Waste Reduction via Computer Vision-based Inspection: Towards Lean Systems in Metal Production

Mohammad Shahin
F. Frank Chen (FF.Chen@utsa.edu)
University of Texas at San Antonio
Ali Hosseinzadeh
Mazdak Maghanaki

Research Article

Keywords: Lean Manufacturing, waste reduction, Machine Learning, Industry 4.0, image classification

Posted Date: April 14th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2782987/v1

Abstract

The global steel demand continues to increase, with steel being used in various industries, including construction, automobile, national defense, and machinery. However, steel production is a delicate process that can result in different defects on the steel surface, negatively affecting the quality of the steel products. Therefore, recognizing metal surface defects is critical in the metal production industry. Manual detection of these defects is the standard method, but it is time-consuming, labor-intensive, and prone to subjective factors, leading to low accuracy and unreliable results. Automated defect detection using computer vision methods can replace or supplement manual detection. In recent years, machine learning algorithms, particularly Convolutional Neural Networks (CNNs), have shown great promise in achieving high accuracy rates in this task. In addition, image classification algorithms can contribute to Lean metal production by identifying defects or anomalies in the manufacturing process, which can be used to reduce waste and increase efficiency. However, the performance and cost of different CNN architectures can vary widely, making it challenging for decision-makers to select the most suitable model. This paper analyzes various CNN-based image classification algorithms, including MobileNet, ShuffleNet, DenseNet, RegNet, and NasNet, in classifying steel surface defects in the NEU-CLS-64 dataset. We evaluate their performance using metrics such as accuracy, precision, sensitivity, specificity, F1 score, and G-mean, and benchmark these models against each other. Our findings revealed that RegNet achieved the highest accuracy, precision, sensitivity, specificity, F1 score, and G-mean performance but at a higher cost than other models. Meanwhile, MobileNet had the lowest performance. The results provide decision-makers with valuable insights into selecting the most suitable CNN model for steel surface defect detection based on their performance.

1. Introduction

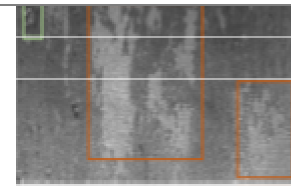
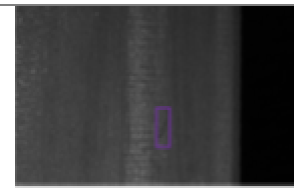
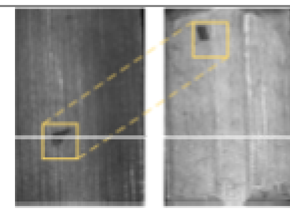
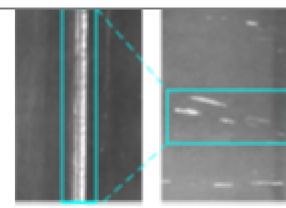
Adopting Lean manufacturing aims to enhance the quality of products or services by minimizing waste, thereby adding value. This makes it crucial for meeting customer needs and staying competitive in today's global markets. The concept of Lean Manufacturing originated in Japan, where Toyota was the first to implement it as the Toyota Production System. Nowadays, Lean is widely applied across industries, defined as providing quality products at low manufacturing costs. The primary goal of Lean Manufacturing is to reduce waste in the enterprise, as waste represents an extra burden on resources that do not add value to the company. Lean Manufacturing tools help minimize operational costs by reducing waste, optimizing product quality, and increasing efficiency [1]. Lean manufacturing is widely accepted and adopted in various industries to achieve maximum productivity, high quality, and cost reduction. Implementing Lean practices enhances organizational productivity, reduce manufacturing costs, eliminates unnecessary downtime, improves resource utilization, and increases profitability. Consequently, adopting Lean principles enhances the competitiveness of any organization in the market [2]. As a result, enterprises seek to implement the Lean Manufacturing concept to become more financially viable and operationally efficient. Applying the Lean manufacturing concept has led to continuous organizational improvement [3]. Implementing Lean Manufacturing tools, including Total

Productive Management (TPM), Overall Equipment Effectiveness (OEE), and Jidoka, directly impacts the enterprise's environment and contributes to its social and economic sustainability [4]. Lean manufacturing has operational benefits regarding quality, delivery, cost, and volume flexibility [5, 6]. Studies have demonstrated that adopting digital technology and Lean manufacturing can reduce production costs despite varying environmental conditions. The integration of Lean manufacturing and digital technology enhances operational performance and the efficiency of production management. Implementing Lean practices has also improved an enterprise's financial standing by adopting a safe shop floor management approach [7]. Integrating the Lean manufacturing concept into railway transport management has optimized labor and material resources. [8]. This implementation of Lean tools has resulted in a reduction in operation time, an improvement in quality, and an expansion of the customer base [8, 9]. Implementing Lean manufacturing tools in enterprise management has increased efficiency and production quality while minimizing waste in workstations, resulting in reduced operational costs [10, 11]. An additional example demonstrated that incorporating Lean manufacturing tools has prevented order returns. As a result, there is a decrease in waste production, lower production costs, and improved operational performance [12]. Despite the documented benefits of the Lean manufacturing concept in the literature, certain barriers hinder its implementation in organizations. These obstacles include a lack of responsibility, knowledge, training, and budgetary constraints [13]. On a positive note, research has demonstrated that there is still significant potential to benefit from applying Lean tools in various enterprises. [14–16]. Developing new and innovative technologies is essential to achieve sustainability in manufacturing. Introducing Lean manufacturing can aid in this pursuit by reducing waste [17]. Green Lean integrates environmentally sustainable practices into the Lean manufacturing methodology. It involves reducing waste and minimizing the environmental impact of production processes by optimizing resource utilization, reducing greenhouse gas emissions, and adopting environmentally-friendly practices. Green Lean practices focus on achieving operational efficiency while minimizing environmental harm, contributing to sustainable manufacturing [18]. This paper deployed machine-based vision technology to identify defects and reduce waste in the process, thus showing how an Industry 4.0 technology combined with Lean tools can help achieve a sustainable manufacturing practice. Different types of waste exist in manufacturing, referred to as "MUDA," a Japanese word that includes defects, motions, over-production, inventory, scrap and waste, waiting, skill, and transportation. Implementing Lean manufacturing tools in the manufacturing industry aimed to eliminate waste [19]. Statistical evidence has confirmed that defects are the most commonly reported issue in manufacturing. Waste can exist in various forms, including inventory, motion, waiting, and transportation, and can be caused by defects, over-production, scrap, and unnecessary motion [20]. Disorganized work and non-standardized production processes can also contribute to waste [21]. Eliminating waste through continuous improvement helped balance the workload and remarkably reduced the lead time [22, 23]. Furthermore, it reduces non-value activities, eliminates bottlenecks in the production process, lowers costs, and maximizes the production rate. [24, 25]. Furthermore, applying Lean Manufacturing tools such as 5s, Kaizen [26], and Kanban can help reduce the defects rate [27] and in increasing efficiency [24, 28]. Eliminating waste through Lean Manufacturing tools [29] helps keep end customers happy, produces quality products, and reduces cost and delivery time [30]. Previously, Information Technology (IT) was often regarded as a waste source in the context of Lean manufacturing. However, this perception has gradually diminished as the integration of automated manufacturing with the potential of IT has transformed the roles of managers, engineers, and operators, leading to the emergence of knowledge-based workplaces [31]. Hence, it is crucial to incorporate automation and IT more extensively into manufacturing procedures [32]. Using data science enables the conversion of large amounts of data into useful information, leading to enhanced transparency and improved product quality. Combined with Lean Manufacturing techniques, it can significantly improve production systems [33, 34]. The utilization of sensors during the quality control process and advancements in analytics to interpret the data gathered from these sensors have been proven useful in optimizing tasks [35]. In addition, various Machine Learning (ML) algorithms, including Artificial Neural Networks (ANN), have been applied to optimize the efficiency of operations in logistics, supply chain, production, and marketing [36]. Research has shown that integrating Industry 4.0 (I4.0) technologies with Lean manufacturing can have a synergistic and complementary effect, resulting in improved production performance. This is achieved by utilizing the real-time monitoring capabilities of I4.0 to optimize the manufacturing process and enhance the benefits of Lean manufacturing [37, 38]. One of the main goals for businesses is to gain a competitive edge by increasing profits and decreasing costs. One way to achieve this is through continual enhancement of productivity and quality. By integrating Lean Six Sigma tools with Industry 4.0, companies can achieve these objectives by reducing waste [39] and creating opportunities to achieve operational excellence [40–42]. The Industrial Internet of Things (IIoT) devices are utilized in intelligent manufacturing systems to monitor production performance and enhance reliability [43]. They can detect failures and wear conditions in real-time with high accuracy [44]. Motion capture technology can improve safety measures by tracking and recording workers' movements to analyze them and prevent early injuries, ultimately improving the overall efficiency of the manufacturing process [45]. By integrating IIoT into manufacturing processes and material handling operations, Lean automation can attain substantial efficiency gains. [46]. The integration of cutting-edge technologies, including big data, IIoT, cloud computing, and cloud-based services, alongside the anticipated fifth and sixth-generation networks, has facilitated the extraction of vast amounts of data from the production processes. This data plays a critical role in designing high-quality products and ensuring consistency in the manufacturing process. The advanced analytics offered by big data have also improved forecasting accuracy, enabling better decision-making and ultimately reducing waste in the manufacturing system while adding value to the final product [47]. Combining Lean manufacturing tools with Machine Learning (ML) has been suggested as a way to eliminate waste and add value to manufacturing [48], as sometimes the means of Lean manufacturing alone may not be sufficient [49]. In addition, integrating ML into manufacturing can enhance production, efficiency, and quality [50]. Achieving environmental responsibility can be facilitated by implementing Lean Six Sigma (LSS) techniques [51, 52]. However, integrating the emerging technologies of I4.0 with Lean Six Sigma (LSS) makes it possible to sustain environmental responsibility in the long run. Thus playing a crucial role in achieving an eco-friendly and sustainable approach to green manufacturing initiatives [53].

2. Metal Production And Inspection

The global need for steel has continued to rise annually; steel is mainly used in infrastructure projects. The United States alone consumed 98 million metric tons in 2021 [54]. In addition, steel plates are crucial in automobile, national defense, machinery, chemical, and light industries. Defect waste refers to producing any item unsuitable for use, such as defective materials, components, or products not made according to specifications [55]. Steel production is a delicate process involving various stages, such as heating, rolling, drying, and cutting. During these stages, the steel surface may come in contact with several machines before it is ready for shipment. Consequently, various defects that must be identified on the steel plate may arise. Imperfections such as cracks, scabs, curling edges, cavities, and abrasions on the surface can occur due to raw material and technological issues, negatively affecting corrosion resistance, strength, and the economic benefits of the factory [56]. The expanding demand for steel has created a need for a highly efficient manufacturing system in the steel processing industry to produce products that meet consumers' expected quality. Manual detection is the standard steel surface defect detection method, which increases the labor cost while being vulnerable to subjective factors and produces low accuracy and unreliable results [57]. Computer vision methods can replace or supplement manual detection in the defect classification process. Therefore, it is imperative to investigate deploying a computer-based vision algorithm to detect surface defects on the production line automatically. Automated defect detection is a specialized area of object detection that focuses on identifying specific defects on the surfaces of various industrial products, such as steel, glass, printed circuit boards (PCBs), wafers, wood, and more. Steel is crucial in multiple fields, including construction, vehicle parts, infrastructure, etc. Therefore, implementing an automated system for detecting defects on steel surfaces could enhance the quality of steel products [58]. However, detecting metal surface defects remains challenging due to the randomness of their generation. Table 1 summarizes these challenges.

Table 1. Challenges encountered during detecting metal surface defects [59]

Challenge	Description	Example
The ambiguity between normal and defect	Detecting metal surface defects in the complex industrial production environment can be challenging as many normal areas (inside the orange box) in the samples appear similar to defective regions (inside the green box)	
Tiny defective area	Detecting metal surface defects can be challenging due to several factors. The area of the defect is usually small, and the camera angle may influence its acquisition. Additionally, in a complicated industrial production scenario, recognizing defects similar to the background color (inside the purple box) can be even more challenging	
Inter-class similarity	Minor morphological variations can cause visually similar appearances of defects in different classes, leading to high interclass similarity (inside the yellow boxes)	
Intra-class variance	Drastic differences in appearances can exist among the defects belonging to the same class, resulting in an intra-class variance (inside the blue boxes)	

An additional challenge is a prevalence of imbalanced data [60], a common occurrence in manufacturing. In terms of sample size, there are typically more products of good quality compared to those of bad quality. Additionally, specific defect categories may have fewer samples than others, further complicating matters and making it difficult to extract features of the defects. The problem is compounded by the fact that the background area is significantly larger than the defect area, resulting in an imbalanced dataset. As a result, advanced machine vision algorithms have been employed in metal surface detection. These methods are more efficient, flexible, and accurate than human inspection [61].

In the context of metal production, Lean principles can be applied to various stages of the production process, such as raw material handling, metal cutting and shaping, assembly, and finishing with quality control and inspection processes. By adopting Lean practices, metal manufacturers can optimize production, improve quality, and reduce lead times, improving competitiveness and increased profitability. Image classification algorithms can contribute to Lean metal production by identifying defects or anomalies in the manufacturing process. Using computer vision, images of metal products can be analyzed to detect any defects or abnormalities in their shape, size, or texture. This information can then be used to identify areas where the manufacturing process can be improved to reduce waste and increase efficiency. For example, an image classification algorithm can be used to identify surface defects on metal products, such as scratches, cracks, or dents. This information can be used to improve the quality control process, reduce scrap, and improve overall product quality. Similarly, image classification algorithms can be used to identify defects in the shape or size of metal products, which can help to optimize the production process and reduce waste. Overall, image classification algorithms can help to streamline the manufacturing process, reduce waste, and improve product quality in Lean metal production.

3. Dataset

Northeastern University (NEU) has released a dataset on steel surface defects called NEU-CLS-64 [62], containing nine common types of defects found in hot-rolled steel strips. The dataset consists of more than 7000 grayscale images. The nine types of defects included in the dataset are rolling scale (RS), plaque (Pa), cracking (Cr), pitting surface (PS), inclusions (In), scratches (Sc), grooves and gouges (GG), rolling dust (RD), and spots (Sp). Figure 1 shows an illustration of these defects.

Hot-rolled steel strips are thin sheets of steel that are produced through the hot-rolling process. This process involves heating the steel above its recrystallization temperature, which typically ranges from 1,000 to 1,200 degrees Fahrenheit. Thus, making it easier to shape and form. In addition, the process allows the steel to be deformed without cracking. Hot-rolled steel is commonly used in applications where precise shapes and tolerances are not required, such as in the construction industry for structural steel components and in manufacturing for various types of machinery and equipment. They are known for their strength, durability, and affordability. Hot-rolled steel is typically characterized by a scaly surface texture and a blue-grey oxide scale that forms on the surface during cooling [63]. Hot-rolled steel strips can suffer from a variety of surface defects that can affect the quality and performance of the material. These defects include but are not limited to r rolling scale (RS), plaque (Pa), cracking (Cr), pitting surface (PS), inclusions (In), scratches (Sc), grooves and gouges (GG), rolling dust (RD), and spots (Sp). A summary of these defects with their illustrations is given in Table 2. In addition, the dataset was subjected to image augmentation, a technique in computer vision and deep learning that involves applying various transformations to existing training images to create new, slightly modified versions of the original data. This can include rotation, scaling, cropping, flipping, adding noise, changing brightness or contrast, and many others. Applying these transformations to the training data exposes the model to a wider variety of possible inputs, which can help it generalize better to new, unseen data. Image augmentation is beneficial when the available training data is limited or imbalanced, as it can effectively increase the size and diversity of the training set without requiring additional data collection.

Table 2
Description of hot-rolled steel surface defects

RS [63], (1589 total defects)
Definition: Rolling scale defect, or mill scale, is a surface defect that occurs on hot-rolled steel strips. It appears as a dark, flaky residue on the surface of the steel, caused by the formation of iron oxide during the heating and rolling process. Root Cause: Rolling scale defects occur during the hot rolling process of steel strips when the scale layer, which forms on the surface of the steel due to oxidation during the high-temperature heating, is not entirely removed during the subsequent rolling process. The incomplete scale removal can result in cracks and voids on the steel surface, known as rolling scale defects. These defects can occur due to various factors, such as inadequate cleaning of the steel surface before the rolling process, insufficient rolling pressure, incorrect temperature control, or a combination of these factors. Effects: The rolling scale can adversely affect the surface quality of the steel, as it can interfere with subsequent coating or painting operations. Rolling scale can also be a potential source of corrosion if left untreated. In addition, the formation of rolling scale defects can negatively affect the surface quality of the steel strips, reducing their value and usability in various applications. Therefore, detecting and classifying this type of defect during the manufacturing process is essential.
Pa [64], (1148 total defects)
Definition: The plaque defect is a surface defect that occurs in hot-rolled steel strips. It appears as a raised area or bumps on the surface of the steel strip, which can vary in size and shape. Root Cause: Plaque defects are caused by the formation of hard and brittle iron oxide layers during the cooling of the hot-rolled steel strip. These oxide layers can crack and spall off, leaving behind irregularly shaped pits or craters on the surface of the strip. The formation of plaque defects is influenced by several factors, including the steel's composition, the strip's thickness, the cooling rate during processing, and the presence of contaminants such as sulfur and phosphorus. The plaque defect is also caused by non-metallic inclusions in the steel, which can become trapped on the surface during the rolling process. Effects: The plaque defect can make the surface of the steel strip look rough, uneven, and aesthetically unappealing. And if the plaque is present on a critical surface of the steel strip, it can reduce product quality, as the defect can cause performance issues in the final product. In addition, the plaque defect can cause processing issues during subsequent manufacturing processes, such as welding or coating, as it can interfere with the adhesion of coatings or the welding process. And in extreme cases, the plaque defect can result in material loss, as the affected portion of the steel strip may need to be trimmed or discarded altogether. The plaque defect can negatively impact the quality and appearance of the steel and can also affect the performance of downstream processes that use the steel strip.
Cr [65], (1210 total defects)

RS [63], (1589 total defects)
Definition: Cracking is a defect that can occur on the surface of hot-rolled steel strips. It refers to the appearance of cracks, fissures, or splits on the surface of the steel strip. These cracks can be either longitudinal (running parallel to the rolling direction) or transverse (running perpendicular to the rolling direction). Cracking can range from fine hairline cracks to much larger, deeper ones. Root Cause: Cracking defects in hot-rolled steel strips can be caused by several factors, such as the presence of impurities, improper cooling during the manufacturing process, or high levels of stress due to improper handling or transportation. Rapid cooling of the steel strip can also cause thermal stress, leading to cracking. Additionally, improper lubrication during the rolling process can cause friction, generating heat that can lead to cracking. Finally, a seam or inclusion in the steel strip can also create stress concentrations, making it more susceptible to cracking. Effects: Cracking defects can significantly reduce the mechanical properties of the hot-rolled steel strip, making it weaker and more prone to failure under stress. This can cause the strip to break or fracture during processing or use, leading to production delays, equipment damage, and potentially hazardous situations. Cracks may also propagate and grow over time, making the strip susceptible to failure even at lower stresses. In addition, cracks can create rough and uneven surfaces, affecting the quality of downstream processing operations and the final product.
PS [66], (797 total defects)
Definition: Pitting is a surface defect that appears as small holes or cavities on the material's surface. It can occur in various materials, including metals, due to corrosion, erosion, or mechanical damage. Root Cause: In the context of hot-rolled steel strips, the pitting surface can be caused by impurities in the material or by the interaction of the steel surface with the surrounding environment during the production process. Other factors include non-metallic inclusions, roll wear, scale and debris, and localized overheating during hot rolling. Pitting corrosion can also occur in aggressive environments like salt water or acidic solutions. Effects: Pitting can reduce the aesthetic appeal of the steel strip, and in severe cases, it can lead to corrosion and compromise the material's structural integrity. Therefore, pitting is a vital defect to detect and prevent to ensure the steel product's quality and reliability and prevent reducing the integrity and strength of the material.
Sc [67], (773 total defects)
Definition: Scratches are linear grooves or marks on the surface of the hot-rolled steel strip. These are considered a type of surface defect that can affect the quality and appearance of the steel. Root Cause: Scratches can vary in depth and length, and various factors, such as improper handling during transportation, processing, or storage, abrasive materials, or contact with sharp objects, can cause them. Scratches can also occur during the rolling process if there is an issue with the rolling equipment or if the strip comes into contact with foreign material. Effects: Scratches on the surface of steel can have several effects. Firstly, they can affect the appearance and aesthetics of the steel, making it look unattractive and reducing its value. Secondly, scratches can compromise the integrity of the steel, making it more susceptible to corrosion and reducing its strength. Thirdly, scratches can interfere with the functionality of the steel, particularly in applications where a smooth surface is required for efficient operation (e.g., in bearings or moving parts). Additionally, scratches can provide sites for accumulating dirt, bacteria, or other contaminants, which can be problematic in applications where cleanliness is essential (e.g., food processing or medical equipment).
In [68], (775 total defects)

RS [63], (1589 total defects)
Definition: Inclusions are a type of defect that occurs in metals, including hot-rolled steel strips. Inclusions refer to any foreign material that becomes embedded in the metal during the manufacturing process. These foreign materials can include non-metallic particles such as oxides, sulfides, and silicates or metallic particles such as iron, manganese, and aluminum. In addition, inclusions can be categorized as either internal or external. Internal inclusions are embedded within the metal, whereas external inclusions are present on the surface of the metal strip. Root Cause: Inclusions can be caused by various factors, including the quality of the raw materials used, the manufacturing process, and the temperature and pressure conditions during the production process. Inclusions can also be caused by environmental contamination, such as dirt, dust, or other debris. Effects: The effect of inclusions on the hot-rolled steel strip depends on their size, shape, and location within the metal. Inclusions can weaken the metal and reduce its strength and ductility, making it more susceptible to cracking and other forms of failure under certain conditions. They can also affect the appearance of the metal, causing it to have a rough or uneven surface.
GG [69], (296 total defects)
Definition: Grooves and gouges are surface defects that can occur in steel during manufacturing, handling, or use. A groove is a narrow, elongated depression or channel on the surface of the steel. A gouge is a deep, irregularly shaped groove on the surface of the steel. Root Cause: Grooves can be caused by abrasive wear, scratches, or gouging. They can also occur during manufacturing due to inadequate cooling or lubrication, improper handling, or improper grinding or polishing. At the same time, gouges can be caused by mechanical damage, such as impact or scraping, or by exposure to corrosive substances. Gouges can also occur during manufacturing due to inadequate cooling or lubrication, improper handling, or improper grinding or polishing. Effects: Both grooves and gouges can affect the surface finish and cause weaknesses in the steel, reducing its strength and increasing the likelihood of failure.
RD [70], (200 total defects)
Definition: Rolling dust is a surface defect that appears as dark, irregularly shaped patches or lines on the surface of the steel. Root Cause: It is caused by the accumulation of debris, such as mill scale and iron oxide, on the rolls during the rolling process. This debris is then pressed into the surface of the steel as it is rolled, resulting in the appearance of rolling dust. Effects: The defect can affect the surface quality of the steel, reducing its visual appeal and potentially impacting its performance in certain applications.
SP [71], (438 total defects)
Definition: Spot is a steel surface defect that appears as a circular or elliptical depression or indentation on the steel surface. It is usually tiny, with a few millimeters or less in diameter. Root Cause: Spots can occur for various reasons, such as inclusions, improper grinding or polishing, improper coating, or surface contamination. Effects: The presence of spots on steel surfaces can reduce the aesthetic appearance of the material, as well as affect its mechanical and physical properties. Therefore, detecting and preventing spot defects is essential in quality control processes.

Detecting surface defects in hot-rolled steel strips and monitoring their frequency during quality control can provide valuable data for root cause analysis. By identifying and quantifying the frequency of each type of defect, it is possible to identify the areas of the manufacturing process where the defects are occurring and their possible causes. For example, if there is a high frequency of cracking defects in a particular batch of steel strips, it may indicate that the steel was cooled too quickly or that there is an issue with the rolling process. By analyzing the data, engineers, and technicians can identify the specific cause of the defects and make changes to the process to prevent them from happening in the future. Furthermore, tracking the frequency of defects over time makes it possible to identify trends and patterns that can help identify long-term issues in the manufacturing process. This can allow for continuous improvement efforts to be implemented to reduce the overall frequency of defects and improve product quality. Figure 2 shows the framework for the computer-based vision inspection of metal surface defects.

4. Methodology

Implementing IIoT tools and equipment has enabled automated defect detection in intelligent manufacturing systems, improving quality and efficiency [72]. This study aims to demonstrate how I4.0 technologies, mainly computer-based vision, can reduce waste by accurately classifying steel surface defects. To achieve this, a dataset comprising various instances of steel surface defects could be classified using different image classification algorithms. Traditionally, such inspection processes were carried out manually with cameras transmitting images to a screen monitored by human operators. However, integrating automation, big data, and computer-based vision detection systems can significantly enhance manufacturing systems [73]. In the last decade, Deep Learning (DL) models have gained recognition due to their ability to power computer vision-related tasks [74, 75]. Convolutional Neural Network (CNN) [76] uses a feed-forward topology to propagate signals and is being widely used for image classification and object detection [77]. Surface defect segmentation has gained significant attention in industrial inspection in recent years due to its high precision. Various studies have been conducted in this area. For example, Bian et al. [78] introduced a multiscale fully convolutional network to segment the defect of aeroengine blades. Similarly, Yu et al. [79] proposed a two-stage fully convolutional network for predicting defect areas in an industrial setting. Tabernik et al. [80] developed a segmentation-based deep-learning architecture for surface-crack detection. The first stage employs a segmentation network to identify defect areas, and the second stage uses an additional decision network to predict the entire image's abnormality. Moreover, Liong et al. [81] proposed a method for segmenting leather defects, which utilizes convolutional and deconvolutional neural networks. Delconte et al. [82] proposed a novel structure using a relief map image and convolutional neural network for wood defect segmentation. Xie et al. [83] presented a main and secondary net for defect segmentation on textured surfaces. The secondary net extracts features in the frequency domain, and the main net extracts features in the spatial domain and fuses the extracted features from the secondary net. Wu et al. [71] also designed a ResMask framework for generating generic defect images and a coarse-to-fine module to detect and segment generic defects. All these studies have contributed to developing machine learning-based methods for surface defect segmentation, which have potential applications in industrial inspection. The concept of

Graph Convolution Network (GCN), an analog convolution operation, was first introduced by Bruna et al. [84]. It was mainly used for graph structures in non-Euclidean space, such as social networks, chemical molecular structures, and knowledge maps [85]. This paper evaluates the fidelity and efficiency of multiple image processing algorithms. A list of these algorithms presented in Table 3. An 70/15/15 data split was used for training, validating, and testing respectively. During training, the system learns to identify patterns and features that are associated with defects and uses this information to make predictions on new, unseen examples.

Table 3
List of deployed algorithms.

AlexNet	CancerNet	DenseNet	EfficientNet
GoogleNet	InceptionNet	LeNet	MobileNet
NasNet	RegNet	ResNet	ShuffleNet
SqueezeNet	VggNet	XceptionNet	ZfNet

4.1 LeNet

LeNet is a CNN that was introduced in 1998 by Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. It was one of the first successful CNN architectures for handwritten digit recognition [86]. The LeNet architecture consists of seven layers: the input layer, two convolutional layers, two subsampling (pooling) layers, a fully connected layer, and an output layer [87]. The input layer takes in the images, which are typically preprocessed to a fixed size. The first convolutional layer applies a set of learnable filters to the input images to extract features. The subsampling layers reduce the dimensionality of the feature maps, making the model more computationally efficient. The second convolutional layer and subsampling layer are applied in the same way [88]. The fully connected layer connects all the neurons in the previous layer to the output layer. The output layer produces a probability distribution over the possible classes. The LeNet architecture has been used for various image classification tasks, such as handwritten digit recognition, facial recognition, traffic sign classification, and medical imaging [89]. It has also served as a foundation for many more advanced CNN architectures that have been developed since its introduction.

4.2 AlexNet

AlexNet is a type of CNN developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, and presented at the ImageNet Large Scale Visual Recognition Challenge in 2012 [90]. It was one of the first deep learning models to achieve high accuracy in image classification tasks, and it revolutionized the field of computer vision. AlexNet consists of 5 convolutional layers, followed by three fully connected layers, and uses various techniques such as ReLU activation functions, overlapping pooling, and dropout regularization to reduce overfitting [91]. The architecture of AlexNet includes around 60 million parameters, making it a large and complex model for its time. AlexNet has been deployed in the detection and segmentation tasks.

4.3 EfficientNet

EfficientNet has a CNN architecture that Google researchers developed in 2019 to achieve state-of-the-art performance on image classification tasks while being computationally efficient. EfficientNet uses a compound scaling method that uniformly scales a CNN's depth, width, and resolution to improve its efficiency while maintaining or improving its accuracy. The architecture is based on the idea that scaling up all network dimensions will lead to improved performance. However, uniformly scaling all dimensions reduces accuracy returns and increases computational costs [92]. EfficientNet introduces a new scaling method that involves balancing network depth, width, and resolution in a principled way. The authors propose a novel compound scaling method that uniformly scales the network depth, width, and resolution with fixed scaling coefficients. They demonstrate that this scaling method leads to better performance than traditional methods that independently scale these dimensions [93]. EfficientNet has achieved state-of-the-art performance on several benchmark datasets, including ImageNet, CIFAR-10, and CIFAR-100, while requiring fewer parameters and less computation compared to other state-of-the-art CNN architectures [94].

4.4 VggNet

VGGNet, short for Visual Geometry Group Network, is another CNN architecture proposed by the Visual Geometry Group at the University of Oxford in 2014 [95]. The VGGNet architecture consists of convolutional layers with 3x3 filters and max pooling layers with 2x2 filters. The number of filters increases as the spatial resolution decreases. The final feature maps are then fed into fully connected layers for classification [96, 97]. VGGNet achieved state-of-the-art performance on the ImageNet dataset in 2014, and its architecture has since been used as a basis for many other CNN architectures [98, 99].

4.5 XceptionNet

XceptionNet is a deep CNN architecture that was proposed by Google researchers in 2016. The name "Xception" stands for "Extreme Inception", as it builds on the Inception architecture while significantly changing its basic modules [100]. The critical innovation of XceptionNet is the use of depthwise separable convolutions, which factorize a standard convolution into a depthwise convolution and a pointwise convolution. This allows the network to capture spatial information and channel-wise interactions separately, making parameters more efficient and effective [101]. XceptionNet achieved state-of-the-art results on the ImageNet classification task and has since been used as a backbone architecture in many computer vision applications, such as object detection and semantic segmentation [102, 103].

4.6 ResNet

ResNet (short for Residual Network) is a deep neural network architecture developed by Microsoft researchers in 2015. It was designed to solve the problem of vanishing gradients in very deep CNN by introducing residual connections or "shortcut" links that allow gradients to flow directly through the network without passing through many layers. The ResNet architecture consists of a series of residual blocks, each containing several convolutional layers and shortcut connections. These blocks allow the network to be very deep (e.g., hundreds of layers) while still being able to train effectively [104]. ResNet has been widely used in various computer vision tasks, such as image classification, object detection, and segmentation. It has achieved state-of-the-art performance on several benchmark datasets, including ImageNet and COCO [105, 106].

4.7 GoogleNet

GoogleNet, also known as Inception V1, is a CNN architecture developed by Google researchers in 2014. It was designed to have a deeper architecture while reducing the computational cost of training such a deep network. The main innovation of GoogleNet is the Inception module, which uses multiple filter sizes in parallel to capture features of different scales within the same layer. This allows the network to capture many features without requiring multiple layers with varying filter sizes. GoogleNet also uses a global average pooling layer to reduce the number of parameters in the final fully connected layer, reducing the computational cost of training the network [107]. GoogleNet achieved state-of-the-art performance on the ImageNet dataset at its release, and its architecture has influenced the design of many subsequent types of CNN [108].

4.8 InceptionNet

Inception V3 is a CNN architecture that Google developed as a follow-up to GoogleNet. The architecture uses multiple convolutional filters of different sizes in parallel to extract features from an input image. The architecture also includes inception-v3 modules [109]. The InceptionNet architecture achieved state-of-the-art performance on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and has since been used in various computer vision applications such as fault detection [110], underwater image classification [111], plant disease detection [112], and other medical applications.

4.9 MobileNet

MobileNet is a CNN light architecture designed for efficient computation on mobile devices with limited computational resources, such as smartphones or embedded systems. It uses depthwise separable convolutions, which split the standard convolution operation into a depthwise convolution and a pointwise convolution, reducing the number of parameters and operations required while maintaining accuracy [113]. This makes it a popular choice for mobile applications where power and memory constraints are important factors [114, 115].

4.10 ShuffleNet

ShuffleNet is a neural network architecture designed for efficient deep learning on mobile and embedded devices. It was proposed in 2018 by Xiangyu Zhang et al. from Megvii Research. ShuffleNet is based on pointwise group convolution, which allows for efficient computation of feature maps by reducing the number of operations required for convolutions. The network architecture includes a shuffling process performed after pointwise group convolutions, which helps to increase the mixing of feature maps across different channels [116]. This improves the ability of the network to capture complex features and increases its accuracy. ShuffleNet has achieved state-of-the-art performance on several image classification benchmarks while being computationally efficient and requiring a low memory footprint. ShuffleNet was used extensively in autonomous Unmanned Aerial Vehicle (UAV), among other applications such as tool wear recognition, detecting and classifying fruits, leaf disease recognition, as well as medical applications [117, 118].

4.11 DenseNet

DenseNet (short for Densely Connected Convolutional Networks) is a CNN architecture design that was introduced in 2016. It is similar to the ResNet architecture in that it aims to address the vanishing gradient problem that can occur in deep neural networks. In a DenseNet, each layer receives the feature maps from all preceding layers and passes on its feature maps to all subsequent layers. This dense connectivity pattern means that information can flow more quickly through the network, which can help to mitigate the vanishing gradient problem and improve gradient flow during training [119]. DenseNets are also known for their parameter efficiency since they can achieve high accuracy using fewer parameters than other architectures. DenseNet has been used in deep fake image detection [120] and bird song classification [121].

4.12 RegNet

RegNet is a neural network architecture developed by Facebook AI Research (FAIR) that aims to optimize the computational efficiency of deep neural networks. RegNet uses a regularization technique called Adaptive Regularization of Weights (AROW) to reduce the parameters and operations required to train a neural network while maintaining accuracy. The architecture also combines grouped convolutions and bottleneck blocks to improve computational efficiency [122]. The performance of RegNet has been demonstrated on several benchmark datasets, with results comparable to or better than other state-of-the-art architectures while requiring fewer computational resources. RegNet has been used in surface defect detection [123], identification of fish species [124], and analysis of medical imagery [125].

4.13 NasNet

NasNet is a deep neural network architecture developed by Google researchers designed to learn the structure of neural networks automatically. "Nas" stands for "neural architecture search", which automatically discovers the best neural network architecture for a given task. The architecture comprises a search space of building blocks and a search algorithm that determines the optimal combination of these building blocks to achieve the best performance on a given task [126]. NasNet has achieved state-of-the-art performance on several image recognition tasks [127].

4.14 CancerNet

A unified deep-learning network architecture was developed for cancer detection and recognition-related tasks. It has a custom design that consists of an encoder, decoder, and classifier. The input and output layers of CancerNet have 24,565 nodes each [128]. CancerNet has been used to detect breast cancer [129] and in various medical applications [130].

4.15 ZfNet

ZfNet (Zeiler and Fergus Net) is a CNN architecture designed for image classification tasks. It was proposed in 2013 by Matthew D. Zeiler and Rob Fergus. ZfNet consists of five convolutional layers followed by two fully connected layers and a softmax classifier. The architecture uses a smaller filter size than AlexNet, with the first layer using a 7x7 filter and the remaining layers using 3x3 filters [131]. ZfNet was one of the early CNN architectures that achieved high accuracy on the ImageNet dataset. ZfNet has been used in Kiwifruit detection in field images [132], palm vein recognition [133], and fruit-on-plant detection [134].

4.16 SqueezeNet

SqueezeNet is a deep CNN architecture designed to have a small number of parameters and low memory requirements. It was introduced in 2016 by the University of California, Berkeley researchers. The architecture aims to achieve high accuracy on image classification tasks while using significantly fewer parameters than other models [135]. This is accomplished through a combination of techniques, including using 1x1 convolutions to reduce the number of filters in the network and eliminating fully connected layers in favor of global average pooling [136]. SqueezeNet is particularly well-suited to applications with limited computational resources, such as mobile devices and embedded systems. Figure 3 shows the accuracy and loss vs. epochs for both training and validation for MobileNet. Figure 4 tries to show a general representation of the CNN architecture that was utilized at the core of all these models.

5. Results And Discussion

A confusion matrix is a table used to evaluate the performance of a classification algorithm or detection model. It is a matrix of true and predicted class labels for a test data set. It displays the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). In other words, it shows how well a model predicts the correct class labels and how often it makes mistakes. The matrix obtains a model's accuracy, precision, recall, and F1-score, which are important metrics in machine learning evaluation. Figure 5 shows the confusion matrix that was generated for the MobileNet model. And Table 5 shows the TP, FP, TN, and FN values calculated based on it.

Table 4
TP, FP, TN, and FN values for MobileNet model for all defect classes.

	FP	FN	TP	TN
Cr	5	1	181	904
Gg	77	4	41	969
In	5	0	117	969
Pa	200	1	172	718
Ps	13	2	119	957
Rp	12	12	18	1049
Rs	0	229	10	852
Sc	3	52	65	971
Sp	0	14	53	1024

One of the main goals of this paper was to evaluate the fidelity and efficiency of different CNN-based image-processing algorithms in recognizing multi-class steel surface defects. Table 5 shows all utilized models' average values for accuracy, precision, sensitivity, specificity, G-mean1, and F1 scores for all classes of steel surface defects.

Table 5
Average values for performance measurements for all classes of steel surface defects.

Model	Accuracy	Precision	Sensitivity	F1-Score	Specificity	G-Mean
AlexNet	98.64%	90.55%	88.31%	88.88%	99.24%	93.24%
CancerNet	99.02%	93.14%	90.30%	91.51%	99.45%	94.56%
DenseNet	99.55%	96.24%	96.04%	96.10%	99.75%	97.84%
EfficientNet	99.63%	96.74%	97.52%	96.99%	99.80%	98.64%
ExceptionNet	99.08%	94.85%	95.45%	94.88%	99.47%	97.36%
GoogleNet	99.43%	95.49%	95.22%	95.31%	99.69%	97.39%
InceptionNet	99.37%	94.84%	94.77%	94.72%	99.65%	97.12%
LeNet	98.72%	92.42%	89.25%	90.27%	99.27%	93.88%
MobileNet	93.58%	79.99%	76.35%	70.05%	96.33%	82.07%
NasNet	99.55%	96.47%	96.10%	96.26%	99.75%	97.86%
RegNet	99.71%	97.46%	98.17%	97.78%	99.84%	99.00%
ResNet	99.37%	94.88%	94.60%	94.67%	99.65%	97.05%
SuffleNet	98.27%	89.25%	84.87%	86.17%	99.03%	91.18%
SqueexNet	98.82%	92.81%	89.65%	90.98%	99.33%	94.18%
VggNet	99.47%	95.76%	95.85%	95.73%	99.71%	97.73%
ZFNet	98.43%	88.05%	85.28%	86.03%	99.13%	91.12%

Accuracy is a measure of how well a model correctly predicts the class of a given input. It is defined as the ratio of the number of correct predictions to the total number of predictions made by the model. In other words, accuracy is the percentage of correctly classified instances out of the total number of instances. It is a commonly used metric to evaluate the performance of classification models.

$$Accuracy = {TP + TN}/{TP + FN + TN + FP} \quad (1)$$

In the context of steel surface detection, accuracy refers to the ability of a machine learning model to correctly identify and classify surface defects in steel with a high degree of precision. Accurate detection is crucial in ensuring that the quality of the steel is up to standard, as defects in the surface of steel can compromise its structural integrity and cause safety concerns in its end-use applications. In addition, by accurately identifying and classifying surface defects, manufacturers can take appropriate actions to correct the defects and ensure the quality of their products. High accuracy in steel surface detection is important because it ensures that defects are correctly identified and can be addressed, which helps to maintain the quality of the steel and prevent potential failures or safety issues. Most models have scored an accuracy value of more than 98%, so it's fair to say that they have all performed almost equally in this regard. Image preprocessing techniques such as data augmentation can increase the diversity of the training data, which can help to improve the accuracy of all the models.

Precision is a metric used in machine learning and statistics to evaluate the accuracy of a model's positive predictions. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model, i.e., the proportion of correctly predicted positive instances among all instances predicted as positive by the model.

$$Precision = {}^{TP}/_{TP + FP} \qquad (2)$$

Precision is useful in cases where false positives (i.e., instances incorrectly identified as positive by the model) are costly or undesirable. In the context of surface steel defect detection, precision refers to the ability of a detection algorithm to accurately identify only the relevant defects without including false positives. In other words, precision measures the ratio of true positives (i.e., correctly identified defects) to the total number of defects identified by the algorithm (i.e., true positives and false positives). High precision is desirable in surface steel defect detection, as it ensures that the algorithm is not flagging areas of the steel surface that do not contain actual defects. This can help minimize the need for further inspection or rework, which can be costly and time-consuming. RegNet has scored the highest precision value of almost 98% while MobileNet has scored the lowest precision value compared to the other models. MobileNet and other lightweight image classification algorithms, such as ShuffleNet, typically have lower precision than larger, more complex models because they sacrifice some level of accuracy for efficiency. These models are designed to be deployed on mobile devices or other resource-constrained environments, where they have to operate within strict memory and computational limitations. To achieve this, they often use techniques such as quantization, pruning, and other optimization methods that can compromise their performance slightly in favor of reducing their computational cost. Additionally, MobileNet and similar models may use smaller kernel sizes and fewer layers than larger models, making them less capable of capturing intricate details and nuances in the images they process. However, despite their lower precision, these models can still be highly effective in many real-world applications, mainly when speed and efficiency are more important than absolute accuracy.

In machine learning, sensitivity is a performance metric that measures the proportion of actual positive cases correctly identified as positive by the model. It is also known as recall or true positive rate (TPR). The sensitivity score is calculated as the number of true positives divided by the sum of true positives and false negatives. Sensitivity is a vital evaluation metric in machine learning because it helps us understand how well our model can correctly identify positive cases for each class. In addition, it is instrumental in cases where the cost of a false negative is high, such as in medical diagnoses.

$$\text{Sensitivity} = {}^{TP}/_{TP + FN} \tag{3}$$

In the context of detecting steel surface defects, sensitivity can be used as an evaluation metric to measure the performance of a machine learning model in correctly identifying positive cases, i.e., the presence of defects. A high sensitivity means that the model can accurately detect most of the defects, which is a critical characteristic for ensuring the quality control of steel products. For example, in an automated system for inspecting steel surfaces, the sensitivity of the machine learning model can determine how reliable the system is in detecting and identifying defects. High sensitivity can help minimize the number of missed defective products, leading to a better quality of steel products. MobileNet has scored the lowest sensitivity value among all models. The sensitivity of an image classification algorithm depends on its architecture and the specific trade-offs made during its design. In the case of MobileNet, its lower sensitivity can be attributed to the use of depthwise separable convolutions, which are computationally efficient but may not capture fine details as well as other convolutional layers. Additionally, MobileNet may have fewer parameters or layers than other architectures, which can limit its ability to capture complex features and patterns. These design choices make MobileNet a good choice for applications where computational efficiency is a priority, but may not be the best option for applications where high sensitivity is critical. In contrast, RegNet has achieved the highest sensitivity value among all models, and this can be attributed to the fact that RegNet was explicitly designed to achieve high performance and efficiency in image classification tasks. It uses a novel network architecture that emphasizes building a highly modular and scalable network with many small building blocks, allowing for better feature extraction and learning.

Specificity is the proportion of correctly identified negative instances (true negatives) out of all the actual negative instances. In other words, it measures how well the model can correctly identify the classes not present in the target variable. Specificity is a useful metric in evaluating the overall performance of a multi-class classification model, especially when the class distribution is imbalanced or when some classes are more important than others.

$$Specificity = {}^{TN}/_{TN + FP} \tag{4}$$

In the context of steel surface defect detection, specificity refers to the ability of a machine learning model to correctly identify negative samples, i.e., samples that do not have any defects on the surface. High specificity means that the model can correctly identify the absence of defects on the surface, which is important for ensuring the quality of the steel product. However, it is important to note that specificity should be balanced with sensitivity to ensure the model does not miss any defects on the surface while detecting false positives. Most models have scored a specificity value of more than 99% except for MobileNet, so it's fair to say that almost all of them have performed almost equally in this regard. In addition, the inherent different hyperparameters of the models, such as the learning rate, batch size, and regularization parameters, can significantly impact the model's specificity. A well-tuned model can improve the specificity of the model to detect features and patterns in the images. In the case of MobileNet, and as mentioned earlier, MobileNet is designed to be a lightweight convolutional neural network (CNN) architecture suitable for mobile and embedded devices. It achieves this by using depthwise separable convolutions, which consist of a depthwise convolution layer followed by a pointwise convolution layer. While depthwise separable convolutions reduce the number of parameters and computational complexity of the network, they can lead to lower specificity due to the information loss during the depthwise convolution layer. This can cause some features to be missed, resulting in misclassifications and lower specificity.

The F1 score, or F-score or F-measure, measures a classification model's accuracy. The F1 score is the harmonic mean of precision, and sensitivity gives a balanced measure of a model's performance. The F1 score ranges between 0 and 1, with a higher score indicating better performance. In other words, F1 score provides a single score that balances both precision and recall in a classification model, making it a useful metric for evaluating the overall performance of the model.

$$F1 = 2 * \left({}^{Precision\, *\, \text{Sensitivity}} / {}_{Precision\, +\, \text{Sensitivity}} \right) \qquad (5)$$

In the context of steel surface defect detection, F1 score can be used to evaluate the performance of a machine learning model in identifying different types of defects. A higher F1 score indicates that the model is better at correctly identifying the defects, while a lower score indicates that the model is making more errors. Therefore, F1 score is an important metric in ensuring the quality of steel products and preventing defects. RegNet achieved the highest F1 score, while MobileNet achieved the lowest F1 score. RegNet architecture is designed to achieve higher performance than previous models, so it has more layers and wider channels than other models like MobileNet. This allows RegNet to capture more complex patterns and features in the data, which leads to better classification performance and higher F1 scores. On the other hand, MobileNet is a lightweight model that sacrifices some accuracy for faster inference and lower memory usage. This means that MobileNet may be unable to capture all the relevant information in the data, resulting in a lower F1 score than other more powerful models like RegNet.

G-mean1, also known as geometric mean, is a classification performance measure that considers sensitivity and specificity. It is defined as the square root of the product of sensitivity and specificity, where sensitivity is the true positive rate and specificity is the true negative rate. The G-mean is useful in situations where the classes are imbalanced, and there is a need to balance the importance of correctly classifying both the positive and negative samples. It is commonly used in machine learning applications, such as fraud detection, anomaly detection, and medical diagnosis. In summary, the G-mean measures how well a model can identify both positive and negative samples correctly, and it is advantageous in imbalanced classification problems.

$$G - mean1 = \sqrt{Specificity * \text{Sensitivity}}$$

6

In the context of steel surface defect detection, G-mean can be used to assess the overall performance of a defect detection system. This is important in ensuring that only the actual defects are identified while minimizing false alarms or misclassifications. In the context of steel surface defect detection, it is important to achieve a high G-mean to ensure that the classifier can accurately detect defects in all classes, including those that occur less frequently. If the dataset used to train the model is imbalanced, with one type of defect being more common, the G-mean can provide a more accurate measure of how well the model can detect all categories of defects. A high G-mean score indicates that the model can accurately identify all types of defects. In contrast, a low score means that the model may be biased towards certain types of defects and may not perform well in detecting others. RegNet achieved the highest G-mean1 value, while MobileNet achieved the lowest G-mean1 value. RegNet reached the highest G-mean1 value because it simultaneously achieved high sensitivity and specificity, indicating that it was better at identifying positive and negative cases. On the other hand, MobileNet achieved the lowest G-mean value because it had lower sensitivity and specificity, meaning it struggled to identify both positive and negative cases accurately. This could be because MobileNet has fewer parameters, which may limit its ability to capture complex features in the data accurately. Figure 6 shows the performance measurement curves of accuracy, precision, sensitivity, specificity, G-mean1, and F1 scores for all classes of steel surface defects.

Table 6 shows the number of epochs used to train each model to achieve the lowest loss and highest accuracy values and the time needed per epoch. An epoch refers to a single iteration through the entire dataset during the training phase of a neural network. During each epoch, the neural network processes all the training examples in the dataset and updates its weights and biases accordingly to minimize the loss function. After completing one epoch, the model has seen all the training data once. The number of epochs is a hyperparameter determined before training the model. It controls how many times the model will go through the entire dataset. Choosing the appropriate number of epochs is vital to ensure that the model is not underfitting or overfitting the data. Underfitting occurs when the model has not learned enough from the training data. At the same time, overfitting occurs when the model has memorized the training data and does not generalize well to new data.

Table 6
Time and number of epochs for all models.

Model	Number of Epochs	Time per Epoch (s)	Total Time (min)
AlexNet	109	5	9.08
CancerNet	67	6	6.70
DenseNet	66	11	12.10
EfficientNet	55	16	14.67
ExceptionNet	108	19	34.20
GoogleNet	51	36	30.60
InceptionNet	92	8	12.27
LeNet	72	5	6.00
MobileNet	40	35	23.33
NasNet	154	16	41.07
RegNet	47	14	10.97
ResNet	47	7	5.48
SuffleNet	41	6	4.10
SqueexNet	52	5	4.33
VggNet	61	5	5.08
ZFNet	89	12	17.80

The number of epochs used during training can have a significant impact on the computational resources and cost required for the task. Running more epochs can improve model performance but also increase computational time and resource requirements. On the other hand, using fewer epochs may save computational resources and time, but it can also result in an underfit or poorly performing model. The trade-off between the number of epochs used and the performance of the model must be carefully considered to balance the need for accuracy and computational efficiency. In some cases, it may be possible to use techniques like early stopping, where the training process is stopped before the model begins to overfit, to reduce the number of epochs required while still achieving high accuracy. Decision-makers can use this information to compare the costs and benefits of different machine-learning models with varying numbers of epochs. They can evaluate the trade-offs between the model performance and the required resources and cost, and select the model that best fits their needs and constraints.

Steel surface defect detection plays a crucial role in achieving a Lean in metal production. Lean metal production aims to minimize waste, reduce costs, and maximize efficiency in the manufacturing process. Detecting surface defects early in the production process enables corrective action to be taken promptly, thereby reducing waste and improving efficiency. In addition, by using computer-based vision techniques, defects can be detected with high accuracy, reducing the need for manual inspection, which can be time-consuming and prone to errors. This increases productivity and reduces costs, contributing to a Lean production process. Moreover, early detection of surface defects helps prevent defects from propagating further down the supply-chain line, reducing the risk of producing defective products. This reduces the need for returns and recalls, further contributing to cost savings and improving overall efficiency. With a more efficient and effective defect detection system, manufacturers can also optimize their quality control processes, reduce variability in production, and ultimately improve customer satisfaction. Additionally, a Lean approach to metal production can involve continuous improvement efforts that rely on data-driven decision-making, which is enabled by defect detection technologies to observe patterns and frequency in production defects and can lead to further improvements in production efficiency and quality. In summary, computer-based vision inspection for metal surface defects is a critical component of a Lean system in metal production. Detecting defects early and accurately leads to reduced waste, increased efficiency, and cost savings.

5.1 Limitations

While computer-based vision techniques have shown promising results in detecting and classifying steel surface defects, their use has several limitations. One major limitation is the requirement for large amounts of annotated data to train the models, which can be time-consuming and costly. Additionally, there may be variations in lighting, angle, and surface texture that can affect the accuracy of the models. Moreover, computer-based vision models may struggle to detect subtle defects that are not easily distinguishable from the background or noise. The models may also be affected by overlapping or complex defects that are difficult to separate and classify. Furthermore, the performance of these models may depend on the quality and resolution of the images used for training and testing. In some cases, the models may not generalize well to new images not included in the training dataset, leading to reduced accuracy and reliability. Also, implementing and maintaining computer-based vision systems can be costly and require specialized hardware and software, which may be a barrier to adoption for some industries. Finally, an existing model can tailor to a particular material or defect. So sometimes, these approaches are not versatile enough to be applied to various types of defects in different materials. As a result, investigating the detection and classification of surface defects could be an important target for steel producers.

6. Conclusion

This paper presents a comprehensive study of CNN-based deep learning image algorithms in detecting multi-class surface defects on steel. Overall, our findings demonstrate the effectiveness of CNN-based models in accurately detecting and classifying surface defects on steel, with RegNet performing the best in terms of all utilized performance measurement metrics. These results have important implications for developing automated systems for surface defect detection in industrial settings. In addition, image classification algorithms can contribute to Lean metal production by identifying defects or anomalies in the manufacturing process, which can be used to reduce waste and increase efficiency. Overall, image classification algorithms can help streamline manufacturing, reduce waste, and improve product quality in Lean metal production.

Declarations

Funding

The reported research work received partial financial support from U.S. Office of Naval Research MEEP Program (Award Number: N00014-19-1-2728), U.S. Department of Energy/NNSA (Award Number: DE-NA0004003), as well as from the Lutcher Brown Distinguished Chair Professorship fund of the University of Texas at San Antonio.

Data transparency

All the data have been presented in the manuscript.

Ethical approval

The paper follows the guidelines of the Committee on Publication Ethics (COPE).

Consent to participate

The authors declare that they all consent to participate this research.

Consent for publication

The authors declare that they all consent to publish the manuscript.

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

Authors Contribution

Mohammad Shahin took care of conceptualization, methodology, investigation, original draft, methodology, and final revisions. Ali Hosseinzadeh took care of the dataset and the review. Mazdak Maghanaki conducted initial literature review. F. Frank Chen contributed to resources and coordination efforts. Finally, all authors read and approved the final manuscript.

References

1. Radecka K (2022) Improving Bu Siness Maintenance Processes Using Lean Manufacturing Tools – Case Study. Sci Pap Silesian Univ Technol Organ Manag Zesz Nauk Politech Slaskiej Ser Organ Zarzadzanie 395–405. https://doi.org/10.29119/1641-3466.2022.156.27

2. Nguyen D-M, Duong T-K, Enterprises (2022) Characteristics and Lean Outcome: An Empirical Evidence from Vietnam Manufacturing Enterprises. Management Systems in Production Engineering,2022, Volume 30, Issue 2, pp. 98–108,DOI 10.2478/mspe-2022-0013Volumen 30 - Edición 2 (June 2022)
3. U R, C K, D. S, A. P (2022) Analysis of the Systemic Approach to the Concept of Lean Manufacturing - Results of Empirical Research: ANALIZA SYSTEMOWEGO PODEJŚCIA DO KONCEPCJI LEAN MANUFACTURING - WYNIKI BADAŃ EMPIRYCZNYCH. 25:375–395. https://doi.org/10.17512/pjms.2022.25.2.24
4. García Alcaraz JL, Morales García AS, Díaz Reza JR et al (2022) Machinery Lean Manufacturing Tools for Improved Sustainability: The Mexican Maquiladora Industry Experience. Mathematics 10:1468. https://doi.org/10.3390/math10091468
5. Singer E, Becker K (2013) A single-source content management system for lean manufacturing. Int J Lean Six Sigma 4:83–103. https://doi.org/10.1108/20401461311310535
6. Sancha C, Wiengartena F, Longonia A, Pagell M (2019) The moderating role of temporary work on the performance of lean manufacturing systems. Int J Prod Research: Vol 58. https://doi.org/10.1080/00207543.2019.1651458,Pages 4285–4305, Aug 2019
7. Tripathi V, Chattopadhyaya S, Mukhopadhyay AK et al (2022) Development of a Data-Driven Decision-Making System Using Lean and Smart Manufacturing Concept in Industry 4.0. A Case Study
8. ADREEVA T, SHVIDENKO D, POPOVA L (2021) Development of a Lean Quality Management System: An Integrated Management System | SpringerLink. IHNS 2010. Lecture Notes in Business Information Processing, vol 46. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12494-5_13
9. Nascimento DL, de Goncalvez Quelhas M, Gusmão Caiado OL RG, et al (2020) A lean six sigma framework for continuous and incremental improvement in the oil and gas sector. Int J Lean Six Sigma 11:577–595. https://doi.org/10.1108/IJLSS-02-2019-0011
10. Argiyantari B, Simatupang TM, Basri MH (2022) Transportation performance improvement through lean thinking implementation. Int J Lean Six Sigma 13:622–647. https://doi.org/10.1108/IJLSS-06-2020-0075
11. Katarzyna RADECKA, IMPROVING BU SINESS MAINTENANCE PROCESSES USING LEAN, MANUFACTURING TOOLS. SCIENTIFIC PAPERS OF SILESIAN UNIVERSITY OF TECHNOLOGY ORGANIZATION AND MANAGEMENT SERIES, NO. 156,pp396–405,2022,doi:0000-0001-8389-5652
12. Sancha C, Wiengarten F, Longoni A, Pagell M (2020) The moderating role of temporary work on the performance of lean manufacturing systems. Int J Prod Res 58:4285–4305. https://doi.org/10.1080/00207543.2019.1651458
13. Gupta S, Jain SK (2013) A literature review of lean manufacturing. Int J Manag Sci Eng Manag 8:241–249. https://doi.org/10.1080/17509653.2013.825074

14. Swarnakar V, Vinodh S (2016) Deploying Lean Six Sigma framework in an automotive component manufacturing organization. Int J Lean Six Sigma 7:267–293. https://doi.org/10.1108/IJLSS-06-2015-0023
15. Medonos M, LEANNESS LEVEL OF MANUFACTURING COMPANIES - A SURVEY ON LEAN MANUFACTURING IMPLEMENTATION (2021) Acta Acad Karviniensia 21:54–65. https://doi.org/10.25142/aak.2021.012
16. Chauhan G, Chauhan V (2019) A phase-wise approach to implement lean manufacturing. Int J Lean Six Sigma 10:106–122. https://doi.org/10.1108/IJLSS-09-2017-0110
17. Venugopal V, Saleeshya PG (2019) Manufacturing system sustainability through lean and agile initiatives. Int J Sustain Eng 12:159–173. https://doi.org/10.1080/19397038.2019.1566411
18. Kalemkerian F, Santos J, Tanco M et al (2022) Analysing the alignment between the Green Lean and Circular strategies: towards a Circular Lean approach. https://doi.org/10.1108/JMTM-11-2021-0480. J Manuf Technol Manag ahead-of-print
19. Leksic I, Stefanic N, Veza I (2020) The impact of using different lean manufacturing tools on waste reduction. Adv Prod Eng Manag 15:81–92. https://doi.org/10.14743/apem2020.1.351
20. Vaněček D, Pech M, Rost M (2018) Innovation and Lean Production. Acta Univ Agric Silvic Mendel Brun 66:595–603. https://doi.org/10.11118/actaun201866020595
21. Putri NT, Dona LS (2019) Application of lean manufacturing concept for redesigning facilities layout in Indonesian home-food industry: A case study. TQM J 31:815–830. https://doi.org/10.1108/TQM-02-2019-0033
22. Vinodh SKB, Devarapu S, Siddhamshetty S G (2017) Application of Lean approach for reducing weld defects in a valve component: a case study. Int J Lean Six Sigma 8:181–209. https://doi.org/10.1108/IJLSS-01-2016-0001
23. Suhardi B, Anisa N, Laksono PW (2019) Minimizing waste using lean manufacturing and ECRS principle in Indonesian furniture industry. Cogent Eng 6:1567019. https://doi.org/10.1080/23311916.2019.1567019
24. Panat R, Dimitrova V, Selvamuniandy TS et al (2014) The application of Lean Six Sigma to the configuration control in Intel's manufacturing R&D environment. Int J Lean Six Sigma 5:444–459. https://doi.org/10.1108/IJLSS-02-2014-0004
25. Almira S, Bašić H, Lulic H, Nakic A (2020) Application of the Lean Concept for Analysis and Optimization of the Automobile Filter Production Company. 31:0708–0716. https://doi.org/10.2507/31st.daaam.proceedings.098
26. Kumar Arya A, Kumar Jain S (2014) Impacts of Kaizen in a small-scale industry of India: a case study. Int J Lean Six Sigma 5:22–44. https://doi.org/10.1108/IJLSS-03-2013-0019
27. Sharma A, Bhanot N, Gupta A, Trehan R (2021) Application of Lean Six Sigma framework for improving manufacturing efficiency: a case study in Indian context. Int J Product Perform Manag 71:1561–1589. https://doi.org/10.1108/IJPPM-05-2020-0223

28. Setiawan S, Setiawan I, Jaqin C et al (2021) Integration of Waste Assessment Model and Lean Automation to Improve Process Cycle Efficiency in the Automotive Industry. Qual Innov Prosper 25:48–64. https://doi.org/10.12776/qip.v25i3.1613
29. Purushothaman MB, Seadon J, Moore D (2022) A relationship between bias, lean tools, and waste. Int J Lean Six Sigma 13:897–936. https://doi.org/10.1108/IJLSS-03-2021-0045
30. Jasti NVK, Kota S, Sangwan KS (2019) An application of value stream mapping in auto-ancillary industry: a case study. TQM J 32:162–182. https://doi.org/10.1108/TQM-11-2018-0165
31. Torri M, Kundu K, Frecassetti S, Rossini M (2021) Implementation of lean in IT SME company: an Italian case. Int J Lean Six Sigma 12:944–972. https://doi.org/10.1108/IJLSS-05-2020-0067
32. Caldeira T, Ciarelli PM, Neto GA (2020) Industrial Optical Character Recognition System in Printing Quality Control of Hot-Rolled Coils Identification. J Control Autom Electr Syst 31:108–118. https://doi.org/10.1007/s40313-019-00551-1
33. Goshime Y, Kitaw D, Jilcha K (2018) Lean manufacturing as a vehicle for improving productivity and customer satisfaction: A literature review on metals and engineering industries. Int J Lean Six Sigma 10:691–714. https://doi.org/10.1108/IJLSS-06-2017-0063
34. Pozzi R, Cannas VG, Ciano MP (2021) Linking data science to lean production: a model to support lean practices. Int J Prod Res 1–22. https://doi.org/10.1080/00207543.2021.1946192
35. Ltd I-IB (2021) Artificial Intelligence-driven Big Data Analytics, Real-Time Sensor Networks, and Product Decision-Making Information Systems in Sustainable Manufacturing Internet of Things. Econ Manag Financ Mark 16:81–93
36. Sharma S, Gahlawat VK, Rahul K et al (2021) Sustainable Innovations in the Food Industry through Artificial Intelligence and Big Data Analytics. Logistics 5:66. https://doi.org/10.3390/logistics5040066
37. Santos BP, Enrique DV, Maciel VBP et al (2021) The Synergic Relationship Between Industry 4.0 and Lean Management: Best Practices from the Literature. Manag Prod Eng Rev MPER 12:94–107. https://doi.org/10.24425/mper.2021.136875
38. Buer S-V, Semini M, Strandhagen JO, Sgarbossa F (2021) The complementary effect of lean manufacturing and digitalisation on operational performance. Int J Prod Res 59:1976–1992. https://doi.org/10.1080/00207543.2020.1790684
39. Ejsmont K, Gladysz B, Corti D et al (2020) Towards 'Lean Industry 4.0 – Current trends and future perspectives. Cogent Bus Manag 7:1781995. https://doi.org/10.1080/23311975.2020.1781995
40. Tripathi V, Chattopadhyaya S, Mukhopadhyay AK et al (2022) Development of a Data-Driven Decision-Making System Using Lean and Smart Manufacturing Concept in Industry 4.0: A Case Study. Math Probl Eng 2022:e3012215. https://doi.org/10.1155/2022/3012215
41. Chiarini A, Kumar M (2021) Lean Six Sigma and Industry 4.0 integration for Operational Excellence: evidence from Italian manufacturing companies. Prod Plan Control 32:1084–1101. https://doi.org/10.1080/09537287.2020.1784485

42. Demirkol İ, Al-Futaıh AA (2020) The Relationship Between Industry 4.0 and Lean Production: An Empirical Study on Bursa Manufacturing Industry. https://doi.org/10.20491/isarder.2020.897
43. Xi T, Benincá IM, Kehne S et al (2021) Tool wear monitoring in roughing and finishing processes based on machine internal data. Int J Adv Manuf Technol 113:3543–3554. https://doi.org/10.1007/s00170-021-06748-6
44. Vertical data continuity with lean edge analytics for industry 4.0 production.Comput Ind125:103389. https://doi.org/10.1016/j.compind.2020.103389
45. Wierschem DC, Jimenez JA, Méndez Mediavilla FA (2020) A motion capture system for the study of human manufacturing repetitive motions. Int J Adv Manuf Technol 110:813–827. https://doi.org/10.1007/s00170-020-05822-9
46. Vlachos IP, Pascazzi RM, Zobolas G et al (2021) Lean manufacturing systems in the area of Industry 4.0: a lean automation plan of AGVs/IoT integration. Prod Plan Control 1–14. https://doi.org/10.1080/09537287.2021.1917720
47. Wang J, Xu C, Zhang J, Zhong R (2022) Big data analytics for intelligent manufacturing systems: A review. J Manuf Syst 62:738–752. https://doi.org/10.1016/j.jmsy.2021.03.005
48. Demirdöğen G, Diren NS, Aladağ H, Işık Z (2021) Lean Based Maturity Framework Integrating Value, BIM and Big Data Analytics: Evidence from AEC Industry. Sustainability 13:10029. https://doi.org/10.3390/su131810029
49. Shahin M, Chen FF, Bouzary H, Krishnaiyer K (2020) Integration of Lean practices and Industry 4.0 technologies: smart manufacturing for next-generation enterprises. Int J Adv Manuf Technol 107:2927–2936. https://doi.org/10.1007/s00170-020-05124-0
50. Belhadi A, Zkik K, Cherrafi A et al (2019) Understanding Big Data Analytics for Manufacturing Processes: Insights from Literature Review and Multiple Case Studies. Comput Ind Eng 137:106099. https://doi.org/10.1016/j.cie.2019.106099
51. Ratnayake RMC, Chaudry O (2017) Maintaining sustainable performance in operating petroleum assets via a lean-six-sigma approach. Int J Lean Six Sigma 8:33–52. https://doi.org/10.1108/IJLSS-11-2015-0042
52. Powell D, Lundeby S, Chabada L, Dreyer H (2017) Lean Six Sigma and environmental sustainability: the case of a Norwegian dairy producer. Int J LEAN SIX SIGMA 8:53–64. https://doi.org/10.1108/IJLSS-06-2015-0024
53. Belhadi A, Kamble SS, Zkik K et al (2020) The integrated effect of Big Data Analytics, Lean Six Sigma and Green Manufacturing on the environmental performance of manufacturing companies: The case of North Africa. J Clean Prod 252:119903. https://doi.org/10.1016/j.jclepro.2019.119903
54. Indrawati S, Azzam A, Ramdani A (2019) Manufacturing Efficiency Improvement Through Lean Manufacturing Approach: A Case Study in A Steel Processing Industry. In: IOP Conference Series: Materials Science and Engineering. Institute of Physics Publishing
55. Wahaba A, Mukhtara M, Sulaiman R (2013) A Conceptual Model of Lean Manufacturing Dimensions. 4th Int Conf Electr Eng Inform ICEEI 2013 11:1292–98

56. Qian K (2019) Automated Detection of Steel Defects via Machine Learning based on Real-Time Semantic Segmentation. Proc 3rd Int Conf Video Image Process 42–46. https://doi.org/10.1145/3376067.3376113
57. Luo Q, Fang X, Liu L et al (2020) Automated Visual Defect Detection for Flat Steel Surface: A Survey. IEEE Trans Instrum Meas Instrum Meas IEEE Trans IEEE Trans Instrum Meas 69:626–644. https://doi.org/10.1109/TIM.2019.2963555
58. Akhyar F, Lin C-Y, Muchtar K et al (2019) High Efficient Single-stage Steel Surface Defect Detection. 2019 16th IEEE Int Conf Adv Video Signal Based Surveill AVSS Adv Video Signal Based Surveill AVSS 2019 16th IEEE Int Conf On 1–4. https://doi.org/10.1109/AVSS.2019.8909834
59. Guo B, Wang Y, Zhen S et al SPEED:Semantic Prior and Extremely Efficient Dilated Convolution Network for Real-Time Metal Surface Defects Detection.IEEE Trans Ind Inform Ind Inform IEEE Trans IEEE Trans Ind Inf PP:1–11. https://doi.org/10.1109/TII.2022.3233674
60. Shahin M, Chen FF, Hosseinzadeh A et al (2023) Enhanced safety implementation in 5S + 1 via object detection algorithms. Int J Adv Manuf Technol 1–21. https://doi.org/10.1007/s00170-023-10970-9
61. Liu W, Yang X, Gao H (2022) A Novel Industrial Chip Parameters Identification Method Based on Cascaded Region Segmentation for Surface-Mount Equipment. IEEE Trans Ind Electron Ind Electron IEEE Trans IEEE Trans Ind Electron 69:5247–5256. https://doi.org/10.1109/TIE.2021.3082072
62. Bao Y, Song K, Liu J et al (2021) Triplet-Graph Reasoning Network for Few-Shot Metal Generic Surface Defect Segmentation. IEEE Trans Instrum Meas 70. https://doi.org/10.1109/TIM.2021.3083561
63. Utsunomiya H, Hara K, Matsumoto R, Azushima A (2014) Formation mechanism of surface scale defects in hot rolling process. CIRP Ann 63:261–264. https://doi.org/10.1016/j.cirp.2014.03.022
64. Sarkar PP, Dhua SK, Thakur SK, Rath S (2017) Analysis of the Surface Defects in a Hot-Rolled Low-Carbon C–Mn Steel Plate. J Fail Anal Prev 17:545–553. https://doi.org/10.1007/s11668-017-0281-8
65. Bahrami A, Kiani Khouzani M, Mokhtari SA et al (2019) Root Cause Analysis of Surface Cracks in Heavy Steel Plates during the Hot Rolling Process. Metals 9:801. https://doi.org/10.3390/met9070801
66. Shi Y, Liang Q, Wen L (2022) Causes and Prevention Measures of Bright Pitting Defects on Surface of Hot-Rolled Galvalume Steel Sheets. Int J Heat Technol 40:685–692. https://doi.org/10.18280/ijht.400304
67. Wen X, Shan J, He Y, Song K (2023) Steel Surface Defect Recognition: A Survey. Coat 2079–6412 13:17. https://doi.org/10.3390/coatings13010017
68. Yotsuji J, Koshihara T (2014) Detection system for inclusion defects in hot-rolled steel plates using MFLT with two different magnetizing strengths. AIP Conf Proc 1581:1315–1319. https://doi.org/10.1063/1.4864973
69. He Y, Wen X, Xu J (2022) A Semi-Supervised Inspection Approach of Textured Surface Defects under Limited Labeled Samples. Coatings 12:1707. https://doi.org/10.3390/coatings12111707

70. Chakraborty A, Mondal A, Agnihotri S et al (2016) Investigation of a surface defect and its elimination in automotive grade galvannealed steels. Eng Fail Anal 66:455–467. https://doi.org/10.1016/j.engfailanal.2016.05.010
71. Qi S, Yang J, Zhong Z (2020) A Review on Industrial Surface Defect Detection Based on Deep Learning Technology. In: Proceedings of the 2020 3rd International Conference on Machine Learning and Machine Intelligence. Association for Computing Machinery, New York, NY, USA, pp 24–30
72. Althubiti SA, Alenezi F, Shitharth S et al (2022) Circuit Manufacturing Defect Detection Using VGG16 Convolutional Neural Networks. Wirel Commun Mob Comput 1–10. https://doi.org/10.1155/2022/1070405
73. Kelm A, Laußat L, Meins-Becker A et al (2013) Mobile passive Radio Frequency Identification (RFID) portal for automated and rapid control of Personal Protective Equipment (PPE) on construction sites. Autom Constr 36:38–52. https://doi.org/10.1016/j.autcon.2013.08.009
74. Kolar Z, Chen H, Luo X (2018) Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images. Autom Constr 89:58–70. https://doi.org/10.1016/j.autcon.2018.01.003
75. Wang H, Pan C, Guo X et al (2021) From object detection to text detection and recognition: A brief evolution history of optical character recognition. WILEY Interdiscip Rev-Comput Stat 13:e1547. https://doi.org/10.1002/wics.1547
76. Shahin M, Chen FF, Bouzary H et al (2022) A novel fully convolutional neural network approach for detection and classification of attacks on industrial IoT devices in smart manufacturing systems. Int J Adv Manuf Technol 1–13. https://doi.org/10.1007/s00170-022-10259-3
77. Ciaburro G (2017) Neural Networks with R. Packt Publishing
78. Bian X, Lim SN, Zhou N (2016) Multiscale fully convolutional network with application to industrial inspection. 2016 IEEE Winter Conf Appl Comput Vis WACV Appl Comput Vis WACV 2016 IEEE Winter Conf On 1–8. https://doi.org/10.1109/WACV.2016.7477595
79. Yu Z, Wu X, Gu X (2017) Fully Convolutional Networks for Surface Defect Inspection in Industrial Environment. In: Liu M, Chen H, Vincze M (eds) Computer Vision Systems. Springer International Publishing, Cham, pp 417–426
80. Tabernik D, Šela S, Skvarč J, Skočaj D (2020) Segmentation-based deep-learning approach for surface-defect detection. J Intell Manuf 31:759–776. https://doi.org/10.1007/s10845-019-01476-x
81. Liong S-T, Zheng D, Huang Y-C, Gan YS (2020) Leather defect classification and segmentation using deep learning architecture. Int J Comput Integr Manuf 33:1105–1117. https://doi.org/10.1080/0951192X.2020.1795928
82. Delconte F, Ngo P, Debled-Rennesson I et al (2021) Tree Defect Segmentation Using Geometric Features and CNN. In: Kerautret B, Colom M, Krähenbühl A et al (eds) Reproducible Research in Pattern Recognition. Springer International Publishing, Cham, pp 80–100
83. Xie Y, Zhu F, Fu Y (2020) Main-Secondary Network for Defect Segmentation of Textured Surface Images. 2020 IEEE Winter Conf Appl Comput Vis WACV Appl Comput Vis WACV 2020 IEEE Winter

Conf On 3520–3529. https://doi.org/10.1109/WACV45572.2020.9093578

84. Bruna J, Zaremba W, Szlam A, LeCun Y (2013) Spectral Networks and Locally Connected Networks on Graphs. ArXiv E-Prints arXiv:1312.6203. https://doi.org/10.48550/arXiv.1312.6203
85. Zhao L, Song Y, Zhang C et al (2020) T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. IEEE Trans Intell Transp Syst Intell Transp Syst IEEE Trans IEEE Trans Intell Transp Syst 21:3848–3858. https://doi.org/10.1109/TITS.2019.2935152
86. Yann LeCun, Bottou L, Bengio Y, Haffner P (2001) Chap. 9: Gradient-Based Learning Applied to Document Recognition.Intell Signal Process306–351
87. Refianti R, Mutiara A, Priyandini R p (2019) Classification of melanoma skin cancer using convolutional neural network. Int J Adv Comput Sci Appl 10:409–417. https://doi.org/10.14569/IJACSA.2019.0100353
88. Jain N, Nandakumar K, Ratha N et al (2021) CryptInfer: Enabling Encrypted Inference on Skin Lesion Images for Melanoma Detection. https://doi.org/10.1145/3486001.3486233. First Int Conf AI-ML-Syst 1–7
89. Behura A (2021) 1 - Congruence of deep learning in biomedical engineering: future prospects and challenges. Handb Deep Learn Biomed Eng 1–24. https://doi.org/10.1016/B978-0-12-823014-5.00003-X
90. Alam TM, Shaukat K, Khelifi A et al (2022) Disease diagnosis system using IoT empowered with fuzzy inference system.Comput Mater Contin5305–5319
91. Yang X, Khushi M, Shaukat K (2020) Biomarker CA125 Feature Engineering and Class Imbalance Learning Improves Ovarian Cancer Prediction. In: 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). pp 1–6
92. Tan M, Le QV (2020) EfficientNet:Rethinking Model Scaling for Convolutional Neural Networks
93. Tama BA, Vania M, Kim I, Lim S (2022) An EfficientNet-Based Weighted Ensemble Model for Industrial Machine Malfunction Detection Using Acoustic Signals. IEEE Access 10:34625–34636. https://doi.org/10.1109/ACCESS.2022.3160179
94. Ab Wahab MN, Nazir A, Zhen Ren AT et al (2021) Efficientnet-Lite and Hybrid CNN-KNN Implementation for Facial Expression Recognition on Raspberry Pi. IEEE Access 9:134065–134080. https://doi.org/10.1109/ACCESS.2021.3113337
95. Ye M, Ruiwen N, Chang Z et al (2021) A Lightweight Model of VGG-16 for Remote Sensing Image Classification. IEEE J Sel Top Appl Earth Obs Remote Sens 14:6916–6922. https://doi.org/10.1109/JSTARS.2021.3090085
96. Chollet F (2018) Deep learning with Python. Manning Publications
97. Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: A simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
98. Zan X, Zhang X, Xing Z et al (2020) Automatic Detection of Maize Tassels from UAV Images by Combining Random Forest Classifier and VGG16. Remote Sens 12:3049.

https://doi.org/10.3390/rs12183049

99. Yang H, Ni J, Gao J et al (2021) A novel method for peanut variety identification and classification by Improved VGG16. Sci Rep 11:15756. https://doi.org/10.1038/s41598-021-95240-y
100. Chollet F (2017) Xception: Deep Learning with Depthwise Separable Convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 1800–1807
101. Chen B, Liu X, Zheng Y et al (2022) A Robust GAN-Generated Face Detection Method Based on Dual-Color Spaces and an Improved Xception. IEEE Trans Circuits Syst Video Technol 32:3527–3538. https://doi.org/10.1109/TCSVT.2021.3116679
102. Stark T, Wurm M, Zhu XX, Taubenböck H (2020) Satellite-Based Mapping of Urban Poverty With Transfer-Learned Slum Morphologies. IEEE J Sel Top Appl Earth Obs Remote Sens 13:5251–5263. https://doi.org/10.1109/JSTARS.2020.3018862
103. Thitisiriwech K, Panboonyuen T, Kantavat P et al (2022) The Bangkok Urbanscapes Dataset for Semantic Urban Scene Understanding Using Enhanced Encoder-Decoder With Atrous Depthwise Separable A1 Convolutional Neural Networks. IEEE Access 10:59327–59349. https://doi.org/10.1109/ACCESS.2022.3176712
104. He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 770–778
105. Sun L, Liang K, Song Y, Wang Y (2021) An Improved CNN-Based Apple Appearance Quality Classification Method With Small Samples. IEEE Access 9:68054–68065. https://doi.org/10.1109/ACCESS.2021.3077567
106. Karaduman G, Akin E (2020) A Deep Learning Based Method for Detecting of Wear on the Current Collector Strips' Surfaces of the Pantograph in Railways. IEEE Access 8:183799–183812. https://doi.org/10.1109/ACCESS.2020.3029555
107. Alsharman N, Jawarneh, I(2) (2020) GoogleNet CNN neural network towards chest CT-coronavirus medical image classification. J Comput Sci 16:620–625. https://doi.org/10.3844/JCSSP.2020.620.625
108. Reis HC, Turk V, Khoshelham K, Kaya S (2022) InSiNet: a deep convolutional approach to skin cancer detection and segmentation. Med Biol Eng Comput 60:643–662. https://doi.org/10.1007/s11517-021-02473-0
109. Szegedy C, Vanhoucke V, Ioffe S et al (2016) Rethinking the Inception Architecture for Computer Vision. pp 2818–2826
110. Xu Z, Wang Y, Xiong W, Wang Z (2022) A Novel Attentional Feature Fusion with Inception Based on Capsule Network and Application to the Fault Diagnosis of Bearing with Small Data Samples. MACHINES 10:789. https://doi.org/10.3390/machines10090789
111. Yang M, Wang H, Hu K et al (2022) An Inception–Attention-Module-Based Network for Classifying Underwater Images From Others. IEEE J Ocean Eng Ocean Eng IEEE J IEEE J Ocean Eng 47:704–717. https://doi.org/10.1109/JOE.2021.3126090

112. Upadhyay SK, Kumar A (2022) An Accurate and Automated plant disease detection system using transfer learning based Inception V3Model. 2022 2nd Int Conf Adv Comput Innov Technol Eng ICACITE Adv Comput Innov Technol Eng ICACITE 2022 2nd Int Conf On 1144–1151. https://doi.org/10.1109/ICACITE53722.2022.9823559
113. Howard AG, Zhu M, Chen B et al (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. https://doi.org/10.48550/arXiv.1704.04861
114. Hartanto CA, Wibowo A (2020) Development of Mobile Skin Cancer Detection using Faster R-CNN and MobileNet v2 Model. 2020 7th Int Conf Inf Technol Comput Electr Eng ICITACEE Inf Technol Comput Electr Eng 2020 7th Int Conf On 58–63. https://doi.org/10.1109/ICITACEE50144.2020.9239197
115. Liu J (2022) VGG, MobileNet and AlexNet on Recognizing Skin Cancer Symptoms. 2022 3rd Int Conf Electron Commun Artif Intell IWECAI Electron Commun Artif Intell IWECAI 2022 3rd Int Conf IWECAI. 525–528. https://doi.org/10.1109/IWECAI55315.2022.00107
116. Abu Al-Haija Q (2022) Leveraging ShuffleNet transfer learning to enhance handwritten character recognition. Gene Expr Patterns 45. https://doi.org/10.1016/j.gep.2022.119263
117. Hou W, Guo H, Yan B et al (2022) Tool wear state recognition under imbalanced data based on WGAN-GP and lightweight neural network ShuffleNet. J Mech Sci Technol 1–17. https://doi.org/10.1007/s12206-022-0912-1
118. Rezaee K, Mousavirad S, j., Khosravi M et al (2022) An Autonomous UAV-Assisted Distance-Aware Crowd Sensing Platform Using Deep ShuffleNet Transfer Learning. IEEE Trans Intell Transp Syst Intell Transp Syst IEEE Trans IEEE Trans Intell Transp Syst 23:9404–9413. https://doi.org/10.1109/TITS.2021.3119855
119. Chen W, Tao C (2022) Object-Oriented Extraction of Land Occupation Types in Mining Areas by Using Densenet. IGARSS 2022–2022 IEEE Int Geosci Remote Sens Symp Geosci Remote Sens Symp IGARSS 2022–2022. IEEE Int 6348–6351. https://doi.org/10.1109/IGARSS46834.2022.9883297
120. Sahib I, AlAsady TAA (2022) Deep fake Image Detection based on Modified minimized Xception Net and DenseNet. 2022 5th Int Conf Eng Technol Its Appl IICETA Eng Technol Its Appl IICETA 2022 5th Int Conf On 355–360. https://doi.org/10.1109/IICETA54559.2022.9888278
121. Liu H, Liu C, Zhao T, Liu Y (2021) Bird Song Classification Based on Improved Bi-LSTM-DenseNet Network. 2021 4th Int Conf Robot Control Autom Eng RCAE Robot Control Autom Eng RCAE 2020 3rd Int Conf On 152–155. https://doi.org/10.1109/RCAE53607.2021.9638962
122. Xu J, Pan Y, Pan X et al (2022) RegNet: Self-Regulated Network for Image Classification. IEEE Trans NEURAL Netw Learn Syst. https://doi.org/10.1109/TNNLS.2022.3158966
123. Zhu L, Baolin D, Xiaomeng Z et al (2022) Surface Defect Detection Method Based on Improved Semisupervised Multitask Generative Adversarial Network. Sci Program 1–17. https://doi.org/10.1155/2022/4481495
124. Mujtaba DF, Mahapatra NR, Similar Fish Species Identification (2021) Convolutional Neural Networks for Morphologically. 2021 Int Conf Comput Sci Comput Intell CSCI Comput Sci Comput Intell CSCI

2021 Int Conf CSCI 1553–1559. https://doi.org/10.1109/CSCI54926.2021.00303
125. Peng L, Wang C, Tian G et al (2022) Analysis of CT scan images for COVID-19 pneumonia based on a deep ensemble framework with DenseNet, Swin transformer, and RegNet. Front Microbiol 13:995323. https://doi.org/10.3389/fmicb.2022.995323
126. Abbas Q, Gul, A(2) (2023) Detection and Classification of Malignant Melanoma Using Deep Features of NASNet. SN Comput Sci 4. https://doi.org/10.1007/s42979-022-01439-9
127. Naskinova I (2022) Transfer learning with NASNet-Mobile for Pneumonia X-ray classification. Asian-Eur J Math. https://doi.org/10.1142/S1793557122502400
128. Gore S 1), Azad R k. (1 2) (2022) CancerNet: a unified deep learning network for pan-cancer diagnostics.BMC Bioinformatics23:. https://doi.org/10.1186/s12859-022-04783-y
129. Platt R (2000) Breast Cancer on CancerNet. Breast Cancer Res. https://doi.org/10.1186/bcr-2000-2-webreport0001. 2:
130. Meng X1), Wang J et al (2015) 1 2), Yuan C(1), CancerNet: A database for decoding multilevel molecular interactions across diverse cancer types. Oncogenesis 4:. https://doi.org/10.1038/oncsis.2015.40
131. Zeiler MD, Fergus R (2014) Visualizing and Understanding Convolutional Networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision – ECCV 2014. Springer International Publishing, Cham, pp 818–833
132. Fu L 2,3), Feng Y 4), Majeed Y et al (2018) (5), Kiwifruit detection in field images using Faster R-CNN with ZFNet. Elsevier B.V., pp 45–50
133. Kaddoun SS, Aberni Y, Boubchir L et al (2021) Convolutional Neural Algorithm for Palm Vein Recognition using ZFNet Architecture. 2021 4th Int Conf Bio-Eng Smart Technol BioSMART Bio-Eng Smart Technol BioSMART 2021 4th Int Conf On 1–4. https://doi.org/10.1109/BioSMART54244.2021.9677799
134. Gao F, Fu L, Zhang X et al (2020) Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. Comput Electron Agric 176. https://doi.org/10.1016/j.compag.2020.105634
135. Akpinar KN, Genc S, Karagol S (2020) Chest X-Ray Abnormality Detection Based on SqueezeNet. 2020 Int Conf Electr Commun Comput Eng ICECCE Electr Commun Comput Eng ICECCE 2020 Int Conf On 1–5. https://doi.org/10.1109/ICECCE49384.2020.9179404
136. Bernardo LS, Damaševičius R, De Albuquerque VHC, Maskeliūnas R (2021) A hybrid two-stage SqueezeNet and support vector machine system for Parkinson’s disease detection based on handwritten spiral patterns. Int J Appl Math Comput Sci 31:549–561. https://doi.org/10.34768/amcs-2021-0037

Figures

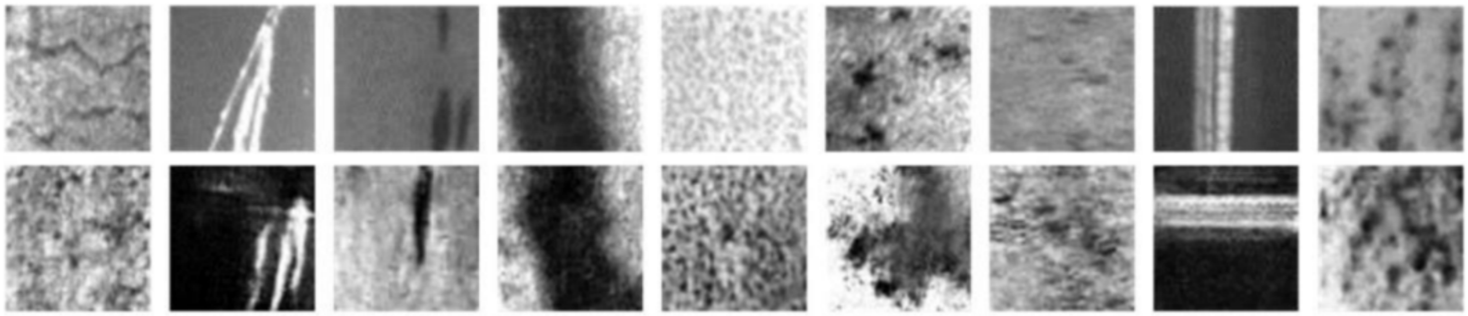


Figure 1

Examples of steel surface defects, left to right, Cr, GG, In, Pa, PS, RD, RS, Sc, and Sp

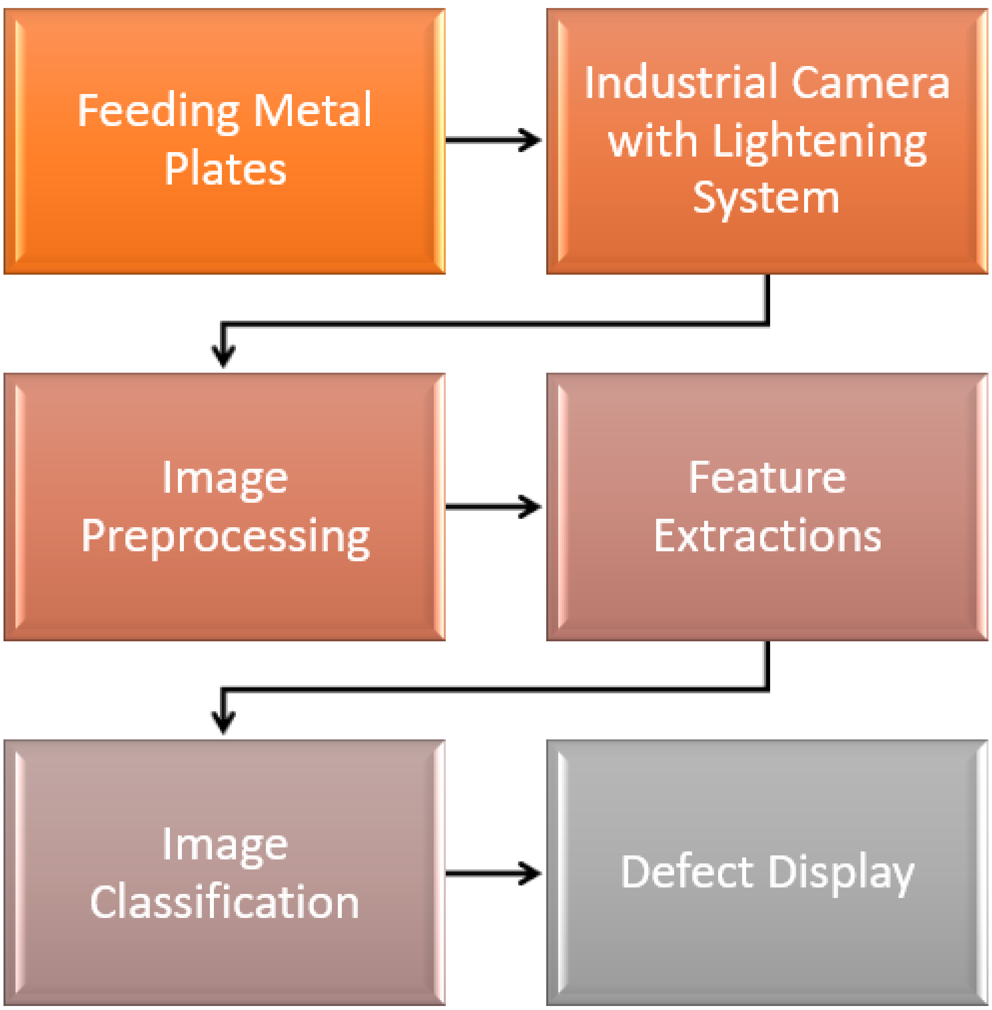


Figure 2

Metal surface computer based-vision inspection framework.

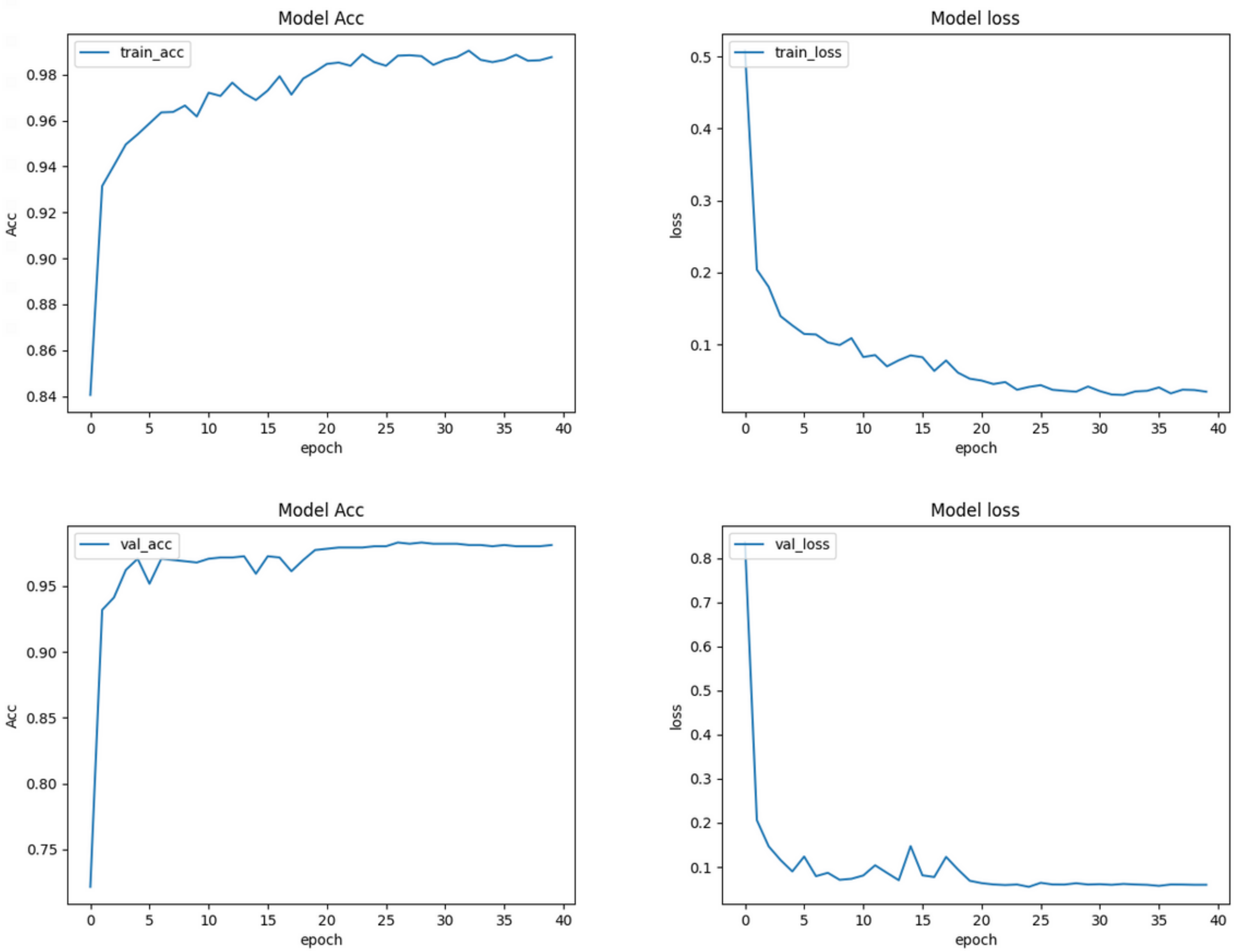


Figure 3

The accuracy and loss vs. epochs for both training and validation for MobileNet.

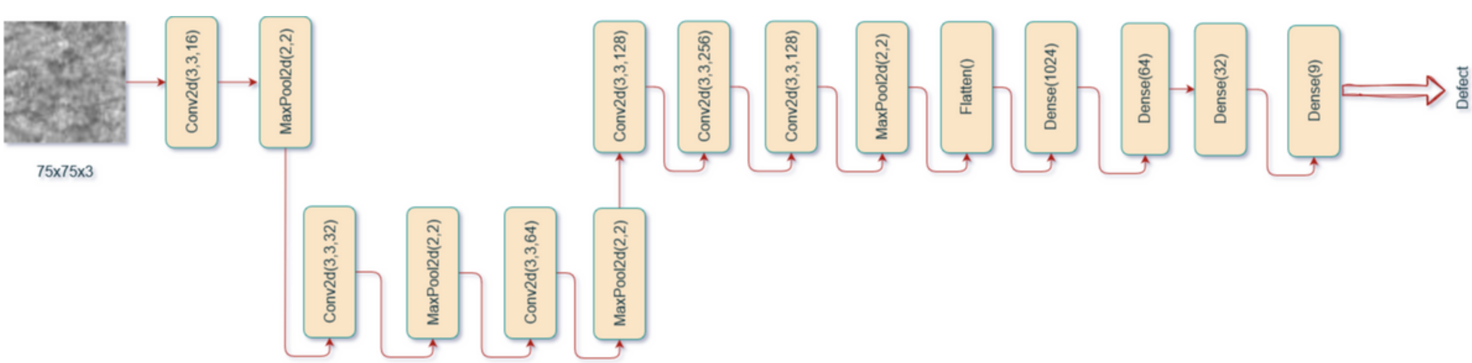


Figure 4

CNN core model.

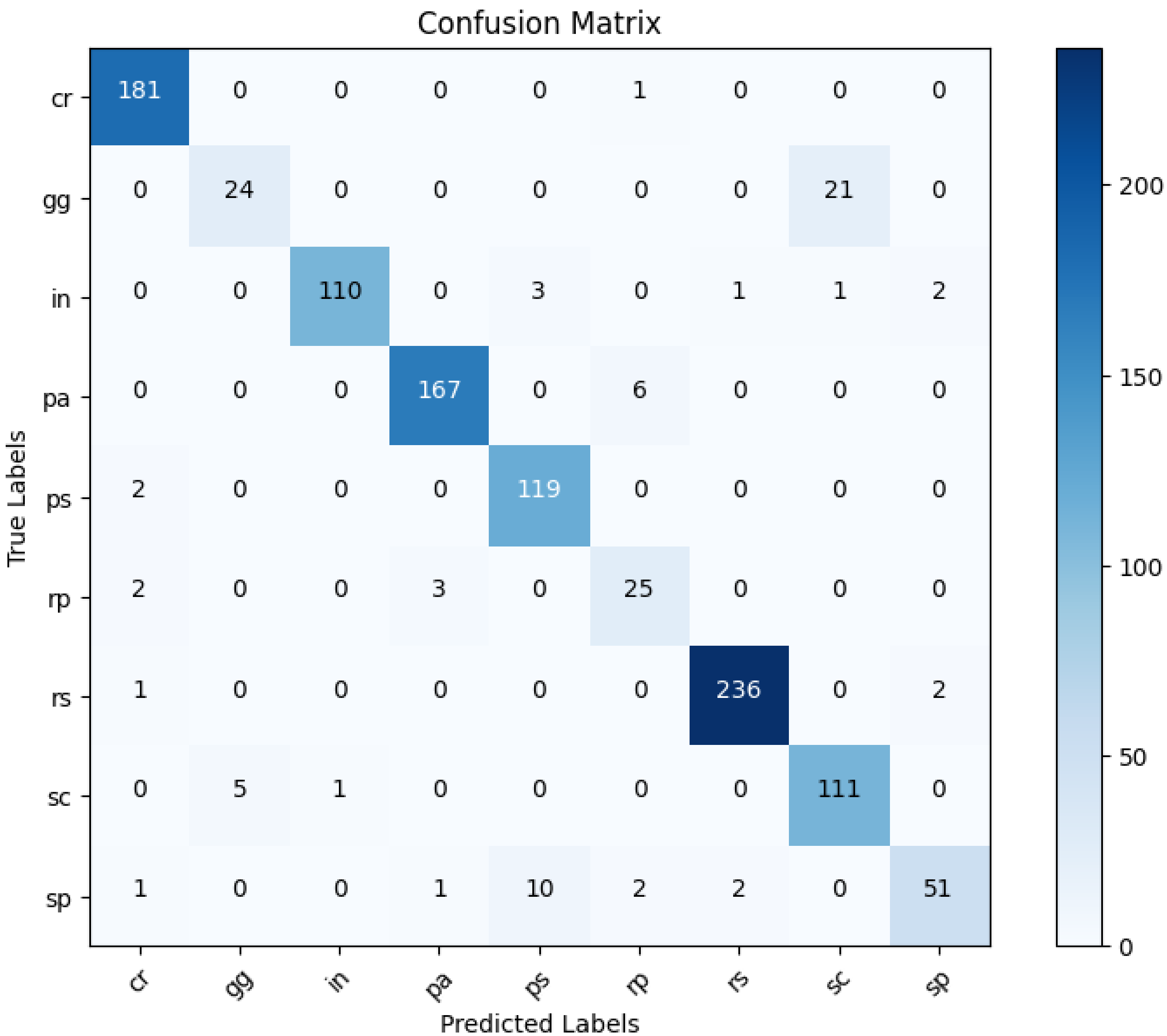


Figure 5

Confusion matrix generated for MobileNet.

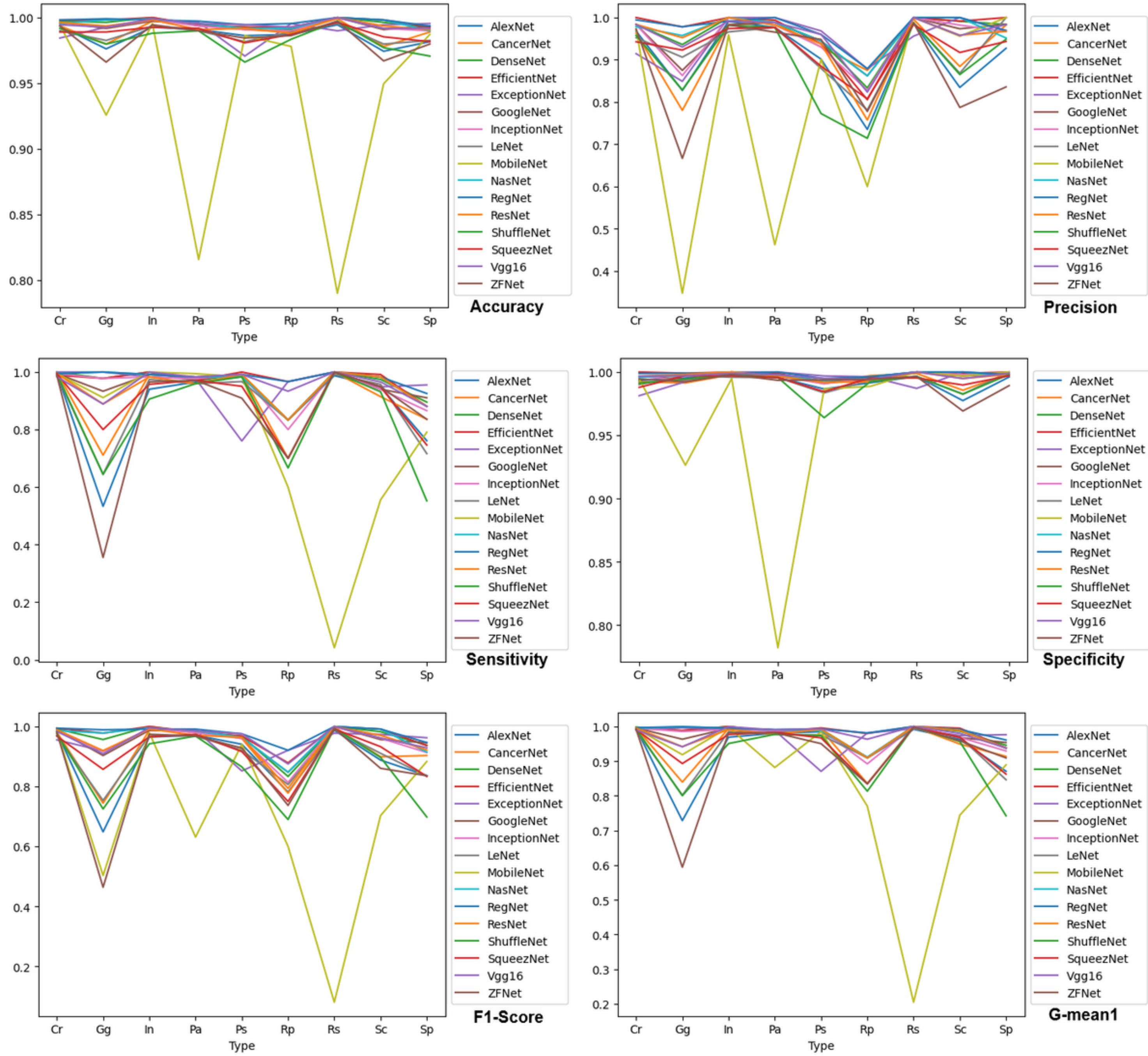


Figure 6

Values of performance measurements for all models per all classes.