Generative Adversarial Networks (GAN) Model for Dynamically Adjusted Weld Pool Image toward Human-Based Model Predictive Control (MPC)

Tianpu Li[1], Yue Cao[1], Qiang Ye[2], YuMing Zhang[1*]

1: Department of Electrical and Computer Engineering and Institute for Sustainable Manufacturing, University of Kentucky, Lexington, KY 40506, USA
2: Department of Mathematics, University of Kentucky, Lexington, KY 40506, USA
* yuming.zhang@uky.edu

Abstract: Gas Metal Arc Welding (GMAW) is a critical industrial technique known for its high productivity, flexibility, and adaptability to automation. Despite the significant advancements in robotic welding, challenges remain in fully automating the arc welding process, particularly due to the complex dynamics of the weld pool associated with GMAW. A human-robot collaborative (HRC) system where humans operate robots may conveniently provide the needed adaptive control to the complex GMAW. While in conventional HRC systems humans receive process feedback to make adaptive adjustments, we propose provide humans with predictive future feedback to further ease the human decision and reduce the needed skills/trainings. To this end, this study explores the integration of deep learning models, specifically Generative Adversarial Networks (GANs) combined with Gated Recurrent Units (GRUs), to model and predict the dynamic behavior of the weld pool during GMAW. By leveraging time-series data of torch movement and corresponding weld pool images, the proposed GRU-GAN model generates high-fidelity weld pool images, capturing the intricate relationship between speed variations and weld pool morphology. Through extensive experimentation, including the design of an acceptable Encoder-Decoder structure for the GAN, we demonstrate that incorporating both temporal and speed sequence information significantly enhances the model's predictive capabilities. The findings validate the hypothesis that dynamic torch speed adjustments, akin to those performed by skilled human welders, can be effectively modeled to improve the quality of automated welding processes. Future work will be devoted to human-based model predictive control (MPC) in an HRC environment.

Keywords: Deep Learning, GAN, GRU, Welding, Weld Pool, GMAW.

1. Introduction

According to the International Federation of Robotics, "50% of the world's robots are used for welding. Specifically, 33% are used for spot welding, 16% perform arc welding, and 1% perform other types of welding operations"[1] . While this clearly indicates that welding is a major application of robots, it also suggests that the robotization of arc welding processes lags behind, as there are far more arc welding operations than spot welding ones. This is primarily due to the lack of generalizable methods for robots to intelligently adapt to the process to ensure the quality of the welds produced. However, due to the slower-changing dynamics of arc welding processes compared to spot welding, skilled human welders are able to make adaptive adjustments to overcome the effect from process uncertainty to produce quality welds through observing the process[2]. Human-robot collaboration (HRC) combining robot advantages with human welder flexibility and cognitive skills [3] provides a generalizable method to robotize arc welding through human making high-level decision while robot performing high-precision and adaptive welding based on this decision. To

be effective, humans are provided with immersive process feedback so that they can observe the process as at the site [4]. For gas tungsten arc welding (GTAW) where the process is less fluctuating, weld pool image provides an immersive feedback [5] so that HRC has been effectively used to robotize the process.

Gas metal arc welding (GMAW) is much more dynamic than GTAW and its fine control is difficult even for skilled welders. As such, despite its high productivity, GMAW is used for applications where the required welds may be made without fine adaptive control. For critical applications, GTAW is currently the primary choice despite the lower productivity. To extend GMAW and other highly dynamic welding processes beyond their current capability in making high quality welds, we propose to provide humans with predictive immersive process feedback so that the human can perform predictive control similar like a model predictive control which has proved to be effective in many applications [6,7]. To this end, we propose to generate future process feedback. For GMAW, this will be the future weld pool images.

We can consider an arc welding process as a dynamic system $y[t] = f(y[t-1], \dots y[t-p], u[t-1], \dots u[t-q]) \triangleq f(Y[t-1], U[t-1])$ where the process output y is the weld pool (image). The input u is the welding parameters that are adaptively adjusted. Welding conditions and other welding parameters that are not intentionally adjusted during welding process are reflected in the model f. For a human operated GMAW process, the wire feed speed which dictates the welding current I [8] and welding voltage V are pre-determined, thus fixed during welding process. The arc power $P = VI\ (J/s)$ and wire melting speed $\dot{M}$ (g/s) are thus considered not adjustable during the process. As the welding process is primarily controlled by the heat input $h = P/v$ (J/mm), the heat supplied in the workpiece along a unit length of weld, and the mass input $m = \dot{M}/v$ (g/mm), the welding speed v is the primary variable to realize the real-time control. Its adjustment can be conveniently realized by the welder through manipulating the welding torch. We note that human welder can also manipulate the torch to change the torch angle to provide more complex control on the process. In this study, we will only adjust the welding speed with the torch being vertical to the workpiece similar as in most robotic GMAW processes. As such, the input will be $u = v$.

We envision that in an HRC welding system, a human observes the welding process through the real-time captured weld pool image and adjusts a virtual tool to command the robot to change the welding speed. In addition to this typical way of HRC, we propose that the system also predicts the future weld pool (image) per the change in the virtual tool (welding speed). If the future weld pool is not desirable (is not in the desired trajectory to converge to the desired output y^*), the human will keep his/her fine adjustment. In this way, the human performs a model predictive control on a complex process whose output (weld pool image) is unconventional. The future output as a function of the input is predicted through the generated future weld pool image. The optimized input corresponding to the desirable future weld pool is realized by the human through stopping the adjustment of the virtual tool.

This study aims at developing a model $y[t+1] = f(x[t], u[t])$ to predict the next weld pool $y[t+1]$ using the available history of weld pool images $y[t], \dots y[t-p+1]$ and of the input $u[t-1], \dots u[t-q+1]$ denoted as $x[t]$, and the proposed control $u[t]$. Using the predicted $y[t+j]$ and $u[t+j] = u[t]$, the model can be used to predict $y[t+j+1]$ so that the future output can be recursively predicted. By using $u[t+j] = u[t]\ (j \geq 1)$, we are implementing an MPC with the control horizon $N_u = 1$ which is popular in MPC applications. We can predict $y[t+j]$ for $1 \leq j \leq N_y$ corresponding an MPC with the prediction horizon N_y. As $N_u = 1$ ($u[t+j] = u[t]\ (j \geq 1)$), the future output trajectory expects to be smooth. This

allows us to just display $y[t+N_y]$ to the human for the human to see the expected control result and decide if the proposed $u[k]$ can be accepted.

Section 2 and 3 will review related works to analyze the state of the art to identify the challenges and then outline our overall strategy to address the complexity of underlying modeling problem. The experimental system and experimental data will be introduced in Section 4 before Section 5 is devoted to the details of the modeling methodology. The results are presented and discussed in Section 6, before the conclusions are drawn in Section 7.

2. Related Works

HRC welding may provide a solution to enable a quick response to dynamic changes in the GMAW process. Since HRC has already been recognized in micro-manufacturing, which refers to the processes of manufacturing, aligning, connecting, packaging, and testing devices across multiple size scales (typically from 1 micron to 1 centimeter) [9], theoretically, using HRC in welding offers sufficient precision. A fundamental function of HRC is that humans control the machine/robot. Therefore, the robot needs to perceive human instructions and provide feedback to the human operator to assist in making better decisions[4]. An important question is what information the machine should provide to the human, and how this information should be presented to maximize its effectiveness. Typically, in HRC environments, information feedback and AI assistance are presented to humans through various devices and technologies. Since the welding process is complex and dynamic, using high-definition multimedia can better help humans perceive and interpret the environment around the robot for high-precision welding trajectory planning. We believe that using virtual reality technology to present information to humans is the best option. Wang et al.[5] were among the first to use virtual reality technology for coordinated control in human-robot collaborative welding. By using an HDR (high dynamic range) camera in front of the welding torch to construct a digital twin welding system, it allows humans to be immersive as in the welding site.

Conventional HRC welding as in [5] provides human welders with the welding process feedback at the present time, either directly by the weld pool images and or by enhanced weld pool images. The predictions are made in welders mind to decide on the control. The performance of the resultant control largely depends on welder skills which require significant training and many years of practices to gain. Providing process prediction as proposed in this work can reduce the skill levels required. However, the predictive model needed and its complexity are process dependent. In GMAW, the weld pool, as process feedback to humans, is not only influenced by many input parameters, but also by the impact of the welding torch's movement speed on the pool. This is a complex dynamic process, as the fluid dynamics of the pool itself are intricate, and the speed's effect on the pool's morphology exhibits both lag and limitations [10]. Therefore, developing a predictive model for human welders in HCR appears to be challenging and an innovative approach is needed.

The incorporation of machine learning techniques to make welding systems intelligent is a solution that can result in high product quality and cost-effective solutions, which has led to extensive consideration by researchers and industrialists [11]. Yu et al. [12] studied how the deep learning has been a game changer in predicting the weld penetration occurring underneath the workpiece from other measurable phenomena. Mucllari et al. [13] used GAN models to correlate measurable weld pool as represented by weld pool images

to the images on the back-side of the workpiece representing the weld penetration state. It shows that the current weld pool is not, but the history of the weld pool is, sufficient to predict the back-side image. As such, it suggests that the human welders control the weld penetration based on the dynamic evaluation of their observed weld pool. Liu et al. [14] concluded that the empirical skills of the human welders have a crucial role in the production of high quality welding, and proposed an Adaptive Neuro-Fuzzy Inference System (ANFIS) to model the response of the skilled human welders to the molten pool surface. Hu et al. [15] described the behavior and forming appearance of the molten pool on a complex surface by developing a computational fluid dynamics model of the molten pool that considers droplet impact, gravity, arc force, heat and mass transfer. Fewer current studies have considered the idea of dynamic adjustment of the welding process by human welders. Although Mucllari et al. [13] constructed a reverse model to validate the forward process, they focused on the GTAW process as well as the effects of the top image and current parameters. While the GMAW process is more complex compared to the GTAW welding molten pool state, the speed of the torch movement has a greater impact on the welding molten pool state. So it is meaningful to explore the effect of welding speed on the weld pool in the GMAW process, and the HRC system constructed by this model allows the human welder to optimize the speed by adjusting the speed in real time according to the state of the weld pool. However, there is a lack of in-depth exploration of the process to build the model, for example, if the welding speed information is sufficient to be responsible to changes in the weld pool, and how long a time series of welding speed information is needed to account for such changes.

The development of deep learning models based on Convolutional Neural Networks (CNNs) has given robotic welding the opportunity to acquire and process the dynamic behavior of the welding process and construct plausible models based on a simplified computational fluid dynamics process. For the welding process with temporal characteristics, Recurrent Neural Network (RNN) [16] solved the problem that traditional neural networks cannot handle sequential data. In order to overcome the shortcomings of RNN in long sequence processing, Hochreiter and Schmidhuber [17] proposed Long Short-Term Memory Network (LSTM) in 1997. In order to improve the computational efficiency, Cho et al. [18] proposed the gated recurrent unit (GRU) in 2014 that simplifies the structure of LSTM by combining the input and forgetting gates of LSTM into an update gate, while omitting the output gate. This makes GRU more efficient with similar performance to LSTM. Li et al. [7] used an LSTM cell integrated into a U-Net network to obtain complex information contained in the time-series welding process and used it to reconstruct the complex state of the weld pool. Exploring the effect of torch movement speed (welding speed) on the dynamics of the weld pool involves the use of low-dimensional features to predict high-dimensional complex image information. Generative Adversarial Networks (GANs, Generative Adversarial Networks) are widely used in generating high-dimensional images from low-dimensional information. GANs were developed by Ian Goodfellow et al. [19] in 2014, and its basic idea is to gradually improve the ability of the generator through the mutual confrontation of two networks, Generator and Discriminator, so that it can generate high-dimensional data similar to the distribution of real data. The low-dimensional information of the welding torch moving speed in the input is used to obtain the high-dimensional information of the dynamic behavioral image of the welding pool using the GAN network in fact to construct a plausible model by avoiding the influence of complex hydrodynamic calculations and other welding parameters.

Accurately modeling the dynamic system of the weld pool is crucial for optimizing welding processes and improving weld quality. It is necessary to dynamically model the controlled welding process in order to predict the state of the weld pool at the next moment. Common methods for building such models are

based on fundamental theories of thermodynamics, fluid mechanics, and heat transfer, using mathematical equations to describe the temperature field, velocity field, and shape changes of the weld pool. Rosenthal, D [20], proposed a mathematical theory to describe heat distribution during welding, building a dynamic model of the welding process based on experimental data. Wu et al. [21] further digitized the physical processes, using numerical simulation to model fluid flow and heat transfer within the weld pool. They developed a transient numerical model to study the dynamic behavior of full-penetration GTAW. Building on previous efforts, for the issue of controlling the depth of the weld pool in the GMAW process, a dynamic model of the weld pool was established, and an MPC (Model Predictive Control) strategy was used to predict and control the pool depth in real time. This brought the accuracy and stability of dynamic weld pool state control to a new level [6]. MPC can handle the complex nonlinearities and system uncertainties in the GMAW process. It has been applied to control the backside weld bead width in pulsed GMAW, optimizing heat input through feedback from electrical characteristic signals to achieve the desired weld geometry [22]. Additionally, MPC has been used to regulate droplet detachment frequency and arc length, allowing consistent weld quality across different operating conditions [23]. Further, a robust MPC algorithm has been developed to control a process where the model parameters are uncertain with the uncertainty being quantified using intervals [24] and been used to control the complex welding processes [25-27] including GMAW [26] and its novel modification double-electrode GMAW [27]. However, in all previous MPC-based welding process controls, the welding process has been represented by carefully chosen states and outputs. It has not been applied to an "original" welding process where the "original" states and outputs are high-dimensional—specifically, approximately infinite-dimensional raw phenomena like welding pool images that have not been reduced to low-dimensional variables.

3. Overall Modeling Strategy

Most deep learning-based weld penetration monitoring studies use observable complex phenomena Ξ (high dimensional data) as input to predict low dimensional labels. The high-dimensional data contains both correlated and uncorrelated features that may be extracted to match with the labels. Validation data may also contain the same irrelevant features, and using the validation results to measure model quality is a compromise since there is no other better measure. However, in generative modeling, low-dimensional data is used to predict complex high-dimensional data. The fundamental role of simple inputs in determining complex outputs can be better secured if modeling is successful [2]. In the images we have captured experimentally, we can clearly see the changes in the size and shape of the resulting molten pool with the change in the moving speed of the torch, as shown in Fig. 1. We can preliminarily conclude that there is a certain relationship between the moving speed of the torch and the size of the molten pool formed. Therefore, we can establish a preliminary hypothesis that the low-dimensional data of the change in speed will have a correlation with the high-dimensional data of the image of the change in the size of the weld pool. Our initial aim is to validate this model in some way.

On this base, we propose conducting research through generative modeling. We denote the weld pool as represented by its image as $y[t]$, and welding speed as $u[t]$. We intend to establish a GAN [16] model to generate/predict the next image $y[t+1]$ from currently available information $x[t]$ as a function of $u[t]$. The image $y[t+1]$ displayed to the human operator will continuously update in real-time as long as the human adjustment continues. The $u[t]$ will be sent to the robot for execution once the adjustment is below

the pre-set threshold. Since the original GAN model lacks direct control over specific attributes of the generated samples during the generation process, we employ a Conditional GAN[28].

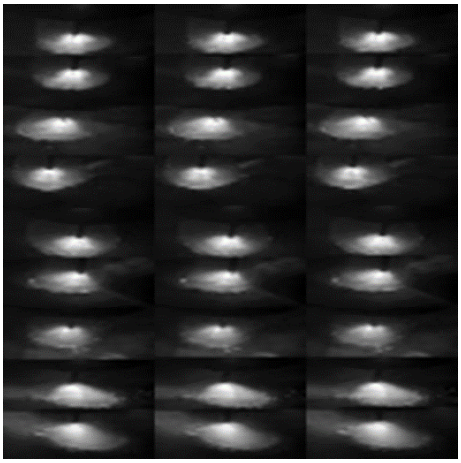

Figure 1 Weld pool dynamics

When designing a generator for Conditional GAN (CGAN) [28], the first thing that should be considered is whether the depth and breadth of the generating structure can generate a clear enough image of the weld pool. If a too deep network is chosen directly it may lead to overfitting or unstable training. We propose to use a symmetric Encoder-Decoder structure and test it with different depths to generate $y[t]$ from itself. Using the same image as the input and output assures the raw input information is 100 percent sufficient so that the acceptability of the proposed structure will only depend on if the Encoder can adequately extract the image features and the Decoder can generate the same image from the features. Because this also allows the high dimension information (image) directly compared, the generatability of the proposed structure can be accurately assessed. Further, as the features extraction from the image and the same image is generated from the same features are inverse mappings, we propose to use the same depth for the Encoder and Decoder so that we can gradually increase the depth to find the optimal depth. This makes the process/effort to determine the optimal structure/depth more manageable. In case there is a difference in the optimal depths for them, as long as it is not too significant, the resultant depth is still optimal in balancing the sufficiency with overfitting/training instability (caused by excessive depth).

We have two types of conditions, image and signal (welding speed). As we have decided an optimal structure to introduce the image condition, we will use it and consider a separate way to introduce the signal (welding speed) condition and then concatenate them together. This results in a change in the Decoder (input) and theoretically may call for an increase in its depth. However, in this first study of this kind, we will still use the same depth with an eye open for the possibility to slightly increase it. Further, the two conditions both dynamically contribute the underlying welding process. We propose to apply a RNN (Recurrent Neural Network) to each of them to form the two separate information/features and finally concatenate them together as the input condition of the Decoder. The final model structure is illustrated in Fig. 2 and details will be discussed in Section 5.

4. Experiments

GMAW process [8] can be simply explained by Figure 3. The heat for melting the metal is generated by the arc between the consumable electrode and the metal. The electrode is a wire that is continuously fed/melt through the arc. During welding, an inert gas (such as helium and argon) or its mixture with of other gases

is supplied to the welding area to shield and protect the arc, the weld pool, the electrode, and the base material nearby from interacting with the atmosphere. The weld pool is formed by the heat from the arc generated between the continuously fed filler metal (electrode) and the workpiece. The arc also melts the electrode, forming droplets at the electrode tip. These droplets detach from the electrode tip and are transferred to the workpiece through the arc. Under the heat from the arc and droplets as well as the periodic impact from the droplets, the weld pool gradually forms. The formation of droplets, their detachment and transfer, and the resultant dynamic changes in the weld pool are controlled in an extremely way by the fluid flow, forces, heat within the droplets and the weld pool, and the arc (thus heat and forces imposed on the weld pool) that fluctuates in association with the periodic formation, growth, detachment, and flight of the droplets. In particular, the arc length, thus the distribution of the arc (heat and force distributions), is subject to a periodic change. The welding current (thus the arc pressure which is proportional to the square of the welding current) is also continuously adjusted by the welding system to balance the melting and feeding of the wire, thus to maintain the arc stability [29]. The state of the weld pool is thus controlled affected by multiple parameters and reflect the state of the welding process and the quality of the resultant welds.

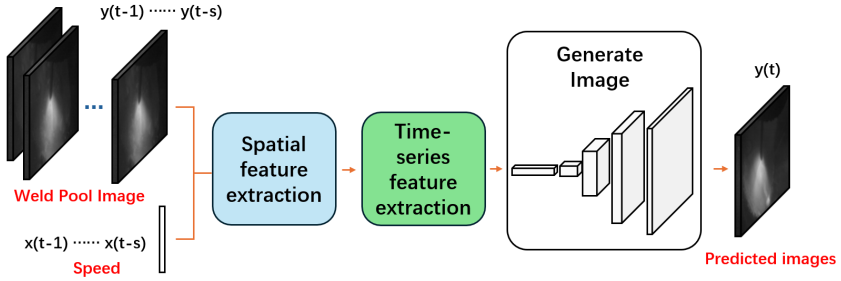

Figure 2 Modeling methodology

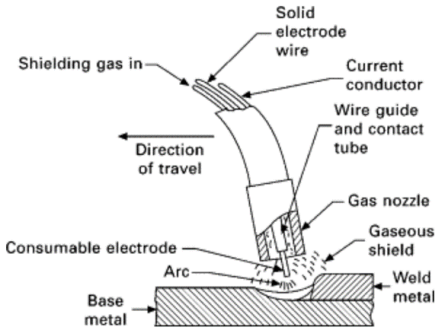

Figure 3 Gas Metal Arc Welding (GMAW) [8]

The question arises why a mapping like $y[t] = f(y[t-1], \dots y[t-p], u[t-1], \dots u[t-q])$ or its derivation $y[t+1] = f(x[t], u[t])$ may model such a complex process as aforementioned. We note that the wire feed speed, which is fixed, dictates the welding current needed for the welding machine to maintain the balance between the feeding and melting of the wire and the current controls the arc force and the electromagnetic force to detect the droplet. The current and arc length (thus the arc voltage) determined by the phase of the droplet in its forming, growing and detaching period decide the power of the arc. The arc power and welding speed decide the heat input into the weld pool. As such, the wire feed speed and welding speed are the source control variables to determine the fluctuating periodical dynamic process of the weld pool. The weld pool reflects the results of the welding process as well as the influence from other uncontrolled variables and the sate of the welding process in the periodical fluctuations. When the wire feed speed is fixed, the source control variable is the welding speed. In the meantime, the dynamically changing welding

process state as reflected in the weld pool image history captures the phase in the periodical change. The proposed model inputs are thus expected to be sufficient to predict the next weld pool (image) and the success in obtaining the mapping will depend on the modeling approach for which we have proposed using CGAN.

We used two 6-degree-of-freedom (DOF) robots, the UR10e and UR16e, as a welding robot and a monitoring robot, respectively (Fig. 4). The Xiris HDR weld camera XVC-1100, a high dynamic range (HDR) camera for monitoring of open-arc processes, was mounted on the monitoring robot to capture images of the arc and the weld pool. A bandpass filter centered at 650 nm was mounted on the lens to filter out strong arc light. We used a PC to control the speed and direction of the welding robot and to monitor the robot's movement according to the control instructions.

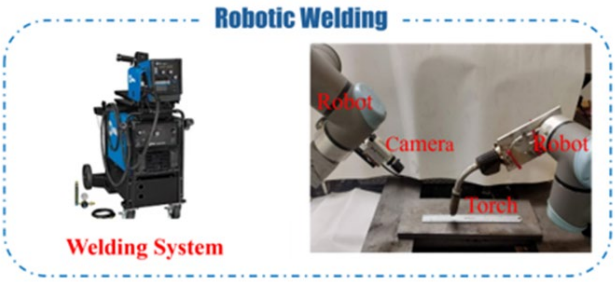

Figure 4 Robotic Welding System

We set the welding power source to operate at a high voltage of 40.8V in constant voltage mode to ensure minimal spatter and good bead appearance in the spray transfer mode. Under these conditions, the welding circuit can be simply regarded as an inductive load, with current undergoing periodic high-frequency variations. The fluctuations in current parameters have negligible effects during this process. The welding parameters were controlled by a computer using a PCI-6229 National Instruments data acquisition card running in C++. We conducted a total of 19 experiments, capturing 30 images per second, resulting in a usable image dataset of 19,184 images after preliminary processing.

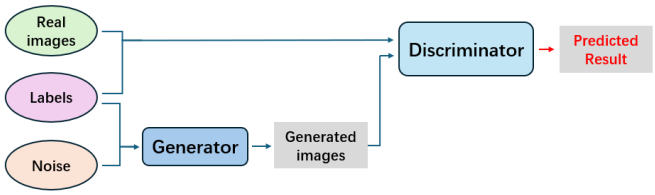

Figure 5 Conditional GAN structure[28]

5. Modeling Methodology

5.1 Conditional GAN

In 2014, Mirza et al. [28] proposed the Conditional GAN (CGAN), as illustrated in Figure 5. The core idea of CGAN is to introduce additional conditional information into the original GAN architecture, allowing the generator and discriminator to consider conditional variables simultaneously during training. CGAN extends the input space of both the generator and discriminator by incorporating a conditional variable c. In the generator, the input consists of noise z combined with the conditional variable c, enabling the generation

of samples $G(z|c)$ that are related to the condition c. The conditional variable c can take various forms, such as class labels, text descriptions, or image attributes. In the discriminator, the task is not only to judge the authenticity of the input sample but also to predict according to the corresponding conditional variable. The objective function of CGAN has two components: first, the ability to distinguish between real and fake samples, and second, the accurate prediction of the conditional variable c. Therefore, the loss function consists of two parts: the generator loss L_G, which encourages the discriminator D to classify the generated sample $G(z|c)$ with the corresponding condition c as "fake" maximizing $log(1 - D(G(z|c), c))$; and the discriminator loss L_G , which encourages D to correctly differentiate between the real sample (X, c) and the generated sample $(G(z|c), c)$, minimizing $E_{(X,c)\sim P(X,c)}[log\,(D(X, c))] + E_{z\sim P(z)}[log\,(1 - D(G(z\,|\,c), c))]$. In this work, we will implement the Wasserstein GAN and use the Wassetstein loss [39]

$$\mathcal{L}_D = E_{x\sim p_{data}(x)}[D(x)] - E_{z\sim p_z}[D(G(z))] \quad (4)$$

5.2 Encoder-Decoder

We use an Encoder-Decoder structure to extract information and then to generate weld pool image. The Encoder-Decoder structure was initially proposed by Ilya Sutskever et al. [30] in 2014 for handling sequence-to-sequence (Seq2Seq) tasks, and it has demonstrated strong expressiveness when adapted to image data. Due to the complexity of weld pool images, traditional Multi-Layer Perceptrons (MLP) [31] struggle to generate usable images. An MLP is a type of feedforward neural network typically consisting of an input layer, multiple hidden layers, and an output layer. Each neuron in a layer is fully connected to the neurons in the previous layer, and the network performs nonlinear transformations using activation functions such as ReLU or Sigmoid. The design of MLPs is relatively simple, and they are generally used for tasks like classification and regression, where both the input and output have fixed dimensions.

However, the Encoder-Decoder structure is more suited for handling complex data like images. It generally consists of two main components: the encoder and the decoder. The encoder's task is to compress the input data into a low-dimensional latent space representation, typically achieved using Convolutional Neural Network (CNN) (Fig.6). The decoder is then responsible for reconstructing this latent representation into the output data (Fig. 6), such as an image or a label map.

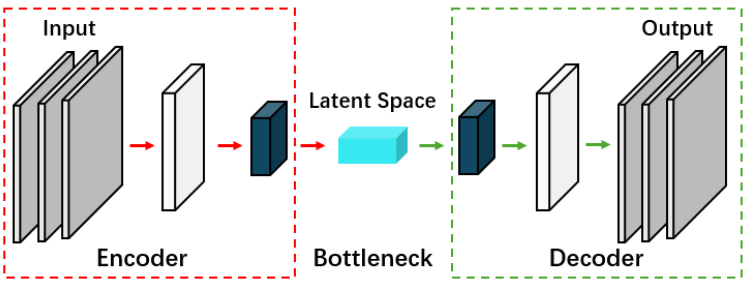

Figure 6 Encoder-Decoder structure[30]

The Encoder-Decoder structure is viable for reconstructing weld pool images. Silva et al. [32] studied the role of the Encoder-Decoder design in machine learning (ML) from an information-theoretic perspective. They validated the usefulness of predictive structures in ML by examining information sufficiency and mutual information loss. In our study, we use 64x64x1 weld pool images as input and propose

constructing Encoder-Decoder models with varying depths to determine the necessary depth for reconstructing usable weld images. We configure each convolutional/deconvolutional layer with a kernel size of 4x4, a stride of 2x2, and padding of 1. We test Encoder-Decoder models with depths of 4, 5, 6, 9, 11, and 13 layers. Since the goal is to assess the model's ability to reconstruct image information, we input a single weld pool image and output a reconstructed image to evaluate the model's effectiveness. The CGAN used in this study (Figure 7) employs an Encoder-Decoder structure for the generator, while the discriminator uses an Encoder structure. The parameters for the GAN are set as shown in Table 1.

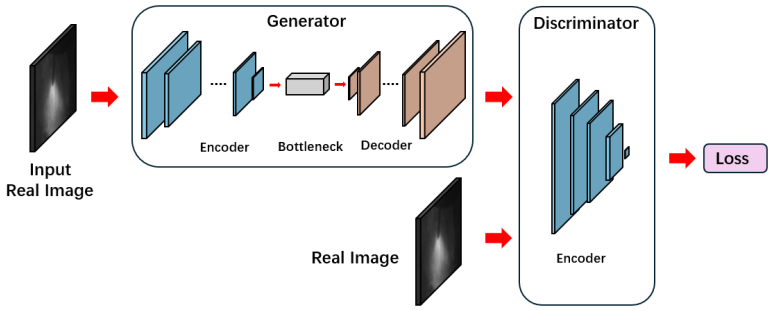

Figure 7 Validating GAN with Encoder-Decoder Layers

Table 1. GAN parameters.

Parameters	Values
Z-dim	100
Epoch	300
Learning rate	9e-5
Loss function	Wasserstein
Batch size	64

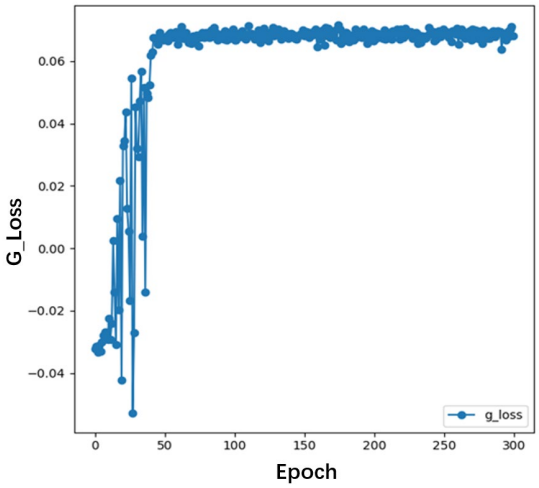

Figure 8 Generator Loss

We set the learning rate appropriately and chose a sufficient number of training epochs, as it is clear from the training loss of the most structurally complex Encoder-Decoder model with 13 layers of depth that the generator loss has stabilized at around 0.06 in about 50 Epochs (Fig.8).

Only use the generator loss to judge the adequacy of the information in the generated images is not a comprehensive measure of the quality of the generated images. Zhao et al [33] pointed out that the

generator loss is not sufficient to assess the quality of the generated images because the generator loss usually focuses on pixel-level differences or the discriminative ability of the model and does not adequately reflect the perceptual quality of the images, especially in preserving the image structure and details. Meanwhile, the most widely used full-reference quality metric, mean square error (MSE), is easy to compute and optimize but does not match well with the perceived visual quality of the human visual system [34]. So, we add the parameter Structural Similarity Index Metric (SSIM), which was proposed by Wang et al [35] in 2004 as a metric to measure the similarity between two images. It evaluates the quality of an image by comparing its brightness, contrast and structural information and is an important tool in image quality assessment. The value of SSIM ranges from 0 to 1, where 1 means that the two images are exactly the same and 0 means that the two images are completely different. Unlike the traditional MSE, SSIM is more in line with the human visual system's perception of image quality. The SSIM is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (1)$$

where, μ_x, μ_y, σ_x, σ_y denote respectively the mean and standard deviation of the images $x := I_t$ and $y := \hat{I}_t$, $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ with K_1, $K_2 \ll 1$ and L = pixel max value. In order to avoid instability when ${\mu_x}^2 + {\mu_y}^2$ and ${\sigma_x}^2 + {\sigma_y}^2$ are very close to zero, we employ C_1 and C_2, which are constants, to prevent division by a small number. SSIM was first introduced in [35] to measure the difference between two images by considering the luminance, contrast and the structure. DSSIM (Difference Structural Similarity Index Measure) is a variant or complement of the SSIM, which is used to measure the difference between two images, defined as:

$$DSSIM(x,y) = \frac{1 - SSIM(x,y)}{2} \quad (2)$$

Also, the loss function we designed to train the generator is the following weighted loss:

$$\mathcal{L}_G = 0.8 * Wass_G + 0.2 * DSSIM \quad (3)$$

$$Wass_G = E_{X\sim Pg}[D(x)] - E_{X\sim Pr}[D(x)] \quad (4)$$

The number of layers of Encoder and Decoder under the same structure is the same. After training the models with different structures, we get the final SSIM values as shown in Table 2.

Table 2. SSIM values for different layer structures

Layer	SSIM Value
4	0.8702
5	0.9023
6	0.9056
9	0.9144
11	0.9146
13	0.9206

From the SSIM values, with the increase in the number of layers the image information is more adequately restored, which is reasonable since the depth and complexity of the network is increasing. It also proves that the generation of images of weld pool requires a deep network. However, the SSIM values for models with more than 6 layers do not change significantly while humans can observe the increase in the details when the generated images approach the real (Fig. 9). So, we need to refer to human perception vision as the final measure, in particular for the intended application in HRC systems.

From Fig. 9, it can be observed that the images generated in the 4-layer, 5-layer, and 6-layer models are very similar and fuzzy, and lack detailed information, indicating that the depth of the network is not sufficient to accurately extract and accurately restore the image information. After exceeding 9 layers, the image clarity increases to a usable state, indicating that the depth of the network at this point is sufficient to extract and restore the information of the original image. Based on the experimental results, we conclude that an Encoder-Decoder model with a depth of at least 9 layers is required to generate the weld pool image. It is also observed that there is little difference in the results at 9 layers compared to 11 layers and 13 layers, but in practice, training with 11 and 13 layers requires exponentially more resources, especially a large amount of GPU memory. Choosing the appropriate depth of the network can reduce the consumption of computational resources while guaranteeing the performance of the model [36], and the Encoder-Decoder model with a 9-layer structure is thus chosen for generating weld pool images.

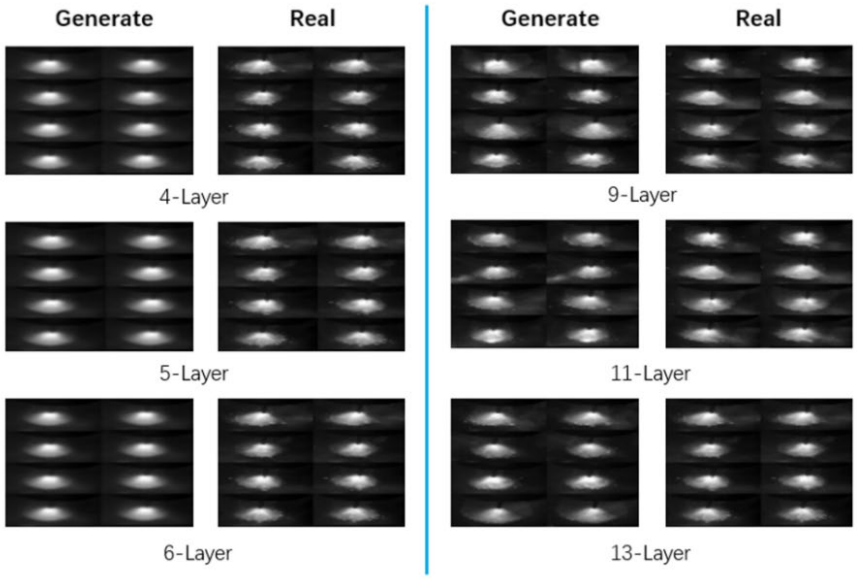

Figure 9 Generate results for different layers

5.3 GRU-GAN Model

5.3.1 GRU Model

We will use a GRU to process the time series of the extracted image features (hidden state). GRU (Figure 10) is a simplified version of LSTM [37], proposed by Cho et al. [18] in 2014. Compared to LSTM, GRU merges the input gate and forget gate into a single update gate and uses a reset gate to control the influence of the previous state on the current state. The update gate determines the extent to which the hidden state at the current time step inherits information from the previous hidden state. The reset gate controls the influence of the previous hidden state on the current candidate hidden state. The candidate hidden state is a candidate value calculated based on the current input and the reset hidden state, and the updated hidden state combines the candidate hidden state with the hidden state of the previous time step to some extent. There are many noteworthy GRU variants with excellent performance, such as GORU [38], etc.

5.3.2 GRU-GAN Generator

We have determined to use a 9-layer Encoder-Decoder structure and thus design the Encoder shown in Fig.10 to process image inputs, first resizing each input to 64x64x1, followed by 9 layers of Conv2D, where all convolutional layers use a kernel size of 4, a stride of 2, and a padding size of 1. Batch normalization is applied after the 3rd and 6th layers. Finally, all $y[t-i]'s$ are concatenated together.

The speed data is processed using the same method as the CNN 1D approach employed by Mucllari et al. [2]. This input consists of $u[t-j]'s$, which are passed through two 1D convolutional layers (CNN 1D). Both layers use a kernel size of 3, with the number of channels set to 3 and 6, respectively. The output of these layers is then merged with the output of the GRU and reshaped into a 512x1x1 format. This output is then used as the input to the Decoder. The Decoder model consists of 9 layers of 2D transposed convolution, with each layer using a kernel size of 4, a stride of 2, and a padding size of 1. Batch normalization is applied after the third and sixth layers. Finally, the output is resized to 64x64x1, which is the predicted weld pool image $y[t]$ (Fig.11).

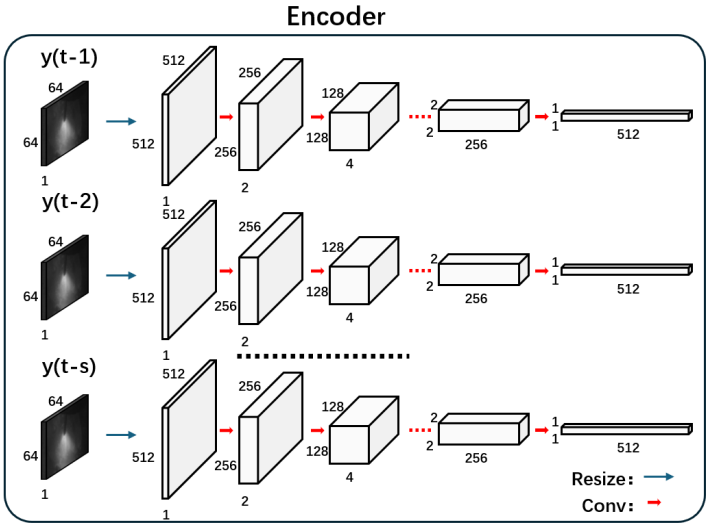

Figure 10 GRU-GAN Generator Encoder

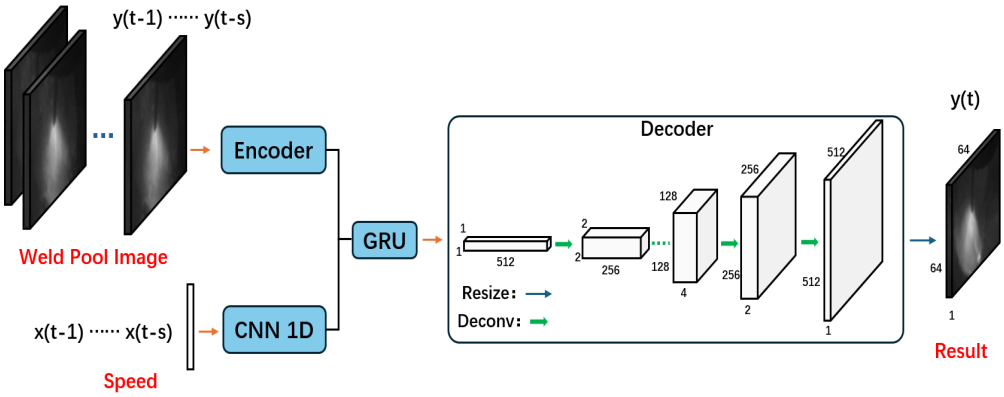

Figure 11 GRU-GAN Generator Structure

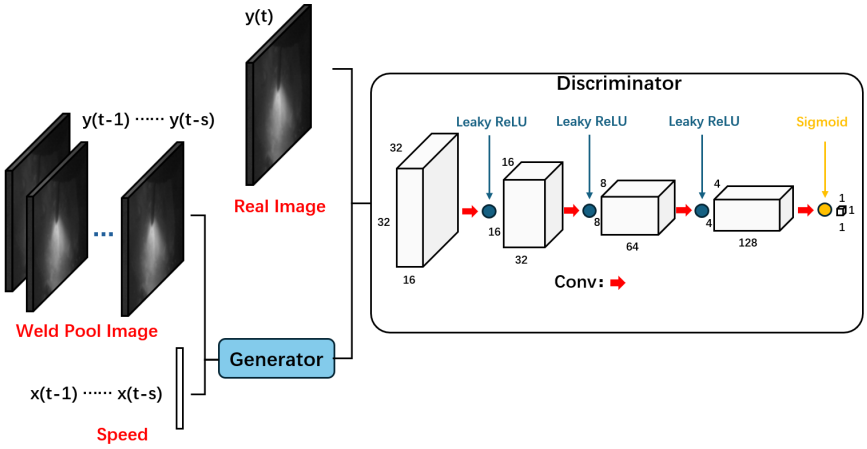

Figure 12 GRU-GAN Discriminator Structure

The loss function for training the GRU model is based on the "Wasserstein" loss [39]: see (??).
In our training, an additional setting was set to apply exponential learning rate decay every 10 epochs with a decay coefficient of 0.9.

5.3.4 GRU-GAN Discriminator

The discriminator uses a 5-layer Encoder structure, where all convolutional layers have a kernel size of 4 and a stride of 2. The padding size is set to 0 for the final layer and 1 for the other layers. Batch normalization [40] is applied between layers, and a Leaky ReLU with a coefficient of 0.2 is used. The last layer of the discriminator employs a sigmoid activation function. This architecture is illustrated in Figure 12. The inputs to the discriminator are the real image at time $y[t]$ and the generated image created by the generator using the previous images and speed data.

6. Results and Discussion

6.1 Dataset

The training was conducted using two datasets: a large dataset with 19,184 images and a small dataset with 1,000 images. To demonstrate that the impact of speed and temporal information on the weld pool is significant and easily identifiable, it is equivalent to proving that our model performs well even when trained on the small dataset. We divided the datasets into training and testing subsets, with 80% used for training and the remaining 20% for testing. We conducted comparative experiments on these two datasets, using generator loss ($\mathcal{L}_G$) and SSIM [35] as evaluation metrics.

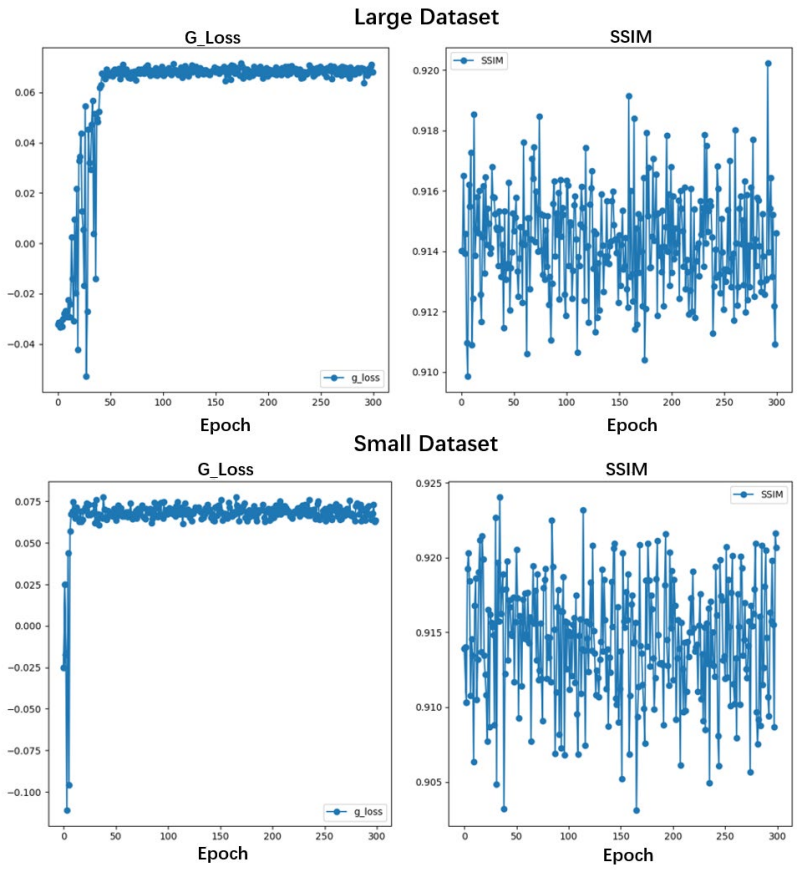

Figure 13 Generator loss and SSIM values

Figure 13 shows the changes in $\mathcal{L}_G$ and SSIM for both the Large Dataset and Small Dataset across epochs. It is clear that the $\mathcal{L}_G$ for the small dataset is only slightly larger than that for the large dataset, by less than 0.01, which can be considered negligible. The SSIM values for both datasets fluctuate around 0.915, with the small dataset generally exhibiting slightly higher SSIM values. The most notable difference between the two datasets is that the $\mathcal{L}_G$ for the small dataset stabilizes more quickly. Therefore, using the small dataset for this model over 300 epochs is sufficient. (Fig.14)

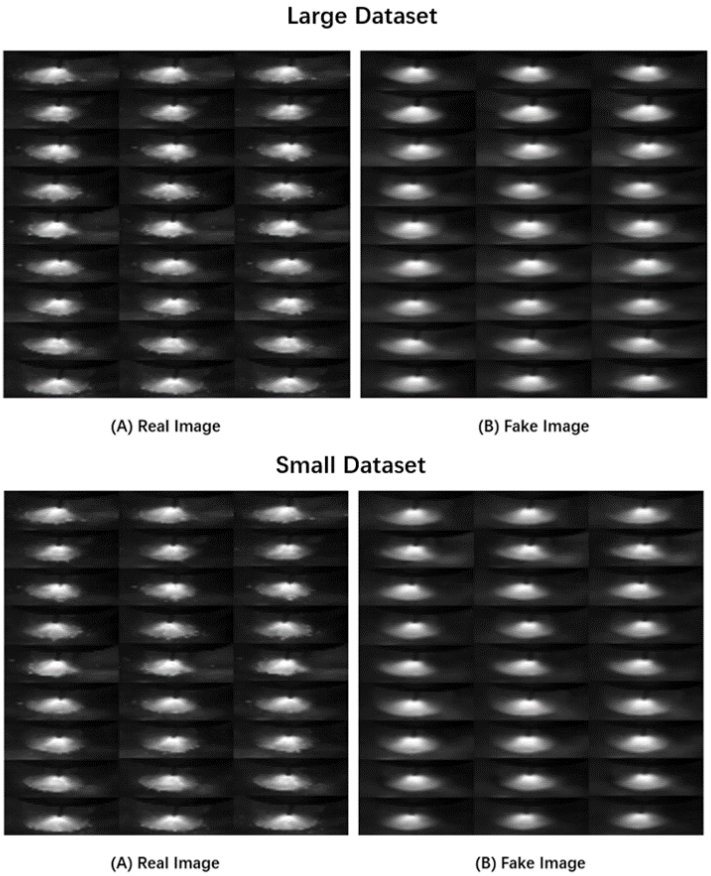

Figure 14 Generate results with different dataset sizes

6.2 Effects of Time-series Images

To study the effect from previous images, we need first determining the appropriate length p which has been introduced in the Introduction. Clearly, calculating this theoretically is very difficult because too many factors influence the dynamic process. We observed that, whenever there was a significant change in speed, the weld pool would exhibit the expected changes within 1 to 2 seconds. Our camera captures 30 images per second, so we designed a comparative experiment with p being 1, 10, 30, and 60. We compared the MSE and SSIM metrics for each p (Table 3).

Table 3. Effects of time-series images

Sequence length p	MSE	SSIM
1	0.068	0.9144
10	0.0021	0.9537
30	0.0032	0.9244
60	0.0028	0.9376

It can be observed that when p is 10, 30, and 60, the MSE values are all lower than when p=1, while the SSIM values are all higher. This validates the hypothesis that the image information from previous time sequences contains hidden information about the current image. At the same time, it can be observed that the model's performance does not increase with longer time sequences. Although the MSE value for the sequence length of 60 is lower than that for 30, its SSIM value is nearly 0.01 lower than that of 30. This

difference is even more pronounced when comparing the sequences of 10 and 30. The sequence length p of 10 not only has a lower MSE value than the sequence length of 30 but also lower than that of 60. Additionally, the SSIM value reaches 0.9537, the highest among the four sequences.

From the generated images (Fake Images) in Figure 15, it can be observed that although the differences in SSIM and MSE values between the sequence lengths are small, the image generated with a sequence length of 10 contains more detailed information about the weld pool. This indicates that a sequence length of 10 provides the best predictive performance. This phenomenon might suggest that there is a limit to the significant impact of time sequences on the weld pool. It is also possible that using longer sequences causes the GRU to lose important hidden information relevant to that specific moment. However, since this study focuses on the impact of speed rather than on the weld pool, we will introduce the speed parameter to observe changes in the model's performance.

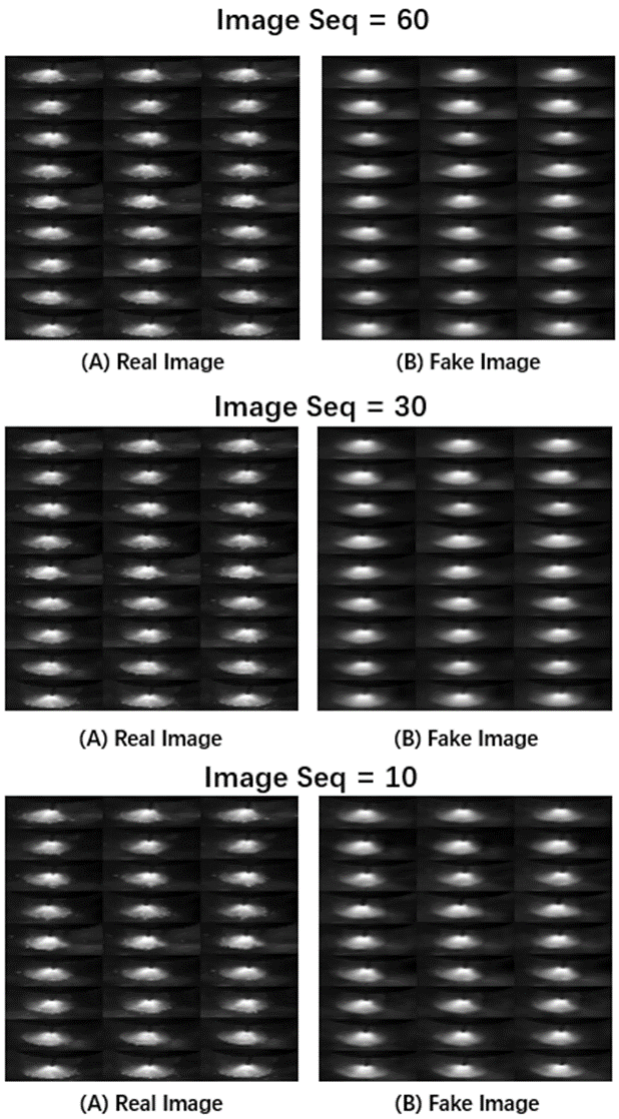

Figure 15 Results of time series images as input

6.3 Effects of Time-series Speed

Using the same method to incorporate speed, we concatenated the previous time sequence image information with the speed sequence information and then input this combined data into the GRU. The output of the GRU was used as the input for the Decoder to generate realistic weld pool images. In the comparative experiments, the selected image sequences were p =1, 10, 30, and 60, corresponding to speed sequences of q =1, 10, 30, and 60 (Table 4).

Table 4. Effects of time-series Speed

Speed Seq length q	Image Seq p	MSE	SSIM
1	1	0.051	0.9132
10	10	0.0021	0.9549
30	30	0.0021	0.9549
60	60	0.0019	0.9534

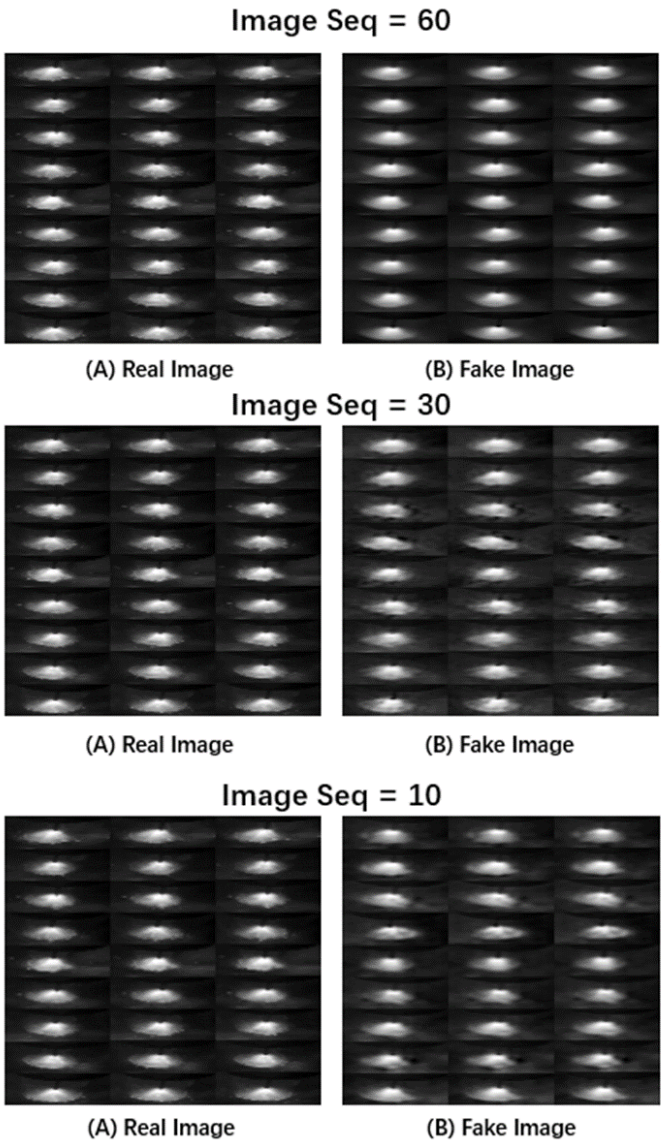

Figure 16 Results of adding time series speed data as input

It can be observed that after introducing the speed sequence, the model's two evaluation metrics, MSE and SSIM, showed significant improvement. Additionally, when the image sequence length was 10, the results were the same as those with an image sequence length of 30. However, while the MSE value was lower with an image sequence length of 60, the SSIM value did not perform as well, possibly due to model overfitting [41].

From the generated images (Fake Image) in Figure 16, it can be observed that the weld pool images generated with an image sequence length of 30 contain more detailed information than those generated with a sequence length of 60, which aligns with the SSIM value trends. The difference between the images generated with sequence lengths of 30 and 10 is not significant. This may be because the introduction of two time sequence datasets enhanced the model's robustness, allowing it to focus more effectively on important latent information. This demonstrates that incorporating the speed sequence is effective for generating weld pool images with the model.

We believe this result also indirectly confirms that experienced welders can adjust the speed of the torch based on real-time feedback from the weld pool to achieve high-quality welds. In the dynamic welding process, the information from speed changes is sufficient to generate the corresponding changes in the weld pool under the given conditions (wire feed speed and arc voltages setting etc. in GMAW).

7. Conclusions

We argue that HRC (human-robot collaboration) can ease the robotization of complex welding processes where both human adaptation and physical strength are required, and propose to reduce the reliance on human skills through a human-based model predictive control (MPC). The basic question is whether a complex welding process can be effectively modeled using a generalizable approach so that the future output can be predicted in terms of human perception. This study predicts the weld pool image as the future output of human perception under the control of welding speed in the complex GMAW process. We tested the feasibility using a conditional GAN with a GRU structure, allowing us to account for the history of the output (weld pool image) and control (welding speed) without prior knowledge of the dynamics of the welding process. To this end, we first explored the depth needed to verify the Encoder-Decoder structure of the generator in the GAN to generate an image containing sufficient information about the weld pool. According to our experimental results, it is necessary to have more than nine layers in the Encoder-Decoder structure, and considering efficiency, we believe that a nine-layer structure is optimal. We then constructed a conditional GAN with a GRU structure to generate an image of the weld pool using its previous time-series information, demonstrating that this time-series information is useful for predicting the dynamic welding process. Furthermore, we showed that adding control data (the time series of welding speed) significantly improved our predictions. The dynamic time series of the output (weld pool images) and control (welding speed) together effectively predict the future output (weld pool image).

Acknowledgement

This work is partially funded by the University of Kentucky Institute for Sustainable Manufacturing and Department of Mathematics and the National Science Foundation under grants IIS-2327113 and CMMI-2024614. The assistance from Edison Mucllari is greatly appreciated.

References:

[1] Force Design. "Robotic Welding System Design: A Need-to-Know Guide for Automated Welding." Force Design, 16 Apr. 2020, https://forcedesign.biz/blog/robotic-welding-system-design-a-need-to-know-guide-for-automated-welding/.

[2] Mucllari, E., Cao, Y., Ye, Q., & Zhang, Y. (2024). Modeling imaged welding process dynamic behaviors using Generative Adversarial Network (GAN) for a new foundation to monitor weld penetration using deep learning. Journal of Manufacturing Processes, 124, 187-195.

[3] Lei, T., Rong, Y., Wang, H., Huang, Y., & Li, M. (2020). A review of vision-aided robotic welding. Computers in Industry, 123, 103326Lei, T., Rong, Y., Wang, H., Huang, Y., & Li, M. (2020). A review of vision-aided robotic welding. Computers in Industry, 123, 103326.

[4] Ajoudani, A., Zanchettin, A.M., Ivaldi, S. et al. Progress and prospects of the human–robot collaboration. Auton Robot 42, 957–975 (2018). https://doi.org/10.1007/s10514-017-9677-2.

[5] Wang, Q., Jiao, W., Yu, R., Johnson, M.T. and Zhang, Y., 2019. Modeling of human welders' operations in virtual reality human–robot interaction. IEEE Robotics and automation letters, 4(3), pp.2958-2964.

[6] Y. K. Liu and Y. M. Zhang, "Model-Based Predictive Control of Weld Penetration in Gas Tungsten Arc Welding," in IEEE Transactions on Control Systems Technology, vol. 22, no. 3, pp. 955-966, May 2014, doi: 10.1109/TCST.2013.2266662.

[7] Li, T., Cao, Y., & Zhang, Y. (2024). Analysis of weld pool region constituents in GMAW for dynamic reconstruction through characteristic enhancement and LSTM U-Net networks. Journal of Manufacturing Processes, 127, 573-588.

[8] Mallick, P. K. (2010). Joining for lightweight vehicles. In P. K. Mallick (Ed.), *Materials, Design and Manufacturing for Lightweight Vehicles* (pp. 275-308). Woodhead Publishing.

[9] D. Wei et al., "Automated Fabrication of Tactile Sensors Using a Custom Additive Manufacturing Platform," 2022 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS), Toronto, ON, Canada, 2022, pp. 1-6, doi: 10.1109/MARSS55884.2022.9870485.

[10] Cho, WI., Na, SJ. Impact of driving forces on molten pool in gas metal arc welding. Weld World 65, 1735–1747 (2021). https://doi.org/10.1007/s40194-021-01138-8

[11] Kumar, S., Gaur, V. & Wu, C. Machine learning for intelligent welding and manufacturing systems: research progress and perspective review. Int J Adv Manuf Technol 123, 3737–3765 (2022). https://doi.org/10.1007/s00170-022-10403-z

[12] Yu, R., et al., 2023. 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, pp.15-46.

[13] E. Mucllari, R. Yu, Y. Cao, Q. Ye and Y. Zhang, "Do We Need a New Foundation to Use Deep Learning to Monitor Weld Penetration?," in IEEE Robotics and Automation Letters, vol. 8, no. 6, pp. 3669-3676, June 2023.

[14] Liu, Y. K. (2014). Skilled Human Welder Intelligence Modeling and Control: Part 1 Modeling. Welding Journal., 93(2), 46S-52S.

[15] Hu, Z., Hua, L., Qin, X., Ni, M., Ji, F., & Wu, M. (2021). Molten pool behaviors and forming appearance of robotic GMAW on complex surface with various welding positions. Journal of Manufacturing Processes, 64, 1359-1376.

[16] McCulloch, W. S., & Pitts, W. (1943). "A logical calculus of the ideas immanent in nervous activity." Bulletin of Mathematical Biophysics, 5(4), 115-133.

[17] Hochreiter, S., & Schmidhuber, J. (1997). "Long short-term memory." Neural Computation, 9(8), 1735-1780.

[18] Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Conference on Empirical Methods in Natural Language Processing (EMNLP 2014).

[19] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). "Generative adversarial nets." Advances in Neural Information Processing Systems, 27.

[20] Rosenthal, D. (1941). Mathematical theory of heat distribution during welding and cutting. Welding Journal, 20(5), 220s-234s.

[21] Wu, C. S., & Zhu, X. K. (2004). Numerical simulation of three-dimensional transient heat transfer and fluid flow in GTA welding. International Journal of Heat and Mass Transfer, 47(17-18), 3745-3758.

[22] Tukahirwa, G., & Wandera, C. (2023). Influence of Process Parameters in Gas-Metal Arc Welding (GMAW) of Carbon Steels. IntechOpen. doi: 10.5772/intechopen.1002730.

[23] Ye, Y., Wang, Q., Chen, C. et al. Multi-objective Optimization of Short-Circuit Gas Metal Arc Welding for Overhang Structures by Grey Relational Analysis Integrated Taguchi Method. J. of Materi Eng and Perform (2024). https://doi.org/10.1007/s11665-024-10052-1.

[24] Y. M. Zhang and R. Kovacevic, “Robust control of interval plants: A time domain method,” IEE P-CONTR THEOR AP 144 (4): 347-353 JUL 1997.

[25] W. Lu, Y. M. Zhang, and W.-Y. Lin, “Nonlinear interval model control of quasi-keyhole arc welding process,” Automatica, 40(5): 805-813, 2004.

[26] Y. M. Zhang, Liguo E, and B. L. Walcott. "Robust control of pulsed gas metal arc welding," Journal of Dynamic Systems, Measurement, and Control -Transactions of the ASME 124(2): 281-289, 2002.

[27] Kehai Li, YuMing Zhang, “Interval model control of consumable double-electrode gas metal arc welding process,” IEEE Transactions on Automation Science and Engineering, 7(4): 826-839, 2010.

[28] Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[29] Hu, J., & Tsai, H. L. (2007). Heat and mass transfer in gas metal arc welding. Part I: The arc. *International Journal of Heat and Mass Transfer, 50*(5–6), 833-846.

[30] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

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

[32] Silva, J. F., Faraggi, V., Ramirez, C., Egana, A., & Pavez, E. (2024). Understanding Encoder-Decoder Structures in Machine Learning Using Information Measures. arXiv preprint arXiv:2405.20452.

[33] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss Functions for Image Restoration With Neural Networks," in *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47-57, March 2017, doi: 10.1109/TCI.2016.2644865.

[34] "Whats wrong with mean-squared error" in Digital Images and Human Vision, MA, Cambridge:MIT Press, pp. 207-220, 1993.

[35] Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004, doi: 10.1109/TIP.2003.819861.

[36] Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295-2329.

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

[38] Mucllari, E., Zadorozhnyy, V., Pospisil, C., Nguyen, D. and Ye, Q., 2022. Orthogonal Gated Recurrent Unit with Neumann-Cayley Transformation. arXiv preprint arXiv:2208.06496.

[39] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein Generative Adversarial Networks. *International Conference on Machine Learning*.

[40] Ioffe, S. and Szegedy, C., 2015, June. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR.

[41] Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, *4*(1), 1-58.