Journal of Nondestructive Evaluation (2020) 39:68
https://doi.org/10.1007/s10921-020-00719-9

Using Deep Learning for Defect Classification on a Small Weld X-ray Image Dataset

Chiraz Ajmi[1] · Juan Zapata[1] · José Javier Martínez-Álvarez[1] · Ginés Doménech[1] · Ramón Ruiz[1]

Received: 17 January 2020 / Accepted: 30 August 2020 / Published online: 4 September 2020

Abstract
This document provides a comparative evaluation of the performance of a deep learning network for different combinations of parameters and hyper-parameters. Although there are numerous studies that report on performance in deep learning networks for ordinary data sets, their performance on small data sets is much less evaluated. The objective of this work is to demonstrate that such a challenging small data set, such as a welding X-ray image data set, can be trained and evaluated obtaining high precision and that it is possible thanks to data augmentation. In fact, this article shows that data augmentation, also a typical technique in any learning process on a large data set, plus that two image channels, such as channels B (blue) and G (green), both are replaced by the Canny edge map and a binary image provided by an adaptive Gaussian threshold, respectively, gives to the network a 3% increase in accuracy, approximately. In summary, the objective of this work is to present the methodology used and the results obtained to estimate the classification accuracy of three main classes of welding defects obtained on a small set of welding X-ray image data.

Keywords Industrial X-ray images · Welding defects · Heterogeneities classification · Deep learning · Machine learning

1 Introduction

Welds are customarily used to attach two or more metal parts in a wide range of industrial activities. Because these junctions may suffer loads and fatigue during product lifetime, there is a possibility that they may be deficient due to typical welding defects such as lack of fusion or porosity inside the weld could cause a assemble to break or a structure to rupture. Therefore, it is often necessary to test specific pieces or materials to determine whether the structure is suitable for its designed use. Ideally such testing should be done without damaging the material, piece or structure. Nondestructive testing (NDT) is a wide group of analysis techniques used in science and technology industry to evaluate that the requirements of a material, component or system are satisfied without causing damage over the piece. In fact, NDT have gone from being a simple laboratory curiosity to an essential tool in industry in the last decades. Welds may be tested using NDT techniques such as industrial radiography, industrial computed tomography (CT) scanning using X-rays or gamma rays, ultrasonic testing, liquid penetrate testing, or magnetic particle inspection. Although it is one of the oldest techniques of non-destructive inspection, radiography is still accepted as essential for the control of welded joints in many industries such as the nuclear, naval, chemical, and aeronautical.

But unlike the past, these X-ray films are digitised or acquired digitally to be treated and processed on a digital computer. In this sense, an image classification algorithm must be applied to understand the content of the welding image, which is the task of using computer vision and machine learning algorithms to extract meaning from an image and to classify accordingly. This classification could be as simple as assigning a label to what the radiography image contains, or as advanced as interpreting its contents and returning a human-readable sentence. Without a doubt, image classification and image understanding are the most popular field of computer vision, and possibly will be for the next years. Information and communications technology (ICT) companies is acquiring image understanding startup companies in order to build applications for consumers (based on smartphones), companies and industries that can understand and interpret the content of an image or video.

✉ Juan Zapata
juan.zapata@upct.es

[1] Dpto. de Electrónica y Tecnología de Computadoras, Universidad Politécnica Cartagena, Cartagena, Spain

In the context of machine learning applied to image classification, the goal of a machine learning algorithm is to take a set of images and identify patterns that can be used to discriminate classes (objects) from one another. Traditional machine learning classifiers are simple image classification algorithms. For instance, the k-Nearest Neighbour classifier (k-NN) is so extremely simple that it actually learn nothing. Without specialised data structures, traditional machine learning classifiers scale linearly with the number of data points, making it challenging to use in high dimensions. This behaviour is in contrast to parameterised learning models [28] which spend a large amount of time upfront training a model to obtain high accuracy, and, in turn, have very fast classifications at testing time. This is the foundation on which all deep learning networks are built on. Using parameterised learning, these models can actually learn from input data and discover underlying patterns [7,17].

In the past, hand-engineered features were used as input data such as: Local Binary Patterns [23], Haralick texture [8], shape (Hu Moments [11], Zernike Moments [14]), color (color moments, color histograms, color correlograms [12]), keypoint detectors (FAST [27], Harris [9], DoG [19], etc) and local invariant descriptors (SIFT [19], SURF [1], BRIEF [2], ORB [4], to name a few too), but now we can use raw pixel intensities as inputs to our machine learning models, as is now common with deep learning.

Deep learning is a sub-field of machine learning, which is, in turn, a sub-field of artificial intelligence (AI). While AI embodies a large, diverse set of work related to automatic machine reasoning (inference, planning, heuristics, etc.), the machine learning sub-field tends to be specifically interested in pattern recognition and learning from data. Artificial Neural Networks (ANNs) are a class of machine learning algorithms that learn from data and specialise in pattern recognition, inspired by the structure and function of the brain. Deep learning belongs to the family of ANN algorithms, and in most cases, the two terms can be used interchangeably. Deep neural networks are ANNs with several hidden layers which have recently become a highly successful and popular research topic in machine learning due to their excellent performance in many benchmark problems and applications.

In the literature there is a poverty of research data on machine learning methods in general and deep learning in particular for weld defect classification. We are in the process of researching and analyses that would provide more sophisticated capabilities to our system for weld defects classification as our model integrates the best features of the deep learning and image processing techniques. Deep learning is flexible, tolerant of the imprecise data and can be built on top of the experience of experts which labelling the ground truth in direct contrast to no parameterised techniques, which take training data and generate opaque, impenetrable models. Deep learning relies on the experience of people who already understand your system.

The aim of this approach is to present the methodology used and the results obtained to estimate the classification accuracy of the three main classes of weld defects. Our methodology tries to solve some shortcomings of the works carried out in the past. First, we try to obtain very good accuracy in the classification using a very small challenge dataset, second, we use different combination of data augmentation with the aim to know the best combination that improves the performance of the classification and third, our deep learning method for weld defect classification was used to automate the process of classification in the three main types of weld defects met in practice.

In the following section, some published works on the classification of defects in automated radiographic inspection are presented; next, in Sect. 3 the experimental methodology is explained, how the dataset was obtained and why it is a challenging dataset, how data augmentation was carried out in order to avoid overfitting, the network architecture is presented and finally, how dataset curation and hyper-parameters setting are performed. Section 4 shows results of training and validation for different settings and, in the end, Sect. 5 shows conclusions and main contributions of this work.

2 Related Works

It is interesting to note the relative lack of published work on the classification of defects in automated radiographic inspection applications using neural networks. Carvalho et al. [3] evaluated the use of artificial neural networks (ANNs) for pattern recognition of magnetic flux leakage (MFL) signals in weld joints of pipelines. ANNs were applied to classify signal patterns with three types of defects in the weld joint: external corrosion (EC), internal corrosion (IC) and lack of penetration (LP). Di et al. [5] presented a method based on classification of the obtained features using self-organizing feature map (SOM) neural networks in order to get the weld quality information. Subsequently, Liao and Tang applied a multilayer perceptron [18] (MLP [26]) for extracting welds from digitized radiographic images. The procedure consists of three major components: feature extraction, MLP-based object classification, and postprocessing.

Peng [24] discusses an effective method to extract the features of the defects much simple algorithm which is based on perceptron model to recognise and classify the defects. Experimental results show that the pretreatment of the images of welding lines is very important to the feature extraction and defects recognition and the and method of recognition and classification of defects put forward is effective. Shen et al. [29] suggested three classifiers (one-versus-rest SVM, one-versus-one SVM and MLP neuron network) and a group of feathers are used to compare with the classifier and the feature group we proposed. The bootstrap estimate is used to estimate their performances. Zapata et al. [33] describe an automatic system to detect, recognise, and classify welding defects in radiographic images and evaluate the performance for two neuro-classifiers based on an artificial neural network (ANN) and an adaptive-network-based fuzzy inference system (ANFIS). Shitole et al. [30] presents a research using advanced methods for automatic interpretation and classification of weld defects in Time-of-Flight Diffraction (TOFD) data. In the classification stage three different classification techniques are employed and compared: an artificial neural network-based classifier, a fuzzy logic-based classifier and a hybrid neural-fuzzy classifier.

Sutcliffe et al. [32] describes the development of an automatic defect recognition system applicable to full matrix capture (FMC) imaged data. Computer vision principles were used on FMC-reconstructed images for feature extraction and combined with a multi-layer perceptron artificial neural network for classification. Necceredine et al. [21] used an unsupervised classifier based on a finite mixture model using the multivariate generalised Gaussian distribution (MGGD). The parameters of the nonzero-mean MGGD-based mixture model are estimated using the Expectation-Maximisation algorithm where, exact computations of mean and shape parameters are originally provided. Hou et al. [10], presented a recent work based on a deep neural network model for an automatic detection of weld defects, contained in 88 X-ray images taken from a public database called GDXray [20]. Their proposed model obtains a maximum classification accuracy rate of 91.84%.

3 Experimental Methodology

3.1 A Small Welding X-ray Image Dataset

The images that populate the dataset were obtained from an X-ray image acquisition system. X-ray films can be digitised by several systems. An overview of the applicability of existing film digitisation systems to non-destructive testing can be found in [6,35]. The most usual way of digitisation is through scanner, which work with light transmission—usually called transparency adapters. In this present work, an UMAX[1] scanner was used, model: Mirage II (maximum optical density: 3.3; maximum resolution for films: 2000 dpi) to scan the International Institute of Welding (IIW)[2] films. The spatial resolution used in the study was 500 dpi (dots per inch), totalling an average image size of 2900 pixels (horizontal length) × 950 pixels (vertical length) in colour, which resulted in an average pixel size of 50 mm. Such resolution was adopted for the possibility of detecting and measuring defects of hundredths of millimetres, which, in practical terms of radiographic inspection, is much higher than the usual cases.

The data set is characterised by a great diversity of image sizes (they were resized in a range which varies from 640 × 480 to 720 × 576 pixels on three channels red, green and blue) due to different distributions and sizes of heterogeneities that the radiography film conforms. The database only includes X-ray welding images organised in three folders. Each folder represents a class or image tag. Each class has a number of images that differ from the rest of the folders. Therefore, the lack of penetration group has 57 images, the pinhole group only has 44 images and the porosity group has 115 images, being the most populous class of all.

The semantic gap is the difference between how a human perceives the contents of an image versus how an image can be represented in a way a computer can understand the process because it acquires an image like if it were a big matrix of pixels. The semantic gap can be salved if feature extraction were applied to quantify the contents of image. There are two solutions to accomplish this feature extraction process. Firstly, hand-engineered feature extractors (such Histograms of Gradients (HoG), Local Binaries Patterns (LBPs), or other traditional approaches) may be applied. Secondly, deep learning may be applied to automatically learn a set of features that can be used to quantify and ultimately label the contents of the image itself. But this deep learning procedure can face itself to other challenges proposed for the set of images. Our dataset can be considered a challenging dataset due to the dramatic factors of variation in how an image appears such as changes in: view point, or how an image can present oriented/rotated in multiple dimensions due to how the object is acquired; scale, or how one image can be represented with a same object but with different size; varying lighting conditions, or how there is the possibility of that the acquisition of image can be different among films due to thick of material; background clutter, or how an image can be composed of several similar objects with similar colours and textures and how these objects occlude one another, and how parts from different objects may be intermingled; and intra-class variation, or how an image can contents the same object but with (slightly or not) different geometrical forms.

Therefore, our dataset, with only a few dozens images per class, becomes challenging for deep learning models to learn a representation for each class without overfitting. As a general rule of thumb, it's advisable to have 1000–5000 example images per class when training a deep neural network [7]. In this paper we will present methods to improve classification using data augmentation and image preprocessing on small datasets. Figure 1 shows diverse examples of dataset.

[1] www.umax.org

[2] www.iiwelding.org

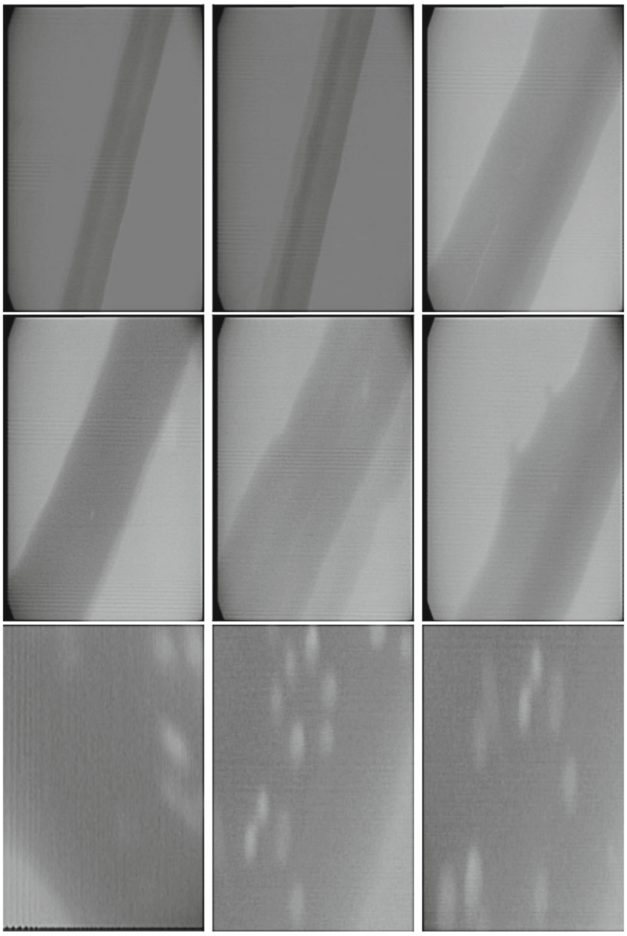


Fig. 1 Our dataset can be considered a challenging dataset due to the dramatic factors of variation in how an image appears: first row shows lack of penetration examples, second row shows pinhole examples, and third row shows porosity examples

3.2 Data Augmentation

One of the most ordinary issue for learning algorithms is to avoid overfitting which they is faced. Overfitting is a situation where learning model performed exceptionally well on training data, but was not able to predict well with testing data. Regularisation helps in overcoming the overfitting problem and improves the performance of learning model. Regularisation is any modification which is made to a learning algorithm that is intended to reduce its generalisation error, but not its training error [7]. In short, regularisation seeks to reduce testing error perhaps at the expense of increasing training error slightly.

There are different forms of regularisation such as: L2 and L1 regularisation, modify the network architecture itself (dropout), early stopping and data augmentation. Data augmentation modifies training samples, changing their appearance slightly by applying random jitters and transformations such that the classes labels are not changed, before passing them into the network for training. The end result is that a network consistently sees "new" training data generated (data are augmented) from the original training data. The need to gather more training data because they are not exist more in dataset is partially alleviating. This usually provides a big impulse in improving the accuracy of the model. It can be considered as a mandatory technique in order to improve our predictions.

A main aim when applying data augmentation is to increase the generalization power of the model. Given that network is constantly seeing new, slightly modified versions of the input data, it is able to learn more robustly. At testing time, data augmentation are not applied and trained network is evaluated obtaining an increase in testing accuracy, and perhaps at the expense at a slight down in training accuracy.

Two types of data augmentation has been performed in this work. Firstly, a series of small modifications have been made with training images were introduced to the network. Specifically, there have been made random rotations of 30 degrees, horizontal displacements of 10%, vertical displacements of 10%, shearing of 10% and horizontal flip of 10%. In all cases the interpolation of the image has been carried on with the nearest pixels. In all these slight modifications, the essential semantics of the images did not changed, i.e., the original label of its training image. In Fig. 2 is shown ten different augmented images from training set image to visualise how data augmentation is carried on.

Secondly, channels B (blue) and G (green) have been replaced by the map of Canny edges and its binary image provided by an adaptive Gaussian threshold, respectively. In this way, the network is fed with a group of data that provide not only the grey levels of the image but also its edges and binary image. This increase in information gives to the network a small increase of approximately 3% in accuracy, as shown in Sect. 4. In order to visualise this data augmentation, in Fig. 3 is shown an X-ray image and its augmented channels.

3.3 A Network Architecture Based on VGGnet

VGGNet was first introduced by VGG (Visual Geometry Group) from University of Oxford in 2014 [31]. VGGNet was the first runner-up in the classification task and winner in the localisation task in ILSVRC 2014. The main contribution of Visual Geometry Group was demonstrating that an architecture as VGGNet with very small (3×3) filters can be trained to increasingly higher depths (VGG16 and VGG19 layers) and obtain state-of-the-art classification on the challenging ImageNet classification challenge. The main ideas was that by using layers of 3×3 filters, it actually have already covered bigger effective area. Previously, network architectures in the deep learning literature used a mix of filter sizes, large-size filters such as 11×11 in AlexNet [15] and 7×7 in ZFNet

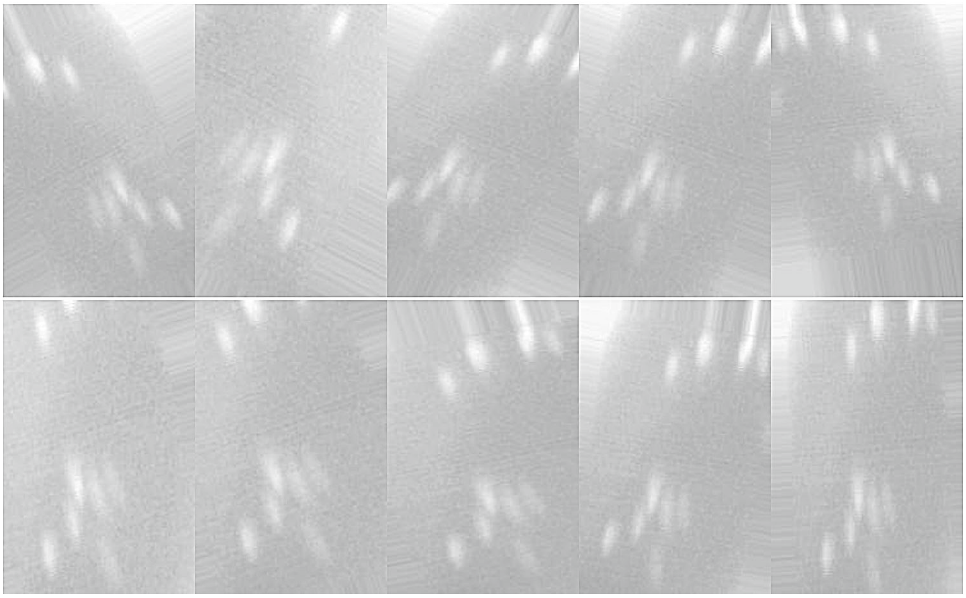


Fig. 2 Data augmentation over image originates 10 new different images with the same label. Each image has been randomly rotated, sheared, shifted, zoomed, and flipped, within the following ranges: degree range for random rotation = 30, fraction of total width shift = 0.1, fraction of total height shift = 0.1, Shear angle in counter-clockwise direction in degrees = 0.2, zoom = 0.2 or range for random zoom [0.8,1.2] and flip = true

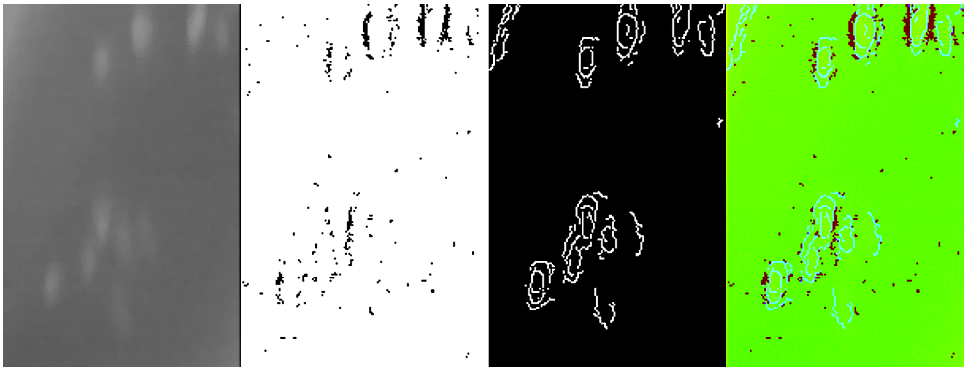


Fig. 3 From left to right, grey-level X-ray image or R channel, adaptive Gaussian thresholding image or G channel, Canny edges image or B channel, merged image (Color figure online)

[34] indeed are not needed. From there, filter sizes progressively reduced to 5×5. Finally, only the deepest layers of the network used 3×3 filters. VGGNet was the first in using 3×3 kernels throughout the entire architecture. Firstly, the use of these small kernels helps VGGNet to generalise classification problems outside what the network was originally trained on and, secondly, the number of parameters to learn are fewer which is better for faster convergence, and for reduction of overfitting problem. Another characteristics is that VGGNet stacks multiple convolution layers using rectified linear units (ReLU) as activation function before applying a pool operation.

Our network is based on VGG but is not a VGG16 or VGG19. The choice of VGG is because we believe that it is an architecture suitable for our purposes due to it is shallow, with good and fast convergence and therefore excellent for quick testing. In Fig. 4 is shown a scheme of our model. This model of VGGNet disposes of two sets convolution with rectified linear unit (ReLU) as activation function layer before applying a pool operation; it is due to it is pretended the network should learn more rich features from the convolutional layers before downsampling the spatial input size via the pool operation. Following, a full connected set with rectified linear unit (ReLU) activation function and a full connected set with softmax activation function layers are stacked to network. The first two convolutional layers will learn 32 filters, each of size 3×3. The second two convolutional layers will learn 64 filters, again, each of size 3×3. Pool layers will perform max pooling over a 2×2 window with a 2×2 stride. Batch normalisation layers after the activations along with dropout layers after the pool and full connected layers are also inserted. The initial input image size is assumed to be 64×64×3.

There is some debate on whether the batch normalization layer should be before or after the activation layer [13]. We have carried out a preliminary study that has not been conclusive, it seems that in our case some better performance is obtained after the activation layer. In any case, it is not in our interest to analyze the network performance, but rather to demonstrate that through a relatively shallow network, good performance can be obtained with a small and unbalanced database through data augmentation and reusing the

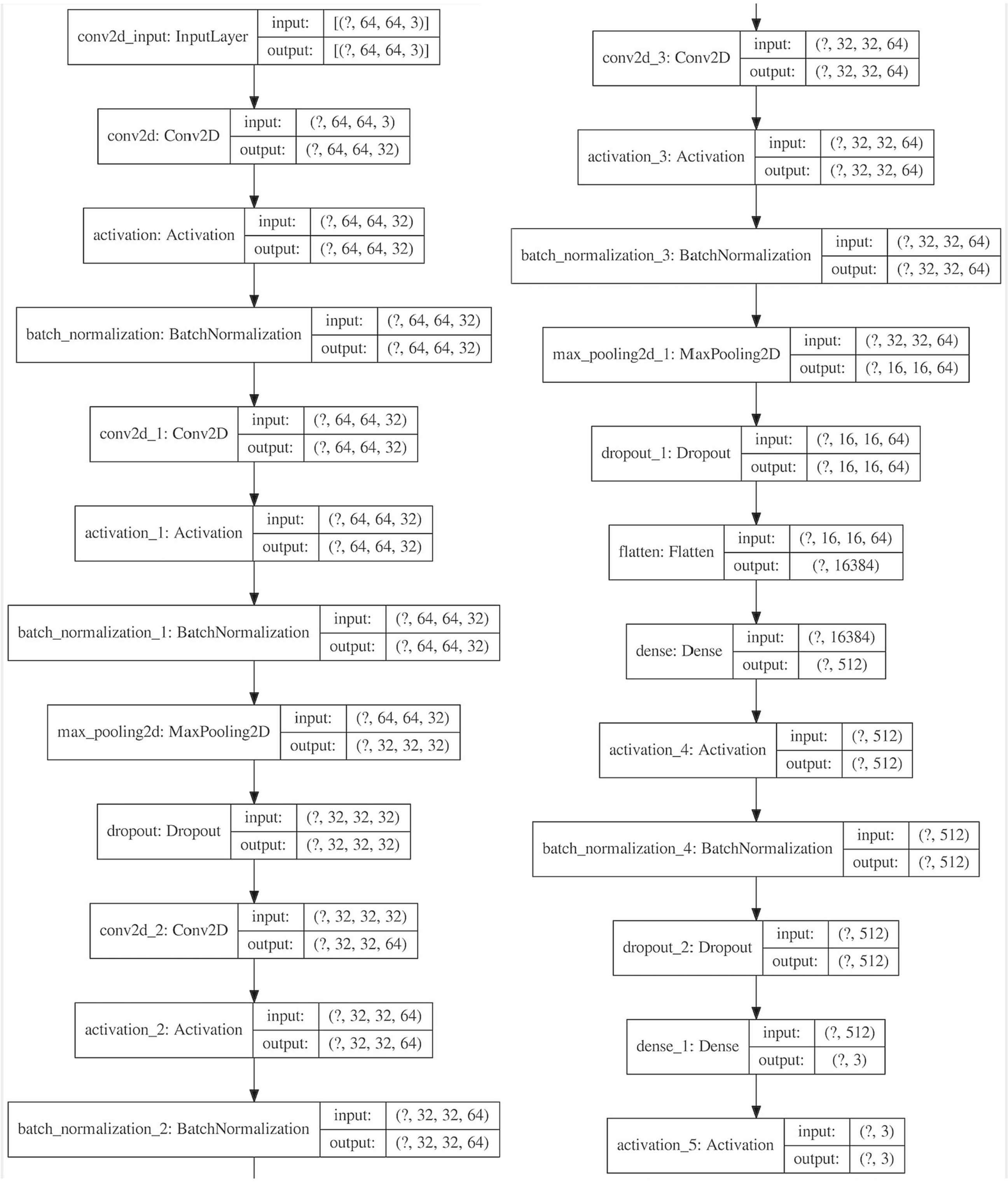

conv2d_input: InputLayer input: [(?, 64, 64, 3)] output: [(?, 64, 64, 3)]
conv2d: Conv2D input: (?, 64, 64, 3) output: (?, 64, 64, 32)
activation: Activation input: (?, 64, 64, 32) output: (?, 64, 64, 32)
batch_normalization: BatchNormalization input: (?, 64, 64, 32) output: (?, 64, 64, 32)
conv2d_1: Conv2D input: (?, 64, 64, 32) output: (?, 64, 64, 32)
activation_1: Activation input: (?, 64, 64, 32) output: (?, 64, 64, 32)
batch_normalization_1: BatchNormalization input: (?, 64, 64, 32) output: (?, 64, 64, 32)
max_pooling2d: MaxPooling2D input: (?, 64, 64, 32) output: (?, 32, 32, 32)
dropout: Dropout input: (?, 32, 32, 32) output: (?, 32, 32, 32)
conv2d_2: Conv2D input: (?, 32, 32, 32) output: (?, 32, 32, 64)
activation_2: Activation input: (?, 32, 32, 64) output: (?, 32, 32, 64)
batch_normalization_2: BatchNormalization input: (?, 32, 32, 64) output: (?, 32, 32, 64)
conv2d_3: Conv2D input: (?, 32, 32, 64) output: (?, 32, 32, 64)
activation_3: Activation input: (?, 32, 32, 64) output: (?, 32, 32, 64)
batch_normalization_3: BatchNormalization input: (?, 32, 32, 64) output: (?, 32, 32, 64)
max_pooling2d_1: MaxPooling2D input: (?, 32, 32, 64) output: (?, 16, 16, 64)
dropout_1: Dropout input: (?, 16, 16, 64) output: (?, 16, 16, 64)
flatten: Flatten input: (?, 16, 16, 64) output: (?, 16384)
dense: Dense input: (?, 16384) output: (?, 512)
activation_4: Activation input: (?, 512) output: (?, 512)
batch_normalization_4: BatchNormalization input: (?, 512) output: (?, 512)
dropout_2: Dropout input: (?, 512) output: (?, 512)
dense_1: Dense input: (?, 512) output: (?, 3)
activation_5: Activation input: (?, 3) output: (?, 3)

Fig. 4 Summary of the model architecture. input/output volume sizes are included for each layer. Only 3×3 convolutions are applied. The sign ? is used in order to indicate what any number of input images is possible

Table 1 Hyper-parameter and settings used in our network based on VGGnet

Image size	64 × 64
R Channel	Grey-level X-ray image
G Channel	Adaptive Gaussian thresholding image
B Channel	Canny edge image
Train set	60%
Test set	20%
Validation set	20%
Optimisation algorithm	SGD
Learning rate (lr)	0.05, 0.025, 0.015
Momentum (m)	0.0, 0.9
Nesterov acceleration (N)	true, false
Batch size	32
Data augmentation	10 images
Epochs	100
Loss function	Categorical cross entropy

color channels of the image to increase the information that is delivered to the network.

3.4 Dataset Curation and Hyper-parameters Setting

As the input of the deep network expects a 64 × 64 input, the images of the dataset had to be resized to this size without keeping the aspect or the ratio of the width to the height of the image. Although be from a strictly aesthetic point of view, the aspect ratio of the image when resizing an image should be maintained, most neural networks and Convolutional Neural Networks applied to the task of image classification assume a fixed size input which meaning that the dimensions of all images must be passed through the network must be the same. Common choices for width and height image sizes used as input to Convolutional Neural Networks include 32×32, 64×64, 224×224, 227×227, 256×256, and 299×299. Actually, the size of the input of the network was absolutely selected in a heuristics way, i.e., a 64 × 64 was adopted because VGGnet has a input size of 224×224 and therefore, it is 64 is 4 times less than 224 approximately and indeed it is a power of two. Therefore if dataset consists of images that are 312×234, 800×600, and 770×300, among other image sizes, the aspect ratio the images can be ignored and permitted the distortion.

But the real first component of building a deep learning network is to present an initial tensor to the network formed by the images themselves as well as the associated labels which come from a finite set of categories. Furthermore, the number of images for each category should be approximately uniform (i.e., the same number of examples per category). If we have twice the number of class A than class B, and five times the number of class C than class B, then our classifier will become naturally biased to overfitting into these heavily-represented categories. Class imbalance is a common problem in machine learning and there exist a number of ways to overcome it but the best method to avoid learning problems due to class imbalance is to simply avoid class imbalance entirely or to use an appropriate metric.

Generally, that appropriate metric deals with the trade-off between recall (R, percent of truly positive instances that were classified as such) and precision (P, percent of positive classifications that are truly positive). In situations where detecting instances of a minority class is required, it is usually more concerned recall than precision, as in the context of a binary classification, it is usually more costly to miss a positive instance than to falsely label a negative instance. Thus, when comparing approaches to imbalanced classification problems, using metrics beyond accuracy such as recall, precision, F-measure and AUROC (Area Under the Receiver Operating Characteristics) must be consider.

Next, the data must be split in two parts, a training set and and testing set, where it is extremely important that the training set and testing set are independent of each other and do not overlap. It is said, if testing set is part of training set, then classifier has an unfair advantage since it has already seen the testing examples before and learned from them. Then, we have obtained the validation set using the training set. Therefore, there are 3 datasets in a split 60/20/20. Training set are sample of data used to fit the model, validation set provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper-parameters, and test set provide an unbiased evaluation of a final model.

These data splits make sense, but deep networks have a number of hyper-parameter (e.g., learning rate, decay, regularisation, etc.) that need to be tuned and dialled to obtain

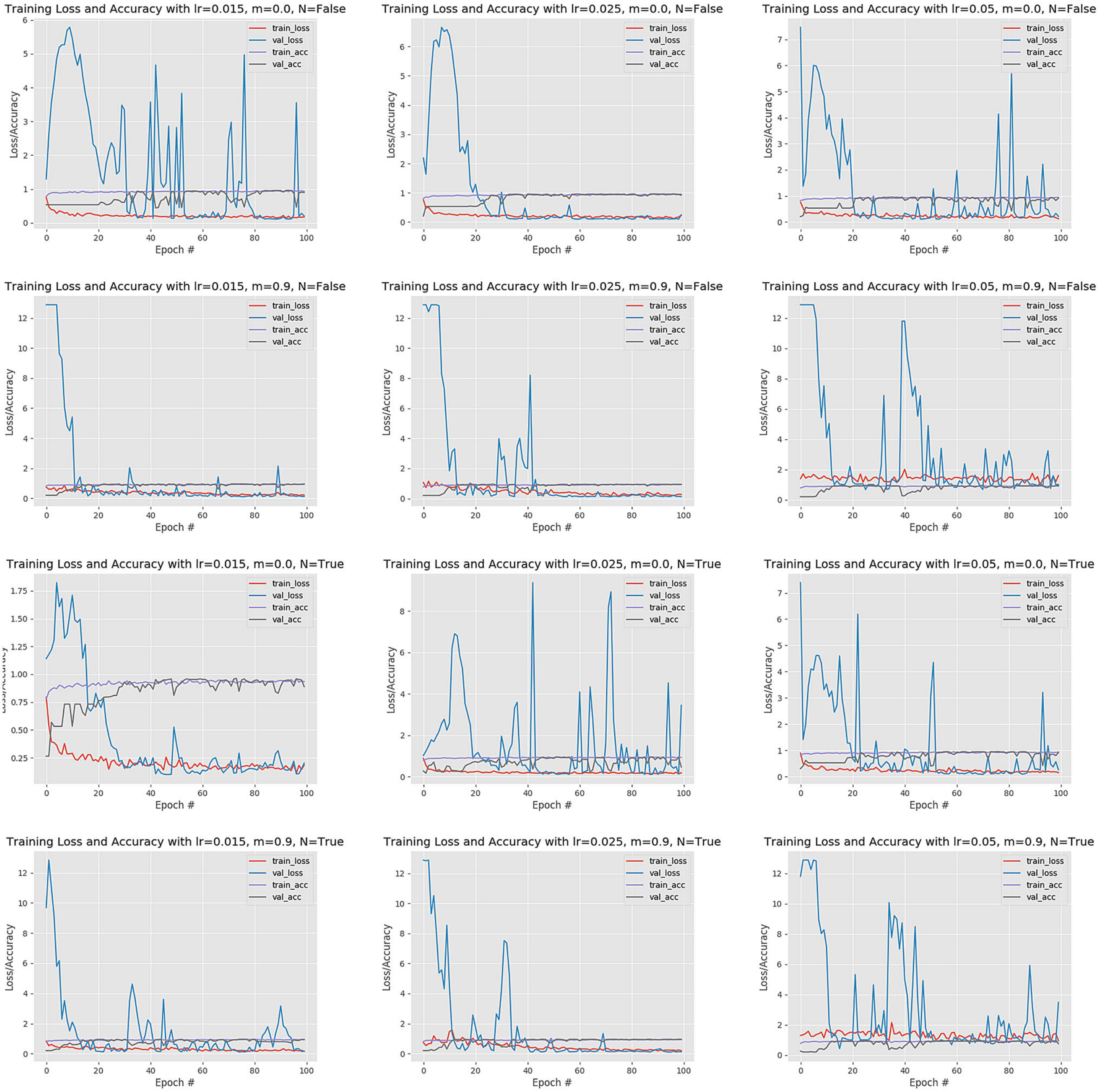


Fig. 5 Examples of training and validation curves using data augmentation without channel substitution (*lr* learning rate, *m* momentum, *N* Nesterov acceleration). The graph in row 4 column 2 is the best result. In general, we can see that good results are obtained with a lr = 0.025 (column = 2) and momentum = 0.9. The worst results are obtained along column = 3, that is, with lr = 0.05

optimal performance and therefore it is critical that they get set properly. In practice, testing data could be used to tweak these values, but again the test set must be only used in evaluating the performance of deep network.

Therefore, given a training set of images the deep learning network can be trained with a goal which is to learn how to recognise each of the categories in labelled data set. When the deep learning model commits a mistake, it learns from that mistake and improves itself. This is performed applying a form of gradient descent which explanation is out of scope of this work, but briefly, gradient descent is an optimisation algorithm used to minimise some function by iterative movement in the direction of steepest descent as defined by the negative of the gradient. In machine learning, gradient descent is used to update the parameters of model.

Our model uses categorical cross entropy as loss function. Categorical cross entropy is a loss function that is used for the categorization of a single label. This is when only one category is applicable for each data point. In other words, an example can belong to a single class. Categorical cross entropy is different from binary cross entropy. Binary cross entropy only allows two classes to be classified. With categorical cross entropy, classification is not limited to two classes, and therefore, how many classes your model has can be classified. It is a Softmax activation plus a cross entropy loss. With this loss function, a deep network can be trained to output a probability over the dataset classes for each image. So, this loss function is used for multi-class classification. The cross entropy compares the model's prediction with the label which is the true probability distribution. The cross entropy goes down as the prediction gets more and more accurate. It becomes zero if the prediction is perfect. As such, the cross entropy can be a loss function to train a classification model.

Last, evaluating trained network is necessary. For each of the images in testing set, it must be presented to the network and ask it to predict what it thinks about the label of the image must be. The predictions of the model for an image in the testing set must be evaluated. Finally, these model predictions are compared to the ground-truth labels from testing set. The ground-truth labels represent what the image category actually is. From there, the number of predictions our classifier got correct and compute aggregate reports such as precision, recall, and F-measure, which are used to quantify the performance of network as a whole.

One of the first hyper-parameters to explore is the learning rate for the optimisation algorithm that is used to learn a set of classifier weights for parameterised deep learning. The most famous of all them is Stochastic Gradient Descent (SGD) [16], a simple modification to the standard gradient descent algorithm that computes the gradient and updates the network weights on small batches of training data. Typical batch sizes include 32 (our case), 64, 128, and 256. There are two primary extensions to SGD. The first is momentum [25], a method used to accelerate SGD, enabling it to learn faster by focusing on dimensions whose gradient point in the same direction. The second method is Nesterov acceleration [22], an extension to standard momentum. Nesterov acceleration can be conceptualised as a corrective update to the momentum which lets us obtain an approximate idea of where our parameters will be after the update. Although is not enough clear, in practice SGD with momentum is used with large dataset (such as ImageNet) and Nesterov acceleration is used with small dataset. Table 1 shows all hyper-parameters used in our model.

4 Results

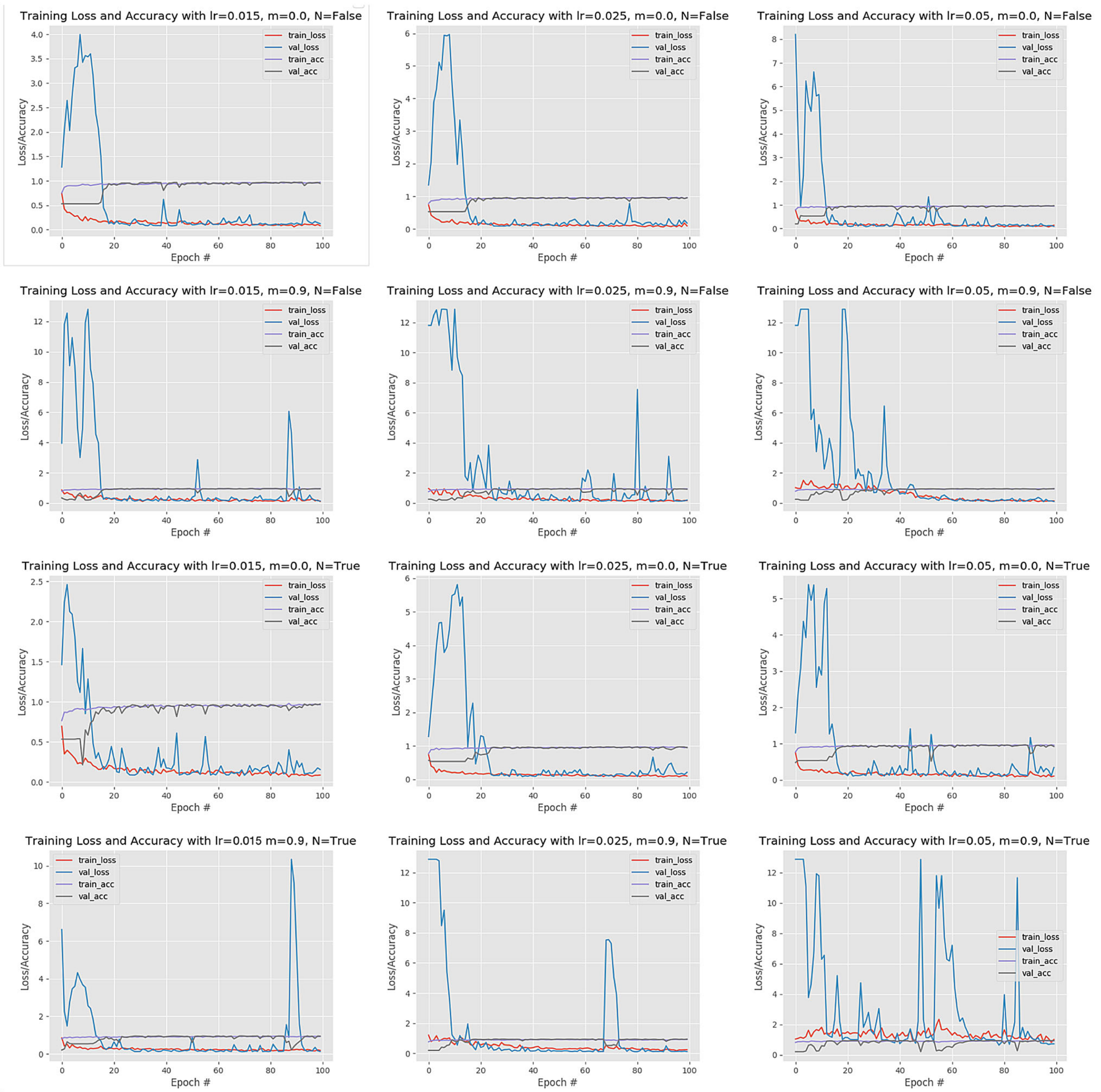
In order to obtain quantitative results which allow to compare the benefits of the techniques employed, different training on the network have been carried out using distinct approaches as it was explained above. But this paper has not got as goal to obtain a precise and complete repository which results of each one of combinations possible among different techniques, parameters and hyper-parameters in order to obtain the best classification. Our goal is to show that on small and challenge dataset is possible obtain excellent results in classification using deep learning with data augmentation and replacing channels B (blue) and G (green) by maps of Canny edges and its binary image provided by an adaptive Gaussian threshold, respectively.

Our results pointed that with momentum and Nesterov acceleration the training and validation curves are smoother with better accuracy with learning rates about 0.025. In Fig. 5 is shown some examples of these training curves using only data augmentation without augmentation of channels.

Table 2 shows the best report obtained in terms of metrics, which is training with a learning rate equal to 0.025 moment equal to 0.9 and using Nesterov acceleration only with data augmentation without channel substitution. For reference to the reader, this table matches the graph in row 4 column 2 of Fig. 5. The ideal system corresponds to precision and recall equal to one. In practice, a compromise between these two quantities exists: a system with a high recall is likely to have false positives, and a system with high precision is likely to miss some true annotations. Our main goal is proof that a deep learning network can be trained on a small database using an ad-hoc data augmentation, and that obtaining of good results on that unusual small dataset is possible in terms of classification. The report shows precision, recall, F-measure and support for each class. The precision is the ratio $t_p/(t_p + f_p)$; and the recall is the ratio $t_p/(t_p + f_n)$ where t_p is the number of true positives, f_p the number of false positives, and f_n the number of false negatives. In other words, the precision is intuitively the ability of the classifier not to label as positive a sample that is negative and the recall is intuitively the

Table 2 Classification report for a learning rate equals to 0.025, momentum equals to 0.9 and using Nesterov acceleration using data augmentation without channel substitution

	Precision	Recall	F1-score	Support
Lack penetration	0.93	0.89	0.91	57
Pinholes	0.87	0.91	0.89	44
Porosities	1.00	1.00	1.00	115
Accuracy			0.95	216
Macro avg	0.93	0.93	0.93	216
Weighted avg	0.95	0.95	0.95	216



Training Loss and Accuracy with lr=0.015, m=0.0, N=False

Training Loss and Accuracy with lr=0.025, m=0.0, N=False

Training Loss and Accuracy with lr=0.05, m=0.0, N=False

Training Loss and Accuracy with lr=0.015, m=0.9, N=False

Training Loss and Accuracy with lr=0.025, m=0.9, N=False

Training Loss and Accuracy with lr=0.05, m=0.9, N=False

Training Loss and Accuracy with lr=0.015, m=0.0, N=True

Training Loss and Accuracy with lr=0.025, m=0.0, N=True

Training Loss and Accuracy with lr=0.05, m=0.0, N=True

Training Loss and Accuracy with lr=0.015 m=0.9, N=True

Training Loss and Accuracy with lr=0.025, m=0.9, N=True

Training Loss and Accuracy with lr=0.05, m=0.9, N=True

Fig. 6 Examples of training and validation curves using data augmentation with channel substitution (*lr* learning rate, *m* momentum, *N* Nesterov acceleration). In general, we can see that good results are obtained with any lr and momentum using channel substitution. The worst results are obtained along column = 3, that is, with lr = 0.05

ability of the classifier to find all the positive samples. Often, the two quantities are summarised into a single number, F, defined as the harmonic mean of precision (P) and recall (R): $F = 2PR/(P + R)$. The value of F measures the accuracy of a test taking values from 0 to 1 with 1 being the best possible case and 0 the worst case. The harmonic mean between precision and recall has been considered, known as traditional F-measure or balanced F1Score. The support is the number of occurrences of each class. The reported averages include macro average (averaging the unweighted mean per label) and weighted average (averaging the support-weighted mean per label)

Channel augmentation is the next parameter to consider in order to compare the benefits of this technique. Figure 6

Table 3 Classification Report for a learning rate equals to 0.015, momentum equals to 0.0 and using Nesterov acceleration using data augmentation with channel substitution

	Precision	Recall	F1-score	Support
Lack penetration	0.90	1.00	0.95	57
Pinholes	1.00	0.86	0.93	44
Porosities	1.00	1.00	1.00	115
Accuracy			0.97	216
Macro avg	0.97	0.95	0.96	216
Weighted avg	0.97	0.97	0.97	216

Table 4 Mean for classification report using data augmentation without channel substitution

	Precision	Recall	F1-score	Support
Lack penetration	0.85	0.94	0.90	57
Pinholes	0.93	0.80	0.89	44
Porosities	0.99	0.99	0.99	115
Accuracy			0.92	216
Macro avg	0.92	0.93	0.92	216
Weighted avg	0.95	0.95	0.95	216

Table 5 Mean for classification report using data augmentation with channel substitution

	Precision	Recall	F1-score	Support
Lack penetration	0.87	0.95	0.91	57
Pinholes	0.93	0.82	0.87	44
Porosities	1.00	0.99	0.99	115
Accuracy			0.95	216
Macro avg	0.96	0.96	0.96	216
Weighted avg	0.97	0.97	0.97	216

shows that both curves, training and validation, are smoother and present better results in theirs metric, as in Table 3 is shown. Table 3 shows the best report obtained in terms of metrics, which is training with a learning rate equal to 0.015 moment equal to 0.0 and using Nesterov acceleration with data augmentation and channel substitution. For reference to the reader, this table matches the graph in row 3 column 1 in Figure 6. In general, it can be said that the set of curves where channel substitution has been used has had a better performance than the previous ones where channel augmentation has not been used. In gist, they have had higher metrics than without channel augmentation. In fact, Tables 4 and 5show the average of the metrics of each set without and with channel substitution. These tables have been obtained by averaging the classification reports of each of the hyperparameter combinations shown in the graphs of Figs. 5 and 6. As can be seen, Table 5 shows better results than Table 4, which implies that channel substitution provides some boost in performance for classification.

For some graphs in Figs. 5 and 6, there is a high fluctuation in the validation loss even for higher epoch numbers. The possible reason is precisely because it is a very unbalanced dataset that causes validation loss in some epochs. One way to avoid this is to get more samples for that class, or to make a specific augmented data on that class more than the rest to compensate this unbalance.

5 Conclusions

In view of the results obtained, different statements can be made. First, and the most important, and that is the main goal of this work, is that a small and unbalanced dataset can be classified with good metrics using data augmentation techniques. Second, is that these data augmentation techniques do not have to be limited to displacements, turns and scaling. By the way, these data augmentation procedures are widely used in deep neural networks with much larger ordinary data sets. That is, for a small data set such as welding of X-ray images, that data augmentation can be extended, since colour channels can be reused as feature vectors that provide the deep neural network of augmented information that allows get more learning. In our case, we have replaced two channels with an edge map and a binarised image. We believe that other operators can be used with some or replace them to allow the introduction of other more specific information. The third is that the network does not necessarily have to be very deep to obtain a classification of very few classes with great accuracy. This is important, because this implies that network like our can be trained on Commercial Off-The-Shelf GPUs. The network was trained with 100 epochs and took 18 s for epoch on a GPU Ge Force GTX 1080 Ti that shows that our deep network is relatively shallow.

Without a doubt, there is work to be done in the imminent future. The first is to try to discover which other feature maps can be introduced into the channels of the welding X-ray images that are more discriminating for the deep neural network. It is a challenge since there is a pleiad of possibilities. We believe that the relative shallowness of the network would allow some flexibility when it comes to training in the search for these better discriminating maps. The second approach would try to discover other network architectures that fit the dataset more adequately. Siamese networks are clearly candidates, as they have a flexible architecture when adapting to the dataset. Siamese networks do not learn to classify, they learn to compare, and comparison is the main basis of classification. Data augmentation with elastic deformation which has successfully be applied as a more sophisticated augmentation type for several classification tasks and can be used in the future.

Acknowledgements The authors wants to acknowledge the work of the rest of the participants in this project, namely: J.A. López-Alcantud, P. Rubio-Ibañez, Universidad Politécnica de Cartagena, J.A. Díaz-Madrid, Centro Universitario de la Defensa - UPCT and T.J. Kazmierski, University of Southampton. This work has been partially funded by Spanish government through project numbered RTI2018-097088-B-C33 (MINECO/FEDER,UE).

References

1. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: In ECCV, pp. 404–417 (2006). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.85.2512
2. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: binary robust independent elementary features. In: Proceedings of the 11th European Conference on Computer vision: Part IV, ECCV'10, pp. 778–792. Springer-Verlag, Berlin, Heidelberg (2010). http://dl.acm.org/citation.cfm?id=1888089.1888148
3. Carvalho, A., Rebello, J., Sagrilo, L., Camerini, C., Miranda, I.: Mfl signals and artificial neural networks applied to detection and classification of pipe weld defects. Ndt & E Int. **39**(8), 661–667 (2006)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886–893 (2005). http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1467360&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1467360
5. Di, L., Yonglun, S., Feng, Y.: Online monitoring of weld defects for short-circuit gas metal arc welding based on the self-organizing feature map neural networks. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, vol. 5, pp. 239–244 (2000)
6. Ewert, U., Zscherpel, U., Horkey, M., Kennedy, J., Hutchinson, M.: A new computer based concept for digital radiographic reference images. J. Nondestr. Test. **7**(12), 1–13 (2002)
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
8. Haralick, R.M., Shanmugam, K.S., Dinstein, I.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **3**(6), 610–621 (1973)
9. Harris, C., Stephens, M.: A combined corner and edge detector. In: In Proc. of Fourth Alvey Vision Conference, pp. 147–151 (1988). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.231.1604
10. Hou, W., Wei, Y., Guo, J., Jin, Y., Zhu, C.: Automatic detection of welding defects using deep neural network. J. Phys. **933**, 012006 (2018). https://doi.org/10.1088/1742-6596/933/1/012006
11. Hu, M.K.: Visual pattern recognition by moment invariants. Inf. Theory IRE Trans. **8**(2), 179–187 (1962). https://doi.org/10.1109/TIT.1962.1057692
12. Huang, J., Kumar, R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: CVPR, pp. 762–768. IEEE Computer Society (1997). http://dblp.uni-trier.de/db/conf/cvpr/cvpr1997.html#HuangKMZZ97
13. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR arXiv:abs/1502.03167 (2015). http://dblp.uni-trier.de/db/journals/corr/corr1502.html#IoffeS15
14. Khotanzad, A., Hong, Y.: Invariant image recognition by zernike moments. IEEE Trans. Pattern Anal. Mach. Intell. **12**(5), 489–497 (1990). https://doi.org/10.1109/34.55109
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, pp. 1097–1105. Curran Associates Inc., New York (2012)
16. Laboratories, S.U.S.E., Widrow, B., of Naval Research, U.S.O., Corps, U.S.A.S., Force, U.S.A., Navy, U.S.: Adaptive "adaline" neuron using chemical "memistors.". (1960). https://books.google.es/books?id=Yc4EAAAAIAAJ
17. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015). https://doi.org/10.1038/nature14539
18. Liao, T.W., Tang, K.: Automated extraction of welds from digitized radiographic images based on mlp neural networks. Appl. Artif. Intell. **11**(3), 197–218 (1997)
19. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
20. Mery, D., Riffo, V., Zscherpel, U., Mondragón, G., Lillo, I., Zuccar, I., Lobel, H., Carrasco, M.: Gdxray: the database of x-ray images for nondestructive testing. J. Nondestr. Eval. **34**(4), 42 (2015)
21. Nacereddine, N., Goumeidane, A.B., Ziou, D.: Unsupervised weld defect classification in radiographic images using multivariate generalized gaussian mixture model with exact computation of mean and shape parameters. Comput. Ind. **108**, 132–149 (2019). https://doi.org/10.1016/j.compind.2019.02.010
22. Nesterov, Y.E.: A method for solving the convex programming problem with convergence rate $o(1/k^2)$. Dokl. Akad. Nauk SSSR **269**, 543–547 (1983)
23. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
24. Peng, J.J.: A method for recognition of defects in welding lines. In: 2009 International Conference on Artificial Intelligence and Computational Intelligence, vol. 2, pp. 366–369. IEEE (2009)
25. Qian, N.: On the momentum term in gradient descent learning algorithms. Neural Netw. **12**(1), 145–151 (1999). https://doi.org/10.1016/S0893-6080(98)00116-6
26. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psych. Rev. **65**, 386–407 (1958)
27. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. IEEE Int. Conf. Comput. Vis. **2**, 1508–1511 (2005). https://doi.org/10.1109/ICCV.2005.104
28. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Prentice Hall Press, Upper Saddle River (2009)
29. Shen, Q., Gao, J., Li, C.: Automatic classification of weld defects in radiographic images. Insight-Non-Destr. Test. Cond. Monit. **52**(3), 134–139 (2010)
30. Shitole, C.N., Zahran, O., Al-Nuaimy, W.: Combining fuzzy logic and neural networks in classification of weld defects using ultrasonic time-of-flight diffraction. Insight-Non-Destr. Test. Cond. Monit. **49**(2), 79–82 (2007)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014). arXiv:1409.1556
32. Sutcliffe, M., Lewis, J.: Automatic defect recognition of single-v welds using full matrix capture data, computer vision and multi-layer perceptron artificial neural networks. Insight-Non-Destr. Test. Cond. Monit. **58**(9), 487–493 (2016). https://doi.org/10.1784/insi.2016.58.9.487

33. Zapata, J., Vilar, R., Ruiz, R.: Performance evaluation of an automatic inspection system of weld defects in radiographic images based on neuro-classifiers. Expert Syst. Appl. **38**(7), 8812–8824 (2011)

34. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. CoRR **abs/1311.2901** (2013). http://dblp.uni-trier.de/db/journals/corr/corr1311.html#ZeilerF13

35. Zscherpel, U., Berlin, B.: Film digitisation systems for dir: standards, requirements, archiving and printing. J. NDT Ultrason. **5**, 5 (2000)