IEEE - 56998

Surface Crack Detection Using Deep Learning Models

Satwika Prathapani
School of CSE
VIT-AP University
Amaravati, India
prathapanisatwika1525@gmail.com

Poornimai Mekala
School of CSE
VIT-AP University
Amaravati, India
mekala.poornimai09@gmail.com

Ch. Venkata Rami Reddy
School of CSE
VIT-AP University
Amaravati, India
chvrr58@gmail.com

Suneetha Muvva
Independent Researcher
Amaravati, India
sunee.muvva@gmail.com

2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 979-8-3503-3509-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCNT56998.2023.10307851

***Abstract* — Surface cracks are a widespread issue that affects many different businesses, including the building industry, the transportation industry, and the manufacturing industry. In order to preserve safety, minimize expensive repairs, and ensure the lifespan of buildings and equipment, it is essential to detect and identify surface cracks. The research made use of a dataset of surface cracks that is freely accessible to the public from Kaggle website. A level of accuracy of 99.62% was attained by the VGG16 model. The accuracy of the Inception model was measured at 99.89%. The Xception model was successful in achieving an accuracy of 99.76%. According to these findings, the Inception model performed far better than the other two models when it came to identifying surface cracks.**

Keywords: Surface crack detection, Convolutional neural networks, Deep learning, Image segmentation, Object detection, Feature extraction, Image classification, Computer vision

I. INTRODUCTION

Surface cracks are a widespread issue that affects many different businesses, including the building industry, the transportation industry, and the manufacturing industry. In order to preserve safety, minimize expensive repairs, and ensure the lifespan of buildings and equipment, it is essential to detect and identify surface cracks. Surface cracks may be induced by a number of different things, including tension, variations in temperature, chemical reactions, and even natural disasters. Manual examination by qualified persons is required in traditional approaches to surface crack detection. This technique may be difficult, subjective, and prone to mistakes. Because of advances in deep learning, researchers have developed a number of ways to automatically detect surface cracks using computer vision. Deep learning algorithms have lately gained popularity in a variety of areas, including Pattern Recognition [1–8], Medical Imaging, Driver Drowsiness Detection [9–10], Video Analysis, Spam Detection [11], Healthcare, Clustering [12], and many more. Deep learning, a subset of machine learning, trains neural networks to recognize data patterns. This training may make use of a large number of data points. Deep learning has been found to be highly effective in a variety of computer vision applications, including object detection, image classification, and image segmentation. Deep learning models are well-suited for identifying surface cracks because they can autonomously extract characteristics from the images they are given as input and can learn from enormous amounts of data. The goal of this work is to compare the performance of three cutting-edge deep learning models in detecting surface cracks: VGG16, Inception, and Xception. We used transfer learning approaches to fine-tune the pre-trained models on a publicly accessible dataset of surface cracks. This dataset is comprised of pictures taken by various imaging modalities, such as optical.

II. LITERATURE REVIEW

The identification of surface cracks is a difficult problem that has attracted a substantial amount of interest from the scholarly community. Manual examination, which is often used in traditional techniques of surface crack identification, may be difficult, subjective, and prone to errors. Deep learning models have been shown to be extremely successful at detecting surface cracks. Deep learning algorithms for surface crack recognition have included convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders in recent years. CNNs are particularly useful for identifying surface cracks because they can autonomously extract features from the images they are given as input and capture the spatial information of the cracks.

Liu et al. [13] developed a CNN-based technique for locating surface cracks in concrete structures. Their CNN architecture consisted of two completely connected layers, two convolutional layers, and two max-pooling layers, which they had designed themselves. When it came to identifying surface cracks on concrete photographs, the suggested approach was able to obtain an accuracy of 92.1%.
Deep learning was presented as a technique for locating surface cracks in asphalt pavements by Tian et al. [14]. They started with a VGG16 model that had already been trained and then fine-tuned it using photographs of asphalt pavement. When it came to identifying surface cracks in asphalt photographs, the suggested approach was able to obtain an accuracy of 92.8%. Ma et al. [15] suggested a technique that makes use of deep learning for the purpose of locating surface cracks in metal plates. They started with a ResNet-50 model that had already been trained and then fine-tuned it using a dataset of photos of metal plates. When it came to identifying surface fractures on metal photos, the suggested approach was able to reach an accuracy of 95.2%.

These works show the promise of deep learning models in surface crack detection tasks. Despite this, there are still several obstacles to overcome in order to design surface crack detection systems that are successful. One of the most significant obstacles is the scarcity of datasets that are both varied and extensive, which are required for both the training and the evaluation of the models. Another obstacle is the difficulty of recording and recognizing tiny cracks or breaks in complicated patterns. This is a challenge since it makes it more difficult to find cracks. It is vital to construct models that can be generalized across many imaging techniques since different imaging methods may have varying capacities for identifying particular kinds of cracks. As a result of this, it is important to develop models.

III. Proposed model

The model in this study is based on transfer learning methodologies. Among these techniques the fine-tuning of pre-trained deep learning models, such as VGG16, Inception, and Xception, using a large dataset of surface cracks captured by multiple imaging modalities. CNN architectures that have been pre-trained on huge datasets like ImageNet and can be customized for particular purposes include VGG, ResNet, Inception, and MobileNet. Each of these designs is notable in its own right.

A. Inception

The Inception-v3 neural network is a convolutional one, and it has a total depth of 48 layers. The enormous collection of more than one million photos in the ImageNet database can be used to train a network that you can import. The trained network can categorize photographed things into a thousand different groups, including "keyboard," "mouse," "pencil," and "many animals." As a direct result, the network now has the ability to collect intricate feature representations for many sorts of images. 299 by 299 pixels is the utmost dimension of an image that can be submitted to the network. Deep learning officially began in 2014 with the development of the Inception architecture by Google researchers. It was designed to improve the efficiency of convolutional neural networks (CNNs) by extracting features from input images using multiple parallel convolutional layers with differing filter sizes. This action was taken to enhance CNN. The design of Inception is predicated on the concept of developing a network that is capable of both convolution and pooling operations at a variety of different sizes. In order to do this, the design employs many parallel convolutional layers with varying filter sizes. This provides the network with the capability to collect features operating at a variety of scales. After that, these parallel convolutional layers are joined together, and the features that are produced by that step are sent into a pooling layer. The Inception architecture not only makes use of parallel convolutional layers, but it also makes use of 1x1 convolutional layers to lower the dimensionality of the input features before sending them on to the bigger convolutional layers. As a result, the number of network parameters is decreased, and the network's computational efficiency increases.

B. VGG16

As a straightforward but effective convolutional neural network, the VGG16 architecture has achieved outstanding results in a number of image categorization tasks.
1. Input: A picture with dimensions of 224 × 224 × 3 is read in as the input to the network.
2.Convolutional Layers: The first thirteen layers of the network are convolutional layers, with each layer having a 3x3 filter and a stride of 1. The number of filters increases as we move deeper into the network, starting with 64 filters in the first layer and ending at 512 filters in the final layer.
3.Max Pooling Layers: The network receives a max pooling layer with a 2x2 filter and stride value after every two convolutional layers. This contributes to a reduction in the output's spatial dimension, which in turn increases the computational efficiency.
4.Fully Connected Layers: There are a total of three entirely connected layers following the thirteen convolutional layers. 4,096 neurons are present in each of the first two fully connected layers, whereas only 1,000 neurons are present in the third fully connected layer. The number of classes in the ImageNet dataset correlates with the number of neurons in this population.
5.SoftMax Layer: Using the output of the last fully connected layer, a SoftMax activation function is used to generate a probability distribution over the classes. This task is performed by the neural network's fifth and final layer. The VGG16 design is generally straightforward, but its potency should not be underestimated. It is well-suited for a variety of computer vision tasks due to its deep architecture and small filter size, which enable it to capture both low-level and high-level information in images. Therefore, it can be utilized in a variety of computer vision applications.

C. Xception

The Xception architecture is a powerful convolutional neural network that achieves cutting-edge performance on a variety of benchmark datasets while being more computationally efficient than conventional CNNs.
1. Input: A picture with dimensions of 299 × 299 × 3 is read in as the input to the network.
2.Entry Flow: The entry flow comprises of the initial few layers of the network and is made up of a number of convolutional layers, batch normalization layers, and activation functions. This portion of the network is located at the network's inception.
3.Middle Flow: The middle flow contains multiple repeated, depth-separated convolution segments. Each block in this flow consists of a pointwise convolution layer, a depth-wise convolution layer, a batch normalization layer, a ReLU activation function, and an additional batch normalization layer.
4.Exit Flow: The exit flow consists of numerous convolutional layers, batch normalization layers, and activation functions. These layers collaborate to reduce the spatial scale of the output and provide the final classification results.
5.SoftMax Layer: Using the output of the last convolutional layer, a SoftMax activation function generates a probability distribution across classes. This is the fifth and final layer of the neural network.

The Xception architecture is a robust neural network that produces cutting-edge performance. It is appropriate for a variety of computer vision applications because it makes use of depth-wise separable convolutions to collect both low-level and high-level picture information. It may thus be used to address a variety of computer vision problems.

IV. METHODOLOGY

The methodology for surface crack detection using VGG16, Inception, and Xception architectures typically involves the following steps:

1. Dataset Preparation: The first thing that has to be done is to either develop or compile a dataset of photos that show surface cracks. It is important that the dataset be sufficiently vast so that it can capture a wide range of crack kinds, sizes, and orientations. It is important that the photos be labelled with the locations of the cracks as well as their degrees of severity. Dataset is taken from Kaggle website.
2. Data Preprocessing: The next step is to do some preliminary processing on the pictures in order to prepare them for use in the CNN model training process. In order to do this, the photos may need to be resized to a constant size, the pixel values may need to be normalized to a common scale, and data augmentation methods such as rotation, translation, and flipping may need to be utilized to increase the training data's diversity.
3. Model Selection: The next thing that has to be done is select the most suitable CNN architecture for the surface fracture detection assignment. VGG16, Inception, and Xception are all well-known and efficient architectures that have been pre-trained on huge picture datasets. These designs may be fine-tuned to perform very well on a specific task.
4. Model Training: After the selection of the CNN architecture, the subsequent step is to train the model using the preprocessed dataset. This is accomplished by providing the CNN with the pictures to analyze as input and modifying the model parameters in order to get the lowest possible prediction error. The training procedure could involve several epochs or iterations; to avoid overfitting, the model's performance should be evaluated using a validation set.
5. Model Evaluation: Once the model has been trained, the next step is to evaluate its performance on a test set of photos that were not utilized during the training process The common measures used for evaluation are accuracy, precision, recall, and F1 scores. These metrics reflect the capacity of the model to accurately detect the existence of surface cracks as well as the severity of the cracks.
6. Model Optimization: If the performance of the model is not sufficient, the next step is to optimize the model. This may be done by altering the hyperparameters such as the learning rate, batch size, and regularization strength, or it can be done by fine-tuning the pre-trained weights on a related dataset.
7. Deployment: Once the model has been optimized and verified, it will be ready to be used in a surface crack detection application that will be used in the real world. For this purpose, the model may need to be integrated with a user interface or an API, or it may need to be deployed on a cloud server so that it can be accessed remotely. The performance of the model should be continually checked, and updates should be applied whenever new data is made available.

4.1 Dataset

A. Surface crack image dataset

Fig.1. illustrates some of the dataset's sample photos. This dataset is taken from the Kaggle website. Images of a variety of concrete surfaces, both with and without cracks, are included in the databases. For the purposes of image classification, the image data are separated into two folders, one labelled "negative" (indicating that there is no crack) and the other "positive," which indicates that there is a crack. There are a total of 40.000 photos that are 227 x 227 pixels in size and have RGB channels, and each class contains 20,000 images.

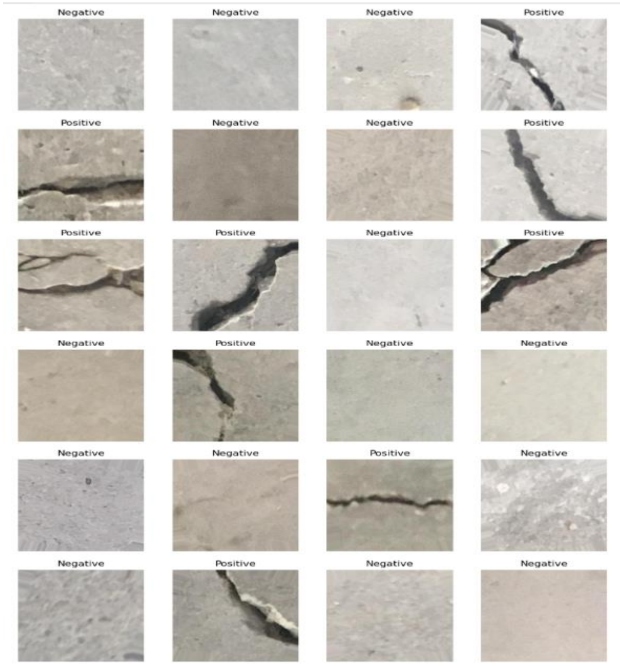

Fig. 1. Illustrations of crack and non-crack images

V. RESULTS AND DISCUSSIONS

In this study, the suggested model capabilities were assessed using Accuracy, Recall, F1-score, and the confusion matrix.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - \text{score} = 2 * \frac{precision * recall}{precision + recall}$$

TABLE 1. ACCURACIES OF PRE-TRAINED MODELS (%)

Model	Precision	Recall	F1-score	Accuracy
Inception	99.62	99.89	99.75	99.89
VGG-16	99.94	99.44	99.68	99.62
Xception	99.77	99.97	99.86	99.76

Table 1. depicts the proposed Inception method and the other pretrained CNN models' total performance analysis is shown in the confusion matrix, showing that Inception performs better than the other algorithms in comparison. Table 1 shows the Accuracies of pre-trained models.

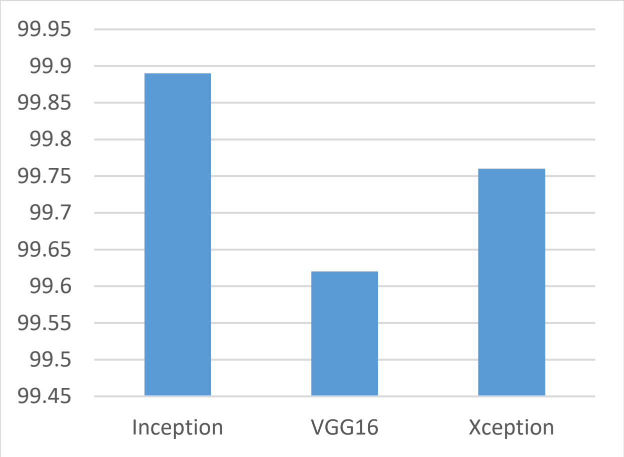

Fig 2. Performance comparison.

Fig. 2. shows the performance comparison of pre-trained models.

VI. CONCLUSION

Some well-known deep learning architectures for surface crack detection are compared and contrasted in this study. These are the VGG16, Inception, and Xception architectures. The research entailed training and testing the models on a dataset of surface pictures that had cracks and photos that did not have cracks. The findings demonstrated that all three architectural designs were successful in identifying surface cracks, with Inception attaining the greatest overall performance metrics among the three. However, the research also showed that the most effective architecture and hyperparameters differ depending on the particular application and dataset, and that transfer learning has the potential to dramatically increase the model's overall performance. This study demonstrates the potential of deep learning approaches for surface crack detection, which has the potential to have substantial ramifications for the infrastructure's maintenance and safety. The study also sheds light on the benefits and drawbacks of various deep learning architectures and points the way towards potential directions for further investigation, such as the investigation of more architectural styles and datasets as well as the incorporation of many types of data sources.

REFERENCES

[1] Reddy C.V.R., Reddy U.S., Kishore K.V.K. (2019). Facial emotion recognition using NLPCA and SVM, Traitement du Signal, 36(1): 13-22. https://doi.org/10.18280/ts.360102

[2] VenkataRamiReddy, C., Kishore, K.K., Bhattacharyya, D., Kim, T.H. (2014). Multi-feature fusion-based facial expression classification using DLBP and DCT. International Journal of Software Engineering and Its Applications, 8(9): 55-68. https://doi.org/10.14257/ijseia.2014.8.9.05

[3] Ramireddy, C.V., Kishore, K.K. (2013). Facial expression classification using Kernel-based PCA with fused DCT and GWT features. In 2013 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1-6. https://doi.org/10.1016/10.1109/ICCIC.2013.6724211

[4] Reddy, C.V.R., Kishore, K.K., Reddy, U.S., Suneetha, M. (2016). Person identification system using feature-level fusion of multi-biometrics. In 2016 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1-6. https://doi.org/10.1109/ICCIC.2016.7919672

[5] Meena, K., Veni, N.N.K., Deepapriya, B.S. et al. A novel method for prediction of skin disease through supervised classification techniques. Soft Comput 26, 10527–10533 (2022). https://doi.org/10.1007/s00500-022-07435-8

[6] Palakodati, S.S.S., Chirra, V.R., Dasari, Y., Bulla, S. (2020). Fresh and rotten fruits classification using CNN and transfer learning. Revue d'Intelligence Artificielle, Vol. 34, No. 5, pp. 617-622. https://doi.org/10.18280/ria.340512

[7] Chirra, V.R.R., Uyyala, S.R. & Kolli, V.K.K. Virtual facial expression recognition using deep CNN with ensemble learning. *J Ambient Intell Human Comput* **12**, 10581–10599 (2021). https://doi.org/10.1007/s12652-020-02866-3

[8] Banothu, B., Murthy, T.S., Reddy, C.V.R., Yakobu, D. (2020). High-order total bounded variation approach for gaussian noise and blur removal. International Journal of Advanced Science and Technology, 29(3): 10152-10161.

[9] Chirra, V.R.R., Uyyala, S.R., Kolli, V.K.K. (2019). Deep CNN: A machine learning approach for driver drowsiness detection based on eye state. Revue d'Intelligence Artificielle, Vol. 33, No. 6, pp. 461-466. https://doi.org/10.18280/ria.330609

[10] B. V. Kumar, K. K. Srinivas, P. Anudeep, N. S. Yadav, G. V. Kumar and P. A. Harsha Vardhini, "Artificial Intelligence Based Algorithms for Driver Distraction Detection: A Review," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 383-386, doi: 10.1109/ISPCC53510.2021.9609349.

[11] Chirra, V.R., Maddiboyina, H.D., Dasari, Y., Aluru, R. (2020). Performance evaluation of email spam text classification using deep neural networks. Review of Computer Engineering Studies, Vol. 7, No. 4, pp. 91-95. https://doi.org/10.18280/rces.070403

[12] V. R. Reddy, D. Yakobu, S. S. Prasad and P. A. H. Vardhini, "Clustering Student Learners based on performance using K-Means Algorithm," 2022 International Mobile and Embedded Technology Conference (MECON), 2022, pp. 302-306, doi: 10.1109/MECON53876.2022.9752165.

[13] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In D. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), European Conference on Computer Vision (pp. 21-37). Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2

[14] Han, X., Zhao, Z., Chen, L., Hu, X., Tian, Y., Zhai, C., ... & Huang, X. (2022). Structural damage-causing concrete cracking detection based on a deep-learning method. Construction and Building Materials, 337, 127562. https://doi.org/10.1016/j.conbuildmat.2021.127562

[15] Ma, Y., Li, H., Wang, H., Wang, X., & Gao, Y. (2019). Deep learning for detecting surface cracks on metal plates. Journal of Intelligent Manufacturing, 30(1), 149-160. https://doi.org/10.1007/s10845-017-1345-7