Automated defect classification of Aluminium 5083 TIG welding using HDR camera and neural networks

Bacioiu, Daniel; Melton, Geoff; Papaelias, Mayorkinos; Shaw, Rob

DOI:
10.1016/j.jmapro.2019.07.020

Automated defect classification of Aluminium 5083 TIG welding using HDR camera and neural networks

Daniel Bacioiu[a,b], Geoff Melton[b], Mayorkinos Papaelias[a], Rob Shaw[b]

[a] *Department of Metallurgy and Materials, University of Birmingham*
[b] *TWI Ltd, Abington, Cambridge, UK*

Abstract

Weld defect identification represents one of the most desired goals in the field of non-destructive testing (NDT) of welds. The current study investigates a system for assessing tungsten inert gas (TIG) welding using a high dynamic range (HDR) camera with the help of artificial neural networks (ANN) for image processing. This study proposes a new dataset [1] of images of the TIG welding process in the visible spectrum with improved contrast, similar to what a welder would normally see, and a model for computing a label identifying the welding imperfection. The progress (accuracy) achieved with the new system over varying degrees of categorisation complexity is thoroughly presented.

Keywords: automation, convolutional neural networks, HDR camera, vision, process monitoring, quality assessment

1. Introduction

Tungsten Inert Gas (TIG) welding [1, 2] is an arc welding process which is commonly employed in joining high-value precision components across a range of industries including automotive and aerospace. It was invented in the late 1930s when the requirement for joining magnesium became apparent. Since then, it has been adapted for joining steel, copper, aluminium and many other engineering alloys. It is traditionally a manually implemented process, capable of achieving higher quality weldments in comparison with other arc processes. However, it currently has many limitations, all of which are inherently related to its manual nature. It heavily relies on highly-skilled manual welders who are expensive and in very short supply. Furthermore, it lacks the flexibility to perform the welding of complex geometries and causes poor repeatability and relatively high scrappage rates.
To address these limitations, the automation of TIG weld process is crucial. The monitoring and control of the weld pool during the process are critical to the automation in order to ensure quality, consistency and repeatability [3, 4, 5, 6].

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

Email addresses: dvb594@student.bham.ac.uk (Daniel Bacioiu), geoff.melton@twi.co.uk (Geoff Melton), m.papaelias@bham.ac.uk (Mayorkinos Papaelias), robert.shaw@twi.co.uk (Rob Shaw)

The welding process control system using some form of neural network started with the introduction of neurofuzzy architecture by Zhang and Kovacevic [7, 8]. The neurofuzzy algorithm was capable of adapting and controlling the non-linear nature of TIG welding by establishing a correlation between the weld pool boundary and the top-side and back-side bead widths.

Over the past decades, a number of sensing approaches have been proposed in the literature as a means to monitor, in real-time, the weld pool. Among these approaches are the pool oscillation method [9, 10], ultrasonic testing [11, 12, 13], infrared sensing [14, 15] and specular weld pool surface [16, 17]. Lee et al.[18], Song et al. [19] and Lucas et al. [20] approach the concept of capturing and processing images of the weld pool. Most of these studies dedicate ample analysis for weld defects categorisation at different stages during or post-welding. The raw information for image processing and the entire approach deriving from it take two main avenues: radiographic images and visible spectrum images of welding. The two approaches have inspired each other with regards to the signal processing applied to the data, but fundamentally they approach the problem from two different points of view. While the radiography cannot monitor the weld pool, the acquisition of images in the visible spectrum has no limitation with regards to image collection. Conversely, the visible spectrum image acquisition needs to cope with the intense light emitted by the welding arc. To illustrates this, Zou et al. [21] proposed an algorithm for real-time weld defect detection in radiographic NDT of spiral pipes. The method is based on greyscale image thresholding and application of Kalman filtering for defects tracking and false alarms exclusion. The technique provides further resilience to noise and background inhomogeneity. In another work, Boaretto et al. [22] made use of double wall double image (DWDI) method for capturing radiographic images. These authors subsequently located the weld bead, segmented the discontinuities and extracted the features. These features were then processed with a feed-forward multilayer perceptron (MLP) for the classification of discontinuities into 'defect' and 'no-defect'. Hou et al. [23] proposed a system for identifying defects in x-ray images of welds based on Stacked Sparse Auto-Encoders (SSAE) and softmax classification. After training the system on a subset of GDXray dataset, the author achieved an accuracy of 91.84%.

On the other hand, using input that is more analogous to human vision, Jiang et al. [24] studied the extraction of weld pool width from an image by passing it through a series of transformations: intra-group variance thresholding, fill hole treatment (advance morphological operations), FFT, edge detection (Canny), edge curve fitting and finally weld pool width extraction. Liu et al. [25] designed a procedure for capturing the top side of keyhole plasma arc welding using a CCD camera and a filter with different rates of attenuation to counter the strong arc characteristics. The authors applied the Canny filter on the image to extract weld pool edges followed by a series of custom designed and very specific steps, tailored to plasma arc welding, to reconstruct the exact weld pool boundary. In another study, Fidali et al. [26] investigated a technique which uses two cameras to acquire the visual and infrared spectrum of MAG welding process, fuse the images, then extract arc area statistics, welding area topological features and assesses intensity profile along a line in arc and welding pool areas. The authors used the multitudes of these parameters to detect different welding con-

ditions. The end classification was carried out using k-nearest neighbour classifier achieving an accuracy around 50%.
In a suite of publications, Liu and Zhang [27, 28, 29] combined the specular weld pool surface representation of the TIG welding process with the adaptive neuro-fuzzy inference system (ANFIS) for establishing the correlation between the welding current and speed and the back-side bead width. In a subsequent study [30] the same authors attempted to fuse the human welders' robustness to weld pool variations and the robot's quick response capability for offsetting varying welding currents and input disturbances.
In our work, visual testing is the main focus of study, by identifying reliably and consistently defects present in weldments of aluminium 5083 using the TIG welding process. The current study assesses the suitability of 12 neural networks (NN) architectures covering convolutional neural networks (CNN) and fully-connected neural networks (FCN) with varying internal layouts. More precisely, the principal aim of the study is to provide an alternative method for analysing images of the TIG welding process by combining the HDR camera and novel processing paradigms as neural networks.

2. Methodology

Our approach for defect identification involves utilising a camera trailing the welding process oriented directly towards the weld pool to obtain the real-time images. Figure 1 depicts the schematic layout of the system while Figure 2 is a real-life setup.

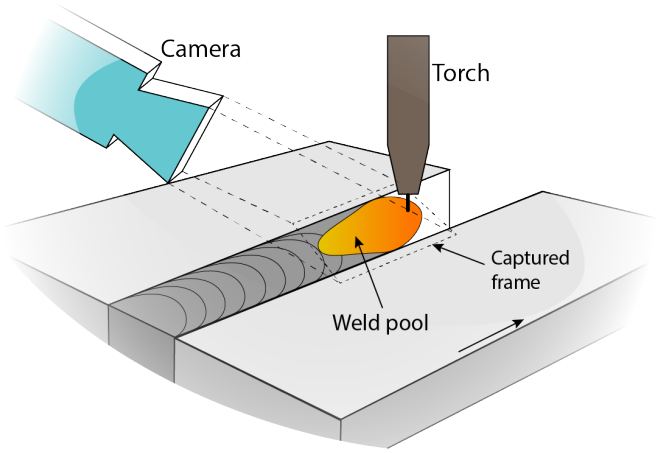

Figure 1: TIG welding process layout.

First, images of weld pools from "good welds" were captured and recorded then several defects were introduced and weld pools from "defective welds" were also recorded. The alteration of welding parameters, one at the time, helped with the defects introduction. For instance, a series of weld trials were performed in which the welding current was decreased in stages until "lack of penetration" was obtained. Note that all trials started with a standard "control" welding conditions given in subsection 2.1.

The images captured by the camera were then categorised based on the data representative of the defects targeted in this work. Note that this categorisation ranges from the basic "good weld vs defective weld" to more complex 6-class problem of "good weld vs burn through vs contamination vs lack of fusion vs misalignment and lack of penetration".

2.1. Welding process

The welding process of choice in the current study is TIG welding due to the arc stability, minimal spatter and weld pool stability, used for high-quality welds, which simplifies the images analyses.

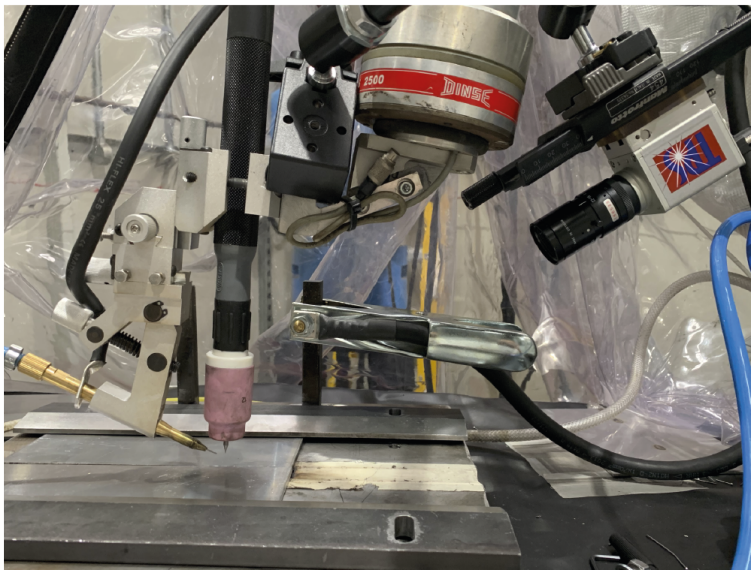

Figure 2: Camera and robot setup.

Aluminium 5083 grade was selected as the work material because it is extensively used in TIG welding. Its composition is detailed in Table 1. The plates thickness is 2mm, and the groove angle is 90°. Table 2 lists the welding parameters used in this work as a standard

Table 1: Aluminium 5083 composition.

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

"control" to achieve good welding conditions.

Table 2: Standard baseline welding parameters.

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

Figures 3 and 4 show the distribution of the input current and the travel speed for each class examined in this study by deviating from the baseline parameters.

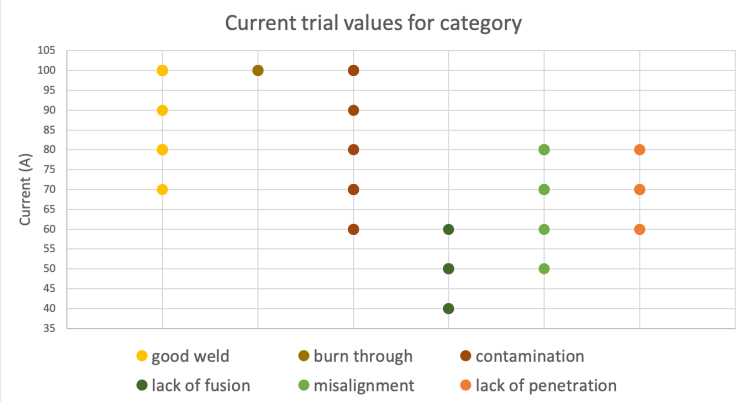

Figure 3: Current values for each category

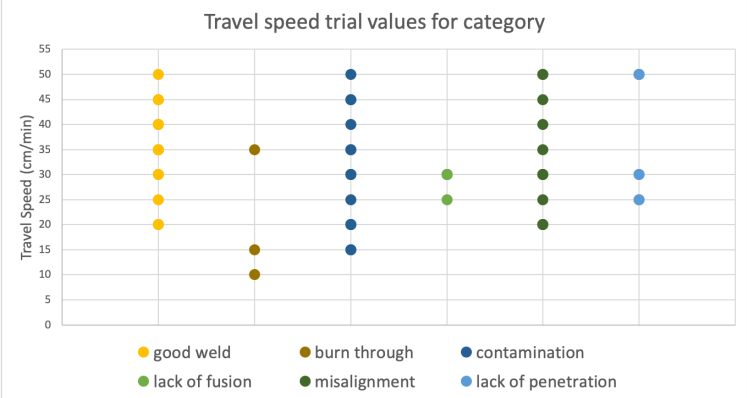

Figure 4: Travel speed values for each category

2.2. Camera capabilities

The camera used to record images is a Xiris XVC-1000. It uses high dynamic range (HDR) capability, of 140+dB, to diminish the arc luminosity and bring up details from the weld pool and surrounding area, generating tones with greater accuracy. Table 3 details the

camera specifications.
Figure 5 shows the workpiece after welding with the weld appearance as recorded by the camera attached.

Table 3: Xiris XVC-1000 specifications

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

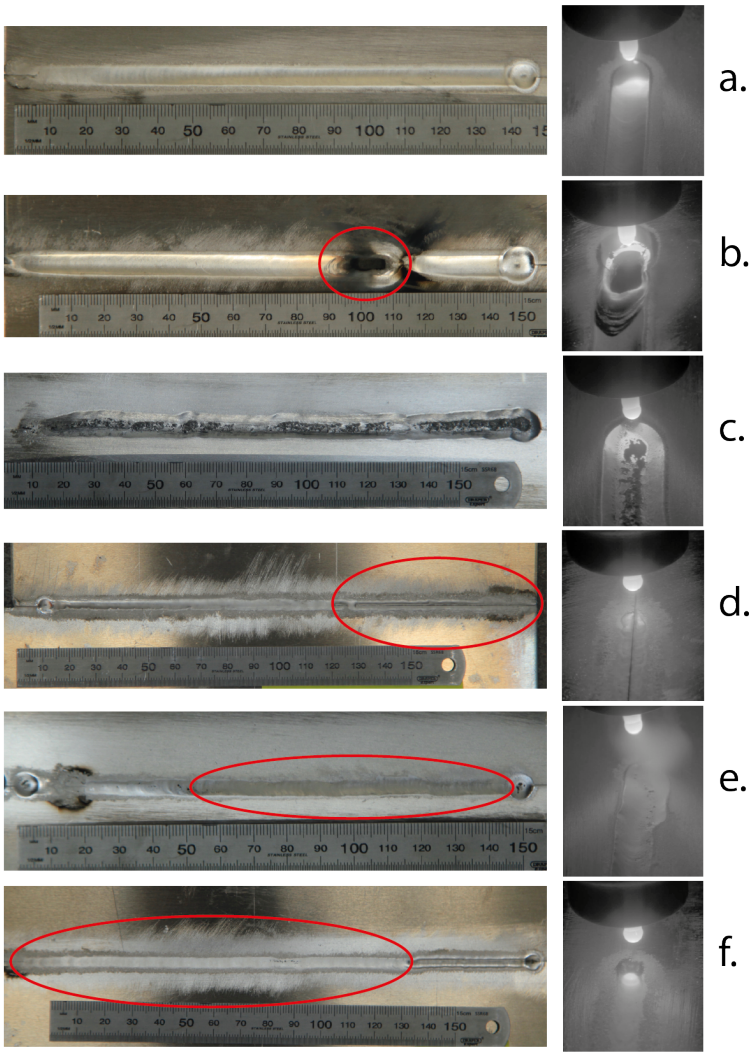

Figure 5: Dataset samples of aluminium TIG welding. a) good weld; b) burn through; c) contamination; d) lack of fusion; e) misalignment; f) lack of penetration

2.2.1. Data

The study aim is to extract information from an image representing the weld pool and surrounding area in the visible spectrum. We use a new dataset, recorded with the HDR camera, containing images with tone contrast similar to human eye vision, an example of which is given in Figure 5, composed of 60 welding trials.
Each trial generates images at 55 frames per second, producing relatively quickly a large amount of data. The dataset contains 33254 images of TIG welding of aluminium 5083. The images were split into 6 classes, listed in Table 4, by an experienced operator.

Table 4: Dataset split between training and test for 6-class test.

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

Table 5: Dataset split between training and test for 4-class test.

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

The processing architecture requires the data to be split into two main subsets: training and testing, with approximatively 75% and 25% share respectively. The data split are described in Table 4 and 6 are for the 6-class and 2-class analysis, respectively. The exception is the 4-class analysis, described in Table 5, which uses an 88%-12% split due to limitations in the generated data. The dataset for 4-class trial is a smaller version of 6-class and 2-class datasets where "misalignment" and "lack of penetration" are removed and classes with excessive number of examples reduced. The reduction excludes images from classes with excessive representation, i.e. "good weld". As a consequence it is less representative compared to the other two. All 6-class, 4-class and 2-class analysis use the data generated within the same set of welding experiments. Table 6 "defective" category incorporates all the defects from Table 4 under one single class.

Table 6: Dataset split between training and test for 2-class test.

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

As each weld generates hundreds of images, the problem of correlation between the train and test splits became apparent. To reduce the correlation, it was ensured that no weld trial could be found in both the train and test subsets, meaning that the dataset was split based on welding trial only and not by frame setting.

2.2.2. Preprocessing

The camera delivers frames of size 1280×1024 pixels, centred on the weld pool. The images contain a substantial amount of black pixels surrounding the weld pool and the welding arc as seen in Figure 6. Therefore, the cropping reduces the original size of 1280×1024 to 800×974. Further to cropping, the images are subsampled, reducing the size to 400×487. The subsampling operation is necessary because of the hardware constraints during the training stage of the networks. The model receiving a higher resolution requires significantly more GPU memory.

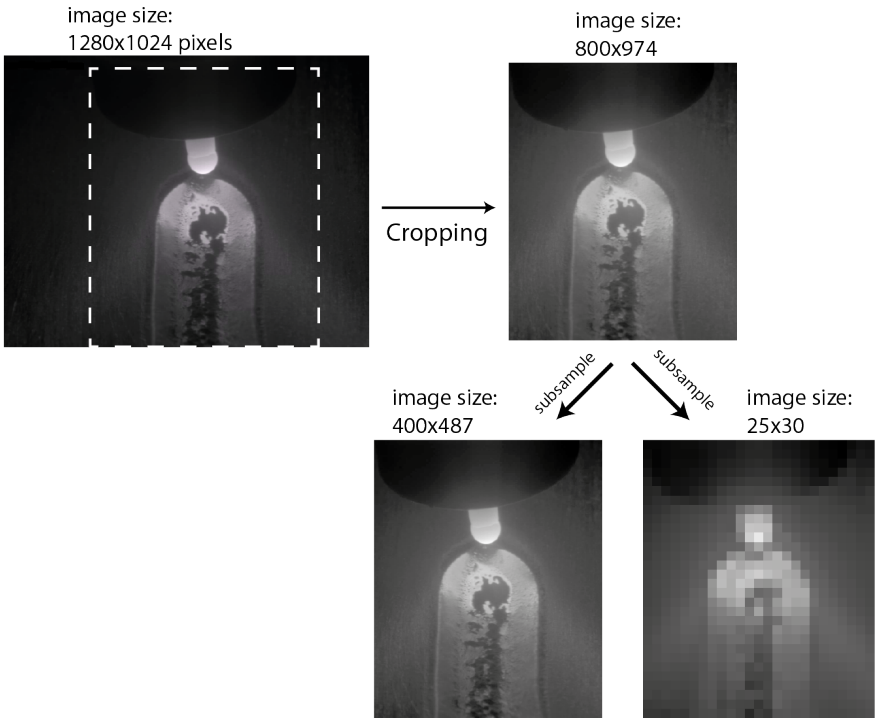

Figure 6: Image subsampling

The resolution reduction has the potential of influencing the resulting model accuracy. This study performs an ablation analysis on the effect of resolution reduction on the final accuracy. The images of 400×487 pixels are subsampled to 25×30 pixels and unsampled back to 400×487. The subsampling followed by upsampling has two reasons: it maintains the input image size of 400×487 pixels for neural networks and removes details from the baseline image of 400×487. The result is the isolation of fidelity as the only changing parameter across comparison.

2.3. Processing architecture

The processing paradigm evaluated in the current study is based on an artificial neural network (ANN) architecture [31, 32, 33]. ANN is a lattice of computations which reduces the input dimensionality, eventually arriving at desired labels. The reason for choosing ANN rests mostly in the processing adaptability across various lighting and appearance conditions. The study applies the fully connected neural network architecture (FCN) [34, 35, 36, 37] and convolutional neural network (CNN) [38] for processing the weld images. FCN and CNN are subsets of the larger ANN processing paradigm [39]. Both architectures have an input layer for receiving the image and an output layer for the classification label. In FCN there is a connection between each input pixel and every node from the subsequent layer. In case of CNN, the region for kernel application is limited to the kernel dimension. One of the first application of CNN [31] was character identification [40]. A representation of convolution dynamics is Figure 7, where the input image (green) convolves with the kernel (red) outputting one single value (purple).

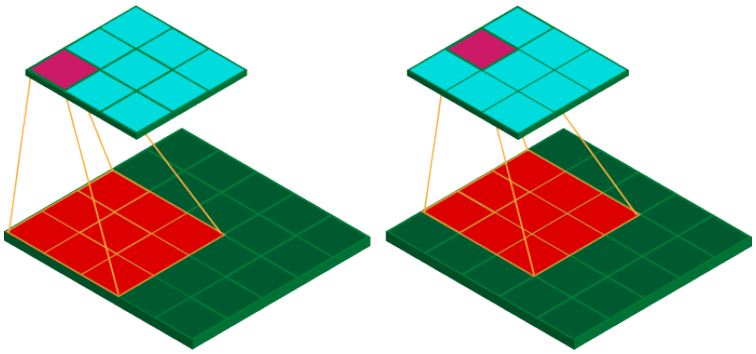

Figure 7: Convolution between a feature map of dimension 5×5 (green) and a kernel of dimension 3×3 (red), stride 1 and image padding 0. The result is another feature map of dimension 3×3 (turquoise).

By traversing the input image one pixel at a time (stride 1) horizontally or vertically, another image (feature map) is created (turquoise). The reduction in the number of parameters required to produce the same output is significant. As an example in the current displayed setting of Figure 7 there are $3 \times 3 = 9$ weights producing 9 outputs, while in the case of fully-connected networks (FCN), producing the same output would require $5 \times 5 \times 9 = 225$ weights because the number of parameters grows quadratically with the input image size.

The analysis in the current study covers 12 architectures, 6 CNNs and 6 FCNs. The main parameters defining the architecture variations for CNN are the convolutional kernel size, the number of kernels in each layer and stride, while for FCN the important parameters are number of layers and the number of units in each layer. Tables 7 and 8 describe in more details how layers are composed for each architecture.

Table 7: Fully connected neural network architectures

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

Table 8: Convolutional neural network architectures

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

The first layer, "conv:[5, 5]×[16]/1-relu", in model 1 in Table 8, contains 16 kernels with each kernel of dimension 5×5, stride 1, and the rectified linear unit, ReLu [41], activation function. It produces 16 feature maps of size 396×483 each. This implies 396×483×16 = 3060288 output values with 5×5×16 (weights) + 16 (biases) = 416 parameters. For comparison purpose, the same number of outputs, with a FCN, would require (396×483×16)*(400×487 + 1) = 596,147,162,688 parameters - almost impossible to process.

Following each convolution layer, there is maximum pooling sampling layer [42]. "max pool:[5, 5]/3" translates to a kernel size 5×5 and stride 3, which samples the largest value in the receptive field.

The basic blocks in this study are the 5×5 kernels with stride 1 and 2, 3×3 kernels with stride 1 and 2, maximum pooling layers of sizes 5×5 with stride 3 and 3×3 with stride 2. The convolution and maximum pooling operators are placed at the beginning of the network having an effect of feature reduction, minimising input from 194800 (multiplying 400×487) pixels to few hundreds.

After the convolution and pooling layers within each architecture, the image is flattened i.e., unrolled into a vector and denoted "flatten:[384]", followed by fully connected layers, denoted "matmul:[384, 256]" as they are matrix multiplications.

The last two hidden layers "matmul:[384, 256]-relu" and "matmul:[256, 128]-relu" are the same for all the networks to preserve some similarity between different architectures and assess the power of representations built in previous convolutional layers.

An example of the feature maps shrinkage due to convolution and maximum pooling is presented in Figure 8.

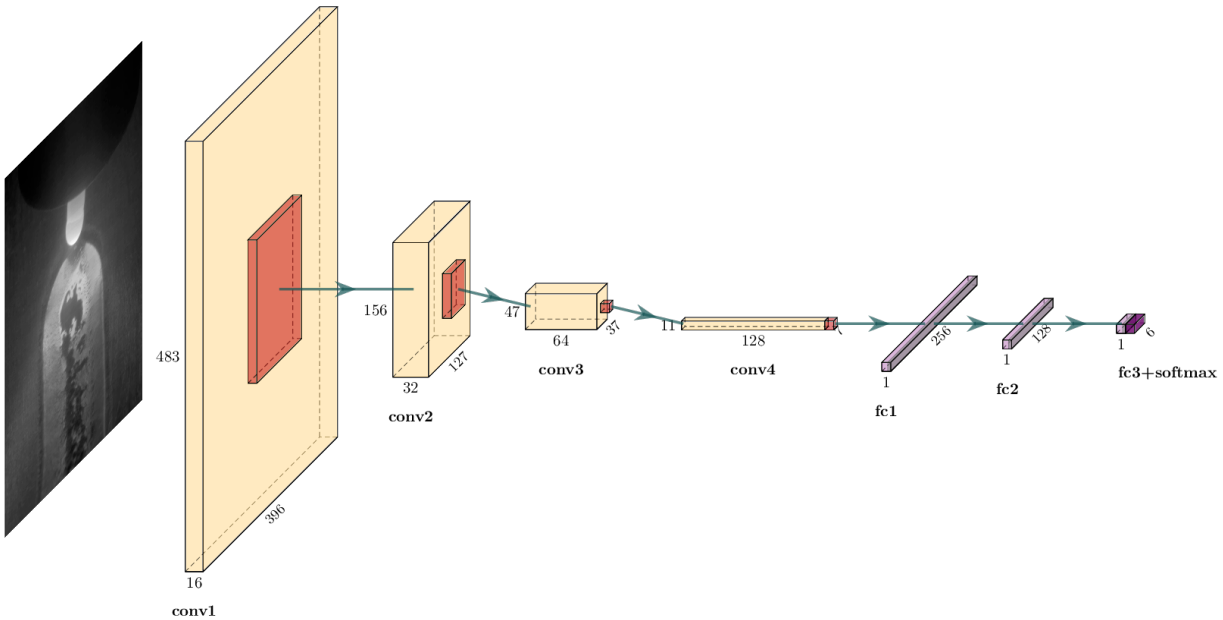

Figure 8: An example of visualisation of feature maps dimensions for each convolution and fully connected layers for Model 1.

2.3.1. Convergence

On top of the architecture variation, there is the process of learning, or more precisely, finding the probability distribution describing the dataset. The probability distribution is parametrised by the weights composing the kernels and fully connected layers in the architecture described in Subsection 2.3. The convergence algorithm used in the current study is called Adaptive Moment Estimation (Adam) [43]. The training parameters are described in Table 9

Table 9: Training parameters.

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

In summary, this study evaluates the neural networks accuracy on classifying weld images with defects. The variable under investigation are as follows:

Table 10: Varying parameters.

Model	12 architectures
Learning rate	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4} and 10^{-5}
Classification difficulty	2-class 4-class 6-class
Input images size	400x487 px 30x25 px

3. Results & Discussion

3.1. 6-class test

The most challenging test for the neural networks is the 6-class classification, for which performance over a range of architecture and learning rate variations are described in Table 11.

The architectures highlight a significant degree of performance difference between CNN and FCN. The CNN models outperform and build better representations from HDR input images compare to FCN with a performance gap of 18% on average.

Table 11: Model accuracy for the learning rate and model for 6-class classification.

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

The performance numbers presented in Table 11, on the last column (dark grey background), represents the average of the three light grey columns, 10^{-3}, 10^{-4} and 10^{-5} and it measures the model's stability across a range of values for learning rate. The first two columns (10^{-1} and 10^{-2}) are omitted because the networks are not successful in representing the underlying dataset probability distribution for the learning rate chosen. 10^{-1} and 10^{-2} learning rates were too high for the architecture's internal parameters to converge to a state representative of dataset's probability distribution. In conclusion, all the learning rate greater than 10^{-2} are unsuitable for converging to an acceptable solution.
Figures 9 shows the same set of models as Table 11, displayed graphically, while Figure 10 clusters the models based on the number of layers, highlighting the importance for increasing the networks depth.
The limit in this case is the processing power available, which for this study, was 4GB Nvidia GeForce GTX 980, able to accommodate a model of up to 12 layers. The pattern is also found in FCN, architectures with 5 hidden layers exhibiting an accuracy advantage over architectures with 4 hidden layers.

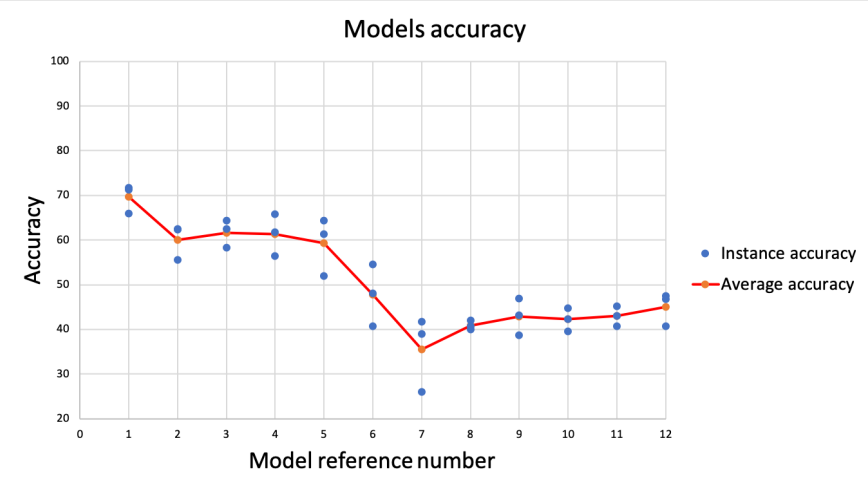

Figure 9: Average accuracy for models trained with the learning rates 10^{-3}, 10^{-4} and 10^{-5} for 6-class classification.

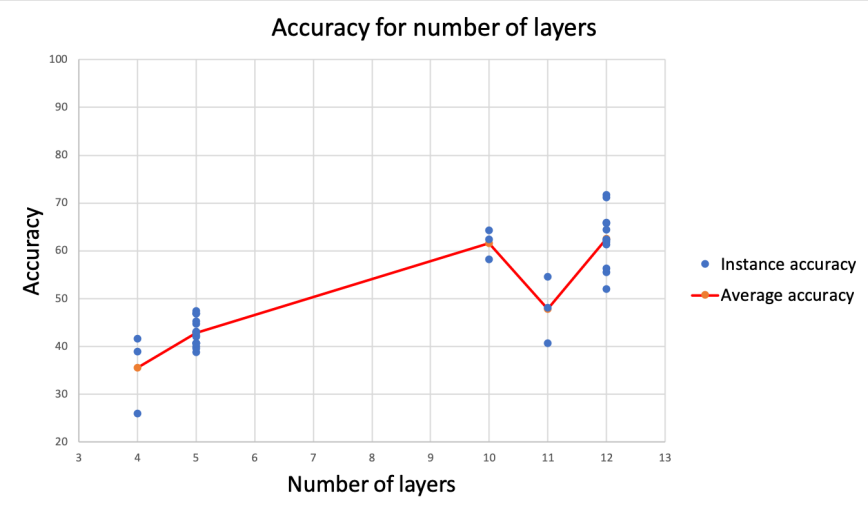

Figure 10: Average accuracy as a function of the number of layers for 6-class classification.

The main interest of the study is to perform hyper-parameters' (e.g. learning rate) ablation analysis on values that led to relatively successful models. A relatively successful model is one that achieves an accuracy exceeding 40% since the models tend to classify everything as the same class. In the current case that class is "good weld".

3.2. 4-class test

Table 12 shows the performance results for training the the same architecture on classifying 4 types of welds.

Table 12: Model accuracy for the learning rate and model for 4-class classification.

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

The average performance difference between CNN and FCN shrinks to 9.5%. Figures 11 and 12 show the average accuracy for each class and average accuracy variation for model depth increase.

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

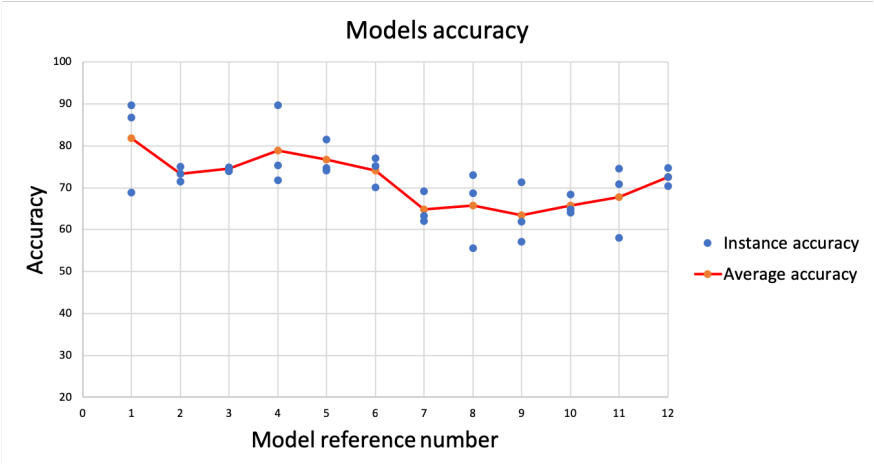

Figure 11: Average accuracy for models trained with the learning rates 10^{-3}, 10^{-4} and 10^{-5} for 4-class classification.

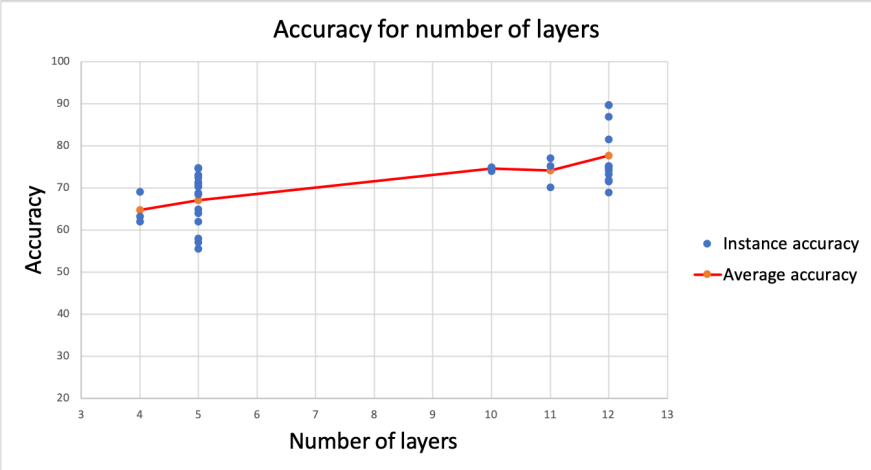

Figure 12: Average accuracy as a function of the number of layers for 4-class classification.

3.3. 2-class test

The reason almost all architectures achieve an accuracy of 67% over 2 classes in Table 13 when the random classification probability is 50% is due to a slight imbalance in test dataset (2189 good weld samples and 4399 defective). All examples are categorised as defective (4399/(2189+4399) = 0.67), therefore no representation of data was achieved.

Table 13: Model accuracy for the learning rate and model for 2-class classification.

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

The results in Figure 11 and 12 show smaller performance gap between CNN and FCN with a difference of 7.5% in favour of CNN. Analysing the samples labelled incorrectly, it is observed that most images of "contamination" are misclassified, in almost all cases, indicating the defect is not represented sufficiently well in training dataset.
The 2-class performance mirrors the pattern seen in 6-class with the exception of overall

accuracy. Since only two classes are present the entire graph is shifted up reflecting simpler categorisation (see Figures 13 and 14).

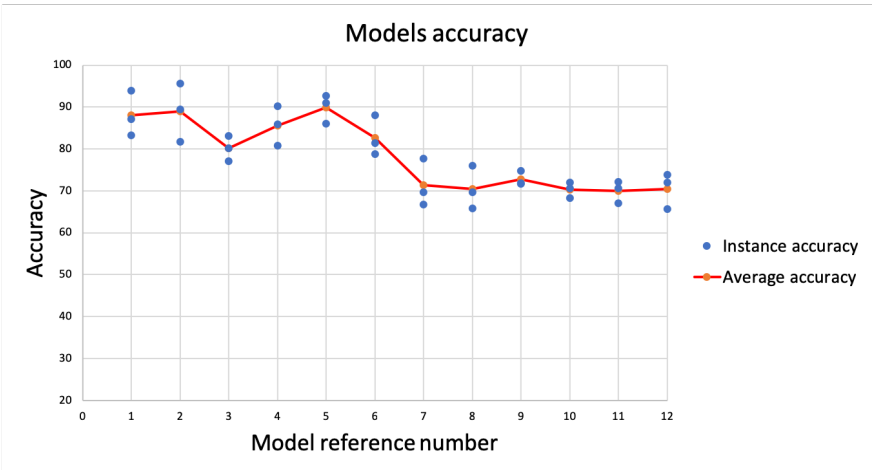

Figure 13: Average accuracy for models trained with the learning rates 10^{-3}, 10^{-4} and 10^{-5} for 2-class classification.

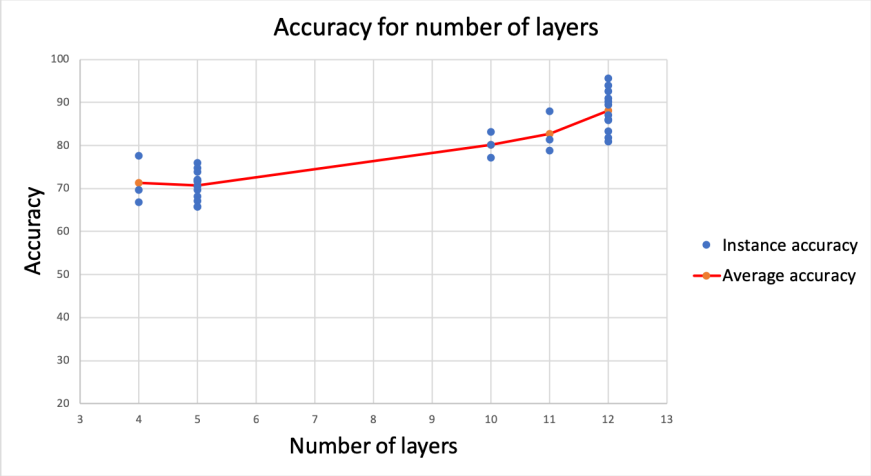

Figure 14: Average accuracy as a function of the number of layers for 2-class classification.

3.4. Training stability

The training duration is 5 epochs for all the neural network architectures. Figure 15 shows the loss evolution during training stage, significant of the convergence towards a solution. In this case the loss graph represents an instance of model reference 4, but the same trend could be observed for all the models.
Figure 16 shows the test accuracy at different points during training, notably at every epoch's end.

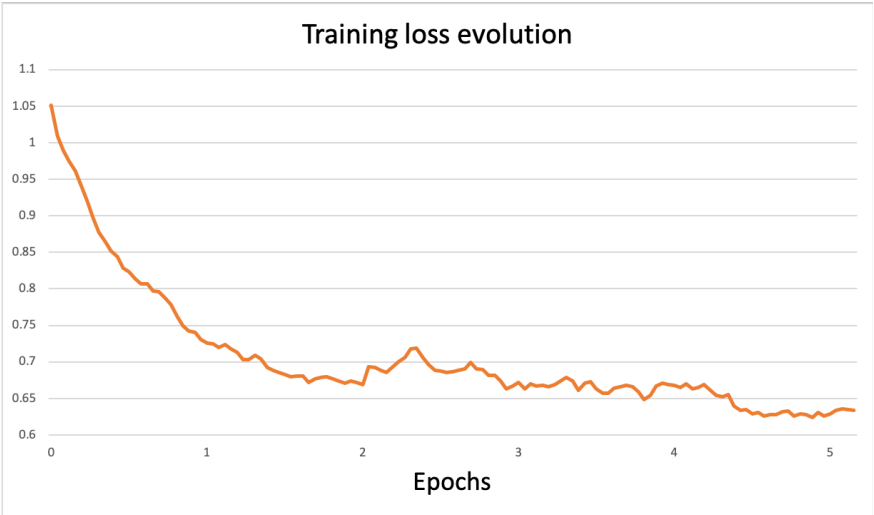

Figure 15: Training loss evolution.

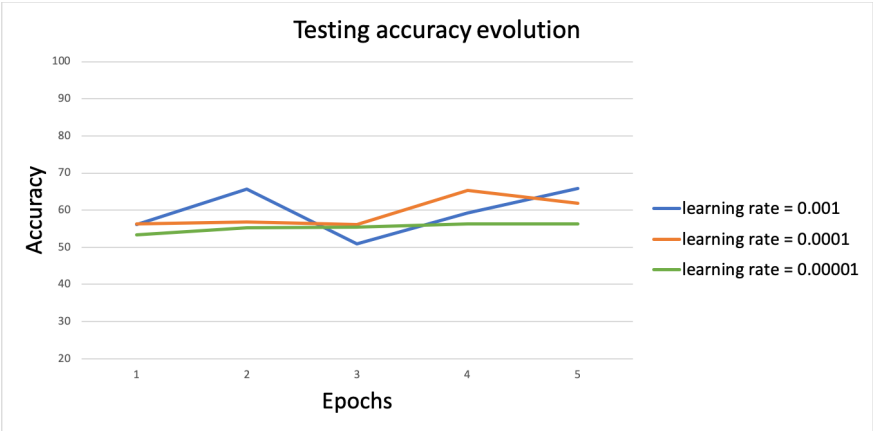

Figure 16: Test accuracy evolution.

Figures 15 and 16 highlight the models' rapid converge during the first epoch, while during the next 4 epochs, the models either maintain the same accuracy or at best improves it marginally.

3.5. Subsampling influence

This study analysis the impact of the resolution reduction on the final accuracy performance. The result are presented in Figures 17, 18 and 19, for the 6-class, 4-class and 2-class, respectively. Models trained using the images subsampled to 25×30 pixels then upsampled to 400×487 are compared against models trained with the images subsampled to 400×487 pixels. The fidelity reduction impacts severally the CNN architectures, particularly when the problem difficulty increases and the distinction between classes became harder, as in 6-class test. The FCNs show small decrease in accuracy performance, therefore concluding

the architectures are agnostic to inaccurate pixel values and less sensitive to the gradient between pixels values.

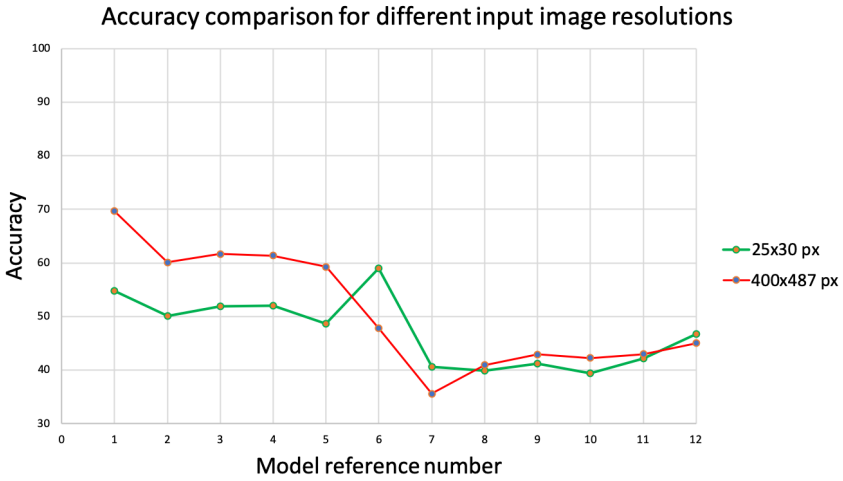

Figure 17: Subsampling accuracy impact for 6-class defect.

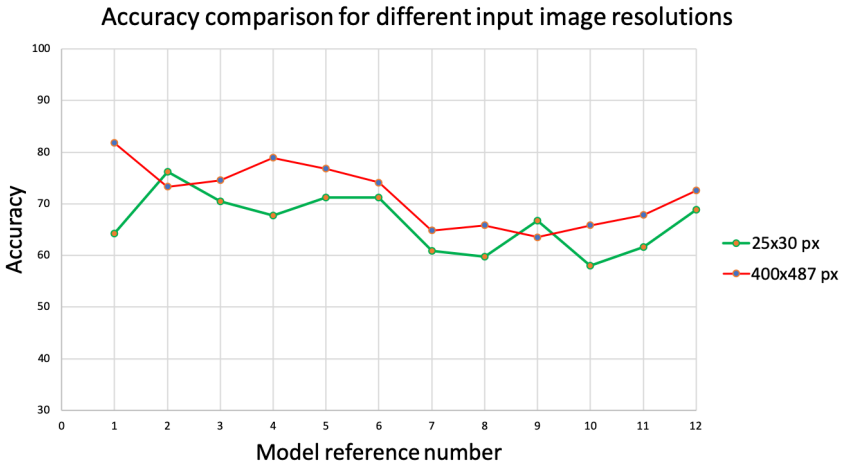

Figure 18: Subsampling accuracy impact for 4-class defect.

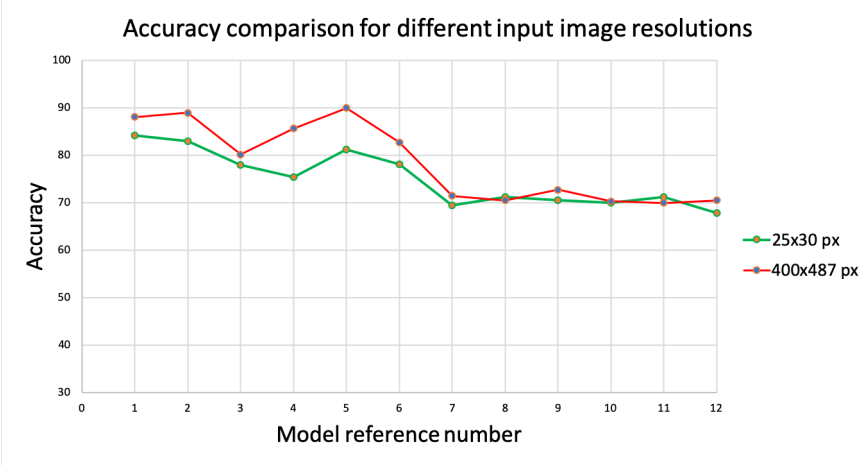

Figure 19: Subsampling accuracy impact for 2-class defect.

4. Conclusion

This study investigates a system for weldment classification using a vision system composed of a HDR camera able to offset powerful arc light and capture more detailed imaging of weld pool paired with the new and adaptive classification paradigm of neural networks. The analysis involved the construction of models based on CNN and FCN, varying internal architecture and hyper-parameters influencing the convergence of internal parameters for representing dataset's probability distribution. The models were trained using 6-class, 4-class and 2-class tests with highest accuracies of 71%, 89% and 95%, respectively. Furthermore, this study performed neural networks' robustness examination over a set of problems (6-class, 4-class and 2-class), highlighting the parameters influencing the model performance, concluding the architecture is the most important aspect given the learning rate is adequately chosen. The study also cover, the critical analysis of the errors linked to the input images fidelity reduction, highlighting the decrease in the accuracy performance.
The system required the generation of a new TIG welding dataset representing good welds as well as different types of common defects. This study contributes with 33254 images covering five welding defects.

Acknowledgements

The authors are indebted to the Engineering and Physical Sciences Council (EPSRC) and TWI Limited for providing the funding for this work.
The authors also gratefully acknowledge the provision of facilities from TWI Limited, the National Structural Integrity Research Centre and the School of Metallurgy and Materials of the University of Birmingham.

References

[1] L.F. Jeffus. *Welding: Principles and Applications*. Thomson/Delmar Learning, 2004.

[2] A. O'Brien and American Welding Society. *Welding Handbook: Welding processes*. Welding Handbook. American Welding Society, 2004.
[3] R. Kovacevic, Y. Zhang, and L. Li. Monitoring of weld joint penetration based on weld pool geometrical appearance. *Welding Journal - WELD J*, 75, 01 1996.
[4] M. Vasudevan, N. Chandrasekhar, V. Maduraimuthu, A. K. Bhaduri, and B. Raj. Real-time monitoring of weld pool during gtaw using infra-red thermography and analysis of infra-red thermal images. *Welding in the World*, 55(7):83–89, Jul 2011.
[5] J. Liu, Z. Fan, S.I. Olsen, K.H. Christensen, and J.K. Kristensen. Boosting active contours for weld pool visual tracking in automatic arc welding. *IEEE Transactions on Automation Science and Engineering*, 14(2):1096–1108, April 2017.
[6] C. Li, Y. Shi, Y. Gu, and P. Yuan. Monitoring weld pool oscillation using reflected laser pattern in gas tungsten arc welding. *Journal of Materials Processing Technology*, 255:876 – 885, 2018.
[7] Y.M. Zhang, L. Li, and R. Kovacevic. Neurofuzzy model based control of weld fusion zone geometry. In *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, volume 4, pages 2483–2487 vol.4, June 1997.
[8] R. Kovacevic and Y.M. Zhang. Neurofuzzy model-based weld fusion state estimation. *IEEE Control Systems Magazine*, 17(2):30–42, April 1997.
[9] R.J. Renwick and R.W. Richardson. Experimental investigation of GTA weld pool oscillations. *Welding Journal*, 62(2):29 – 35, 1983.
[10] Y.H. Xiao and G. den Ouden. A study of GTA weld pool oscillations. *Welding Journal*, 69(8):298 – 293, 1990.
[11] L.A. Lott. Ultrasonic detection of molten/solid interfaces in weld pools. *Material Evaluation*, 42:337 – 341, 1983.
[12] D.E. Hardt and J.M. Katz. Ultrasonic measurement of weld penetration. *Welding Journal*, 63(9):273 – 281, 1984.
[13] N.M. Carlson and J.A. Johnson. Ultrasonic sensing of weld pool penetration. *Welding Journal*, 67(11):239 – 246, 1988.
[14] W. Chen and B.A. Chin. Monitoring joint penetration using infrared sensing techniques. *Welding Journal*, 69(4):181 – 185, 1990.
[15] P. Ghanty, M. Vasudevan, D.P. Mukherjee, N.R. Pal, N. Chandrasekhar, V. Maduraimuthu, A.K. Bhaduri, P. Barat, and B. Raj. Artificial neural network approach for estimating weld bead width and depth of penetration from infrared thermal image of weld pool. *Sci. Technol. Weld. Join.*, 13(4):395 – 401, 2008.
[16] W. Zhang, Y. Liu, X. Wang, and Y. Zhang. Characterization of three-dimensional weld pool surface in gas tungsten arc welding. *Welding journal*, 91, 07 2012.
[17] W. Zhang, X. Wang, and Y. Zhang. Analytical real-time measurement of a three-dimensional weld pool surface. *Measurement Science and Technology*, 24:5011–, 11 2013.
[18] S.K. Lee and S.J. Na. A study on automatic seam tracking in pulsed laser edge welding by using a vision sensor without an auxiliary light source. *Journal of Manufacturing Systems*, 21(4):302 – 315, 2002.
[19] H.S. Song and Y.M. Zhang. Three-dimensional reconstruction of specular surface for a gas tungsten arc weld pool. *Measurement Science and Technology*, 18(12):3751, 2007.
[20] W Lucas, D Bertaso, G Melton, J Smith, and C Balfour. Real-time vision-based control of weld pool size. *Welding International*, 26(4):243–250, 2012.
[21] Y. Zou, D. Du, B. Chang, L. Ji, and J. Pan. Automatic weld defect detection method based on kalman filtering for real-time radiographic inspection of spiral pipe. *NDT & E International*, 72:1 – 9, 2015.
[22] N. Boaretto and T.M. Centeno. Automated detection of welding defects in pipelines from radiographic images dwdi. *NDT & E International*, 86:7 – 13, 2017.
[23] W. Hou, Y. Wei, J. Guo, Y. Jin, and C. Zhu. Automatic Detection of Welding Defects using Deep Neural Network. In *Journal of Physics Conference Series*, volume 933 of *Journal of Physics Conference Series*, page 012006, January 2018.

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

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

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

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

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

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

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

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

[32] Y. LeCun, L. Bottou, G.B. Orr, and K-R Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 9–50, London, UK, UK, 1998. Springer-Verlag.

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

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

[35] R. Salakhutdinov and G. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 412–419, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.

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

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

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

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

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

[41] V. Nair and G.E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Frnkranz and Thorsten Joachims, editors, *ICML*, pages 807–814. Omnipress, 2010.

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

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