accurate labels X. However, we introduce inaccuracies in the labels by using $X_k^0 = X_k - \varepsilon(k)$, where $-1 \leq \varepsilon(k) \leq 1$ is a uniformly distributed random number independent of $\varphi(k)$, to generate inaccurate labels $X^0 = (X_1^0, ..., X_N^0)^T$.

Figure 6.2 provides the results obtained from 100 simulations with $N = 100$ in Fig. 2(a) and $N = 1000$ in Fig. 2(b). The abscissa represents the order in which 100 independent simulations are conducted with $\theta = [0.5, 0.4]^T$, showing that the results may vary across each simulation. A point on the red line represents $\frac{var(\varepsilon)}{var(X)}$, which measures the normalized/relative labeling error in a particular simulation, while a point on the blue line represents $\frac{var(e)}{var(\varepsilon)}$, representing the normalized modeling error. $E\left(\frac{var(\varepsilon)}{var(X)}\right)$ and $E(\frac{var(e)}{var(\varepsilon)})$ are the means of their normalized inaccuracies/errors from all simulations.

From Figure 5.2, it can be observed that $\frac{var(\varepsilon)}{var(X)}$ is approximately 0.7, and this normalized labeling error is independent of the identification and is thus independent of N. For $N = 100$, $E\frac{var(e)}{var(\varepsilon)} = 1.26\%$. The modeling error is much smaller than the labeling error. When $N = 1000$, this reduces to $0.7091 * 0.0019 = 0.13\%$. The large error in labeling is compensated by increasing the number of samples, as expected.

(a) $N = 100$ (b) $N = 1000$
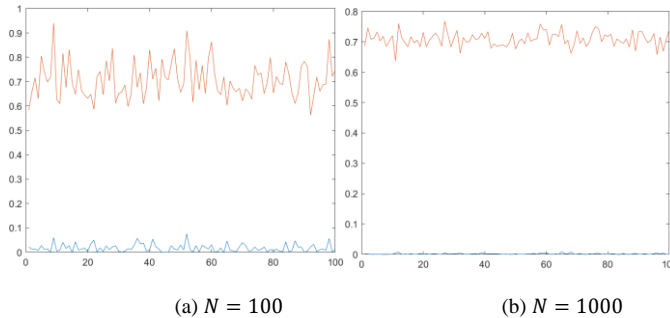
Figure 5.2 Normalized labeling error (red) and normalized modeling error (blue)

Using a simple linear model to achieve the global minimum and guarantee the best possible fit to the data is not practically useful because this guarantee does not apply to complex models. For nonlinear models, obtaining meaningful statistics like those presented above from hundreds or thousands of modeling fitting simulations can be challenging. Convergence of parameter estimation for nonlinear models depends on the

optimization algorithm used and is not always guaranteed. Nevertheless, regardless of model complexity and optimization algorithm, the main objective of model identification is to find the best model parameters that can minimize the prediction error. The minimal achievable prediction error is dependent on the model and algorithm chosen. Therefore, it is important to numerically illustrate the feasibility of using inaccurate labels through simulations. Deep learning models are also considered nonlinear mappings, but they are much more complex in terms of nonlinearity, higher input dimension, more parameters, and larger data size. Despite these complexities, using inaccurate labels may still be feasible for deep learning models. The inherent flexibility and adaptability of deep learning architectures make them more robust to noisy or imperfect labels, allowing them to learn and generalize effectively from noisy data.

5.3 Train Deep model via datasets with inaccurate labels

To compare the feasibility of training deep learning models with alternative/inaccurate labels, a deep learning model take accurate labels as input to predict w_b from weld pool images which has been trained. The dataset used is $[\Xi, X]$ and final accuracy achieved can be saved as the benchmark for later reference.

Zero-mean random errors E is then introduced into the X to generate $X^0 = X + E$ where E has the same dimensions with X. Model training with dataset X^0 can not achieve the same accuracy level as the model training with X, although the labeling inaccuracy has a theoretical zero-mean. After analyzing and identifying the reason, we propose to subtract the same/realized random errors, ensuring that the zero-mean is not only theoretically achieved (which would require infinite data for practical guarantee), but also practically realized. By using the dataset $[\Xi_L, X_L^0] = [\Xi\ X + \mathrm{E}; \Xi\ X - \mathrm{E}]$ with inaccurate labels but

$\widetilde{N} = 2N$, the modeling accuracy is greatly improved. Furthermore, by using an even larger dataset $[\Xi_L, X_L^0] = [\Xi X + E^{(1)}; \Xi X - E^{(2)}; \Xi X + E^{(2)}; \Xi X - E^{(2)}]$ with $\widetilde{N} = 4N$, where $E^{(1)}$ and $E^{(2)}$ are two separately random number sequences, the accuracy is further enhanced. This trend can continue by adding more realized random number sequences.

Experiments:

A Gas Tungsten Arc Welding (GTAW) experimental system has been used to perform the experiments to generate the dataset $[\Xi X]$ used in this chapter. The system consists of two cameras placed at the top and backside of the workpieces, capturing front-side images as Ξ and backside images during the welding process simultaneously. The backside images are subsequently utilized to calculate using a threshold [44] after the welding process.

During the experiments, the welding speed was maintained at a fixed rate of 2mm/s. The welding current I_k was defined as the sum of the baseline current I_b, which remains constant throughout each experiment, and the deviation ΔI_k, representing the actual current's fluctuation from the baseline. Every 2 seconds, ΔI_k was randomly varied within the range of $[-5A, 5A]$. This random variation in the current was applied to generate a diverse distribution within the dataset $[\Xi X]$, encompassing various penetration states (backside bead widths) and welding phenomena. $90A$ and $100A$ has been chosen as two baselines current, thus, the actual current ranging is $[85A, 105A]$. This current range allows $2mm$ thick stainless-steel workpieces achieve an appropriate penetration range, i.e., ensuring complete penetration without an excessively large backside bead width.

Each individual welding experiment was carried out for a duration of 40 seconds. For each experiment, we performed two sets of trials: one using $I_k = I_b + \Delta I_k$ with a

specific and realized random sequence $\Delta I_k's$, and the other using $I_k = I_b - \Delta I_k$ with the same $\Delta I_k's$. This resulted in a total of 12 welding experiments, forming six pairs of experiments for each baseline current level of $90A$ and $100A$. Each current level was utilized for three pairs of experiments. Throughout the experiments, the cameras recorded a high-speed capture rate of 60 images per second, leading to the acquisition of a total of 28,800 data pairs for analysis. After removing the data associated with the initial phase of each experiment, where complete penetration had not yet been achieved, a total of 16,200 data/image pairs remain to construct the dataset $[\Xi\ X]$ for this study. The precise labels, representing the backside bead width derived from X, are illustrated in Figure 6.3(a) and (c). These figures display the accurate labels separately for a specific division of the training set, with a size of 13,000 for Figure 6.3(a) and the validation set with a size of 3,200 for Figure 6.3(c). Other divisions of the data can also be employed and specified accordingly.

Network: A Convolutional Neural Network (CNN) is carefully designed to predict X_k from Ξ_k. The CNN architecture is composed of four repetitions of a standard convolution layer, followed by a pooling layer. The specific parameters for these layers are (1, 16, 5, 1, 2), (16, 32, 3, 2, 1), (32, 64, 3, 2, 1), and (64, 128, 3, 2, 1), respectively. Subsequently, the output from the final pooling layer is directed into two fully connected layers, progressively reducing the number of channels from 128 to 64, and finally to 1.

To train the model, the mean square error (MSE) is chosen as the loss function, and the optimization is achieved using the stochastic gradient descent (SGD) method [XX]. Throughout the training process, a learning rate of 0.0001 is employed to facilitate
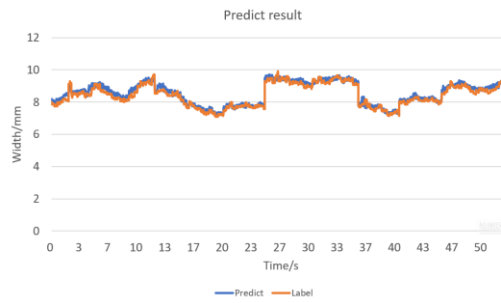
convergence and accuracy. These design choices aim to ensure robust performance and precise prediction of X_k based on Ξ_k in our research.
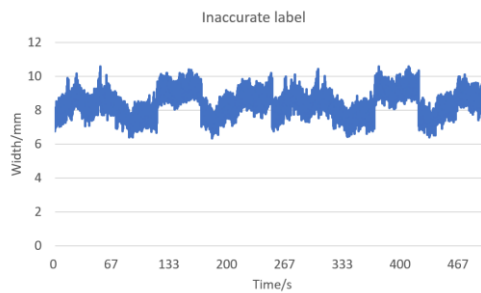

(a)
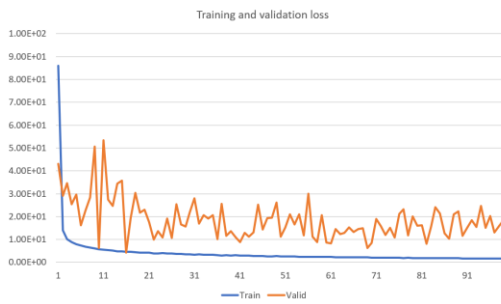
(b)

(c)

Figure 5.3 Training using accurate label. (a) Training label; (b) Training and validataion loss; (c) Validation label and prediction
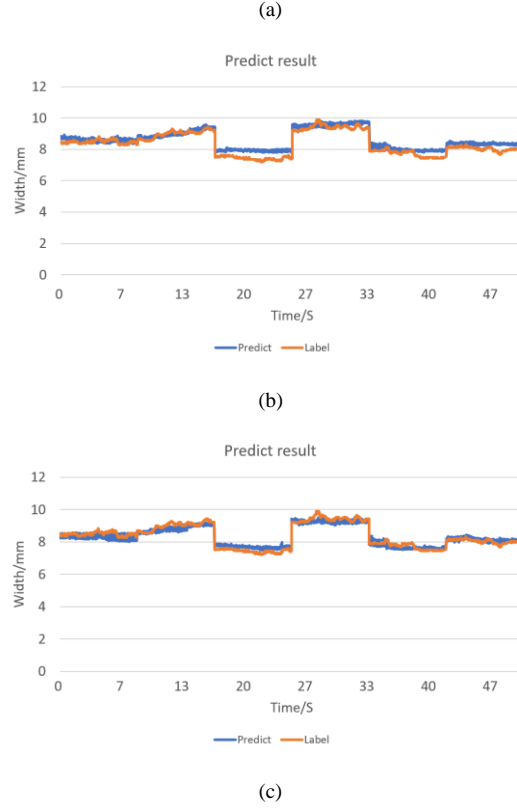
(b)

(c)

Figure 5.4 Training using inaccurate label. (a) Inaccurate training label; (b) Validation and prediction with $\tilde{N} = 2N$; (c) Validation and prediction with $\tilde{N} = 10N$

Benchmark from Training Using Accurate Labels: As previously discussed, a benchmark was established as reference by training network using the accurate labels X. As illustrated in Figure 6.4, this benchmark model achieved a validation accuracy of $0.2mm$, i.e., $\sqrt{\kappa} = 0.2\ mm$. Training and validation losses are shows in Fig. X. Followed by and early stop method, the best prediction model is chosen at the epoch 16, where the loss reached its minimum point.

Training Using Inaccurate Labels: A uniformly distributed random number is generated and added/subtracted to the accurate training labels to form a new training set which doubles the size of the training data from $[\Xi\ X]$ to $[\Xi_L, X_L^0] = [\Xi\ X + E; \Xi\ X - E]$ as shown in Figure 5.4(a), where each element in E is a random number within $[-1.2mm, 1.2mm]$. The same CNN structure is trained with the new/inaccurate dataset of

the doubled size ($\widetilde{N} = 2N$). The validation result is shown in Figure 6.4(b) achieving the validation accuracy with the mean square error at 0.32mm ($\sqrt{\kappa^0} = 0.32mm$). As previously discussed and demonstrated by using a linear model, increasing the dataset size can reduce the error. As such, the model is trained again with a dataset $[\Xi_L, X_L^0] = [\Xi X +$ $E^{(1)}; \Xi X - E^{(1)}; \Xi X + E^{(2)}; \Xi X - E^{(2)}, \cdots\cdots, \Xi X + E^{(5)} \Xi X + E^{(5)}]$, where $E^{(j)}(j = 1, \dots, 5)$ are separately realized random number sequences, resulting in a ten-fold size of the training dataset ($\widetilde{N} = 10N$). The results are shown in Figure 6.4(c). The mean square error $\sqrt{\kappa^0}$ is reduced from $0.32mm$ to $0.25mm$ which is much closer to that of the benchmark trained using the accurate labels. This demonstrates that training a deep learning model using inaccurate labels is possible for the monitoring of weld penetration.

5.4   Using current as inaccurate label

While we have demonstrated the feasibility for using inaccurate labels for model training, the demonstration uses accurate labels that are not supposed to be available to generate inaccurate labels. The availability of accurate labels provides us an easy way to assure the zero mean of the labeling inaccuracy. When the accurate label is not available, finding inaccurate labels that can be easily and automatically obtained to assure the zero mean in the labeling accuracy is challenging, in particular when the number of the samples is not extremely large. However, although finding such inaccurate labels is not easy, we have demonstrated the feasibility of using inaccurate labels for model training and can work on finding "better inaccurate labels".

We first test the ability to use the welding current as the inaccurate label as it is easily and automatically obtainable in manufacturing. The data size can be almost infinitely large

through IoT-D^3 to continuously increase the size of the available dataset for training. The weld penetration is determined by the heat input and the penetrating force during the welding process for a given welding condition. The heat input is the heat/energy supplied into the workpiece in a unit length. In welding, it is calculated as the welding current multiplied by the welding voltage divided by the welding speed. In GTAW, for a given application, the welding voltage is approximately constant. If the welding speed is also given, the heat input is proportional to the welding current such that the welding current is the only parameter determining the heat input. For the penetrating force in GTAW, it is primarily from the arc pressure that is proportional to the square of welding current. Hence, welding current can be considered the only parameter that controls weld penetration so that $X_k = f(I_k)$, where I_k and X_k are the welding current and weld penetration at instant k. The current is thus a good candidate as an inaccurate penetration label. If the welding speed varies, we can use the welding speed and current together to form the inaccurate label. This work focuses on using welding current as inaccurate labels.
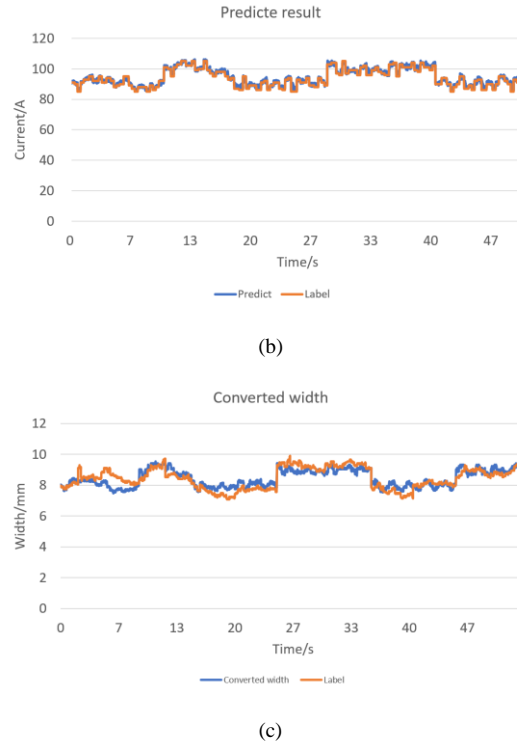

(a)

(b)

(c)
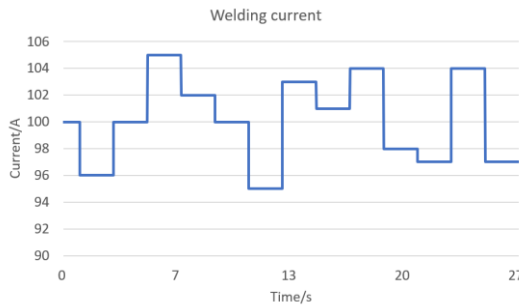
Figure 5.5 Training using welding current as inaccurate label. (a) Welding current in one experiment; (b) Welding current validation label and its prediction; (c) Accurate label and predicted label (converted from the predicted current)

In each application for which we are building a penetration monitoring model, penetration is determined by welding current and welding conditions. Each given application corresponds to a nominal welding condition. It is the deviation of the actual welding condition from the nominal one that causes the weld penetration produced by the designed welding procedure (welding current, welding speed etc.) to deviate from the expected one. As such, although the deviation of a welding condition is difficult to quantify, we can envision that the effect of such deviation causes the penetration to vary around the desired one in both directions (greater or smaller). Hence, we can easily assume $X_k = X_k^0 + \varepsilon(k)$, where $X_k^0 = f(I_k)$, with the labeling error $\varepsilon(k)$ being caused by the deviation of a welding condition from the nominal one. It is reasonable to hypothesize that the labeling error is zero mean when the data size is sufficiently large. This can be realized in

72

manufacturing through IoT-D^3. $X^0 = f(I)$ can thus be used as a candidate of inaccurate label. Our task is now to use the dataset $[\Xi\ X^0]$ to train a model that can predict X, rather than predicting X^0, although X is not used for training. As f is unknown but it is deterministic, we can identify it from all or a subset of $[X\ I]$. We thus can first train the same benchmark model structure for $\hat{I}$ (in A) and then predict the penetration using $f(\hat{I})$ (in mm). In this study, we simply fit a linear model for f from available $[X\ I]$.

Figure 5.5(a) shows welding current in a single welding experiment that changes each 2 seconds. We discretize the current at 60 Hz to match with the image acquisition frequency to obtain the inaccurate label forming I in the dataset $[\Xi\ I]$. Figure 5.5(b) shows the welding current label used for training and the training result $\hat{I} = g(\Xi)$ where g is the trained CNN model. As can be seen in Figure 6.5(b), the current is accurately predicted from the top side image. However, as there is a labeling inaccuracy, the welding current does not exactly match with the actual weld penetration. The final effectiveness for training by using the current as an inaccurate label is shown in the validation result in Figure 5.5(c). The mean square error $\sqrt{\kappa^0}$ is 0.46mm.

Please note that we have used current sequence pairs $\{I_k = I_b + \Delta I_k, I_k = I_b - \Delta I_k\}$ to conduct an experiment pair. The purpose is to reduce the mean error in the labeling error when using the inaccurate label. This is because there is a dynamic process from a change in the welding current to its produced change in the weld pool and weld penetration. If ΔI_k is positive, $f(I_k = I_b + \Delta I_k)$ is more likely to be greater than the actual label such that labeling error $\varepsilon(k)$ is negative. However, $f(I_k = I_b - \Delta I_k)$ would be more likely to be smaller such that $\varepsilon(k)$ is positive. This helps achieve the desired "zero mean" property for the error of the inaccurate labels. While this helps, the response dynamics may be further

utilized to reduce the labeling error by filtering the current. Ideally, the filter needs to be designed per dynamic response property. In Figure 5.6(a), we use filter $I_{f,k} = \alpha I_{f,k-1} + (1 - \alpha)I_k$ $(0 < \alpha < 1)$ with $\alpha = 0.9$ for I_k that may change in 1/60 second. The original current I and filter current I_f in an experiment are illustrated in Figure 5.6(a). Using I_f as the inaccurate label, the trained model whose training result is shown Figure 5.6(b) can better predict the actual label in the validation data (Figure 5.6(c)). The prediction error $\sqrt{\kappa^0}$ is reduced from 0.46 mm (using the current) to 0.39mm (using the filtered current). Changing α to better match the actual dynamics from the current change to the backside bead width change may further improve the accuracy but it is not the focus of this study that aims to demonstrate the feasibility.
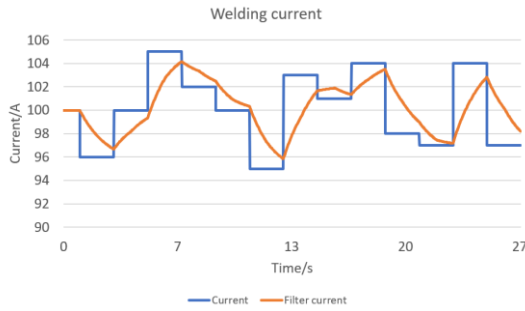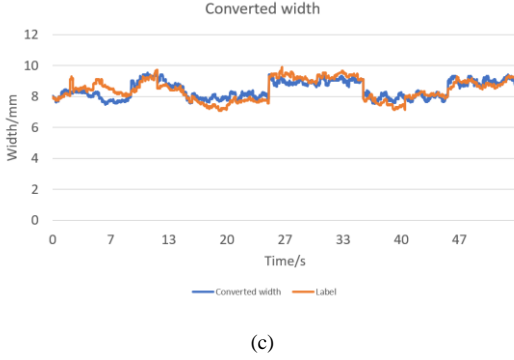

(a)
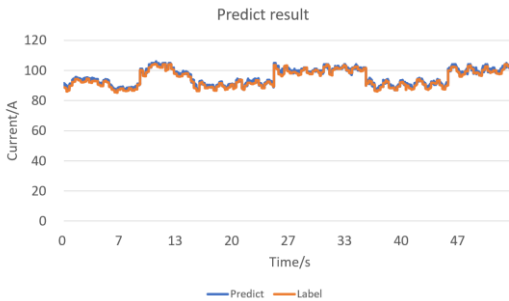
(b)

Figure 5.6 Training using filtered welding current as inaccurate label. (a) Welding current and filter current in one experiment; (b) Filtered welding current validation label and its prediction; (c) Accurate validation label and predicted label (converted from the predicted current)

5.5   Effectiveness of Using Larger Size Data with Inaccurate Welding Current Labels

We have just demonstrated the feasibility to train a deep learning model from the filtered current as a substitutive inaccurate label. However, its potential in improving the accuracy by taking advantage of the increased data size, for example through $IoT\text{-}D^3$ in manufacturing, has not been fully demonstrated. To further demonstrate, we may conduct more experiments to increase $\widetilde{N}$, for example increasing the number of experiments from 12 to 120 or reducing N (data size for training using accurate labels). Increasing $\widetilde{N}$ can better realize the zero-mean property but using a smaller $\widetilde{N}$ (by reducing N accordingly) provides a more convincing demonstration. In addition, 120 experiments are too extensive for an initial study aiming at demonstrating feasibility. Hence, we reduce N from 13,000 to 1,300 while $\widetilde{N} = 10N = 13,000$..

The same model structure is trained using $N = 1,300$ accurate weld penetration labels to compare with the model trained using filtered current. The validation set is the same for these two comparative models. As shown in Figure 5.7, the mean square error $\sqrt{\kappa} = 0.38$ mm. This is the result for using $[\Xi, X]$. In Figure 5.6(c) with $\widetilde{N} = 13,000$,

75

$\sqrt{\kappa^0} = 0.39$mm after the linear conversion. This is the result from using $[\Xi_L, X_L^0]$. We can thus see that the same accuracy is achieved through by using $[\Xi_L, X_L^0]$. The solution of the problem as specified in Section II is thus experimentally demonstrated.
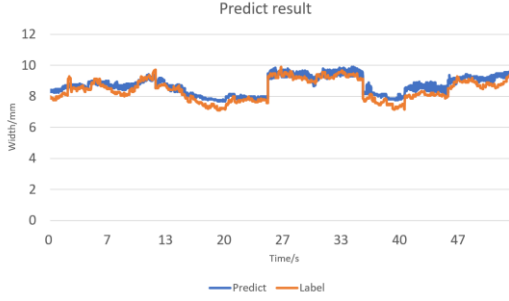

Figure 5.7 Validation result for training using a smaller dataset

## 5.6 Calibration and Semi-supervised Pre-training Approach

The model trained using the filtered current outputs the estimated label in the unit of current (ampere). We have converted it from ampere to millimeter in order to compare with the backside bead width. We argue that this conversion may not be absolutely needed as we can easily find the maximum current I_{max} and minimum current I_{min} each of which in average produces the maximally/minimally allowed backside width $w_{b,max}/w_{b,min}$. The degree of the estimated penetration is a normalized number $r = 100 * (M(\Xi_k) - I_{min})/(I_{max} - I_{min})$ percent with $r = 0$ corresponding to $w_{b,min}$ and $r = 100\%$ to $w_{b,max}$. However, it is still preferred that a calibrated model can be used to convert the model output to the bead width. To this end, at least a small set of $[X, I]$ is needed where X is a small set of accurate labels. Now the question is if we can do more in addition to the calibration if we have a small set of accurate labels and if we can deal with nonlinear calibration.

Denote $I = \mathrm{M}_{nn}(\varphi)$ as the deep learning model to be trained using inaccurate labels where 1) the feature map φ is generated by $\mathrm{M}_{conv}(\Xi)$, layers of repetitive conventions each followed by pooling, from the topside weld pool image Ξ, and 2) M_{nn} is the fully connected neural network. Assume it has been trained by using a large number of inaccurate labels (filtered current) resulting in M_{conv}^* and M_{nn}^*.

We now propose to train a model $X = \mathrm{M}_{nn}(\mathrm{M}_{conv}^*(\Xi))$ from the small dataset $[\Xi\ X]$ where M_{conv}^* is fixed and only M_{nn} is trained. This first serves the purpose for calibration. However, we are also using a pre-training approach, a model training methodology which is pioneered by OpenAI in GPT1 [18]. It allows us to learn high-level representations of data by training a model on a vast amount of unlabeled data. This approach uses the learned representations as a foundation for fine-tuning the model on a smaller, task-specific dataset, improving performance on the target task. Pre-training can help mitigate overfitting issues, especially when the target task has limited labeled data available. Additionally, it enhances the model's ability to generalize effectively to novel data.

We propose a pre-training approach in the supervised setting using inaccurate labels. As such, we pre-trained the model with a large dataset of easily obtained inaccurate labels (filtered current) to establish a correlation between the topside weld pool image and welding current. Then, we transferred the pre-trained model M_{conv}^* using a much smaller dataset with accurate labels X to ensure that the model can effectively establish a correlation between the topside weld pool image and the accurate backside bead width. By introducing pre-training in this way, we can significantly reduce the cost of obtaining accurate labels during the welding process while improving the model's prediction accuracy. The model used to predict the filtered current, shown in Figure 5.6 (b), was
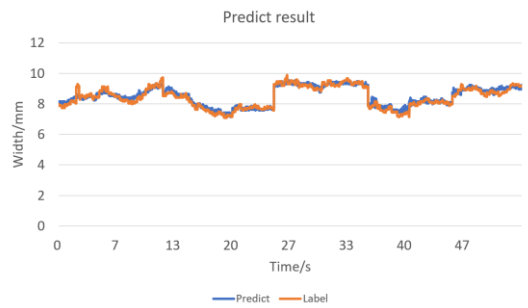
selected as the model for transfer learning. The training of the fully connected layers of our network using the accurate labels, with its capacity to approximate a nonlinear function, enables our network to learn nonlinear calibration between current and penetration.
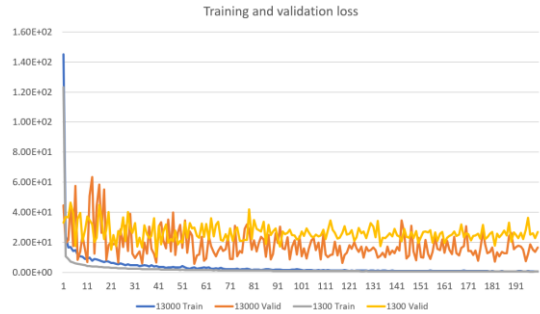
(a)
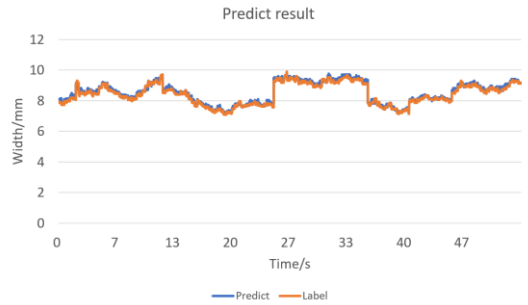
(b)

(c)

Figure 5.8 Validation results of proposed pre-training approach. (a) Training and validation losses; (b) Re-trained with 13,000 accurate labels; (c) Re-trained with 1,300 accurate labels

To test the effectiveness of the proposed transfer learning approach, we still use the entire dataset of size 16,200 of the accurate label. However, we keep all the parameters trained using the filtered current as the inaccurate label but re-train the fully connected

layers by 13,000 accurate labels and 3,200 (3200/60=53.3 seconds) for validation. The

training and validation losses are shown in Figure 5.8(a). The validation result is shown in

Figure 5.8(b). The prediction error is reduced to 0.13mm from that was from model trained

using the same accurate label. This demonstrates the effectiveness of our proposed transfer

learning approach.

Now we assume that we only have a much smaller set of data with accurate labels X.

To this end, we reduce the size of the data for transfer learning to just 1,300 while still

keeping the validation data size unchanged at 3,200. The training and validation losses are

shown in Figure 5.8. As shown in Figure 5.8(c), the error achieved 0.18mm. This is smaller

than 0.2 mm, i.e., the error achieved using the entire data of accurate labels which is ten

times of the accurate label data used in the proposed method, or the 0.38 mm error achieved

by using the initial model trained on a dataset of 1,300 images. As such, the significant

benefit of using the pre-trained approach to reduce the error due to (possible) infinite size

of the used inaccurate labels is demonstrated. In this particular case, better accuracy is

achieved by using 1/10 of the accurate labels by leveraging the readily available welding

current data.

5.7   Summary

This chapter experimentally demonstrated that 1) the filtered current is a reasonable

inaccurate label to train a deep learning model to predict the weld penetration and 2)

increasing the size of the filtered current in the dataset as the alternative inaccurate label

can achieve the same accuracy as training using a smaller portion of accurate labels. A

possibility is thus demonstrated through the weld penetration monitoring case for training

deep learning models using automatically obtained labels, that may be inaccurate,

suggesting a direction for more effectively taking advantage of large data available in manufacturing through Internet of Things from different operations at different locations in different time.

In addition, we have proposed to use a large number of inaccurate labels (filtered current) to pre-train a model and then re-train its fully connected neural network by using a small set of accurate labels. Although the size of the accurate labels is reduced to one tenth, better accuracy has been achieved. This provides an effective method to calibrate and better train a deep learning model by taking advantage of a large number of inaccurate, automatically obtained labels.

CHAPTER 6. ROBOTIZE DE-GMAW PROCESS AND ITS MONITORING

6.1   Introduction

DE-GMAW welding modified conventional GMAW process by introduce a second electrode to bypass a portion of the current flowing from the wire to reduce the heat imposed on the workpieces. To establish the system which can perform the DE-GMAW process with a stabilize bypass arc, it is important to have the ability to justify the quality of the bypass arc, which parameters in the DE-GMAW welding process will influence the quality of the bypass arc, which parameters can represent the quality of the bypass arc. The major challenge with DE-GMAW process is to precisely position the second electrode in close proximity to the main wire to sustain the bypass arc. The arcing phenomenon need to be investigated to dynamically adjusting the bypass electrode in real-time.

6.2   Process background

As shown in the Figure 6.1. It is obvious that raw image information can effectively represent the bypass arc state, but influenced by the arc illumination, it is challenging to process the raw image to identify the bypass electrode, main wire, bypass arc in real-time. Experience human welder can distinguish those components with careful inspection, but it takes years training to enhance their ability. Consider the weld pool view is the majority of the information source human welder will receive during the manual welding process, they take weld pool view as input and output their control operation. Consequently, to replicate the human expertise, raw image representing their observation should be an ideal input information source to train the deep learning model.

Recognizing that the robotization of the DE-GMAW process is a novel endeavor, we acknowledge the absence of a fully controllable robotized procedure, and its control mechanisms remain uncharted territory. In response, we propose to draw insights from human welders as a means to tackle this challenge. Consequently, an experimental setup has been devised, involving a tractor manipulating the GMAW torch, while a human welder monitors the arc and adjusts the bypass electrode. Given the unfamiliarity of the human welder with this specific process, their bypass electrode control is not optimally refined. This dynamic offers a spectrum of scenarios from which we can glean lessons on both successes and failures, informing the development of an adaptive control algorithm. This algorithm is designed to enable the follower robot to adjust the bypass electrode based on insights learned from human welders. In essence, these scenarios, emulating real manufacturing conditions within a controlled laboratory environment, facilitate a more pragmatic exploration of how operational modes may evolve during actual production.

Figure 6.1 Arc image captured with high-speed camera [92]

6.3   Experiments and data collection

Follow the system setup illustrate in Figure 6.2. An experimental system is established as shown in the Figure 6.3. a skilled human welder will operate a GTAW torch while collaborating with a tractor that has an attached GMAW torch. The tractor will move at a fixed speed, and the human welder will adjust her weld trajectory based on the relationship between the weld pool and the two torches that she observes. A Point Grey

camera FL3-FW-03S1C was attached to the tractor to move with it and observe the weld pool on the side during the welding process, shows in the Figure 6.4. The GMAW power source used was the Miller Auto-Continuum 350, and the GTAW power source was the Miller Maxstar 210. During the welding process. the weld image, voltage, and current are sensing synchronously. The current for the GTAW power source I_2 was set as $100A$, while the voltage for the GMAW power source V_1 was set as $33V$ and the wire feed speed was $270IPM$.
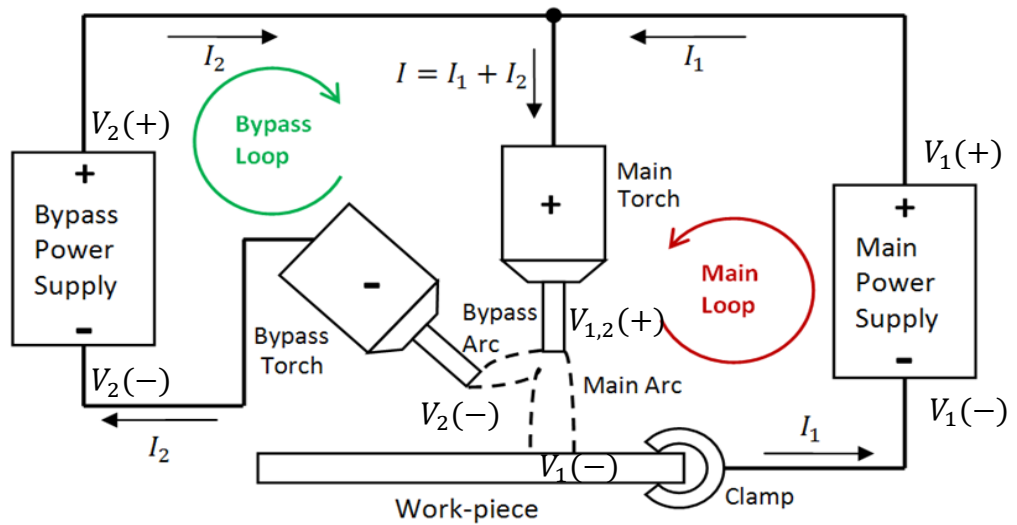

Figure 6.2 DE-GMAW system setup

Figure 6.3 Experimental setup, with power not in the view

Examples of the recorded images are shown in Figure 6.5. Among the manual welding process, various phenomena will occur cause by humans' uncertainty. Which provide a comprehensive dataset for the later training process.
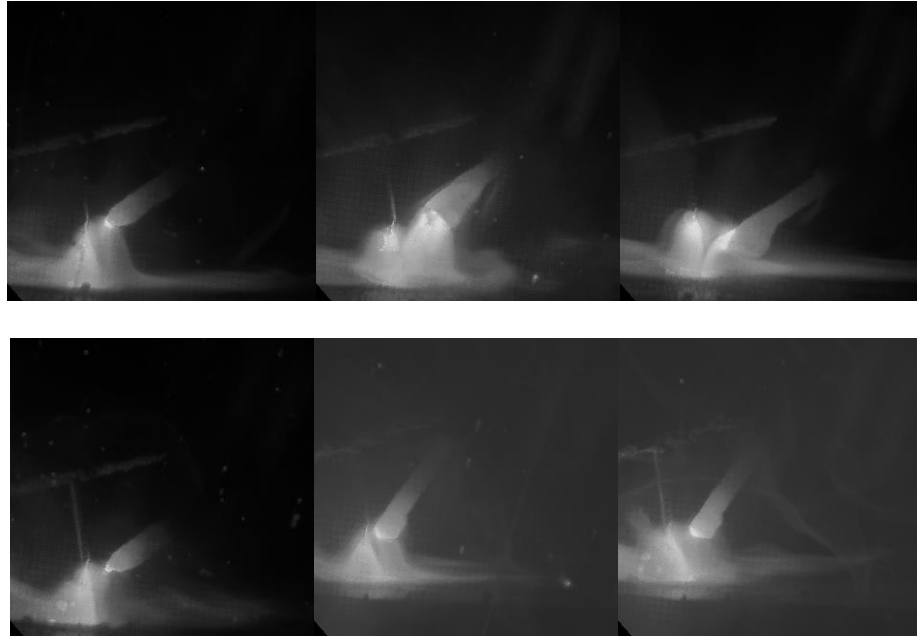

Figure 6.4 Examples of the recorded images

As we discussed before, experienced human welder can distinguish between those welding image, evaluate the bypass quality based on their years of experience. Thus, manual labeling the image to obtain their corresponding label is feasible but very time consuming, makes it not practically feasible. There, an alternative label needs to be discovered which can automatically generated and reflect the arc mode. To this end, we analyze the underlying process.

Inspect the image captured during the DE-GMAW process, as shown in Figure 6.5. There are typically three states of the arc, as shown separately in the figure. In Figure 6.5 (a), there is no bypass arc between the GMAW torch and the GTAW torch, and the GTAW arc is not ignited. Therefore, no current passes through the GTAW torch, and the voltage

behaves as an open-loop voltage, as seen in the current and voltage plot, called as open arc. If the GTAW arc successfully ignites, as shown in Figure 6.5 (b), a bypass arc is formed between the two torches, and the current passing through the GTAW torch will be maintained at the value set on the power source, which is 100A in this case. This 100A current is involved in melting the wire, increasing the melting speed without passing through the base workpieces, thus decoupling the melting current and the base current. The voltage ranged between [20A,40A], and it exhibited a linear relationship with the distance between the two torches, called as parallel arc. Figure 6.5 (c) illustrates another state that may occur during the DE-GMAW process, where the distance between the two torches is not manipulated appropriately, resulting in a serial arc appearing between the two torches, and each torch will establish its own arc with the base metal. These two arcs caused the voltage to exceed 40A, are called as serial arc.

(a)   (b)   (c)
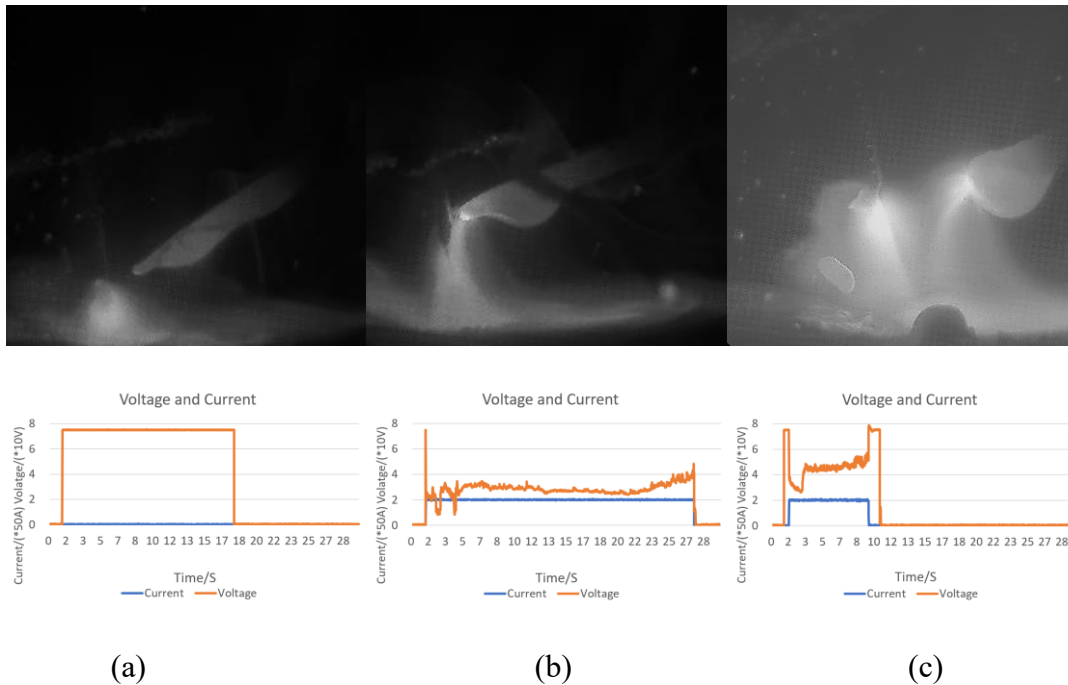
Figure 6.5 Bypass arc at different state; (a) Open arc; (b) Parallel arc (c) Serial arc
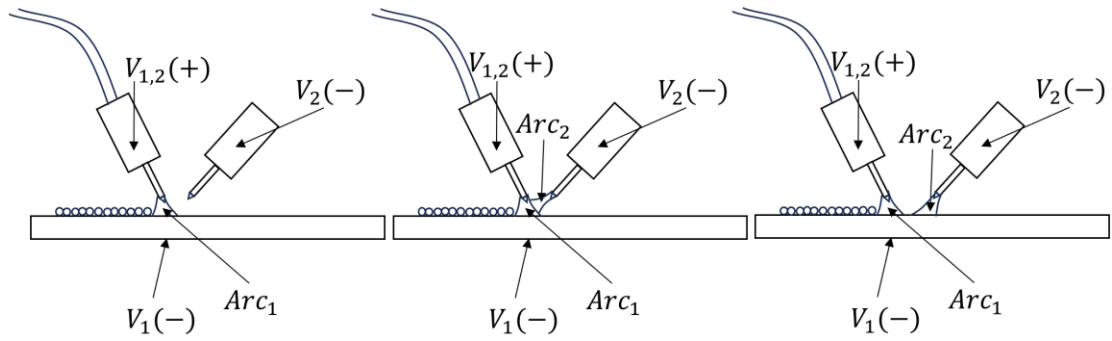
Figure 6.6 Electrical principles of different operation modes

The details of these three arc states are illustrated in the Figure 6.6. As discussed, both states (a) and (c) are undesirable in real applications. Only state (b) achieves the intended purpose of the DE-GMAW process, which is to reduce heat accumulation on the base metal while maintaining the heat input on the feed wire. Therefore, distinguishing between these three states is crucial in practical applications. Utilizing voltage as a classifier for these three states may be intuitive, but it can vary due to different power sources, feeding wires, or current setups. Therefore, we propose using a deep learning model to process the captured images and classify the states accurately, regardless of the variations in voltage caused by different factors, for more reliable and consistent results.
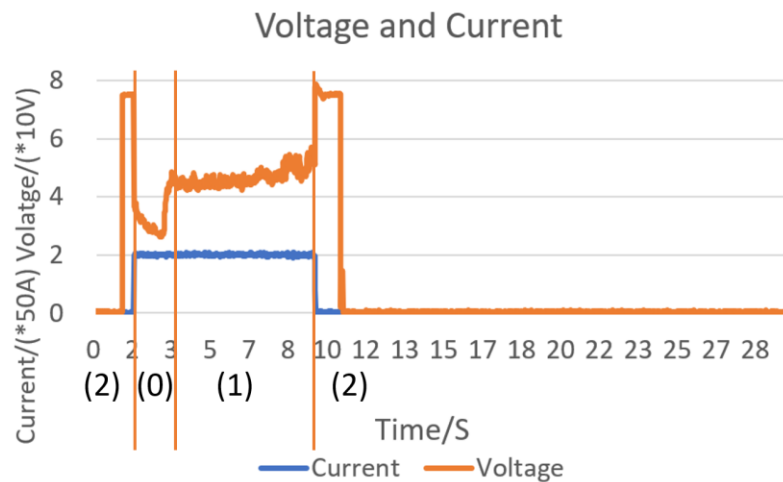

Figure 6.7 Voltage based segmentation

To accurately label the image into 3 categories based on the voltage information, we take one experiment process for a detail analysis. As shown in Figure 6.7. the voltage profile during a DE-GMAW process can be segmented into distinct parts, namely: (0) state, representing the desired bypass arc; (1) state, indicating the undesired serial arc; and (2) state, denoting the undesired open arc. These segments help to distinguish the different arc states and contribute to a more comprehensive understanding of the DE-GMAW process.

With the assistance of a skilled human welder, 20 weld trials were conducted, and a total of 10,062 data pairs $[I_k, S_k]$ have been collected, where $k \in [1, 10062]$.

6.4 Network and training

To predict the label S from the input image I, a convolution neural network architecture has been designed, shown in the Figure 6.8. The model takes the input image with size of $1 * 256 * 256$. Follow the five repeated convolution process, the feature's channel increased from 1 to 16, 32, 64, 128, 256. ReLu activation and batch normalization are applied between each convolution and pooling process. After processing the input image into a 256*1 feature vector, it is passed through two continuous fully connected layers, reducing its dimension from 256 to 128 and finally to 3. Softmax process is then performed on the resulting 3*1 feature vector to obtain the real class of the input image.
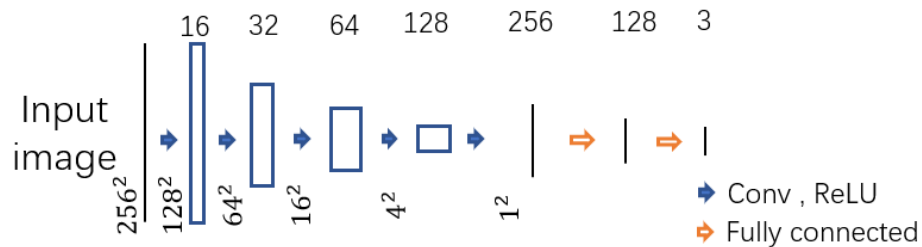

Figure 6.8 CNN for classification.

The training and validation process has been conducted on a NVIDIA GTX 2080 graphic card. With the dataset collected containing 10062 data pairs, a total of 9,692 data pairs were used for the training process, while the remaining 470 data pairs were utilized for the validation process. The model was trained iteratively 200 times with SGD optimizer and cross-entropy loss under Python environment with Pytorch library. During the training and validation process, the dataset has been shuffled to ensure that the data has been drawn randomly. The learning rate was set as 0.0001.

The loss results during the training and validation processes are displayed in Figure 6.9. The validation loss stabilizes after epoch 33, while the training loss continues to decrease until the end. The model with the minimum validation loss at epoch 67 is selected, and the test results are presented in Figure 6.10.
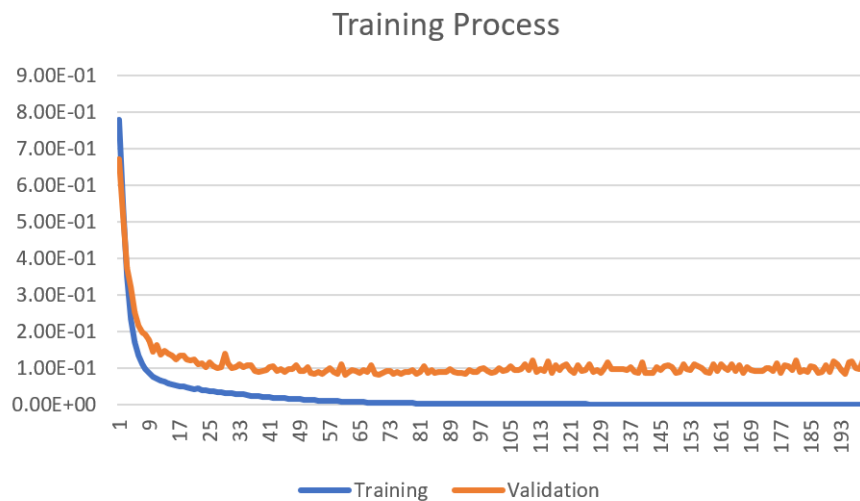

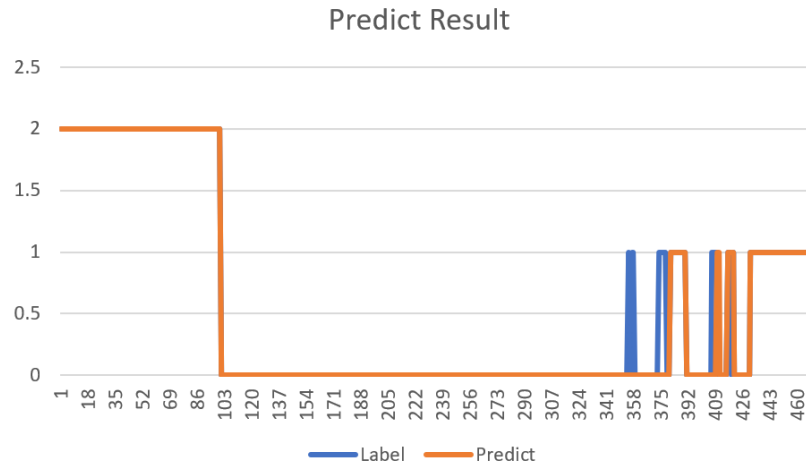Figure 6.9 Training and validation curve

Figure 6.10 Validation result

For a more intuitive view of the results, Figure 6.11 displays the voltage, two reference lines, and the errors obtained from the Label-Predict result. Clearly, in the first segment of the data, where the voltage exceeds 70V (indicating only GMAW arc with open-loop voltage), the model's accuracy is high with no errors observed. Similarly, during the second segment, where the voltage fluctuates around 30V, representing a successful DE-GMAW process, the model also performs accurately. However, in the third segment, which marks the transition between the successful by-pass arc and the undesired serial arc, the model makes several mistakes. Use voltage to calculate mean square loss for the wrong predictions is 0.039.
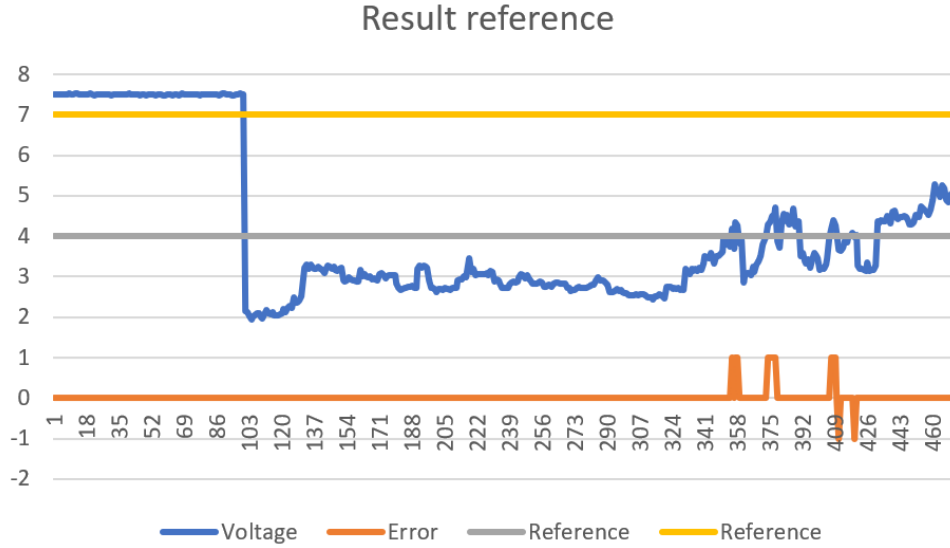
89

Figure 6.11 Analysis of the prediction error: where errors occur.

6.5 K-means classification

K-means clustering, as described in [94], is an approach employed to cluster data points into distinct groups. The central objective is to minimize the aggregate of squared distances between individual points and the centroids of their respective clusters. The iterative process encompasses the assignment of points to the closest centroid and subsequent centroid updates. The overarching aim is to identify centroids that effectively reduce the cumulative distance within each cluster. For every data point x_i, the nearest centroid c_j is determined, leading to the assignment of x_i to cluster j:

$$min_j \|x_i - c_j\|^2 \tag{24}$$

For each cluster j, update its centroid c_j to the mean of all data points assigned to it:

$$c_j = \frac{1}{|s_j|} \Sigma_{x_i \in s_j} x_i \tag{25}$$

During this procedure, s_j represents the collection of data points allocated to cluster j. Through iterating of these two phases, a point of convergence is reached, characterized by stable assignments and centroids. This algorithm efficiently divides the data into K clusters, where K is a pre-established hyperparameter. The selection of K holds substantial impact on the clustering quality. To ascertain the suitable value for K, the gap statistic method [95] is utilized by:

$$Gap(K) = Avg[\log(J_{null})] - \log(J_{actual}) \tag{26}$$

Gap statistic method generate several random data set and sum their squared distances as J_{null}, $Avg = g[\log(J_{null})]$ represents the average of the logarithms of the sum of squared distances for the null data clusters. Compare the difference between the logarithm of the sum of squared distances for the actual data clusters J_{actual} and the average logarithm of the sum of squared distances for the null data clusters. Determining the optimal number of K that provides a good balance between capturing meaningful patterns in the data and avoiding overfitting or underfitting.

With the voltage data collected during the experiment. The result was shown in Figure 6.12. 3 clusters achieve minimum gap. This verifies our classification clusters proposed based on physical analysis of the process. However, K-means algorithm avoids manually assigning the thresholds.
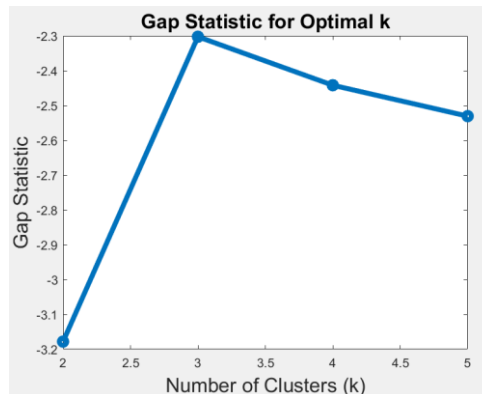
Figure 6.12 Gap statistics result for an optimal K

     With K-means algorithm performed on the dataset which $K = 3$. The result is

shown in Figure 6.13.

Figure 6.13 K-means clustering result
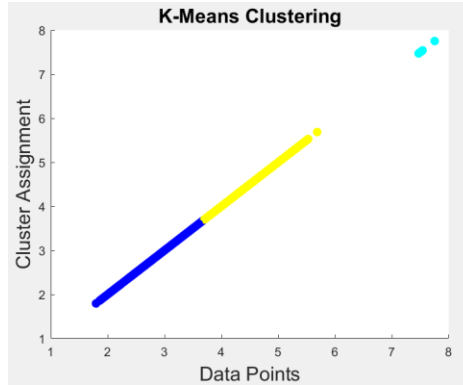
     The 3 Cluster Centroids were determined as 2.9, 4.5, and 7.5, where 2.9

corresponds to our desired arc state (parallel arc). With the new labeling categorize, the

identical model structure was trained using the same input data, while the labels substituted

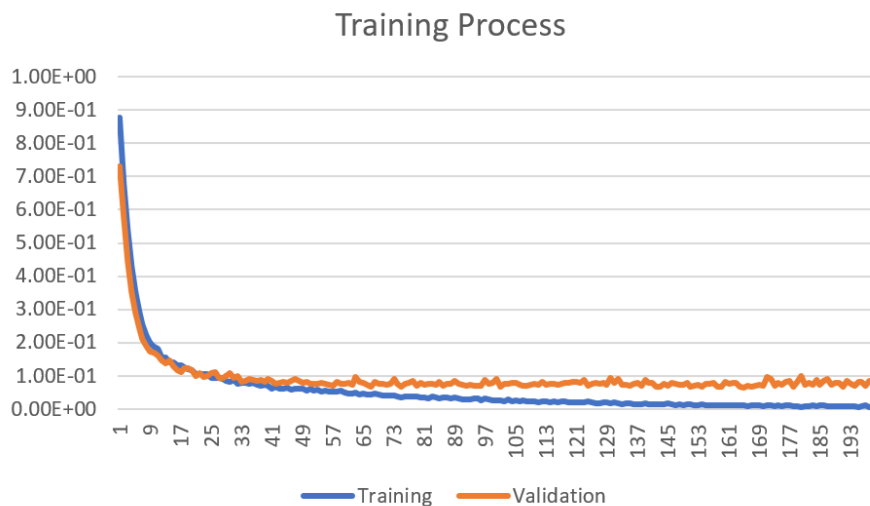from the K-means clustering results. The training curve is illustrated in Figure 6.14.

Figure 6.14 Training and validation curve

The validation error plot, following the format of Figure 11, is presented in Figure 6. 15.
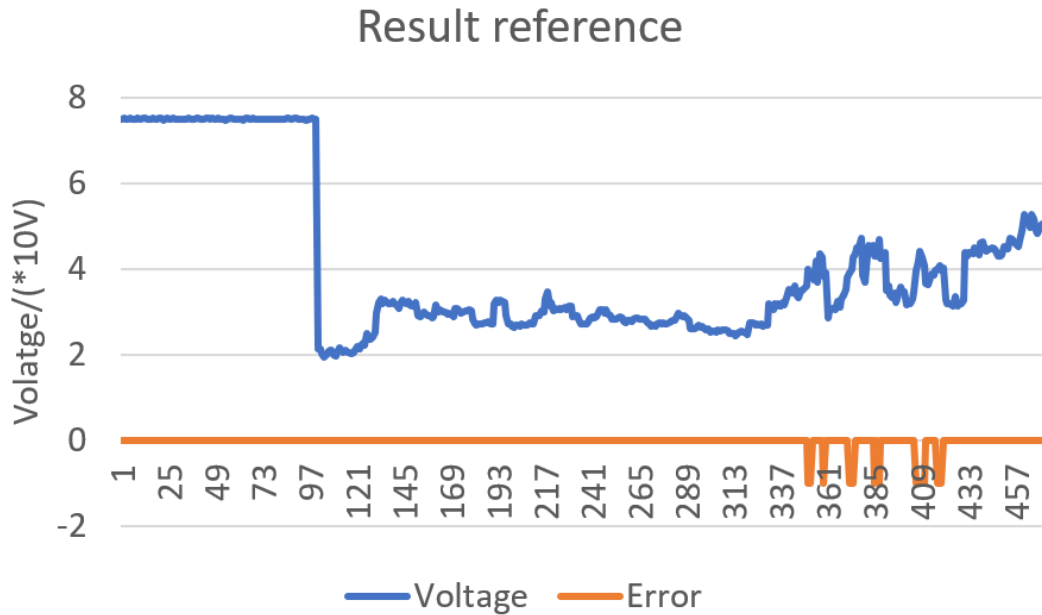

Result reference

Figure 6.15 Analysis of the prediction error in classification using K-means clustering based thresholds: where errors occur.

Just as in the case of model training using manually assigned threshold labels, the K-means based labeling also demonstrates good performance for the initial two segments of the data. Errors only occur when the voltage experiences substantial fluctuations. This highlights that an accurate classification model can be trained without the need for prior information about the welding power source, as evidenced by the fact that obtaining a sensible label grouping does not demand knowledge of the welding system's voltage/current.

6.6 Fine classification focused on desired mode

While the voltage represents the bypass arc state, a question may arise that the voltage here is a continuous signal, plot the histogram in the Figure 6.16 clearly shows that

while open arc state can be completely separable, parallel arc and serial arc has no significant boundary to separate them. Thus, this phenomenon implies that there is a mixed model where a portion of current flows through the bypass electrode, a portion of current go through the wire and flows to the workpieces, a portion of the current go through the wire, flows to the workpieces and form an arc between the bypass electrode and the workpieces (which is similar to a GTAW welding process). A careful inspection of the weld image proved out suspect, as illustrate in the Figure 6.17.
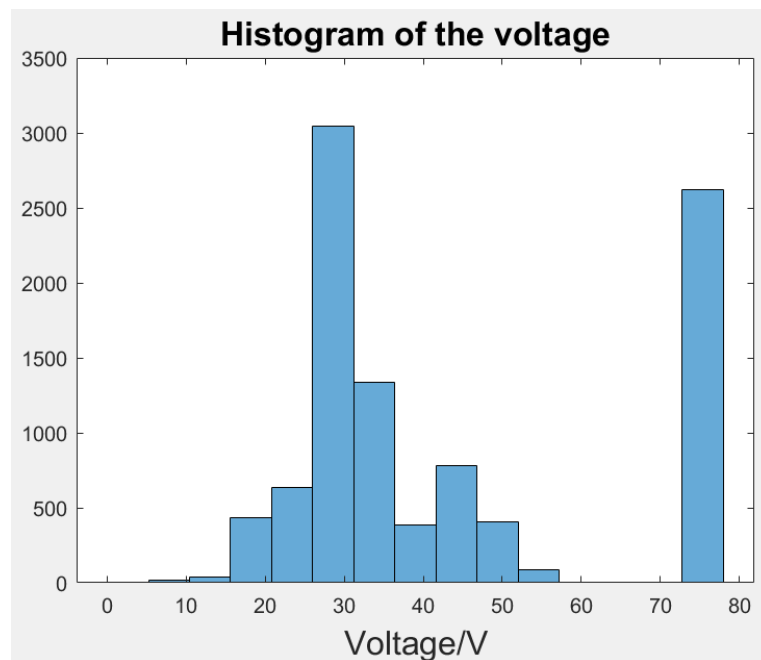

Figure 6.16 Histogram of bypass voltage measured across the wire and bypass electrode.

Figure 6.17 Examples of the mixed mode in DE-GMAW.

As the Serial Arc mode is the worst, it is beneficial to prevent a mixed mode from being accepted during supervisory monitoring. To this end, for the image that has been classified to and accepted as the Parallel Arc mode, we propose to pass the image also into a fine classification network. We note there are cases where we have the perfect Parallel Arc mode but the bypass electrode appears to be too close to the wire to cause a possible collision. As such, we propose to further classify the Parallel Arc mode, based V_2, into three sub-clusters: Too Close, Most Desired, and Low Confidence. Since the middle part of the voltage can be considered as the optimal state, the new dataset has been classified based on the voltage value, where the upper threshold is $minV_2^* + 2\Delta V_2$ and lower threshold is $inV_2^* + \Delta V_2$. We trained the same model with the new dataset. The result is shown in Figure 18. The prediction accuracy is approximately 80%.

It needs to be mentioned that although the performance is worse than the previous two model, this fine classification process represent an advanced attempt to classify the bypass arc within a very narrow segment, the input image share a lot similarity and label as the voltage only exhibited small difference. The attempt only to determine which state

could be considered as the most ideal arc state while the other two states also being

acceptable. This training process proved that the deep learning model has the ability to

identify the arc states barely different. The performance expected to be improved with an

increasing dataset.

Result reference
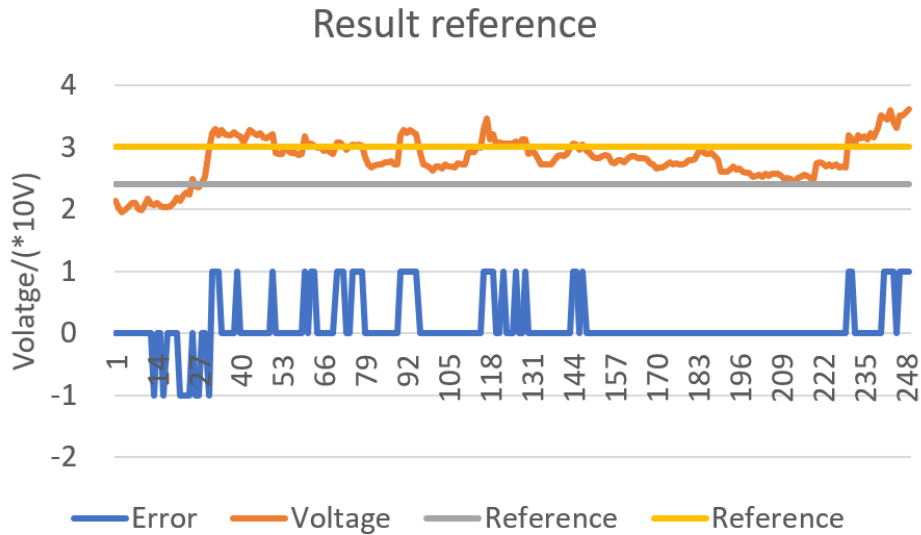
Figure 6.18 Analysis of the prediction error in fine classification within desired mode: where errors occur.

6.7   Summary

In this chapter, a DE-GMAW experiment system has been established, arc phenomenon

which happened during the DE-GMAW process been analyzed, a deep learning model is

proposed to distinguish the arc state based on the voltage data collected during the welding

process, the validation prove the effectiveness of trained model, the label from the voltage

data can be classified by human pre-knowledge or K-means clustering algorithm. The

obtained arc state will be utilized in the next chapter, to perform a DE-GMAW process

automatically with a stable bypass arc.

CHAPTER 7. DEEP-LEARNING BASED SUPERVISORY CONTROL OF ROBOTIZED DE-GMAW PROCESS

7.1 Introduction

As mentioned above, all the exist research about DE-GMAW process use a pre-fixed relationship between the main torch and the bypass torch. The placement of the bypass electrodes and main electrodes is established before the welds start. However, to learn and control the DE-GMAW process as a skilled human welder. To combine the human's adaptability and robot's precision, the DE-GMAW process needs to be robotized, i.e., to perform the process with a robot collaborate with another robot or a tractor to obtain the flexibility during the process. To achieve this goal. The skilled human welder's operation information needs to be collected. In this chapter, we enhance the DE-GMAW system with an IMU sensor to obtain comprehensive human operation information, overcome the IMU sensor's inaccurate problem with the inaccurate training approach we discussed in the previous chapter. Training a deep learning model to sense the welding view as a human welder. Control the process following the methodology generalized from the human.

7.2 Principles

7.2.1 Human movement sensing

Weld pool, as the most crucial area during the welding process attract most of the human welder's attention. Through carefully observation of the weld pool's phenomenon, human welder controls the torch to maintain the stable arc conditions to producing a high-quality weld. If considering the weld pool's image as the input information source to the human model, the output from the human will be torch movement information. Human welder precisely adjusts his/her arm posture to place the torch's tip at the optimal position

to ensure the weld stability. Among this process, the torch's tip position served as a visible connection between the weld pool's visualizing information and the human welder's physical manipulation. Hence, to understand the human control methodology, a robust correlation between the weld pool's visual information I and the torch's end point position P_{end} need to be established. The weld pool's visual information can be captured by an industrial camera and the torch's end point position can be obtained through the IMU sensor.
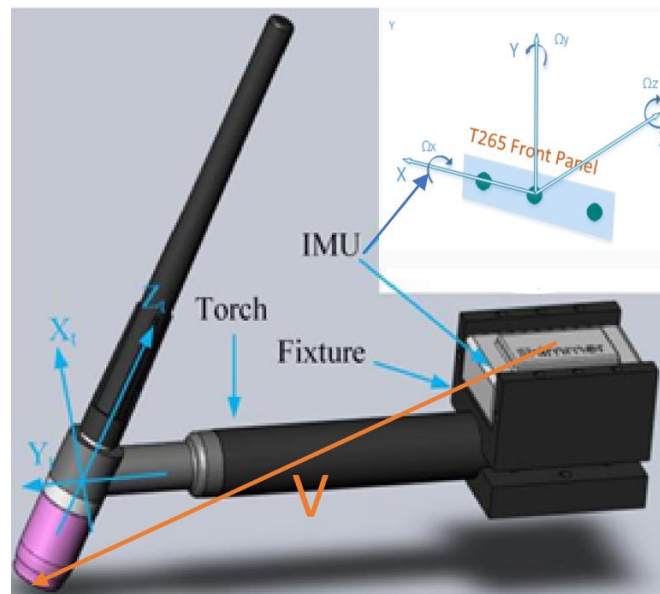
Figure 7.1 A welding torch equipped with IMU sensor

As illustrated in Figure 7.1. The showed GTAW torch has attached an IMU sensor on the topside of the handle as Intel Realsense T265. When the human welder manipulates this torch to perform the DEGMAW process, this sensor will provide rotation and movement information based on its intrinsic coordinate system. As the electrode used in the GTAW process is non-consumable, with geometry correlation, a relationship can be derived from the sensor intrinsic coordinate system to the electrode coordinate system, i.e., from the sensor's position to the torch's end point position:

$$P_{end} = R * V + T \tag{27}$$

Here, V is the constant vector from the sensor's coordinate system to the end position of the torch, T is the translation vector based on the sensor itself. P_{end} represent the end point position we desired, and R stands for the rotation matrix derived from the sensor's output quaternions (q_w, q_x, q_y, q_z):

$$R = \begin{bmatrix} 1 - 2 * q_y{}^2 - 2 * q_z{}^2 & 2 * q_x * q_y - 2 * q_z * q_w & 2 * q_x * q_z + 2 * q_y * q_w \\ 2 * q_x * q_y + 2 * q_z * q_w & 1 - 2 * q_x{}^2 - 2 * q_z{}^2 & 2 * q_y * q_z - 2 * q_x * q_w \\ 2 * q_x * q_z - 2 * q_y * q_w & 2 * q_y * q_z + 2 * q_x * q_w & 1 - 2 * q_x{}^2 - 2 * q_y{}^2 \end{bmatrix} \tag{25}$$

To solve this equation, quaternions (q_w, q_x, q_y, q_z) and translation vector T is readily available from the sensor. The V can be determined by utilizing its inherent feature that its constant. If we find a place to fix the tip of torch and rotate it. From the equation (24), the following equation should be satisfied:

$$\begin{aligned} R_1 * V + T_1 &= R_2 * V + T_2 \\ R_3 * V + T_3 &= R_4 * V + T_4 \\ &\cdots \\ R_{n-1} * V + T_{n-1} &= R_n * V + T_n \end{aligned} \tag{28}$$

Consider that:

$$\begin{aligned} (R_1 - R_2) * V &= T_2 - T_1 \\ (R_3 - R_4) * V &= T_4 - T_3 \\ &\cdots \\ (R_{n-1} - R_n) * V &= T_n - T_{n-1} \end{aligned} \tag{29}$$

If replace:

$$X = \begin{bmatrix} R_1 - R_2 \\ R_3 - R_4 \\ \cdots \\ R_{n-1} - R_n \end{bmatrix}, Y = \begin{bmatrix} T_2 - T_1 \\ T_4 - T_3 \\ \cdots \\ T_n - T_{n-1} \end{bmatrix} \tag{30}$$

With least square method, constant V can be solved by:

$$V = (X' * X)^{-1} * X' * Y \tag{31}$$

Once the accurate V is obtained, P_{end} can be solved through equation (24). Collect all the

P_{end} data generated during a DEGMAW welding process. A human welder trajectory can
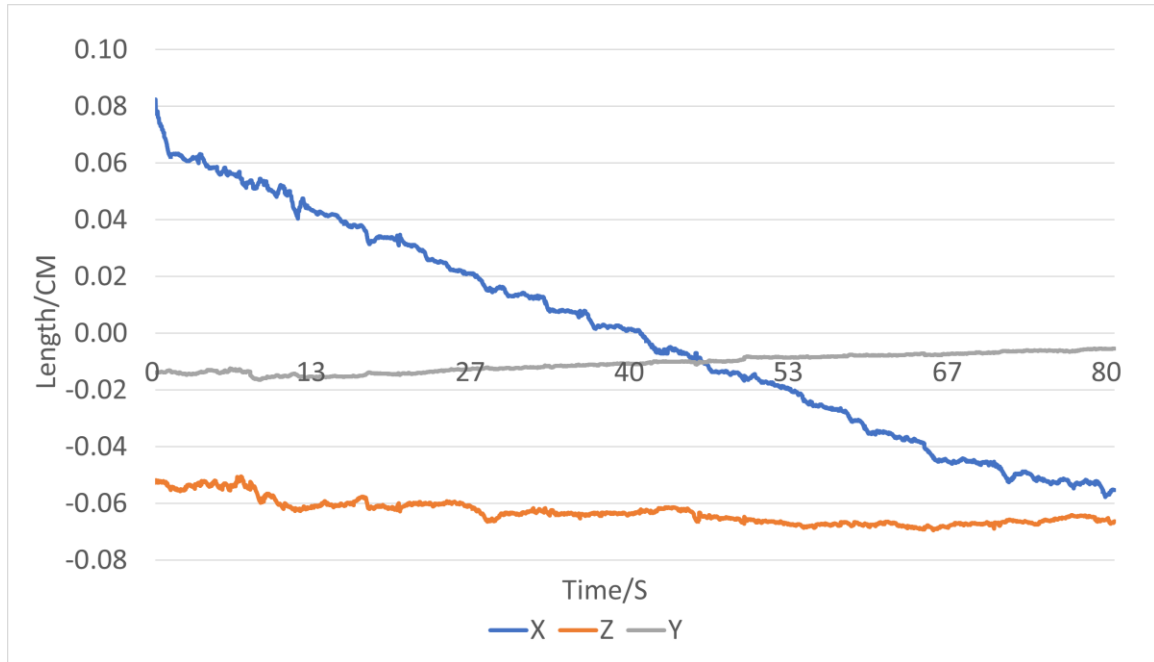
be retrieved shown in Figure 7.2.

Figure 7.2 Human welder position information from one DEGMAW experiment

In this data, the human welder manipulates the GTAW torch collaborating with a tractor

attached with a GMAW torch moving along the X direction. Following the tractor's fixed

speed, the human maintains a steady movement statue at X direction, while some minor

unintended shifts are observed in the Y and Z directions caused by human's natural

inconsistency.

As the torch's position and tractor's position both available, the distance between them

which decided by the human welder can be collected through $P_{tractor} - P_{end}$. As shown
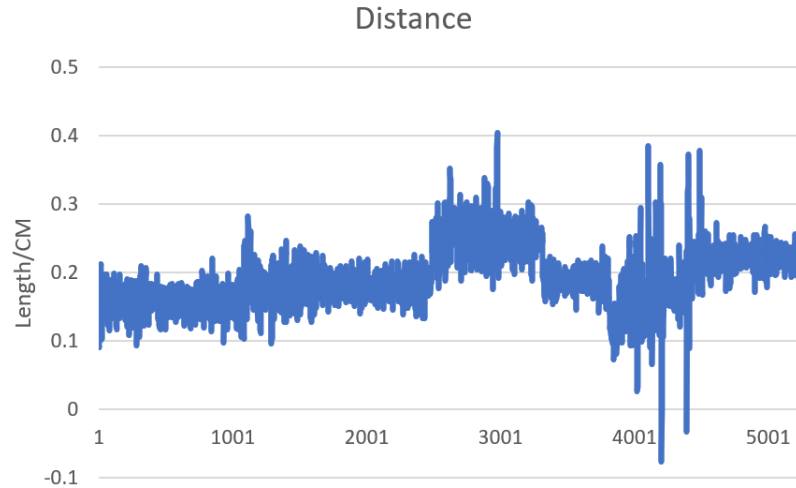
in Figure 7.3.

Figure 7.3 Distance between two torch

View the distance between the torches. Sensor's system error introduces a fluctuation around $[-0.5mm, 0.5mm]$ to the information. Control the robot to perform the welding task based on this level of accuracy is insufficient. On the one hand, welding images contain the information we need but it's hard to design feature-based algorithm and ensure its robustness following the conventional approach. One the other hand, deep learning model have the ability but needs a comprehensive dataset to finish the training process to overcome the overfitting issue, but as shown in the Figure 7.4, manual labeling the data may be feasible, however, it is proving to be time consuming and restrict the size of manually labeled data.

Therefore, considering the approach we discussed in the previous chapter, an effective solution can be proposed by pre-training an initial model with the inaccurate label generated from the sensor, after the pre-training process, a fine-tuning process will be performed with a small set of data which labeled manually. Through this process, all the data generated during the experiment will be utilized effectively and the final accuracy of the model will be improved.
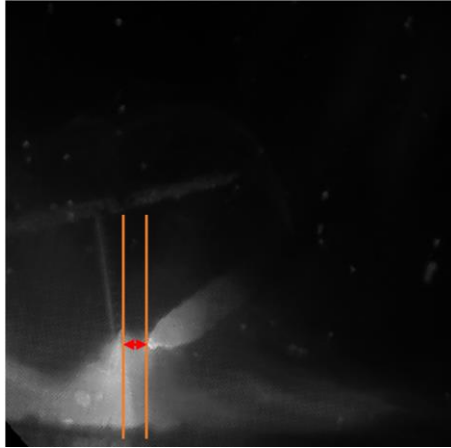
Figure 7.4 Manual labeling process

7.2.2　　Human movement sensing

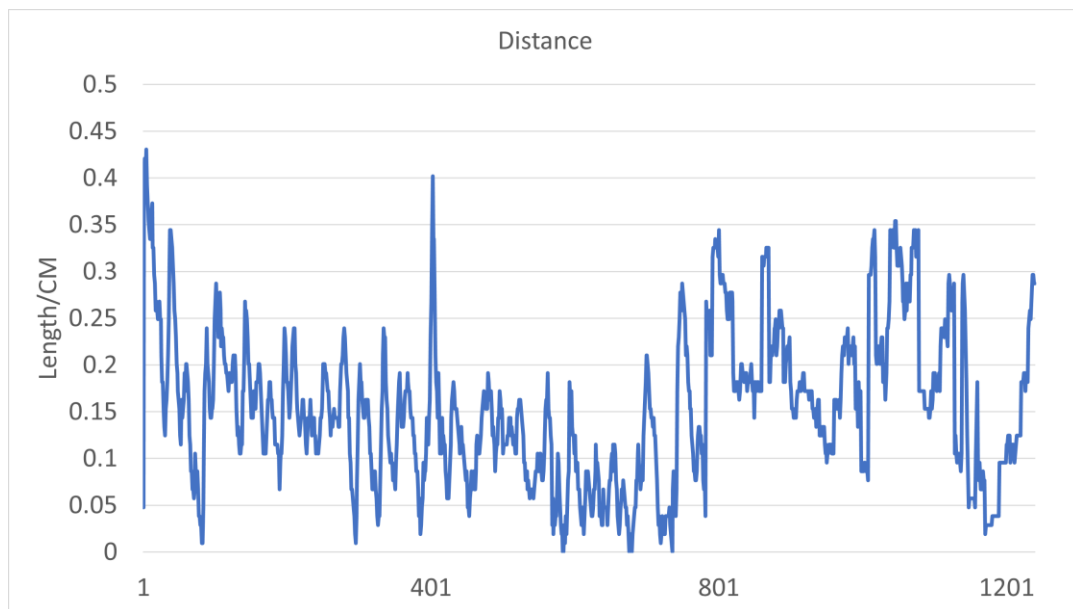Figure 7.5 illustrates a part of distance data labeled manually.

Figure 7.5 Manual label data

Analysis of the data from the human welder reveals a pattern of adjustments characterized by both overcorrection and subsequent correction. The optimal welding process is achieved when the human welder minimizes torch distance adjustments, maintaining a stable gap between the two torches.

Probability density is computed based on the dataset, as illustrated in Figure 7.6. The optimal distance between the two torches is established at 1.7mm.
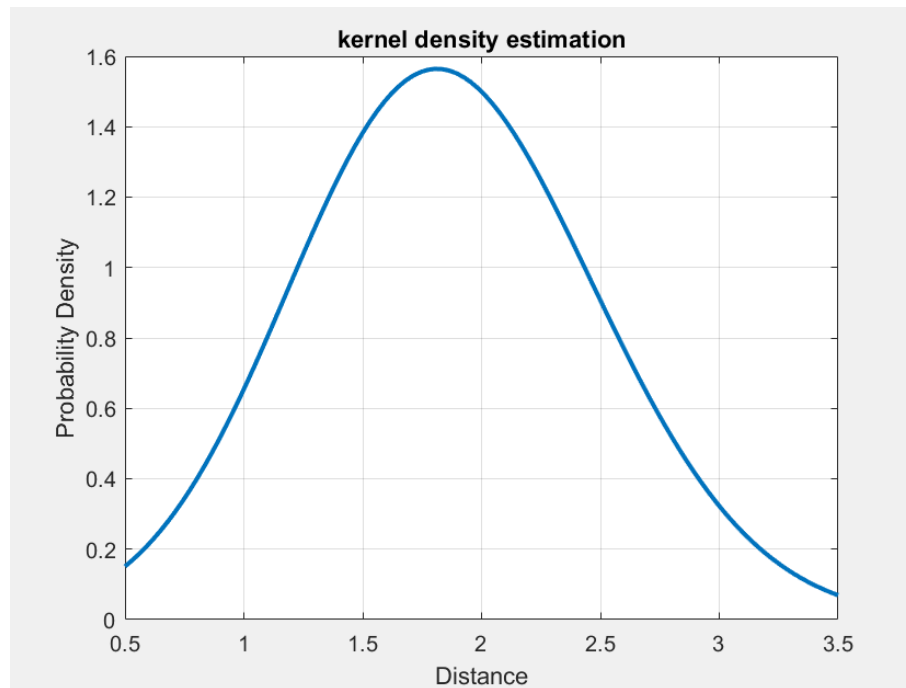
Figure 7.6 Probability density of human operation distance

7.3   Network and training

With the input image has size 256*256, the adopt CNN architecture consists of 5 convolution layers, each followed by a max pooling layer. In the end, the feature vector output from the last max pooling layer serve as input fed into 2 constitutive fully connected layer. Batch normalization and ReLU activation functions were performed between each convolution and max pooling layer. The parameters among 5 convolution layers were (1, 16, 5, 1, 2), (16, 32, 5, 1, 2), (32, 64, 3, 2, 1), (64, 128, 3, 2, 1), (128, 256, 3, 2, 1), separately. The last two fully connected layer reduce the feature dimensions from 256 to 128 and finally to 1.

The Training process which trains the model from a scratch to a pre-training model was carried on a NVIDIA GTX 2080 graphic card. With 200 times iteratively training and learning rate set as 0.001 at the first and reduce to 80% every 20 iteration, SGD optimizer was selected and mean square error loss was used to evaluate the training performance. Each train epoch will shuffle the dataset to ensure the sequence was randomly. 10504 data pairs were used for training and 480 data pair were used for evaluation. The training process was illustrated in Figure 7.7 and the validation result was shown in Figure 7.8.

Figure 7.7 Training and validation curve

General model
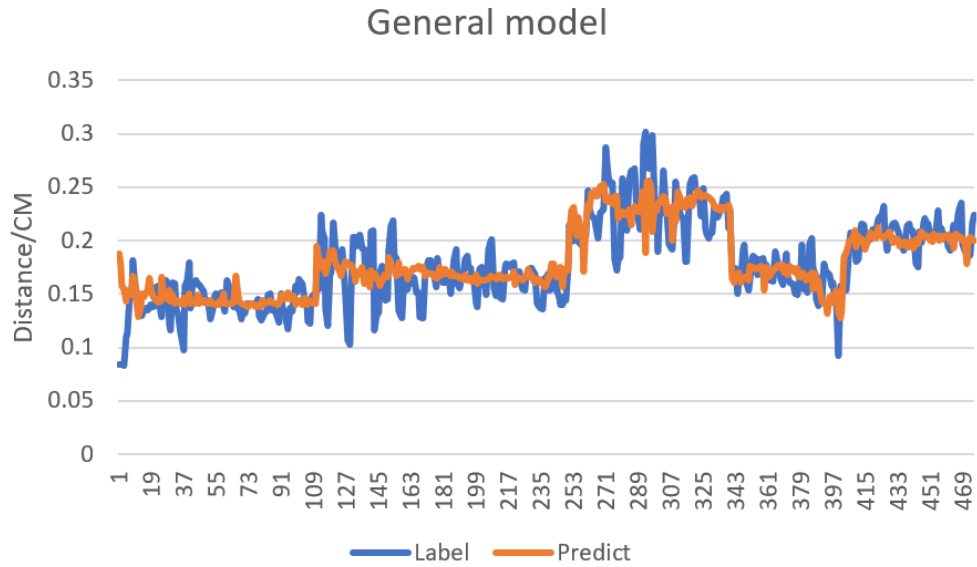
Figure 7.8 Validation result

After the model was initialized from the scratch. The model which has the minimum validation loss at epoch 26 was selected for the fine-tuning process. All the hyperparameter was the same as the pre-training process except the learning rate fixed at 0.0001. The model's convolution layer has been freezing and only update the two fully connected layers. With 1600 data pairs used for training and 93 data pair used for validation. The training process was show in Figure 7.9 and the validation result was plot in Figure 7.10.
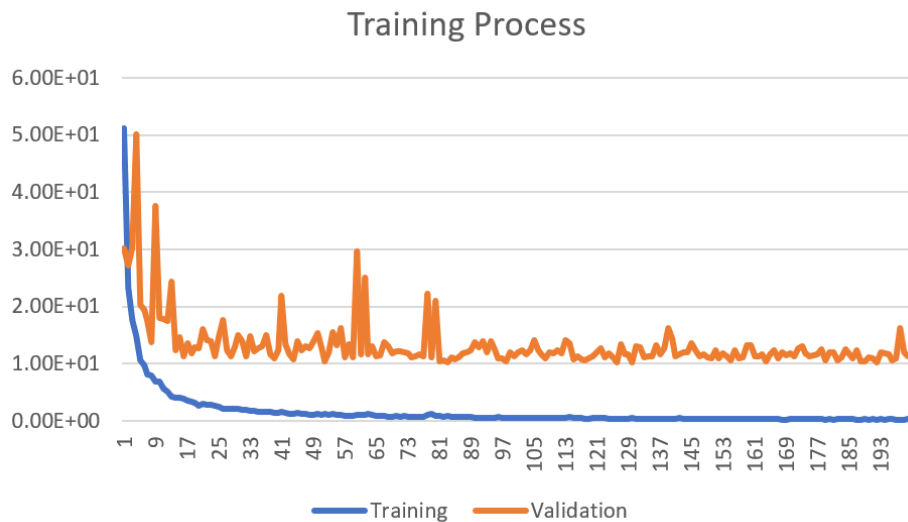
Training Process
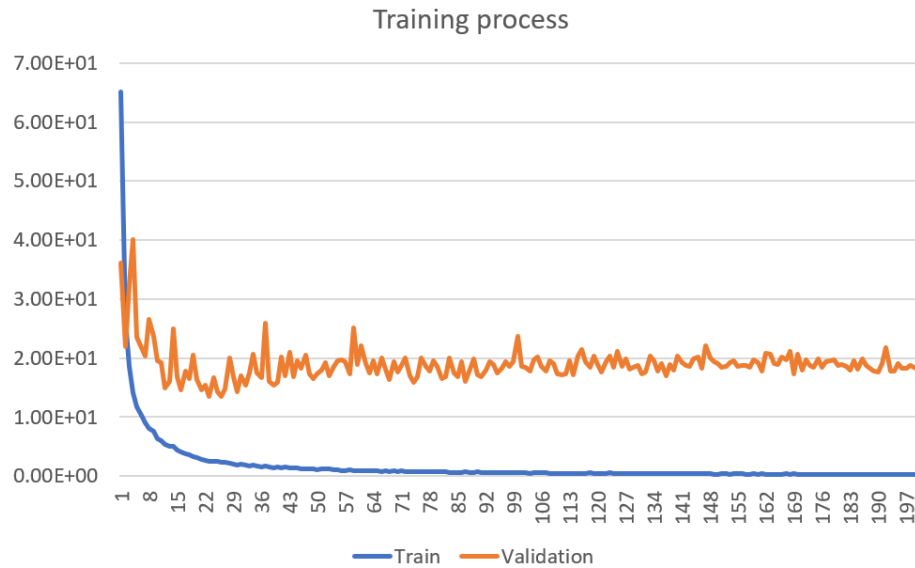
Figure 7.9 Training and validation curve for the fine-tuning process
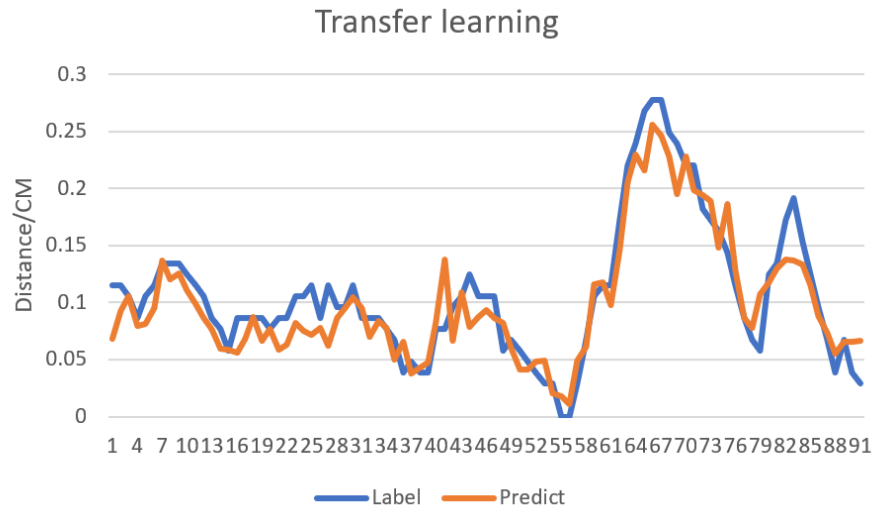
Figure 7.10 Validation result

Validation results proves that the distance could be accurately sensed with the model.

7.4 Robot welding

The model was then tested during the real welding process. The experiment was performed by a UR-15e robot collaborate with the tractor. The robot will replace the human task in the previous DEGMAW experiments. Following the control protocol shown in Figure 7.11.
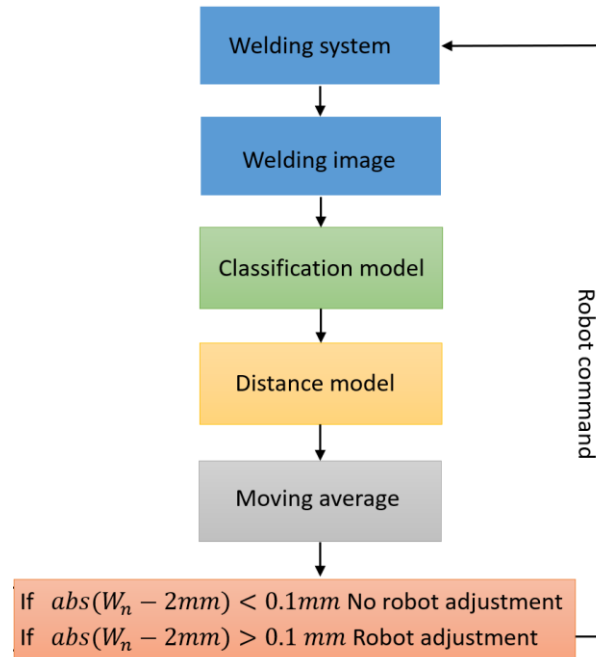
Figure 7.11 Robot control protocol

Before the weld start, the distance between the two torches was 1.1mm. The tractor first starts, the same as the model, and the moving information was sent to the robot synchronize. The robot moves at the same speed with the tractor. After the welds stable, which was approximately 4 seconds after arc ignite. The robot was intentional move away from the tractor to create a large gap between two torches. The output distance information has been processing by a moving average computation with the window size set as 20.

After the intentional move occur 2 seconds, the system grants the model the authority to direct the robot's actions, which the model was already identified that the gap had exceeded the desired threshold, it sends a command to move robot back to the desired position to hold the distance learning from the human welder. It needs to be mentioned that the model although the model can detect the distance instantly, we insert a 2 second delay to ensure a better visual assessment. This process repeats again after two seconds with the moving distance increase 1mm longer.

Figure 7.12 shows the distance between the two torches, while (a) is the distance output from the model during the welding process and (b) is the distance calculated from the sensor after the welds. The result proves that this system can successfully maintain the complex DEGMAW process without pre-fixing the torches relationship before the welding.

(a)

(b)
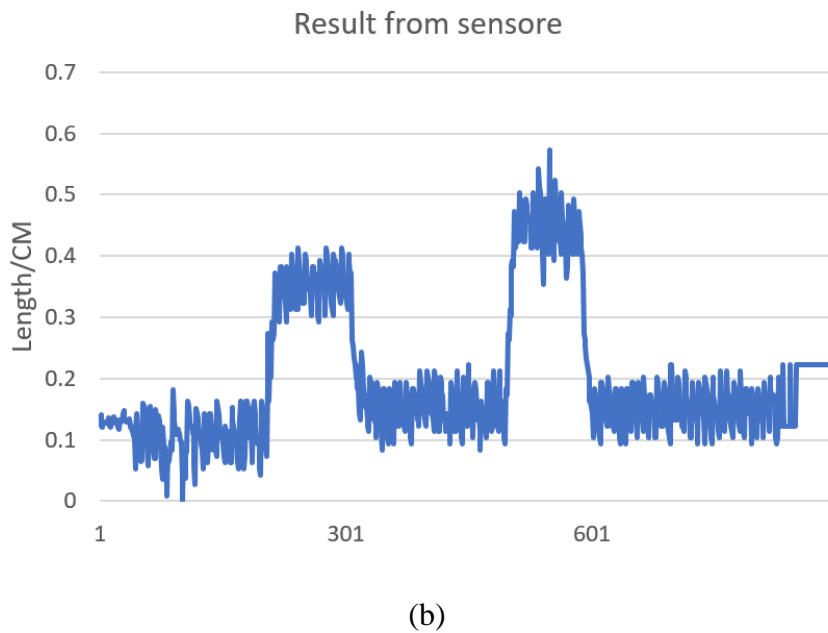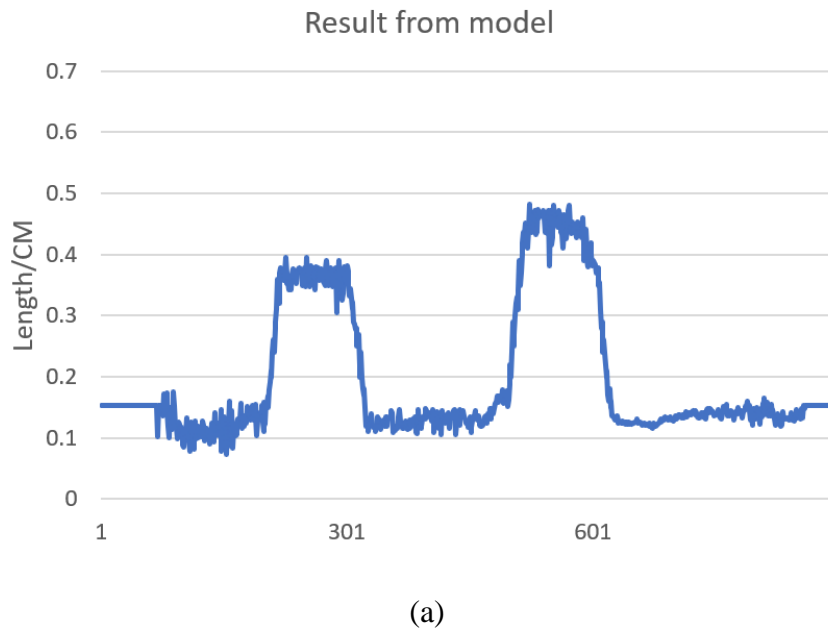
Figure 7.12 distance from the sensor and CNN model

7.5   Summary

Reviewing all the previous work focused on the DEGMAW process, the fixed torch relationship limits the adaptability. This chapter build a system to perform this complex welding process with more flexibility. The control strategy derives from an experienced human welder. This systems flexibility offers a promising solution for enhancing arc-based AM processes and their potential for automation and precision.

CHAPTER 8. CONCLUSION AND FUTURE WORKS

8.1   Conclusion

According to the American Welding Society, the welder shortage will reach a deficit of 400,000 workers by 2024 [1]. This suggest more welding position will have to be replaced by robotics. However, although most of the mass production with low unit profits has already been automated, complex welding process is still challenging to be robotized. To establish a scientific foundation to robotize the complex welding process which can only be handled by the human in the past. This work contributes in the following 3 aspect: (1) developing a robotized DE-GMAW welding system to collect comprehensive data generated during the welding process; (2) offer a strong analyzing/reasoning ability to replicate human welder expertise from data driven approach; (3) to learn from human welder, human heterogeneity issue needs to be considered, and this work proposes a solution to address this problem by training with inaccurate label first then followed by a fine-tuning process. The detailed contributions and achievements are summarized as following:

1. By utilizing image segmentation network, the weld pool's position can be obtained in real-time, which will allow the follower torch to follow the main torch automatically.

2. With the CNN-LSTM network, the criterion to justify the welding quality, back side bead width, can be predicted accurately, with the error achieved $0.3\,mm$, it is remarkably accurate for a welding process that involves a broad arc heat distribution and numerous factors affecting the heat transfer to the bottom of the workpiece.

3. Can a deep learning model be effectively trained using inaccurate labels is a major issue about utilizing human expertise knowledgebase. This work conducts a series of

experiments prove that as the mean error equal to zero. The model can be effectively trained following a fine-tuning process with a small portion of manual labeled data.

4. By developing a DE-GMAW welding system, the DE-GMAW welding process has first been performed by two separate operators. This platform allows the system learning from a skilled human welder, to steeply improve the performance of this complex welding system.

8.2   Future works

The DE-GMAW system currently can only maintain the stable bypass current by control the distance between the two torches at a fixed gap 0.17. This may not consider all the welding scenarios occured during the DE-GMAW process. Therefore, to have an adaptive model which can predict the next control action, we propose to use a CNN-LSTM model illustrated in Figure 8.1. The CNN structure is the same as the previous model while the LSTM structure has 3 layers with 256 as input size and 128 as hidden size.
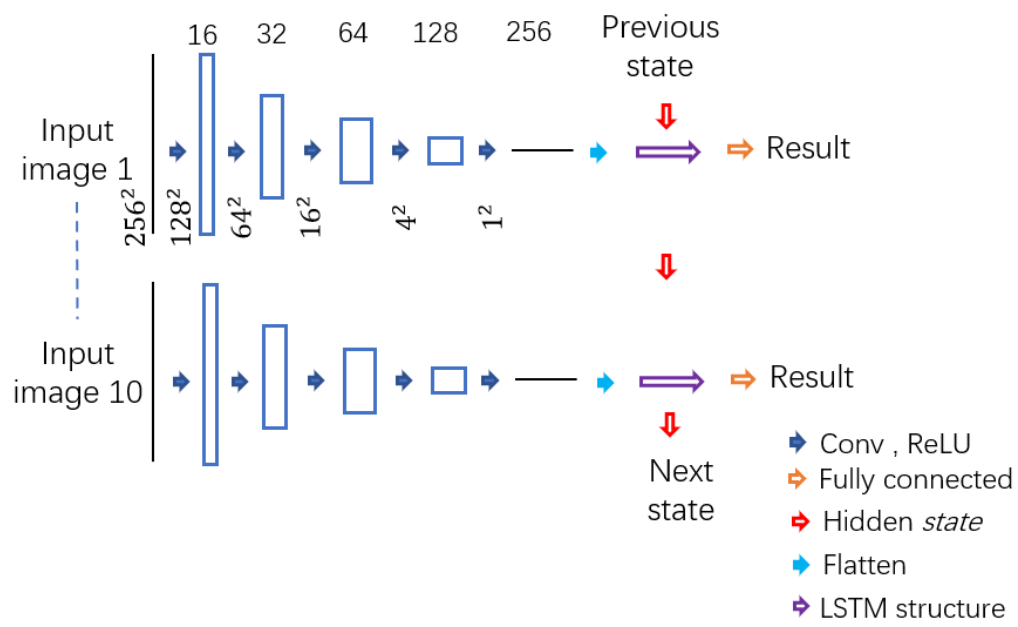
Figure 8.1 CNN-LSTM structure

Utilizing the LSTM structure, the model acquires the capability to forecast the subsequent movement action based on the historical trends. i.e., decide the current operation based on the welding pool state in the past. Inspect the human welder's velocity data which shown in Figure 8.2. It is obvious that is influenced by the inherent instability of the human body. Velocity information contains a significant amount of error, making it challenging for the training model to converge. Therefore, the distance still picked as reference to evaluate the control operation. The model takes the previous 10 images as input and predicts the current distance between two torches as the output. After training the model using the identical setup outlined in chapter 7, the validation results were presented in Figure 8.3. This demonstrates that the model can effectively replicate the behavior of a skilled human welder in maintaining the desired torch distance.
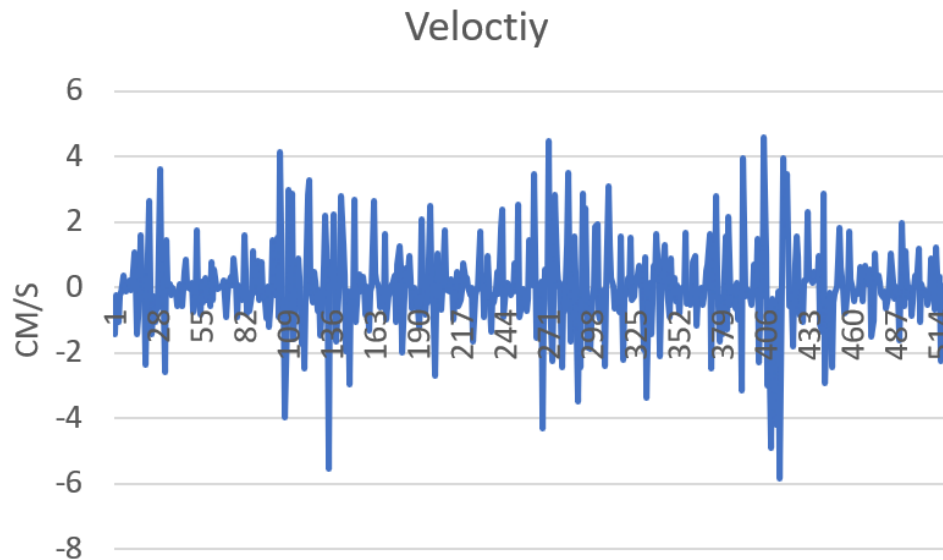
Figure 8.2 Human welder's velocity

Figure 8.3 Validation result

More work can be done to improve the current DE-GMAW system. Which involves not only ensuring a stable bypass arc but also enabling the adjustment of torch speed and orientation according to varying welding geometries. Additionally, we aim to manipulate the voltage from the GMAW power source and the current from the GTAW power source to accommodate different deposition rates.
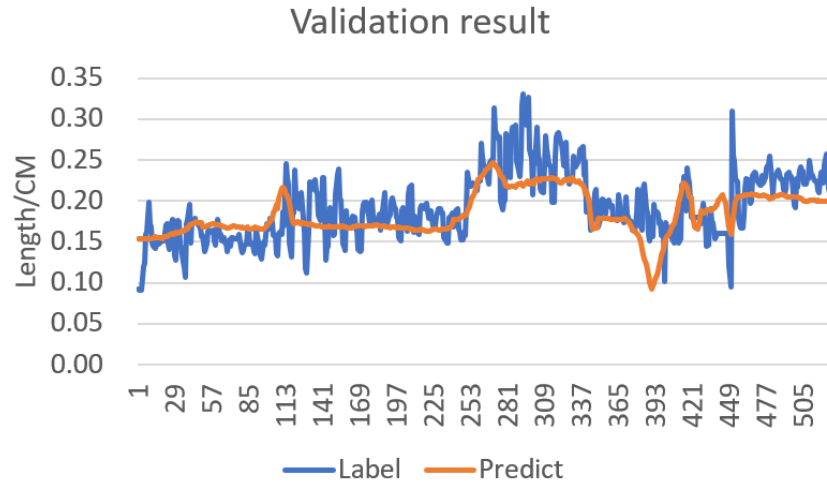
BIBLIOGRAPHY

[1] American Welding Society, www.aws.org

[2] Applied Market Research, Robotic Welding Market Statistics – 2016, https://www.alliedmarketresearch.com/robotic-welding-market

[3] 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, vol. 22, no. 3, pp. 955-966, 2014.

[4] Y. Liu and Y.M. Zhang, "Control of 3D weld pool surface," Control Engineering Practice, vol. 21, no. 11, pp. 1469-1480, 2013.

[5] K. Zhou and L. Cai, "Online nugget diameter control system for resistance spot welding," The International Journal of Advanced Manufacturing Technology, vol. 68, no. 9-12, pp. 2571-2588, 2013.

[6] P. Podržaj and S. Simončič, "Resistance spot welding control based on the temperature measurement," Science and Technology of Welding and Joining, vol. 18, no. 7, pp. 551-557, 2013.

[7] T. Purtonen, A. Kalliosaari, and A. Salminen, "Monitoring and adaptive control of laser processes," Physics Procedia, vol. 56, pp. 1218-1231, 2014.

[8] Z. Wang, "Monitoring of GMAW weld pool from the reflected laser lines for real-time control," IEEE Transactions on industrial informatics, vol. 10, no. 4, pp. 2073-2083, 2014.

[9] N. Lv, J. Zhong, H. Chen, T. Lin, and S. Chen, "Real-time control of welding penetration during robotic GTAW dynamical process by audio sensing of arc length," The International Journal of Advanced Manufacturing Technology, vol. 74, no. 1-4, pp. 235-249, 2014.

[10] Waszink, J.H. and Van den Heuvel, G.J.P.M., 1982. Heat generation and heat flow in the filler metal in GMA welding.

[11] Sun, T., Roy, M.J., Strong, D., Withers, P.J. and Prangnell, P.B., 2017. Comparison of residual stress distributions in conventional and stationary shoulder high-strength aluminum alloy friction stir welds. Journal of Materials Processing Technology, 242, pp.92-100.

[12] Kumar, N., Bhavsar, H., Mahesh, P.V.S., Srivastava, A.K., Bora, B.J., Saxena, A. and Dixit, A.R., 2022. Wire arc additive manufacturing–a revolutionary method in additive manufacturing. Materials Chemistry and Physics, 285, p.126144.

[13] Li, K.H., Chen, J.S. and Zhang, Y., 2007. Double-electrode GMAW process and control. WELDING JOURNAL-NEW YORK-, 86(8), p.231.

[14] LI, K., CHEN, J. and ZHANG, Y., 2007. Double-electrode GMAW process and control: A novel welding process adds a GTAW torch to a conventional GMAW system to create a bypass arc for increasing melting current while controlling base current. Welding journal, 86(8).

[15] D. Yang, G. Wang, and G. Zhang, "A comparative study of GMAW-and DE-GMAW-based additive manufacturing techniques: thermal behavior of the deposition process for thin-walled parts," The

[16] Zhang YM, Jiang M, Lu W. Double electrodes improve GMAW heat input control. Welding Journal (Miami, Fla) 2004;83(11):39–41.

[17] Li KH, Zhang YM. Consumable double-electrode GMAW – Part 1: the process. Welding Journal (Miami, Fla) 2008;87:11s–7s.

[18] Li KH, Zhang YM. Consumable double-electrode GMAW part II: monitoring, modeling, and control. Welding Journal 2008;87. p. 44-s-50-s.

[19] Ming, Z., Chunkai, L., Gang, Z., Yu, S., Ding, F. and Jiankang, H., 2016. Modeling and control of consumable DE-GMAW process. Journal of Manufacturing Processes, 24, pp.293-297.

[20] Lu, Y., 2014. Model Analysis and Predictive Control of Double Electrode Submerged Arc Welding Process for Fillet Joints with Root Opening.

[21] Zhang, S., Ma, G., Peng, X. and Xiang, Y., 2017. Numerical simulation of the effects of bypass current on droplet transfer during AZ31B magnesium alloy DE-GMAW process based on FLUENT. The International Journal of Advanced Manufacturing Technology, 90, pp.857-863.

[22] Yu, R., Kershaw, J., Wang, P. and Zhang, Y., 2022. How to accurately monitor the weld penetration from dynamic weld pool serial images using CNN-LSTM deep learning model?. IEEE Robotics and Automation Letters, 7(3), pp.6519-6525.

[23] Cheng, Y., Yu, R., Zhou, Q., Chen, H., Yuan, W. and Zhang, Y., 2021. Real-time sensing of gas metal arc welding process–A literature review and analysis. Journal of Manufacturing Processes, 70, pp.452-469.

[24] Xiao, Y.H. and Den Ouden, G., 1990. A study of GTA weld pool oscillation. Weld. J., 69(8), p.289.

[25] K. Andersen and G. E. Cook, "Synchronous weld pool oscillation for monitoring and control," IEEE Trans. Ind. Appl., vol. 33, no. 2, pp. 464–471, 1997.

[26] A. J. R. Aendenroomer and G. Den Ouden, "Weld pool oscillation as a tool for penetration sensing during pulsed GTA welding," Weld. J., vol. 77, no. 5, pp. 181s-187s, 1998.

[27] Nagarajan, S., Chen, W.H. and Chin, B.A., 1989. Infrared sensing for adaptive arc welding. Welding Journal, 68(11), pp.462-466.

[28] Chen, W. and Chin, B.A., 1990. Monitoring joint penetration using infrared sensing techniques. Welding Journal, 69(4), pp.181s-185s.

[29] Seppala, J.E. and Migler, K.D., 2016. Infrared thermography of welding zones produced by polymer extrusion additive manufacturing. Additive manufacturing, 12, pp.71-76.

[30] Lin, Z.B., Azarmi, F., Al-Kaseasbeh, Q., Azimi, M. and Yan, F., 2015. Advanced ultrasonic testing technologies with applications to evaluation of steel bridge welding-an overview. Applied mechanics and materials, 727, pp.785-789.

[31] Petrov, P., Georgiev, C. and Petrov, G., 1998. Experimental investigation of weld pool formation in electron beam welding. Vacuum, 51(3), pp.339-343.

[32] Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), pp.679-698.

[33] Singhal, T.S., Jain, J.K., Kumar, M. and Saxena, K.K., 2022. Effect of filler wire preheating and nozzle cooling with advanced submerged arc welding process on bead geometry and microstructure. Advances in Materials and Processing Technologies, 8(sup2), pp.504-518.

[34] Liu, Y., Zhang, W. and Zhang, Y., 2013. Dynamic neuro-fuzzy-based human intelligence modeling and control in GTAW. IEEE Transactions on Automation Science and Engineering, 12(1), pp.324-335.

[35] Zheng, M., Wei, L., Chen, J., Zhang, Q., Zhong, C., Lin, X. and Huang, W., 2019. A novel method for the molten pool and porosity formation modelling in selective laser melting. International Journal of Heat and Mass Transfer, 140, pp.1091-1105.

[36] N. M. Carlson and J. A. Johnson, "Ultrasonic sensing of weld pool penetration," Weld. J., vol. 67, no. 11, pp. 239s-246s, 1988.

[37] B. Mi and C. Ume, "Real-time weld penetration depth monitoring with laser ultrasonic sensing system," J. Manuf. Sci. Eng., vol. 128, no. 1, pp. 280–286, Feb. 2006.

[38] Ma, Z., Zhuang, M. and Li, M., 2020. Effect of main arc voltage on arc behavior and droplet transfer in tri-arc twin wire welding. Journal of Materials Research and Technology, 9(3), pp.4876-4883.

[39] Wang, X., 2014. Three-dimensional vision-based sensing of GTAW: a review. The International Journal of Advanced Manufacturing Technology, 72, pp.333-345.

[40] Chu, Y.X., Hu, S.J., Hou, W.K., Wang, P.C. and Marin, S.P., 2004. Signature analysis for quality monitoring in short-circuit GMAW. Welding journal, 83(12), pp.336s-343s.

[41] Wang, Q., Jiao, W., Wang, P. and Zhang, Y., 2021. A tutorial on deep learning-based data analytics in manufacturing through a welding case study. Journal of Manufacturing Processes, 63, pp.2-13.

[42] Jena, D.P. and Kumar, R., 2011. Implementation of wavelet denoising and image morphology on welding image for estimating HAZ and welding defects. Measurement Science Review, 11(4), pp.108-111.

[43] Zou, Y., Wei, X. and Chen, J., 2020. Conditional generative adversarial network-based training image inpainting for laser vision seam tracking. Optics and Lasers in Engineering, 134, p.106140.

[44] Li, C., Wang, Q., Jiao, W., Johnson, M. and Zhang, Y.M., 2020. Deep learning-based detection of penetration from weld pool reflection images. Weld J, 99(9), pp.239S-245S.

[45] Přibyl, B., Chalmers, A., Zemčík, P., Hooberman, L. and Čadík, M., 2016. Evaluation of feature point detection in high dynamic range imagery. Journal of Visual Communication and Image Representation, 38, pp.141-160.

[46] Bacioiu, D., Melton, G., Papaelias, M. and Shaw, R., 2019. Automated defect classification of Aluminium 5083 TIG welding using HDR camera and neural networks. Journal of Manufacturing Processes, 45, pp.603-613.

[47] Cui, Y.X., Shi, Y.H., Ning, Q., Chen, Y.K. and Zhang, B.R., 2021. Investigation into keyhole-weld pool dynamic behaviors based on HDR vision sensing of real-time K-TIG welding process through a steel/glass sandwich. Advances in Manufacturing, 9, pp.136-144.

[48] Yu, R., Kershaw, J., Wang, P. and Zhang, Y., 2021. Real-time recognition of arc weld pool using image segmentation network. Journal of Manufacturing Processes, 72, pp.159-167.

[49] Xia, C., Pan, Z., Fei, Z., Zhang, S. and Li, H., 2020. Vision based defects detection for Keyhole TIG welding using deep learning with visual explanation. Journal of Manufacturing Processes, 56, pp.845-855.

[50] Pak, M. and Kim, S., 2017, August. A review of deep learning in image recognition. In 2017 4th international conference on computer applications and information processing technology (CAIPT) (pp. 1-3). IEEE.

[51] Otter, D.W., Medina, J.R. and Kalita, J.K., 2020. A survey of the usages of deep learning for natural language processing. IEEE transactions on neural networks and learning systems, 32(2), pp.604-624.

[52] Gaikwad, S.K., Gawali, B.W. and Yannawar, P., 2010. A review on speech recognition technique. International Journal of Computer Applications, 10(3), pp.16-24.

[53] Vorontsov, E., Trabelsi, C., Kadoury, S. and Pal, C., 2017, July. On orthogonality and learning recurrent networks with long term dependencies. In International Conference on Machine Learning (pp. 3570-3578). PMLR.

[54] Ying, X., 2019, February. An overview of overfitting and its solutions. In Journal of physics: Conference series (Vol. 1168, p. 022022). IOP Publishing.

[55] Cheng, Y., Wang, D., Zhou, P. and Zhang, T., 2017. A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282.

[56] Sharma, S., Sharma, S. and Athaiya, A., 2017. Activation functions in neural networks. Towards Data Sci, 6(12), pp.310-316.

[57] Shinohara, R.T., Sweeney, E.M., Goldsmith, J., Shiee, N., Mateen, F.J., Calabresi, P.A., Jarso, S., Pham, D.L., Reich, D.S., Crainiceanu, C.M. and Alzheimer's Disease Neuroimaging Initiative, 2014. Statistical normalization techniques for magnetic resonance imaging. NeuroImage: Clinical, 6, pp.9-19.

[58] Labach, A., Salehinejad, H. and Valaee, S., 2019. Survey of dropout methods for deep neural networks. arXiv preprint arXiv:1904.13310.

[59] Mazilu, S. and Iria, J., 2011, December. L1 vs. l2 regularization in text classification when learning from labeled features. In 2011 10th international conference on machine learning and applications and workshops (Vol. 1, pp. 166-171). IEEE.

[60] Bentoumi, M., Daoud, M., Benaouali, M. and Taleb Ahmed, A., 2022. Improvement of emotion recognition from facial images using deep learning and early stopping cross validation. Multimedia Tools and applications, 81(21), pp.29887-29917.

[61] Rebuffi, S.A., Gowal, S., Calian, D.A., Stimberg, F., Wiles, O. and Mann, T.A., 2021. Data augmentation can improve robustness. Advances in Neural Information Processing Systems, 34, pp.29935-29948.

[62] Jiao, W., Wang, Q., Cheng, Y. and Zhang, Y., 2021. End-to-end prediction of weld penetration: A deep learning and transfer learning based method. Journal of Manufacturing Processes, 63, pp.191-197.

[63] Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), pp.3212-3232.

[64] Haralick, R.M. and Shapiro, L.G., 1985. Image segmentation techniques. Computer vision, graphics, and image processing, 29(1), pp.100-132.

[65] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[66] Lydia, A. and Francis, S., 2019. Adagrad—an optimizer for stochastic gradient descent. Int. J. Inf. Comput. Sci, 6(5), pp.566-568.

[67] Zeiler, M.D., 2012. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

[68] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[69] Amari, S.I., 1972. Learning patterns and pattern sequences by self-organizing nets of threshold elements. IEEE Transactions on computers, 100(11), pp.1197-1206.

[70] Hossain, M.Z., Sohel, F., Shiratuddin, M.F. and Laga, H., 2019. A comprehensive survey of deep learning for image captioning. ACM Computing Surveys (CsUR), 51(6), pp.1-36.

[71] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.

[72] Chung, J., Gulcehre, C., Cho, K. and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[73] Erden, M.S. and Marić, B., 2011. Assisting manual welding with robot. Robotics and Computer-Integrated Manufacturing, 27(4), pp.818-828.

[74] Chao Li, H., Ming Gao, H. and Wu, L., 2007. Teleteaching approach for sensor-based arc welding telerobotic system. Industrial Robot: An International Journal, 34(5), pp.423-429.

[75] Liang, Z., Gao, H., Nie, L. and Wu, L., 2007, August. 3D reconstruction for telerobotic welding. In 2007 International Conference on Mechatronics and Automation (pp. 475-479). IEEE.

[76] Erden, M.S. and Billard, A., 2015. Robotic assistance by impedance compensation for hand movements while manual welding. IEEE transactions on cybernetics, 46(11), pp.2459-2472.

[77] Zhang, T., McCarthy, Z., Jow, O., Lee, D., Chen, X., Goldberg, K. and Abbeel, P., 2018, May. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5628-5635). IEEE.

[78] Argall, B.D., Chernova, S., Veloso, M. and Browning, B., 2009. A survey of robot learning from demonstration. Robotics and autonomous systems, 57(5), pp.469-483.

[79] Zinchenko, K., Wu, C.Y. and Song, K.T., 2016. A study on speech recognition control for a surgical robot. IEEE Transactions on Industrial Informatics, 13(2), pp.607-615.

[80] Du, G., Zhang, P. and Liu, X., 2016. Markerless human–manipulator interface using leap motion with interval Kalman filter and improved particle filter. IEEE Transactions on Industrial Informatics, 12(2), pp.694-704.

[81] Zhou, M.A. and Ben-Tzvi, P., 2014. RML glove—An exoskeleton glove mechanism with haptics feedback. IEEE/Asme Transactions on mechatronics, 20(2), pp.641-652.

[82] Wang, Q., Cheng, Y., Jiao, W., Johnson, M.T. and Zhang, Y., 2019. Virtual reality human-robot collaborative welding: a case study of weaving gas tungsten arc welding. Journal of Manufacturing Processes, 48, pp.210-217.

[83] Wang, Q., Jiao, W., Yu, R., Johnson, M.T. and Zhang, Y., 2019. Virtual reality robot-assisted welding based on human intention recognition. IEEE Transactions on Automation Science and Engineering, 17(2), pp.799-808.

[84] Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

[85] Russell, B.C., Torralba, A., Murphy, K.P. and Freeman, W.T., 2008. LabelMe: a database and web-based tool for image annotation. International journal of computer vision, 77, pp.157-173.

[86] Wang, Q., Jiao, W. and Zhang, Y., 2020. Deep learning-empowered digital twin for visualized weld joint growth monitoring and penetration control. Journal of Manufacturing Systems, 57, pp.429-439.

[87] Wang, Z., Li, L., Chen, H., Wu, X., Dong, Y., Tian, J. and Zhang, Q., 2023. Penetration recognition based on machine learning in arc welding: a review. The International Journal of Advanced Manufacturing Technology, 125(9-10), pp.3899-3923.

[88] Liang, R., Yu, R., Luo, Y. and Zhang, Y., 2019. Machine learning of weld joint penetration from weld pool surface using support vector regression. Journal of Manufacturing Processes, 41, pp.23-28.

[89] Rabiner, L. and Juang, B., 1986. An introduction to hidden Markov models. ieee assp magazine, 3(1), pp.4-16.

[90] Jiao, W., Wang, Q., Cheng, Y., Yu, R. and Zhang, Y., 2020. Prediction of weld penetration using dynamic weld pool arc images. Weld J, 99(11), pp.295-302.

[91] Chen, J., Chen, J., Zhang, K., Feng, Z. and Zhang, Y.M., 2018. Dynamic reflection behaviors of weld pool surface in pulsed GTAW. Welding Journal, 97.

[92] Li, K. and Zhang, Y., 2007. Metal transfer in double-electrode gas metal arc welding. Journal of Manufacturing Science and Engineering, 129(6): 991-999.

[93] Dai, J., Li, Y., He, K. and Sun, J., 2016. R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29.

[94] Lloyd, S., 1982. Least squares quantization in PCM. IEEE transactions on information theory, 28(2), pp.129-137.

[95] Tibshirani, R., Walther, G. and Hastie, T., 2001. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2), pp.411-423.

VITA

RUI YU

Education:

B.E. in Information Engineering at Xi'an University of Posts and Telecommunications, Shaanxi, China

M.E. in Electrical and Computer Engineering at University of Kentucky, Kentucky, U.S

Publications in pursuing Ph.D.:

1. **Rui Yu**, Joseph Kershaw, Peng Wang, YuMing Zhang*. "Real-time recognition of arc weld pool using image segmentation network" Journal of Manufacturing Processes, 72: 159-167, 2021.

2. **Rui Yu**, Joseph Kershaw, Peng Wang, YuMing Zhang*. "How to Accurately Monitor the Weld Penetration From Dynamic Weld Pool Serial Images Using CNN-LSTM Deep Learning Model" in IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 6519-6525, 2022.

3. **Rui Yu**, Joseph Kershaw, Peng Wang, YuMing Zhang*. "Monitoring of Backside Weld Bead Width from High Dynamic Range Images Using CNN Network" In 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), 2022.

4. **Rui Yu**, Yue Cao, Heping Chen, Qiang Ye, YuMing Zhang*. "Deep learning based real-time and in-situ monitoring of weld penetration: Where we are and what are needed revolutionary solutions" Journal of Manufacturing Processes. 93: 15-46, 2023.