Neural Computing and Applications
https://doi.org/10.1007/s00521-024-09664-4

ORIGINAL ARTICLE

End-to-end tire defect detection model based on transfer learning techniques

Radhwan A. A. Saleh[1] · Mehmet Zeki Konyar[2] · Kaplan Kaplan[2] · H. Metin Ertunç[1]

Received: 26 November 2022 / Accepted: 25 March 2024
© The Author(s) 2024

Abstract
Visual inspection of defective tires post-production is vital for human safety, as faulty tires can lead to explosions, accidents, and loss of life. With the advancement of technology, transfer learning (TL) plays an influential role in many computer vision applications, including the tire defect detection problem. However, automatic tire defect detection is difficult for two reasons. The first is the presence of complex anisotropic multi-textured rubber layers. Second, there is no standard tire X-ray image dataset to use for defect detection. In this study, a TL-based tire defect detection model is proposed using a new dataset from a global tire company. First, we collected and labeled the dataset consisting of 3366 X-ray images of faulty tires and 20,000 images of qualified tires. Although the dataset covers 15 types of defects arising from different design patterns, our primary focus is on binary classification to detect the presence or absence of defects. This challenging dataset was split into 70, 15, and 15% for training, validation, and testing, respectively. Then, nine common pre-trained models were fine-tuned, trained, and tested on the proposed dataset. These models are Xception, InceptionV3, VGG16, VGG19, ResNet50, ResNet152V2, DenseNet121, InceptionResNetV2, and MobileNetV2. The results show that the fine-tuned VGG19, DenseNet21 and InceptionNet models achieve compatible results with the literature. Moreover, the Xception model outperformed the compared TL models and literature methods in terms of recall, precision, accuracy, and F1 score. Moreover, it achieved on the testing dataset 73.7, 88, 80.2, and 94.75% of recall, precision, F1 score, and accuracy, respectively, and on the validation dataset 73.3, 90.24, 80.9, and 95% of recall, precision, F1 score, and accuracy, respectively.

Keywords Tire defect detection · Deep learning · Transfer learning · X-ray images · Tire faults classification

1 Introduction

With the increase in human population over the world in recent years, the need for transportation vehicles such as cars, buses, and trucks has increased considerably. This fact drives many factories to increase the production of tires, which are one of the most important pieces of equipment for transportation. Nevertheless, the return rate of defective tires is 7 percent of all tires annually, resulting in annual restitution of $100 million [1]. Quality inspection, which includes fault detection of tires utilizing X-ray images, is required in order to reduce the quantity of returned tires. In tire manufacturing businesses, the task of defect detection utilizing X-ray images is handled manually, which causes time delays and more cost. Moreover, it is a subjective, inefficient, time-consuming, and even biased process that requires a high level of working focus [2].

Mehmet Zeki Konyar, Kaplan Kaplan and H. Metin Ertunç contributed equally to this work.

✉ Radhwan A. A. Saleh
radhwan.saleh@kocaeli.edu.tr

Mehmet Zeki Konyar
mzeki.konyar@kocaeli.edu.tr

Kaplan Kaplan
kaplan.kaplan@kocaeli.edu.tr

H. Metin Ertunç
hmertunc@kocaeli.edu.tr

[1] Mechatronics Engineering Department, Kocaeli University, Umuttepe, 41001 Izmit, Kocaeli, Turkey

[2] Software Engineering Department, Kocaeli University, Umuttepe, 41001 Izmit, Kocaeli, Turkey

Published online: 22 April 2024

In order to get a better view of the difficulties of tires' fault detection problem, an overview of tires' X-ray images is illustrated. Figure 1 shows an X-ray image of a tire which demonstrates the tire's internal structure in detail. A slashed and flattened tire is a good metaphor for a 'long' X-ray image. Stripes are a visual representation of the arrangement of steel wires and rubber. Stripes' unique shape can be considered a type of defect. The steel wire is distributed evenly throughout the tire's right, middle, and left sides. Analysis of these images reveals errors, particularly those created by steel wires. Quality inspectors evaluate a 'long' X-ray image to detect and find image faults. This role is critical for quality inspection and reducing the quantity of returned tires.

Most tire defect detection methods presented in the literature face two types of difficulties. The first one is that the tire images vary due to the fact that there are over 200 different specifications and designs [1]. The other difficulty is that different defects exhibit completely distinct characteristics, and there are over 20 different sorts of defects in tire manufacturing [1].

In tire production, natural rubber, synthetic rubber mixture, various chemicals added to the synthetic mixture for different parts of the tire, and carbon black, which gives the tire its black color, are used as the main material. Afterward, these doughs with different properties were obtained, textile belt coating to be used in tire production, metallic belt coating, patterned and unpatterned tread structure on the outermost part of the tire, the sidewall structure used on the side parts of the tire, the coating of the tire circles on which the tire is placed on the rim and especially It is used as a filling material in the areas where the tire is desired to be strengthened. Various faults may occur in the process, from the preparation of the tire dough to the cooking process of the tire carcasses prepared in various ways.

In order to detect these faults, a 360-degree radiographic image of the tire is taken by using an X-ray camera during the quality control phase of the tire. In order to detect all these faults, the obtained image is examined. After the image is obtained, the fault control is done by an expert operator with eye control. Some types of faults are unacceptable failures, and the relevant tire is scrapped directly, while some types are evaluated according to various thresholds, and tires are classified according to the decision made by the operator.

Several methods have been proposed in the literature to address these challenges. It is possible to classify the proposed solutions in the literature into two types as follows. Some researchers suggested that tire defect detection is a segmentation problem where the defected regions can be extracted automatically using specific approaches. Image segmentation is a hot field of study which is divided into five main categories: thresholding-based, edge-based, region-based, matching-based, and clustering-based segmentation [3]. However, researchers focus on region-based and clustering-based segmentation to solve the tire defects problem. For example, the authors in [4] came up with a way to find tire defects by comparing the distribution of representation coefficients of tires' images. Particularly, the representation coefficients of defect-free images follow a Gaussian distribution, whereas the defect images follow a non-Gaussian distribution.

Later on, the authors in [2] used wavelet multi-scale edge detection to analyze faulty edges in high-frequency multi-textural backgrounds. They suggested a special framework to distinguish the region of the defect from the background. This framework uses edge detection models after optimizing their parameters, such as the threshold value.

Moreover, in [5], a fast detection approach for automatic quality control is presented. This approach uses the feature similarity of tire pictures to capture texture distortion in each pixel by weighting the dissimilarity between pixels. The proposed method was tested on both sidewall and tread images and delivered competitive results. Latterly, the authors in [6] presented a high-precision approach for pixel-level defect identification. They used local inverse difference moment (LIDM) features to create a feature distribution. The defect feature map (DFM) is formed to make it easier to find the defect by comparing the LIDM feature distributions of the original tire image and each

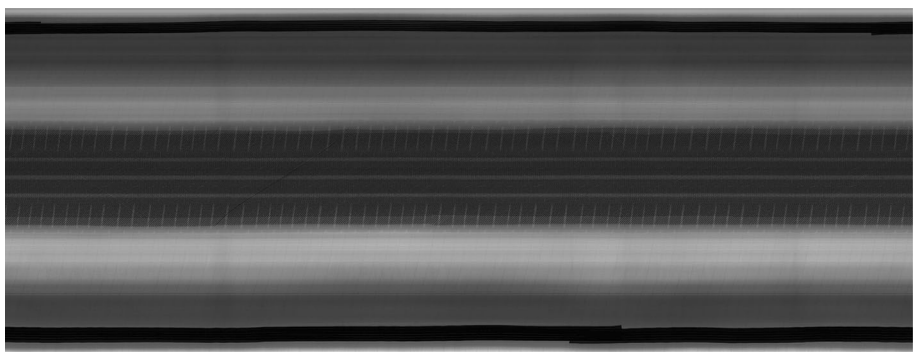


Fig. 1 Example of free-defects long X-ray tire image

sliding image patch. This is done by computing the Hausdorff distance between these two distributions.

Furthermore, in [7], an auto-detection of tire defects in an X-ray image is presented using the inverse transformation of the principal component residual. In this model, three defect types were detected as follows: cords break, bubbles, and foreign matter. In [8], the fine-tuning of the STDC-Net encoder is used to extract the texture feature of the tire's different regions. After that, they proposed a special decoder that compresses the encoder features output and searches the boundary of the bead toe for defect detection purposes. This segmentation approach has achieved 92.4% L-mIoU and 97.1% mIoU for 512×512 input image.

Although the segmentation-based approaches can succeed in solving the first challenge, the variation of tire specifications and designs, they suffer in solving the second challenge, the distinct characteristics of defect types. Thus, they are rarely used in industrial applications [1]. Table 1 summarizes the segmentation-based approaches in recent studies.

The second group of researchers dealt with tire defect detection as a classification problem. Consequently, they have suggested different classifiers using both machine learning and deep learning techniques. For example, the author of the [9] detects tire bubbles with texture-based features and machine learning methods. The convolutional neural networks (CNN) and the faster regions CNN are employed in [10] for the detection of bubble errors. In [11], an improved extremum filter and an enhanced locally adaptive-threshold binarization are used to detect impurities and bubble errors in tire X-ray images. The authors tested their approach by utilizing 280 tires from a tire factory with impurities and bubbles faults. Nonetheless, the models presented in these studies were used to detect only impurities and bubbles defects, whereas, in reality, there are around 20 types of defects [1]. This fact highlights the fact that they could not be embedded in real applications.

In [12], tire faults are classified using a multi-column CNN (MC-CNN) that combines five individual CNNs wherein the predictions of the five CNN are averaged to provide the final output, while in [13], AlexNet-based tire defect classifier is developed. The X-ray images used in these two pieces of research are derived from six types of defects called: Normal-Cords (NC), Bulk-Sidewall (BS), Cords-Distance (CD), Belt-Joint-Open (BJO), Sidewall-Foreign-Matter (SFM), and Belt-Foreign-Matter (BFM).

Due to the rapid success of transfer learning models in many applications, the authors in [14] suggested the use of VGG16 pre-trained model with a fully convolutional network (FCN) to detect tire defects. They fine-tuned the parameters and structure of FCN in order to acquire coarse detection results, which they then enhanced using a fusion technique to obtain fine detection results. However, only four types of defects for both the tread and sidewall tire images have been detected in this study, making it impractical in factory applications. Recently, TireNet has been offered as an end-to-end technique for the practical use of X-ray image-based tire defect identification [1]. This model used the Siamese network as part of a downstream classifier to collect faulty features inspired by periodic

Table 1 Segmentation-based approaches in previous studies

References	Approach	Dataset	# Defects types	Measurement metrics
[4]	Comparing the distribution of features	60 defect images 120 defect-free images	Not known	Error ratio (1.9%) and Detection Ratio (93.4%)
[2]	Wavelet multi-scale edge detection	200 defect images 200 defect-free images	Not known	EM (0.915), Success rate (96.9%), False alarm rate (1.8%) Non-detected rate (1.3%)
[5]	Weighting the dissimilarity between pixels	40 defect sidewall images 20 defect-free sidewall images 40 defect tread images 20 defect-free tread images	Two	Sidewall detected rate (93.3%) Tread detected rate (80%)
[6]	Local inverse difference moment	6 defect images	Not known	F1-measure (90.95%)
[7]	Inverse transformation of PCA	8 defect images 7 defect-free images	Not known	Not known
[8]	STDC-Net Encoder (Unsupervised)	1218 training defect images 854 testing defect images	Not known	EM (0.915), L-mIoU (92.4%), mIoU (97.1%)

features of tire X-ray pictures. The labeled dataset used in this research was 120,000 tire images (100,000 qualified tires and 20,000 defective tires). Moreover, the proposed model was compared to YOLO, SSD, and Faster R-CNN and achieved better results in terms of the recall metric.

In [15], a novel two-stage CNN model is developed for tire defect detection by merging an improved pyramid scene parsing network with an optimized YOLOv3. The authors utilized the K-means algorithm to find the best anchor box and optimize the YOLOv3 network. This model was tested on six types of defects using the CIoU loss function and achieved an average precision of 91.39%. Later on, they proposed another novel model based on a deep convolutional sparse-coding network (DCScNet) [16]. Since sparse coding is used to extract the tire features, it is classified as unsupervised learning. Finally, DCScNet was tested on the same dataset and achieved an accuracy of 96.8%. Recently, gray-level co-occurrence matrix (GLCM) has been used with 22 texture features for the feature extraction stage. After that, different classifiers were evaluated on a dataset consisting of two types of faults (higher deviance and impurities) and defected free images of size 235 and 1276 images, respectively. However, artificial neural network (ANN) classifier achieved the best performance [17].

Indeed, this article is an extension to our previous work wherein the dataset gathered from 50 different design patterns with 15 types of defects for 3366 images of faulty tires and 20,000 images of qualified tires, unlike the previous dataset [17], which was gathered from some design patterns with only two types of faults. Table 2 summarizes the classification-based approaches in recent studies.

It is obvious from Tables 1 and 2 that although the considerable success achieved by these models, they have not yet met the standards for application-level implementation. This is because there are around 20 types of defects in real applications. Moreover, there are around 50 different patterns in tire design. These two challenges were not considered deeply in the previous works, which motivated us to do this research.

Furthermore, we can conclude that the segmentation-based models cannot recognize the complex patterns of different defects better than the classification-based models. Therefore, this paper proposes and compares classification-based models using fine-tuned transfer learning models. The contributions of this paper are listed as follows.

- The proposed methods in this study can detect defective tires in spite of the variety of specifications, designs, and defect types, as our dataset consists of 15 types of defects and 50 different design patterns. In fact, this is one of the most difficult challenges facing previous studies.
- New dataset of tires' X-ray images is collected and labeled. The new dataset consists of images with 15 types of defects that come from around 50 different design patterns. The collection of this dataset continued for more than nine months since defective tires rarely occurred.
- To the best of our knowledge, the literature is limited due to the lack of publicly available tire X-ray datasets. The proposed method contributed to the literature with competitive results. The fact that the enhanced TL models will be adapted to an existing tire X-ray device also makes the proposed study valuable.
- Fine-tune the hyper-parameters of the nine state-of-the-art pre-trained models for tire defects detection application.
- The nine state-of-the-art pre-trained models are compared in terms of accuracy, recall, precision, and F1 score. Based on the results, the best model for tire defect detection is determined.

The following is the structure of this paper. Section 2 presents a brief background about transfer learning models that were employed in this study. The proposed methods and datasets are presented in Sect. 3. The experiment setup, the descriptive and analytical results, and their discussions are covered in Sect. 4. Finally, Sect. 5 presents the study's conclusions and future works.

2 Background

Deep learning (DL) techniques automatically learn from data, discover patterns, and make accurate decisions. As a result, recently, they have played a significant role in industrial applications. Wherein the use of DL can transform industrial operations into highly efficient smart facilities [18], in manufacturing, for example, DL models can extract insight from ambiguous sensory input, resulting in intelligent manufacturing [19]. A major advantage of DL over standard machine learning is that feature learning is accomplished automatically without the need for any outside interaction [20, 21].

In the context of AI, end-to-end learning is a technique where the model learns every step from the initial input stage to the final output result simultaneously [22]. End-to-end learning is also introduced as the process of using gradient-based learning to train an overall, potentially complicated learning system [23]. All the proposed models in this paper are trained using gradient-based learning. Moreover, there are not any external middle stages within the models, like preprocessing or feature extraction. Since

Table 2 Classification-based approaches in previous studies

References	Approach	Dataset	# Defects types	Measurement metrics
[9]	Texture based features	207 defect images 793 defect-free images	Only bubbles	Accuracy (85.9%) Recall (92.62%) Precision (89.65%)
[10]	Combining CNN and Faster R-CNN	136 defect images 262 defect-free images	Only bubbles	Accuracy (89.16%) Sensitivity (100%) Specificity (83.09%)
[11]	Extremum filter and Self-adapt local binarization	280 defect images	Bubbles and Impurities	Impurity Precision (97%), Impurity Recall (95.7%) Clear Bubble Precision (95.8%), Clear Bubble Recall (92%)
[17]	GLCM with Texture Features and ANN	235 defect images 1276 defect-free	Higher deviance and Impurities	Precision (100%), Recall (96%) F1 score (97.9%) Accuracy (99.34%)
[12]	MC-CNN	1266 defect-free images 316 defect images	Six	Accuracy (98.47%)
[13]	AlexNet	1266 defect-free images 316 defect images	Six	Accuracy (96.51%)
[14]	VGG16 with FCN	700 training images 214 testing images	Four	Accuracy (78.91%)
[15]	Merging an improved pyramid scene parsing network with an optimized YOLOv3	1450 defect images	Six	Precision (91.39%) mIoU (87.86%)
[1]	TireNet Based on Siamese Network	20,000 defect images 100,000 defect-free images	Eleven	Recall (83.31%) Precision (93.55%)
[16]	Unsupervised DCScNet	1450 defect images	Six	Accuracy (96.8%)
[8]	STDC-Net Encoder (Unsupervised)	1218 training defect images 854 testing defect images	Not known	EM (0.915), L-mIoU (92.4%), mIoU (97.1%)

the model receives the image as an input and predicts whether there is a fault or not as an output. Thus, the term end-to-end model is used to define all models in this paper.

Transfer learning (TL) is a common research subject in classification problems wherein a pre-trained model is applied to learn a new related task. TL begins with training the model using an enormous dataset and challenging task, followed by transferring the learned features to the second model for training on the target dataset and task [24]. The advantage of TL is manifested when the amount of the available dataset is not huge enough for training.

ImageNet is an open-source picture collection with over 1.2 million images and over 1000 classes. Most of the common TL models, such as VGG16, ResNet50, DenseNet121, ResNet152V2, Xception, InceptionResNetV2, EfficientNetB0, and MobileNetV2 have been trained on the ImageNet dataset. All ImageNet-based models accept either $224 \times 224 \times 3$ image size or $299 \times 299 \times 3$ as input and produce a vector of size 1000 that represents the probability of belonging the image to each class. In this research, we used these TL models as feature extractors and fine-tuned their hyper-parameters for the classification tasks. Consequently, the following subsections illustrate the structure of the general CNN framework in addition to an overview of these TL models.

2.1 Convolutional neural network (CNN)

Convolutional neural networks are one of the most widely used deep learning methods, with applications ranging from object detection to image classification and recognition. CNN can be broadly classified into seven different classes based on various improvements (such as structural reformulations, regularization techniques, and parameter optimizations), namely feature map exploitation, channel boosting, width, multi-path, depth, spatial exploitation, and attention-based CNN [25].

Using CNN, it is possible to automatically recognize hidden characteristics within the pixels and convert them into a map of numbers. These numerical maps are then processed and fed into a deep neural network capable of learning and making predictions. Thus, CNN does not require a separate manual feature extraction stage like other machine learning algorithms. Figure 2 shows the typical

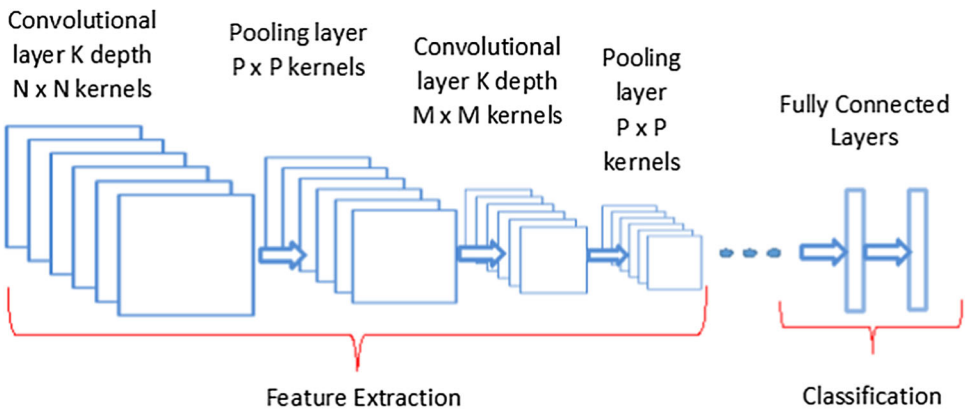


Fig. 2 CNN architecture

basis of CNN architecture. As a starting point, the network is pushed with an image, or 'input image.' The convolutional section of the network is where the input image slides through an infinite number of steps. Once this is done, the fully connected layers will take the final decision.

- Convolutional layer It consists of a collection of convolutional kernels, each associated with a small image region referred to as a receptive field. It is used to extract valuable features from an image. The convolution process results in the multiplication of the weights and the sliding window associated with the input image. Due to the convolutional operation's weight-sharing capability, various sets of features inside an image can be retrieved by sliding kernels with the same set of weights on the picture, which makes CNN parameters more efficient than fully connected networks [25]. Figure 3 illustrates how this layer works.
- Pooling layer The pooling layer is used to minimize the dimension of the image dataset's representations created by the convolution layer, resulting in a reduced sample size for faster calculations. There are several types of pooling layers, including max-pooling, which retains the maximum values from the filter's shape, average pooling, which retains an average value, and min pooling, which retains the filter's minimum value [26]. Figure 4 illustrates the max-pooling layer process with an example.
- Flattening layer Flattening is the process of transforming a 2-dimensional array of pooled feature map results into a single long continuous linear vector.
- Fully connected layer The vector of features obtained from previous layers will be used as an input for this layer which uses nonlinear activation functions to classify the input image by creating a nonlinear combination of selected features [27]. Every neuron in the previous layer is linked to every neuron in the following layer in the fully connected layers, as shown in Fig. 5 [28].
- Activation layers The activation function acts as a decision-making mechanism and aids in the recognition of complicated patterns. By selecting a

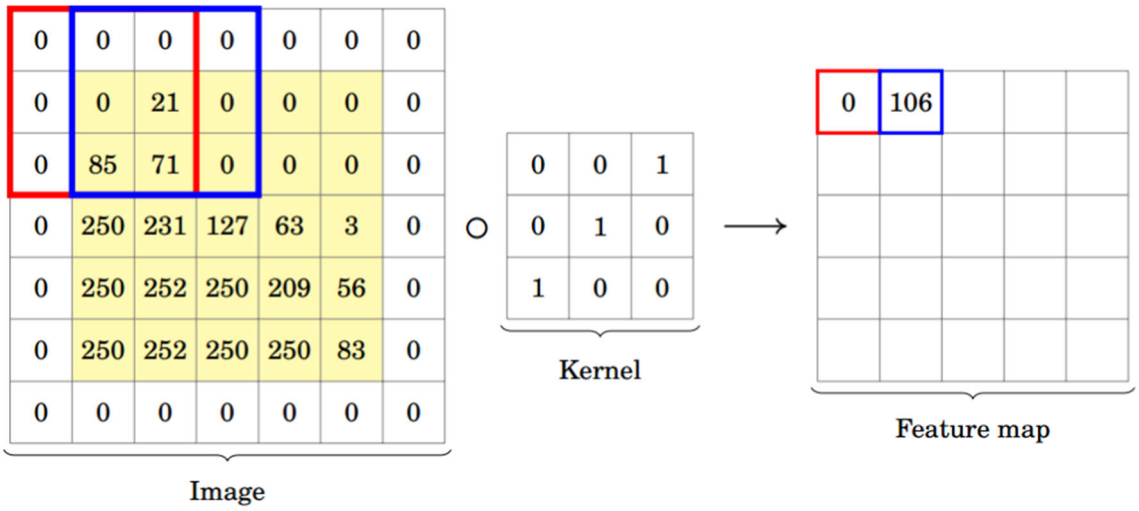


Fig. 3 Convolutional process

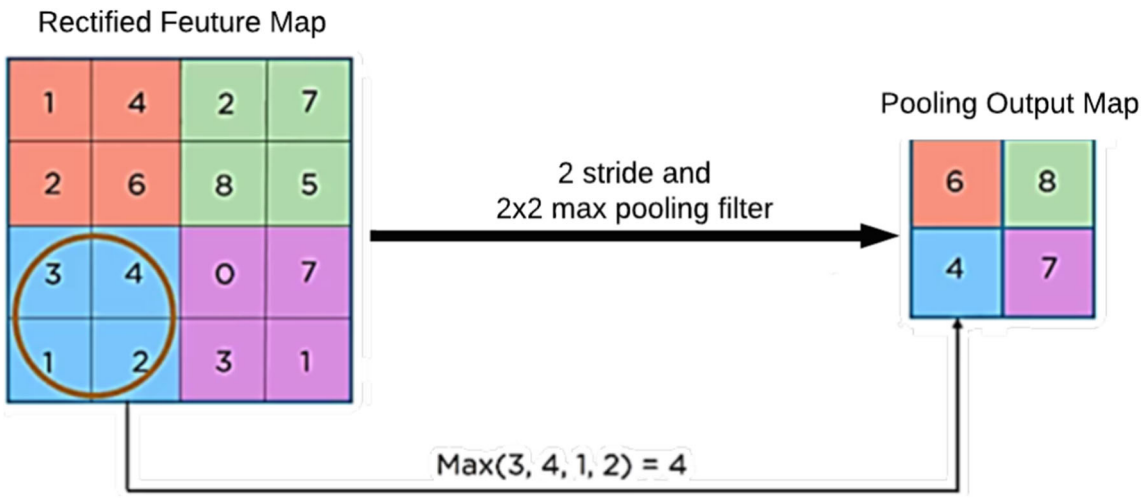


Fig. 4 Max pooling example

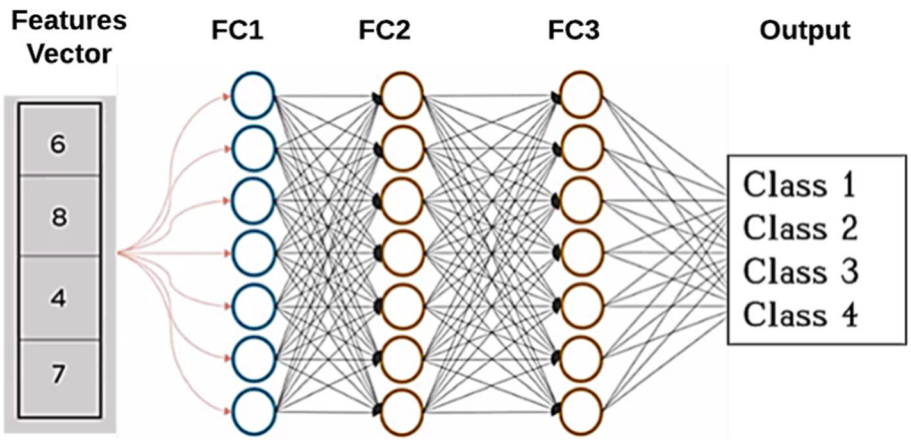


Fig. 5 Connection between fully connected neurons

suitable activation function, the learning process can be accelerated [29]. Different activation functions have been suggested and utilized in the literature. However, sigmoid, tanh, SoftMax, and ReLU are the most common of them [30].

- Dropout layer Overfitting occurs in NNs when many connections are co-adapted when they learn a nonlinear relationship. However, dropout is used to increase generalization by randomly skipping particular units or connections with a specific probability, resulting in better generalization. This random dropping of connections or units results in multiple thinning network topologies, from which a representative network with low weights is picked. It is therefore assumed that the chosen network architecture is an approximate representation of the suggested networks [26].

2.2 VGG16

The VGG16 architecture is the most widely used for ImageNet in the literature [31]. In VGG16, there are 13 convolution layers (each has multiple filters of 3×3, with a stride of 1px and ReLU as an activation function), three pooling layers, and five fully connected layers in total [32]. Although it consists of 16 layers in total, it has 15.3 billion FLOPs.

2.3 VGG19

The VGG19 architecture is an extension structure of VGG16. The difference is that in VGG19, there are 16 convolution layers (each has multiple filters of 3×3, with a stride of 1 px and ReLU as an activation function), five pooling layers, and three fully connected layers in total. The feature extraction layers are divided into five groups, each of which is followed by a max-pooling layer [33].

2.4 ResNet50

ResNet50, which stands for residual learning framework, in which layers inside a network are reformed to learn a residual mapping between inputs and outputs rather than the desired unknown mapping. It consists of 50 layers and has 3.8 billion FLOPs which is lower compared to the FLOPs of the VGG16 model [34]. ResNet50 is comprised of five stages of convolution. Conv1 contains a single convolution block comprised of a single convolution layer (1×1). There are three, four, six, and three convolution blocks in each of the remaining convolution layers (Conv2, Conv3, Conv4, and Conv5), respectively. There are three convolution layers in each convolution block, which are denoted by (1×1), (3×3), and (1×1). The average pooling layer is used to downsample the feature map. In addition, a fully linked convolution layer is used for classification at the end of the network [35].

2.5 ResNet152V2

ResNet152V2 is another version of residual learning framework-based models whose accuracy based on the ImageNet dataset exceeded 94.2% [36]. ResNet152V2 is built similarly to ResNet50 by adding more 3-layer blocks in sequence. Therefore, ResNet152V2 consists of 152 layers and has 11.3 billion FLOPs which is still lower compared to the FLOPs of the VGG16 model [34].

2.6 DenseNet121

DenseNet121 is a dense convolutional network-based pre-trained model that eliminates the layer-to-layer connectivity pattern found in other architectures, ensuring maximal information (and gradient) flow [35]. It is composed of four dense blocks, each of which has six, twelve, twenty-four, and sixteen convolution blocks. Each convolution block contains (1×1) and (3×3) convolution layers. Aside from that, there are (1×1) convolution as well as (2×2) average pooling layers between the dense blocks. The network also includes a convolution layer (7×7) at the input and a fully connected convolution layer at the output; thus, it is composed of 121 layers [35].

2.7 InceptionV3

The third version of the Inception neural network is the most recent and available version. Its structure is made up of a layered pattern that is repeated along the net. There are modules that extract distinct picture features with different filter sizes in parallel using multiple convolutional layers, which are concatenated at the end of the module. The idea here is to try different sizes of filters to increase the depth of the feature search [25].

2.8 InceptionResNetV2

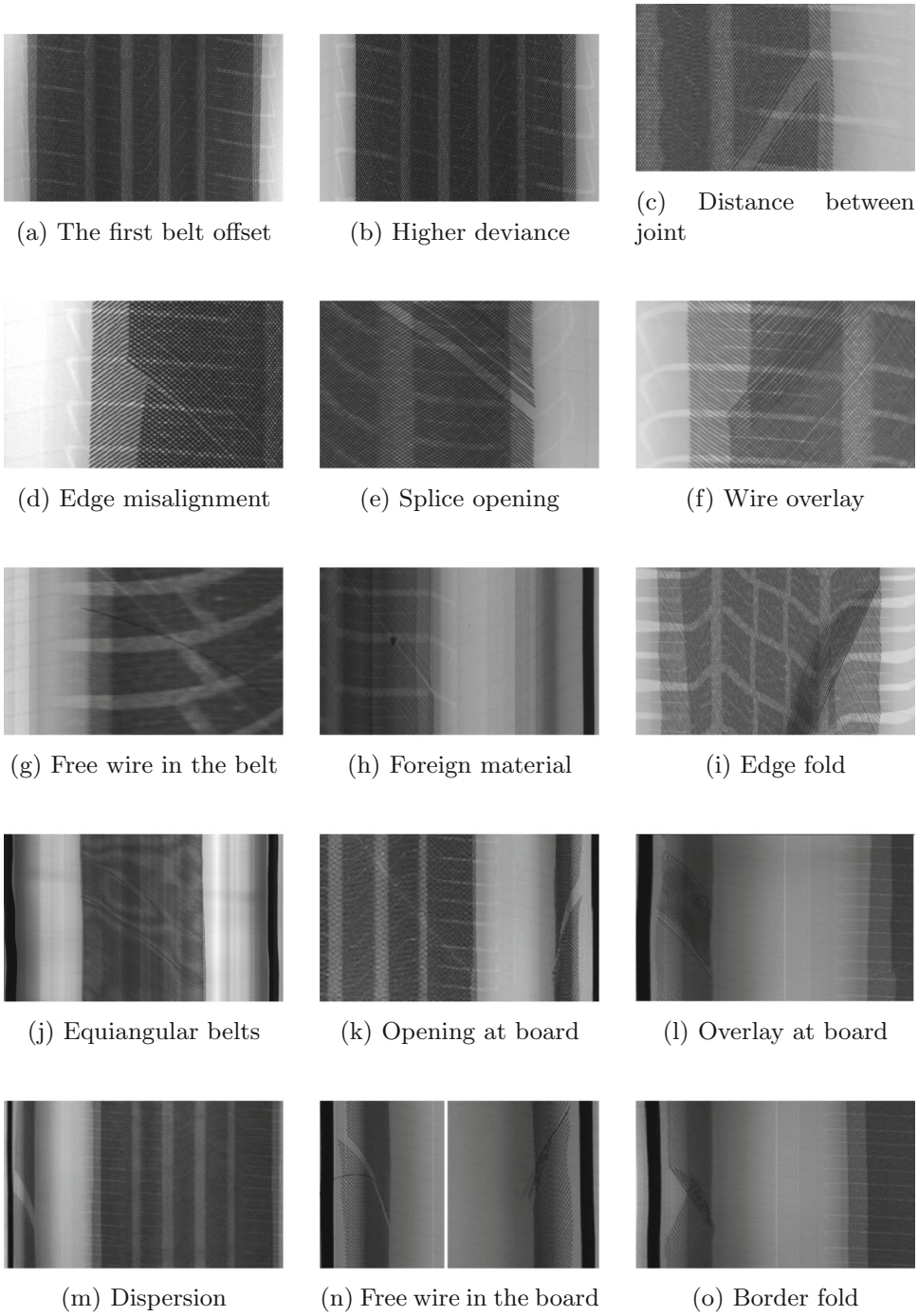
The InceptionResNetV2 is a quite deep pre-trained model that obtained 95.3% of accuracy on the ImageNet dataset. It extends the concept of CNN construction by utilizing blocks rather than merely convolutional layers. Additionally, they divided convolutional operations into geographically distinguishable ones to better utilize computational resources, enhancing model depth and width while holding computational costs unchanged [37]. It contains many residual blocks, such as Repeat, Repeat1, and Repeat2, which are all connected by other residual blocks. It comprises the main stem, followed by the inception and reduction resents A, B, and C, respectively. Using the stem module, the inception ResNet-A block receives input from a succession of 3×3 convolutions, 3×3 maxpools, and filter concats. All of the inception blocks are activated by the ReLU activation function, which is applied to each reduction block. However, it did not apply batch-normalization on top of summation layers [38].

2.9 MobileNetV2

MobileNetV2 is a convolutional architecture that minimizes the cost and size of networks. It was designed for usage with constrained devices such as mobile devices. Its architecture begins with a full convolution layer that has 32 filters. Then, there are 19 residual bottleneck layers [39]. The main part of MobileNetV2 is the depth-wise separable convolution which forms point-wise 1×1 convolutions and depth-wise convolutions. The factorized shape of these convolutions minimizes the number of multiplication operations compared to the ordinary convolutions, resulting in a lower computing cost [40].

2.10 Xception

Xception is a depth-wise separable CNN model that beats InceptionV3 in terms of performance [41]. It consists of 36 convolutional layers, which serve as the network's feature extraction foundation. These layers are divided into 14 modules, which all have linear residual connections surrounding them, except the very beginning and last modules. Its structure begins with the entrance flow, divided into four modules, each having two convolution layers. The convolution is accomplished in the first module through the use of 32 and 64 filters with a 3×3 filter size, while the other three modules use 128, 256, and 728 filters with a 3×3 filter size. Thereafter, a total of eight separable convolution processes with 728 filters of 3×3 size are



(a) The first belt offset (b) Higher deviance (c) Distance between joint

(d) Edge misalignment (e) Splice opening (f) Wire overlay

(g) Free wire in the belt (h) Foreign material (i) Edge fold

(j) Equiangular belts (k) Opening at board (l) Overlay at board

(m) Dispersion (n) Free wire in the board (o) Border fold

Fig. 6 Sample X-ray images for tire defects

repeated eight times in the middle flow. There are two modules in the exit flow wherein the first module convolution is performed with 728 and 1024 filters in 3×3 sizes, and the other module convolution is performed with 1536 and 2048 filters. The architecture is finally completed by adding fully connected layers [42].

3 Methodology

This section discusses the methods and datasets used in this work and the suggested method's evaluation and comparison details. TensorFlow and Keras were used to implement the proposed model. Thus, in this section, the used dataset and proposed models are discussed in detail.



Fig. 7 The utilized Tire X-ray imaging system for image acquisition (**a**) outside, (**b**) inside

3.1 Dataset

In tire production, various malfunctions may occur in the process, from the preparation of the tire dough to the cooking process of the tire carcasses prepared in various ways. For example, foreign matter may have been incorporated into the tire structure at any of the tire manufacturing stages. Although metal detectors can detect many metal-containing foreign materials at any production stage, non-metallic and invisible foreign materials can only be detected by radiographic quality control devices. Another example is the faults in the joints of the textile and metallic belts used in the tire's inner structure. These types of faults occur due to the following reasons: overlapping between layers, the overlap of the joint being open, the joint being made at wrong angles or slipped, the horizontally offset joints, and impossible end-to-end joining, etc. Cord yarn and wires in the belts used in the tire are prepared at various angles. Mostly, the winding is done using more than one belt so that the angles are diagonal. Winding ropes or wires in parallel instead of diagonally is another type of error. The tires used in this study contain the following types of faults. Figure 6 shows a sample of X-ray images of them.

- The first-belt offset: the first belt in the tire is circumferentially offset from the side.
- Higher deviance: the first-belt deviance is higher than the second-belt gap.
- Distance between joints: the distance between the two ends of the joint on the belt edge exceeds the threshold distance.
- Edge misalignment: Saw tooth-shaped misalignment at the edge of the belt.
- Splice opening: wire thinning, lack of wire, or splice opening in the tire.
- Wire overlay: overlaying of excess wire in the splice.
- Free wire in the belt: presence of free wire inside or outside the belt package.
- Foreign material: presence of foreign matter inside or outside the belt package.
- Edge fold: having folds at the edges of the belt.
- Equiangular belts: having two equiangular belts instead of belts that should be in opposite directions.
- Opening at the board: opening at the end of the board.
- Overlay at the board: overlaying at border joint.
- Dispersion: dispersion or separation of the joint.
- Free wire in the board: the presence of free wire in the board.
- Border fold: folding on the board.

The dataset for this article's experiments was obtained from the Pirelli Automobile Tyre Factory in Turkiye. To begin, we collected 'long' X-ray images with X-ray machine outputs as the initial dataset, which includes images of faulty and qualified tires. Due to the rarity of faulty tires in actual production, we have been collecting images for more than a year. Finally, we gathered 3366 images of faulty tires and 20,000 images of qualified tires, all of which were identified and labeled by quality inspectors from the Pirelli Factory.

The dataset is generated by the Alfautomazione Tire-X 3000 system, an advanced tire X-ray inspection machine, as illustrated in Fig. 7. This cutting-edge technology consists of two main components: the diode and the receiver. This arrangement offers the highest level of safety during operation by encasing it in a lead-lined chamber to prevent radioactive leakage.

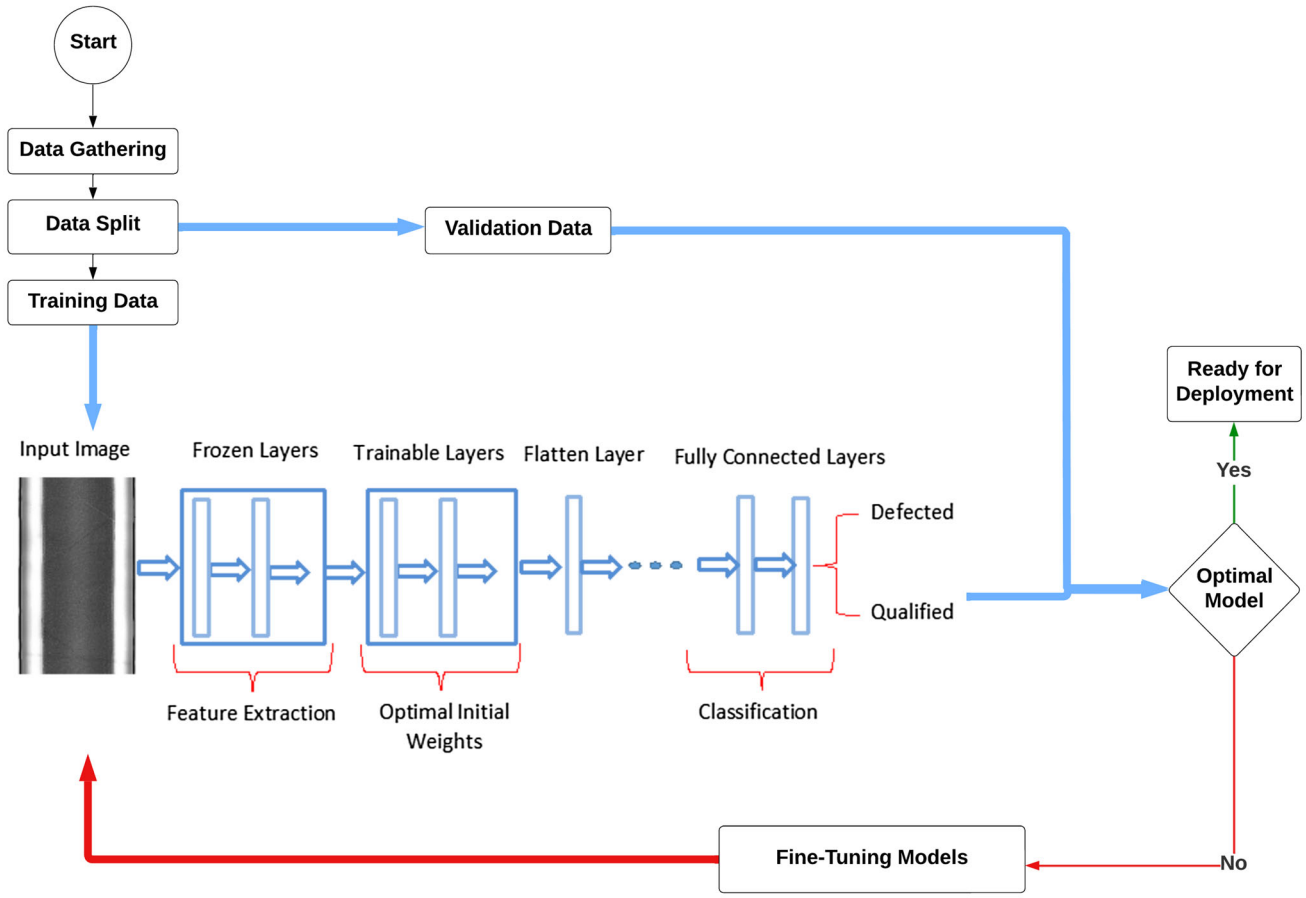


Fig. 8 Flowchart of the proposed method

Table 3 Parameter details setting of the proposed structure

Model	Frozen	Trainable	Total	# HL
VGG16	7,635,264	17,696,769	25,332,033	15
VGG19	10,585,152	20,056,577	30,641,729	18
ResNet50	14,656,384	34,621,953	49,278,337	51
ResNet152_V2	49,402,368	34,619,905	84,022,273	153
DenseNet121	6,549,696	13,333,377	19,883,073	123
MobileNetV2	1,371,904	16,942,913	18,314,817	55
InceptionV3	21,802,784	42,467,841	64,270,625	50
InseptionResNet50_V2	54,336,736	25,166,337	79,503,073	574
Xception	14,073,096	59,217,697	73,290,793	73

Upon closing the cabin, a high voltage is applied to the diode within, causing X-rays to be emitted. As the tire enters the chamber, the cabin seals close, triggering a precise 360-degree rotation process. Meanwhile, outside the tire, a U-shaped receiver records the resulting X-ray image, mirroring the same principles as medical X-ray devices.

The diode uses a water-cooling system to regulate temperature, assuring operational efficiency and safety. The inspection process normally takes one minute and varies with tire diameter. The device interfaces with a computer user interface, allowing quality workers to thoroughly review tire X-ray images in order to maintain the highest levels of integrity and quality.

3.2 Fine tuning of TL models

We can demonstrate a strong capacity to generalize images beyond the ImageNet dataset by fine-tuning the pre-trained model. In general, the TL fine-tuning approaches are divided into four categories as follows.

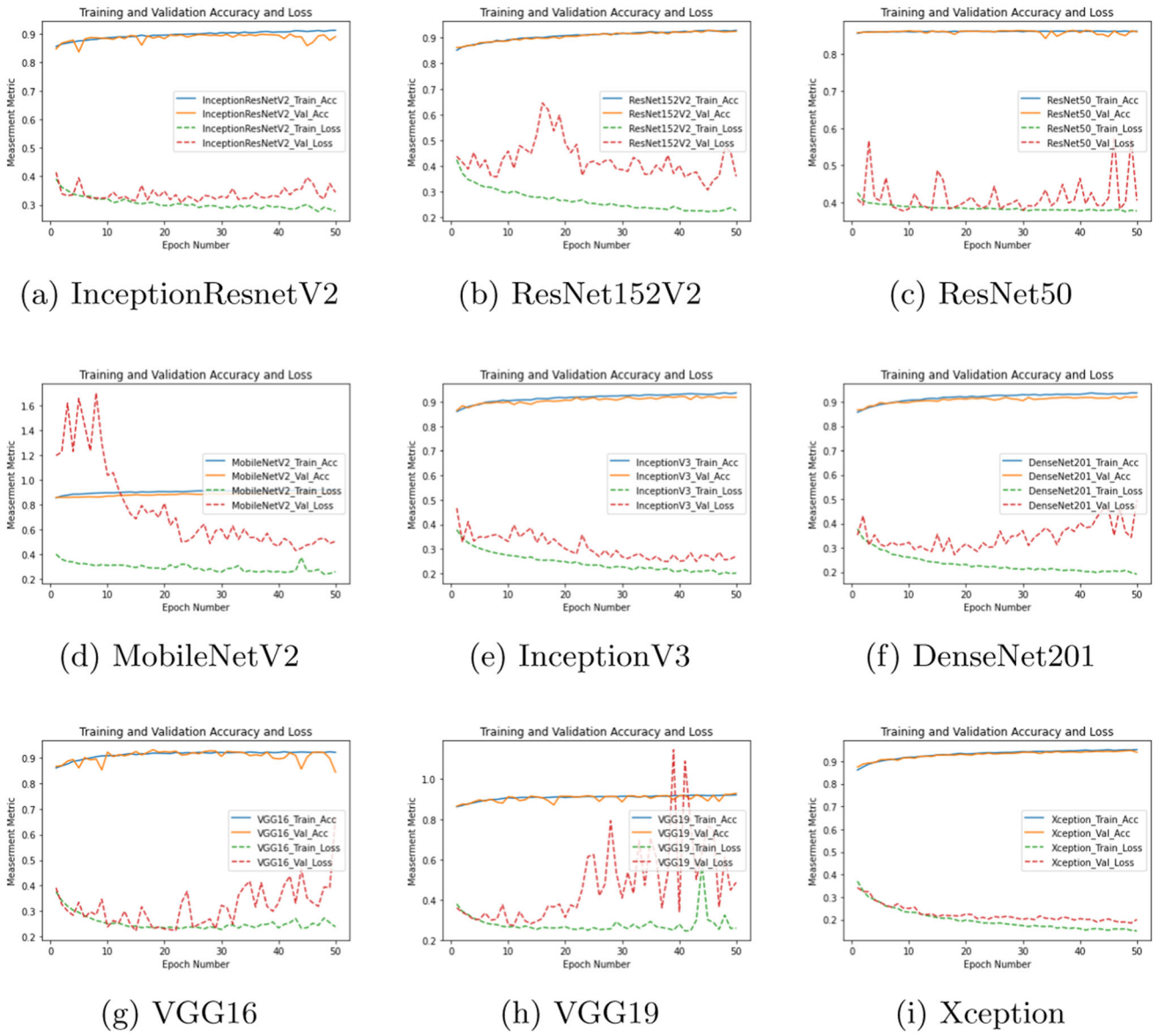


Fig. 9 Convergence rates for all models 'training and validation accuracy and loss'

- *Feature extraction*: by removing the output layer, we may turn a pre-trained model into a feature extraction tool. This approach is helpful when we have a small amount of dataset that is highly similar to the ImageNet dataset.
- *Freeze some layers while training others*: we freeze the weights of the model's early layers and retrain only the upper layers. This approach is preferred when we have a small amount of dataset that has low similarity to the ImageNet dataset.
- *Train the architecture from scratch*: in this approach, we reuse the model's architecture by randomly initializing all the weights and retraining the model on our dataset. This approach is utilized when we have enough amount of dataset that has low similarity to the ImageNet dataset.
- *Initialize the model weights*: in this approach, we use the pre-trained model's weights as initial weights and retrain the model using our dataset. This approach is used in the ideal cases when we have enough amount of dataset that has high similarity to the ImageNet dataset.

Since our dataset has low similarity to the ImageNet dataset and is not large enough compared to the challenges of defect types and design patterns, we follow the second TL fine-tuning approach. Thus, we froze the weights of the model's early layers and retrained only the upper layers.

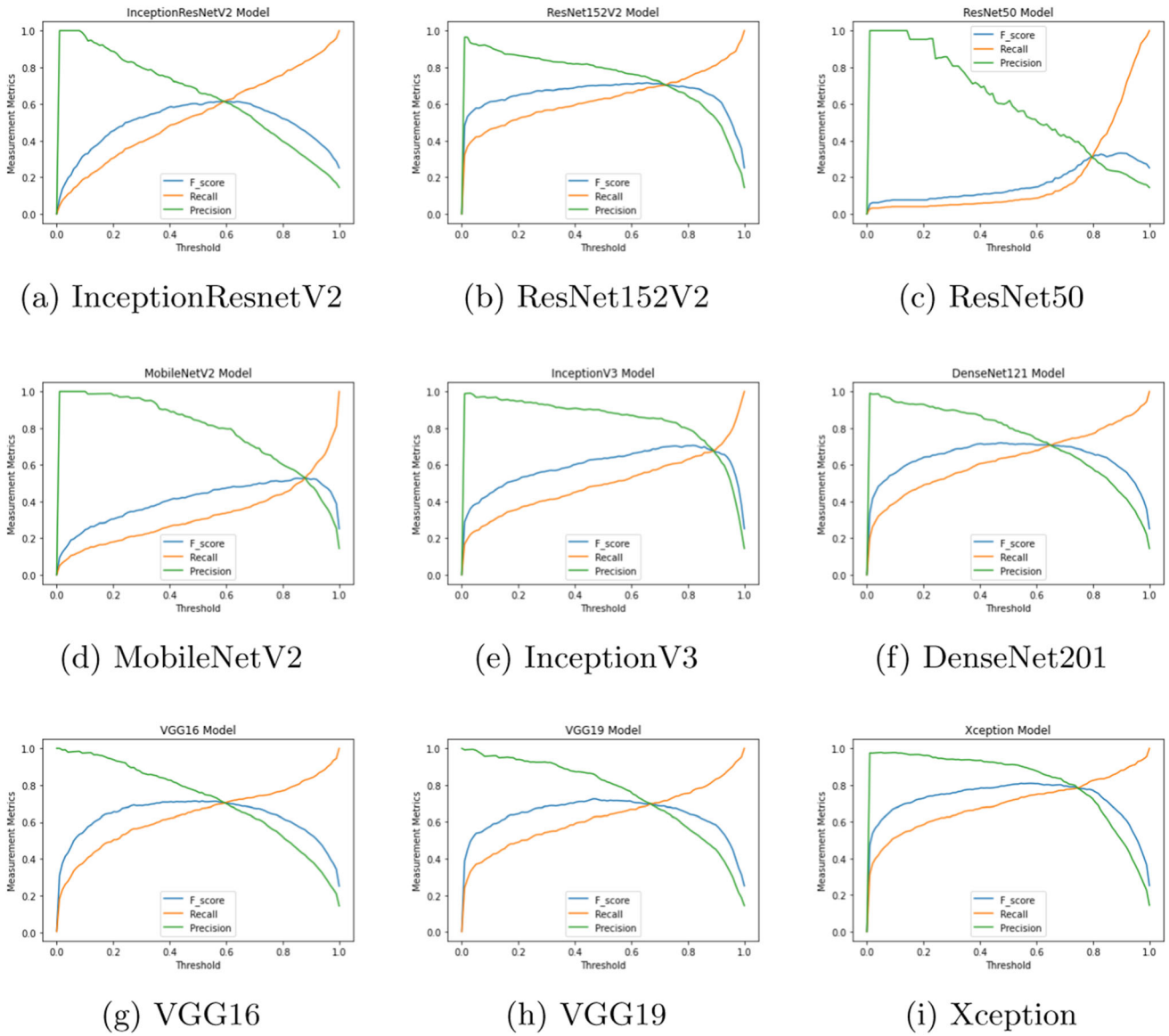


Fig. 10 Thresholding Effects on Recall, Precision, and F1 score

Table 4 Optimal threshold value for the nine models

Model	VGG16	VGG19	ResNet50	ResNet152_V2	DenseNet121	MobileNetV2	InceptionV3	InseptionResNet50_V2	Xception
Threshold	0.515	0.666	0.891	0.657	0.472	0.883	0.828	0.687	0.585

After a series of tests, the criteria for fine-tuning were established. Using the frozen layer parameter, one may specify how many layers of the CNN are untrainable because their weights do not change during model training. The bottleneck features parameter specifies the last feature map that was flattened during pre-training in order to feed a fully connected deep neural network classifier. For the tire defects detection problem, a variety of state-of-the-art pre-trained models are suggested and fine-tuned in this work. Figure 8 depicts the flowchart outlining the basis for this research.

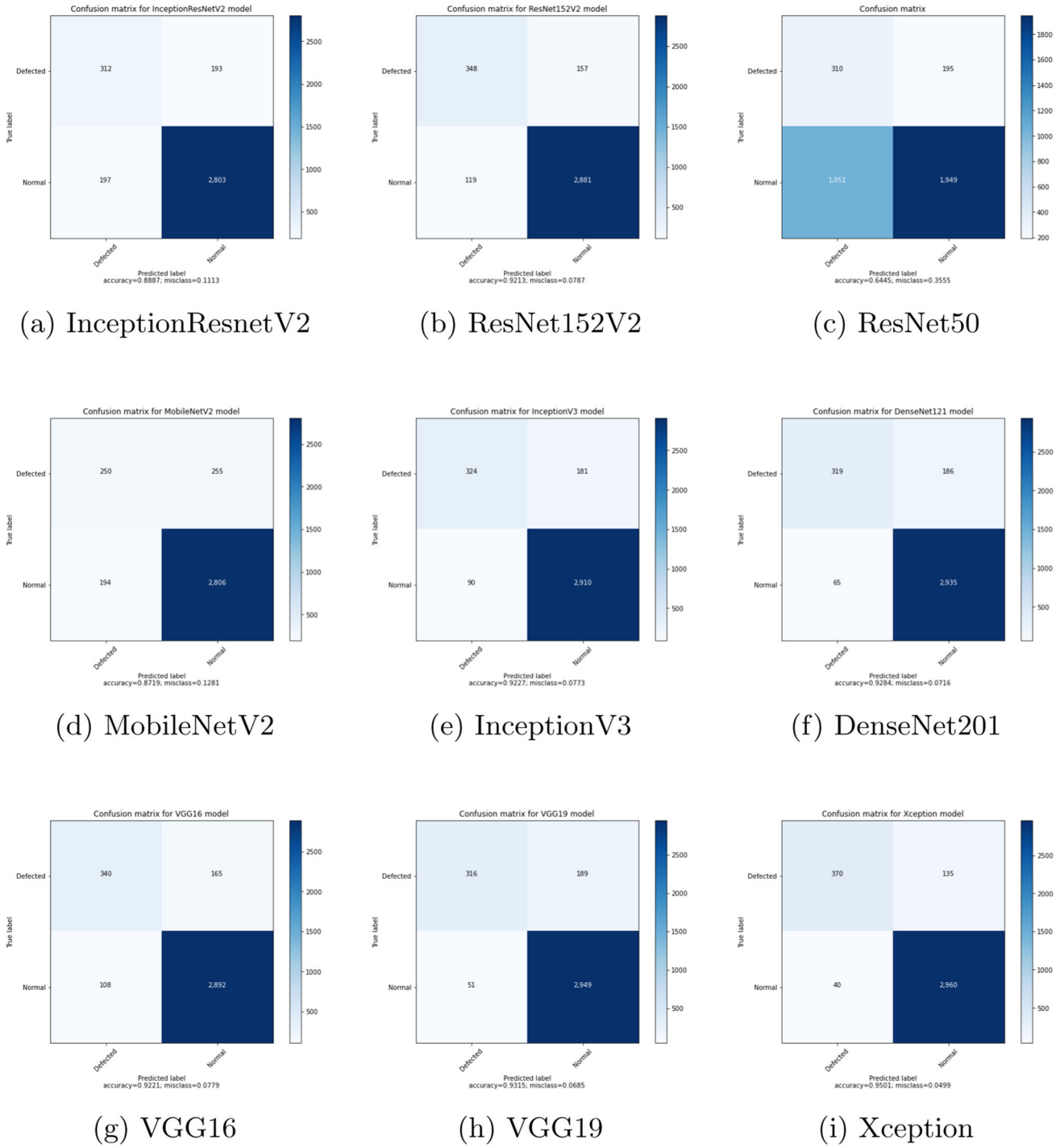


Fig. 11 Confusion matrices for validation dataset

The layers closer to the output features were trained to extract additional information from the subsequent convolution layers. As illustrated in Fig. 8, we added two additional fully connected layers to each pre-trained model and deleted its top layers. The first has 256 neurons and uses the ReLU activation function, while the second has one neuron and uses the sigmoid as the output function. The network was trained for 50 epochs with a learning rate of 0.00001 and a batch size of 32 using an RMSprop optimizer. Table 3 shows the best parameters for the frozen layer and bottleneck features in addition to the number of hidden layers (HL) for each model.

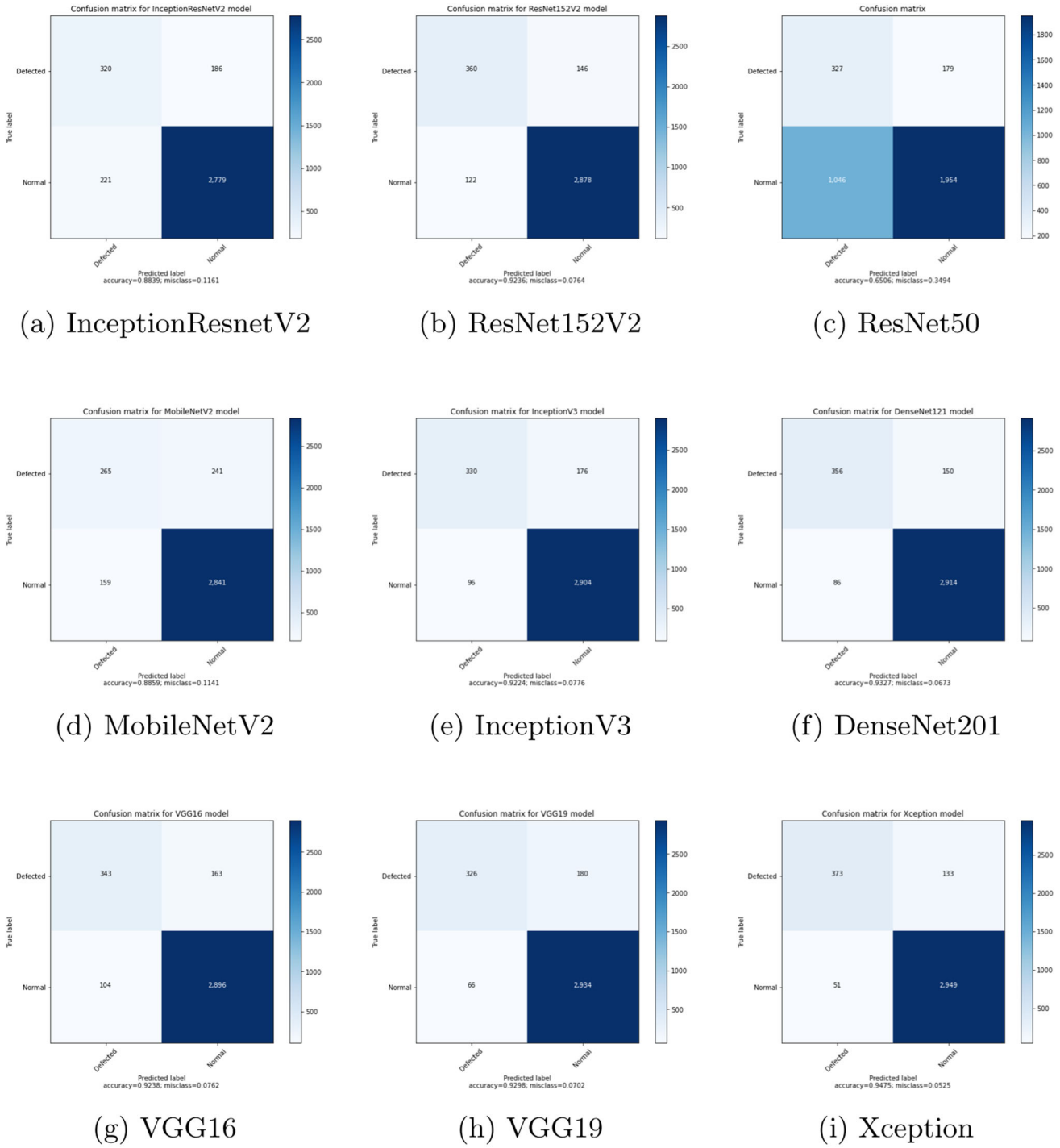


Fig. 12 Confusion matrices for testing dataset

4 Results and discussion

This section is divided into three subsections. First, the evaluation metrics used in this work are illustrated. Then, the experimental setup of our experiments is discussed. Afterward, the obtained results are discussed in the experimental results subsection. Ultimately, a comparison with classification-based previous works is presented and discussed.

Table 5 Testing results for the nine models

Model	Recall %	Precession %	F1 score %	Accuracy %
VGG16	67.8	75.9	71.62	92.38
VGG19	64.4	83.2	72.6	92.98
ResNet50	61.4	23.8	34.8	65
ResNet152_V2	68.9	74.7	74.42	92.13
DenseNet121	70.35	80.5	75.1	93.27
MobileNetV2	52.4	62.5	57	88.6
InceptionV3	65.2	77.5	70.82	92.24
InseptionResNet50_V2	61.8	61.3	61.55	88.4
Xception	**73.7**	**88**	**80.2**	**94.75**

The outcomes of the best model are highlighted in bold

Table 6 Validation results for the nine models

Model	Recall %	Precession %	F1 score %	Accuracy %
VGG16	67.3	76.7	72	92.21
VGG19	62.6	86.1	72.5	93.15
ResNet50	64.6	22.8	33.7	64.45
ResNet152_V2	71.15	74.5	72.79	92.36
DenseNet121	63.2	83.1	71.8	92.84
MobileNetV2	49.5	56.3	52.7	87.19
InceptionV3	64.2	78.3	70.55	92.27
InseptionResNet50_V2	63.2	59.1	61.1	88.87
Xception	**73.3**	**90.24**	**80.9**	**95**

The outcomes of the best model are highlighted in bold

4.1 Evaluation metrics

In this subsection, the performance of the proposed classification models is evaluated. To do this, accuracy, precision, recall, F1 score, and confusion matrix are used.

- *Accuracy*: it is the basic performance metric used in many models to compare them. This metric tells us how the model correctly classifies the tire's image, whether the tire is defective or not. Its value is between 0 and 100, where 100 means the model has the best classification rate and can correctly classify all tires' images. The formula used to calculate the classification rate of the model is given in Eq. (1).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

  where TP and TN are the number of correctly predicted frames for the defected and not defected tire classes, respectively. FP and FN are the number of wrongly predicted frames for the defected and not defected tire classes, respectively.
- *Recall*: the ratio of accurately predicted defected tires' observations to the total actual defected tires' observations is known as recall. Low false-negative rates are associated with high recall. The formula used to calculate the recall of the model is given in Eq. (2).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

- *Precision*: the ratio of accurately predicted defected tires' observations to the total predicted defected tires' observations is known as precision. Low false-positive rates are associated with high precision. The formula used to calculate the precision of the model is given in Eq. (3).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

- *F1 score*: weighted averages of precision and recall are used to get the F1 score. As a result, this score accounts for both true and false positives. The formula used to calculate the F1 score of the model is given in Eq. (4).

$$\text{F1 score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Recall} + \text{Precision}} \tag{4}$$

- *Confusion matrix*: confusion matrices are summaries of classification predictions that show a breakdown of the number of correct and wrong predictions for each class based on the count values.

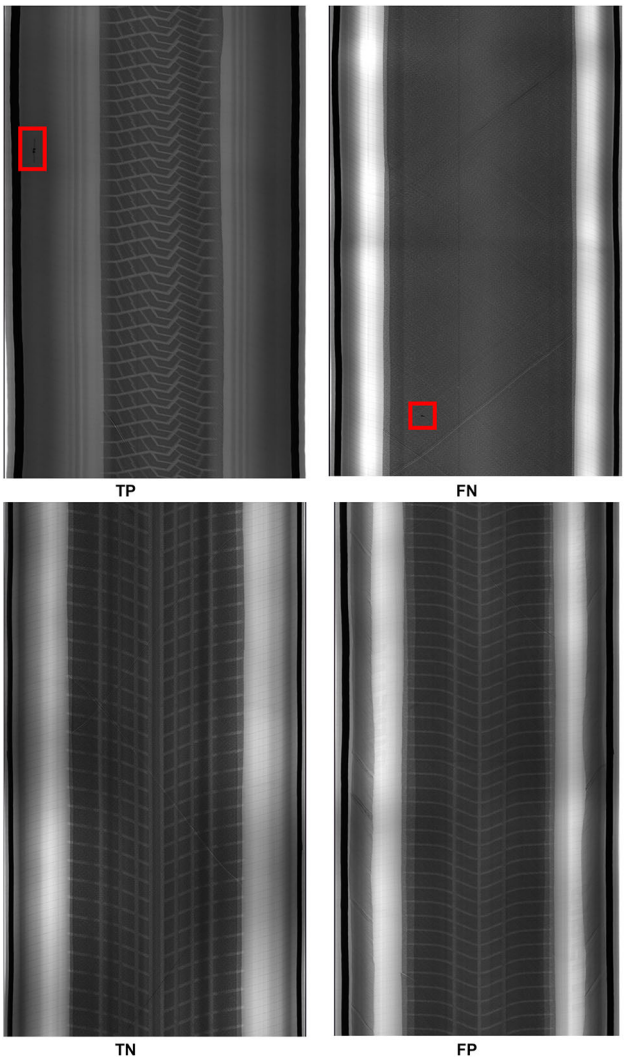


Fig. 13 Exemplar images representing true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) outcomes

4.2 Experimental setup

Implementing a learning process might be resource-intensive and time-consuming, depending on the quality and quantity of the dataset provided. Therefore, in this work, we have utilized the Nvidia GeForce RTX 3060 GPU to do our experiments. Moreover, the 70/15/15 rule is often utilized in the literature as a basis for learning, validation, and testing datasets [31]. Thus, in our experiments, we shuffled all images randomly and split them into 70% for training, 15% for validation, and 15% for testing. The same initial hyper-parameters, optimization algorithm, and loss function were used to train each individual model.

4.3 Experimental results

The proposed dataset for tire defect detection has been used to train and test the proposed architecture of the nine fine-tuned TL models. The convergence rates of the accuracy and the loss metrics of all models with respect to epoch number are shown in Fig. 9. This figure summarizes the convergence rates for both training and validation datasets. From the convergence rates of all models, it is clear that there is no over-fitting or under-fitting behavior. However, the Xception model shows the best performance in terms of over-fitting and under-fitting.

Furthermore, the inspection of the convergence curve figures reveals a significant divergence in the convergence of training and validation losses, which is especially noticeable for VGG19. This divergence can be attributed to the imbalanced nature of the dataset, acting as a significant contributing factor to this phenomenon. With a six-to-one advantage in qualified tire images over defective ones, the model's learning is skewed toward the dominant class, limiting convergence, particularly as training advances. This observation highlights the difficulties inherent in our research, which include not just the complexity of tire images with anisotropic multi-textured rubber layers and various patterns, but also the presence of 15 distinct fault kinds in an unbalanced dataset.

In binary classification, if a model returns a score rather than a prediction, we must normally apply a threshold to convert the score to a prediction. Because the score's meaning is to provide us with the perceived probability of obtaining 1 based on our model, 0.5 is an obvious threshold to apply. However, in most cases, 0.5 is not the optimal threshold. Thus, we tested all models on the validation dataset to find the optimal threshold for each model within the range between 0 and 1.

We face a complicated trade-off between precision and recall, and determining the ideal threshold proves to be a hard challenge. This complexity results from a variety of inherent difficulties in our issue, such as the complex structure of anisotropic multi-textured rubber layers, a wide range of faults, the sophisticated nature of tire designs, imbalanced datasets, and variability in environmental conditions. As a result, in our research, we focus on the F1 score metric, which provides a comprehensive assessment of model performance by combining precision and recall. By taking into account both false positives and false negatives, the F1 score provides a more complete assessment of the model's capacity to correctly identify defective tires while minimizing misclassifications.

Figure 10 shows the effect of changing threshold value on recall, precision, and F1 score. Table 4 shows the optimal threshold values for each model on the validation

Table 7 Comparison with previous works

References	Approach	Dataset	# Defects types	# Pattern designs	Measurement metrics
[9]	Texture based features	207 defect images 793 defect-free images	Only bubbles	Not Mentioned	Accuracy (85.9%) Recall (92.62%) Precision (89.65%)
[10]	Combining CNN and Faster R-CNN	136 defect images 262 defect-free images	Only bubbles	Not Mentioned	Accuracy (89.16%) Sensitivity (100%) Specificity (83.09%)
[11]	Extremum filter and Self-adapt local binarization	280 defect images	Bubbles and Impurities	Not Mentioned	Impurity Precision (97%), Impurity Recall (95.7%) Clear Bubble Precision (95.8%), Clear Bubble Recall (92%)
[17]	GLCM with Texture Features and ANN	235 defect images 1276 defect-free	Higher deviance and Impurities	Precision (100%), Recall (96%) F1 score (97.9%) Accuracy (99.34%)	
[12]	MC-CNN	1266 defect-free images 316 defect images	Six	Not mentioned	Accuracy (98.47%)
[13]	AlexNet	1266 defect-free images 316 defect images	Six	Not mentioned	Accuracy (96.51%)
[14]	VGG16 with FCN	700 training images 214 testing images	Four	Not mentioned	Accuracy (78.91%)
[15]	Merging an improved pyramid scene parsing network with an optimized YOLOv3	1450 defect images	Six	Not mentioned	Precision (91.39%) mIoU (87.86%)
[1]	TireNet Based on Siamese Network	20,000 defect images 100,000 defect-free images	Eleven	Not mentioned	Recall (83.31%) Precision (93.55%)
[16]	Unsupervised DCScNet	1450 defect images	Six	Not mentioned	Accuracy (96.8%)
[8]	STDC-Net Encoder (Unsupervised)	1218 training defect images 854 testing defect images	Not known	Not Mentioned	EM (0.915), L-mIoU (92.4%), mIoU (97.1%)
Our	**Xception-based**	**3366 defect images 20,000 defect-free images**	**Fifteen**	Fifty	**Recall (73.7%), Precision (88 %), F1 score (80.2%), Accuracy (94.75%)**

Outcomes of the proposed method

dataset. The table presents the recall and precision values corresponding to the best F1 score, highlighting the effectiveness of our approach in navigating the complexities of tire defect detection.

After choosing the optimal threshold for each model to get the maximum F1 score value on the validation dataset, we generated the confusion matrices using the validation and testing datasets. Thus, Figs. 11 and 12 show the confusion matrices of the validation and testing datasets, respectively. We notice the high similarity in confusion matrices of the same models in the validation and testing datasets which highlights the generalization property of the proposed models. Thus, the proposed models could successfully achieve this work's main two objectives: detecting defective tires regardless of the variety of specifications and designs and detecting defective tires despite the variety of defect types.

From confusion matrices, we can calculate the classification measurements metric as shown in Tables 5 and 6 for the validation and testing datasets, respectively. It is clear

from these results that the Xception model achieved the best results in terms of recall, precision, F1 score, and accuracy. It achieved 73.7, 88, 80.2, and 94.75% of recall, precision, F1 score, and accuracy, respectively, on the testing dataset. Moreover, it achieved 73.3, 90.24, 80.9, and 95% of recall, precision, F1 score, and accuracy, respectively, on the validation dataset.

To provide a more vivid demonstration of the effectiveness of the premier Xception model, Fig. 13 showcases exemplar images representing true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) outcomes. The figure depicts a noticeable trend in the Xception model's performance in terms of tire defect detection. Specifically, it demonstrates that the Xception model excels at detecting significant tire defects but suffers from tiny variations close to acceptable limits. Quality inspectors at the Pirelli Factory provide evidence to support this observed practice. They clarify that acceptable fault size limits differ depending on customer requirements. As a result, during labeling, certain tires that were declared okay may be considered defective by high-standard race companies due to the proximity of defect sizes to permitted limits. As a result, it is critical to recognize the inherent tradeoffs between sensitivity and specificity in defect detection models. While improving the model for greater sensitivity to tiny defects may reduce false negatives, it may also increase the chance of false positives, potentially resulting in unwarranted interventions or resource allocations.

4.4 Comparison with previous works

Table 7 summarizes the results of our proposed model compared with the classification-based approaches in recent studies. This table highlights how tire defects detection problem become more challenging with the increase of defective types. The accuracies of previous works' models were higher with limited defective types and lower with higher defective types. Thus, proposing a highly accurate model for this problem despite the number of defective types is the main challenge that we tried to overcome. Furthermore, our dataset comes from fifty different pattern designs, whereas each pattern has different structures and dimensions. These two facts make us believe that the proposed model shows impressive behavior.

5 Conclusion

We proposed a transfer learning-based model for tire defect detection in this study. The suggested model can identify whether there is a defect in the tire's image, regardless of the types of defective tires and the design patterns. First, we gathered and labeled a novel dataset consisting of 3366 images of faulty tires and 20,000 images of qualified tires. The gathered dataset comes from 15 types of defects that come from around 50 different design patterns. This challenging dataset was split into 70, 15, and 15% for training, validation, and testing, respectively. After that, Xception, InceptionV3, VGG16, VGG19, ResNet50, ResNet152V2, DenseNet121, InceptionResNetV2, and MobileNetV2 pre-trained models were fine-tuned, trained, and tested on the proposed dataset. The experimental findings demonstrate how the suggested fine-tuned Xception model outperformed the other fine-tuned models in terms of recall, precision, accuracy, and F1 score. Moreover, the fine-tuned Xception model achieved 73.7, 88, 80.2, and 94.75% of recall, precision, F1 score, and accuracy, respectively, on the testing dataset and 73.3, 90.24, 80.9, and 95% of recall, precision, F1 score, and accuracy, respectively, on the validation dataset. To sum up, this research could successfully achieve this work's main two objectives: detecting defective tires regardless of the variety of specifications and designs and detecting defective tires despite the variety of defect types. The success of the proposed fine-tuned model in building a highly generalizable system motivates us to extend this work in the future and deploy it in the real production line. More data must be gathered for future work to enhance the system's performance and try other deep learning techniques.

Acknowledgements The authors would like to acknowledge the support provided by The Scientific and Technological Research Council of Türkiye (TUBITAK, Project Number: 5210056), the Pirelli Automobile Tyres İzmit factory, and the Kocaeli University Software Technologies Research (STAR) Laboratory for this study. This article is an extension to our previous work wherein the dataset is gathered from 50 different design patterns with 15 types of defects for 3366 images of faulty tires and 20,000 images of qualified tires, unlike the previous [17] dataset, which was gathered from some design patterns with only two types of faults.

Author contributions Radhwan A. A. Saleh contributed to conceptualization, methodology, validation, investigation, data curation, writing—original draft, and visualization and provided software. Mehmet Zeki KONYAR was involved in review and editing, resources, and project administration. Kaplan KAPLAN contributed to review and editing, resources, and project administration. H. Metin ERTUNC was involved in review and editing, resources, project administration, and supervision.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availability The dataset that was used in this work is not openly available due to the policy that the Pirelli Automobile Tyres Izmit factory follows. However, all figures and training processes that support the findings of this study are included in the manuscript.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Li Y, Fan B, Zhang W, Jiang Z (2021) Tirenet: A high recall rate method for practical application of tire defect type classification. Futur Gener Comput Syst 125:1–9
2. Zhang Y, Lefebvre D, Li Q (2015) Automatic detection of defects in tire radiographic images. IEEE Trans Autom Sci Eng 14(3):1378–1386
3. Akay R, Saleh RA, Farea SM, Kanaan M (2022) Multilevel thresholding segmentation of color plant disease images using metaheuristic optimization algorithms. Neural Comput Appl 34(2):1161–1179
4. Xiang Y, Zhang C, Guo Q (2014) A dictionary-based method for tire defect detection. In: 2014 IEEE International conference on information and automation (ICIA). IEEE, pp 519–523
5. Guo Q, Zhang C, Liu H, Zhang X (2016) Defect detection in tire X-ray images using weighted texture dissimilarity. J Sens 2016
6. Zhao G, Qin S (2018) High-precision detection of defects of tire texture through X-ray imaging based on local inverse difference moment features. Sensors 18(8):2524
7. Cui X, Liu Y, Wang C (2016) Defect automatic detection for tire X-ray images using inverse transformation of principal component residual. In: 2016 Third international conference on artificial intelligence and pattern recognition (AIPR). IEEE, pp 1–8
8. Yi X, Peng C, Zhang Z, Xiao L (2022) The defect detection for X-ray images based on a new lightweight semantic segmentation network. Math Biosci Eng 19(4):4178–4195
9. Saleh NAA, Konyar MZ, Kaplan K, Ongir S, Ertunç HM (2022) Detection of air bubbles from tire shearography images. In: 2022 International congress on human-computer interaction, optimization and robotic applications (HORA22), pp 1–4
10. Chang C-Y, Wang W-C (2018) Integration of CNN and faster R-CNN for tire bubble defects detection. In: International conference on broadband and wireless computing, communication and applications. Springer, Berlin, pp 285–294
11. Zheng X, Ding J, Pang Z, Li J (2018) Detection of impurity and bubble defects in tire X-ray image based on improved extremum filter and locally adaptive-threshold binaryzation. In: 2018 International conference on security, pattern analysis, and cybernetics (SPAC). IEEE, pp 360–365
12. Cui X, Liu Y, Zhang Y, Wang C (2018) Tire defects classification with multi-contrast convolutional neural networks. Int J Pattern Recognit Artif Intell 32(04):1850011
13. Zhang Y, Cui X, Liu Y, Yu B (2018) Tire defects classification using convolution architecture for fast feature embedding. Int J Comput Intell Syst 11(1):1056
14. Wang R, Guo Q, Lu S, Zhang C (2019) Tire defect detection using fully convolutional network. IEEE Access 7:43502–43510
15. Zheng Z, Zhang S, Shen J, Shao Y, Zhang Y (2021) A two-stage CNN for automated tire defect inspection in radiographic image. Meas Sci Technol 32(11):115403
16. Zheng Z, Shen J, Shao Y, Zhang J, Tian C, Yu B, Zhang Y (2021) Tire defect classification using a deep convolutional sparse-coding network. Meas Sci Technol 32(5):055401
17. Saleh RA, Konyar MZ, Kaplan K, Ertunç HM (2022) Tire defect detection model using machine learning. In: 2022 2nd International conference on emerging smart technologies and applications (eSmarTA). IEEE, pp 1–5
18. Khalil RA, Saeed N, Masood M, Fard YM, Alouini M-S, Al-Naffouri TY (2021) Deep learning in the industrial internet of things: potentials, challenges, and emerging applications. IEEE Internet Things J 8(14):11016–11040
19. Espitia FA, Soto LR (2020) Novel methods based on deep learning applied to condition monitoring in smart manufacturing processes. In: New trends in the use of artificial intelligence for the industry 4.0. IntechOpen, p 49
20. Al-Areqi F, Konyar MZ (2022) Effectiveness evaluation of different feature extraction methods for classification of covid-19 from computed tomography images: a high accuracy classification study. Biomed Signal Process Control 76:103662
21. Wang J, Jiang C, Zhang H, Ren Y, Chen K-C, Hanzo L (2020) Thirty years of machine learning: the road to pareto-optimal wireless networks. IEEE Commun Surv Tutor 22(3):1472–1514
22. Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang J et al (2016) End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316
23. Glasmachers T (2017) Limits of end-to-end learning. In: Asian conference on machine learning. PMLR, pp 17–32
24. Couture J, Lin X (2022) Image-and health indicator-based transfer learning hybridization for battery RUL prediction. Eng Appl Artif Intell 114:105120
25. Subramanian M, Shanmugavadivel K, Nandhini P (2022) On fine-tuning deep learning models using transfer learning and hyper-parameters optimization for disease identification in maize leaves. Neural Comput Appl 1–18
26. Chen Y, Yi Z (2021) Adaptive sparse dropout: learning the certainty and uncertainty in deep neural networks. Neurocomputing 450:354–361
27. Roy AM (2022) Adaptive transfer learning-based multiscale feature fused deep convolutional neural network for EEG MI multiclassification in brain–computer interface. Eng Appl Artif Intell 116:105347
28. Saleh RAA, Ertunç HM (2022) Development of a neural network model for recognizing red palm weevil insects based on image processing. Kocaeli J Sci Eng 5(1):1–4
29. Saleh RAA, Rüştü A (2019) Classification of melanoma images using modified teaching learning based artificial bee colony. Avrupa Bilim ve Teknoloji Dergisi 225–232
30. Dubey SR, Singh SK, Chaudhuri BB (2022) Activation functions in deep learning: a comprehensive survey and benchmark. Neurocomputing
31. Coulibaly S, Kamsu-Foguem B, Kamissoko D, Traore D (2019) Deep neural networks with transfer learning in millet crop images. Comput Ind 108:115–120

32. Chakrapani G, Sugumaran V (2023) Transfer learning based fault diagnosis of automobile dry clutch system. Eng Appl Artif Intell 117:105522
33. Bansal M, Kumar M, Sachdeva M, Mittal A (2021) Transfer learning for image classification using VGG19: Caltech-101 image data set. J Ambient Intell Humaniz Comput 1–12
34. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
35. Polat Ö, Polat A, Ekici T (2021) Automatic classification of volcanic rocks from thin section images using transfer learning networks. Neural Comput Appl 33(18):11531–11540
36. Hamida S, El Gannour O, Cherradi B, Raihani A, Moujahid H, Ouajji H (2021) A novel covid-19 diagnosis support system using the stacking approach and transfer learning technique on chest X-ray images. J Healthc Eng 2021
37. Garg A, Salehi S, La Rocca M, Garner R, Duncan D (2022) Efficient and visualizable convolutional neural networks for covid-19 classification using chest CT. Expert Syst Appl 116540
38. Arora G, Dubey AK, Jaffery ZA, Rocha A (2022) A comparative study of fourteen deep learning networks for multi skin lesion classification (MSLC) on unbalanced data. Neural Comput Appl 1–27
39. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
40. Alkhawaldeh RS, Alawida M, Alshdaifat NFF, Alma'aitah W, Almasri A (2022) Ensemble deep transfer learning model for Arabic (Indian) handwritten digit recognition. Neural Comput Appl 34(1):705–719
41. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258
42. Ghosh S, Bandyopadhyay A, Sahay S, Ghosh R, Kundu I, Santosh K (2021) Colorectal histology tumor detection using ensemble deep neural network. Eng Appl Artif Intell 100:104202

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.