Article

Deep learning and conventional machine learning for image-based in-situ fault detection during laser welding: A comparative study

Christian Knaak [1,*], Moritz Kröger [2], Frederic Schulze [1], Peter Abels [1] and Arnold Gillner [1]

[1] Fraunhofer-Institute for Laser Technology ILT, Steinbachstrasse 15, 52074 Aachen, Germany
[2] RWTH Aachen University Chair for Laser Technology, Steinbachstrasse 15, 52074 Aachen, Germany
* Correspondence: christian.knaak@ilt.fraunhofer.de

Abstract: An effective process monitoring strategy is a requirement for meeting the challenges posed by increasingly complex products and manufacturing processes. To address these needs, this study investigates a comprehensive scheme based on classical machine learning methods, deep learning algorithms, and feature extraction and selection techniques. In a first step, a novel deep learning architecture based on convolutional neural networks (CNN) and gated recurrent units (GRU) is introduced to predict the local weld quality based on mid-wave infrared (MWIR) and near-infrared (NIR) image data. The developed technology is used to discover critical welding defects including lack of fusion (false friends), sagging and lack of penetration, and geometric deviations of the weld seam. Additional work is conducted to investigate the significance of various geometrical, statistical, and spatio-temporal features extracted from the keyhole and weld pool regions. Furthermore, the performance of the proposed deep learning architecture is compared to that of classical supervised machine learning algorithms, such as multi-layer perceptron (MLP), logistic regression (LogReg), support vector machines (SVM), decision trees (DT), random forest (RF) and k-Nearest Neighbors (kNN). Optimal hyperparameters for each algorithm are determined by an extensive grid search. Ultimately, the three best classification models are combined into an ensemble classifier that yields the highest detection rates and achieves the most robust estimation of welding defects among all classifiers studied, which is validated on previously unknown welding trials.

Keywords: real-time quality prediction; spatio-temporal features; feature importance; recurrent neural network; high-speed infrared imaging; convolutional neural network; lack of fusion (false friends)

1. Introduction

Process monitoring and fault detection is an essential requirement for a multitude of manufacturing processes. In particular, complex joining processes such as laser welding (LW) require suitable quality monitoring procedures in order to satisfy the constantly increasing demands for high-quality products in modern and flexible production environments. In laser deep-penetration welding, a laser beam is focused on the material's surface. The energy provided by the laser radiation heats the welding material and as a result, the temperature in the laser beam focus exceeds the boiling point of the material. This leads to a vapor capillary (keyhole) which increases the penetration depth of the laser beam into the material due to the occurrence of multiple reflections within the keyhole. Although laser welding processes are well known, automated in-line quality diagnosis still remains a challenge [1]. In practice, weld quality is affected by several factors, such as thermal conditions during laser-material interaction, variations in material properties, impurities on the workpiece surface, and changes in the properties of the laser beam, all of which result in an unacceptable product [2,3]. During laser welding the complex interaction between laser beam and the weld material can lead to weld imperfections such as cavities, solid inclusion, lack of fusion as well as lack of penetration, weld seam deformations, cracks, and other deviations from the desired weld quality. A reliable quality diagnosis tool must provide high sensitivity for critical defects but also a certain adaptability in case of required process changes.

A common method for monitoring the laser welding process is to observe the radiation emitted by the keyhole via high-speed photo diodes. The keyhole is an out-gassing channel for vaporized material and process gases. As a result of the outflowing gases and the incoming laser radiation, a plasma plume originates above the material's surface on the keyhole position. With respect to in-process monitoring, the electromagnetic signature of the keyhole and plasma plume can be observed and correlated with quality-related phenomena, occurred during the weld process [4,5]. Unfortunately, the correlations of those signals to certain quality criteria are often ambiguous, so that statistical proof of quality by destructive testing is necessary.

However, recent advances in sensing technology and an increasing number of sensors applied on laser machines and processes, enables online weld quality monitoring with higher precision by combining multiple data sources. Similarly, complex sensors such as thermal camera systems have become reasonably priced and can be used as a data source for in-process weld quality monitoring. Recently used sensors for laser welding process monitoring are image-based sensors such as cameras in the infrared wavelength range [6], acoustic emission sensors, optical sensor such as high-speed photodiodes and pyrometer [7]. Also techniques such as x-ray imaging, spectrographically sensors [8] and combined sensing techniques have been investigated [9]. Especially, camera sensors provide important information from various process zones that emerge during laser welding. The keyhole is typically surrounded by molten material, the weld pool. Size and shape of weld pool are important geometrical parameters that correlate with weld shape and quality [10,11].

Due to high process dynamics and partially chaotic keyhole behaviors [12], an approach based on precise physical modelling of the welding process is not practical for real-time quality diagnosis of laser welds [13]. On the other side, the incorporation of new technologies such as Industrial Internet of Things (IIoT) and advanced analytics into manufacturing systems aims to produce individualized products at high quality and low costs. In the manufacturing domain, such data-driven approaches have been extensively studied in the past and are based on autoregressive (AR) models, cluster analysis, fuzzy set theory or on supervised learning algorithms such as multivariate regression, multi-layer perceptron and decision trees, as well as k-nearest neighbors [14,15]. Therefore, recent development led to advanced process monitoring systems which integrate machine learning techniques for process control and prediction of critical defects [16,17]. An advantage of data-driven methods is that it is not necessary to explicitly model the physical behavior of the system in order to build a statistical model. However, process understanding can help to design and develop the right feature set and to select relevant sensors and signal sources as input for the data-driven model. A data-driven model utilizes input variables (features) extracted from the raw signals to establish a statistical model between those features and the observed phenomena, e.g., weld defects during the welding process. Therefore, features that describe the significant characteristics of the signal are required for classical supervised learning algorithms and are often manually designed and depend on signal type (e.g., image data, data from high-speed photo diodes) and the output variable. For example, You et al. [18] proposed diagnosis system for autonomous laser beam welding. This system is based on extracting features with wavelet packet decomposition and dimensionality reduction techniques (PCA) in combination with SVM-based classification for defect detection. An extensive experimental setup has been established to evaluate the proposed methods comparing measurements signals from photodiodes, image sensors and x-ray analysis. However, the question remains which features are necessary to achieve high defect detection accuracies and how different learning algorithms may improve the detection performance. Still, machine learning is not only used for defect detection in laser welding. Different machine learning regression algorithms, i.e., different types of artificial neural networks and support vector regression (SVR), were used by Cai et al. [19] to predict the weld bead width. The authors used seven features extracted from the welding images recorded during the process to describe geometrical properties of the keyhole and weld pool. Overall, the results show that the investigated algorithms can accurately predict welding quality in real time.

From the field of computer vision and pattern recognition, deep learning methods have emerged as an effective technique to solve signal- and image processing tasks [20,21]. Deep learning is different from classical machine learning as it integrates the process of feature extraction within the data-driven model. Deep learning models with multiple layers of artificial neurons are based on the findings in neuro-science that multi-stage deep neural networks allow humans to perform complex signal processing tasks such as object- and voice recognition [22,23]. As a result, deep learning models are capable of extracting more refined and complex image characteristics and are therefore expected to provide higher classification accuracies than conventional approaches based on feature engineering and traditional classifiers. With the advent of deep learning, especially convolutional neural networks (CNN), top rankings in classification performance were achieved in several image recognition competitions such as ImageNet in 2012. CNNs have therefore become a common solution for many computer vision tasks [24]. Nowadays, it is possible to train large multi-layered CNN networks, typically consisting of many types and numbers of layers on GPU-hardware, with the help of open source deep learning frameworks such as TensorFlow [25], PyTorch [26] or Caffe [27].

This has led to various applications of CNNs in industrial production sector to recognize defects and improve product quality [28]. Therefore, it is no surprise that deep learning has recently been used in laser welding applications to predict defects.

For example in 2014, Günther et al. [29] suggested a deep learning scheme for extracting relevant features from in-process laser welding data. They used a deep learning-based auto-encoder with fully connected layers to create a new latent feature space of 16 features that describe the welding images. With the help of these features they used an SVM to predict the photodiode welding signal in the near feature based on image features. Higher prediction accuracies were achieved compared to an approach using PCA. In 2019 Zhang et al. [30] presented a CNN-architecture that uses features extracted from image and photodiode signals recorded during laser welding to detect welding imperfections. The approach shows promising results compared to a traditional ANN model, although it was not used to extract features from raw sensor signals.

Thermal images and convolutional neural networks work well in combination, as shown by Gonzales-Val et al. [31]. They proposed a CNN architecture to predict dilution in laser metal deposition as well as defects in laser welding based on infrared images. Experimental results show promising results with respect to the prediction accuracy.

For CO2 laser welding, a combination of CNN and a recurrent neural network (RNN) was applied to extract primary features from weld pool images. Although, RNNs are used to model sequence-based problems such as voice recognition. In this approach, RNNs were used to fuse features extracted via CNN from a single image with the help of an RNN to recognize good and imperfect weld images [32].

Besides the manufacturing domain, architectures based on CNN and RNN turned out to be successful in applications such as action and emotion recognition in video data [33,34]. Additionally, a group of researchers utilized CNN and RNN architectures to improve prediction accuracy of the steering angle of an autonomous vehicle. They achieved lowest error compared to other approaches in the literature [35].

In this work, geometrical and statistical features are extracted from thermal image data (MWIR & NIR) recorded during the laser welding process to determine the keyhole- and weld pool characteristics for each time step (section **Error! Reference source not found.**). The features are based on higher order image moments, shape descriptors and descriptive image statistics as well as statistics in the time domain, that are used to establish a high-dimensional feature vector. Subsequently, the performance of manually extracted features in combination with different classical machine learning algorithms such as SVC, LogReg, DT, ANN, KNN and RF is evaluated for quality prediction during the welding process. In addition, we determine the significance of individual features and the relevance of different feature subsets in terms of their classification performance. Furthermore, a new deep learning-based approach for data-driven feature extraction and weld defect detection is introduced and investigated. The architecture is based on convolutional neural networks (CNN) which are often used for image classification and is further described in section **Error! Reference source not found.**. Although in-process data are available in form of images, some important information may only be available in the time-domain of the welding video stream. Therefore, the CNN is combined with a recurrent neural network (RNN), specifically the gated recurrent unit (GRU) architecture as described in section **Error! Reference source not found.**, that was recently used to solve pattern recognition tasks in the time-domain [36]. The advantage of CNNs to extract relevant spatial information and the ability of GRUs to learn meaningful temporal characteristics are combined to automatically extract a spatio-temporal feature representation of a given image sequence. Additionally, an architecture based on CNN only is employed as a reference. Subsequently, all models are optimized using a grid search process combined with nested cross validation. In a further step, the deep learning architectures are compared with classical machine learning approaches based on the individual prediction performance in four unseen welding trials. Ultimately, a combination of three unique models is proposed as an ensemble classifier to predict the seam quality during the welding process based on majority vote (section **Error! Reference source not found.**). A schematic overview of the data processing and evaluation steps applied in this work is given in **Error! Reference source not found.**. Overall, the main contributions of this work include the following points:

- Assessment of the significance of geometric and statistical features extracted from the keyhole and weld pool region of two different image data sources (i.e., MWIR and NIR) with respect to the ability to detect particular weld defects.
- Development and evaluation of a unique deep learning architecture combining CNNs and GRUs to extract spatio-temporal features from image sequences.
- Comparison of classical machine learning methods (i.e., DT, kNN, LogReg, SVM, ANN, RF) and modern deep learning architectures with respect to prediction accuracy, F1-score as well as training and inference time using an experimental data set.
- Combination of the top-three classification models as an ensemble classifier based on majority vote to robustly detect critical defects such as lack of fusion, sagging, seam width deviations and lack of penetration during laser welding.

From here, the remaining part of this paper shows the following structure. Section **Error! Reference source not found.** provides the background knowledge for different classification algorithms as well as a definition of the proposed CNN-GRU architecture. Section **Error! Reference source not found.** describes the experimental setup and the process of feature extraction. Experimental results are presented and analysed in Section **Error! Reference source not found.**. Finally, a conclusion is given in section **Error! Reference source not found.**.

2. Methodology and Background Knowledge

In this work several conventional machine learning algorithms are compared to each other in terms of prediction performance and processing time. These algorithms and the resulting prediction model often require feature engineering as a preliminary stage, especially in the field of image recognition, in order to create predictive models. not only with a high prediction performance and less overfitting, but also with fast execution times and a higher degree of comprehensibility. The investigated conventional machine learning algorithms are listed below:

- Decision tree (DT),
- K-nearest neighbor (kNN),
- Random forests (RF),

- Support vector machines (SVM),
- Logistic regression (LogReg),
- Artificial neural networks (ANN).

A detailed overview and discussion of these algorithms can be found in several textbooks such as [37,38] and [39,40]. The method of feature engineering and classification using a conventional algorithm is additionally compared to modern deep learning approaches, which include the process of feature extraction as part of the model. The following types of deep learning algorithms [41],

- Convolutional neural networks (CNN) and
- Gated recurrent units (GRU),

are used in this work for defect detection during laser welding processes. Both, conventional and deep learning algorithms use the following data set D as input to establish a data-driven model:

$$D = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) | \, \boldsymbol{x}_i \in \mathbb{R}^p, \boldsymbol{y}_i \in \mathbb{R}^m, i = 1{,}2 \ldots, Q\} \tag{1}$$

Where $\boldsymbol{x}_i$ denotes the ith feature vector, which for conventional machine learning methods consists of numerous features p, that are explained in **Error! Reference source not found.** and

Table **4** more detailed. For deep learning algorithms, the feature vector $\boldsymbol{x}_i$ represents a raw image or image sequence in the data set. The label vector, described by $\boldsymbol{y}_i$, belongs to the feature vector $\boldsymbol{x}_i$ while m denotes the number of classes, which in this work represents the six different welding quality states as stated in section **Error! Reference source not found.**. For this study, the DT, LogReg, SVM, ANN, kNN and RF implementations of scikit-learn 0.22.1 and Python 3.6 are used to train classification models [42]. All hyperparameters that were optimized via grid search and 4-fold nested cross validation, can be obtained from **Error! Reference source not found.**. For all other algorithm hyperparameters not listed in **Error! Reference source not found.**, the default values of the scikit-learn implementation are used. In the subsequent section, a more detailed description regarding the combination of CNN and GRU architectures used in this work is given.

2.1. *Deep learning with CNN and GRU*

Although a multilayer perceptron (MLP) with multiple hidden layers can be viewed as a deep neural network, these networks are not necessarily capable of extracting relevant information from complex raw data such as images or audio signals. For example, connecting every pixel of an image to each node in a hidden layer results in a high amount of parameters that need to be trained, which is computationally intensive and may result in overfitting. Modern deep learning architectures consist of multiple layers that extract relevant features from high-dimensional input data. While these architectures usually end with fully connected layers to determine the output, the topology of the network at the beginning, often differs.

In this work, the focus lies on CNN and GRU architectures, which are combined to extract features from image sequences. CNNs are an advanced version of feed-forward neural networks for image processing that significantly reduce the number of parameters that needs to be determined during training, while maintaining the high predictive capabilities of the model.

2.2. *Convolutional neural network (CNN)*

CNNs can not only be used for image data, but they bring certain advantages to these applications, such as translation invariance through weight sharing, and local connectivity that takes the spatial structure of images into account. For some other applications, where spatial relations are important, these model assumptions of CNNs may also be applicable.

A simple CNN usually consists of three types of layers, which are stacked to create a deep neural network model. These layers are usually defined as pooling layer, fully connected layer, and convolutional layer. In the convolutional layer, small patches (filter) convolve over the input array, which in the first convolutional layer is the original image.

The coefficients of each filter kernel defined in a certain convolutional layer are determined during training process. The output of a convolutional layer can be denoted as follows [43]:

$$\boldsymbol{X}_d^l = \boldsymbol{f}(\sum\nolimits_{i \in M_d} \boldsymbol{X}_i^{l-1} \times \boldsymbol{K}_{id}^l + \boldsymbol{b}_d^l). \tag{2}$$

Where $\boldsymbol{X}_d^l$ is the dth output feature map (image) of the lth convolutional layer. On the right side, the ith output feature map $\boldsymbol{X}_i^{l-1}$of the previous layer $l-1$ is convolved with the idth kernel $\boldsymbol{K}$ of the current layer. $\boldsymbol{b}_d^l$ denotes the offset (bias), and M_d represent the input feature maps while $\boldsymbol{f}$ represents the activation function.

Convolutional layer is frequently followed by a pooling layer to reduce the input dimensions for the following layers by down-sampling feature maps from the previous layer. Typical types of pooling layers are max pooling and average pooling. The output x_d^l is stated by the following equation:

$$\boldsymbol{X}_d^l = \boldsymbol{f}(\delta_d^l \text{ subsample}(\boldsymbol{X}_d^{l-1}) + \boldsymbol{b}_d^l \tag{3}$$

Where l is the number of the pooling layer, $\boldsymbol{f}$ can be an activation function, δ_d^l denotes the resample factor and $\text{subsample}(.)$ represents the down-sampling function (e.g., mean or max pooling), and $\boldsymbol{b}_d^l$ is the bias (offset). Pooling, especially max pooling, is a convolution-based operation that is applied to reduce overlapping in feature maps and can help to avoid over fitting and may lead to a more generalized model [20].

2.3. *Recurrent neural networks and gated recurrent units (GRU)*

In this work, CNNs are utilized to automatically extract relevant characteristics from raw camera images. It is also possible to extract spatio-temporal information from video streams using 3D-CNNs, to extract patterns from temporal changes between adjacent frames. For example, 3D-CNNs are often used to recognize gestures or emotion in videos [33,34,44]. However, compared with approaches that combine CNN with RNN structures such as long-short term memory (LSTM) or gated recurrent units (GRU), 3D-CNN has a disadvantage that derives from its high computational complexity and excessive memory consumption, which can be a major burden for several applications that require high inference rates, especially on embedded devices [45]. Additionally, RNN architecture can be used to extract long-term temporal characteristics, whereas 3D-CNN are mostly used for the extraction of short-term temporal pattern [46]. Therefore, the combination of CNN and LSTM has been used recently for action recognition in video data that is still a challenging problem in computer vision [32,47]. LSTMs have become especially popular due to high performances achieved in domains such as natural language processing, but recent findings suggest that GRU architectures offer very comparable accuracies compared to LSTM with lower computational costs. [36,48].

GRUs belong to the group of gated RNNs, one of the most effective neural networks to approximate complex temporal dynamics. As a unique implementation of the RNN architecture, GRUs use gating mechanisms to manage the exchange of information within the neural network. GRUs were proposed by Cho et al. [49] in 2014 as an alternative architecture to the commonly used long short term memory (LSTM), which was proposed in 1997 [50]. The GRU is a slightly more simplified variation of the LSTM, as it has fewer parameters and thus may train faster and needs less data to generalize. Compared to LSTM, the entire memory is exposed to the network, while for LSTMs the exposure to other units is controlled by the output gate. Additionally, GRU can control the information flow from the previous activation, whereas LSTM is not able to manage this information flow [48]. Potentially lower calculation costs and the data-efficient structure are the reason why GRU is used for this work. The main advantage is that gated units in RNNs can store information in their units that is accessible in a later time step. The decision when to store, read or erase information is learned from the data. A GRU with unit u in layer l can be described as follows [51]:

$$\tilde{h}^t_{l,u} = g_1(\boldsymbol{w}_{l,u}\,\mathbf{x}^t + \boldsymbol{u}_{l,u}\boldsymbol{h}_l^{t-1} + b^l_{d,u}) \tag{4}$$

$$\beth^t_{l,u} = g_2(\boldsymbol{m}_{l,u}\,\mathbf{x}^t + \boldsymbol{o}_{l,u}\boldsymbol{h}_l^{t-1} + c^l_{d,u}) \tag{5}$$

$$\mathrm{h}^t_{l,u} = \beth^t_{l,u}\tilde{h}^t_{l,u} + \left(1 - \beth^t_{l,u}\right)h^{t-1}_{l,u} \tag{6}$$

$$\mathbf{h}^t_l = \left[h^{t-1}_{l,1}, \dots, h^{t-1}_{l,n\_unit}\right] \tag{7}$$

$$y^t_l = g_3(\boldsymbol{V}_{l,}\mathbf{h}^t_l + a^l_{d,u}) \tag{8}$$

Where the parameter vectors $\boldsymbol{w}_{l,u}\,\mathbf{x}^t$, $\boldsymbol{u}_{l,u}$, $\boldsymbol{m}_{l,u}$, $\boldsymbol{o}_{l,u}$ and $\boldsymbol{V}_{l,}$ as well as the parameter $b^l_{d,u}$, $c^l_{d,u}$, $a^l_{d,u}$ are determined during the training via backpropagation through time. g_1 represents the tanh activation function and g_2 is implemented as sigmoid function. If the gate value $\beth^t_{l,u}$ is close to zero, the GRU keeps the state values h^{t-1}_l ,but saves a new state $\tilde{h}^t_{l,u}$ if the gate value is close to 1. The input of the GRU is a feature vector $\mathbf{x}^t$ at time step t and a vector $\boldsymbol{h}^{t-1}_l$ that contains state values from all unit in the previous layer. g_3 is an activation function and is represented in this work by the sigmoid function.

In our architecture, the feature vector extracted by the CNN is consecutively fed into the RNN layer, which is represented by a GRU. The overall CNN-GRU architecture is shown in **Error! Reference source not found.**. For each measurement, the network takes a sequence of last $n_{sequence}$ consecutive weld images as input. Instead of using only the most recent image, the network is able to use information from the last $n_{sequence}$ images to predict the local weld quality. The image sequence represents the input of the first convolution layer, where convolution kernels with a size of 2×2 are applied on the input images. Based on Eq. 2, this results in a specific number of feature maps defined by the hyperparameter $conv\_1\_depth$. A second convolution layer uses the previously calculated feature maps as input and convolves a 3×3 kernel to compute the second layer feature maps with the help of the activation function $(Activation)$, number of feature maps $conv\_2\_deph$ and Eq. 2. The results are transmitted to the pooling layer that applies maximum pooling on each feature map, where a kernel of size 2×2 moves with a step size of 2 in both directions (Eq. 3).

The GRU network is implemented at the end of the convolutional stack of the network. The flattened feature maps (i.e., 9 x 2352 matrix) of the nine images are used as input for the GRU layer that consists of a specific number of units (GRU_units) that use tanh activation function. Based on equation (8), the GRU layer combines the feature vectors of a sequence of $n_{sequence}$ consecutive weld images to obtain a spatio-temporal feature representation. The last fully connected layer represents a hidden layer that consists of a specific number $(Dense\_units)$ of nodes and uses the activation function $(Activation)$. The softmax function is selected as the activation function of the last output layer. Additionally, a reference CNN was trained based on a modified architecture compared to **Error! Reference source not found.**, that uses a single image as input and has no GRU layer.

For both architectures, hyperparameters such as depth of each convolutional layer $(conv\_1\_depth)$ and $(conv\_2\_depth)$ as well as the activation function $(Activation)$, the number of units of GRU layer (GRU_units), the number of units in the fully connected layer $(Dense\_units)$, and the length of the input image sequence $n_{sequence}$ were determined via grid search on the basis of the values provided in **Error! Reference source not found.**. For each training process, Nesterov-accelerated Adaptive Moment Estimation (Nadam) optimizer was used to minimize the categorical cross-entropy loss function within 100 training epochs. Both architectures were implemented using TensorFlow 2.0 and Python 3.6.

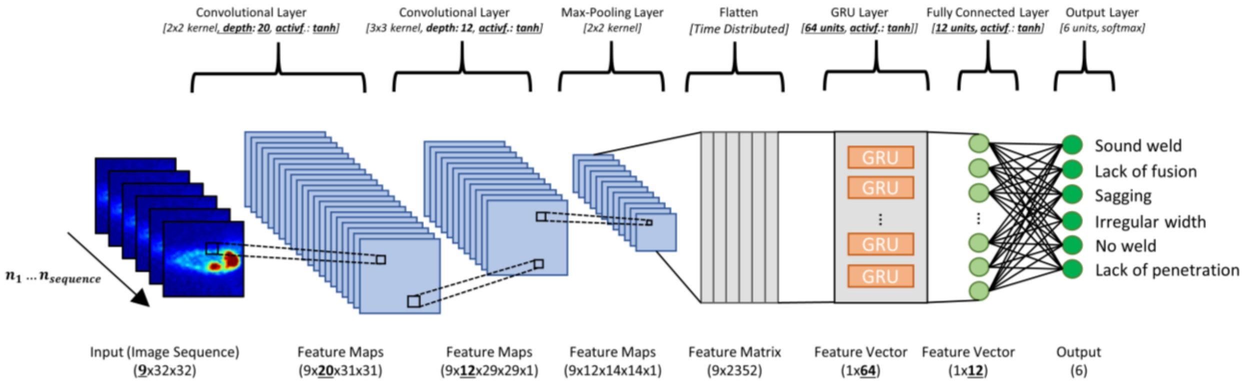

Figure 1. Proposed deep neural network architecture based on convolutional layers and gated recurrent units (GRU) for image sequence classification.

3. Experiment Setup and data preprocessing

3.1. Multi-camera welding setup

In order to detect changes in process conditions and quantify process imperfections, online process monitoring based on two cameras, as shown in **Error! Reference source not found.**, was applied. A CMOS-based camera (NIR) was used to visualize the keyhole and its surrounding area during the welding process. To monitor the weld pool in real-time, a PbSe-sensor (MWIR) was engaged, since the maximum of temperature radiation occurs according to Eq. 9 within the wavelength range of the sensor's sensitivity. The relation between specific temperatures and its wavelength of maximum thermal radiance can be expressed by the following equation according to Wien's displacement law [52]:

$$\lambda_{max} = \frac{2897.8\,\mu m \times K}{T_{melt}} \tag{9}$$

Substituting $T_{melt}$by a value of 1737 K, which represents the melting point of low carbon steel (FE P05) used for these experiments, leads to the wavelength of maximum thermal radiance at $\lambda_{\text{max}} = 1634$nm. In front of the camera sensor, narrow bandpass filters reduce the effect of chromatic aberration on the measurement signal. To meet the λ_{max} calculated above, the infrared camera uses a filter that provides a bandwidth of 82 nm at a central wavelength of 1690 nm as shown in Table 1. Both cameras start capturing image data when triggered by a signal from the robot control system. However, the data acquisition rates of the cameras used for this experiment differ. Considering the MWIR-camera sample rate of 500 Hz, each frame of the NIR-camera (100 Hz) is multiplied by five to avoid down sampling of the 500 Hz signal and to synchronize the data streams.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202105.0272.v1

Table 1. Description of the sensors and optical components used for the welding experiments

Type of camera	Sensor material / Sensitivity range	Resolution	Acquisition rate	Field of view	Bandpass filter [CWL / FWHM]
Photonfocus D1312IE-160-CL (NIR)	Si / 0.4-0.9 μm	1312x1080	100 Hz	11.6 x 5 mm²	840 nm / 40 nm
NIT Tachyon μCore 1024 (MWIR)	PbSe / 1-5 μm	32x32	500 Hz	9x9 mm²	1690 nm / 82 nm

Experiments have been conducted by applying different welding parameters using a high-power disk laser at a focus diameter of 0.6 mm and argon as shielding gas. The experiment was performed with galvanized low-carbon steel in overlapping configuration. The geometric dimensions can be obtained from **Error! Reference source not found.**. A welding configuration, which consisted of three galvanized steel sheets (FE P05) of different thickness, was considered for the experiment. For some welding trials, a modified middle sheet was used to provoke lack of fusion in certain areas due to a larger gap size as shown in **Error! Reference source not found.**-b. To allow outgassing of vaporized zinc during welding, a gap of 0.15 mm was established between all welding sheets.

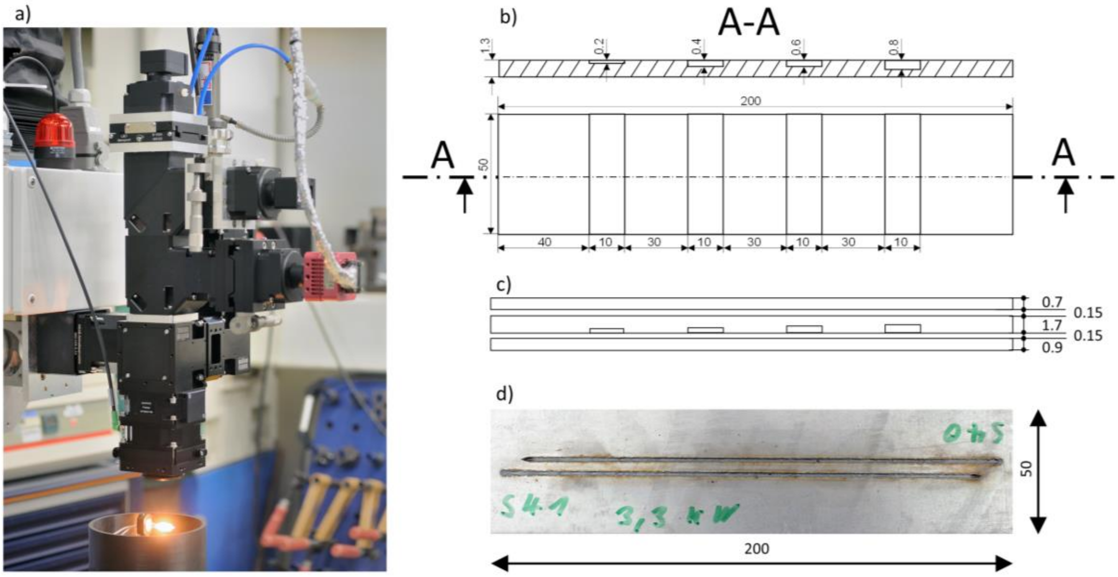

Figure 2. a) Photography of the welding optics with coaxially integrated cameras; b) Drawing of welding sheets with different slot sizes (middle sheet); c) Side view of the sheet configuration used during the welding experiments; d) Photography (top view) of two welding trails (P=3.3KW, v=50mm/s, ds=0.6mm, Argon shield gas flow=60l/min).

3.2. Feature extraction for in-situ weld image data

This chapter describes the features being extracted from the MWIR and NIR image data that were recorded during welding processes. As stated above, the recorded video data of the welding processes contain spatio-temporal information regarding the optical emission of the weld pool and the keyhole. While the proposed deep learning approach extracts relevant features directly from the raw input data, conventional classification algorithms investigated in this work require the extraction of handcrafted features from the original data as input to work properly. Overall, 172 unique features are extracted from the two process image types shown in **Error! Reference source not found.** to reduce the amount of data to be processed and to counteract the effect of over-fitting when using the raw images as input. The process of feature extraction is based on the following image processing steps:

- Binarize image based on the target object threshold
  (keyhole threshold > weld pool threshold)
- Detect contour (connected boundary line of an object) using the algorithm of Suzuki et al. [53] and select largest contour from all contours found in image
- Calculate contour properties such as centroids, and other image moments (**Error! Reference source not found.**)
- Fit an ellipse to the found contour
- Obtain geometrical parameters of the ellipse (**Error! Reference source not found.**)
- Calculate additional features such as statistical and
  sequence-based features (
-
- Table **4**)

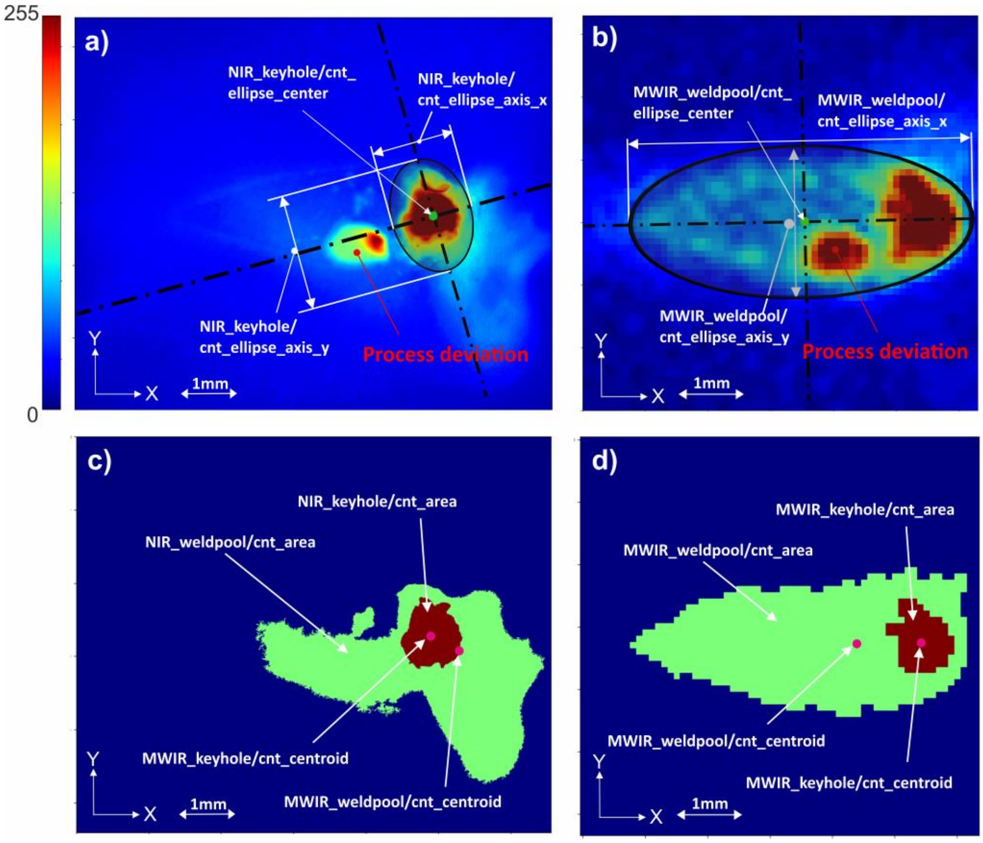

Figure 3. (a) & b): Original image and geometrical features extracted from keyhole and weld pool regions. (c) & d): Detected keyhole and weld pool contours (filled) based on two-step binarization of the original images.

The extracted contour was fitted as an ellipse with its principal axes to obtain geometrical parameters such as length and width of the keyhole and weld pool area. An example of the ellipse fitting can be seen in **Error! Reference source not found.** - a) & b). The calculation of image moments based on the extracted contour object yields additional contour properties such as area, geometric center, contour orientation and information on symmetry [54]. The calculation of moments of order p and q, of the gray value-function $I(x,y)$ for discrete images can be approximated by [55]:

$$m_{pq} = \sum_{x}\sum_{y} x^p y^q I(x,y)\,\Delta A \tag{10}$$

Where ΔA describes the area of one pixel. The zero-order moment, m_{00} represents the area of an object. For binary images, these values are proportional to the objects center coordinates. By dividing first-order moments by the zero-order components as shown in **Error! Reference source not found.**, the result can be interpreted as center of gravity of the contour. A visual explanation of some extracted geometrical features is provided in **Error! Reference source not found.**.

Taking into account two different image types (i.e., NIR and MWIR image data), 86 features are calculated for every ith image and for each image type T. Equation 11 shows the aggregated feature list F_i^T which consists of several feature subgroups as stated in **Error! Reference source not found.**. Geometrical features G_i^T based on the extracted keyhole and weld pool contours are defined as one feature subgroup.

$$F_i^T = G_i^T + IS_i^T + TS_i^T + WP_i^T + KH_i^T \tag{11}$$

Additionally, features related to overall images statistics such as *mean, minimum, maximum, variance, median, skewness and kurtosis* define the second subgroup IS_i^T. Furthermore, features based on the statistics of pixels within the keyhole region KH_i^T or the weld pool area WP_i^T are also defined as feature subgroups. Also, features are extracted from the time domain of the welding video data to form the feature subset TS_i^T. For that, statistics are calculated according to

Table **4**, based on the weld pool area of the nine most recent consecutive images, including the current image for each time step. If no image is available for a particular position in the sequence, the values are subsequently filled with the previous value. The weld pool area was chosen as time series reference feature since weld pool features appear to be highly relevant to the predictive power according to **Error! Reference source not found.**. **Error! Reference source not found.** and

Table **4** provides a detailed explanation of the individual features.

Table 2. Description of feature sub-groups used for classical machine learning methods and feature importance evaluation

Feature sub-group (short name)	Expression	Description
Geometrical features (*geometrical*)	G_i^T	Only geometrical features according to **Error! Reference source not found.** based on the weld pool and keyhole region
Overall image statistics (*image stats*)	IS_i^T	Overall image statistics according to Table **4**
Times series statistics (*timeseries stats*)	TS_i^T	Time series statistics according to Table **4** based on weld pool area
Weld pool features (*weld pool*)	WP_i^T	Geometrical and statistical features according to **Error! Reference source not found.** and Table **4** derived from the weld pool region
Keyhole features (*keyhole*)	KH_i^T	Geometrical and statistical features according to **Error! Reference source not found.** and Table **4** derived from the keyhole region

To improve the classification performance and robustness of the trained models, feature normalization was applied for both handcrafted features and raw image data. The following equation normalizes the features to a value between zero and one:

$$x_{norm} = \frac{(x - x_{min})}{x_{max} - x_{min}} \tag{12}$$

Table 3. Features based on shape descriptors and image moments (geometrical features) for a given keyhole or weld pool contour.

Feature name	Feature expression	Feature description
cnt_area	$m_{00} = \sum_x \sum_y I(x,y)\,\Delta A$	0th order moment which represents the area
cnt_centroid_*x*/*y*	$\bar{x} = \frac{m_{10}}{m_{00}};\ \bar{y} = \frac{m_{01}}{m_{00}}$	1th order moments: Center of gravity (COG)
cnt_2nd_order_mom[M*xx*|M00]	$x_2 = \frac{m_{20}}{m_{00}};\ y_2 = \frac{m_{02}}{m_{00}}$	2nd order moments: distribution of contour pixel around COG normalized by m_{00}
cnt_3nd_order_mom[M*xx*|M00]	$x_3 = \frac{m_{30}}{m_{00}};\ x_3 = \frac{m_{03}}{m_{00}}$	3rd order image moments of the given contour normalized by m_{00}
cnt_ellipse_angle *(Ellipse rotation angle α)*		Calculates the ellipse that fits (in a least-squares sense) the given contour best of all
cnt_ellipse_center_*x*/*y* *(x / y coordinate of the center)*	$\frac{(x\ \cos\alpha + y\ \sin\alpha)^2}{a^2} + \frac{(x\ \sin\alpha - y\ \cos\alpha)^2}{b^2} = 1$	The algebraic distance algorithm is used [56]
cnt_ellipse_axis_*x*/*y* *(major semi-axis a/b)*		Algorithm returns 5 ellipse parameters
cnt_equi_diameter	$d = \sqrt{\frac{4 \cdot m_{00}}{\pi}}$	Calculates the diameter of a circle based on the contour area
cnt_aspect_ratio	$Aspect\ ratio = \frac{Width}{Height}$	Defines bounding rectangle of the contour in terms of height and width
cnt_extent	$Extend = \frac{m_{00}}{BR - Area}$	Extent is defined as contour area divided by the area of the enclosing rectangle
cnt_solidity	$Sol = \frac{m_{00}}{Convex\ Hull\ Area}$	Ratio of contour area to the area of the convex hull.

Table 4. Image features based on statistical characteristics [37]

Feature name	Feature expression	Feature description
Prefix[1]_mean	$\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)$	Mean of the data $x_{1,\dots,n}$ depending on prefix
Prefix[1]_variance	$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$	Variance of the data $x_{1,\dots,n}$ depending on prefix
Prefix[1]_skewness	$Skew = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{(x_i - \bar{x})}{\sigma}\right]^3$	Skewness of the data $x_{1,\dots,n}$ depending on prefix
Prefix[1]_kurtosis	$Kurt = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{(x_i - \bar{x})}{\sigma}\right]^4$	Kurtosis of the data $x_{1,\dots,n}$ depending on prefix

[1] Prefix can be “cnt” for pixel intensities within the extracted contour of the keyhole or weld pool, or “axis_x/y” for pixel intensities along the keyhole or weld pool ellipse axis, “ts-area” for nine consecutive weld pool areas (time domain) or no prefix for overall statistics of the given image data.

3.3. Welding Defects and Data Preparation

doi:10.20944/preprints202105.0272.v1

In a further step, several welding trials based on zinc-coated steel sheets were manually characterized by human experts in terms of quality according to international standards (i.e., EN ISO 13919-1 / EN ISO 6520-1) [57,58]. **Error! Reference source not found.** shows examples for MWIR images of different weld quality states investigated in this work. It is also shown that the amount of labeled data available for supervised learning differs greatly between defect classes. Naturally, labels for images showing a satisfactory weld situation are abundant, while image data related to small defects within the weld are rare.

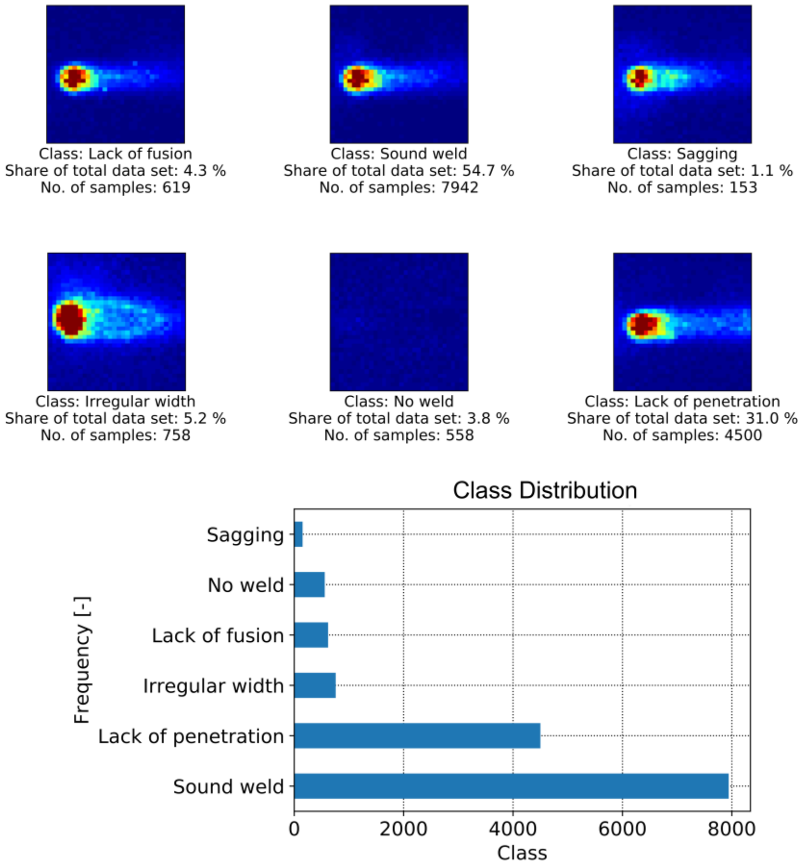

Figure 4. Example for MWIR image data and sample distribution of different quality states based on 13 weld trials (14,530 samples) that form the welding data set.

Examples of different weld defects such as lack of fusion, which often appears as a good weld in the top view, while the cross-sectional view shows a missing connection between the two sheets, are shown in **Error! Reference source not found.**. It can be obtained, that sagging, or an irregular weld width can easily be recognized from the top view photography. However, additional information is required to distinguish the classes of sound weld from lack of fusion and lack of penetration. To generate annotations for supervised machine learning, the image data was compared with the weld seam photography (top/bottom view) and the associated metallographic characterization (cross-sectional view) by matching both data sets via process start and end points. Only image data for which the quality of the weld seam could be reliably determined were annotated accordingly.

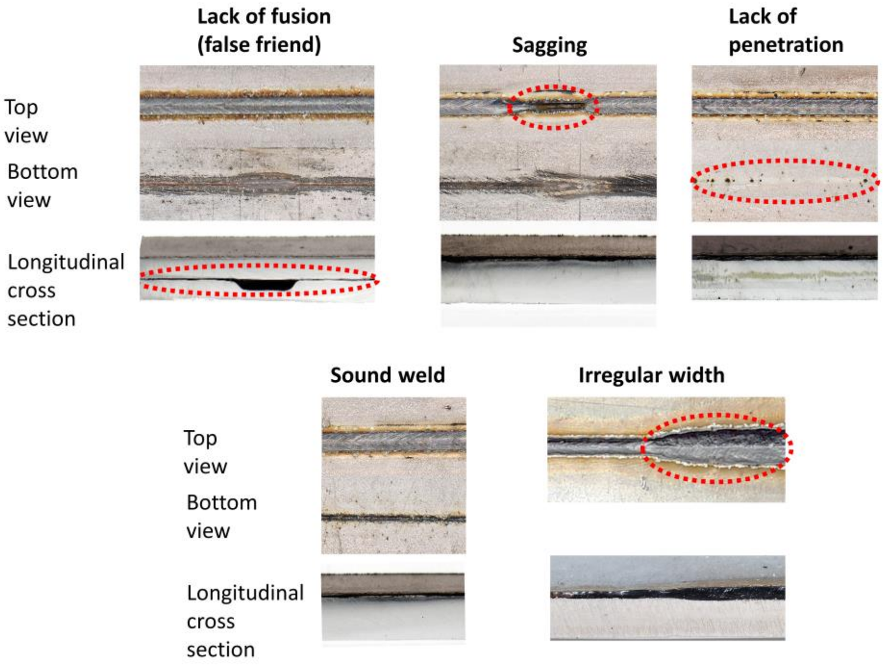

Figure 5. Photographs from different perspectives of the welding defects investigated in this study.

Overall, 14,530 images were manually annotated based on 13 weld trials. To form a temporal data set for the CNN-GRU architecture, nine consecutive images and the associated quality labels are taken in the original temporal order. The last quality label of the image stack is used as a label for a new temporal sample to build a new data set. After moving on from one image in the original data set, the next nine images and the corresponding label are taken and then added to the new data set. In case not all nine images are available, the missing images are filled with the last available image. Finally, the new data set contains as many samples as the original one, but each sample consists of nine images instead of one. To assess the impact of the number of cross-validation folds and thus the size of the training and test data sets, **Error! Reference source not found.** shows the accuracies and F1-Scores for different dataset splits. Each nested cross-validation in this work was performed five times, with a different seed value used in each iteration to randomly split into training and test data. The results suggest that influence of the data split is rather smaller as the scores differ slightly. This can be confirmed by a one-way ANOVA (analysis of variance) analysis using the mean scores for each iteration of each data split. For accuracy and F1-Score, ANOVA yields a p-value of 0.15 and 0.18, respectively, which is above the 0.05 threshold for significant differences. Thus, for the welding data set, it can be assumed that the differences concerning the data splits are randomly generated. Therefore, a common value of four folds is subsequently used for cross-validation.

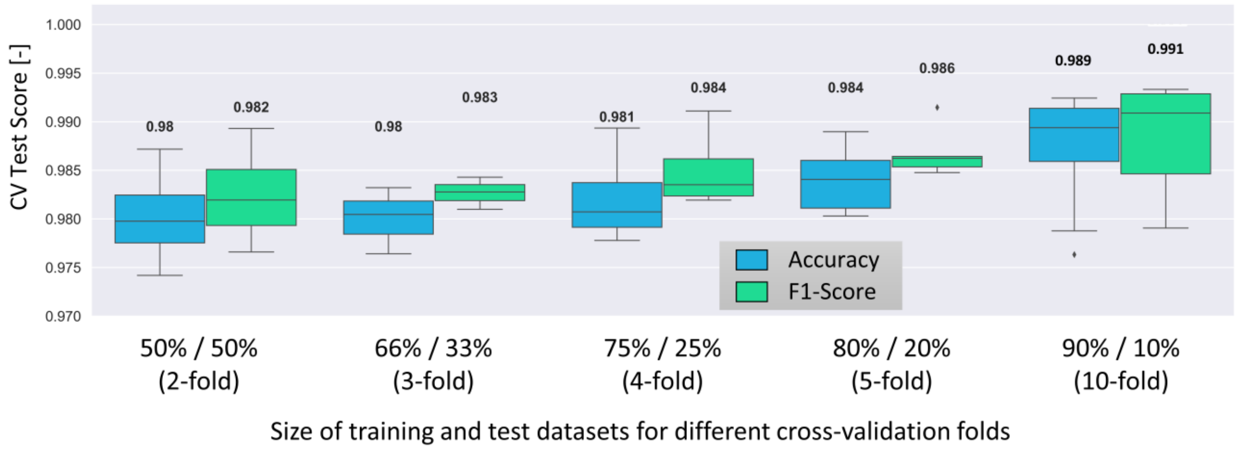

Figure 6. Different training and test splits and the corresponding test set cross-validation-scores for five random iterations. The CNN-model was used for the assessment.

In this work, deep learning models utilize data augmentation to artificially increase the data set to 72,650 images and image sequences. Some weld trials were performed in different directions compared to the sensor alignment (see **Error! Reference source not found.** - Irregular width). To learn features that are independent of the welding direction (or the sensor alignment), image augmentation is performed for all images and image sequences. Mirroring and rotation were chosen because they allow the convolutional structures to learn rotational and directionally invariant features, which leads to a more generalized model [41]. Deep learning methods typically require more data since they come with an increased number of parameters to be trained compared to conventional methods [59]. For this welding data set, experiments have shown that with an increased amount of training data, an increase in performance can be achieved. Data augmentation was also used for image sequences. In this case, all images in the sequence were coherently augmented by using the same method (i.e., rotation, mirroring) for each image in the stack. Data augmentation was not applied to the classical methods, because most of the extracted features do not vary with image mirroring or rotation. For better comprehension, the data processing and evaluation steps utilized in this work are depicted in **Error! Reference source not found.**.

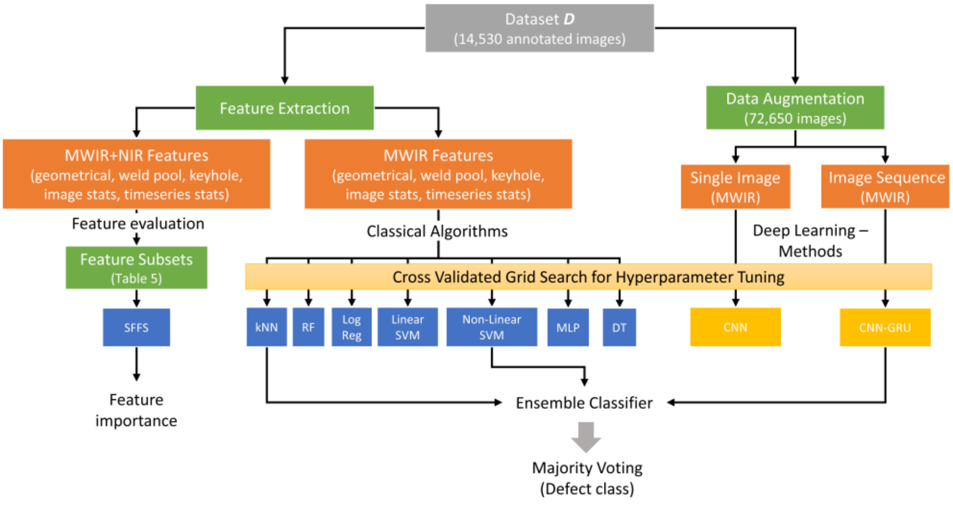
Dataset *D* (14,530 annotated images)
Feature Extraction
Data Augmentation (72,650 images)
MWIR+NIR Features (geometrical, weld pool, keyhole, image stats, timeseries stats)
MWIR Features (geometrical, weld pool, keyhole, image stats, timeseries stats)
Single Image (MWIR)
Image Sequence (MWIR)
Feature evaluation
Feature Subsets (Table 5)
Classical Algorithms
Deep Learning – Methods
Cross Validated Grid Search for Hyperparameter Tuning
SFFS
kNN
RF
Log Reg
Linear SVM
Non-Linear SVM
MLP
DT
CNN
CNN-GRU
Feature importance
Ensemble Classifier
Majority Voting (Defect class)

Figure 7. Schematic overview of the evaluation process established in this work.

4. Results and Discussion

The next section presents the feature evaluation, the results of the comparison among the classification algorithms and the final performance evaluation based on complete and unseen weld trials. Various metrics can be used for assessing the performance of classification models. Accuracy, for example, has the advantage of being simple to interpret as it represents the ratio of correctly classified samples to the number of total samples. However, accuracy is not considered a robust measure when dealing with unbalanced classes, which is the case for the weld data set. Therefore, the F1-Score is introduced as main metric to measure multiclass classification performance on the unevenly distributed weld data set [40]. On the basis of the definition of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) detections, accuracy and F1-Score are defined as follows [37]:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{16}$$

Accuracy score usually is utilized when the true positives and true negatives matter more, whereas F1 score typically applies when the false negative and false positive predictions are more important. In this study, accuracy and F1-Score are reported to describe the classification performance of a machine learning model, however the F1-Score is considered for final evaluation.

4.1. Assessment of Feature Importance

The importance of the features was determined using sequential forward floating selection (SFFS), which represents an extension of the simpler SFS algorithm. SFS starts with an empty feature subset and trains a classification model for each available feature based on a defined algorithm, which in this case was a SVM classifier, as it provides small training and inference times (see **Error! Reference source not found.**). The feature that provides the highest balanced accuracy score is included as the most important feature in the new subset. Afterwards, at every ith step, classifiers are trained for each combination of the $(i-1)th$ important feature and the remaining features to determine the ith most important feature. A comparative alternative would be sequential backward selection (SBS). The algorithm begins with the full feature set and removes features based on their effect on classification accuracy until the specified number of features in the new subset is reached. However, this algorithm is not suitable for large feature subsets because it requires more processing time than SFS. The floating version of SFS (SFFS) has an extra step that allows the removal of features that were previously included (or excluded), resulting in an increased search space to find the optimal feature subset. It has been shown that SFFS enables the selection of appropriate features with high efficiency, especially compared to methods such as "*Min-Max search*", "*branch and bound*" or SFS, which is why it is used in this work [60].

Based on the SFFS algorithm, **Error! Reference source not found.** and **Error! Reference source not found.** show the cumulated accuracies for 20 out of 86 features based on weld pool and keyhole characteristics as well as overall image and time series statistics extracted from MWIR and NIR welding images.

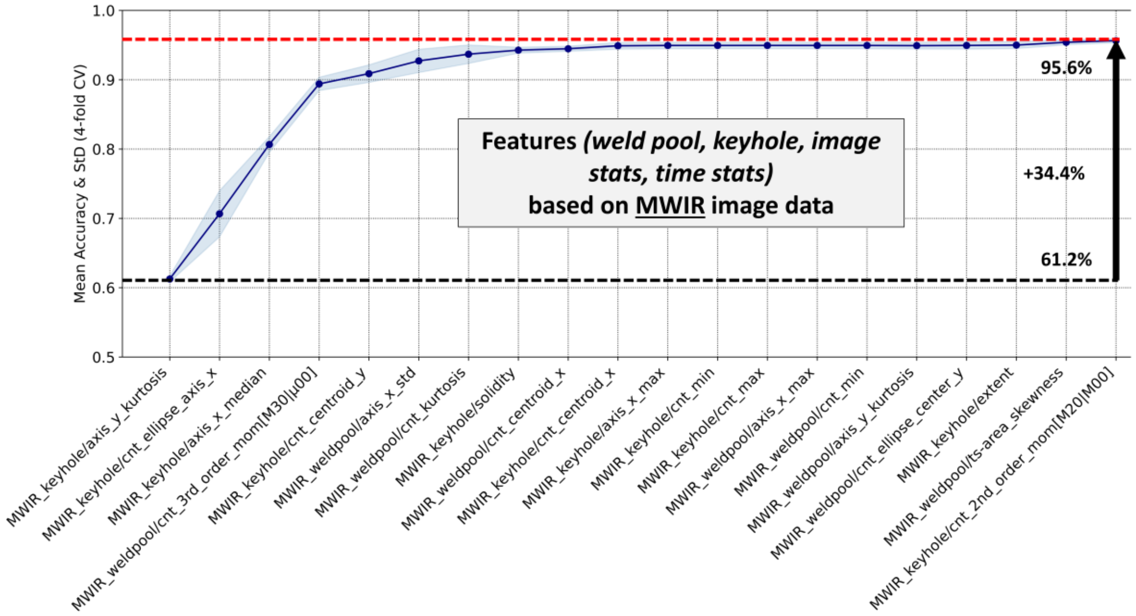

Figure 8. 20 most important features based on 80 geometrical and statistical characteristics of the weld pool, keyhole and overall image statistics extracted from MWIR images (starting with the left).

SFFS only finds the optimal subset of n features that leads to the highest performance score but does provide information about the individual importance of these features. Thus, SVMs are used to estimate the importance ranking of individual features within the list of the most important features found via SFFS. Starting with the far-left feature, SVM models were trained consecutively by adding another feature in a further step, until 20 features were added to the final feature subset. It can be seen, that classification models based on MWIR features achieve higher accuracies with fewer features as compared to models trained with features only based on the NIR features. For both image types, NIR and MWIR, it can be observed that statistics of the pixel distribution of the weld pool and keyhole axis' (i.e., *axis_x / y_kurtosis*) and further geometrical properties such as *ellipse_axis_x/y* are most relevant for weld defect prediction.

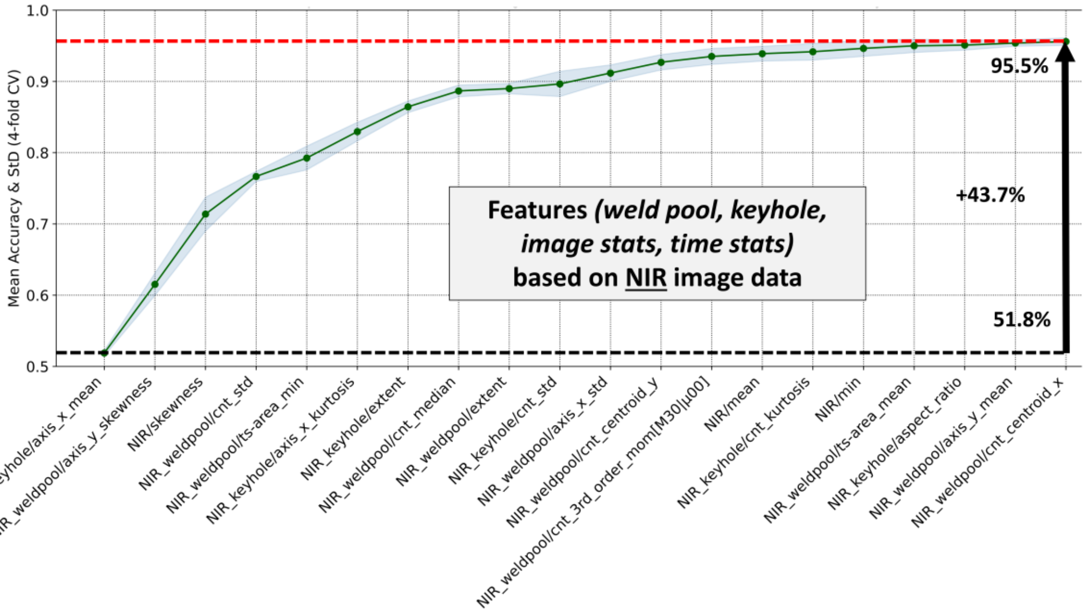

Figure 9. 20 most important features based on 80 geometrical and statistical characteristics of the weld pool, keyhole and overall image statistics extracted from NIR images (starting with the left)

Error! Reference source not found. shows the defect detection performance of several feature subsets derived from the original amount of 172 features as cross-validated F1-Score. The results show that feature subsets based on geometric features (MWIR+NIR (*geometrical*)) extracted from the weld pool and keyhole regions, can almost reach the top F1-Score of 0.978 achieved by the original feature set. Interestingly, if the prediction model is trained only on geometrical features from either MWIR or NIR images, its performance (0.928 & 0.826) is significantly lower than the performance of the combination of these feature subsets (0.970). The general performance level of feature subsets only based on overall image statistics (*image stats*) and time-series statistics (*timeseries stats*) is low compared to all other subsets. One reason for that can be found in the low dimensionality of those subset (i.e., six features). However, **Error! Reference source not found.** and **Error! Reference source not found.** also show that these features are not necessarily important, as only three time series features and two features based on image statistics appear in the lists of the twenty most important features. It should be noted that more complex time series features, such as Fourier or Wavelet coefficients, may lead to more important features. However, their further investigation exceeds the scope of this work. Meanwhile, the F1-Scores for weld pool features extracted from MWIR and MWIR plus NIR images are 0.966 and 0.969 respectively, whereas the score for weld pool features extracted from the NIR images is 0.918. This is probably caused by the low thermal signal obtained with this sensor. Although NIR image data at 840 nm wavelength provide higher spatial resolution of the keyhole area, the thermal signal of the weld pool area was hardly detected by this sensor. As explained in section **Error! Reference source not found.**, the optimal wavelength for weld pool observation is located at around 1634 nm, which is preferably observed by the MWIR camera. Additionally, if the performances of features extracted only from MWIR images are compared, weld pool features outperform keyhole feature by 3.3 %. Overall, most relevant information can be found in MWIR features which reach, according to **Error! Reference source not found.**, generally higher F1-Scores compared to NIR features. However, the highest F1-Scores are achieved by combining features from both cameras. This leads to the assumption that the NIR images with spatially higher resolution, can provide additional information of the keyhole area, to that obtained from the MWIR images. However, as the number of features used to create a classification model increases, the risk of over-fitting also increases.

Table 5. Comparison of several feature subsets with respect to their ability to predict different weld defects (without "no weld" class)

Feature subset		**Cross-validated F1-Score**					
Name	**No. of Feat.**	**Lack of fusion**	**Sound weld**	**Sagging**	**Irregular width**	**Lack of penetration**	**avg**
MWIR+NIR (*weld pool, keyhole, image stats, timeseries stats*)	172	0.983	0.998	0.913	1.0	0.999	0.978
MWIR+NIR (*geometrical*)	64	0.89	0.989	0.976	1.0	0.998	0.970
MWIR+NIR (image stats)	12	0.743	0.908	0.091	0.999	0.951	0.738
MWIR+NIR (*timeseries stats*)	12	0.701	0.867	0.000	1.0	0.914	0.694
MWIR+NIR (*weld pool*)	74	0.953	0.995	0.901	1.0	0.999	0.969
MWIR+NIR (*keyhole*)	74	0.948	0.995	0.829	1.0	0.999	0.954
MWIR (*weld pool, keyhole, image stats, timeseries stats*)	86	0.945	0.993	0.93	1.0	0.998	0.973
MWIR (*geometrical*)	32	0.834	0.96	0.864	1.0	0.98	0.928
MWIR (*image stats*)	6	0.688	0.74	0.000	1.0	0.862	0.658
MWIR (*timeseries stats*)	6	0.569	0.669	0.000	1.0	0.801	0.607
MWIR (*weld pool*)	37	0.896	0.987	0.951	1.0	0.997	0.966
MWIR (*keyhole*)	37	0.851	0.983	0.833	1.0	0.997	0.933
NIR (*weld pool, keyhole, image stats, timeseries stats*)	86	0.904	0.986	0.956	1.0	0.996	0.968
NIR (*geometrical*)	32	0.56	0.907	0.780	0.923	0.961	0.826
NIR (*image stats*)	6	0.403	0.862	0.000	0.922	0.937	0.625
NIR (*timeseries stats*)	6	0.544	0.808	0.000	0.989	0.855	0.639
NIR (*weld pool*)	37	0.787	0.941	0.902	0.993	0.971	0.918
NIR (*keyhole*)	37	0.791	0.955	0.863	0.995	0.978	0.916

4.2. Classifier Comparison Based on Grid Search Results

For a comprehensive comparison of different classification methods and algorithms, a grid search coupled with four-fold nested cross validation was performed to find optimal hyperparameters. For each conventional classification algorithm, every combination of grid values shown in **Error! Reference source not found.** was evaluated by using the complete data set of 14,530 samples and the entire MWIR feature subset. The subset was chosen because the MWIR data already scored high F1-Scores (0.973), compared to the combination of MWIR and NIR (0.978). Therefore, a feature space with fewer dimensions is chosen to prevent the phenomenon of the "*curse of dimensionality*" which may lead to over-fitting. The CNN and CNN-GRU models were trained using the augmented welding data set, consisting of 72,650 samples of raw image data and image sequences respectively (see section **Error! Reference source not found.**).

Table 6. Classification algorithms and hyperparameter values used for cross-validated (nested) grid search

Algorithm name	Hyperparameter	Grid values
Decision Tree Classifier **[DT]**	**max_depth**: Maximum depth of decision tree	[10, 20, 30, 40, 50]
	max_features: Number of unique features used to evaluate the best split	[sqrt(n_features)’, ‘log2(n_features))’]
	criterion: Estimation of the split quality	[‘gini’, ‘entropy’]
KNeighbors Classifier **[kNN]**	**metric**: Metric used to measure distance between two data points in an n-dimensional feature space	[‘minkowski’, ‘euclidean’,‘manhattan’]
	weights: Function used to weight points in each neighbourhood	[‘uniform’,‘distance’]
	n_**neighbours**: number of neighbours to evaluate	[2, 3, 4, 5, 6]
Support Vector Classifier with non-linear kernel **[SVM (non-linear)]**	**C**: regularization strength (L2 penalty) while regularization is inversely proportional to C. Used for all kernels (sigmoid, rbf, polynomial)	[0.01,0.1,1,10, 100,1000, 10000]
	kernel: type of kernel used	[‘rbf‘,‘poly‘,‘sigmoid‘]
	degree: Degree of the polynomial kernel function (poly)	[3,4,5,6]
Support Vector Classifier with linear kernel **[SVM linear]**	**C**: Regularization strength while regularization is inversely proportional to C	[0.01,0.1,1,10,100,1000, 10000]
	loss: Specifies the loss function	[‘hinge’, ‘squared_hinge’]
	penalty: Application of Lasso (L1) or Ridge (L2) regularization	[l2, l1]
Random Forest [RF]	**n_estimators**: Number of overall decision trees	[5, 10, 100, 500]
	max_features: Number of unique features used to evaluate the best split	[sqrt(n_features)’, ‘log2(n_features))’]
	criterion: Estimation of the split quality	[‘gini’, ‘entropy’]
Multi Layer Perceptron [MLP]	**learning_rate_init**: Learning rate at start that manages the weight update rate.	[0.01, 0.05, 0.1, 0.5, 1.0]
	Activation: The hidden layer’s activation function	[‘logistic, ‘relu’, ‘tanh’]
	hidden_layer_sizes: Number of nodes the hidden layer consists of	[25, 50, 100]
Logistic Regression [LogReg]	**C**: Regularization strength while regularization is inversely proportional to C	[0.01,0.1,1,10,100,1000, 10000]
	solver: Algorithm to solve the optimization problem	['liblinear', 'saga]
	penalty: Application of Lasso (L1) or Ridge (L2) regularization	[l2, l1]
Convolutional Neural Network [CNN]	**Activation**: the activation function for convolution and fully connected layer	[ReLU, tanh]
	conv_1_depth: the number of output filters in the first convolutional layer	[32,48,24]
	conv_2_depth: the number of output filters in the 2nd convolutional layer	[50,64,36]
	Dense_units: number of units in the hidden layer	[24,36,48]
Convolutional Neural Network + Gated Recurrent Units [CNN-GRU]	**Activation**: The activation function for convolution and fully connected layer	[ReLU, tanh]
	conv_1_depth: The number of output filters in the first convolutional layer	[20,32,12]
	conv_2_depth: The number of output filters in the 2nd convolutional layer	[16,8,10]
	GRU_units: Number of units in the Gated Recurrent Unit layer	[64,48,96]
	Dense_units: Number of nodes in the hidden layer	[12,10,8] [3,6,9]

nsequence: Length of the input image sequence to be classified

In **Error! Reference source not found.**, the performance and the optimal hyperparameter of all classification methods evaluated during grid search are shown. Finally, the optimal parameter sets were evaluated using five random four-fold cross-validation iterations, resulting in 20 samples (i.e., five iterations and four folds per iteration) for each classifier and score. Overall, the proposed CNN-GRU architecture achieves the highest median scores. However, conventional classification methods such as kNN and non-linear SVM, which are based on geometric and statistical features extracted from the MWIR images, are only slightly lower in terms of their median performance scores. The best performing individual models with respect to F1-Score are kNN (0.992) and CNN-GRU (0.995) classification models. The results show that the average performance level of all methods investigated, is high (>93 %) which leads to rather small differences between the individual methods.

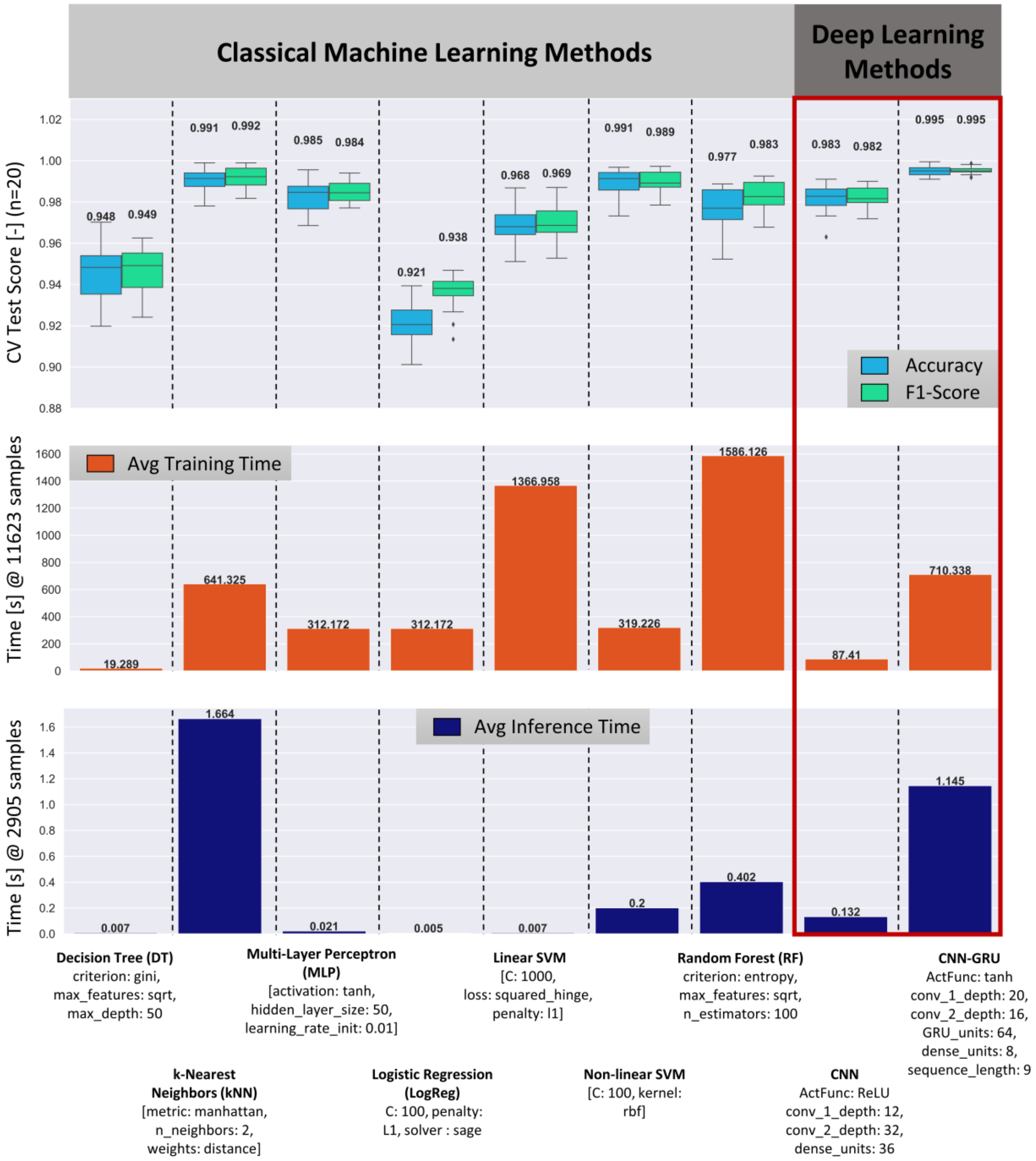

Figure 10. Performance comparison of different conventional machine learning and deep learning classification methods. Optimal hyperparameter for each classifier were found via grid search (Table 6). The median scores are displayed in the top diagram

To analyze the significance of the difference between the performances, a two-way nested ANOVA is utilized. For nested ANOVA, the type of machine learning (ML) a classifier belongs to is represented by two groups (i.e., classical ML and deep learning). The ML type is considered as level one factor, whereas the classifier type is used as nested random variable (i.e., 11 subgroups). For each classifier, the five mean scores of five random 4-fold cross validations were used as input. In a two-level nested ANOVA, one null hypothesis is that the groups have the same mean score. Based on the results depicted in **Error! Reference source not found.**, this hypothesis cannot be rejected (p>0.05), which indicates that the difference between classical and deep learning-based classifiers is negligible. The second level null hypothesis states that all non-deep learning algorithms have the same mean score and that both algorithms based on deep learning have the same mean score. Since the p-values for both scores are below the threshold of 0.05, this hypothesis is neglectable. Therefore, a statistically significant difference must be at least between two of the investigated classifiers. A more detailed post-hoc analysis regarding the significance of the classifiers' performance difference is given in section **Error! Reference source not found.**.

Table 7. Nested ANOVA for the effect of "type of machine learning" and "type of classification algorithm"

2-Way Nested ANOVA for Accuracy

Source of Variation	SS	Df	MS	F	p-value
Between groups (Type of ML)	0.003209	1	0.003209	1.0969	0.3298
Subgroups whithin groups (Algorithm)	0.0205	7	0.002925	392.6607	<0.01
Residuals	0.000261	35	<0.0001		
Total	0.0239	43			

2-Way Nested ANOVA for F1-Score

Source of Variation	SS	Df	MS	F	p-value
Between groups (Type of ML)	0.002306	1	0.002306	1.0877	0.3317
Subgroups whithin groups (Algorithm)	0.0148	7	0.002120	393.5979	<0.01
Residuals	0.000189	35	<0.0001		
Total	0.0173	43			

The algorithms can be evaluated not only according to their prediction performances, but also in terms of individual training and inference times, which are particularly important in practice for real-time measurement and quality prediction. If training and inference times are important, the kNN classifier underperforms clearly in contrast to its competitors. Considering an inference time of 1.664 seconds for 2,905 samples, the kNN classifier can only query 1,745 samples/sec, while processing time for feature extraction is not included. That is because for each classification the dissimilarities with each training vector must be computed, which leads to high computational costs. For a brute force kNN algorithm a time complexity of $O(n \times m)$ can be considered, where m is number of features per sample and n is the amount of samples used for training [39]. On the contrary, the trained CNN-GRU architecture reaches 2,537 images/sec when inference is performed on GPU. All described models and algorithms have been trained on a computer with Intel® Core™ i7-9700 CPU and Nvidia® GeForce® GTX 1080 Ti GPU.

Both inference rates are higher than the frame rate of the MWIR camera (500 Hz). The training time of the CNN-GRU model is approximately 10 % higher than the kNN training time. These timings serve only as a rough and relative estimate, as they depend heavily on the individual implementation and hardware used. It should be noted that the CNN and CNN-GRU models process raw image data or image sequences and perform feature extraction inherently, therefore the overall processing time is not expected to increase compared to traditional machine learning methods, where feature extraction requires additional processing time. Additionally, the generally high level of performance of all algorithms may be due to the conservative annotation process of the weld data. Only image data for which the quality of the weld seam could be reliably identified by the human experts were marked accordingly. Therefore, in a next step, we will evaluate the performance of these models on complete and unseen welding trials.

4.3. Experimental Evaluation

Four different welding trials were employed to assess the performance of the different classification models. On the basis of the top three models according to **Error!**

Reference source not found.Error! Reference source not found., the probability curves of each defect class, predicted by the models are shown in **Error! Reference source not found.** and **Error! Reference source not found.** for different welding trials, together with their cross-sectional and top view.

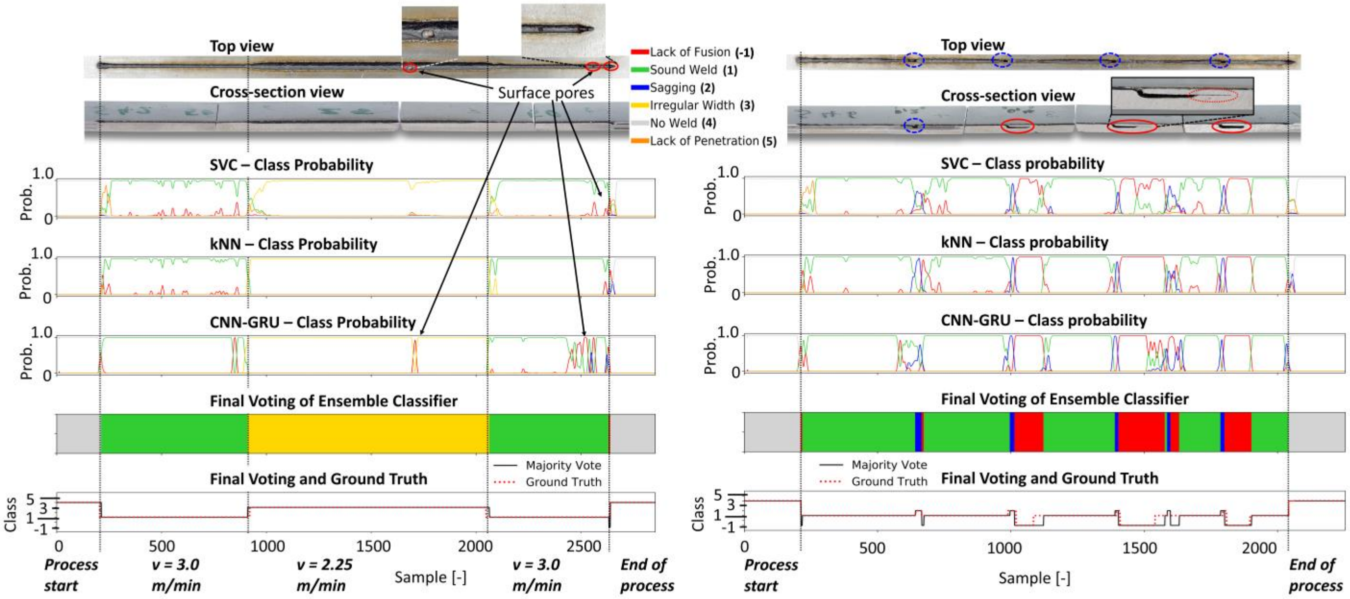

Figure 11. The metallographic characterization, the resulting ground truth data and the classification results for weld 42 (left) and weld 46 (right) based on the majority vote of SVC, kNN and CNN-GRU classifiers

Additionally, according to their F1-Score the three best performing models, namely kNN (0.939), CNN-GRU (0.938) and non-linear SVM (0.926), were combined to an ensemble classifier to increase the robustness and provide a generalized model compared to a single estimator. Based on the sum of all probabilities, the class with the highest probability was chosen for the prediction. The result for each welding trial is shown in the colored bar plot. The bottom plots show the ground truth data of each trial compared to the class predicted by majority vote. In weld 42 (**Error! Reference source not found.** – left), the welding speed was temporally reduced to 75 % of the original welding speed of 3 m/min, which leads to an increased width weld seam that was sufficiently detected by all classifiers and accordingly to the resulting majority vote. Additionally, open pores occurred during the weld at the marked positions (red circles) in the top view of weld 42. While the classifiers were not trained to detect these kinds of defects, the CNN-GRU model shows high sensitivity to these events, as it presents decreased probabilities for a good weld for these specific seam positions (red circle). On the right side of **Error! Reference source not found.**, the prediction results for weld 46 are shown. Based on the experimental setup in **Error! Reference source not found.**, weld defects were provoked by modifications in the form of several slots at the top side of the middle sheet. In the top view, seam collapses and sagging can be observed (blue circles). Lack of fusion (red circles) has occurred at three positions correctly identified by the classifiers. A short section after the first sagging defect is predicted as lack of fusion by the classifiers. However, this cannot be confirmed in the cross-sectional view of the weld.

Table 8. Parameters and sheet configuration of four welding experiments for evaluation

Welding parameters	Weld 42	Weld 46	Weld 48	Weld 216
Laser power (kW)	3.3	3.3	3.3	2.7
Beam focus offset (mm)	0	0	0	-2
Welding speed (mm/s)	50; 37.5	50	50	50

Shielding gas (L/min)	60	60	60	60
Sheet configuration	Three sheets; No slots	Three sheets; Slots point upwards	Three sheets; Slots point downwards	Two sheets; No middle sheet

In **Error! Reference source not found.**, the results of the prediction for weld 48 are shown on the left side. In this weld, slots were made on the underside of the middle sheet to induce welding defects. While the top view of the weld shows a small section where sagging occurred, the cross-sectional view shows three sections of lack of fusion defects. The latter defect type is correctly predicted in terms of their general location, but the exact position was not perfectly recognized. The "sagging" defect in the first part of the weld seam is detected by the CNN-GRU and kNN models, which leads to a sagging classification by the ensemble classifier at this location. On the right side of **Error! Reference source not found.**, the bottom view shows lack of penetration for the entire weld. The welding seam was performed with a laser beam that was positioned -2 mm out of focus. At the end of the weld, the bottom view shows an increased penetration depth. However, full penetration was never achieved during this weld. All models predict the absence of a sufficient weld depth in the first part of the weld with a high probability. In the last third of the weld, according to all classifiers, the probability for lack of penetration decreases and the models predict an increased probability for the sagging defect, which did not occur in this weld.

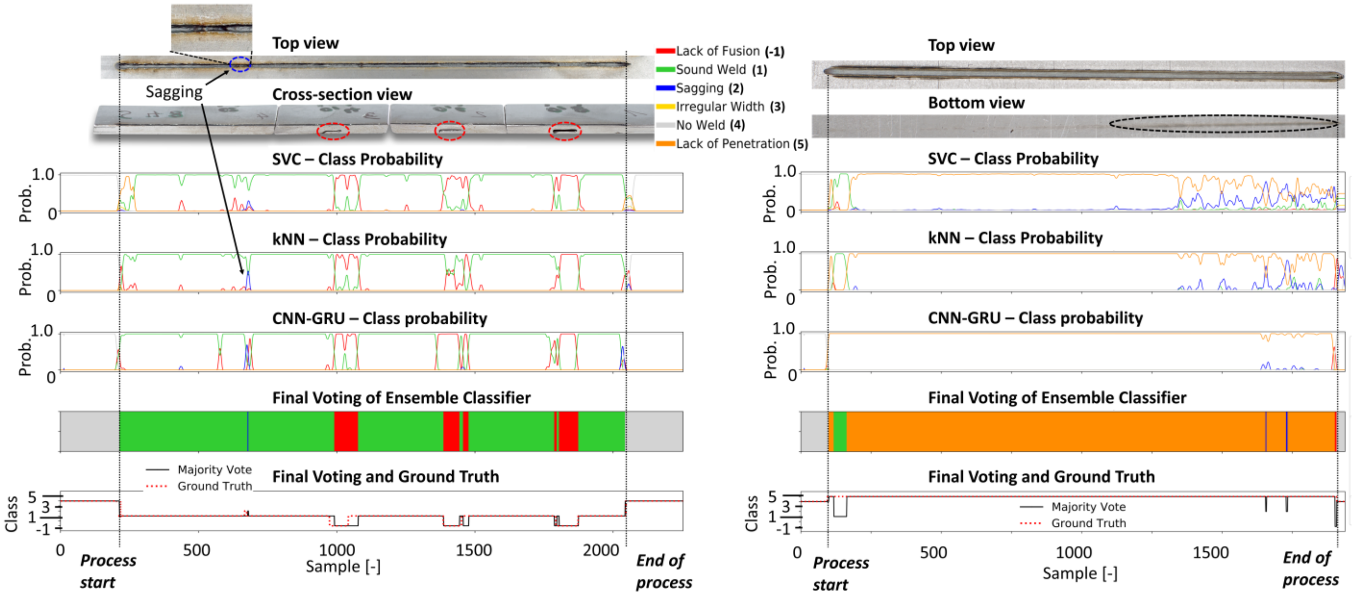

Figure 12. The metallographic characterization, the resulting ground truth data and the classification results for weld 48 (left) and weld 216 (right) based on the majority vote of SVC, kNN and CNN-GRU classifiers

The performance of all classification models used in this work and the resulting majority vote accuracy can be obtained from **Error! Reference source not found.Error! Reference source not found.**. Based on four welding trials, the proposed deep learning architecture (CNN-GRU) achieves an average F1-Score of 93.8 % and outperforms the CNN-model which uses single images. Finally, by combining the three best classifiers via majority vote, the highest average performance of 95.2 % can be achieved. It should be noted that in this evaluation all major defects are properly identified by the ensemble classifier. The inaccuracy is due to imprecise localization and dimensions of the defect predictions as well as false positive detections (false alarms) at some points of the weld.

Table 9. Classification performance for different welding trials (not within the training data set).

Method	Weld 42 (2,856 samples)	Weld 46 (2,255 samples)	Weld 48 (2,254 samples)	Weld 216 (3,140 samples)	Avg. Accuracy	Avg. F1-Score
Decision Tree	0.893	0.861	0.914	0.729	0.849	0.893
kNN*	0.977	0.885	0.921	0.94	**0.931**	**0.939**
MLP	0.962	0.882	0.916	0.831	0.898	0.924
LogReg	0.944	0.873	0.911	0.782	0.878	0.892
Linear SVM-	0.93	0.815	0.906	0.75	0.850	0.867
Non-Linear SVM*	0.958	0.892	0.917	0.888	**0.914**	**0.926**
RF	0.97	0.921	0.927	0.796	0.904	0.923
CNN	0.822	0.895	0.916	0.933	0.892	0.897
CNN-GRU *	0.95	0.88	0.919	0.972	**0.930**	**0.938**
Ensemble Classifier*	**0.985**	**0.909**	**0.946**	**0.963**	**0.951**	**0.952**

*Top-3 classification models selected to build the ensemble classifier.

To evaluate the significance of performance differences between the classification models, McNemar's statistical test continuity correction [61] that belongs to the group of Chi-squared tests was applied. In the context of machine learning models, this method is often considered when comparing the predictive accuracy of two models [62,63]. In the McNemar test, the null hypothesis (H_0) can be formulated such that both models perform equally well. Hence, the alternative hypothesis (H1) implies that there is a significant performance difference between the models. The two-tailed test will evaluate both if the accuracy of model 1 is significantly greater than that of model 2 and if the accuracy is significantly less than that of model 2. The difference is considered to be significant if the resulting p-value is smaller than the threshold of 0.05 [61,64].

The p-values of the different pairs of classifiers examined in this work are shown in **Error! Reference source not found.**. The arrowheads direct towards the classifier that performed better on average in the given welding trials (i.e., trials 142, 146, 148, 216). The figure shows that H_0 can be rejected for the most comparisons since the p-value is below 0.05. Regarding the selection of the three best models based on classification accuracy or F1-Score, as shown in **Error! Reference source not found.**, the results indicate that KNN, non-linear SVM, and CNN-GRU are statistical significantly preferable models, as their high performance is generally associated with low p-values (<0.05) compared to other classifiers. However, there is an exception as the p-value of 0.477 (>0.05) indicates that the CNN-GRU model and the KNN classifier have no significant difference in performance. From the overall perspective this result appears plausible since the metrics (i.e., accuracy and F1-Score) are very similar for both classifiers. Despite that the models behave differently for each welding trial which leads to higher differences in the metrics according to **Error! Reference source not found.** and **Error! Reference source not found.Error! Reference source not found.**. Nevertheless, CNN-GRU and KNN are still ranked higher than the other competitors with statistical significance, and thus are used together with the non-linear SVM model to build an ensemble classifier. Comparing the individual deep learning algorithms (i.e., CNN and CNN-GRU), the proposed architecture using temporal image stacks and gated recurrent units achieves significantly higher scores than the reference CNN, which uses single images as input. Furthermore, the results support the statistical significance of the superior performance of the ensemble classifier.

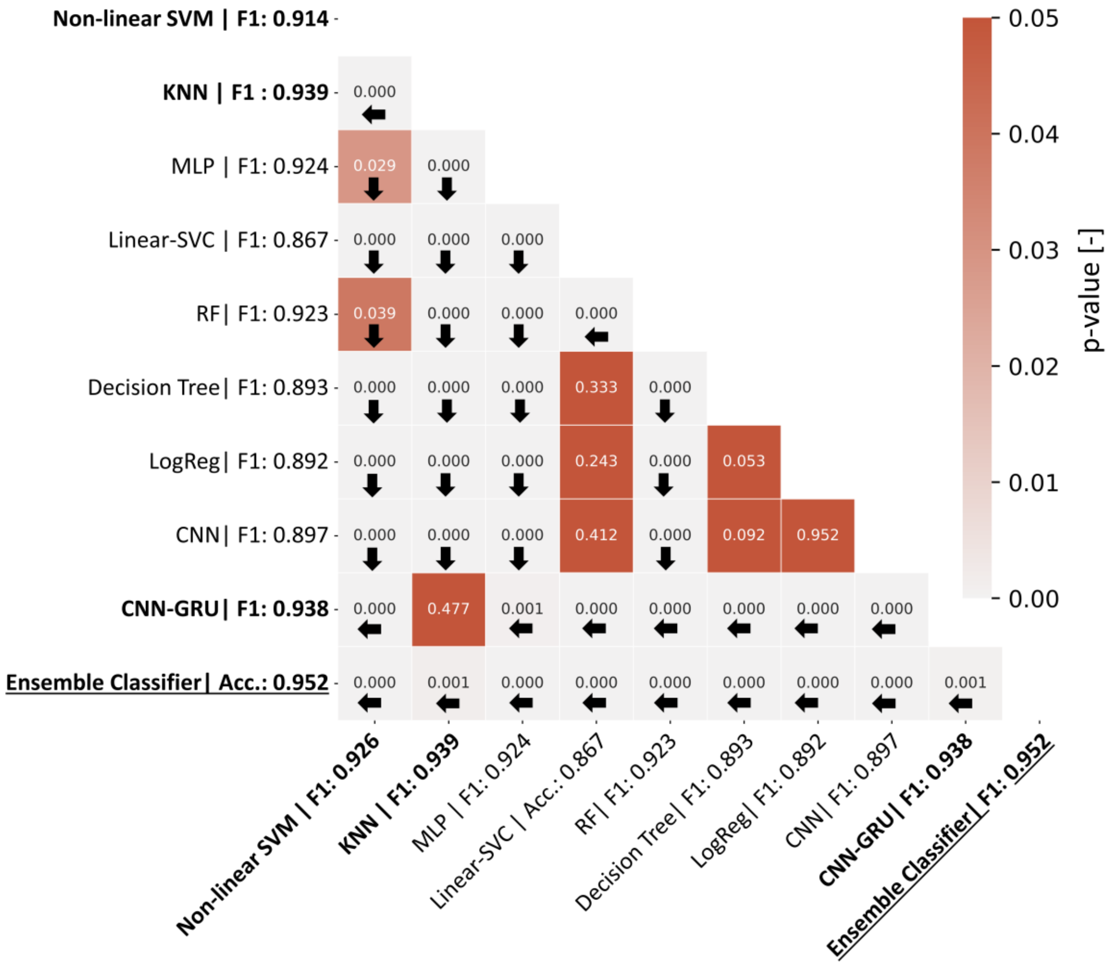

Figure 13. P-values of McNemar's test conducted for different models based on predictions of trial 142, 146, 148 and 216. The arrow direction indicates the classifier with highest F1-score for each tested pair.

The defect detection ability with respect to the different defect classes can be seen in **Error! Reference source not found.**, where normalized confusion matrices for various classification models listed in **Error! Reference source not found.Error! Reference source not found.** are depicted. In this application, sagging is the most difficult defect to detect. Due to the small number of samples available for training and the relative similarity with other classes confusion arises, especially with the classes "sound weld" and "lack of fusion" while the former is more critical. If a defect is mistaken for another defect, at least one insufficient quality has been found and failure of the final product can be avoided. In addition, the figure shows that the CNN-GRU classifier, for example, has an increased ability to detect defects such as "lack of fusion " and " lack of penetration", whereas the detection ability is reduced for the "sagging" and "sound weld" classes. The individual signs within the diagonal squares in **Error! Reference source not found.** indicate the two highest true positive rates for each defect class among the individual classifiers and the corresponding true positive rates in the resulting ensemble classifier. It is shown that the individual models can compensate their low true positive rates in certain defect classes by majority votes, resulting in an ensemble classifier with the highest defect sensitivity and the lowest false alarm rate compared to all individual classifiers.

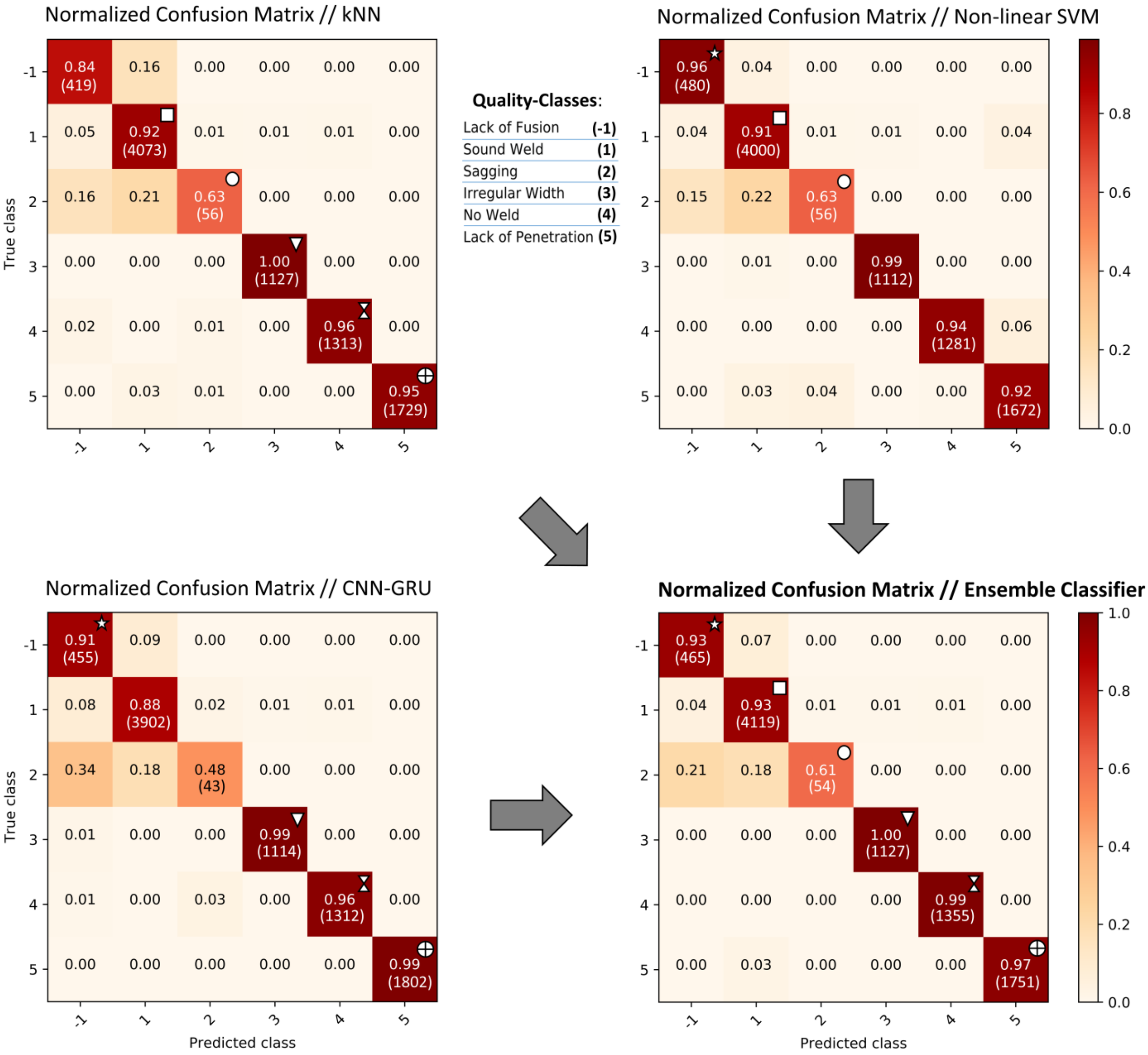

Figure 14. Confusion matrix normalized to total amount of samples per class based on data of four welding trials (weld 42, 46, 48, 216). The number in brackets indicates the absolute number of samples of correct predictions for each class

Overall, the results indicate that classical machine learning is on par with deep learning for this application. Still, deep learning brings several advantages, especially in terms of implementation times (i.e., no requirement for feature engineering), execution times, and scalability. It is assumed that the performance of classical methods of machine learning and deep learning can be further improved by increasing the amount of training data. However, deep learning can also learn to extract more refined features from larger data sets, while traditional methods may reach saturation more quickly in terms of classification performance because their ability to improve feature extraction is not inherently given.

It must be mentioned that the present work was realized with data obtained in a well-controlled laboratory setup. Although the welding head and monitoring equipment studied in this work can also be used for industrial production, the artificially induced faults may not fully reflect the situation for industrial applications. Another factor to be considered critically is the highly imbalanced data set used for training and testing. As documented in literature, highly imbalanced data sets cause heavily biased classification results [65]. This results from the fact, as shown in **Error! Reference source not found.** (class "sagging"), that classes with more labeled instances are given more importance than those with far fewer labeled instances, since the classifier's default learning objective tends to be robust to these minority classes. Therefore, classifiers trained under such a condition tend to categorize the minority classes randomly. However, it is believed that the classification performance of these minority classes can be further improved by addressing the imbalances in the dataset seen in **Error! Reference source not found.**.

5. Conclusion

On the basis of two different imaging sensors, conventional and deep learning techniques are employed to predict critical weld defects such as lack of fusion (false friends), sagging, irregular seam width and lack of penetration. Methods from the field of computer vision and descriptive statistics are used to extract informative features from the image data recorded during weld processes. An extensive study on the importance of the different features and feature subsets was carried out. It is shown that when using a small number (< 36) of features, the most relevant features can be derived from MWIR camera images, especially from the weld pool region. However, the highest detection rates were achieved by combining geometrical and statistical features extracted from both image data sources. Moreover, a deep learning architecture based on CNNs and GRUs was employed to detect weld defects exploiting their ability to extract spatio-temporal features from raw video data. Compared to the conventional classifiers, the model was able to provide indications of undefined errors such as open pores. Additionally, based on the activation maps of the CNN model, insightful information about the visual appearance of various defects in the image data were derived. In a further step, hyperparameters for deep learning methods as well as for classical machine learning algorithms were optimized during an extensive grid search. All methods were finally compared in terms of classification performance (i.e., F1-Score and accuracy), training time and inference time. The top three classification models, specifically non-linear SVM, CNN-GRU and kNN, were finally combined into an ensemble classifier that applies majority voting. Based on the evaluation on four previously unseen welding trials, our proposed architecture achieves the second highest mean F1-Score of 93.8 % of all single classification models and represents a competitive alternative that does not require extensive feature engineering. Ultimately, the experiments showed that for this particular application example, a high average F1-Score of 95.2 % for error detection can be achieved with statistical significance when conventional machine learning and deep learning methods are combined to create an ensemble classifier.

In the future, more emphasis will be placed on unsupervised and semi-supervised methods for detecting anomalies and defects using a small number of training samples. Furthermore, it is envisaged to address the imbalances in the datasets, e.g., through cost-sensitive learning or random resampling techniques. In combination with advanced data augmentation methods this could further increase the performance of the machine learning methods presented in this paper.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, C.K. and M.K.; methodology, C.K.; software, F.S.; validation, P.A., M.K. and Z.Z.; formal analysis, A.G.; All authors have read and agreed to the published version of the manuscript."

Funding: This work is under the framework of EU projects QU4LITY and MAShES. These projects have received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825030 and No 637081 respectively. The dissemination of results herein reflects only the authors' view and the Commission is not responsible for any use that may be made of the information it contains.

Data Availability Statement: Data presented in this study can be obtained from the corresponding author upon request.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. You, D.Y.; Gao, X.D.; Katayama, S. Review of laser welding monitoring. *Science and Technology of Welding and Joining* **2013**, *19*, 181–201, doi:10.1179/1362171813Y.0000000180.
2. Shao, J.; Yan, Y. Review of techniques for on-line monitoring and inspection of laser welding. *J. Phys.: Conf. Ser.* **2005**, *15*, 101–107, doi:10.1088/1742-6596/15/1/017.
3. Stavridis, J.; Papacharalampopoulos, A.; Stavropoulos, P. Quality assessment in laser welding: A critical review. *Int J Adv Manuf Technol* **2018**, *94*, 1825–1847, doi:10.1007/s00170-017-0461-4.
4. Kim, C.-H.; Ahn, D.-C. Coaxial monitoring of keyhole during Yb:YAG laser welding. *Optics & Laser Technology* **2012**, *44*, 1874–1880, doi:10.1016/j.optlastec.2012.02.025.
5. Courtois, M.; Carin, M.; Le Masson, P.; Gaied, S.; Balabane, M. A complete model of keyhole and melt pool dynamics to analyze instabilities and collapse during laser welding. *Journal of Laser Applications* **2014**, *26*, 42001, doi:10.2351/1.4886835.
6. Saeed, G.; Zhang, Y.M. Weld pool surface depth measurement using a calibrated camera and structured light. *Meas. Sci. Technol.* **2007**, *18*, 2570–2578, doi:10.1088/0957-0233/18/8/033.
7. Bertrand, P.; Smurov, I.; Grevey, D. Application of near infrared pyrometry for continuous Nd:YAG laser welding of stainless steel. *Applied Surface Science* **2000**, *168*, 182–185, doi:10.1016/S0169-4332(00)00586-9.
8. Kong, F.; Ma, J.; Carlson, B.; Kovacevic, R. Real-time monitoring of laser welding of galvanized high strength steel in lap joint configuration. *Optics & Laser Technology* **2012**, *44*, 2186–2196, doi:10.1016/j.optlastec.2012.03.003.
9. Purtonen, T.; Kalliosaari, A.; Salminen, A. Monitoring and Adaptive Control of Laser Processes. *Physics Procedia* **2014**, *56*, 1218–1231, doi:10.1016/j.phpro.2014.08.038.
10. Zhang, Y.; Zhang, C.; Tan, L.; Li, S. Coaxial monitoring of the fibre laser lap welding of Zn-coated steel sheets using an auxiliary illuminant. *Optics & Laser Technology* **2013**, *50*, 167–175, doi:10.1016/j.optlastec.2013.03.001.
11. Knaak, C.; Kolter, G.; Schulze, F.; Kröger, M.; Abels, P. Deep learning-based semantic segmentation for in-process monitoring in laser welding applications. In *Applications of Machine Learning.* Applications of Machine Learning, San Diego, United States, 11–15 Aug. 2019; Zelinski, M.E., Taha, T.M., Howe, J., Awwal, A.A., Iftekharuddin, K.M., Eds.; SPIE, 2019 - 2019; p 2, ISBN 9781510629714.
12. Tenner, F.; Riegel, D.; Mayer, E.; Schmidt, M. Analytical model of the laser welding of zinc-coated steel sheets by the aid of videography. *Journal of Laser Applications* **2017**, *29*, 22411, doi:10.2351/1.4983236.
13. Schmidt, M.; Otto, A.; Kägeler, C. Analysis of YAG laser lap-welding of zinc coated steel sheets. *CIRP Annals - Manufacturing Technology* **2008**, *57*, 213–216, doi:10.1016/j.cirp.2008.03.043.
14. Wuest, T.; Weimer, D.; Irgens, C.; Thoben, K.-D. Machine learning in manufacturing: Advantages, challenges, and applications. *Production & Manufacturing Research* **2016**, *4*, 23–45, doi:10.1080/21693277.2016.1192517.
15. Xing, B.; Xiao, Y.; Qin, Q.H.; Cui, H. Quality assessment of resistance spot welding process based on dynamic resistance signal and random forest based. *Int J Adv Manuf Technol* **2018**, *94*, 327–339, doi:10.1007/s00170-017-0889-6.
16. Knaak, C.; Thombansen, U.; Abels, P.; Kröger, M. Machine learning as a comparative tool to determine the relevance of signal features in laser welding. *Procedia CIRP* **2018**, *74*, 623–627, doi:10.1016/j.procir.2018.08.073.
17. Jager, M.; Hamprecht, F.A. Principal Component Imagery for the Quality Monitoring of Dynamic Laser Welding Processes. *IEEE Trans. Ind. Electron.* **2009**, *56*, 1307–1313, doi:10.1109/TIE.2008.2008339.
18. You, D.; Gao, X.; Katayama, S. WPD-PCA-Based Laser Welding Process Monitoring and Defects Diagnosis by Using FNN and SVM. *IEEE Trans. Ind. Electron.* **2015**, *62*, 628–636, doi:10.1109/TIE.2014.2319216.
19. Cai, W.; Wang, J.; Cao, L.; Mi, G.; Shu, L.; Zhou, Q.; Jiang, P. Predicting the weld width from high-speed successive images of the weld zone using different machine learning algorithms during laser welding. *Mathematical Biosciences and Engineering* **2019**, *16*, 5595–5612, doi:10.3934/mbe.2019278.
20. Schmidhuber, J. Deep Learning in Neural Networks: An Overview. *Neural Networks* **2015**, *61*, 85–117, doi:10.1016/j.neunet.2014.09.003.
21. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.; et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97, doi:10.1109/MSP.2012.2205597.

22. Krüger, N.; Janssen, P.; Kalkan, S.; Lappe, M.; Leonardis, A.; Piater, J.; Rodríguez-Sánchez, A.J.; Wiskott, L. Deep hierarchies in the primate visual cortex: what can we learn for computer vision? *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1847–1871, doi:10.1109/TPAMI.2012.272.
23. Mohamed, A.-r.; Sainath, T.N.; Dahl, G.; Ramabhadran, B.; Hinton, G.E.; Picheny, M.A. Deep Belief Networks using discriminative features for phone recognition. *2011 IEEE International Conference*; pp 5060–5063.
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90, doi:10.1145/3065386.
25. Mart\'ın Abadi; Ashish Agarwal; Paul Barham; Eugene Brevdo; Zhifeng Chen; Craig Citro; Greg S. Corrado; Andy Davis; Jeffrey Dean; Matthieu Devin; et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015. http://tensorflow.org/.
26. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, R. Garnett, Eds.; Curran Associates, Inc, 2019; pp 8024–8035.
27. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. *Caffe: Convolutional Architecture for Fast Feature Embedding*, 2014. http://arxiv.org/pdf/1408.5093v1.
28. Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems* **2018**, *48*, 144–156, doi:10.1016/j.jmsy.2018.01.003.
29. Günther J.; Patrick M. Pilarski; Gerhard Helfrich; Hao Shen; Klaus Diepold. First Steps Towards an Intelligent Laser Welding Architecture Using Deep Neural Networks and Reinforcement Learning **2014**, doi:10.1016/j.protcy.2014.09.007.
30. Zhang, Y.; You, D.; Gao, X.; Zhang, N.; Gao, P.P. Welding defects detection based on deep learning with multiple optical sensors during disk laser welding of thick plates. *Journal of Manufacturing Systems* **2019**, *51*, 87–94, doi:10.1016/j.jmsy.2019.02.004.
31. Gonzalez-Val, C.; Pallas, A.; Panadeiro, V.; Rodriguez, A. A convolutional approach to quality monitoring for laser manufacturing. *J Intell Manuf* **2019**, *30*, 2505, doi:10.1007/s10845-019-01495-8.
32. Liu, T.; Bao, J.; Wang, J.; Zhang, Y. A Hybrid CNN–LSTM Algorithm for Online Defect Recognition of CO2 Welding. *Sensors (Basel)* **2018**, *18*, doi:10.3390/s18124369.
33. Ouyang, X.; Xu, S.; Zhang, C.; Zhou, P.; Yang, Y.; Liu, G.; Li, X. A 3D-CNN and LSTM Based Multi-Task Learning Architecture for Action Recognition. *IEEE Access* **2019**, *7*, 40757–40770, doi:10.1109/ACCESS.2019.2906654.
34. Fan, Y.; Lu, X.; Li, D.; Liu, Y. Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In *ICMI'16*, Proceedings of the 18th ACM International Conference on Multimodal Interaction : November 12-16, 2016, Tokyo, Japan. the 18th ACM International Conference, Tokyo, Japan, 11/12/2016 - 11/16/2016; Nakano, Y., Ed.; The Association for Computing Machinery, Inc: New York, NY, 2016; pp 445–450, ISBN 9781450345569.
35. Valiente, R.; Zaman, M.; Ozer, S.; Fallah, Y.P. Controlling Steering Angle for Cooperative Self-driving Vehicles utilizing CNN and LSTM-based Deep Networks. In *IV19*, 30th IEEE Intelligent Vehicles Symposium : 9-12 June 2019, Paris. 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 6/9/2019 - 6/12/2019; IEEE: [Piscataway, New Jersey], 2019?; pp 2423–2428, ISBN 978-1-7281-0560-4.
36. Yin, W.; Kann, K.; Yu, M.; Schütze, H. *Comparative Study of CNN and RNN for Natural Language Processing*, 2017. http://arxiv.org/pdf/1702.01923v1.
37. Bishop, C.M. *Pattern recognition and machine learning*, 11. (corr. printing); Springer: New York, 2013, ISBN 9780387310732.
38. *Feature extraction & image processing for computer vision, third edition;* Nixon, M.S.; Aguado, A.S., Eds., 3rd ed.; Academic Press: Kidlington, Oxford, U.K., 2012, ISBN 9780123965493.
39. Runkler, T.A. *Data Analytics. Models and Algorithms for Intelligent Data Analysis;* Vieweg+Teubner Verlag: Wiesbaden, 2012, ISBN 9783834825889.
40. Raschka, S. *Python machine learning. Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics;* Packt Publishing open source: Birmingham, Mumbai, 2016, ISBN 9781783555130.
41. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep learning;* MIT Press: Cambridge, Massachusetts, London, England, 2016, ISBN 9780262035613.
42. Pedregosa et al. Scikit-learn: Machine Learning in Python **2011**, *JMLR 12*, pp. 2825-2830.
43. Lee, K.B.; Cheon, S.; Kim, C.O. A Convolutional Neural Network for Fault Classification and Diagnosis in Semiconductor Manufacturing Processes. *IEEE Trans. Semicond. Manufact.* **2017**, *30*, 135–142, doi:10.1109/TSM.2017.2676245.
44. Arif, S.; Wang, J.; Ul Hassan, T.; Fei, Z. 3D-CNN-Based Fused Feature Maps with LSTM Applied to Action Recognition. *Future Internet* **2019**, *11*, 42, doi:10.3390/fi11020042.
45. Kim, D.; Cho, H.; Shin, H.; Lim, S.-C.; Hwang, W. An Efficient Three-Dimensional Convolutional Neural Network for Inferring Physical Interaction Force from Video. *Sensors (Basel)* **2019**, *19*, doi:10.3390/s19163579.

46. Zhu, G.; Zhang, L.; Shen, P.; Song, J. Multimodal Gesture Recognition Using 3-D Convolution and Convolutional LSTM. *IEEE Access* **2017**, *5*, 4517–4524, doi:10.1109/ACCESS.2017.2684186.
47. Ullah, A.; Ahmad, J.; Muhammad, K.; Sajjad, M.; Baik, S.W. Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. *IEEE Access* **2018**, *6*, 1155–1166, doi:10.1109/ACCESS.2017.2778011.
48. Rana, R. *Gated Recurrent Unit (GRU) for Emotion Classification from Noisy Speech*, 2016. http://arxiv.org/pdf/1612.07778v1.
49. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, 2014. http://arxiv.org/pdf/1406.1078v3.
50. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780, doi:10.1162/neco.1997.9.8.1735.
51. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *NIPS 2014 Workshop on Deep Learning, December 2014.*
52. DeWitt, D.P.; Nutter, G.D. *Theory and practice of radiation thermometry;* Wiley: New York, 1988, ISBN 9780471610182.
53. Suzuki, S.; be, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing* **1985**, *30*, 32–46, doi:10.1016/0734-189X(85)90016-7.
54. Flusser, J.; Suk, T.; Zitov, B. *Moments and moment invariants in pattern recognition;* J. Wiley: Chichester, West Sussex, U.K, Hoboken, N.J, 2009, ISBN 0470699876.
55. Feature Extraction & Image Processing for Computer Vision. In *Feature extraction & image processing for computer vision, third edition,* 3rd ed.; Nixon, M.S., Aguado, A.S., Eds.; Academic Press: Kidlington, Oxford, U.K., 2012; pp i–iii, ISBN 9780123965493.
56. Fitzgibbon, A.W.; Fisher, R.B. A Buyer's Guide to Conic Fitting. In *Proceedings of the 6th British Machine Vision Conference*, 11 - 14 September 1995, The University of Birmingham, Birmingham. British Machine Vision Conference 1995; Pycock, D., Ed.; British Machine Vision Association: Birmingham, 1995; 51.1-51.10, ISBN 0-9521898-2-8.
57. EN ISO 13919-1: Welding – Electron and laser-beam welded joints – Guidance on quality levels for imperfections – Part 1: Steel, nickel, titanium and their alloys (ISO/DIS 13919-1:2018); German and English version prEN ISO 13919-1:2018.
58. DIN EN ISO 6520-1: Classification of geometric imperfections in metallic materials – Part 1: Fusion welding (ISO 6520-1:2007); Trilingual version.
59. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era.
60. Zongker, D.; Jain, A. Algorithms for feature selection: An evaluation. In *Proceedings of 13th International Conference on Pattern Recognition.* Proceedings of 13th International Conference on Pattern Recognition, Vienna, Austria, 29/08/1996 - 29/08/1996; IEEE, 1996 - 1996; 18-22 vol.2, ISBN 0-8186-7282-X.
61. Edwards, A.L. Note on the correction for continuity in testing the significance of the difference between correlated proportions. *Psychometrika* **1948**, *13*, 185–187, doi:10.1007/BF02289261.
62. Vinodhini, G.; Chandrasekaran, R.M. Opinion mining using principal component analysis based ensemble model for e-commerce application. *CSIT* **2014**, *2*, 169–179, doi:10.1007/s40012-014-0055-3.
63. Hussain, E.; Mahanta, L.B.; Das, C.R.; Talukdar, R.K. A comprehensive study on the multi-class cervical cancer diagnostic prediction on pap smear images using a fusion-based decision from ensemble deep convolutional neural network. *Tissue Cell* **2020**, *65*, 101347, doi:10.1016/j.tice.2020.101347.
64. McNemar, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **1947**, *12*, 153–157, doi:10.1007/BF02295996.
65. Kotsiantis, S.; Kanellopoulos, D.; Pintelas, P. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering* **2005**, *30*, 25–36.