IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 May 2024, accepted 16 July 2024, date of publication 23 July 2024, date of current version 20 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3432597

RESEARCH ARTICLE

Spliced Multimodal Residual18 Neural Network: A Breakthrough in Pavement Crack Recognition and Categorization

ASAD ULLAH[1], ZHAOYUN SUN[2], SYED AGHA HASSNAIN MOHSAN[3], AMNA KHATOON[4], ADIL KHAN[1], AND SHABEER AHMAD[5]

[1]School of Information Engineering, Xi'an Eurasia University, Xi'an, Shaanxi 710065, China
[2]Department of Information Engineering, Chang'an University, Xi'an 710064, China
[3]College of Information Science and Technology, Eastern Institute for Advanced Studies, Ningbo 315200, China
[4]School of Information Engineering, Chang'an University, Xi'an, Shaanxi 710064, China
[5]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding authors: Zhaoyun Sun (zhaoyunsun@126.com) and Asad Ullah (asadullah@eurasia.edu)

ABSTRACT Machine learning and deep neural network developments have evolved image classification. This study offers a new neural network design with segmentation and localization, different crack types, categorization accuracy, and efficiency. In the proposed work, an advanced and innovative Spliced Multimodal Residual18 Neural Network (SMR18-NN) Model is presented that departs from previous models. For data authentication, augmentation is done by proposing the new and effective model known as an Augmented Minority Over-sampling Technique (AMOST). The SMR18 NN model combines the well-established ResNet18 framework with the Faster RCNN architecture, incorporating parts from the modified Fast RCNN and the Region Proposal Network. This work aims to revolutionize crack-type recognition and categorization in images. A Support Vector Machine strategically improves the network's data classification. A modified ResNet18 model is ultimately implemented and compared with the proposed innovative SMR18-NN model. Both network's parameters, such as epochs, number of iterations, etc., were kept the same for fair evaluation. Innovative frameworks and properly selected benchmark datasets supported this. The empirical results of this comparison study are convincing. ResNet18 training and testing accuracy was 90.60% and 85.20%, respectively. The SMR18-NN outperformed these results with 96.20 training and 92.00% testing accuracy. The experiment concluded with SMR18-NN accurately detecting image features, proving its superiority in image classification.

INDEX TERMS Faster RCNN, ResNet18, road crack image, road cracks, spliced multimodal Residual18 neural network, localization.

I. INTRODUCTION

Any image's background significantly impacts the brightness or contrast of an image [1]. Traditionally, visionary engineers know the value and worth of paved road maintenance and rehabilitation. Its significance is now being understood in a new context and with a new concept [2]. In the past, civil inspectors were used to gathering information on the road surface as they drove or walked [3]. It was a dangerous job to be done. The entire process considered was time-consuming, laborious, and expensive. It is more challenging because the task must be completed in busy traffic. The safety of the impacted members will be in jeopardy. The pavement crack quality is tested using various methods and techniques, including image processing, infrared detection, and ultrasonic testing. Another technique called WiseCrax employs infrared imaging to find cracks [4]. Umer Farooq uses infrared images for pedestrian detection [5]. These techniques are still being made, and impressive research is being done on advancing pavement cracks. The literature has

The associate editor coordinating the review of this manuscript and approving it for publication was Domenico Rosaci.

suggested a strategy for crack width detection to enhance the precision of crack detection. This strategy involves extracting crack images from surface disturbances on paved surfaces.

The biological neural network system inspires Artificial Neural Networks (ANN) [6]. It isn't easy for the computer to recognize and give output as humans do. More particularly, when faced with an unfamiliar object, humans can quickly learn the patterns from even a single example and correctly identify similar ones. It is a programming worldview with which a system can gain from accessible observational information. The pattern recognition approach is also used for different purposes [7]. A feed-forward neural network is created by forming a directed acyclic graph for a neural network (NN) [8]. Artificial neurons receive input signals from previous neurons (x0), and this signal is multiplied by a weight (wi) to model the interaction of dendrites. Subsequently, these weighted input signals are summed, and a fixed bias is added and fed into the activation function, which generates the output. The weights are adjusted based on the data labels to learn and estimate the inference [9]. Current automatic real-time detection systems have a low entry rate and difficulty in soldering. However, no precise algorithm is available for identifying and classifying pavement crack images [10].

The size of a CNN model can have a significant impact since it comprises several layers, including convolution, pooling, ReLU, and fully connected layers [11]. The core layer is the convolution layer because it plays a vital role in performing complex matrix multiplication. Several crack recognition and segmentation techniques are available. Nevertheless, several of these techniques aim to decrease the model size, considerably decreasing accuracy. Although much research has been conducted on pavement crack detection [12], there is still room for improvement, specifically in reducing the processing time, increasing accuracy, and building unique neural networks. Multiple research studies were done using different techniques, such as the researcher's work on masonry structure in the article published in the Journal of Geomate [13]. Many machine learning techniques, including traditional techniques like the general linear model and multivoxel techniques, have been used for categorization [14]. Finding a suitable algorithm for suitable kinds of cracks was also a problem. The same issue was finding the most appropriate algorithms like VGG, which did not give satisfactory results for finding non-crack or crack images in the dataset. However, the same VGG is good for finding abnormal human behaviour [15]. The deep learning algorithms utilized are ResNet18 and SMR18-NN. The purpose of using these neural networks is that the ResNet18 model from Microsoft Research Asia won the ILSVRC-2015 object identification competition. This model is also used to compare AlexNet and SqueezeNet. So, this model is utilized here for better comparison and analysis [16]. The proposed research concerns pavement cracks, their identification, classification, and road safety [17].

The aim and the achievement after the completion of this research are:

- The proposed new algorithm SMR18-NN is compared with existing altered networks ResNet18. Its modification, implementation, and accuracy result is compared.
- The scope of the study is to automatically identify and classify cracks in the pavement surface based on the dataset. Pavement crack detection and segmentation through a deep neural network are to be implemented to find the best and most accurate model for such a small dataset with minimal possible computational time.
- To build Geo-Free dataset robust pavement crack datasets and classify them into respective classes. Automatic dataset building will decrease human interference errors.

One of the contributions also involves resizing and reshaping the images into a format suitable for processing by multiple networks. A small dataset of 4333 images is built, though other researchers did the research using 21k, 60K, and 100K images. Training and testing are done using the algorithms ResNet18 and SMR18. Classification into 8 different classes is done, and the localization task is performed at the end. Road accidents are preventable only with good paving and crack care.

II. LITERATURE REVIEW

The review was conducted by researchers Byunghyun Kim and Soojin Cho from the University of Seoul's Department of Civil Engineering. Their investigation only involved two binary detections in determining whether an image has a crack or crackles, but the average accuracy was good (96.64%), and the precision was 86.73 [18]. The exploration in a publication by Zhun Fan, Senior Member, IEEE, Yuming Wu, Jiewei Lu, and Wenji Li, shows a precision gain of 91.78%; nevertheless, their training images are 7,41,932 their testing images are 70,65,600 [19]. A significant number of images in the dataset will produce impressive results. According to Steward and Tian, segmenting images recorded in sunny environments was more prone to pixel noise and edge variability [20]. The classification accuracy (CA) may be impacted by the images' brightness, size, and colour (Amith, 2020) [21]. According to researchers, image processing helps remove typical features that may be used to train and evaluate the algorithms that produce the CA [22]. The same deep learning measures COVID-19 for decision-making and aggregation via the RISTEC-B model [23]. It has demonstrated exceptional achievements in object detection, language processing, biology, etc. Some researchers are working on nighttime images, as the researcher did research and published the article in Scientific Reports [24]. All the disciplines mentioned earlier and other engineering have used deep learning, but some research is also taking place without deep learning [25]. Due to their benefits, including safety, uncrewed

aerial vehicles (UAVs) have been the subject of more and more studies on bridge inspection. The advancement of deep learning has prompted significant research towards creating datasets, automation, and discriminative methods for feature extraction [26].

The maintenance of roads is of utmost importance for the development of any country. The Ministry of Local Transportation must maintain roads effectively to reduce road accidents. Road cracks can be broadly categorized as structural or non-structural [27]. Structural cracks may be caused by incorrect design, faulty construction, or overloading, while non-structural cracks are typically the result of internally induced stresses, moisture penetration, and thermal variation. Cracks can also be classified as thin or small (less than 1 mm in width) or medium-sized (1 to 2 mm in width). This research has focused on wide or big cracks over 2 mm in width [28]. Scientists M. Hofacker and his colleague scientist C. Miehe explain the diversity of complex cracks with mathematical proof and deep analysis in the research article, ''A phase field model of dynamic fracture: Robust field updates for the analysis of complex crack patterns'' [29]. This research aims to discover a cost-effective method for upgrading road cracks using different kinds of neural networks [30] and [31]. In Korea, the proportion of structures older than thirty years was estimated at 3.8% in 2014, slightly developed in the 1970s. Experts predict that this percentage will grow exponentially and reach 13.8% in 2024 and 33.7% in 2029. In the US, the overall condition of transportation infrastructure is rated as D+ on average, and the cost of rehabilitation is anticipated to be $123 billion [32].

III. FRAMEWORK OF IMPLEMENTED MODEL

Prior research has demonstrated that deep learning models are highly effective for recognizing cracks and classifying images [33]. The models are capable of automatically and efficiently extracting features from data. Deep neural networks (DNNs) function as hierarchical feature representation frameworks that use back-and-forward propagation methods to fine-tune the network, producing the desired classification outcome [34]. Although there was complexity due to the complicated background, this experiment used a low-budget device instead of the specialized optical devices that the present methods demanded. The performed research work is unique because of the internally modified structure and the unique parameters during the experimentation. Like in ResNet18, by default, there is a fully connected layer with an output size of 1000, but in the proposed case, the output layer finds and differentiates 8 classes. So, changes are made accordingly.

This research route and process are crystal clear in Figure 1. Initially, data was collected with the help of a built setup. After that, manipulation is done by mixing different available datasets online. After filtering the dataset, a new dataset is formed. After training, the testing phase was implemented, and the result was in the form of F1 score, precision, and recall.

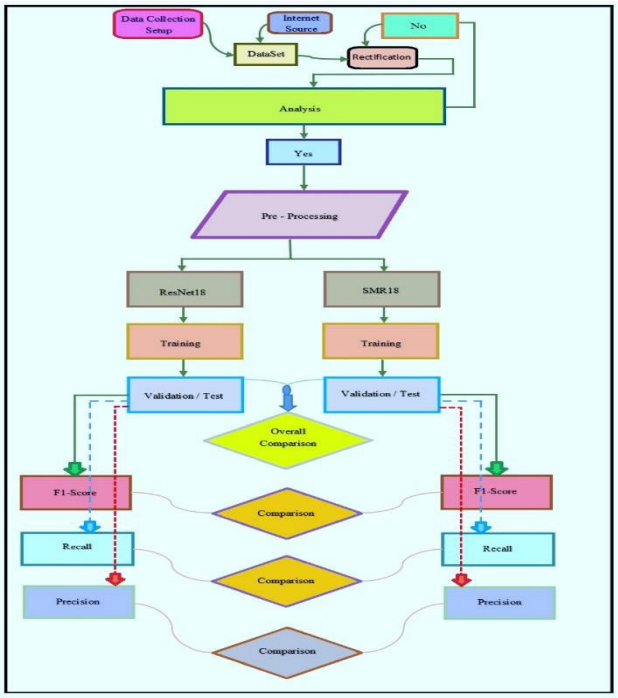


FIGURE 1. Overall outline of the research.

IV. DATA COLLECTION AND ANALYSIS PHASE

Data collection serves as the foundation and backbone of any research project. While creating a dataset is not considered innovative work, it is a difficult task under a complex background [35]. Still, the difference between the proposed dataset and the existing one is that the proposed dataset is built in a developing country. In contrast, the existing datasets are built mainly in developed countries [36]. The pavement images data-gathering system comprises three parts, as shown in Figure 2. The pavement surface images are illuminated by a vehicle-mounted high-definition linear array CCD camera in the data collection process. The relating line laser gives extra lighting so the camera can store the image in the system's hard drive. Cracked and crackles images are obtained from the assembled setup. As the images were acquired through the setup, it was challenging for the researcher to validate them concerning crack or non-crack images. For this purpose, help was given by an expert, Amna Khatoon. She helped to differentiate between the images and whether the images had cracks or crackles.

Once it is done, with the help of another expert, different classes are made, and different crack types are assigned to the concerned crack class. It was done by an expert, Ishfaq Ahmad, and it was a paid task costing 2,000$. The total collected images were 80000 by the established setup.

A. AUGMENTATION

In the image classification domain, deep convolutional neural networks have made remarkable progress in recent years and have also been employed for pixel-level pavement

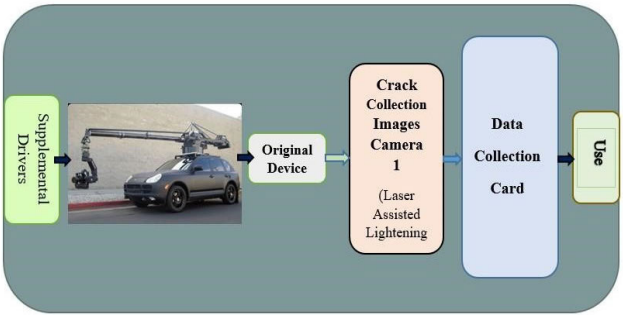
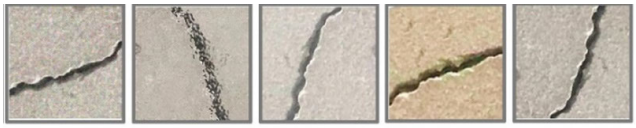


FIGURE 2. System composition of data collected from the vehicle.

crack segmentation [37]. Nevertheless, a diverse set of image datasets is required to achieve accuracy and robustness in training deep convolution models. If there isn't enough diverse training data, regularization techniques, including dropout and batch normalization, are frequently used to avoid overfitting [38]. Data augmentation was implemented during the research. Data augmentation is a technique used to generate new samples similar to those in the training set, which can be seen as a form of regularization. Data augmentation has been extensively used in early works for image classification tasks. Once the data is collected, augmentation makes the dataset robust [39]. The research study showed that the quantity and variety of data determine most neural network models' success. The initial aim of data augmentation is to create or add up data for models. Data augmentation enhances machine learning models' performance and outcomes, which generates novel and varied instances for training datasets. Data augmentation techniques are practical tools to address the challenges faced by the artificial intelligence field in neural network applications.

Additionally, a rich and diverse dataset results in better accuracy and performance of the model [40]. Another benefit of data augmentation is that it minimizes operating expenses by transforming datasets. Data augmentation approaches improve the robustness of neural network models by introducing variables the model encounters in the real world. Some researchers did not implement augmentation and class-making algorithms but directly implemented them in real-time for road crack detection [41]. Computer vision applications use standard data augmentation techniques for training data. Simple and advanced data augmentation techniques are available for image datasets. Modifications to visual data are commonly used in data augmentation [42].

Random rotation, rescaling, and horizontal/vertical flipping are examples of image processing operations utilized in data augmentation. Additionally, cropping, zooming, color adjustment, darkening, and brightness are good techniques used in image augmentation [43]. Advanced data augmentation techniques, such as adversarial training/machine learning, are being developed wherein adversarial instances that disrupt machine learning models are created and added to the dataset for training [44]. Generative Adversarial Networks (GANs) utilize input datasets to learn patterns and generate new instances similar to the training data [45]. Data augmentation has various advantages, including minimizing data overfitting. Overfitting occurs when a function closely fits a limited number of data points and increases the data variability. Data augmentation also assists in improving the models' generalization ability, resolving class imbalance issues in categorization, and reducing data collection and labeling costs. Moreover, some category-free image transformation methods are utilized for image classification tasks to generate new samples from the training set. The commonly used image transformation techniques are listed in Figure 3.



(a) Random-cropping of Original Image (b) Color-jittering (c) Blurring (d) Color noise (e) PCA-jittering

FIGURE 3. From a-e data augmentation with an original image and augmented images.

As data augmentation techniques become more prevalent, verifying the quality of their outputs becomes essential. Creating synthetic data with advanced applications using data augmentation requires further research and study. For instance, generating high-resolution images using GANs can pose challenges. Thus, data augmentation may have some drawbacks that require addressing. The enhanced data will also have biases if the original dataset has biases. Data augmentation increases the training sample's size through various data transformations. However, the transformations may occasionally differ depending on the dataset being used. For example, the MNIST dataset, which consists of handwritten digits, can be transformed by rotations or changes in brightness [46]. However, horizontal or vertical reflections of these images may violate the logical consistency of the dataset. To balance the crack classes, a new over-sampling technique, known as the Augmented Minority Over-sampling Technique (AMOST), is implemented in the research work. In the proposed algorithm, the improvement is done in AMOST in terms of weights of variables: for each minority class instance g_i with p variables, randomly select one of K neighbour instances g_i^k, usually K = 5. The distance between two instances is calculated by Euclidean distance gap_i^k is shown in equation 1.

$$gap_i^k = \left(g_i - g_i^k\right) = \sqrt{\sum_{j=1}^{p}} \left(g_{ij} - g_{ij}^k\right)^2 \tag{1}$$

The j^{th} variable element generates a new instance, as in Equation 2.

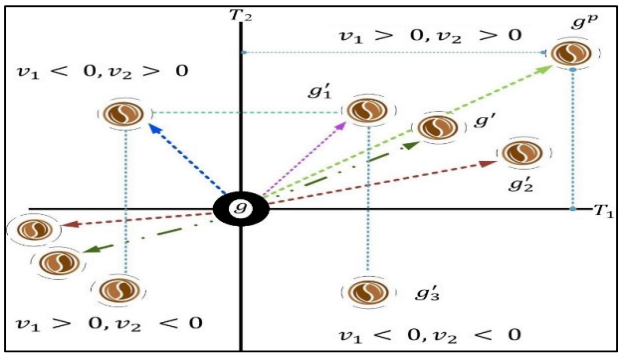
$$g_{ij}^{\prime k} = g_{ij} + w_j \times r_i^k \times gap_i^k \tag{2}$$

where r_i^k is in-between 0 and 1, a random number j, and k is the number from 1,2 …upto p and K. Equation 2 shows each variable weight by w_j. The generated sample by these

variable weights is symmetric. The proposed algorithm works on two algorithms. Algorithm 1 calculates variable weight and minority sample. There is a direct relationship between the new samples and minority sample weights. Algorithm 2 calculates the Case Weight. Different initial weights are assigned with support and non-support vectors to the minority samples in algorithm 1. This sample is predicted to obtain additional weights. It works on the line segmentation portion, where the algorithm randomly selects a Z neighbor based on Equation 3.

$$g' = g + (r \times (g - g^p) \tag{3}$$

According to equation 3, the random number r and g^p is always in-between 0 and 1, randomly elected among Z's nearest neighbors g. The g is the minority sample. The above equation can be repeated to obtain the required synthetic minority instances. Figure 4 shows the variables $T1$ and $T2$ on the x-axis and y-axis respectively. The weights of these variable $T1$ and $T2$ are shown as $v1$ and $v2$. These variable weights have an impact on both the dimensions of the generated samples. AMOST also generates the synthetic sample of the minority class, using the Z minority nearest neighbor to the nearest sample. It creates synthetic sample g' along the line segment connecting the minority sample g, and nearest neighbor g^p. Due to the weights and direction dissimilarity, the generated samples are non-linear blends of g and g^p. When both $v2$ and $v1$ are greater than 0, the generated $g'1$ is close to its nearest neighbor g^p. In the scenario, where $v2 > v1$, then the value and direction of $v2$ are supposed to have a significant impact on generated samples. As a result, the direction of g'_1 is more inclined to $v1$. On the other hand, when both $v1$ and $v2$ are less than 0, the direction of g'_3 will be opposite to g'_1.



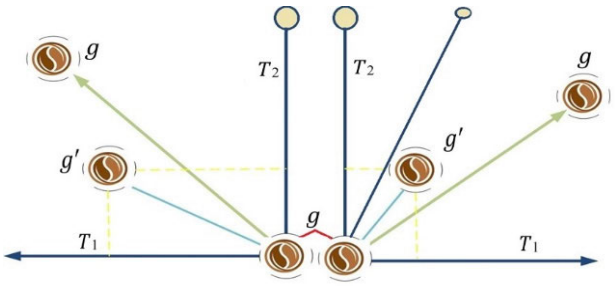
g^p *and* g = Minority Sample, $g'_1, g'_2,$ *and* g'_3 are Synthetic Sample, g' is Synthetic Sample by AMOST.

FIGURE 4. The distribution of the generated sample with different variable weight.

In the proposed algorithm, the element of j^{th} variable g'_{ij} of a newly generated sample g'_i is define as.

$$g'_{ij} = g_{ij} + r \times (g_i - g_i^k) \times w_j \tag{4}$$

where r is a random number between [0, 1]. The variable g_i^k is K's neighbors in the minority sample g_i, and w_j stands for the weight of the variable, the value of j is in-between 1 and p. When the value of each weight is shown as w_1 to w_p, is equal to w, then all these weights have the equal impact on the generation samples. The same phenomenon is true for AMOST synthesizes to create new instances by the combination of minority instances with its neighbor. The minority instance is shown by g_i and it's neighbor is shown by g_i^k in the Equation 4. As a result, the variable has a varied influence on sample generation. For example, w_1 is more significant than other variable weights $w_2 \ldots w_P$; thus it is assumed that g'_{i1} plays a more critical role on g'_{ij}, resulting in more impact of g'_{i1} on the generation of g'_i. The minority class distribution is optimized by introducing the variable weight. These weights prevent to creation of the resembled minority new samples. When the boundary samples are oversampled, AMOST may intensify overlaps because of the neighboring point on the classification edge. Class overlap is prevented, and the generated sample doesn't always interchange with the neighboring samples. Figure 5 shows the variable correlation.

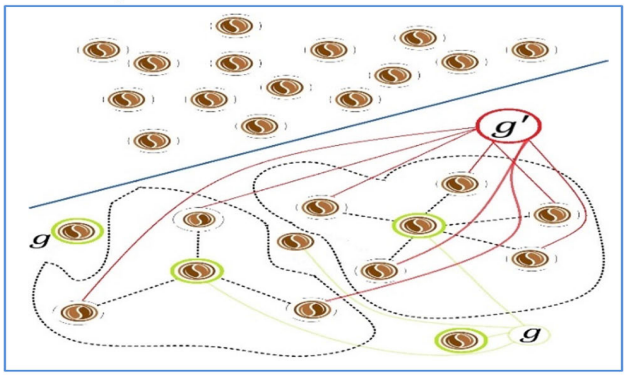


g = Minority Sample, **g′** are Synthetic Samples on the axis of T1 and T2.

FIGURE 5. The Distribution of the generated samples with correlative variables.

Our algorithm proposes a new adaptive oversampling method that weighs minority instances in sample space. Based on the SVM classifier, the minority instances are divided into non-support vectors and support vectors. In contrast, the number of support vectors is s, and the number of non-support vectors is m−s (m denotes the number of minority samples). We assign the initial weight of the support vector to (s/m) and the initial weight of the non-support vector to (1−(s/m)). Since the separating hyperplane is mainly affected by support vectors, we further add different weights to the support vectors according to the accuracy of predicting the sample generation of each support vector; after that, the weight of each support vector is the initial weight plus additional weight, while the weight of non-support vectors remains unchanged. According to the different weights of each minority sample, the new samples are generated by selecting the nearest neighbor adaptively. As shown in Figure 6, different minority instances are due to the difference in weight, so the generated instances with the nearest neighbors are different.

In this method, new fictitious minority samples are chosen by interpolating between original minorities. This method is unique as it didn't do simple duplication based on the



g = Minority Sample, g' are Synthetic samples, and the majority of the samples are above the incline plan line (Without the Assign symbol).

FIGURE 6. The influence of case weight on the distribution of the generated samples.

original sample. The essential advantage of the other methods and the proposed method is that introducing the variable weight improves the distribution and weakens the collinearity of minority samples. In the proposed method, the estimation is done with a vector v from the hyperplane. While ADASYN applies different weights on each minority sample, different numbers of samples are generated. However, in high-dimensional space, the distance between the samples is approximately equal, adversely affecting the definition of the nearest neighbor and thus affecting the distribution of the generated samples. After the augmented and AMOST, Table 1 shows the overall number of images in each class at the end of this research article.

TABLE 1. Summary of dataset numbers.

Crack Type	Number of Images
Alligator	538
Transverse	529
Complex	503
Non-cracks (Crackles images)	491
Wide Crack (Big Crack images)	590
Longitudinal	567
Pothole	555
Sealed	560

Table 1 shows that each class has a specific number of pavement cracks/crackles images. The dataset is indispensable to evaluate the network model's performance. The dataset used in this research was about 4333 images with 8 different classes. These classes cover a broad range of crack types. These classes are alligator, transverse, complex, non-cracks, wide crack, longitudinal, pothole, and sealed classes with the number of images of 538, 529, 503, 491, 590, 567, 555, and 560, respectively. The framework includes stages such as database creation, network architecture development, crack detection, classification, and localization.

B. RESIZING OF IMAGE

The dataset can directly handle RGB (red, green, and blue), which are true color images, but the size is an issue due to the imbalanced image size during the building of the dataset. The dimensions are converted to processible dimensions concerning neural networks. It is possible to convert all images or use another methodology manually. However, BULKRESIZE online is the quickest and easiest option [47]. Height, width, format, and image quality can all be changed along with the background color using the BULKRESIZE option. Engineer Asad Ullah also used the same research method [48]. In addition to this, there is yet another method [49]. With this method, the images are cropped following the necessary input for the respective neural networks to process.

As seen in Figure 7, the image's left side was originally $512\times512\times3$, but it was changed to a size that could processed by the neural network. Like the illustration above, all images are resized and saved in the storage device. After that, the final cropped images are assigned to the appropriate class. The classes are manually configured to ensure accuracy. Additionally, this is done to prevent the display of any images that have been cropped or resized so that the crack no longer exists. Figure 8 displays some randomly selected training images that have been cropped. They are chosen randomly and contain images of cracks and non-cracks, including pothole, alligator, big or wide, complex, etc. cracks images, respectively.

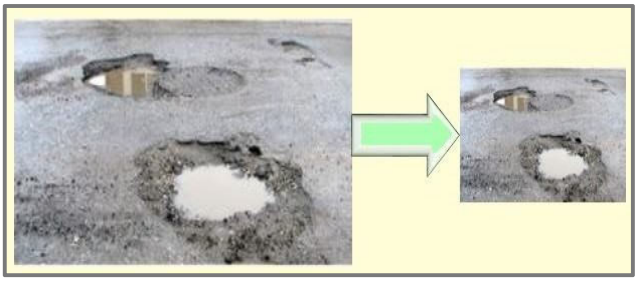


FIGURE 7. Resizing of image.

FIGURE 8. Some randomly selected dataset images.

C. NEURAL NETWORK MODEL AND MODIFICATION

In many cases, the training set is divided into two subsets - one is used for actual training, and the other is employed to monitor how the training is progressing. This subset is referred to as the validation set [50]. The dataset provides a predefined train/ validation and test split. The performed

experiment on a dataset has a training, validation, and test set divided into 70, 15, and 15%, respectively. After the data collection, the experiment is performed using MATLAB 2018b as software. The Device name used is DESKTOP-G5S8E03, with an Intel(R) Core (T.M.) i7-8565U processor. The 11 pro window edition is used, having built an operating system of 22000.434 with a version of 21H2. The installed CPU is 1.80GHz and 1.99 GHz with a GPU of Intel UHD Graphics 620 (300 - 1150 MHz). Random access memory of 8.00 Gigabytes GB having experience pack of 1000.22000.434.0. The system is implemented for AVX, AVX2, and hyper-threading with virtualization. The default ResNet18 NN architecture is shown in Table 2.

ResNet18 training necessitates many computations, which may take more training time, but accuracy is excellent. The manipulation is done for the sake of performed research, and the last layer is designed for 8-crack class classification. In ResNet18 NN, the total trainable parameters are 11,511,784 [51]. Table 2 shows the padding, stride, input, and output layers. Residual block is one of ResNet18's distinctive characteristics in the implemented research. To learn the residual between the input and output instead of the direct mapping, the implemented network learned the "identity shortcut link," which introduces a shortcut between two layers. By incorporating skip connections with identity mapping, the residual block solves the problem of disappearing gradients while training intense models. There is a set weight of 1.0 for these skip connections. Because of the skip connections, the model can become quite deep while still trainable. The residual block tries to fit a different mapping, as shown in Equation 5.

$$H(x) = F(x) + x \tag{5}$$

Concerning identity, (x) is a residual mapping. Learning the weights as 0 is simple if the best mapping is identity. Otherwise, finding tiny fluctuations besides the identity mapping is easier if the ideal mapping is close to the identity. The proposed method utilizes a projection connection instead of an identity connection when the output and input dimensions differ. The function G(x) transforms input x into output F(x) dimensions [52]. There is a difference between the regular and residual blocks of the ResNet18 algorithm, shown below in Figure 9.

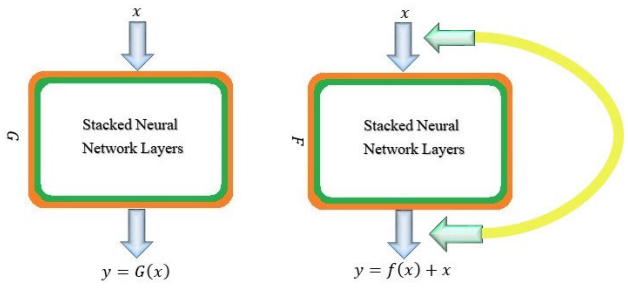


FIGURE 9. Comparison between CNN stable block and residual block of Resnet18.

A residual block involves learning the residual (x) between the output and the input. The residual module determines whether or not a connection is allowed to skip its present connection.

D. PROPOSED INNOVATIVE MODEL SMR18-NN

SMR18-NN is built so that faster RCNN is used to share the computation of DNN between the region proposal network and fast RCNN. One of the key contributions is the enlargement of this algorithm of multiple classes, comprising non-crack and crack classes. The built dataset is capable of covering an assortment of on-field environments. Using a variety of cameras and vision equipment, this technique intends to make a routine inspection of pavement structures easier while also speeding up the accurate assessment of the broad crack distribution. Secondly, convolutional maps are shared between regional proposed networks. This is done to make Faster RCNN faster. The proposed new algorithm is an automated vision-based crack detection method using deep learning.

The pavement road crack detection algorithm using Faster RCNN is divided into two main networks, as shown in Figure 10. Fast RCNN and RPN execute crack detection using or sharing the same deep neural network architecture, and here, the deep neural network used is ResNet18. For modification in ResNet18, the last two layers are modified: max pooling and fully connected. Modifications are done in the CONV and FC layers, while the Softmax layer is replaced with Softmax and regression. For ResNet18 fast RCNN modification, the last max pooling layer is modified by the RoI (Region of interest) pooling layer. The modification is shown below in Table 3.

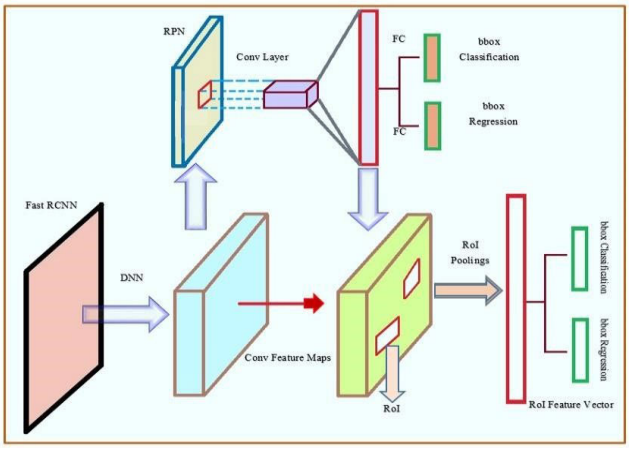


FIGURE 10. The schematic architecture of Faster R-CNN (RPN upper and FRCNN lower portion).

A regional proposal network (RPN) is a fully convolutional network that generates a set of object proposals in the form of rectangular bounding boxes from input images. The literature demonstrates that the RPN is a critical component of the Faster-RCNN model. The RPN comprises two networks

TABLE 2. ResNet18 architecture.

Input			Output			Layer	Stride	Pad	Kernel	Input	Output		Parameters
224	224	3	112	112	64	Conv1	2	1	7	7	3	64	9472
112	112	64	56	56	64	Maxpool 1	2	0.5	3	3	64	64	0
56	56	64	56	56	64	Conv2-1	1	1	3	3	64	64	36928
56	56	64	56	56	64	Conv2-2	1	1	3	3	64	64	36928
56	56	64	56	56	64	Conv2-3	1	1	3	3	64	64	36928
56	56	64	56	56	64	Conv2-4	1	1	3	3	64	128	36928
56	56	64	28	28	128	Conv3-1	2	0.5	3	3	64	128	73856
28	28	128	28	28	128	Conv3-2	1	1	3	3	128	128	147584
28	28	128	28	28	128	Conv3-3	1	1	3	3	128	128	147584
28	28	128	28	28	128	Conv3-4	1	1	3	3	128	128	147584
28	28	128	14	14	256	Conv4-1	2	0.5	3	3	128	256	295168
14	14	256	14	14	256	Conv4-2	1	1	3	3	256	256	590080
14	14	256	14	14	256	Conv4-3	1	1	3	3	256	256	590080
14	14	256	14	14	256	Conv4-4	1	1	3	3	256	256	590080
14	14	256	7	7	512	Conv5-1	2	0.5	3	3	256	512	1180160
7	7	512	7	7	512	Conv5-2	1	1	3	3	512	512	2359808
7	7	512	7	7	512	Conv5-3	1	1	3	3	512	512	2359808
7	7	512	1	7	512	Conv5-4	1	1	3	3	512	512	2359808
1	7	512	1	1	512	Average pool	7	0	7	7	512	512	0
Total												11,511,784	

TABLE 3. RPN layers specification.

Layer	Name	Filter size	Depth	Stride
1	Conv+ReLU	7×7	96	2
2	LRN	--	--	--
3	Max Pooling	3×3	96	2
4	Conv+ReLU	5×5	256	2
5	LRN	--	--	--
6	Max Pooling	3×3	256	2
7	Conv+ReLU	3×3	384	1
8	Conv+ReLU	3×3	384	1
9	Conv+ReLU	3×3	256	1
10	Sliding Conv+ReLU	3×3	256	1
11	FC	--	256	--
12	Softmax & Regressor	--	--	--

that are connected by two parallel convolution layers. Given the feature map extracted by the feature extraction network, the RPN produces a series of highly accurate regions. After receiving the feature map, the RPN initially generates a 1024-dimensional feature map using a 3×3 convolution. Then, it generates anchors, and two parallel branches are used for classification and boundary regression. Unlike traditional

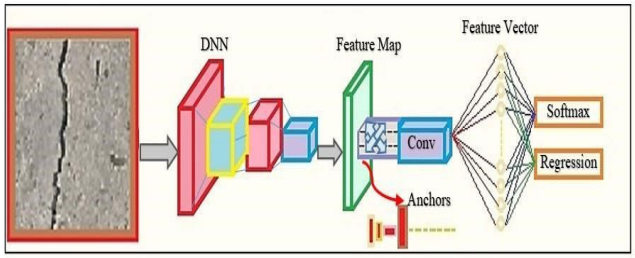


FIGURE 11. The schematic architecture of the region proposal network.

object detection models, the RPN utilizes the anchor mechanism to generate k anchors at the convolutional feature map. The generated anchors are then forwarded to the subsequent two parallel convolution layers. Branch 1, one of the convolution layers, generates 2xkdimensional vectors matching the k anchors' scores classified as targets or backgrounds. The other convolution layer, named branch 2, produces a 4xk-dimensional vector corresponding to the positional transformation parameters of k anchors linked to the actual object bounding box. In this study, branch 2 is utilized for target and background classification. Figure 11 illustrates the RPN's entire schematic architecture.

In the above Figure, there are anchors after the feature map layer. The anchors often number nine and can be found in other aspect ratios, including 1:1, 2:1, and 1:2. The three advised anchors have the following sizes: 128 × 128, 256 × 256, and 512 × 512. Fast RCNN is given the obtained region proposal for additional fine-tuning to enhance the classification process. The obtained region proposal is described by predicting each box's rectangular bounding boxes and probabilities. Here, Table 4 explains Fast RCNN layers in detail.

TABLE 4. Fast RCNN layers specification.

Layer	Name	Filter size	Depth	Stride
1	Conv+ReLU	7×7	96	2
2	LRN	--	--	--
3	Max Pooling	3×3	96	2
4	Conv+ReLU	5×5	256	2
5	LRN	--	--	--
6	Max Pooling	3×3	256	2
7	Conv+ReLU	3×3	384	1
8	Conv+ReLU	3×3	384	1
9	Conv+ReLU	3×3	256	1
10	RoI Pooling	--	256	--
11	FC+ReLU	--	4096	--
12	Dropout	--	--	--
13	FC+ReLU	--	4096	--
14	Dropout	--	--	--
15	FC+ReLU	--	6	--
16	Softmax & Regressor	--	--	--

Like RPN, the Fast RCNN uses DNN to extract feature maps from the object proposal provided as an input image after receiving the object proposal from RPN. The acquired feature maps are overlaid with the RPN region proposal. The Region of Interest (RoI) is presented in the object proposal. When using RoI pooling, RoIs are provided, and the max-pooling procedure is used to create a feature vector for each RoI with a fixed size. The FC layers get the resulting vectors. The softmax layer follows the FC layers, and the regression softmax layer finds the probability of image cracks. In contrast, the regression layer shows the location and size of the anchors, shown in Figure 12.

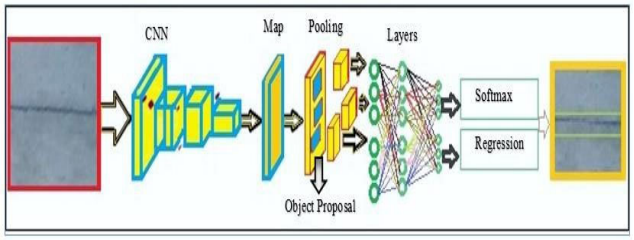


FIGURE 12. The schematic architecture of Fast R-CNN.

Once all these parameters and modifications are done, the architecture extracts features. Support Vector Machine (SVM) is used as a classifier. This classifier is preferred to improve actual positive value. Figure 13 below shows the internal processing structure of SMR18 NN.

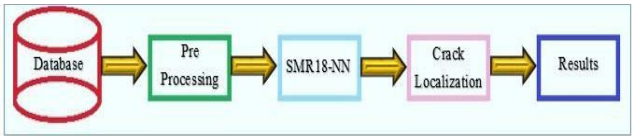


FIGURE 13. The internal processing structure of the algorithm.

The modification in SMR18-NN is done so that faster RCNN is attached with DNN, and Faster RCNN is a combination of two N.N., Fast RCNN and RPN. In SMR18-NN, a new modification is made to attach the support vector machine to the algorithm for good classification results. During the implementation of SMR18-NN, localization is done to emphasize the result at the end, and after gaining the result, it is compared with the ResNet18 result. Ultimately, the decision is made using the results' statistics. The feature maps are acquired by utilizing DNN in RPN. After getting the feature maps, a CONV layer followed by the ReLU activation function is slid on each pixel to get feature maps. Each sliding window feature is planned into a vector and given to a softmax layer and regressions, which predict bounding box locations and the related probabilities.

V. IMPLEMENTATION AND RESULT ANALYSIS

A. PRE-PROCESSING

As already mentioned, deep neural network training is a challenging undertaking. To get the best performance out of the network, thorough training must be done to adjust the weights and biases of each neuron. Region Classification based on Adaptive thresholding has been used for better classification. Equation 6 below shows the formula used for classification.

$$l(a,b) = f(a,b) + I(a-1,b) + I(a,b-1) - I(a-1,b-1)\,H(x) = F(x) + x \tag{6}$$

where $f(a,b)$ is the thresholding pixel of each classified part of the image starting from (a_1, b_1) to the end, shown by (a_{n-1}, b_{n-1}). The Equation 7 is the transform equation of Equation 6.

$$f_s(a,b) = \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} f_s(a,b) = I(a_2,b_2)\} - I(a_2,b_2-1) - I(a_2-1,b_2) + I(a_2-1,b_2-1)\,H(x) = F(x) + x \tag{7}$$

where the square patch size with the 2×2 the threshold value of classification of $T(a,b)$ is below:

$$T(a,b) = \begin{cases} 1 & T(a,b) > \dfrac{f_s(a,b)}{2\times 2} \times \left(1 - \dfrac{t}{100}\right) \\ 0 & otherwise \end{cases} \tag{8}$$

In the above equation 8, $T(a, b)$ is the classification binarized value with the thresholding sensitive value T. The formula is implemented on each pixel (a, b) of the image patch. The pixel value of the image is maximum in the center of the image and lower at the edges of the image. The transform value is represented by $\hat{F}(x, y)$ it is shown in the equation 9.

$$\hat{F}(x, y) = \frac{1}{MN} \sum_{a=0}^{M-1} \sum_{b=0}^{N-1} f(a, b) e^{-j2\pi\left(\frac{xx}{M} + \frac{yy}{N}\right)} \quad (9)$$

In the equation above $f(a, b)$ is the exact pixel value between two pixels of the image, $e^{-j2\pi\left(\frac{xx}{M} + \frac{yy}{N}\right)}$ is the corresponding agent of each pixel value. The complex output of the transformation is produced without losing accuracy. The function of this transformation will make the proposed algorithm fast and robust in terms of time and accuracy, respectively. There is also a conditional probability between two pixels of the image as if they have a similarity up to a non-ignorable level. So, the algorithm is designed in a way that the thresholding point will automatically categorize them in the crack or non-crack part of the image. There is a special Equation 10, for finding the similarity between two-pixel points of the image with conditional probabilities.

$$Q_{j|i} = \frac{e^{-\|a_j - a_i\|^2 / 2\sigma_i^2}}{\sum_{k \neq i} e^{-\|a_i - a_k\|^2 / 2\sigma_i^2}} \quad (10)$$

where σ_i is the variance of the Gaussian centered on the data point a_i. The a_i and a_j are y_i and y_j are the matching parts between points, respectively and it is shown in equation 10. The pairwise similarity is shown by $F_{j|i}$ of image pixel a_i and a_j can be found as in Equation 11.

$$F_{j|i} = \frac{e^{-\|b_j - b_i\|^2}}{\sum_{k \neq i} e^{-\|b_i - b_k\|^2}}; F_{i|i} = 0. \quad (11)$$

The probability of a maximum match of the pixel depends on the output of $Q_{j|i}$ and $F_{j|i}$. These outputs are directly dependent on the variables a_i and a_j with a segmented part of another image part that is b_i and b_j. $F_{j|i}$ can be expended using a series, as shown in the Equation 12.

$$q_{ij} = \frac{\left(1 + \|y_j - y_i\|^2\right)^{-1}}{\left(\sum_{k \neq i} 1 + \|y_j - y_i\|^2\right)^{-1}}. \quad (12)$$

The previous equation has the advantage that $\left(1 + \|y_j - y_i\|^2\right)^{-1}$ approaches an inverse square law for large pairwise distances, $\|y_j - y_i\|^2$. This ensures that the joint probabilities are almost invariant to changes in map scale. A natural measure of faithfulness with $P_{j|i}$ and $q_{j|i}$, is Kullback-Leibler divergence. t-SNE minimizes the sum of the Kullback-Leibler divergence over all data points by using gradient descent. Assuming symmetricity, the cost function is given as in Equation 13:

$$\frac{\partial C}{\partial y_i} = 4 \sum \sum_j \left(P_{ij} - q_{ij}\right) \left(1 + ||y_i - y_j|||2\right)^{-1} \left(y_i - y_j\right) \quad (13)$$

Mapping features in the low-dimensional space using t-SNE enables visualization of the metadata feature space during knowledge transfer across the layers of the deep neural network. t-SNE is analogous to Eigen map analysis to find the relevant clusters, which essentially gives the Eigen gap between successive clusters. The main reason for using ResNet18 and SMR18-NN is that they have distinct architectures, resulting in different training processes and outcomes. To achieve the objective, both networks are evaluated, and the accuracy obtained determines which network is more suitable for the dataset in terms of training, validation, and testing. ResNet18 and SMR18-NN differ in various aspects, which will influence their outputs. Here is the training analysis of ResNet18 and SMR18-NN.

B. RESNET18 MODEL IMPLEMENTATION

During the experiment involving ResNet18, a single GPU was employed as the available resource. The experiment conducted for six epochs resulted in a 90.64% accuracy in training. The learning rate (LR) was set to 0.001, the maximum iterations were 1242, and there were 207 iterations per epoch.

Figure 14 shows the ResNet18 training and that when the epoch is updated, the accuracy increases, and the loss function decreases. This indicates that the model is working correctly, and in the final epoch, the maximum accuracy is gained with a minimum loss. The whole experiment is performed under a constant learning rate.

C. SPLICED MULTIMODEL RESIDUAL18 NEURAL NETWORK OPTIMIZATION

As per the experiment performed and the result is a concern, all the data distribution and various parameters, such as training, validation, testing, leaning rate, epochs, and the number of iterations, are kept the same as the ResNet18. Because the newly proposed method is ultimately compared to ResNet18, it sought to maintain all the parameters at the same level. The dropout layer is also kept the same as before, which is 0.5, preventing the overfitting problem. To create object proposals for the Fast R-CNN, the RPN is trained using pre-trained DNN-initialized weights as a first stage. In the second stage of training, the Fast R-CNN is initialized using the pre-trained weights acquired in the object proposal generation stage. In step 3, object proposals are again generated after RPN has been trained using the weights collected in step 2. The created object proposal is given to the Fast R-CNN in step 4, which trains it using the initial parameters discovered in phase 3. Figure 15 shows the training processing of the network at the end of this research article.

The proposed crack detection algorithm, SMR18-NN, uses a fully connected layer to generate output. This output is

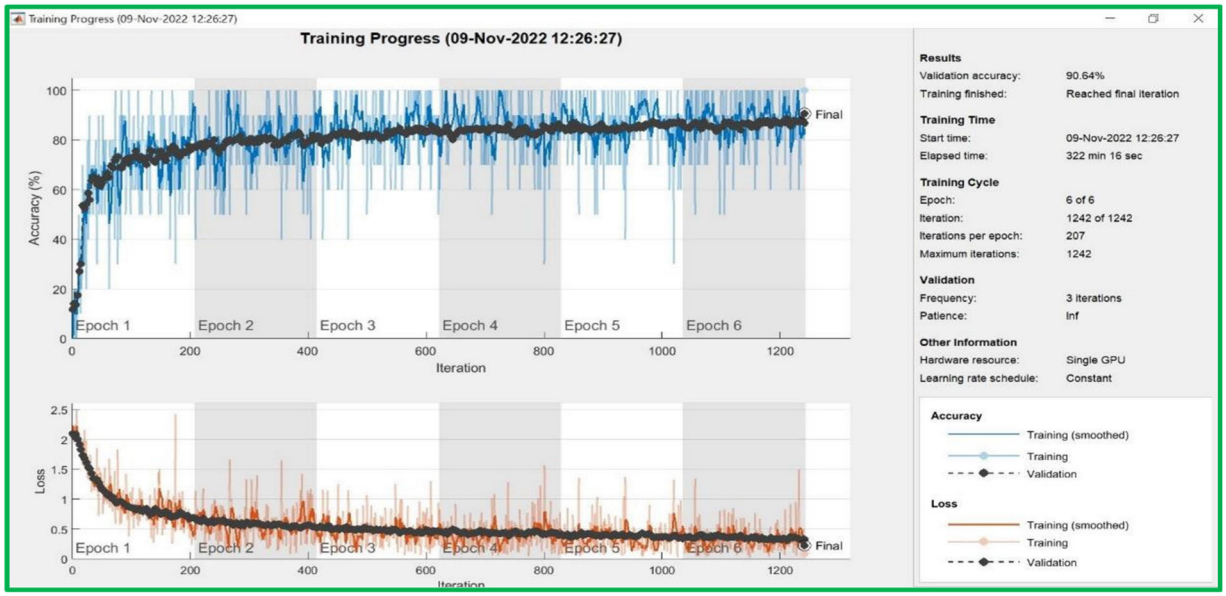


FIGURE 14. ResNet18 training.

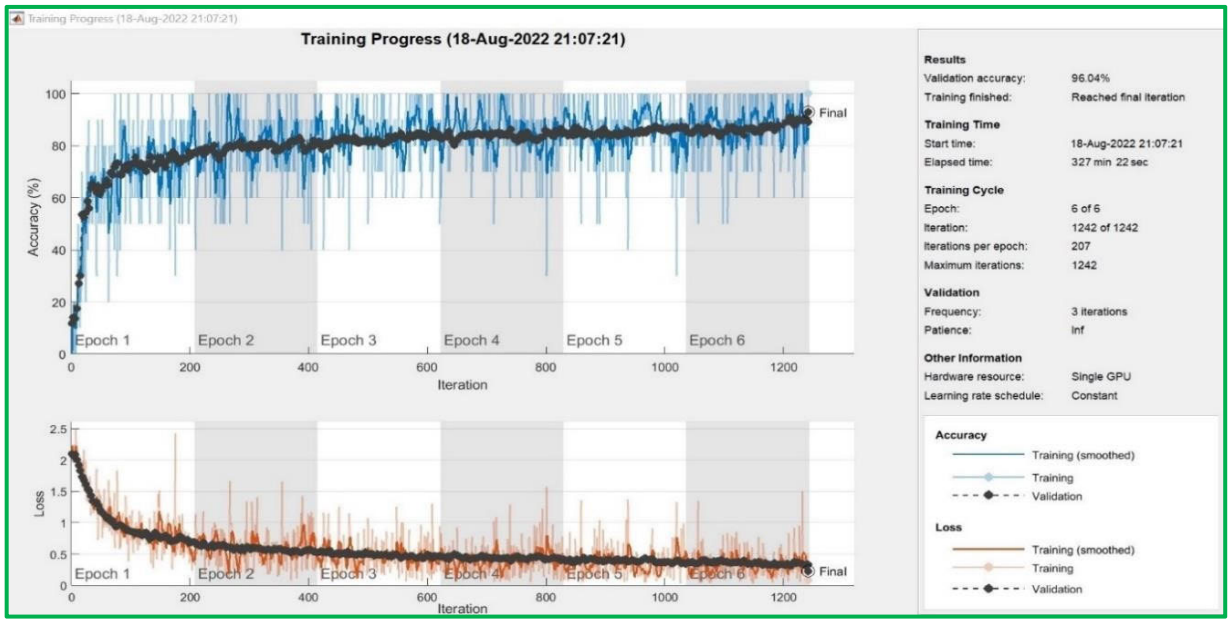


FIGURE 15. Accuracy/ loss graph of SMR18 neural network.

then used as the input feature vectors for an SVM classifier, which is applied during the final stage of the algorithm. The main objective of SVM is to identify a hyperplane that distinguishes most of a labeled dataset for classification. This study used an RBF kernel for SVM, and a cross-validation technique was utilized to obtain optimal results. Generally, DNN is used to extract features from paved images, but SVM is used as an alternative classifier to a softmax layer to enhance the classification ability. Typically, DNNs are used to extract features from paved images, but in this algorithm, SVM is used instead of a softmax layer to improve classification accuracy.

VI. RESULT ANALYSIS

A. RESNET18 EXPERIMENT RESULT ANALYSIS

The target and output classes are shown wit in Figure 16. The Figure is a confusion matrix elaborating every class output as target and output classes.

The best outcome in the above-shown confusion matrix is the pothole class because of the 100% accuracy rate and

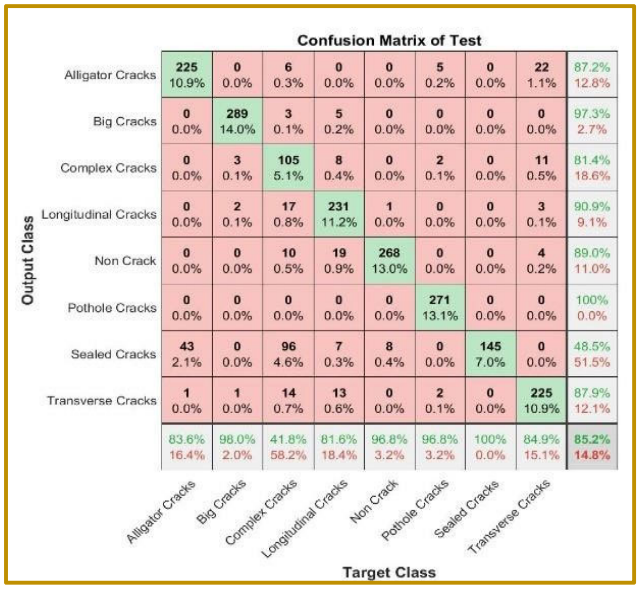


FIGURE 16. Confusion matrix of the Resnet18.

the correct classification of all the images. In Figure 16 of the ResNet18 confusion matrix, the true positive values are 235 for alligators, 289 for big, 105 for complex, 231 for longitudinal, 268 for crackles, 271 for a pothole, 145 for sealed, and 225 for transverse class cracks. The results are satisfactory for the processed dataset since the true positive detected images for wide crack images, which is 289, are also impressive. The confusion matrix shows that ResNet18 performs excellently in every class, resulting in 85.20% overall accuracy.

B. SMR18 COMPARISON AND ANALYSIS WITH RESNET18 NEURAL NETWORKS

After getting the test result of SMR18 NN, which is 92.00%, this result is compared with ResNet18 NN taken from the above experiment. It is crystal clear that all the output results of the SMR18-NN are far better than ResNet18. Figure 17 is more visible.

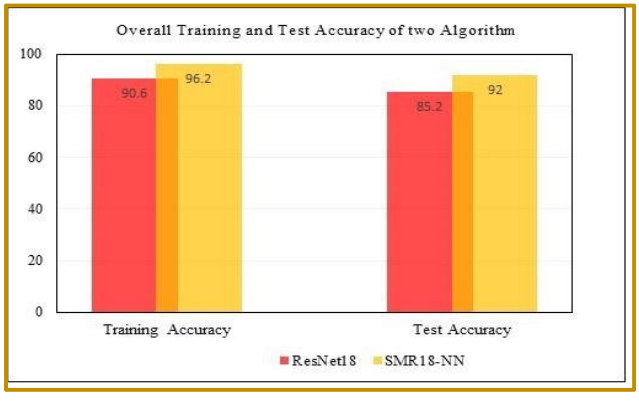


FIGURE 17. Comparative results of ResNet18 and SMR18-NN.

Figure 17 shows comparative results of ResNet18 and SMR18-NN in which training accuracy is shown on the left side while the right side presents test accuracy results. After finding the values, a comparison is made at the end. The value for ResNe18 is 90.60% and 85.20% in the training and test class. At the same time, SMR18-NN is 96.20% and 92.00% for training and test class, respectively.

VII. EXTENDED RESULTS IN TERMS OF OTHER MEASURES

When analyzing the results of a task, several performance metrics can be employed. Different methods exist to evaluate a model, but in engineering, there are three scales based on which it can find the performance score of any network. These are precision, recall, and F1 Score. So, to find the accuracy level of ResNet18 and SMR18-NN, the overall average accuracy is calculated for precision, recall, and F1 score. The last figure, Figure 17, shows the accuracy comparison; it doesn't give information about each class, so it will be more accurate to compare the results of each class for better analysis.

A. PRECISION

Precision considers all retrieved documents and can be evaluated at a specific cut-off rank, considering only the system's top results. Equation 14 for finding the precision value.

$$Precision = \frac{T_p}{T_p + T_F} \tag{14}$$

B. RECALL

Recall is a crucial metric that determines the number of positives the model accurately identifies as true positives.

$$Recall = \frac{T_p}{T_p + F_N} \tag{15}$$

C. F1-SCORE

The F1-Score measures the accuracy of a model on a dataset. It assesses binary classification algorithms categorizing examples as 'positive' or 'negative.' The F-score combines the model's precision and recall. Interestingly, the F1-Score can be adjusted to prioritize either precision or recall. F1-Score, which is a function of precision and recall, the formula is as follows:

$$F1 - Score = \left(\frac{Precision \times Recall}{Precision + Recall}\right) \times 2 \tag{16}$$

The recall is the ratio between correct classified class samples and the class's total number of instances. Table 5 shows the training and test average ResNet18 and SMR18 NN scores. The result-wise performance of SMR18-NN is almost 6-7% better than ResNet18.

In engineering, this is a considerable achievement. It is an eye-opening result for any researcher.

In Figure 18 above, six classes are compared, in which three left sidebars present training accuracy of precision,

TABLE 5. Results comparison of average accuracy.

Method	Training Average Score			Test Average Score		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
ResNet18	85.43	89.40	87.36	84.80	85.38	85.08
SMR18-NN	92.25	91.68	91.96	87.90	90.26	89.06

recall, and F1 score, and on the right side, the same sequence order for the test class. From left to right, the precision value for ResNet18 is 85.43, and SMR18-NN accuracy is 92.25, indicating very high progress in accuracy. The recall value for ResNet18 is 89.40, and SMR18-NN is 91.68. Meanwhile, the average F1 Score is 87.36 and 91.96 for ResNet18 and SMR18-NN, respectively. Test average precision indicates 84.80 for ResNet18 and 87.90 for SMR18-NN. The Figure shows much improvement comparatively because for ResNet18, the outcome is 85.38, but for SMR18-NN, the value is 90.26. The figure's right side presents 85.40 for ResNet18 and 89.06 for SMR18-NN. If the figures are deeply observed, all results are improved significantly irrespective of training or test class.

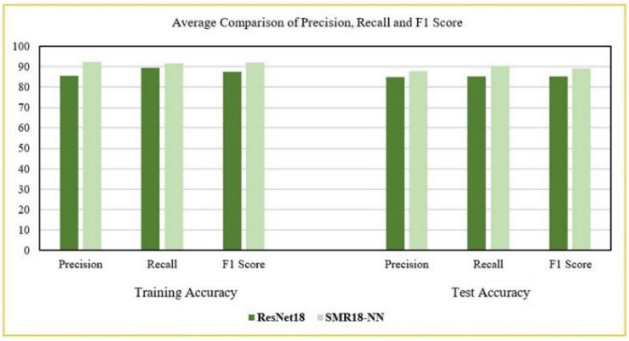


FIGURE 18. Precision, recall, and F1-Score comparison.

In Table 6, each class precision comparison is shown. From alligator to transverse class, each is showing significant improvement in SMR18-NN is visible. The alligator class accuracy for training ResNet18 is 98.10, but the accuracy in SMR18NN is 99.20, while for test precision, the values are 83.60 and 85.10 in ResNet18 and SMR18-NN, respectively. For the big (Wide), non-crack, and pothole classes, the accuracy results are outstanding that is 98.90, 98.20, and 100 for training ResNet18, but in the table, the result of training SMR18-NN it is much better than ResNet18 for these classes. These class values are 99.50, 99.10, and 100 for SMR18-NN with the same sequence of big, non-crack, and pothole classes. For test ResNet18, precision values for the same classes are 98.00, 96.80, and 96.80, while SMR18-NN got 98.90, 98.00, and 98.50 for big, respectively, non-crack and pothole classes, respectively. The complex class result is worse than all, for training ResNet18, its value is 52.30, and for SMR18-NN, it is 58.80, while for the same class, the test ResNet18 is 41.80 and SMR18-NN is 51.20. The training ResNet18 values are 90.60, 93.10, and 91.20 for longitudinal, sealed, and transverse classes, while for respective class SMR18-NN, the values are 92.60, 95.80, and 93.00, respectively. The test precision score is impressive for ResNet18 for the described classes, which is 81.60, 92.40, and 84.90 for longitudinal, sealed, and transverse crack classes, respectively, while for SMR18-NN, the values are 84.40, 94.20, and 87.10 for the above mentioned three classes respectively. Figure 19 shows the recall numbers for ResNet18 and SMR18-NN at the end of this research article.

TABLE 6. Results comparison of average accuracy.

Class	ResNet18 (Training)	SMR18-NN (Training)	ResNet18 (Test)	SMR18-NN (Test)
Alligator	98.10	99.20	83.60	85.10
Big (Wide)	98.90	99.50	98.00	98.90
Non-crack	98.20	99.10	96.80	98.00
Pothole	100.00	100.00	96.80	98.50
Complex	52.30	58.80	41.80	51.20
Longitudinal	90.60	92.60	81.60	84.40
Sealed	93.10	95.80	92.40	94.20
Transverse	91.20	93.00	84.90	87.10

For recall, the worst class is the sealed class because the values for training ResNet18 and SMR18-NN are 57.90 and 61.40, respectively. For test ResNet18, it is 48.50 and 56.30 for SMR18-NN. Wide or big, longitudinal, and pothole are outclass for recall because all the result values are up to 90. For ResNet18 training accuracy for big, longitudinal, and pothole crack classes, the values are 98.30, 93.90, and 100, while for SMR18-NN, the values are 99.20, 95.40, and 100.0, respectively. 92.90 is the value of training ResNet18 in the alligator class, and the SMR18-NN value is 94.50 for the same class. It is a little weird for the same class in the test section because the test alligator score for ResNet18 is 87.20, which is very bad compared to SMR18-NN for the same class, which is 91.50, comparatively defendable. For complex, non-crack, and transverse classes, the training ResNet18 value is 79.80, 98.20, and 94.20. In comparing these scores, training SMR18-NN is almost the same except for complex class, which is 87.70, 99.40, and 94.20 for respective non-crack and transverse classes. In the test section, the values for these complex, non-crack, and transverse classes are 81.40, 89.90, and 87.90 for ResNet18. For the same class, the values are 97.50, 91.9, and 92.30 for SMR18-NN. The most accurate and unbeatable result for all the algorithms, irrespective of ResNet18 or SMR18-NN, is the pothole class, whose result is 100 in all the classes. All the descriptive statistics are shown in Figure 19, and Figure 20 is the figure for the F1 Score comparison at the end of this research article.

Figure 20 shows the F1 Score result. It is evident that the pothole is once again leading the result because the score for training classes for ResNet18 and SMR18-NN is 100, which is unbeatable, but for a test class, the ResNet18 value is 98.37, and SMR18-NN is 99.24. The output for the non-crack and big crack classes is comparatively sound because for training

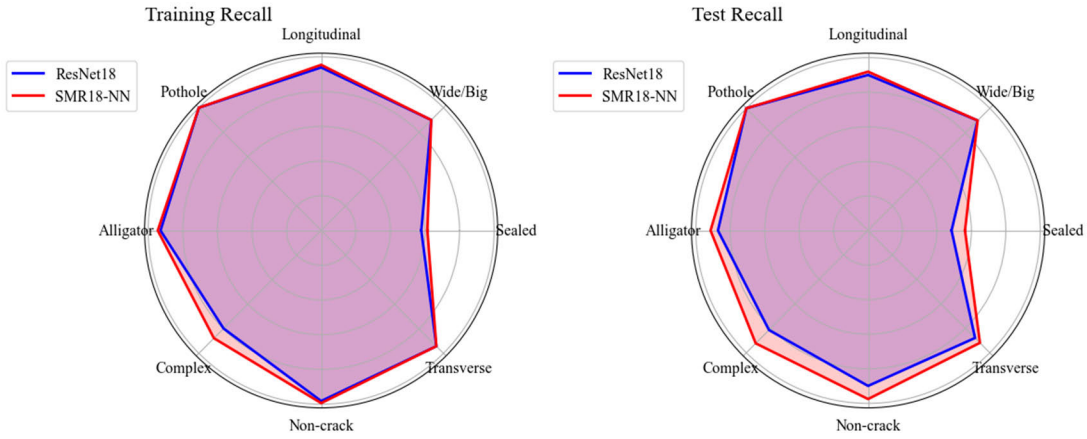


FIGURE 19. Recall comparison.

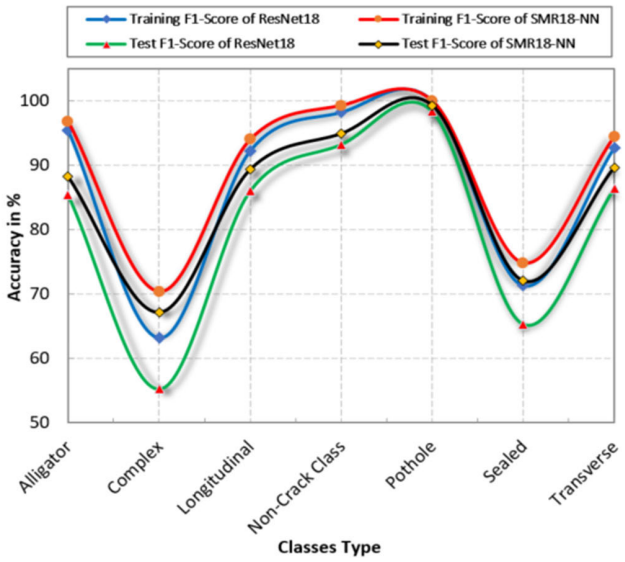


FIGURE 20. F1 Score comparison.

ResNet18, its value is 98.20 and 98.59, while 99.24 and 99.34 for SMR18-NN. For the same classes in the test class, its values are 93.22 and 97.64 for ResNet18 but 94.85 and 98.34 for the SMR18-NN. Training ResNet18 values are 95.42, 63.18, 92.22, 71.30, 92.67, and for SMR18-NN, 96.79, 70.39, 93.97, 74.83, and 94.42 for the respective classes of alligator, complex, longitudinal, sealed and transverse. If this classwise order is followed, the values for the test F1 score of the two algorithms are 85.36, 55.23, 85.99, 65.31, and 86.37 for ResNet18, while the SMR18-NN results are 88.18, 67.14, 89.29, 72.04 and 89.62. From the above discussion, it is clear and a significant achievement that the pothole crack class result is almost ideal and unbeatable. The gain result for this specific class is almost 100% accurate.

D. PAVEMENT CRACKS LOCALIZATION AND ANALYSIS

The crack localization implementation combines the train neural network algorithms with sliding window techniques. The sliding window size in the performed experiment is 1×1, as shown in Figure 21. The purpose of such a small size is to increase accuracy significantly by decreasing false positive detection.

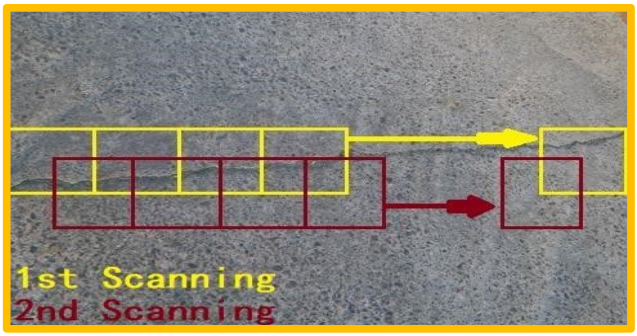


FIGURE 21. Sliding window technique.

Some input images were new to the algorithm during crack localization, but some augmented training images were given as test images. The result can be seen in Figure 22, where the regions are identified correctly and are indicated by the blue boxes after implementing crack localization. One of the other benefits of crack localization is that it can easily find the size and depth of the pavement crack. Though it can find the size of the crack mathematically, it will always be a rough estimation, which might be near the actual crack size. The result shows that the algorithm performance is excellent because new data is fed, and that data has never been used before for training. Some concrete structure images were also given to the system to test whether the proposed algorithm can detect cracks in concrete structures. The image is divided into a grid in the localized cracks figures, and the proposed crack detection algorithm classifies each grid. If the grid is classified as cracked, the highlighted grid is blue and yellow. As shown in the images, most cracked regions are accurately detected. However, there are still instances of incorrect categorization

According to the results of the studies, the SMR18-NN approach offers the potential for automatic pavement crack detection. In this work, SMR18-NN is implemented

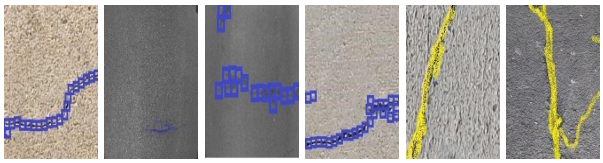


FIGURE 22. Crack localization results.

to automatically detect and localize the cracks in images of paved roads. The algorithm offers better resistance to noise and does not require handcrafted features as it automatically extracts features [53]. Once a deep neural network is trained, it can accurately detect crack regions in images, which is more efficient and cheaper than manual inspection. Creating a quality dataset can present difficulties as it depends on human assessment as the ultimate standard and requires verification from multiple sources. The proposed model has to be realistic, yet the computation cost has to be reasonable, which is fulfilled by our proposed SMR18-NN.

VIII. CONCLUSION, LIMITATION, AND FUTURE WORK

The current research uses modern image processing and genetic algorithms to identify and categorize pavement distress cost-effectively, precisely, and efficiently. It was achieved using advanced Photo processing and genetic algorithms. The studied method automates pothole, longitudinal, and sealed crack evaluations well. It was highly reliable and precise in the samples analyzed. The system is excellent at finding pothole cracks, which can cause road deterioration, accidents, and traffic congestion. This inquiry relies on the algorithm's independence to eliminate manual feature extraction. This quality reduces errors and human involvement expenses. Support Vector Machines in SMR18-NN and ResNet18's testing performance, which yielded an accuracy rate of 85.20%, demonstrate the method's robustness.

The study's shortcomings include its reliance on tiny Photo collections and its inability to obtain interior fracture data. Broader applications, including video processing and medical imaging, have yet to be fully explored. Although no model can attain 100% accuracy, using more datasets, iterations, and hyperparameters can improve precision. It bodes well for academic research in the future. Further research should examine other datasets and models to determine the findings' generalizability and analyze classifiers that may increase algorithm accuracy. This helps researchers decide if the findings are generalizable. Using advanced deep learning models like VGG19, Google Net, ResNet50, and Inception v3 to compare SMR18-NN outputs appears promising. This opportunity is good. The algorithm's architecture may improve pavement crack identification and classification, including max-pooling, Leaky ReLU, clipped ReLU, and BiLSTM, and transfer learning with pre-trained models. This work advances pavement distress evaluation and prepares future studies for more complex, accurate, and comprehensive methods. One ongoing challenge is enhancing accuracy and flexibility when dealing with fractures with similar visual characteristics. The task is challenging since even human inspectors struggle with it. Continued research into innovative designs, more datasets, and different applications will speed up the development system of pavement crack detection and improve its effectiveness.

ABBREVIATION

AMOST	Augmented Minority Over-sampling Technique.
NN	Neural network.
ANN	Artificial Neural Networks.
DNNs	Deep neural networks.
CA	Classification accuracy.
GANs	Generative Adversarial Networks.
RGB	Red, Green, and Blue.
SVM	Support Vector Machine.
RoI	Region of interest.
RPN	Region Proposal Network.
US	United States.

REFERENCES

[1] J. Mun, Y. Jang, Y. Nam, and J. Kim, "Edge-enhancing bi-histogram equalisation using guided image filter," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 688–700, Jan. 2019.

[2] K. Levik, "How to sell the message 'road maintenance is necessary' to decision makers," in *Proc. 1st Road Transp. Technol. Transf. Conf. Africa*, 2001, pp. 460–467.

[3] N.-D. Hoang, "Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding," *Adv. Civil Eng.*, vol. 2018, no. 1, Jan. 2018, Art. no. 3924120.

[4] A. Ullah, Z. Sun, H. Elahi, F. Afzal, A. Khatoon, N. Sayed, and I. Ahmad, "Experimental and numerical research of paved microcrack using histogram equalization for detection and segmentation," *Math. Problems Eng.*, vol. 2022, pp. 1–13, Jun. 2022.

[5] A. Ullah, H. Xie, M. O. Farooq, and Z. Sun, "Pedestrian detection in infrared images using fast RCNN," *Int. J. Neural Netw. Adv. Appl.*, vol. 6, pp. 46–52, Nov. 2019.

[6] M. Y. Seker, A. Ahmetoglu, Y. Nagai, M. Asada, E. Oztop, and E. Ugur, "Imitation and mirror systems in robots through deep modality blending networks," *Neural Netw.*, vol. 146, pp. 22–35, Feb. 2022.

[7] A. Ullah and L. Xuan, "Voice recognition for English language pattern recognition approach," *Amer. Sci. Res. J. Eng., Technol., Sci.*, vol. 17, no. 1, pp. 95–104, 2016.

[8] Y. Zhou, "Natural language processing with improved deep learning neural networks," *Sci. Program.*, vol. 2022, pp. 1–8, Jan. 2022.

[9] T. Mazhar, D. B. Talpur, T. A. Shloul, Y. Y. Ghadi, I. Haq, I. Ullah, K. Ouahada, and H. Hamam, "Analysis of IoT security challenges and its solutions using artificial intelligence," *Brain Sci.*, vol. 13, no. 4, p. 683, Apr. 2023.

[10] O. Che Puan, M. Mustaffar, and T.-C. Ling, "Automated pavement imaging program (APIP) for pavement cracks classification and quantification," *Malaysian J. Civil Eng.*, vol. 19, no. 1, pp. 1–16, May 2018.

[11] J. Zhu and J. Song, "An intelligent classification model for surface defects on cement concrete bridges," *Appl. Sci.*, vol. 10, no. 3, p. 972, Feb. 2020.

[12] S. Ahmed and A. T. Azar, "Adaptive fractional tracking control of robotic manipulator using fixed-time method," *Complex Intell. Syst.*, vol. 10, no. 1, pp. 369–382, Feb. 2024.

[13] L. Ali, "Damage detection and localization in masonry structure using faster region convolutional networks," *Int. J. Geomate*, vol. 17, no. 59, pp. 98–105, Jul. 2019.

[14] J. Behmann, A.-K. Mahlein, T. Rumpf, C. Römer, and L. Plümer, "A review of advanced machine learning methods for the detection of biotic stress in precision crop protection," *Precis. Agricult.*, vol. 16, no. 3, pp. 239–260, Jun. 2015.

[15] C. Liu, J. Ying, H. Yang, X. Hu, and J. Liu, "Improved human action recognition approach based on two-stream convolutional neural network model," *Vis. Comput.*, vol. 37, no. 6, pp. 1327–1341, Jun. 2021.

[16] A. Ullah, H. Elahi, Z. Sun, A. Khatoon, and I. Ahmad, "Comparative analysis of AlexNet, ResNet18 and SqueezeNet with diverse modification and arduous implementation," *Arabian J. Sci. Eng.*, vol. 47, no. 2, pp. 2397–2417, Feb. 2022.
[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[18] B. Kim and S. Cho, "Automated vision-based detection of cracks on concrete surfaces using a deep learning technique," *Sensors*, vol. 18, no. 10, p. 3452, 2018.
[19] Z. Fan, Y. Wu, J. Lu, and W. Li, "Automatic pavement crack detection based on structured prediction with the convolutional neural network," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 1–9.
[20] D. M. Stewart, L. Tian, and D. L. Nelson, "Mutations that cause the wiskott-aldrich syndrome impair the interaction of wiskott-aldrich syndrome protein (WASP) with WASP interacting protein," *J. Immunol.*, vol. 162, no. 8, pp. 5019–5024, Apr. 1999.
[21] T. Rahman, M. E. H. Chowdhury, and A. Khandakar, "Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using using chest X-ray," *App. Sci.*, vol. 10, no. 9, p. 3233, 2020.
[22] Z. Sun, J. Zhai, L. Pei, W. Li, and K. Zhao, "Automatic pavement crack detection transformer based on convolutional and sequential feature fusion," *Sensors*, vol. 23, no. 7, p. 3772, Apr. 2023.
[23] I. Ahmad, S. J. Xu, A. Khatoon, U. Tariq, I. Khan, S. S. Rizvi, and A. Ullah, "Analytical study of deep learning-based preventive measures of COVID-19 for decision making and aggregation via the RISTECB model," *Sci. Program.*, vol. 2022, pp. 1–17, Jan. 2022.
[24] L. Li, W. Wang, M. Wang, S. Feng, and A. Khatoon, "Lane line detection at nighttime of fractional differential and central line point searching with fragi and Hessian," *Sci. Rep.*, vol. 13, no. 1, pp. 1–17, May 2023.
[25] A. Ullah, Z. Sun, H. Rehman, S. Khan, A. Ashraf, and A. Khatoon, "Experimental and mathematical verification of permanent magnet synchronous motor using modified simulink," in *Proc. IEEE 7th Int. Conf. Smart Energy Grid Eng. (SEGE)*, Aug. 2019, pp. 86–91.
[26] T. Rateke, K. A. Justen, and A. Von Wangenheim, "Road surface classification with images captured from low-cost camera–road traversing knowledge (RTK) dataset," *Revista de Informática Teórica e Aplicada*, vol. 26, no. 3, pp. 50–64, Nov. 2019.
[27] J. Kim, S. Shim, S.-J. Kang, and G.-C. Cho, "Transfer learning-based road damage detection for multiple countries," *Struct. Control Health Monitor.*, vol. 2023, pp. 1–16, Jan. 2023.
[28] P. S. Barve, L. S. Thakur, R. P. Barve, J. K. Shah, and N. P. Patel, "Detection and sizing study of cracks," *Int. J. Advance Res. Eng., Sci. Technol.*, vol. 2, no. 8, pp. 84–91, 2015.
[29] J. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzainet, "Blossom-quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm," *Int. J. Numer. Methods Eng.*, vol. 89, no. 9, pp. 1102–1119, Mar. 2012.
[30] N. T. H. Nguyen, T. H. Le, S. Perry, and T. T. Nguyen, "Pavement crack detection using convolutional neural network," in *Proc. 9th Int. Symp. Inf. Commun. Technol.*, 2018, pp. 251–256.
[31] X. Lin, J. Zhang, D. Wu, E. Meng, M. Liu, M. Li, and F. Gao, "Intelligent identification of pavement cracks based on PSA-net," *Frontiers Environ. Sci.*, vol. 11, pp. 1–7, Feb. 2023.
[32] F. Frappart and G. Ramillien, "Monitoring groundwater storage changes using the gravity recovery and climate experiment (GRACE) satellite mission: A review," *Remote Sens.*, vol. 10, no. 6, p. 829, May 2018.
[33] G. Liu, H. Ge, T. Li, S. Su, and S. Wang, "Robust multi-view subspace enhanced representation based on collaborative constraints and HSIC induction," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105585.
[34] Z. Sun, "The pavement laser 3D data denoising algorithm based on the morphology," *J. Inf. Comput. Sci.*, vol. 11, no. 6, pp. 2049–2057, Apr. 2014.
[35] L. Deng, A. Zhang, J. Guo, and Y. Liu, "An integrated method for road crack segmentation and surface feature quantification under complex backgrounds," *Remote Sens.*, vol. 15, no. 6, p. 1530, Mar. 2023.
[36] A. Falendysh, P. Kharlamov, O. Kletska, and N. Volodarets, "Calculation of the parameters of hybrid shunting locomotive," *Transp. Res. Proc.*, vol. 14, pp. 665–671, Jan. 2016.
[37] G. Yu, J. Dong, Y. Wang, and X. Zhou, "RUC-net: A residual-UNet-based convolutional neural network for pixel-level pavement crack segmentation," *Sensors*, vol. 23, no. 1, p. 53, Dec. 2022.
[38] Z. Chen, X. Hao, and Z. Sun, "A modified Bayes shrinkage in shearlet domain combined with p-Laplace anisotropic diffusion for image denoising," *ICIC Exp. Lett.*, vol. 7, no. 10, pp. 2691–2696, 2013.
[39] A. Urbonas, V. Raudonis, R. Maskeliunas, and R. Damasevicius, "Automated identification of wood veneer surface defects using faster region-based convolutional neural network with data augmentation and transfer learning," *Appl. Sci.*, vol. 9, no. 22, p. 4898, Nov. 2019.
[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
[41] Y. Jiang, H. Wei, S. Wang, and S. Tian, "Research on real-time road crack detection algorithm based on improved knowledge distillation," in *Proc. IEEE 11th Data Driven Control Learn. Syst. Conf. (DDCLS)*, Aug. 2022, pp. 1154–1159.
[42] S. C. Magalhães, F. N. dos Santos, P. Machado, A. P. Moreira, and J. Dias, "Benchmarking edge computing devices for grape bunches and trunks detection using accelerated object detection single shot multibox deep learning models," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105604.
[43] J. Shijie, W. Ping, J. Peiyi, and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," in *Proc. Chin. Autom. Congr.*, 2017, pp. 4165–4170.
[44] V. Nath, C. Chattopadhyay, and K. A. Desai, "On enhancing prediction abilities of vision-based metallic surface defect classification through adversarial training," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105553.
[45] J. Yun, D. Kim, D. M. Kim, T. Song, and J. Woo, "GAN-based sensor data augmentation: Application for counting moving people and detecting directions using PIR sensors," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105508.
[46] K. Cheng, R. Tahir, L. K. Eric, and M. Li, "An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset," *Multimedia Tools Appl.*, vol. 79, nos. 19–20, pp. 13725–13752, May 2020.
[47] *Bulk Resize Photos*. Accessed: May 15, 2024. [Online]. Available: https://bulkresizephotos.com/en
[48] A. Ullah, S. Zhaoyun, U. Tariq, M. I. Uddin, A. Khatoon, and S. S. Rizvi, "Gray-level image transformation of paved road cracks with metaphorical and computational analysis," *Math. Problems Eng.*, vol. 2022, pp. 1–14, May 2022.
[49] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 675–678.
[50] X. Niu, Z. Yang, N. Zhou, and C. Li, "A novel method for cage whirl motion capture of high-precision bearing inspired by U-Net," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105552.
[51] P. Navarro, C. Cintas, M. Lucena, J. M. Fuertes, C. Delrieux, and M. Molinos, "Learning feature representation of Iberian ceramics with automatic classification models," *J. Cultural Heritage*, vol. 48, pp. 65–73, Mar. 2021.
[52] G. Jean, F. Banon, B. Junior, and U. de M. Braga-Neto, "Mathematical morphology and its applications to signal and image processing," in *Proc. 8th Int. Symp. Math. Morphol.*, 2017.
[53] W. Li, J. Huyan, S. L. Tighe, Q.-Q. Ren, and Z.-Y. Sun, "Three-dimensional pavement crack detection algorithm based on two-dimensional empirical mode decomposition," *J. Transp. Eng., B, Pavements*, vol. 143, no. 2, pp. 1–12, Jun. 2017.

ASAD ULLAH received the Ph.D. degree in information engineering from Chang'an University.

He has established a strong research foundation during the Ph.D. degree. He is an enthusiastic and committed Associate Professor with the School of Information Engineering, Xi'an Eurasia University. Over the years, he has gained extensive experience in teaching and conducting research, fueling his passion for academia and fostering a positive learning environment. Additionally, he has mentored and supervised undergraduate and graduate students, providing guidance and support in their research projects, and facilitating their professional growth.

ZHAOYUN SUN is Professor with the School of Information and Communication Engineering, Chang'an University. Her research interests include image processing, artificial intelligence, and automation.

SYED AGHA HASSNAIN MOHSAN received the Ph.D. degree in marine information science and engineering from Zhejiang University, China. He is a Joint Postdoctoral Fellow with EIT-USTC, China. With a comulative impact factor of more than 300, he has published more than 95 articles in OSA, Elsevier, Springer Nature, IEEE Transactions, SPIE, and several other journals/conferences. His research interests include optical wireless communication, wireless power transfer, NOMA, IoUT, UWSN, UAVs, IRS, and 5G/6G technology. Currently, he is serving as a Topical Advisory Panel Member for Drones and JMSE. He has served as a peer reviewer for IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *IEEE Communications Magazine*, *IEEE Network Magazine*, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, IEEE TRANSACTIONS ON BIG DATA, and several other SCIE/SSCI journals. He has served as the workshop chair, the session organizer, a TPC member, and delivered invited talks in several international conferences.

AMNA KHATOON is currently pursuing the Ph.D. degree with the College of Information Engineering, Chang'an University, Xi'an, China. She is working on remote sensing, feature extraction, and fuzzy integration. Her research interests include image processing and analysis, feature extraction, remote sensing, and algorithm analysis.

ADIL KHAN received the B.S. degree in telecommunication from Iqra National University, Peshawar, the M.S. degree in electronics and communication engineering, and the Ph.D. degree in electronic science and technology from Beijing University of Posts and Telecommunications, Beijing, China. He is currently an Associate Professor with the Information and Communication Engineering Department, Xi'an Eurasia University. His research interests include wireless communication, UAV-enabled wireless networks, and vehicular ad-hoc networks.

SHABEER AHMAD received the B.S. degree in telecommunication from Iqra National University, Pakistan, and the M.S. degree in electronics and communication engineering from Beijing University of Posts and Telecommunications, Beijing, China, where he is currently pursuing the Ph.D. degree in electronic science and technology. His research interests include the application of machine learning in wireless communication, resource allocation in UAV-enabled wireless networks, software-defined networking, cloud computing, and vehicular ad-hoc networks.