Application of One-Shot Recognition Using Siamese Convolutional Neural Networks for Steel Surface Defect Detection

Gokul Soman, *Student Member, 22209582,* Basil Paul, *Student Member, 22210325,* and Anand Gopinatha Pillai, *Student Member, 22201634*

***Abstract*—In order to evaluate the dissimilarity of surface flaws in steel, this case study presents a novel use of Siamese Neural Networks, providing a reliable and effective method for quality control in industrial settings. The suggested method improves on the conventional defect identification procedure by automatically identifying and quantifying differences in steel surfaces by utilizing deep learning. Moreover, a user-friendly interface of industrial image analysis has been developed to enable smooth integration into real-world settings. This interface facilitates the easy entry of photos and offers clear visualizations of the network's output. By learning discriminative features from pairs of defect photos, the Siamese Neural Network architecture can properly quantify the dissimilarity between distinct defect cases. The training procedure of the model is tuned to capture subtle differences in defect shape, guaranteeing increased sensitivity to surface abnormalities. A user interface that makes it easier to input photographs for analysis has been designed in order to improve practical use. Workers with differing degrees of technical proficiency can utilize the system because of this interface, which makes it simple for operators to upload and process photographs. Furthermore, the interface presents output data in an understandable way, facilitating prompt and well-informed decision-making during quality assessment. The demand for effective, precise, and easily accessible defect analysis in the steel production industry is addressed by the integration of a Siamese Neural Network with an intuitive user interface. The study establishes the foundation for more extensive applications in the field of industrial image analysis and shows how cutting-edge deep learning approaches can transform quality control procedures.**

***Index Terms*—Siamese neural network, CNN, steel surface, surface defects, dissimilarity**

I. Introduction

MATERIAL quality management is crucial in the field of industrial manufacturing, particularly in industries like steel manufacture. Steel surface flaws can have a significant impact on a product's overall performance and structural integrity. Conventional defect detection techniques frequently rely on manual inspection[1][2], which is time-consuming, labor-intensive, and prone to human mistake. This case study investigates the use of a Siamese Neural Network, a potent deep learning architecture, to assess the dissimilarity of surface defects in steel in order to address these issues.

Siamese Neural Nets offer a possible path toward automated defect diagnosis because of its propensity for picking up on minute feature variations. The network can identify complex differences in surface imperfections by training on pairs of defect photos, which helps to produce a more accurate and nuanced defect evaluation. The study prioritizes usability over technical concerns, using an intuitive interface to make the technology accessible to operators with different levels of technical proficiency.

The goal of this program is to simplify the application of cutting-edge neural network technology in practical industrial settings and to increase the accuracy of fault identification. Surface defect analysis in steel manufacturing could undergo a revolution thanks to the combination of an advanced Siamese Neural Network with an easy-to-use user interface. This would offer a flexible and scalable solution for quality control procedures. By demonstrating the effectiveness of this coordinated strategy, this case study hopes to pave the way for deeper uses of deep learning in the field of industrial picture analysis.

This case study expands upon the groundbreaking work presented in the paper titled "One-Shot Recognition of Manufacturing Defects in Steel Surface"[3] by Aditya M. Deshpande, Ali A. Minaia, and Manish Kumara in response to the urgent need for advanced defect detection methodologies in the steel manufacturing industry. Acknowledging the importance of their suggested remedy, our endeavor centers on putting into practice and expanding the system described in the paper. In addition, we present an intuitive user interface to promote smooth operation of the technology with the goal of improving its usefulness in industrial environments.

Our objective of attaining high-efficiency defect analysis is in line with the original paper's emphasis on one-shot recognition of faults in steel surfaces. Through the application of the suggested methodology, which most likely makes use of siamese neural network topologies, our goal is to further improve the accuracy of defect detection in practical production settings. Moreover, we have included a user-friendly interface into the system because we understand how important accessibility and practical usability are. By guaranteeing that operators with differing degrees of technical proficiency can enter photographs and comprehend the network's output with ease, this improvement makes the technology more accessible and useful for industrial quality control procedures.

Our case study aims to confirm and increase the effectiveness of the suggested system by putting the established paper into practice and adding a user interface. By doing this, we hope to set the foundation for a wider adoption of deep learning techniques in the field of industrial image analysis by showcasing not just the original solution's theoretical capability but also its adaptability and efficiency in practical applications.

II. LITERATURE REVIEW

The combination of Industry 4.0 and the Industrial Internet of Things (IIoT) has brought to an unprecedented increase of sensory data in the context of modern production. This increase originates from a variety of causes found in industrial facilities, including labour activities, product lines, environmental factors, and machinery and procedures. The abundance and intricacy of this data pose a difficulty as well as an opportunity for intelligent production, where data-driven choices are essential.[4]

Machine learning has become a revolutionary tool that is revolutionising manufacturing data interpretation. In particular, breakthroughs in computer vision and deep learning stand out as particularly potent instruments that have transformed methods for data analysis[5]. According to this paradigm, it is not only feasible but also essential for modern industrial processes to be able to derive useful insights from complex information.

The relationship between Industry 4.0, IIoT, and the use of machine learning techniques is explored in this literature review, with an emphasis on deep learning and computer vision. The study centres on the use of siamese neural networks for one-shot manufacturing fault identification, with the goal of highlighting knowledge gaps and offering an overview of previous studies. Highlights the significant influence that these technologies have on the industrial environment by looking at how they may be used to solve difficult problems including feature extraction[6], anomaly detection[7], object tracking[8], and object detection[9][10]. The focus is on their use in vision-based inspection, a nondestructive assessment method that is becoming more and more popular in manufacturing quality control.

Computer vision is being used for purposes other than manufacturing flaws, such as identifying fractures and other damage to concrete surfaces[11]. Previous research using conventional image processing[12] methods prepared the way for more current developments. A surpasses conventional techniques like Sobel and Canny edge detection by introducing a strong strategy that uses deep CNNs for fracture classification on concrete surfaces[4].

In navigating the complex field of automated vision systems for industrial quality control, this literature review aims to summarise current understanding, evaluate developments critically, and identify areas that require more research[13]. We want to uncover the story[14] of automated vision systems in manufacturing, highlighting their potential to improve inspection procedures without sacrificing product quality, by looking at the evolution from conventional approaches to modern deep learning techniques.[15]

One especially effective paradigm is the one-shot recognition approach that makes use of the Siamese neural network architecture.[6] When there is just one data sample[16] available for training, this method performs exceptionally well. The siamese network architecture has shown impressive performance in applications beyond conventional picture categorization.[17] It is intended to compare and identify similarities between input pairs. Applications such as picture segmentation, voice recognition[18], natural language processing[19], and drug development[5] are noteworthy.

Effective visual inspection is essential in the field of smart manufacturing to guarantee process integrity and product quality. Given that siamese network-based one-shot recognition may function well with little training data[2], it becomes especially attractive for use in visual inspection tasks. The specific application of this approach to the identification of steel surface defects—a job essential to quality control in manufacturing[20] processes—is the subject of this review of the literature.

The purpose of the literature review is to present the usefulness of Siamese network-based one-shot recognition in the identification of steel surface defects in order to provide empirical support. We provide a comparison study using conventional CNNs, which are commonly used in picture classification applications, and a simple one-shot learning method[21][22][23][24], the nearest-neighbor algorithm with a single neighbour, to contextualise its performance. By means of these analogies, we want to clarify the advantages and disadvantages of every methodology, providing discernments into their pragmatic suitability for visual inspection assignments in the framework of intelligent manufacturing.

Siamese network-based one-shot recognition[5], this literature review contributes to the discourse on efficient and data-efficient methodologies for visual inspection in smart manufacturing. The findings aim to inform researchers, practitioners, and decision-makers in the manufacturing sector, shedding light on the potential impact of these methodologies on the enhancement of quality control processes in the context of modern manufacturing environments.[10]

III. SIAMESE NEURAL NETWORK

An artificial neural network specifically made for tasks involving determining how similar or dissimilar two pairs of input data are is known as a Siamese neural network. Two identical subnetworks with similar weights and parameters make up the design, which was first described in the 2015 publication "siamese neural networks for one-shot image recognition"[16] by Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. The network can acquire representations that can efficiently identify similarities and differences between pairs of input examples because to this special siamese structure.

Pairs of input samples and labels indicating how similar or dissimilar the samples are are fed into a siamese neural network[3] during training as shown in Fig 1. The goal of the network is to maximize the distance between dissimilar pair representations and reduce it for comparable pair representations. The sub-networks acquire transferable representations thanks to the shared weight setting, which makes it easier for the network to identify complex linkages in the data. Siamese Networks use distance metric learning techniques widely in image verification, one-shot learning, and similarity-based classification applications. The difference in the output representations is measured using the Euclidean distance[3]

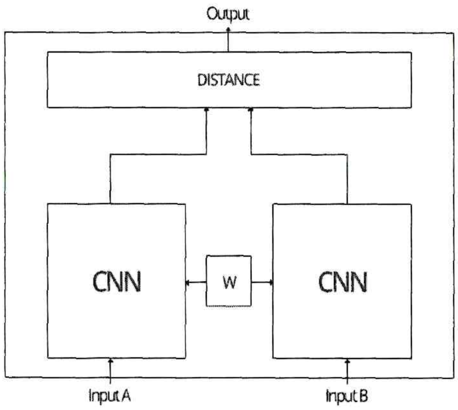

Fig. 1. Siamese neural architecture

or a comparable measure. The network is more successful at learning similarity metrics when contrastive loss or triplet loss functions are used for optimization.

Siamese neural networks are very useful in one-shot learning situations, as the network is highly effective at classifying new classes with few samples. The network creates a spatial arrangement in which similarity between embedding is indicated by closeness between them by mapping input instances into a shared embedding space. This embedding space enables effective comparison and retrieval of similar instances, demonstrating the network's flexibility and versatility across several domains, especially in tasks related to pattern recognition and computer vision.

IV. Proposed System Architecture

The proposed system architecture was taken from the reference paper[3] and it was implemented as it is. A brief explanation of the siamese neural network architecture is as follows:

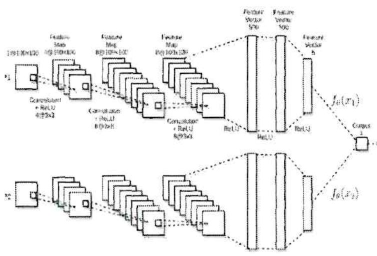

Fig. 2. Architecture of siamese neural network used in reference paper

1) Network Architecture
As seen in figure 2, the siamese network is composed of two identical modules, or branches, that share the same weights and design. A high-dimensional input[3], usually an image of size 100×100 pixels (N=100×100), is processed by each module, and it is compressed into a lower-dimensional encoded vector[3] of size $n = 5$. This architecture makes sure that both modules consistently learn and extract pertinent features from the incoming data, which makes the ensuing similarity comparisons easier.
2) Convolution Layer
Three convolutional layers[3] are used in each module to create feature maps of four, eight, and eight channels, respectively. Every layer's feature map dimensions stay at 100 by 100. The convolutional layers identify spatial hierarchies and patterns from the input images by performing a sliding operation with a 3×3 kernel size and a stride of 1. In order to extract features from the image data that are both local and global, hierarchical feature extraction is essential.
3) Activation Function
The output feature maps from every convolutional layer are subjected to Rectified Linear Unit (ReLU) activation functions. By setting all negative values to zero, this non-linear activation creates element-wise non-linearity and improves the network's learning of intricate representations. ReLU's[3] addition guarantees that the network can recognize complex relationships in the input data and improves feature learning.
4) Fully Connected Layers
Following the convolutional layers, three fully connected layers with sizes of 500, 500, and 5 are included in each module. Convolutional layers extract information, which is then processed further by fully linked layers to produce a more informative and condensed representation of the input. Convolutional layers provide a hierarchical abstraction that fully connected layers[3] use to extract key features and patterns.
5) Encoded Vectors
Each module's final result is an encoded vector of $n = 5$ sizes. These encoded vectors[3], represented as $f_\theta(x_1)$ and $f_\theta(x_2)$, where θ stands for the shared weights, represent the essential qualities and traits of the input images. By guaranteeing that both branches yield similar embeddings, the shared weights allow the network to identify the differences or similarities between the input pairs.
6) Euclidean Distance
The calculation of the Euclidean distance between the respective encoded vectors ($f_\theta(x_1)$ and $f_\theta(x_1)$) of two input images is done in order to quantify their similarity. The dissimilarity[3] or likeness between the input cases is quantitatively represented by this distance measurement. During the training process, the network learns to maximize this distance for dissimilar image pairs and minimize it for comparable image pairs.
7) Training Objective
The goal of training the Siamese network is to minimize the Euclidean distance for similar pairs of images and maximize it for dissimilar pairs at the same time. Typically, a contrastive loss[3] or a related measure learning technique is used to accomplish this training goal. Through optimization towards this goal, the network gains the ability to generate encoded vectors that accurately represent the intrinsic properties of the input data, enabling precise similarity evaluations between image pairs.

A. Software Package and Requirement

Essential libraries for building and training a Siamese neural network are included in the code. It imports a number of elements from the torch library, including transformations, optim (optimization techniques), and nn (neural network modules). Furthermore included are the necessary modules, such as torch, torchvision, matplotlib.pyplot, os, random, PIL (Python Imaging Library), numpy, and torch. Additionally, the code makes use of the torch.utils.data module for DataLoader features and random_split for dataset splitting. Images and results are displayed using Matplotlib.pyplot, which makes visualizations easier. The code's functionality is further increased by the addition of tkinter for GUI elements, filedialog for managing file dialogs, Image and ImageTk from PIL for image processing, and the SiamaseNet from the one_shot module. For further image changes, import torchvision.transforms and transforms. %matplotlib inline preserves the collaboration with Jupyter Notebook and allows for inline visualization.

B. Tkinter

Tkinter[25] is a widely used Python GUI (Graphical User Interface) framework that we used to create the user interface (UI) for our implementation. A straightforward yet effective toolkit for building interactive and user-friendly graphical user interfaces is provided by Tkinter. In our scenario, the Siamese Neural Network's operational code was integrated into the back-end, which was interfaced with with ease using Tkinter as the front-end. The Siamese Neural Network's functionality and the end user were connected by the Tkinter-based user interface. The Tkinter-designed interface allowed users to monitor the output, initiate network processing, and input photos. The operational Python code of the Siamese Neural Network provided the back-end logic behind the scenes, processing the input images, calculating similarity metrics, and producing the final outputs.

V. Project Flowchart

Training a network to determine whether candidate examples belong to the same class as a single sample image provided during the training phase is the idea behind one-shot image recognition.[3] For this, the siamese network design is used in the work that is discussed. The architecture, as seen in Figure 1, aims to provide a reliable representation of surface defects in steel. In order for the model to learn how features differ across pairs of input photos, it uses a contrastive loss function.

Using the surface defect dataset from Northeastern University (NEU)[3], trained the Siamese network model. Six classifications of surface defects on hot-rolled steel strip are included in this database:

I. Crazing (Cr)
II. Pitted surface (PS),
III. Inclusion (In),
IV. Rolled-in scale (RS),
V. Patches (Pa),
VI. Scratches (Sc)

The dataset includes 300 examples for each of the six classes, for a total of 1,800 grayscale images. Every sample image has a resolution of 200 × 200 pixels. In this 3 classes, inclusion (In), rolled-in scale (RS) and patches (Pa) were used for the training process. Rest of the total dataset, the other three classes were considered only for the testing.

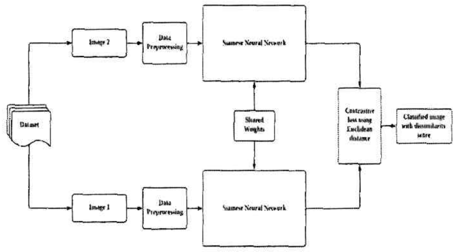

Fig. 3. Flowchart of architecture

The Siamese network shown in Fig 3 is made up of two identical modules that have the same weights. It is possible to think of each module as a parametric function[3] of weights $f_\theta : R^N \rightarrow R^n$, where $N \gg n$. An encoded vector of lesser dimension, n, is obtained by reducing the high-dimensional input (image), represented as R^N. In this particular instance, N is set to 100×100, and n is set to 5. The outputs at layers of size $n = 5$ resulting from the two modules are called the encoded vectors $f_\theta(x_1)$ and $f_\theta(x_2)$. The Euclidean distance between these encoded vectors is the architecture's final result. By giving one instance of each fault, the model is taught to identify many defects. The model receives two pairs of grayscale images[3], x1 and x2, as input. Three convolutional layers, each with 100x100 dimensions and feature maps of sizes 4, 8, and 8, are included in each module. Three completely connected layers with sizes of 500, 500, and 5 come after these convolutional layers. Rectified Linear Unit (ReLU) activation function is used to each layer's output feature maps[3], and a 3x3 kernel size is used for convolutions with a stride of 1. The Siamese network's capacity for one-shot image recognition is demonstrated by this architecture, which makes it possible for it to learn and identify intricate details in steel surface imperfections.

VI. Results

A. Training and Validation

Utilizing a dataset of 900 enhanced picture samples, the Siamese network was trained using particular hyper-parameters. To improve the network's capacity for generalization, these samples were subjected to a number of transformations.[3] There were 32 batches in the training procedure, which lasted 100 epochs. The learning rate of 0.005 and margin[3] $m = 2$ were given as hyper-parameter. The training and validation split was 80 and 20 respectively.

The plot of a one-shot recognition model for steel surface defect identification over 100 epochs shows the training and validation loss curves. The training loss first drops dramatically, suggesting that as the model picks up on trends in the training data, there will be a noticeable decrease in error. Even though the validation loss is reducing, it is more erratic, which indicates that the model had difficulties applying what it had learned to previously unobserved data in the early epochs. The training loss settles at a lower value than the validation loss around the 20th epoch, indicating that the model successfully minimizes error on the training data while retaining a respectable level of generalization. The validation loss does, however, fluctuate toward later epochs, which could be a sign of mild over fitting or sensitivity to the validation data. All things considered, the low ultimate loss values imply that the model does a good job of identifying surface flaws on steel.

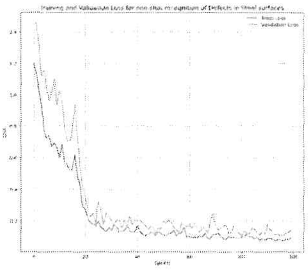

Fig. 4. Training and validation curve

Random selection was used to choose data samples for the training phase. In order to sample a picture pair, two photos with a probability of 0.5 were selected from the same category, and to indicate that they are part of the same class, the associated label was set to y = 1. On the other hand, pictures were selected with a remaining chance of 0.5 from two separate categories, and the label was set to y = 0, signifying that the pictures were taken from different classes. Then, augmentation[3] was applied to this tuple of picture pair and label (x1, x2, y) using the transformations.

Each image's pixel values were normalized before being sent into the network to make sure they fell between [-1, 1]. In order to standardize the input data and enable more reliable and efficient training, this normalization[3] step is essential.

The experiments were conducted on the Ryzen 9 platform equipped with 16GB RAM and an NVIDIA RTX 3060 GPU. These hardware specifications provide the computational power needed to train and evaluate the Siamese network efficiently. The choice of hardware, with a powerful GPU like the NVIDIA RTX 3060, is particularly advantageous for accelerating the parallelized computations involved in deep learning tasks, ensuring robust and timely model training.

As the number of epochs rose during this training, there was a noticeable trend toward a decrease in both the training and validation losses. This pattern indicates that the accumulated loss decreased as the model progressively understood the visual salience's present in the reference and candidate images.

B. User-Interface

The Tkinter framework in Python has been used to create the graphical user interface (GUI), which offers a visually ap-

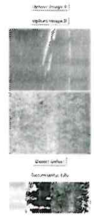

Fig. 5. GUI showing the dissimilarity score of two images

pealing and user-friendly platform. Three well-placed buttons that improve user interaction are part of the purposefully simple UI design. In particular, users can choose and import two photographs from their local system using these two buttons, which act as image upload mechanisms. The photos might represent multiple classes or relate to the same class, giving users choice in the kind of analysis they want to perform. The GUI's third and most important button is titled "Detect Defect." This button opens a connection to the application's back-end, which houses the Python code that implements the Siamese Neural Network.

The GUI's back-end easily connects to the siamese neural network, a potent deep learning model for image comparison and classification. The network has been particularly trained to detect defects in this particular scenario. The siamese neural network analyzes the uploaded photos, determines their dissimilarity scores, and categorizes them according to their features when the "Detect Defect" button is pushed. After that, the outcomes are effortlessly sent back to the GUI for user display.

The siamese neural network's defect detection capabilities can be easily used by users, even without substantial programming skills, thanks to the integration of an easy-to-use interface with a reliable back-end. An effective and efficient tool for picture categorization and defect identification is produced by combining the complex processing capacity of the siamese neural network with a simple user interface design.

C. Step by Step Instructions

I Run GUI Implementation Code:
Run the GUI implementation code that has been provided to start the graphical user interface (GUI). This code is in charge of configuring the GUI environment and outlining its features. Allow the code to finish its execution. The completion of the code execution initializes the GUI components and prepares the user interface for interaction.

II User Interface Appearance:
When the code runs successfully, the user interface screen appears on the screen. This graphical user interface makes it easier for users to engage with the underlying functionalities and allows for their smooth use. With just three buttons for user convenience, the UI design is straightforward. The user interface has two buttons that are dedicated to uploading photographs. These buttons allow users to choose and upload picture files from their

local system. This feature makes it possible to upload photos for later classification.

Fig. 6. Welcome screen of GUI

III Display Classified Image Button:

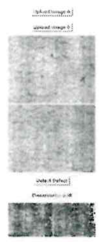

Fig. 7. Dissimilarity displayed after classifying two images.

The identified image and its dissimilarity score are displayed via the third and last button on the user interface (UI). This button, when pressed, causes the underlying siamese network to process the photos that have been submitted and displays the categorization results instantly.

VII. Conclusion

This study focused on applying a novel technique—one-shot recognition using siamese neural networks—for the identification of faults in steel surfaces in an effort to improve industrial quality control procedures. In the context of steel surface defect recognition within smart manufacturing, the main objective was to investigate the efficacy of this methodology and compare it with conventional neural networks (CNNs)[26] and a simple one-shot learning algorithm, the nearest-neighbor[27] algorithm with a single neighbour. Our investigation demonstrated the effectiveness of siamese neural networks in addressing the challenges posed by limited training data. The unique architecture, designed for one-shot recognition, proved to be particularly well-suited for steel surface defect recognition tasks. Through a comprehensive comparative analysis, we bench-marked the siamese network-based one-shot recognition approach against conventional CNNs and a basic one-shot learning algorithm. The results revealed the distinctive advantages of the siamese network in scenarios where training data is scarce. A critical extension of this endeavor involved the integration of a graphical user interface (GUI) to facilitate user interaction and enhance the practical usability of the developed defect recognition system. The integration of a GUI further augmented the user-friendliness and accessibility of the developed system.

Through its use in visual inspection tasks, the project demonstrated the potential uses of siamese networks in smart manufacturing. The emphasis on identifying steel surface defects highlighted how flexible the method is for important quality control procedures. The siamese network is a useful tool for businesses where obtaining large labelled datasets may be difficult or costly because to its ability to function well with little to no training data.

The siamese network-based one-shot recognition technology, when used successfully for steel surface defect recognition[28], has the potential to revolutionise quality control procedures in smart manufacturing. It is a compelling option in situations when getting big labelled datasets is not feasible due to the minimal data requirements. The introduction of a graphical user interface (GUI) addressed the practicality of deploying the defect recognition system in manufacturing environments. The GUI serves as an intuitive platform for users to interact with the system, providing functionalities for parameter tuning, real-time monitoring, and result visualization.

As we wrap up this study, a few new research directions become apparent. In some industrial scenarios, the Siamese network design may function better with additional improvements and optimisations. For practical implementation, it would also be essential to investigate its resilience in identifying a wider variety of problems and integrate it into real-time production systems.

In summary, the siamese neural network-based one-shot recognition method has shown to be a formidable instrument for detecting manufacturing flaws on steel surfaces. The successful integration of a GUI into the defect recognition system opens new possibilities for broader adoption in manufacturing environments. Future research avenues may explore additional features within the GUI, such as real-time analytics, automated reporting, and integration with other manufacturing systems to create a comprehensive and streamlined quality control process. The project showcases this methodology's efficiency and versatility, which adds to the growing field of smart manufacturing. The results of this study open the door for the incorporation of cutting-edge techniques that can transform quality control procedures and achieve the larger goal of intelligent and automated production as companies continue to adopt cutting-edge technology.

Acknowledgment

The authors would like to thank Prof. Dr. Sunil P Survaiya for the guidance and support throughtout the course of the project. We would also appreciate the resources and support from members at Technische Hochschule Deggendorf, campus Cham.

References

[1] Y. Meng, Y. Yang, H. Chung, P. H. Lee, and C. Shao, "Enhancing sustainability and energy efficiency in smart factories: A review," *Sustainability*, vol. 10, p. 4779, 2018.

[2] J. Pfrommer, C. Zimmerling, J. Liu, L. Karger, F. Henning, and J. Beyerer, "Optimisation of manufacturing process parameters using deep neural networks as surrogate models," *Procedia CIRP*, vol. 72, pp. 426–431, 2018.
[3] A. M. Deshpande, A. A. Minai, and M. Kumar, "One-shot recognition of manufacturing defects in steel surfaces," *Procedia Manufacturing*, vol. 48, pp. 1064–1071, 2020.
[4] I. Abdel-Qader, O. Abudayyeh, and M. Kelly, "Analysis of edge detection techniques for crack identification in bridges," *Journal of Computing in Civil Engineering*, vol. 17, pp. 255–263, 2003.
[5] H. Altae-Tran, B. Ramsundar, A. Pappu, and V. Pande, "Low data drug discovery with one-shot learning," *ACS Central Science*, vol. 3, pp. 283–293, 2017.
[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.
[7] B. Kiran, D. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *Journal of Imaging*, vol. 4, p. 36, 2018.
[8] J. Choi, H. J. Chang, T. Fischer, S. Yun, K. Lee, J. Jeong, Y. Demiris, and J. Young Choi, "Context-aware deep feature compression for high-speed visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.
[11] Y. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, pp. 361–378, 2017.
[12] S.-H. Huang and Y.-C. Pan, "Automated visual inspection in the semiconductor industry: A survey," *Computers in Industry*, vol. 66, pp. 1–10, 2015.
[13] J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, "Machine learning-based imaging system for surface defect inspection," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, pp. 303–310, 2016.
[14] J. Richter, D. Streitferdt, and E. Rozova, "On the development of intelligent optical inspections," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2017, pp. 1–6.
[15] T. Vafeiadis, N. Dimitriou, D. Ioannidis, T. Wotherspoon, G. Tinker, and D. Tzovaras, "A framework for inspection of dies attachment on pcb utilizing machine learning techniques," *Journal of Management Analytics*, vol. 5, pp. 81–94, 2018.
[16] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, vol. 2, no. 1, 2015.
[17] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," *arXiv preprint arXiv:1709.03410*, 2017.
[18] D. Droghini, S. Squartini, E. Principi, L. Gabrielli, and F. Piazza, "Audio metric learning by using siamese autoencoders for one-shot human fall detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2019.
[19] P. Neculoiu, M. Versteegh, and M. Rotaru, "Learning text similarity with siamese recurrent networks," in *Proceedings of the 1st Workshop on Representation Learning for NLP*, 2016, pp. 148–157.
[20] D. Mery, V. Riffo, U. Zscherpel, G. Mondragon, I. Lillo, I. Zuccar, H. Lobel, and M. Carrasco, "Gdxray: The database of x-ray images for nondestructive testing," *Journal of Nondestructive Evaluation*, vol. 34, p. 42, 2015.
[21] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.
[22] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2017.
[23] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, and et al., "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.
[24] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning*, 2015, pp. 2152–2161.
[25] P. S. Foundation, "Python programming language," 2001–. [Online]. Available: https://www.python.org/
[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[28] Y. Lee, "Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks," *Neural Computation*, vol. 3, pp. 440–447, 1991.

AUTHOR CONTRIBUTIONS

Gokul Soman: Neural network training and prediction. Contributions to report: Proposed system architecture and Project flowchart.

Basil Paul: Literature search and GUI development. Contributions to the report: Results and Conclusions.

Anand Gopinatha Pillai: Literature review and collection of dataset. Contributions to the report: Abstract, Introdution, Siamese neural network and Literature review.