scientific reports

Check for updates

OPEN

Computer vision and deep transfer learning for automatic gauge reading detection

Hitesh Ninama[1,8], Jagdish Raikwal[1], Ananda Ravuri[2], Deepak Sukheja[3], Sourav Kumar Bhoi[4], N. Z. Jhanjhi[5]✉, Asma Abbas Hassan Elnour[6] & Abdelzahir Abdelmaboud[7]

This manuscript proposes an automatic reading detection system for an analogue gauge using a combination of deep learning, machine learning, and image processing. The study suggests image-processing techniques in manual analogue gauge reading that include generating readings for the image to provide supervised data to address difficulties in unsupervised data in gauges and to achieve better accuracy using DenseNet 169 compared to other approaches. The model uses artificial intelligence to automate reading detection using deep transfer learning models like DenseNet 169, InceptionNet V3, and VGG19. The models were trained using 1011 labeled pictures, 9 classes, and readings from 0 to 8. The VGG19 model exhibits a high training precision of 97.00% but a comparatively lower testing precision of 75.00%, indicating the possibility of overfitting. On the other hand, InceptionNet V3 demonstrates consistent precision across both datasets, but DenseNet 169 surpasses other models in terms of precision and generalization capabilities.

Keywords Gauge detection, Computer vision, Deep learning, DenseNet 169, VGG19, InceptionNet V3

Automation increases productivity, accuracy, and quality of processes and products while reducing human intervention in repetitive and tedious jobs[1]. Automated devices that replicate human capacities through vision are useful outside medicine in various fields, including business, farming, inventory management, controlling product traceability, and production process monitoring[2,3]. Usually, a cloud-based system is connected to the automated devices. Being accessible to authorized users, it can be a useful tool for extracting pertinent information from the environment and interacting with a wide range of devices[4] in real-time[5], as the Internet of Things (IoT) has become essential for communication among devices over the internet. Instrumentation is an extremely important area in the scientific and engineering fields as well as in workstations. It entails consistent and accurate readings of a variety of parameters to produce warnings in the event of unexpected incidents. There are still a number of utility industries that emphasize the use of analog gauges as the cost of reinstalling new digital gauges is too high. Professional drivers of racing cars prefer analog speedometers as they are simple to view and provide an accurate impression of the actual speeds. Specialized scientific instruments known as analog gauges may measure a wide range of physical quantities, such as pressure, tension, speed, etc[6]. In analog gauges, the pointer or needle moves around the dial in a circular motion, stopping at scale marks that are directly proportional to the value being measured[7]. Even though they are ancient, analog gauges are still widely used in various types of businesses.

However, in manufacturing sectors, employees still rely on analog gauges to read physical process value systems such as pressure, temperature, and voltage[8]. Digital gauges that can communicate using wired or wireless connections are preferable. However, upgrading them is not always possible due to high cost and technical limitations. The advantages of analog gauges include their robust anti-interference capability, resistance to water, dust, freezing temperatures, and straightforward design. This topic of research is crucial for creating automated techniques for in-field analog gauge monitoring. Analog gauge data collection needs manual examination, which consumes time, is prone to inaccuracy, and may be dangerous at workstations. As a result, engineers have worked

[1]Institute of Engineering and Technology, Devi Ahilya University, Indore, M.P. 452001, India. [2]Senior Software Engineer, Intel Corporation, Hillsboro, OR 97006, USA. [3]Department of CSE, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad 500090, India. [4]Department of Computer Science and Engineering, Parala Maharaja Engineering College (Govt.), Berhampur, Odisha 761003, India. [5]School of Computer Science, SCS Taylor's University, 47500 Subang Jaya, Malaysia. [6]Computer Science Department, Community College- Girls Section, King Khalid University, 62529 Muhayel Aseer, Saudi Arabia. [7]Humanities Research Center, Sultan Qaboos University, Muscat, Oman. [8]School of Computer Science and Information Technology, Devi Ahilya University, Indore, M.P. 452001, India. ✉email: Noorzaman.jhanjhi@taylors.edu.my

hard to develop automated techniques for reading analog gauges[9]. By installing such systems, the cost of labor, the danger of accidents, and the chance of mistakes may be reduced. One approach to resolving this problem is by using the vital area of Artificial Intelligence (AI) techniques. Computer vision teaches computers how to interpret and distinguish between images[10,11]. A low-cost solution to analyze attempts to comprehend digital streams and monitor the environment to avoid emergency situations is to install still recording devices on workstations[12].

The goal of this research is to use a convolutional neural network based on deep transfer learning to recognize analog gauges in real-time and forecast readings.

Literature review

A number of methods have been suggested to automate reading in analogue gauges for different functional sectors, like the insurance industry and automobile applications[13]. The methods initially perform optically flow-based video stabilization, image processing, and execute segmentation using the HSV color space and morphology operations. The dial region and pointer are then calculated using the Hough transform. Trigonometry is used to calculate an angle produced with the lowest valued scaling mark using the pointer's position in the frame's coordinates, and the result is displayed in the form of text and a progress bar. The method proposed using Analog Gauge Readers (AGR) is developed within the PyCharm environment and deployed using Python programming and open-source libraries like OpenCV, NumPy, and PIL. The experimental findings demonstrate that the technique may be used in several types of analog gauges recorded under various lighting conditions and that adding picture preprocessing significantly reduces inaccuracy. In the experimental investigations, the algorithms' accuracy and precision ranged from 99 to 100%.

Early research[14] suggested using an ESP32-CAM microcontroller and an OV2640 camera with a 65° field of view to capture gauge images, process them on a local computer, and show the results in a web-accessible dashboard. This was accomplished by using the CenterNet HourGlass104 model and convolutional neural network (CNN) to detect the gauge. The pointer angle is determined using pixel projection once the dial has been recognized and segmented using the circular Hough transform. Finally, the angle approach is used to calculate the indicative value with a dataset made of 204-gauge pictures that were divided between training and test groups using a 70:30 ratio. Owing to the small dataset size, numerous improvements were suggested to produce a gauge identification model with acceptable accuracy and a variety of applications. Despite the overall average inaccuracy of 0.95%, the experimental results also show better precision and robustness for industrial applications.

A mobile phone-based system for real-time analog gauge transcription using a convolutional neural network (CNN) system was introduced, which operates efficiently on both high and low-resolution gauge images. A large synthetic image dataset for training and a new real-world dataset for testing were provided. This method showed state-of-the-art performance with pointer angle error less than 1 degree and operates at up to 25 frames per second on mobile devices[15].

In their widely acclaimed work[7], a pipeline based on image processing for automatic pointer movement translation and recognition in analog circular dials was discussed. The suggested method employs a module-based approach to process an input video frame-by-frame. A bilateral filter is used to reduce noise in each image before applying a Gaussian mean level for segmentation of objects. The objects are then classified using probability distributions calculated using Expectation Maximization and described by a set of proposed features. The Mahalanobis distance is used to categorize the pointer, and PCA is used to ascertain its angle. Based on temporal pointer angle estimates, a low-pass filtered electronic time series is the output. The algorithm has successfully processed seven test videos, yielding encouraging results. This contributes to our understanding of the use of inspection robots to monitor crucial equipment and instruments in high-voltage environments. A data station receives the captured images, where they are manually examined, finding gauges in images that have been captured frequently for automatic analysis. The proposed algorithm for gauge detection is based on the geometric fitting approach, where Sobel filters are used for the first time to extract the edges, which may include the gauge shapes. The proposed methodology suggests removing straight lines that don't belong to gauges by using line fitting within the context of random sample consensus (RANSAC). In order to find the most fitted ellipse from the remaining edge points, the RANSAC ellipse fitting is used. The experimental outcomes on a real-world dataset acquired by GuoZi Robotics show that our algorithm delivers precise gauge detection accuracy compared to a number of other methods.

Motivation

The literature review highlights various methods developed to automate the reading of analogue gauges for different functional sectors, such as industrial and automotive applications. The existing methods discussed in the literature involve optical flow-based video stabilization, image processing, segmentation using the HSV color space and morphology operations, and the Hough transform for calculating the dial region and pointer. The proposed methodology aims to detect a circle, line, needle, and center point along with the final angle of the needle and its current value. This enhances the estimation of the current reading of the gauge in an image and labels it for training neural networks using computer vision techniques and deep transfer learning models. The goal is to improve the accuracy and efficiency of gauge detection and readings, particularly in industrial environments where manual inspection may not be feasible or safe.

Methodology

Implementation of the proposed methodology was initiated with the detection of a circle, line, needle, and center point, along with the final angle of the needle, to estimate the current reading of a gauge in an image. These are for the purpose of labeling, essential for training the neural networks. Without them, it would not be possible

to use the neural networks for training. Once the gauge readings are obtained, the images are saved into specific folders using a copy-cut-paste method, and the names of these folders are used as labels.

In our work, the detection of the circle, line, needle, center point with the final angle of the needle and current value is used to estimate the current reading of the gauge in an image to label the first phase of work based on a computer vision library such as OpenCV. The second phase commences with training deep transfer learning models using generated image data. Pandas' data frames were then used to select features and labels[16]. The final phase used the TensorFlow library to train deep learning models, evaluate performance, and perform prediction. The proposed methodology is divided into two sections: the first obtains the current reading using computer vision, OpenCV, and image processing techniques, and the second phase uses to train transfer learning models over generated data with labeled images.

The proposed calibration method primarily aims to refine the accuracy of gauge readings through a multi-phase process. Initially, it involves the detection of critical elements like circles, lines, needles, and center points, crucial for labeling and training neural networks. Accurate labeling is pivotal as it enables the training of transfer learning models, enhancing their ability to interpret and predict gauge data effectively. The methodology integrates computer vision, specifically OpenCV, in the initial phase and transitions to deep transfer learning in the second phase. Pandas' data frames are employed for feature and label selection, streamlining the calibration process for more precise and reliable gauge readings in various applications.

In this paper, the proposed DenseNet169 model with image preprocessing methods shows various hyper-parameters to implement models as listed in Table 1. The hardware required for implementation is an Intel i7 processor, 32 GB of RAM, RTX 3070 DDR 6 8 GB graphics card for neural network training, Python language for implementation, and Jupyter notebook.

Dataset Description

This research utilizes two publicly accessible datasets[17], viz. SyntheticGauges and RealGauges. The SyntheticGauges dataset includes 10,000 training and 1,000 test images of synthetically rendered gauges, all at a resolution of 1024x1024 pixels, aimed at training Convolutional Neural Networks (CNNs). Accompanying JSON files provide ground truth labels in COCO format, detailing gauge bounding boxes, keypoints for perspective points, and gauge parameters such as scale minimum and maximum, pointer center, and tip.

Conversely, the RealGauges dataset, designed to evaluate gauge transcription systems, features real images of six different gauges. It addresses three tasks: gauge detection, where gauges are set against 36 varied backgrounds; pose recovery, assessing gauges from multiple camera angles; and gauge reading, documented in 3 distinct 5-second videos per gauge, showing varied pointer movements. Labels include gauge readings and pointer angles, offering a comprehensive benchmark for gauge transcription system evaluation. Fig.1 displays an overview photo illustrating samples from the dataset, featuring six different types of gauges.

Phase one

In this phase, the data is first collected from[17], which consists of various random gauge images with continuous reading videos. Next, the video of one meter has been selected and converted into individual frames to get manual readings for labeling the images. Finally, these images are saved with high resolution after implementing Hough Circles[18] and drawing circles and the centers of circles. Further division of these circles with points of lines using cos and sin separation at 10-degree angles with 360/10 intervals has been performed.

Briefly, in this phase, after separating the points using the inner and outer circles, the next step is to put a line and some labels on top of it, and then to finally obtain the first calibration images. The intersection of Eqs. (1) and (2) represents the point at which one calculates lines over circles.

$$P(x)_{i,j} = x + 0.9 * r * cos\frac{((s * i * 3.14)}{180} \tag{1}$$

Parameter name	Details
Model	Sequential
Neural network	VGG19, Inception Net v3, Dense Net 169
Layers	Dense, Dropout, Output, Conv2D
Neurons	3,256,9
Optimizer	Adam
Activation function	Softmax, Relu
Learning rate	0.001
Loss	Categorical Cross Entropy
Matrices	Precision, Recall and F1 score
Epochs	100
Batch	14 each step

Table 1. Hyper parameters used to train the models.

Fig. 1. Overview of Dataset Samples (http://www.jjcvision.com/projects/gauge_reading.html).

$$P(y)_{i,j} = y + 0.9 * r * cos\frac{((s * j * 3.14)}{180} \tag{2}$$

where, P(x) and P(y) are the points of the line at the separation point of the circle with radius r, S is the separation of circles with a 10-degree angle, and i,j are rows and columns of image pixels. This model includes three parameters, two of which are for the circle's center and one for the radius. The number of degrees of freedom and consequently the necessary size of the parameter space is constrained if the gradient angle for edges is known. The gradient angle determines the direction of the vector from the circle's center to each of its edge points, leaving the radius value as the only unknowable factor. Equation (3) represents the Hough transformation of a circle where a and b are lines and r is the radius:

$$Hough\ Transform\ Circle = (x-a)^2 + (y-b)^2 - r^2 \tag{3}$$

After the completion of this procedure, a calibration image was obtained. The original image is depicted in Fig. 2a, and the calibration image is shown in Fig. 2b. Finally, the coordinates of the line were obtained from the list of final lines as the process generates N more needle lines than a brute-force approach would. Finding the angle using the arc tangent of angle y divided by angle x.

$$x_{angle} = x_2 - x \tag{4}$$

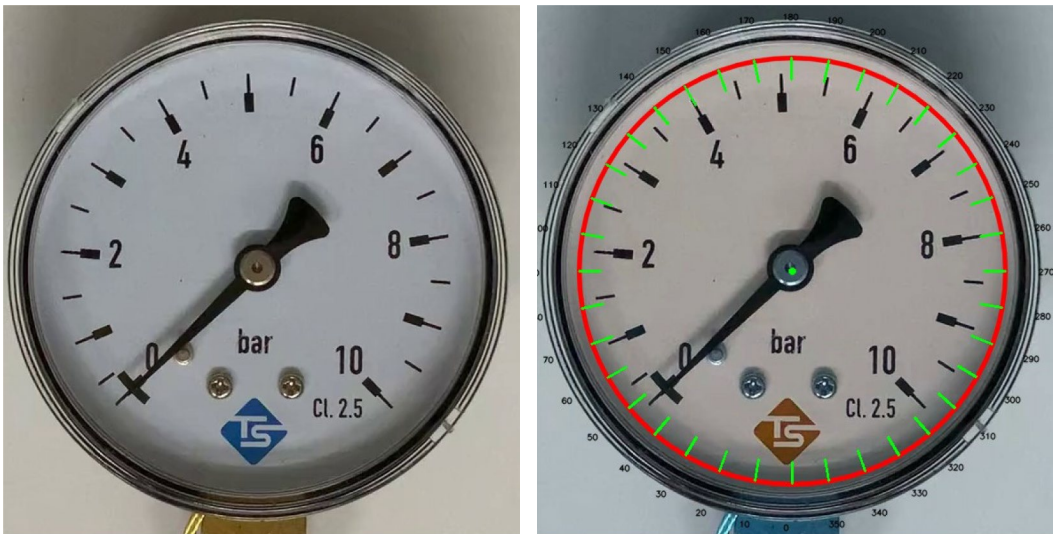
$$y_{angle} = y_2 - y \tag{5}$$

$$arc = arc\_\tan\tan\frac{\mathrm{y_{angle}}}{x_{angle}} \tag{6}$$

After calculating the tangent of the arc using the division of y and x angles, calculate the quadrant of the circle to get the final angle. Equations (7), (8), (9), and (10) show the quadrant calculations:

$$1_{st}Quadrant = x_{angle} > 0 and y_{angle} > 0, final_{angle} = 270 - arc \tag{7}$$

$$2_{nd}Quadrant = x_{angle} > 0 and y_{angle} > 0, final_{angle} = 90 - arc \tag{8}$$



(a) Original gauge **(b)** Calibrated gauge.

Fig. 2. (**a**) Original gauge (**b**) Calibrated gauge.

$$3_{rd}Quadrant = x_{angle} > 0 and y_{angle} > 0, final_{angle} = 90 - arc \tag{9}$$

$$4_{th}Quadrant = x_{angle} > 0 and y_{angle} > 0, final_{angle} = 270 - arc \tag{10}$$

Finally, import the image and set the minimum angle, maximum angle, minimum value of the gauge, and maximum value of the gauge with an image threshold with lower and upper bound limits 0.15, 0.25 for the center coefficient and 0.5, 1.0 for the circle coefficient. After completing phase one, the process generates readings for all images and saves them into labeled folders ranging from 0 to 9. The total number of classes is 9.

Phase Two

This phase is used in the same way that deep transfer learning works[19]. In order to categorize images and predict their properties based on past data, deep learning is essential. In this work, models for gauge reading prediction and gauge detection are used. Supervised training data is obtained using processed visual data; produced by open image datasets and computer vision techniques with circle, line, and angle identification. An artificial neural network based on convolutional neural networks is employed using the transfer learning concept since it was not feasible to use labeled data directly. Three distinct models, such as VGG19[20], Inception V3, and DenseNet169[21], were used to address the above-mentioned problem. Over the ImageNet dataset, these models have already undergone pre-training. Before training, the data is preprocessed by creating additional images with various angles, heights, and widths while incorporating the padding notion. Then, as illustrated in Table 2, a Pandas data frame of information is established, that includes the directory structure, gauge ID, and gauge type.

Data Preprocessing

Data preprocessing alters the data format so that data mining, machine learning, and other data science tasks can process it quickly and efficiently. Data cleansing and removal of superfluous data are the goals of data preparation. All images and the data frame are converted into a particular height and width in this section. There are 1011 images and 9 classes of gauge reading, ranging from 0 to 9. The height and width of the training images are 128 pixels. The following hyperparameters[22] are used to create a batch of images using Image Data Generator: rotational range, width shift, height shift, zooming range, horizontal flip, and vertical flip. The pixel division

S. No	File	Gauge ID	Gauge type
0	4/frame414.png	4	4
1	4/frame410.png	4	4
2	3/frame1223.png	5	3
3	4/frame1231.png	4	4
4	4/frame1242.png	4	4

Table 2. Data frame of images with gauge type.

technique is used to normalize the images. The final graphics include three color channels, a 1011-pixel height and width, and 128 pixels on each side.

Transfer learning and various models

Transfer learning refers to the practice of applying a machine learning method that has been pre-trained to solve a different but related problem. If a simple classifier has been trained to detect the presence of a backpack in an image, it may be applied to detect other objects, such as glasses. Transfer learning involves the use of what you've learned in one task to improve another. The main idea is to apply the weights that a network had learned at "task A" to a new "task B". Rather than starting from scratch, we use patterns from similar activities. Transfer learning is widely used in machine learning and computer vision data processing, such as sentiment analysis, and is an active learning "design philosophy," but not a machine learning specialty. Combining it with neural networks, which require a lot of data and computing power, is now drawing attention. The implementation of our method uses models such as VGG19, InceptionNet V3, and DenseNet169.

VGG19. There are 19 layers in the VGG19 convolutional neural network where you may load a network that has been trained to use more than one million photographs from the database. As it includes 19 layers, the goal is to use these layers to achieve satisfactory data outcomes[20].

Nowadays, convolutional neural networks (CNNs) are used in imaging and computer vision applications for a variety of tasks, e.g., segmentation, classification, and recognition of images. Input layers, hidden layers, and output layers are fundamental elements of simple convolutional neural networks (CNNs). Convolutional layers (CL), pooling layers (PL), and fully connected layers (FCL) are common examples of hidden units (Transfer Learning). Nineteen layers make up the flexible VGG19 neural network convolutional (CNN). Specifically, there are sixteen Convolutional Layers (CL), five Pooling Layers (PL), three Fully Connected Layers (FCL), and one SoftMax (SM) Layer. VGG19's framework is constructed over the course of six stages. The first step is to place the image into the framework; typically, a picture with the coordinates (224, 224, 3) is used. The underlying image structure is then exposed by employing a size-dependent kernel (3, 3)[16]. A fixed-coefficient linearizer (FCL) is used to convert the linear layer output to a non-linear one, as linear output is the norm for the layers. Last but not least, the SoftMax activation layer has been used here. The use of the SoftMax function as a straightforward activation function inside a neural network model is the most widespread use of applied machine learning. Since there are two classes, a SoftMax layer has been used here. The complex patterns of VGG19 are easy to understand and implement, and pre-trained versions learned on larger datasets like ImageNet are more popular than those learned from scratch[23].

Dense net. Despite having a depth of 169 layers, DenseNet-169 is chosen as it has fewer parameters than some other models[21]. It is a design that can address the disappearing gradient issue, requires fewer arguments and runs faster than other methods.

The DenseNet-169 model exists[24,25] and belongs to the DenseNet organization and deviates substantially from the DenseNet- 121 baseline in terms of size and accuracy. The larger DenseNet-169 model is easily distinguishable from the smaller DenseNet- 121 model. To "train" each DenseNet model, the ImageNet database has been utilized that needs a single 1x3x224x224 image in BGR order as the input to run this model. Before the image blob can be transmitted over the network, the average BGR values must be subtracted as shown here: [103.94, 116.78, 123.68]. Additionally, a rounding-off error of 0.017 must be corrected. The results of the DenseNet-169 model are typical of object classifiers for all 1000 categories in ImageNet[6].

Inception v3. The accuracy of picture datasets is increased using the image identification model Inception v3. The model is the result of a variety of ideas that were developed over the course of a year with several studies. This improves the image and allows us to identify it[7].

A substantially higher accuracy of 78.1% on the ImageNet dataset has been demonstrated with the Inception v3 image processing model. The model was created with a number of concepts that various researchers studied over a long period. Convolutions, average pooling, maximum pooling, concatenations, dropout, and layers that are entirely connected are some of the symmetric and asymmetric building blocks of the Inception v3 model. The batch normalization method is applied to activate input in the model and is heavily used. Using the Softmax method, the loss is calculated.

The suggested steps in the methodology are shown in Fig. 3, which include gathering data, identifying circles, lines, and quadrants, dividing them into circles, computing angles, obtaining manual readings for each image, preprocessing the images, implementing neural networks, and dividing the data into 80/20 portions for testing and training. Thereafter, deep transfer learning techniques are used to learn from these trainable images.

Results

Experimental tests and results analysis are essential to evaluate the performance of a proposed methodology. In this paper, the proposed methodology consists of two phases; the first phase involves generating labels for all images using an image processing and computer vision library, and the second phase involves the training of the deep neural network for generating readings of input gauges. To implement the deep neural network, TensorFlow has been used with the Keras library and OpenCV for processing images. This paper implements three neural networks: VGG19, InceptionNetV3, and DenseNet169, for comparison with the existing and proposed models.

The implemented models are trained on the processed data after the picture data has been preprocessed, extracting characteristics from the images, followed by learning through neurons and weight adjustments with some bias. The models' starting learning rate was 0.001, whereas during the training process, the learning rate

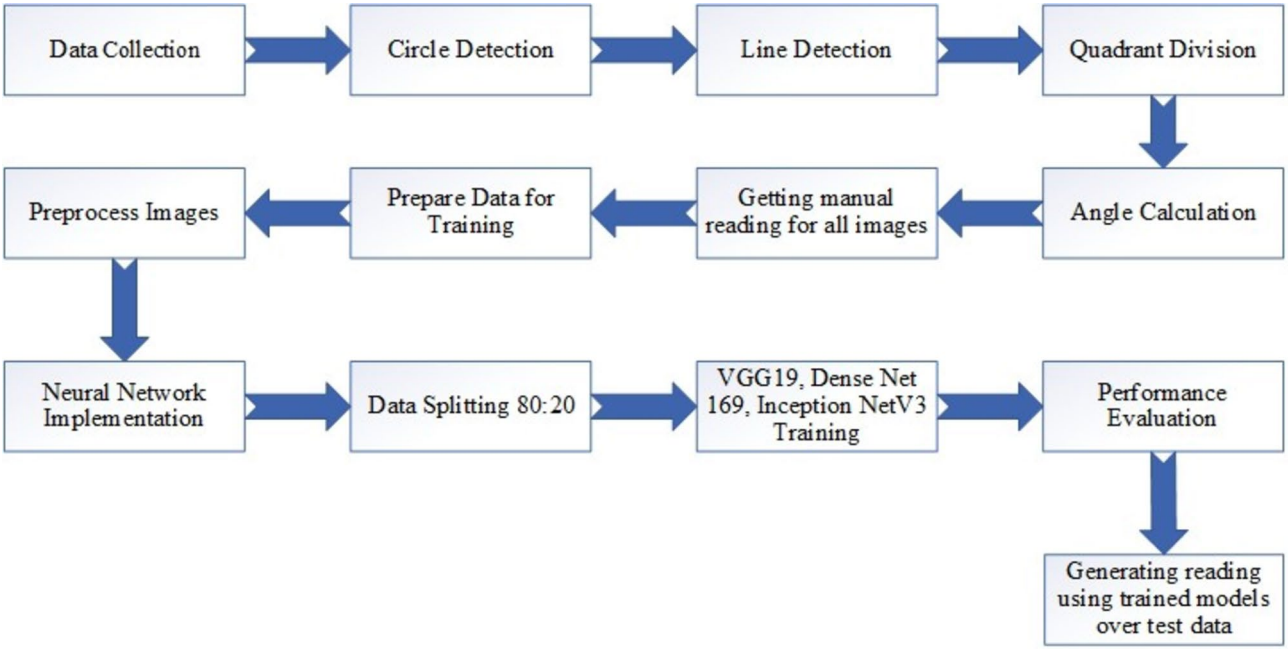


Fig. 3. Flowchart for the proposed approach.

decreased. The graphs shown in Figs. 4, 5, and 6 display the precision, recall, and F1 score graphs for trained models that use more than 80% of the data.

The model demonstrates high accuracy in detecting needle positions, showcasing its proficiency. However, comparisons with human readings reveal some discrepancies, particularly in nuanced variations. The evaluation of the results confirms that the model is competent, but it also highlights the ongoing challenge of achieving human-level precision in interpreting analog gauges.

Table 3 presents a complete summary of the performance metrics, namely precision, recall, and F1 score, for three widely used deep learning models: VGG19, InceptionNetV3, and DenseNet169. The DenseNet169 model is employed for classification purposes. The VGG19 model exhibits a commendable level of precision during the training phase, reaching 97.00%. However, it displays a comparatively lower precision rate of 75.00% on the testing data. This observation implies the possibility of overfitting, wherein the model excessively conforms to the training data and encounters difficulties in generalizing to novel cases. In contrast, the InceptionNetV3 model demonstrates consistent high precision, recall, and F1 scores throughout both the training and testing datasets. This suggests that the model is robust in accurately categorizing positive cases and is capable of generalizing to unfamiliar data. DenseNet169 demonstrates exceptional performance across all criteria. The model demonstrates outstanding precision, recall, and F1 scores on both the training and testing datasets, exhibiting low performance degradation between the two environments. This characteristic is indicative of its robust generalization abilities. The exceptional performance of DenseNet169 indicates its efficacy in accurately categorizing positive cases while retaining a high level of generalization capability. As a result, it is the recommended option among the assessed models for jobs that demand dependable and accurate classification. Figure 7 shows the gauge reading predicted by the DenseNet169 model.

Overall, the experimental results indicate that the proposed methodology and DenseNet169 model with image preprocessing are effective for generating readings of input gauges. The results can be further improved by exploring different hyperparameters, architectures, and optimization techniques.

Discussion

Our proposed method shows significant improvements in gauge reading accuracy compared to existing approaches. For instance, a prior work[15] utilized a MobileNetV2-based CNN for real-time analogue gauge transcription on mobile devices, achieving a pointer angle error of less than 1 degree. Their approach involves training on a large synthetic dataset (10,000 images) and achieving high accuracy and performance metrics in gauge detection, gauge reading, and perspective recovery.

While effective, their method primarily relies on synthetic data for training, which may not fully capture the variations present in real-world scenarios. Additionally, their system operates efficiently on both high and low-resolution gauge images, running up to 25 frames per second on mobile devices.

In contrast, our approach leverages deep transfer learning models, specifically DenseNet169, InceptionV3, and VGG19, which are trained on both synthetic and real-world datasets. Our method demonstrates high precision and generalization capabilities, particularly with DenseNet169 outperforming others in terms of precision, recall, and F1 score. The DenseNet169 model achieves exceptional performance across all criteria, with low performance degradation between training and testing environments, indicating its robust generalization abilities.

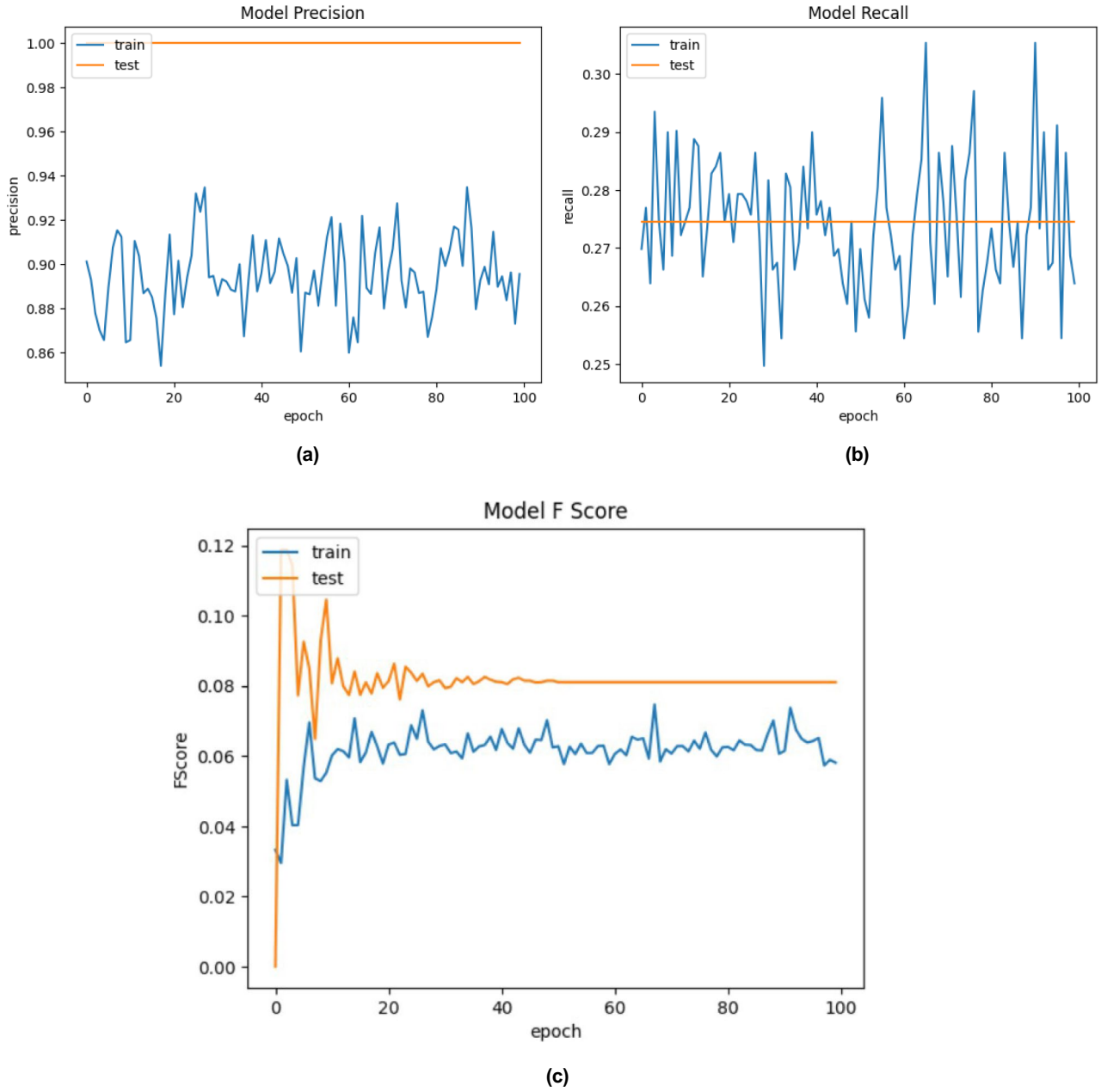


Fig. 4. Precision, recall and F1 score Graph of VGG19.

Moreover, our system addresses the potential overfitting observed in models like VGG19, which exhibits high training precision but lower testing precision. By incorporating a comprehensive dataset and advanced image preprocessing techniques, our approach ensures accurate and reliable gauge reading, even in varied and challenging real-world conditions.

Overall, the superior performance of our proposed system, as evidenced by the higher precision, recall, and F1 scores, along with its ability to generalize well to unseen data, highlights its suitability for real-world applications where high accuracy and reliability are critical. This robust performance makes our method a valuable contribution to the field of automatic analogue gauge reading, offering significant advancements over existing methods.

Conclusion

This paper proposed an automatic reading detection system for analogue gauges. The study used three neural network models for comparative analysis, and the proposed algorithm outperformed the competition with improved accuracy during training and testing. The study also offers details on the hardware used for implementation and compares the performance metrics, such as precision, recall, and F1 score, of the trained models. Performance metrics for VGG19, Inception Net V3, and Dense Net 169 are presented. VGG19 shows strong training precision (97.00%) but lower testing precision (75.00%), suggesting potential overfitting. In contrast, Inception Net V3 maintains high precision across both sets, while Dense Net 169 outperforms the others with exceptional

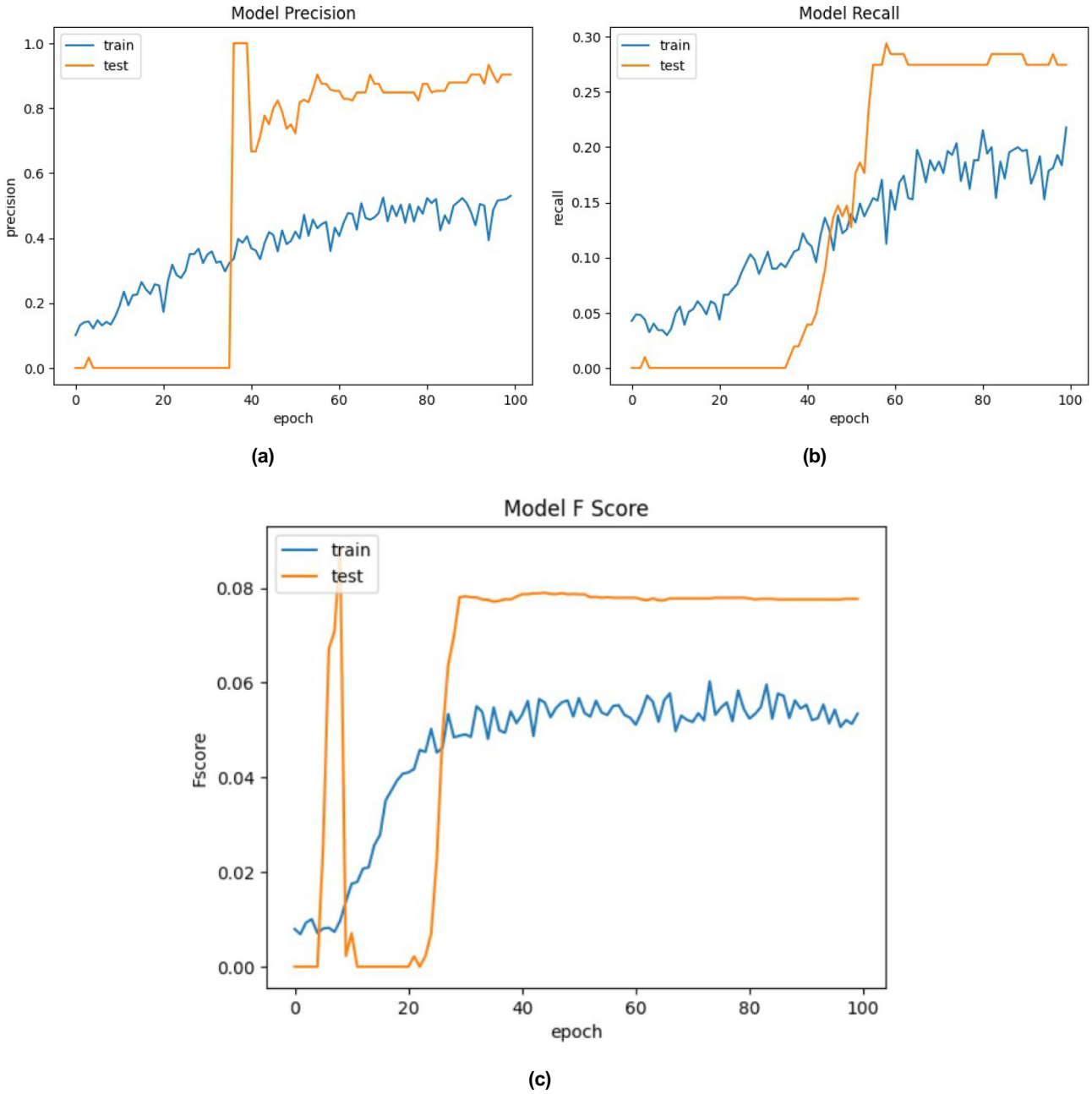


Fig. 5. Precision, recall and F1 score Graph of Inception Net V3.

precision and generalization. Overall, the proposed system has the potential to improve efficiency and safety in facilities that use analog gauges by automating the reading process and reducing the need for manual checking.

Future work

As the average inaccuracy nearly doubles under these circumstances, the results also demonstrate that poor lighting and tilted gauges have a detrimental effect on the algorithm's effectiveness. The solution, however, demonstrated that it was accurate enough for the majority of noncritical applications. The robustness study also showed that the solution has high adaptability to a wide range of gamma, contrast, and brightness values. The only exception was Gaussian noise, which significantly reduced the image quality. As a result, the algorithm's performance is degraded. The viability of this technique in a practical application was also proved by a proof-of-concept machine vision implementation. The small dataset, which may contain individual biases and adversely impacts gauge detection even with data augmentation, is a shortcoming of the method. Furthermore, the final answer depends on external data that may be directly derived from the gauge image. However, these limitations will be investigated in further studies.

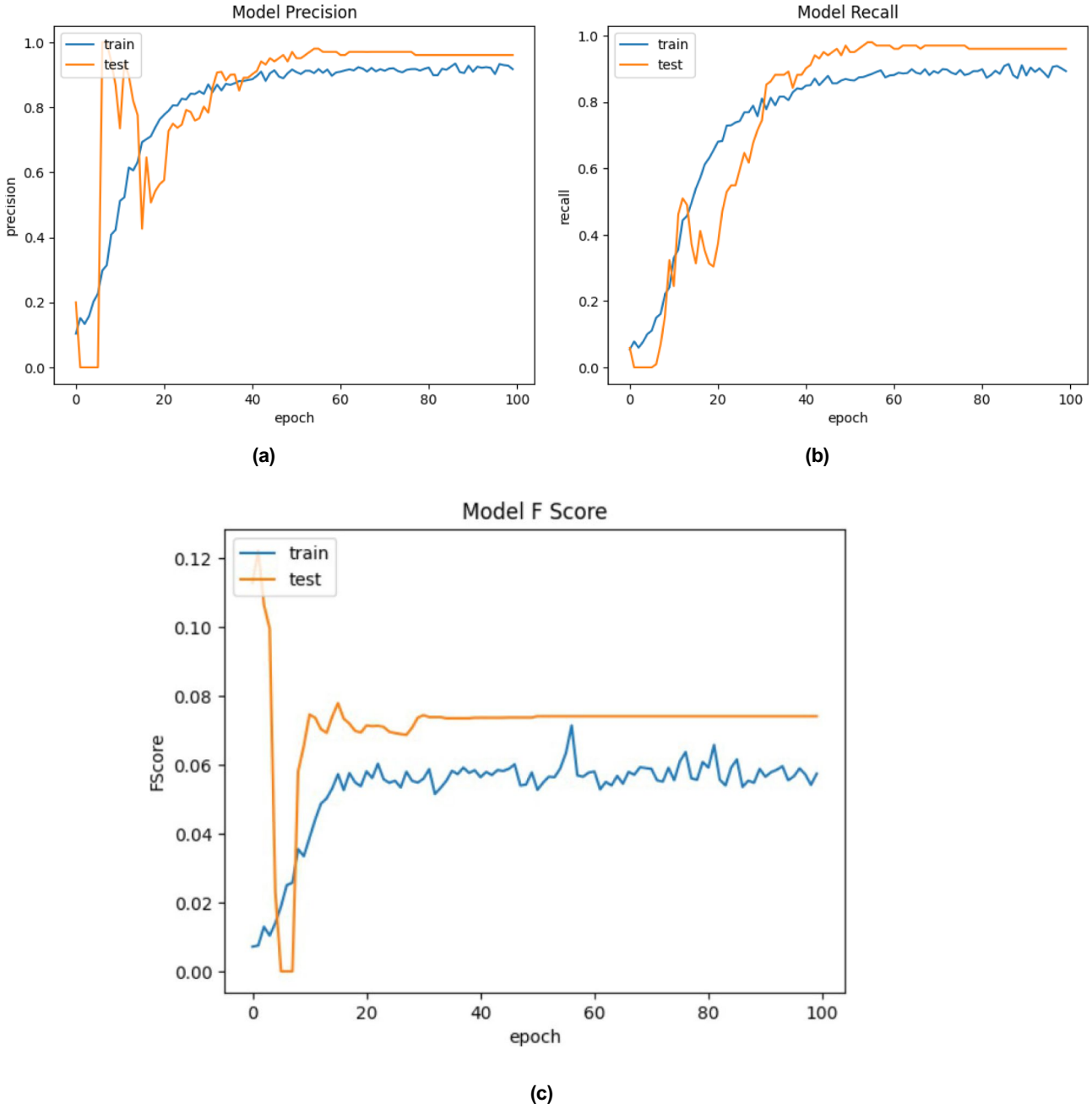


Fig. 6. Precision, recall and F1 score Graph of Dense Net 169.

Model	Train precision	Test precision	Train recall	Test recall	Train F1 score	Test F1 score
VGG19	97.00	75.00	92.23	90.33	97.17	94.87
Inception Net V3	98.76	95.00	94.34	91.45	96.65	94.32
Dense Net 169	99.34	98.01	99.47	97.09	99.62	97.61

Table 3. Comparison table over the used architectures.

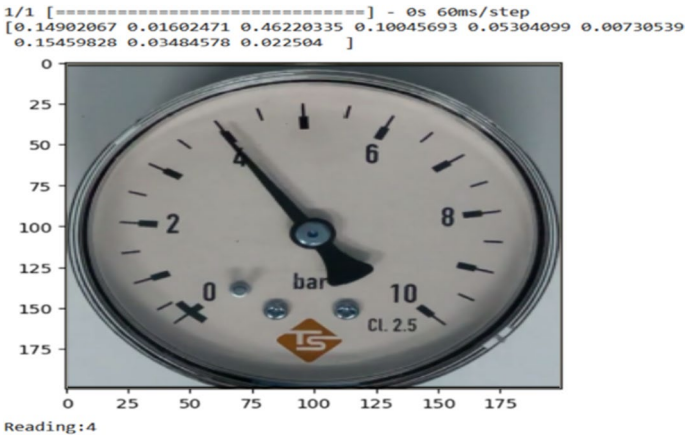


Fig. 7. Gauge reading predicted by Dense Net 169.

Data availability

Data will be available upon request from the designated contact author.

Received: 14 June 2023; Accepted: 26 August 2024
Published online: 03 October 2024

References

1. Chi, J., Liu, L., Liu, J., Jiang, Z. & Zhang, G. Machine vision based automatic detection method of indicating values of a pointer gauge. *Math. Probl. Eng.* https://doi.org/10.1155/2015/283629 (2015).
2. Russakovsky, O. *et al.* Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* **115**, 211–252 (2015).
3. Mohan, C. R. *et al.* Improved procedure for multi-focus images using image fusion with qshiftn dtcwt and mpca in laplacian pyramid domain. *Appl. Sci.* **12**, 9495 (2022).
4. Sowah, R. R., Ofoli, A. R., Mensah-Ananoo, E., Mills, G. A. & Koumadi, K. M. An intelligent instrument reader: using computer vision and machine learning to automate meter reading. *IEEE Ind. Appl. Mag.* **27**, 45–56 (2021).
5. Tang, W.-B. *et al.* A channel rail gauge detection system of modern tram. In *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, 1–5 (IEEE, 2018).
6. Li, B., Yang, J., Zeng, X., Yue, H. & Xiang, W. Automatic gauge detection via geometric fitting for safety inspection. *IEEE Access* **7**, 87042–87048 (2019).
7. Lauridsen, J. S. *et al.* Reading circular analogue gauges using digital image processing. in *14th International joint conference on computer vision, imaging and computer graphics theory and applications (Visigrapp 2019)*, 373–382 (SCITEPRESS Digital Library, 2019).
8. Lin, Y., Zhong, Q. & Sun, H. A pointer type instrument intelligent reading system design based on convolutional neural networks. *Front. Phys.* **8**, 618917 (2020).
9. Zhang, X., Dang, X., Lv, Q. & Liu, S. A pointer meter recognition algorithm based on deep learning. in *2020 3rd International conference on advanced electronic materials, computers and software engineering (AEMCSE)*, 283–287 (IEEE, 2020).
10. Salomon, G., Laroca, R. & Menotti, D. Deep learning for image-based automatic dial meter reading: Dataset and baselines. in *2020 International joint conference on neural networks (IJCNN)*, 1–8 (IEEE, 2020).
11. Madhu, G. *et al.* Imperative dynamic routing between capsules network for malaria classification. *CMC-Comput. Mater. & Continua* **68**, 903–919 (2021).
12. Zuo, L., He, P., Zhang, C. & Zhang, Z. A robust approach to reading recognition of pointer meters based on improved mask-rcnn. *Neurocomputing* **388**, 90–101 (2020).
13. Chavan, S., Yu, X. & Saniie, J. High precision analog gauge reader using optical flow and computer vision. in *2022 IEEE International conference on electro information technology (eIT)*, 171–175 (IEEE, 2022).
14. Peixoto, J. *et al.* Development of an analog gauge reading solution based on computer vision and deep learning for an iot application. In *Telecom*, vol. 3, 564–580 (MDPI, 2022).
15. Howells, B., Charles, J. & Cipolla, R. Real-time analogue gauge transcription on mobile phone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2369–2377 (2021).
16. L, S. & Lakshmi, K. An analysis of convolution neural network for image classification using different models. *Int. J. Eng. Res. Technol. (IJERT)* 1–7 (2020).
17. Howells, B., Charles, J. & Cipolla, R. Real-time analogue gauge transcription on mobile phone. http://www.jjcvision.com/projects/gauge_reading.html (accessed Jan. 06, 2023). (2021).
18. Opencv:hough circle transform. https://docs.opencv.org/3.4/d4/d70/tutorial_hough_circle.html (accessed Jan. 06, 2023).
19. Iqbal, A., Basit, A., Ali, I., Babar, J. & Ullah, I. Automated meter reading detection using inception with single shot multi-box detector. *Intell. Autom. Soft Comput.* https://doi.org/10.32604/iasc.2021.014250 (2021).
20. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
21. Huang, G. Dense connected convolutional neural networks. in *IEEE Computer society conference on computer vision and pattern recognition (CVPR)* (2017).
22. Liao, L., Li, H., Shang, W. & Ma, L. An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* **31**, 1–40 (2022).
23. Krishna, M. M., Neelima, M., Harshali, M. & Rao, M. V. G. Image classification using deep learning. *Int. J. Eng. Technol.* 614–617 (2018).
24. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826, https://doi.org/10.1109/CVPR.2016.308 (2016).

25. Openvino™ toolkit documentation densenet-169. https://docs.openvino.ai/2021.1/omz_models_public_densenet_169_densenet_169.html (accessed Jan. 06, 2023).

Author contributions
H.N, J.R, N.Z conceived the idea, H.N, J.R., and D.S conducted the experiment and wrote the paper, H.N., D.S, A.A., A.A and S.K analysed the results, N.Z supervised the project. All authors reviewed the manuscript.

Funding
The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/319/45.

Competing interests
The authors declare no competing interests.

Human and animal rights
This article does not contain any studies with human participants or animals performed by any of the authors.

Additional information
Correspondence and requests for materials should be addressed to N.Z.J.