

## Advanced Fringe Patterns Denoising and Processing via Ensemble Deep Learning Model

**Vibekananda Dutta, Michał Józwik, Kohei Nimura**

Institute of Micromechanics and Photonics, Faculty of Mechatronics, Warsaw University of Technology, Warsaw, Poland

**Piotr Sosinowski, Adriana Mazurek**

Central Office of Measures, Time and Length Department, Warsaw, Poland

Corresponding author's e-mail address: vibekananda.dutta@pw.edu.pl

### Abstract

The strength of deep learning methods has shown substantial achievements in optical metrology, especially in fringe denoising, fringe analysis, and phase unwrapping. Despite extensive research efforts for decades, the challenge of accurately extracting desired phase distribution information from recorded fringes remains one of the most challenging open problems. This study introduces an ensemble methodology that leverages two deep neural models, such as Single Shot Detector (SSD), and You Only Look Once (YOLO), for automated fringe pattern analysis in decision-making support for the determination of parameters from the calculated phase distribution. The assessment of the proposed methodology, as detailed in this study, was conducted using heterogeneous datasets encompassing data recorded in Kösters interferometer placed in the laboratory of the Polish Central Office of Measures and computer-generated fringe sample images. We benchmark our methodology for the fringe analysis task and find generalization behavior and robustness to noisy recorded data.

**Keywords:** Artificial Intelligence (AI), Neural Networks, Fringe Pattern Analysis, Segmentation, Interferometry, Length of Gauge Block

## 1. Introduction

The analysis of interference fringe patterns constitutes a fundamental technique in interferometric measurements [16]. Fringes produced and observed through an interferometer carry diverse information, mainly reflecting the optical path difference between two light waves. This information can be exploited to obtain topographical data without direct contact with the measured element [5]. Combined with the contemporary image sensors, this method enables measurements with nanometer precision. For various optical measurement techniques, such as interferometry [6], digital holography [15], and fringe projection profilometry (FPP) [22], the accuracy and efficiency of phase retrieval from recorded fringe images are crucial for facilitating decision-making support in optical metrology. Optical metrology experiments are often conducted within customized systems and stringent environments.

The Temporal Phase Shifting (TPS) represents a well-known technique in optical interferometry [24]. It involves acquiring a series of interferograms that undergo phase shifts relative to each other. The intensity function of each interferogram is extracted and utilized in the trigonometric phase equation to obtain a phase function. However, due to the inherent characteristics of the arc-tangent function, the resulting phase map may exhibit discontinuities, commonly referred to as the wrapped phase map. The wrapped phase map is then unwrapped to mitigate these discontinuities, ultimately yielding a continuous phase map. This continuous phase map facilitates visualizing three-dimensional (3D) information about the observed object. TPS methods offer a deterministic and straightforward approach to the phase retrieval problem by acquiring multiple fringe patterns [25]. These methods bestow advantages in terms of both speed and accuracy, thereby facilitating the development of numerous optical metrology instruments.

When conducting analysis using experimentally acquired interferograms, accurately segmenting the fringes observed on the target element from other features present in the image is crucial for obtaining precise results. This is because regions of the image unrelated to the target element serve little purpose in the measurement process, aside from providing contextual information about the positioning of the subject element. Furthermore, excluding excessive background areas from the calculation is essential to minimize the risk of introducing noise or artifacts, such as dust particles, which could distort the resulting phase map. Our goal is to extract the specific fringe pattern of interest from the surrounding areas of the image, which may also contain other fringe patterns. Existing algorithms typically employed for detecting fringe patterns in a general sense are unsuitable for our particular task.

While various image processing software tools can somewhat optimize this process, manual selection becomes highly subjective and time-consuming for the user, as it involves manually selecting the pixels to be extracted from the measurement. Therefore, there is a high demand for methods that automatically detect the region of interest (ROI) within the entire image [8]. Researchers have long sought to attain high-precision real-time measurements. In 2018, various researchers conducted reviews on high dynamic range technology, absolute phase calculation, and phase-shifting methods in fringe pattern profilometry (FPP) [3], [24]. Furthermore, in 2020, the authors in [22] provided a comprehensive summary that addressed the current state, challenges, and future prospects of FPP. However, these aforementioned review papers primarily emphasized conventional FPP algorithms rather than deep learning algorithms.

Despite decades of extensive research efforts, achieving phase measurement with the highest possible accuracy from the minimum number of fringe patterns remains one of the most challenging issues in optical metrology.

In this work, we propose an advanced preprocessing method of fringe patterns that are used in precise interferometric measurements. As a result of the preprocessing block, we obtain denoised images with automatically selected ROI. In our application case, the interferometric measurements of the gauge block length deviation in Kösters interferometer, the denoised fringe patterns are masked in order to remove the edges of the gauge block, marks on its surface, and optimize the measurement area.

With recent advancements in computational power, deep-learning algorithms, which are a subset of machine learning, emerged as a powerful tool that allows the multi-layer neural network to learn the characteristics of data. In recent years, significant efforts have been made to utilize machine learning algorithms [2], [17], [11] for optical metrology. These algorithms present a significant advantage by automating fringe processing tasks with minimal user intervention. Our work aligns with this growing trend as we introduce a machine-learning model that leverages the strength of two deep neural models, such as Single Shot Detector (SSD) [9] and You Only Look Once (YOLO) [18], respectively. This machine learning model eliminates the need for manual selection of ROI, thereby offering an automated approach to fringe segmentation tasks. Furthermore, it requires a minimum training dataset, which makes the model computationally efficient. In addition, the strength of each neural model provides a robust estimation of the feature vectors that are strong and robust against noise. We aim to propose an automated decision-making system that is computationally efficient and robust against severe noise.

Following the inspiration from the potential of deep learning to solve our fringe segmentation task, we introduced a deep ensemble learning approach to achieve high-precision phase measurement results. This approach aims to automate fringe segmentation tasks, thereby improving decision-making support for optical measurement systems. The proposed deep ensemble neural network (DENN) combines the strengths of the SSD neural model and the YOLOv4 neural network, trained using a dataset incorporating noisy fringe patterns and their corresponding noiseless fringe patterns, referred to as ground truth. Each model offers distinct advantages and limitations based on its training objectives. Our task is composed of the following subsets: (a) detection of interference Figure 1. The fringe fraction calculation example: a) phase-shifted interferograms, b) calculated distribution of phase fringes, c) vertical cross-section. ence fringe enclosed within another interference fringe of different phases by employing the capabilities of the SSD neural network, (b) utilization of the YOLOv4 neural network architecture to detect the texts written on the surface of the subject element, (c) detection of visible center lines, and (d) conducting a comparative analysis with existing state-of-the-art methods on both computer-generated fringe and experimental data for our interference fringe segmentation task.

The structure of the paper is as follows: Section 2 facilitates an overview of the data-gathering process, which delineates the collection of experimental and computer-generated fringes and the data pre-processing stage. In section 3, the authors also discuss the theoretical principles of the chosen neural models, followed by the proposed methodology and training strategy. Section 4 presents experimental validation and comparison of the results with those obtained with other popular methods. The article ends with concluding remarks and discussion.

## 2. Data Collection i Preprocessing

### 2.1 Collection of Experimental Interferograms

Multiple wavelength interferometry is one of the widely used techniques for length measurement of gauge blocks and end bars to the highest precision [20]. In this study, the experimental setup was a Kösters interferometer placed in the laboratory of the Polish Central Office of Measures. During a calibration process, the central length deviation of the gauge block is measured by the method of exact fractions [10]. The measurement technique involves the generation of tilt between a reference surface and the front faces of the gauge under test. As a result of tilting, there is an image with three interference fringes on the gauge surface and the bottom reference surface. For each wavelength, five images with exact phase shifts must

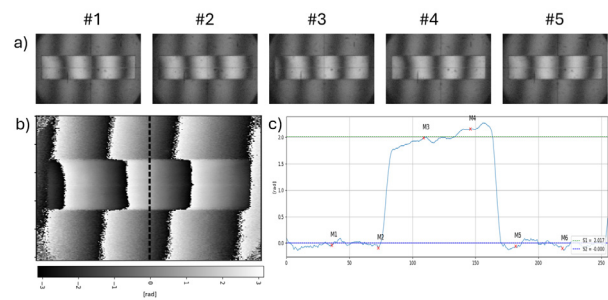


Fig. 1. The fringe fraction calculation example: a) phase-shifted interferograms, b) calculated distribution of phase fringes, c) vertical cross-section.

be acquired (Figure 1a). To calculate the phase fringes distribution (Figure 1b), we are using the TPS method with 5 images. From the calculated phase, the vertical profile in the central area of data is selected, and the value of the fringe fraction is determined as a difference of phase value determined in the upper and lower part of the profile (Figure 1c).

Measurement of the fringe fractional discontinuity at four of known wavelengths allows the unambiguous determination of the integral interference order number with a fringe fraction, and then from fringe fractions, the calculation of the central length deviation of the measured gauge block. It is the edges of the gauge block and the markings on its surface, as well as the crosshairs visible in the camera image 1, that can disturb the process of calculating the phase fraction from the cross-sections of the measured phase profile. In this case, we propose to use automatic finding and masking of problematic areas.

The method of calculation of phase fraction from the wrapped phase has some drawbacks. The interferometric fringes must be adjusted to the vertical position. Inaccurate adjustment will result in tilted fringes and errors in determining phase from the vertical cross-section. That's why, in the system, there is an aiming cross visible in all intensity images. Unfortunately, it causes a local phase error that should be omitted in the calculation. The position of the selected cross-section is very important. It has to be located in the central part of the phase data and the central part of the phase fringe. The calculation is based only on one vertical cross-section, while the system delivers 2D data. The phase fringes may be unwrapped, but in many cases, the resulting phase has many discontinuities. The solution could be a smart selection of the area of the sample and the surrounding area of reference (that is outcome of proposed preprocessing), then individual unwrapping of split areas. In this new scenario, the final step should be a tilt removal based on a calculated plane from the reference data only. In this case, the whole area of the gauge block can be used in the surface-based calculation of the phase fraction.

## 2.2 Preparation of Computer-Generated Fringe Samples

Following our decision to employ deep learning for sample detection, we initiated the development of our object detector by collecting the necessary sample data for training our neural models. This dataset comprised computer-generated fringe images generated using MATLAB, all sized at 256 by 256 pixels, mimicking the outcomes of an interferometer measuring the characteristics of a rectangular sample object. These computer-generated fringe images were derived from a base image representing the ideal state, incorporating various sample conditions and image disturbances. Parameters reflecting sample conditions included the position, size, angle orientation, frequency, and phase shift of observed fringes. Simulated disturbances encompassed alterations in fringe contrast and the introduction of noise, resulting in blurred images. Additionally, variations arising from the presence of text labels on samples and center lines due to optical characteristics were specifically incorporated for our intended experimentation.

For each variable that changed within a defined range, sets of samples were generated alongside samples featuring numerous combinations of variants and inclusions. Ultimately, a collection of 515 computer-generated fringe sample images was prepared. Each image was accompanied by ground truth labels in the form of rectangles, indicating the location and size of the subject element (ROI), as well as the presence of unwanted texts, if applicable. Figure 2 illustrates the list of example images of computer-generated fringe image samples.

The ideal sample image was generated with the following characteristics: (a) resolution – 256 by 256 pixels, (b) simulated vertical fringes for wavelength –  $\lambda = 536.9$  nm, (c) fringe frequency generated as –  $\cos \frac{7\pi}{\lambda}$ , (d) target element (ROI) of size  $128 \times 77$  placed at the center. Based on this ideal sample, the following modifications were made to generate variant samples:

- ROI size ranging from  $232 \times 140$  to  $26 \times 16$ .
- Translation of ROI center position within the area of size  $102 \times 156$  at the center of the whole image.
- Rotation of ROI from  $0$  to  $\pi$ .
- Fringe frequency ranging from  $\cos \frac{\pi}{\lambda}$  to  $\cos \frac{40\pi}{\lambda}$ .
- Phase shift of fringe from  $0$  to  $2\pi$  with step of  $\frac{\pi}{2}$ .
- Modification of fringe contrast by ranging the maximum intensity from  $11$  to  $256$ .
- Applying random noise to the image by adding it to a matrix of size  $256 \times 256$  consisting of randomly generated numbers. The noise matrix was multiplied with a modifier ranging from  $0.1$  to  $1$  for the noise intensity variant.

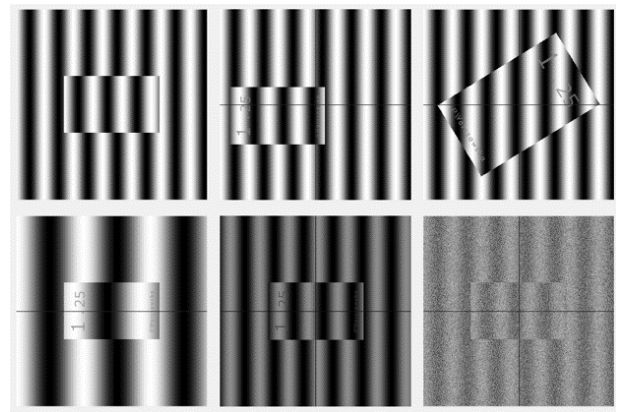


Fig. 2. Examples of computer-generated fringe sample images generated for the training of CNN. Top row from left to right: Ideal sample image with no inclusion or defects; sample displaced from the center with inclusion of unwanted texts and center lines; sample with different size and angle orientation. The bottom row, left to right, is a sample with reduced fringe frequency, a sample with decreased contrast, and a sample with computer-generated fringe noise.

## 2.2 Preparation of Computer-Generated Fringe Samples

Once the sample set was prepared, the data underwent the process of the so-called data augmentation. Data augmentation involves generating additional image sets derived from the original dataset by applying geometrical transformations and color space transformations. Commonly employed modifications include flipping, rotating, scaling, and adjustments in color factors such as saturation, brightness, and hue. This augmentation process not only expands the dataset but also enables the neural network to learn from new conditions of the targeted object that were not covered by the original computer-generated

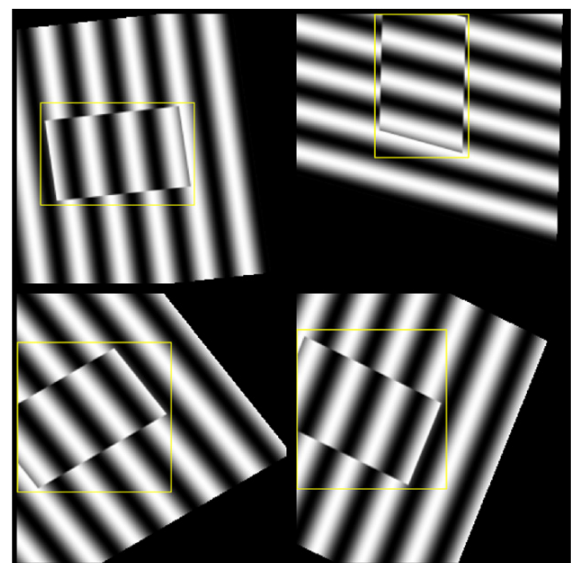


Fig. 3. Derived images from a single computer-generated fringe image after data augmentation. The yellow rectangles are from the label of this sample in the original image and were kept intact with the sample after the modifications.

fringe images. Various works have employed this data augmentation technique to generate datasets [7], [21]. Although this work decided to perform the data augmentation based on geometric transformations, some color-space transformations could be included to improve the quality of the dataset.

For our dataset, the applied modifications include (see Figure 3): (a) rotation with a random value ranging from 0 to 359 degrees, (b) random translation along both axes, with one-third of the image size being the maximum distance, (c) random shearing in both directions within the range of -30 to 30 degrees. These limits were selected to prevent excessive modification of images that could potentially cause the sample to disappear from the image.

### 3. Methodology

#### 3.1 Theoretical Principal

##### 3.1.1 Single Shot Detector (SSD)

Single Shot Detection (SSD) is a method for real-time object detection that processes an image through a neural network once to detect multiple objects [9]. SSD is often compared with the YOLO detector, which operates similarly to SSD but employs a differently structured neural network architecture. Generally, SSD is known for its higher accuracy compared to YOLO, while YOLO is preferred for the very fast detection of small objects. A crucial component of the SSD framework is the feature map, which detects the presence and position of target objects within an image. The feature map overlays a grid on the image, dividing it into smaller sections. Each section has a set of predefined anchor boxes, which are used to approximate the size and aspect ratio of the objects to be detected. For each anchor box applied to the image, the feature map generates a new bounding box that is refined to better match the ground truth bounding box. The output of a feature map includes the following attributes:

- Anchor box offsets - center position and the dimensions of the refined bounding box, related to the ground truth box (x-coordinate of center, y-coordinate of center, width, height).
- Confidence score - value between 0 and 1, describing the probability that the target object is present within the bounding box.

The values of these attributes are used in the loss function to quantify the prediction error relative to the ground truth information. The loss function consists of two components: localization loss ( $L_{loc}$ ) and confidence loss ( $L_{conf}$ ). Localization loss  $L_{loc}$  measures the error based on the difference between

the predicted bounding box and the ground truth box. It is expressed as follows:

$$L_{loc}(y, b, t) = \sum_{k \in pos} \sum_{n \in \{x, y, W, H\}} y_{km}^n \cdot \text{smooth}_{L_1}(b_k^n - t_k^n)$$

where  $y_{km}^n$  is an indicator for matching the k-th default box to the m-th ground truth box of class  $n$ ,  $b_k$  and  $t_k$  are the predicted and ground truth box parameters.  $k \in pos$  is the set of indices k corresponding to positive matches, i.e. default boxes that have been assigned to ground truth objects.  $n \in \{x, y, W, H\}$  represents the bounding box parameters (center x-coordinate, center y-coordinate, width (W), and height (H)). Finally,  $\text{smooth}_{L_1}$  is the loss function, which is less sensitive to outliers compared to  $L_2$  loss and is commonly used for bounding box regression.

The confidence loss  $L_{conf}$  measures how accurately the model predicts the presence of an object and its class label (or background) for each default box.

$$L_{conf}(y, p) = - \sum_{k \in pos} y_{km}^n \log(\hat{p}_k^n) - \sum_{k \in neg} \log(\hat{p}_k^0)$$

where  $\hat{p}_k^n$  is the softmax probability for class  $n$ .  $k \in pos$  is the set of indices for positive matches (boxes assigned to objects) and  $k \in neg$  is the set of indices for negative matches (boxes not assigned to any object—background). Additionally,  $\log(\cdot)$  denotes the natural logarithm, used in the cross-entropy formulation of classification loss. The total loss ( $L$ ) is the weighted sum of localization loss and the confidence loss, expressed as:

$$L(y, p, b, t) = \frac{1}{M} (L_{conf}(y, p) + \beta L_{loc}(y, b, t)),$$

where  $M$  is the number of matched default boxes, and  $\beta$  is a weight factor which balances the contribution of the localization loss relative to the confidence loss. Typically set to emphasize localization more or less depending on task sensitivity. During training, the overall loss is calculated at every iteration to serve as a gauge to define the accuracy of the detector model. The aim of repeating numerous iterations is to get the loss value lower than a certain margin, which can confirm the model's performance for the given task.

##### 3.1.2 You Only Look Once (YOLO)

You Only Look Once (YOLO) object detection [14] is an algorithm developed primarily to address the computational speed issues inherent in conventional methods such as R-CNN. Similar to SSD, YOLO partitions the input image into grids and employs anchor boxes to detect and classify objects simultaneously within each grid. The YOLO v4 network uses CSPDarkNet-53 [18] as the backbone for extracting features from the input images. All versions of YOLO use extensive data augmentation, divided into photometric distortions and geometric distortions. In dealing

with photometric distortions, the brightness, contrast, hue, saturation, and noise of an image are adjusted. The key distinction from SSD lies in how YOLO incorporates the intersection-over-union (IoU) value alongside anchor box offsets and confidence scores. As the name suggests, IoU represents the ratio of the area shared by both the ground truth and predicted bounding boxes to the total area encompassed by their union. This IoU value plays a crucial role in YOLO's loss function, formulated as follows:

$$L_{IoU} = 1 - IoU$$

This general form of IoU loss function was considered to be not precise enough, so new versions of the YOLO algorithm with improved IoU loss functions were introduced. In our development, we employed YOLO version 4, which employs the improved version of the loss function that accounts for the aspect ratio difference of the ground truth and predicted boxes and their relative positions. The IoU of ground truth

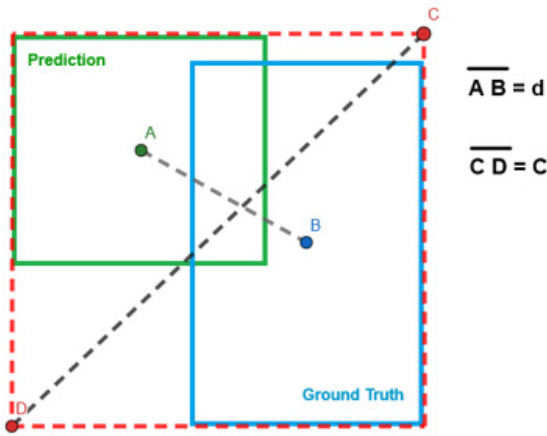


Fig. 4. Visual description of IoU and its relevant distances.

and predicted bounding boxes is visualized in Figure 4.

The red dotted box represents the smallest convex rectangle that encloses both the ground truth and predicted boxes. The distance between the centers of the two boxes is presented as  $d$ . The distance between the two corners of the enclosing box that makes a diagonal line within is presented as  $C$ . Taking those distances into account, the IoU loss function for YOLO version 4, also known as the Complete IoU (CIoU) loss function ( $L_{CIoU}$ ) [19], is presented as the following:

$$L_{CIoU} = 1 - IoU + \frac{d^2}{C^2} + \alpha v,$$

where:

$$v = \frac{4}{\pi^2} \left( \arctan \frac{t^W}{t^H} - \frac{b^W}{b^H} \right)^2,$$

$$\alpha = \frac{v}{(1 - IoU) + v}.$$

Here,  $d$  represents the Euclidean distance between the centers of the predicted bounding box and the ground truth bounding box, and  $C$  denotes the diagonal length of the smallest enclosing box that encloses both the predicted and ground truth bounding boxes.  $\alpha$  is a positive scaling factor that balances the impact of the aspect ratio term  $v$ .  $v$  measures the consistency (difference) between the aspect ratios of the predicted and ground truth boxes.  $(t^W, t^H)$  and  $(b^W, b^H)$  are the width and height of the ground truth and predicted bounding boxes, respectively. The CIoU loss function improves upon traditional IoU loss by considering additional factors such as orientation and aspect ratio discrepancies, leading to more accurate and stable optimization for object detection models.

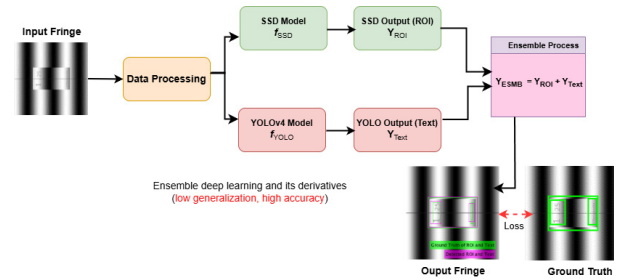


Fig. 5. Overview of the proposed deep ensemble neural network for fringe denoising and analysis. The data-driven deep learning approach - a) SSD - Single Shot Detector, b) YOLO - You Only Look Once.

### 3.2 Proposed Deep Ensemble Neural Network

In Figure 5, a deep ensemble neural network (DENN) model is employed for segmentation, which is an end-to-end network. The proposed deep ensemble learning strategy involves using two base models combined to perform tasks, rather than relying on a single model. By leveraging different architectures that capture distinct information, better decisions can be made through the combination of various networks. Inspired by recent successful applications of deep ensemble learning, we demonstrate that an ensemble leveraging the strengths of two powerful deep neural models can substantially improve the accuracy and stability of fringe-pattern analysis.

In our development, the overall process is separated

into two tasks: the detection of target object fringes (ROI) and the detection of text within the object. The detection of the ROI is handled by the trained SSD network, while the detection of text is managed by the YOLOv4 network due to its capability of detecting smaller features at a relatively high speed. Hence, in the rest of the paper, our developed model will be referred to as the combined model, which operates in a two-stage process to detect both ROI and text.

In the network structure, the input layer is composed of the fringe pattern with noise. The input sample applied to the network is expressed by the row vector  $X \in \mathbb{R}^{H \times W \times C}$ , where  $H$  is the height,  $W$  is the width, and  $C$  is the number of channels (in our work,  $C = 1$ ). The gray level of the image is represented by 8 bits. The base models leverage the strength of SSD and YOLOv4, which are convolutional neural networks that are good at extracting both local and global features. The fringe detection model (ROI detection) using SSD is represented by  $f_{SSD}: \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H' \times W' \times D}$ . Next, the text detection model using YOLOv4 is expressed by  $f_{YOLOv4}: \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H'' \times W'' \times D}$ . Here,  $D$  represents the number of output channels (bounding box coordinates, class probabilities, etc.). To combine the outputs, we used the weighted sum. Here, we consider a simple weighted sum for this purpose:

$$Y_{\text{ensemble}} = \alpha \cdot Y_{\text{ROI}} + \beta \cdot Y_{\text{text}},$$

where  $\alpha, \beta \in \mathbb{R}$  are the weights with  $\alpha + \beta = 1$ . Additionally,  $Y_{\text{ROI}} = f_{SSD}(X) \in \mathbb{R}^{H' \times W' \times D}$  is the output from SSD model (fringe ROI detection result) and  $Y_{\text{text}} = f_{YOLOv4}(X) \in \mathbb{R}^{H'' \times W'' \times D}$  is the output from YOLOv4 model (text detection result), respectively.

During training, the goal was to minimize the total loss  $L_{\text{total}}$  with respect to the parameters of both  $f_{SSD}$  and  $f_{YOLOv4}$ . The optimization problem can be formulated as:

$$\min_{\theta_{SSD}, \theta_{YOLOv4}} L_{\text{total}},$$

where  $\theta_{SSD}$  are the parameters of  $f_{SSD}$  and  $\theta_{YOLOv4}$  are the parameters of  $f_{YOLOv4}$ .  $\theta_{SSD}$  represents all the weights and biases that the SSD model uses to learn how to detect objects (fringes) in the input sample images. Similarly,  $\theta_{YOLOv4}$  presents all the weights, biases, and normalization parameters that the YOLOv4 model uses to learn how to detect objects (texts) in the input images. These parameters are optimised during training to minimize the loss function and improve the model's performance.

The total loss for the ensemble network can be defined as:

$$L_{\text{total}} = \lambda_1 \mathcal{L}_{SSD}(Y_{\text{ROI}}, Y_{\text{ROI}}^{\text{true}}) + \lambda_2 \mathcal{L}_{YOLOv4}(Y_{\text{text}}, Y_{\text{text}}^{\text{true}})$$

where  $\lambda_1, \lambda_2 \in \mathbb{R}$  are weights that balance the importance

of each loss component,  $Y_{\text{ROI}}^{\text{true}}$  is the ground truth ROI map, and  $Y_{\text{text}}^{\text{true}}$  is the ground truth text map. By specifying the SSD and YOLOv4 models in this formulation, we ensure that each model contributes effectively to the final detection task, leveraging their strengths within the ensemble framework.

### 3.3 Training Strategy and Parameters of Networks

Firstly, the proposed DENN model is trained using the datasets obtained in Section 2.2. The purpose of training the DENN model is to confirm its reliability in segmenting the ROI on computer-generated fringe sample images. The training input consisted of 515 computer-generated fringe sample images with ground truth information of the ROI and the texts on the samples. Anchor boxes were manually defined according to the size and aspect ratio of the ROI and the texts, as shown in Figure 6.

During the training process, the learning rate is adjusted automatically after each epoch. The model is trained for a set number of epochs (15, 25, 30, 50, 60, 70). After 70 epochs, our DENN model was fully trained, with the entire training process taking approximately 4 hours. Their selection involves the balance between performance and computational time costs. Each training batch consists of one sample, with the training set comprising 515 groups of sample images and the verification set containing 120 groups of sample images (not seen by the model). A 5-fold cross-validation is adopted to create the data models and estimate the model's efficiency. The data (80% of the total) is split into 3 parts: training on 3, testing on 2, and utilising all combinations of train-test splits. The process is repeated 7 times to obtain a more accurate estimate. After obtaining the trained network model, the test fringe sample images were input, and the proposed DENN recognized the subject optical element as the ROI and the label text on the test samples.

Even though our presented neural network-based solution was trained on the simulated fringe patterns (with fixed phase function and varying carrier fringes period), it was expected to work on the experimentally recorded data. For this purpose, we applied our DENN to the experimentally recorded sample images. The training input for the model consisted of 30 experimental sample images. To compensate for the limited number of experimental samples, these 30 images were augmented to create a final dataset of 2010 images, each modified differently through data augmentation. To increase robustness and prevent high bias in the model, 20% of the total input sample images were used as validation data (unseen by the model).

We adopted a similar training strategy for our DENN model, which was used with the computer-generated fringe sample images, involving adjustments to the learning

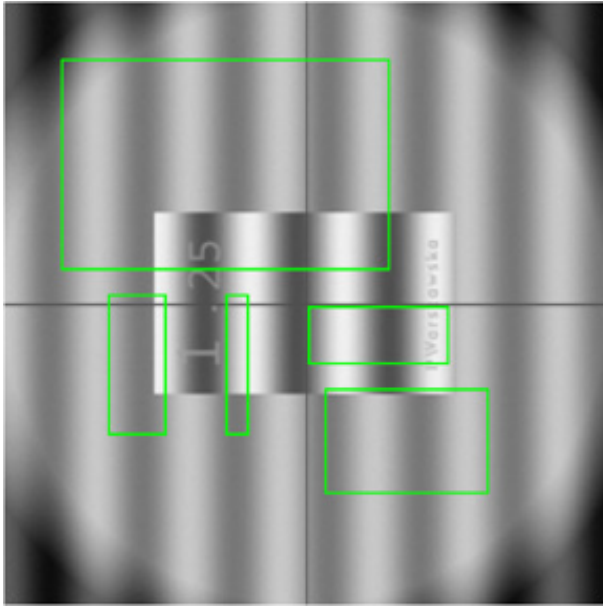


Fig. 6. Computer-generated fringe sample image with the predefined anchor boxes prepared for DENN, represented as green rectangles. The positions are only for this representation and do not correspond to the way they are used by the detector.

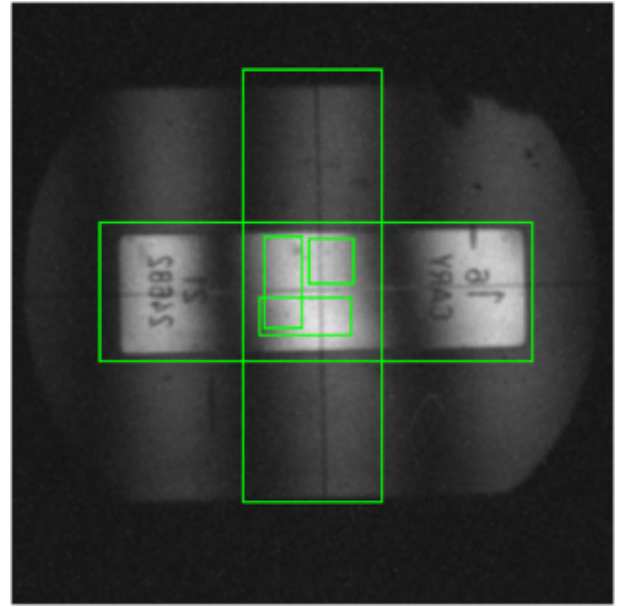


Fig. 7. The experimental test sample image with the predefined anchor boxes prepared for DENN, represented as green rectangles. The positions are only for this representation and do not correspond to the way they are used by the detector.

rate after each epoch and training across 6 different epoch sets. Additionally, we applied 5-fold cross-validation to obtain a more accurate estimate. A set of anchor boxes was adapted to match the characteristics of the predefined experimental samples, as shown in Figure 7.

In our presented DENN network structure, the base pretrained YOLOv4 model was trained using images with dimensions of  $256 \times 256 \times 1$ . The initial learning rate was set to 0.001 with a decay rate of 0.9, and L2 regularization was set to 0.0005. The training was carried out with a mini-batch size of 5 and the Adam optimizer [23]. During the training of YOLOv4, the input images were resized to ensure uniform dimensions. The anchor boxes with aspect ratios were set with the following combinations: {35:20, 36:18, 35:16, 22:22, 24:20, 20:19, 19:17, 20:15, 16:15}.

On the other hand, the pre-trained SSD network in our DENN was modified by adding additional layers. The modified SSD incorporates 50 convolutional layers, 1 max-pooling layer, followed by 43 batch normalization layers, 47 ReLU activation functions [1], and 13 combinations that fuse and aggregate feature maps from different layers to improve detection performance. The network was trained using the sample images with the same dimensions as those used for training the YOLOv4 model. The initial learning rate was set to 0.0001 with a momentum of 0.9, and the Stochastic Gradient Descent with Momentum (SGDM) optimizer was imposed [12]. Similar to YOLOv4, the training was carried out with a mini-batch size of 5. The anchor boxes with aspect ratios were set to the following combinations: {60:30, 60:21, 42:30, 111:60, 111:42,

78:60, 162:111, 162:64, 115:111, 213:162, 256:213, 187:213}.

The network structure examined in this paper was implemented using MATLAB programming language and the framework of MATLAB Deep Learning Toolbox on a PC with Intel®Core™i7-7820X CPU@3.60GHz×13, and the GeForce GTX 1650 (NVIDIA) is used to accelerate the computation.

#### 4. Experiments and results

To quantitatively and qualitatively determine the decision-making support of the presented DENN for recognition of ROI and texts under the scenario of fringe projection profilometry, we analyse two different scenarios: (a) data test - giving the performance accuracy related to the properties of the training sample images, (b) model test - gives the performance of our approach comparing it with other baseline algorithms.

In our experiment, to demonstrate the success of the DENN recognition, 28 different sets of fringe patterns were collected from the Kösters interferometer used for measurement of the central length deviation of the gauge block. Upon evaluating the reliability of the trained model, we applied the binary scores (see Figure 8). That is, a table that visualizes the performance of a model by categorizing each pixel of an image into one of the following conditions:

- True Positive (TP) – area of ROI specified by ground truth that is correctly detected as ROI by the model.
- False Positive (FP) – area of background that is incorrectly detected as ROI by the model.



Fig. 8. Example result of detection of the sample object as ROI. Each pixel in the image is categorized to be in one of the four conditions: True positive, true negative, false positive, and false negative.

- False Negative (FN) – area of ROI specified by ground truth that is incorrectly classified as background by the model.
- True Negative (TN) – area of the background that is correctly classified as background by the model

The following metrics were used to evaluate the model's performance: precision (Pr) =  $\frac{TP}{TP+FP}$ , recall (RE) =  $\frac{TP}{TP+FN}$ , accuracy (AC) =  $\frac{TP+TN}{TP+TN+FP+FN}$ , dice coefficient (DC) =  $\frac{2TP}{2TP+FP+FN}$ , and Jaccard index (JI) =  $\frac{TP}{TP+FP+FN}$ . The dice coefficient is the harmonic mean of precision and recall. It represents the size relation between the prediction and

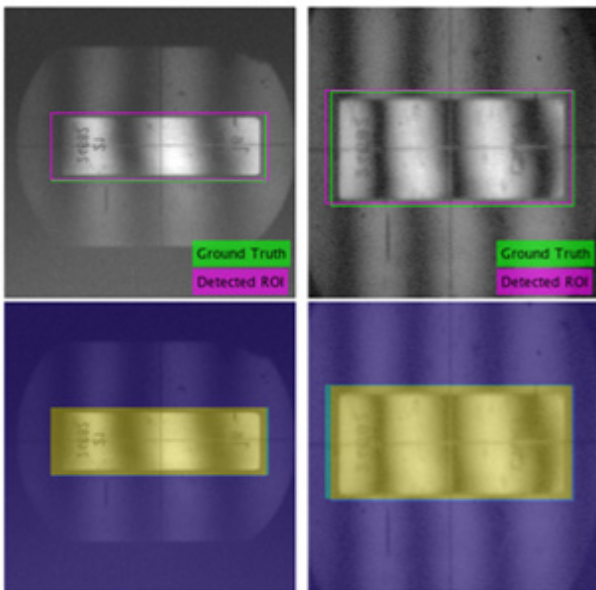


Fig. 9. Example result of detection of the sample object as ROI on the experimental sample images. The ground truth is marked with bounding boxes in green font, and the detected results are marked with bounding boxes in pink font.

Model	Accuracy	DiceCoefficient	JaccardIndex
SSD(9)	<b>0.9840</b>	<b>0.9533</b>	<b>0.9126</b>
YOLOv4(18)	0.2950	0.2353	0.1994
FasterRCNN(13)	0.9389	0.8683	0.7827
FastRCNN(4)	0.9292	0.7989	0.6769

Table 1. Detection of sample fringes (ROI) from 515 computer-generated sample images.

the ground truth, which may not be comprehensible from the precision or recall value alone. The Jaccard index, also known as the intersection over union (IoU), measures the ratio of the area correctly detected as ROI to the sum of the area identified as ROI by the ground truth and the area predicted as ROI by the model.

Additionally, we performed the assessment of Root Mean Square Error (RMSE) for each individual model. The RMSE formula for the ensemble model (proposed DENN) is expressed by:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{Y}_{ensemble,i} - Y_{true,i})^2}$$

where  $N$  represents the total number of sample images,  $\hat{Y}_{ensemble,i}$  is the predicted output for the  $i$ -th sample, and  $Y_{true,i}$  is the true output for the  $i$ -th sample.

**(a) Data test.** Each individual trained model (i.e. SSD and YOLOv4) was tested to detect the ROI and the texts from both computer-generated fringe sample images and experimental sample images. The test was conducted with a set of 515 computer-generated fringe samples and a set of 100 experimental samples. For each run, two pre-trained state-of-the-art (SOTA) deep learning models, Faster RCNN (13) and Fast RCNN (4), were also tested to provide a reference. The experimental results for each stage of the individual model employed in our DENN, in comparison to the SOTA methods, are presented in Tables 1, 2, 3, 4, respectively.

The performance of SSD is better for detecting sample fringes (ROI) in both computer-generated and experimentally recorded sample images, as shown in Tables 1 and 2. However, the YOLOv4 model performs worse for the same task, indicating that its feature extraction capabilities are insufficient for accurately recognizing ROI in fringe sample images. Fringe patterns often consist of regular, repeating lines or waves that might confuse the detection algorithm, leading it to misinterpret the ROI boundaries. Notably, the SOTA methods demonstrate the second and third-best results after the SSD model.

Figure 9 presents the detection results for the region of interest (ROI) within the experimental sample images using the SSD method. The green bounding boxes represent the ground truth ROI, determined through manual annotation or established reference data, while the pink bounding boxes highlight the ROI detected by the ensemble neural

network. The top row of Figure 9 shows the detection results on grayscale images, demonstrating the ability of the SSD method to identify the precise ROI boundaries in challenging lighting and structural conditions. The bottom row illustrates the same detections visualized using a color-heatmap representation, enhancing the visibility of the detected regions for improved interpretability.

It is evident from the figure that the detected ROI aligns closely with the ground truth in both cases, indicating the high accuracy of the detection. The model effectively handles variations in image intensity and fringe pattern complexities, ensuring reliable detection across different experimental samples. These results validate the robustness of the SSD method, particularly in scenarios where interference fringes

Model	Accuracy	DiceCoefficient	JaccardIndex
SSD(9)	<b>0.9806</b>	<b>0.9662</b>	<b>0.9348</b>
YOLOv4(18)	0.1837	0.1653	0.1494
FasterRCNN(13)	0.8754	0.8189	0.6962
FastRCNN(4)	0.7711	0.5487	0.4024

Table 2. Detection of sample fringes (ROI) from 100 experimental recorded sample images.

are complex and partially overlapping, as often observed in Kösters interferometer measurements.

YOLOv4, on the other hand, performed better in detecting text in sample fringes. Tables 3 and 4 present model performances, where YOLOv4 outperforms the other state-of-the-art (SOTA) methods, followed by SSD for computer-generated fringes. However, the SSD network performs poorly in detecting text in experimental fringe sample images. Since SSD is designed to detect objects by looking at features at multiple scales, its local feature extraction capabilities are insufficient for the intricate and small-scale features that text often presents. YOLOv4 uses more sophisticated Feature Pyramid Networks (FPN) and Path Aggregation Networks (PAN) that better preserve high-resolution details and integrate features at multiple scales, making it more adept at detecting smaller and more intricate objects like text. Consistent with previous results, Fast R-CNN and Faster R-CNN rank second and third

Model	Accuracy	DiceCoefficient	JaccardIndex
SSD(9)	0.9769	0.5681	0.4032
YOLOv4(18)	<b>0.9877</b>	0.8280	0.7092
FasterRCNN(13)	0.9180	<b>0.8283</b>	<b>0.7579</b>
FastRCNN(4)	0.9213	0.7789	0.6980

Table 3. Detection of texts from 515 computer-generated sample images.

Model	Accuracy	DiceCoefficient	JaccardIndex
SSD(9)	0.3429	<b>0.0285</b>	<b>0.0165</b>
YOLOv4(18)	<b>0.9541</b>	0.3749	0.3060
FasterRCNN(13)	0.7244	0.2830	0.1579
FastRCNN(4)	0.7197	0.2176	0.1550

Table 4. Detection of texts from 100 experimental recorded sample images.

for experimental sample images, and third and fourth for computer-generated fringes.

Figure 10 illustrates the detection of text regions on experimental sample images using the strength of the YOLOv4 method. The green bounding boxes denote the ground truth regions of the text, as identified through manual annotation, while the pink bounding boxes highlight the text regions detected by the network. The top row of Figure 10 displays the detection results on grayscale images, where the network successfully identifies the text within the sample objects, even under conditions with challenging contrasts and variations in illumination. The bottom row provides the corresponding visualizations using a color-heatmap overlay, which emphasizes the detected text regions.

The results demonstrate that the detected text regions closely align with the ground truth, validating the accuracy of the YOLOv4 network in handling text detection tasks. The method effectively addresses challenges posed by the interference fringe patterns surrounding the text and variations in image quality, ensuring consistent and reliable detection performance.

**(b) Model test.** It is not possible to compare the different

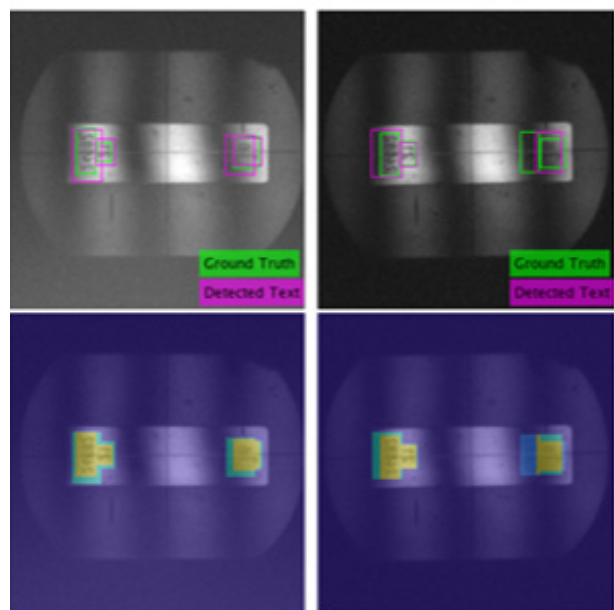


Fig. 10. Example result of detection of the texts on the experimental sample images. The ground truth is marked with bounding boxes in green font, and the detected results are marked with bounding boxes in pink font.

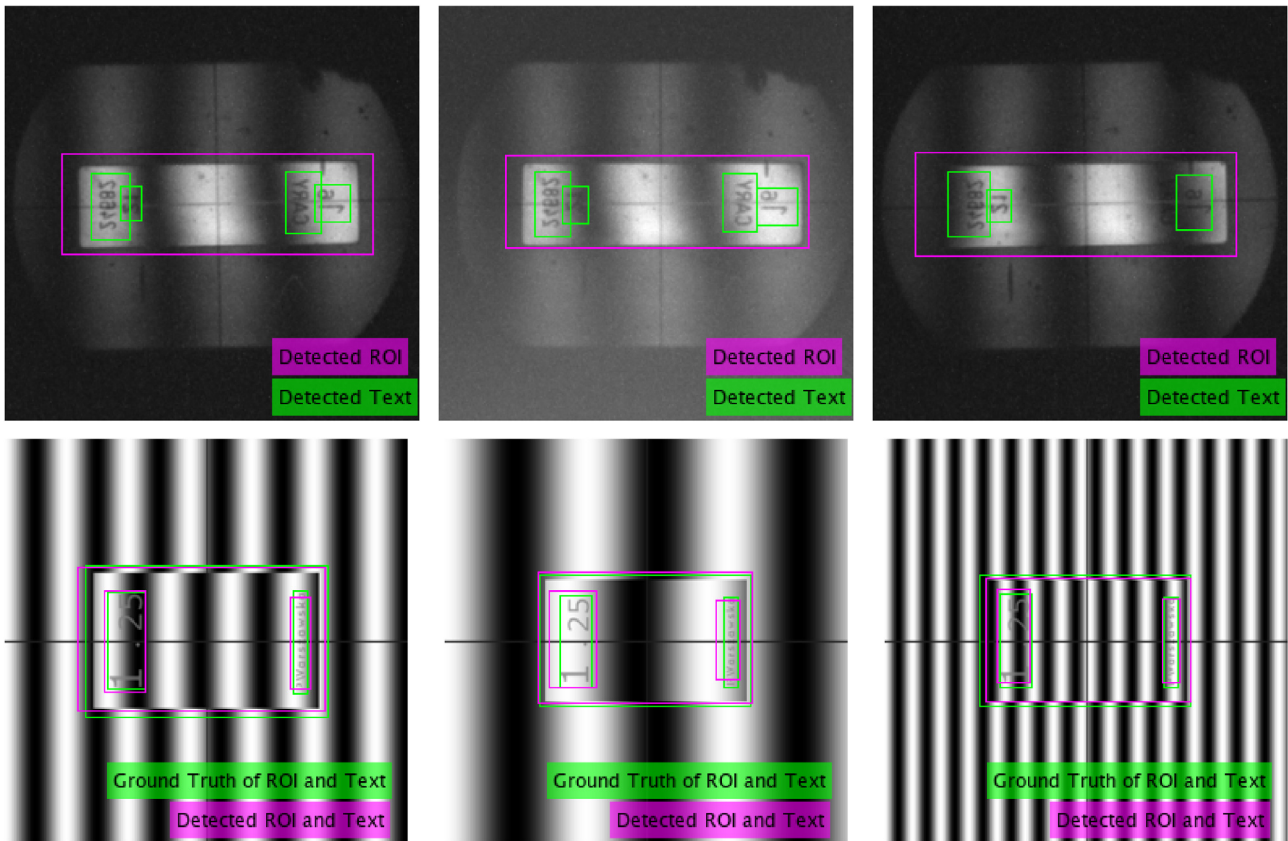


Fig. 11. Example detection of different sample images of fringe pattern characteristics (both experimental & computer-generated samples). In the first row, the detected ROIs are marked with pink bounding boxes, followed by the texts, which are detected by green bounding boxes. In the second row, the ground truth is marked with bounding boxes in green font, and the detection results are marked with bounding boxes in pink font.

deep learning methods for fringe segmentation, as the implementation of the methods described in the literature needs a lot of work. In addition, not all details of the implementation are disclosed. That is why we make the comparison by considering only the state-of-the-art available pre-trained networks. To conduct this study, we compared our DENN with the methods using (a) pre-trained fast RCNN and (b) pre-trained faster RCNN, as they are employed in data tests.

Fast R-CNN (4) integrates the object detection process

Model	Accuracy	RMSE	Executiontime(s)
FasterRCNN(13)	0.8450	0.19	<b>0.0209</b>
FastRCNN(4)	0.7419	0.23	0.8455
DENN(ours)	<b>0.9159</b>	<b>0.11</b>	0.0592

Table 5. Performance comparison of other baseline algorithms against our method on the experimental dataset. The experimental sample images incorporate different sets of fringe patterns with texts.

into a single streamlined network, optimizing both performance and accuracy. It can be used effectively to recognize ROIs and texts in fringe samples by leveraging its robust feature extraction, region proposal, and classification capabilities.

By fine-tuning the network to the specific characteristics of fringe patterns and text, it is possible to achieve the detection and localization of fringes in these complex images. Faster R-CNN (13) utilizes the object detection pipeline by incorporating a Region Proposal Network (PN) that shares convolutional features of the full image with the detection network. This results in a significant speedup and improves the detection accuracy.

Table 5 presents the performance comparison. Accuracy and RMSE were used as evaluation indicators. We achieved good performance in detecting the ROI and texts in fringe sample images. However, the time efficiency experiment showed that our combined model performed poorly compared to Faster RCNN due to the execution capabilities of each individual model before combining their outputs into an ensemble network. The achieved performance is compared to those reported in (4), (13) (Table 5). Additionally, we reported a performance matrix (i.e. confusion matrix) which summarizes performance for both object types together (ROI vs not-ROI and Text vs not-Text). Each cell counts how many objects fall into that category (see Fig. 12). The best accuracy reported is 91.59% as shown in Table 5. Time efficiency was evaluated by measuring the run-time for executing the detection on 150 sample images and then

		Predicted Class	
		ROI	Text
Actual Class	ROI	138	12
	Text	13	137

Fig. 12. Confusion matrix representing the overall object detection performance of the proposed DENN model on 150 recorded experimental samples. Each sample contains both ROI and text bounding box ground truth annotations. The achieved detection reflects the model's ability to correctly detect both ROI and text regions in the combined output.

taking the average runtime per image. Examples of fringe detection (ROI) and text separation using the DENN approach are presented in Figure 11, where it is evident that the network successfully detected different sets of fringe pattern characteristics.

## 5. Concluding remarks

The measurement of the central length deviation of a gauge block depends on many factors, including the environmental conditions (temperature, atmospheric pressure, and relative humidity) and the quality of the calculated phase distributions obtained from the interferograms. In this work, we address the second problem and present an advanced image pre-processing method that leverages the strengths of a deep ensemble neural network. We demonstrate how this deep learning-based approach significantly improves the accuracy of detecting interference fringes nested within other fringes of different phases. Unlike existing standalone deep learning approaches (i.e. single-model approach), the proposed ensemble framework integrates two state-of-the-art deep neural models - SSD and YOLO. The strength of the proposed ensemble method is that it combines the predictions of multiple models, leveraging their strengths and compensating for their weaknesses. This combination allows the system to capture both fine details and broader patterns in the interference fringes, leading to more accurate phase distribution detection. The effectiveness of the proposed method has been verified using simulated fringe patterns and experimental data from the Kösters interferometer. We believe that, after appropriate training with different types of data, the proposed network framework should also be applicable to other forms of fringe patterns (e.g.

exponential phase fringe patterns or closed fringe patterns) and other phase measurement techniques.

## 6. Acknowledgement

This work was prepared within the project "Development and implementation of modern algorithms for the analysis of fringe images at interferometer systems for measuring length deviations and surface microgeometry at the Central Office of Measures", contract number PM/SP/0008/2021/1, financed by the Ministry of Education and Science (Poland) as part of the Polish Metrology Programme.

## References

- [1] K. Eckle and J. Schmidt-Hieber. A comparison of deep networks with relu activation function and linear spline-type methods. *Neural Networks*, 110:232–242, 2019.
- [2] S. Feng, Q. Chen, G. Gu, T. Tao, L. Zhang, Y. Hu, W. Yin, and C. Zuo. Fringe pattern analysis using deep learning. *Advanced photonics*, 1(2):025001–025001, 2019.
- [3] S. Feng, L. Zhang, C. Zuo, T. Tao, Q. Chen, and G. Gu. High dynamic range 3d measurements with fringe projection profilometry: a review. *Measurement Science and Technology*, 29(12):122001, 2018.
- [4] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [5] K. Harding. *Handbook of optical dimensional metrology*. CRC Press, 2013.
- [6] N. Jackson. *Principles of interferometry. Jets from Young Stars II: Clues from High Angular Resolution Observations*, pages 193–218, 2008.
- [7] E. K. Kim, H. Lee, J. Y. Kim, and S. Kim. Data augmentation method by applying color perturbation of inverse psnr and geometric transformations for object recognition based on deep learning. *Applied Sciences*, 10(11):3755, 2020.
- [8] H. Liu, N. Yan, B. Shao, S. Yuan, and X. Zhang. Deep learning in fringe projection: a review. *Neurocomputing*, page 127493, 2024.
- [9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

- [10] G. Michalecki. Automatic calibration of gauge blocks measured by optical interferometry. *Science Review*, 1:15–17, 2001.
- [11] D. Pandey, J. Ramaiah, S. Ajithaprasad, and R. Gannavarpu. Subspace analysis based machine learning method for automated defect detection from fringe patterns. *Optik*, 270:170026, 2022.
- [12] S. Postalcioglu. Performance analysis of different optimizers for deep learning-based image recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(02):2051003, 2020.
- [13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
- [14] A. M. Roy, R. Bose, and J. Bhaduri. A fast accurate fine-grain object detection model based on yolov4 deep neural network. *Neural Computing and Applications*, 34(5):3895–3921, 2022.
- [15] U. Schnars, C. Falldorf, J. Watson, W. Jüptner, U. Schnars, C. Falldorf, J. Watson, and W. Jüptner. *Digital holography*. Springer, 2015.
- [16] M. Servin, J. A. Quiroga, J. M. Padilla, et al. *Fringe pattern analysis for optical metrology*. Wiley Online Library, 2023.
- [17] A. Vishnoi, A. Madipadaga, S. Ajithaprasad, and R. Gannavarpu. Automated defect identification from carrier fringe patterns using wignerville distribution and a machine learning-based method. *Applied Optics*, 60(15):4391–4397, 2021.
- [18] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 13029–13038, 2021.
- [19] X. Wang and J. Song. Iciou: Improved loss based on complete intersection over union for bounding box regression. *IEEE Access*, 9:105686–105695, 2021.
- [20] M. Wengierow, L. Salbut, Z. Ramotowski, and R. Szumski. Measurement system based on multi-wavelength interferometry for long gauge block calibration. *Metrology Journal*, 15:234–245, 2020.
- [21] Y. Xiao, E. Decencièrre, S. Velasco-Forero, H. Burdin, T. Bornschlöggl, F. Bernerd, E. Warrick, and T. Baldeweck. *A new color augmentation method for deep learning segmentation of histological images*. In *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*, pages 886–890. IEEE, 2019.
- [22] J. Xu and S. Zhang. Status, challenges, and future perspectives of fringe projection profilometry. *Optics and Lasers in Engineering*, 135:106193, 2020.
- [23] Z. Zhang. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pages 1–2. IEEE, 2018.
- [24] Y. Zhu, L. Liu, Z. Luan, and J. Sun. Discussions about fft-based two-step phase-shifting algorithm. *Optik*, 119(9):424–428, 2008.
- [25] C. Zuo, S. Feng, L. Huang, T. Tao, W. Yin, and Q. Chen. Phase shifting algorithms for fringe projection profilometry: A review. *Optics and lasers in engineering*, 109:23–59, 2018.