IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A SECURE FRAMEWORK FOR CRACK DETECTION IN UNDERGROUND TUNNEL IMAGES USING MACHINE LEARNING TECHNIQUES

[1]Swathi Ramesh, [2]Nikitha Maria James, [3]Riya John,[4]Shaheez K S,[5]Divya James

[1,2,3,4,5]Department of Information Technology,

[1,2,3,4,5]Rajagiri School of Engineering and Technology, Ernakulam, Kerala, India

Abstract: Crack has an important role within the field of evaluating the standard of concrete structures, which affects the applicability, and sturdiness of the structure. Cracks on the surface of concrete structures like tunnels indicate the strength and durability of that structure. Hence it is required that these cracks be found as early as possible so that maintenance can be done. Thus, there is a need for new and efficient crack detection methods. In secure tunnel crack detection system using machine learning, tunnel surface images are obtained and are then encrypted. These images are then pre-processed and classified into cracks and no cracks classes using transfer learning technology of Inception v3 neural network model. The model achieved an excellent accuracy of 0.998, precision of 0.989, recall of 0.989 and fscore of 0.989.

Index Terms **- Crack, Inception V3, Secure Tunnel Crack, Homomorphic Encryption, Transfer.**

I. INTRODUCTION

It is required that the cracks in concrete surfaces be found as early as possible so that maintenance can be done. It is difficult to identify tunnel cracks because they are tiny, and many filters may have to be applied to remove the noise within the tunnel images. Nowadays, tunnel construction technology is becoming very complex [1]. Traditional monitoring methods are based on manual crack detection, but the method is subjective and operators face difficulties like exposure to dust conditions, insufficient light. It is also a cumbersome procedure to detect the cracks from a huge set of images manually [2]. So, developing an automatic inspection system for identifying cracks will save time and effort. Our main aim is to create a framework for detecting the cracks in the underground tunnel using machine learning techniques and to create a secure framework for the system.

II. BACKGROUND STUDY

A. Homomorphic Encryption:

Homomorphic encryption is a cryptographic method that permits mathematical operations on data to be administered on cipher text, instead of performing these operations on the particular data itself [8]. Once we encrypt the input file, we get the cipher text. Various operations like addition and multiplication can be performed on the cipher text and are then decrypted to get the plain text. The output obtained from decrypting this cipher text will be same as the output we obtain after performing the mathematical operations on the original data. This is one of the major advantages of using homomorphic encryption. Many systems choose to perform computations using the services provided by large companies like Amazon and Microsoft because they are cheap. But these systems are concerned about the security of their data so they encrypt the data and then sent it to these companies for computation and storage. Privacy is very important in today's world. For this purpose, various system makes use of homomorphic encryption.

A homomorphic encryption is an asymmetric encryption system where a public key is used to encrypt the data and private key is used to decrypt the data and vice versa. But the basic difference between homomorphic encryption and other encryption systems [7] is that it allows to perform computation on encrypted data. This means that we can perform computations over the encrypted data and get the same result as what we would get after we perform the computation on the original data. Homomorphic encryption involves using arithmetic functions like addition and multiplication, instead of Boolean functions like other methods of secure computation.

B. Transfer Learning:

Transfer learning is a machine learning approach in which a pre-trained neural network model is used on a similar problem. In this approach, the weights learned from performing a task is used for another related task. Hence generalization is achieved. In machine learning, the pre-trained neural network model is used for performing the related task. The weights that the model gained from learning the base task is used for the other task. Transfer learning is used in many machine learning applications. For example, in computer vision, the base layers of neural networks usually try to detect edges, more complicated patterns like shapes in the middle layers and some intrinsic features in the latter layers. In transfer learning, the weights of the first and middle layers are not updated and are kept as such by freezing those layers. Only the last layers are trained on our dataset. It helps to make use of the weights it learned on the previous task. We attempt to transfer the maximum amount knowledge as possible from the previous task the model was trained on to the new task we have.

Training a new neural network from scratch is a difficult procedure. The main benefit of transfer learning is that a machine learning model is created and trained with less training data because the model is already pre-trained. The model already knows the general features that are common to the task at hand and does not need a lot of training. not only decreases the training time, but also improves the performance of the neural network. Hence transfer learning helps to discover a good combination of features within a short time frame. There are many pre-trained models which can be used for transfer learning and other machine leaning tasks. Inception v3 is one such model.

III. PROPOSED METHODOLOGY

Different modules in our system include user interface, a homomorphic encryption, and Inception V3. The user has access to login into our model to input the image for classification. The image is the encrypted using homomorphic encryption for the securing the data. The image is then decrypted and is given to the inception V3 model where it is trained and classified into images with cracks or images with no cracks.

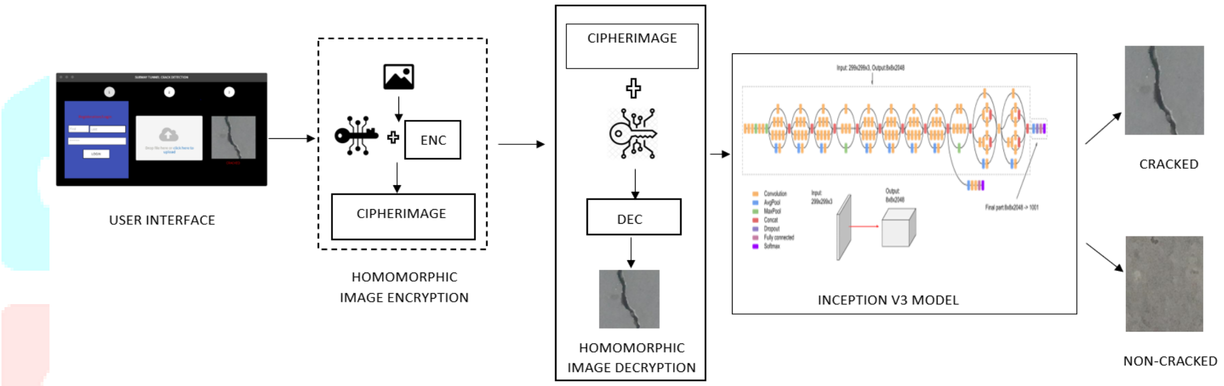

Figure 1: Proposed Model

A. User Interface:

The first module in our system is user interface, in order to check whether an image contain crack or not the user should log in to their account and if the user does not have one, he/she should create one. Then by using the login credentials the user can access the application.

B. Homomorphic Encryption:

The project aims at deploying our code on two nodes, one acting as client and the other one as severs machine. The client is responsible for generating keys, encrypting the image and sending the (key, image) pair to the server. The server in turns perform homomorphic operation.

Step 1: Here first the image is converted into RGB pairs. The image is transformed into 2-D array.

Step 2: Then are using paillers algorithm to generate the public key and the private key.

Step 3: The image is then encrypted on client side before sending it to the model. Using public key, we encrypt the image. The output after encryption will be cipher image. The formula for this is cipher = (a % b) mod (n *n), where a = pow (g, m, n * n) and b = pow (r, n, n * n) here g and r are random numbers, n is the public key and m is the input image.

Step 4: We decrypt the image using the private key (λ, μ), the formula for that is m = (l * μ) % n where l = (pow (cipher, λ, n * n) – 1) // n.

C. Image classification Model:

Input: The input to the neural network model is the raw images of tunnel surface. These images contain tunnel surface with cracks and tunnel surface without cracks.

Image preprocessing: The raw tunnel images are first converted into RGB format in order to make feature extraction easier. The other image preprocessing step is reducing dimension. The default input dimension of Inception v3 is 299-by-299.So the images are rescaled to match the input size of Inception v3.

Transfer learning using Inception v3:

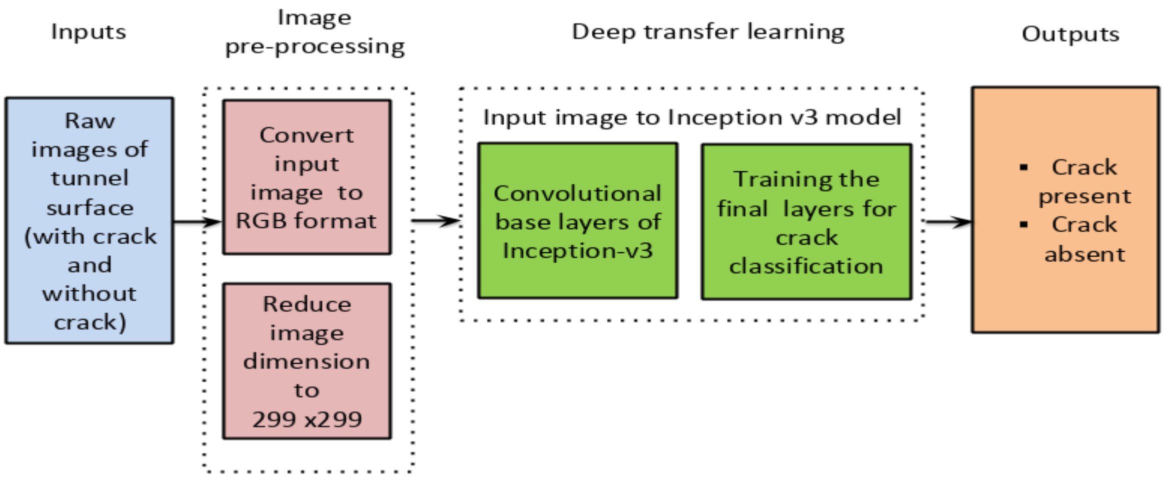

Figure 2: Inception V3

Inception-v3 is a convolutional neural network model with twenty-four million parameters. It achieved an accuracy of 78.8 percent in the ImageNet dataset classification with a very low error rate. It is a pre trained version of the Inception network trained on a large number of images from the ImageNet database. The pretrained network is able to classify images into 1000 classes with excellent accuracy. Hence, the network already knows intrinsic feature representations for a wide variety of images. The network also has a default image input size of 299-by-299.Any image input to the network must be resized to this dimension. The network has two parts; the feature extraction part which consists of the convolutional layers. The other part is the classification part which consists of the fully connected layer and the SoftMax layer [5]. The convolutional layers extract the features that are common to all the images like edges and patterns. The classification part makes use of these features to classify the images.

Inception-v3 uses an "Inception module", a sparse CNN, with 22 layers in a parallel processing workflow, and benefits from several auxiliary classifiers within the intermediate layers. In contrast to conventional CNNs such as Alex Net and VGG, wherein either a convolutional or a pooling operation can be used at each level, the Inception module could benefit from both at each layer.

Furthermore, filters (convolutions) with varying sizes are used at the same layer, providing more detailed information and extracting patterns with different sizes. Importantly, a 1×1 convolutional layer, the so-called bottleneck layer, was employed to decrease both the computational complexity and the number of parameters. To be more precise, 1×1 convolutional layer were used just before a larger kernel convolutional filter (e.g., 3×3 and 5×5 convolutional layers) to decrease the number of parameters to be determined at each level (i.e., the pooling feature process).

In addition, 1×1 convolutional layer make the network deeper and add more non-linearity by using ReLU after each 1×1 convolutional layer. In this network, the fully connected layers are replaced with an average pooling layer. This significantly decreases the number of parameters since the fully connected layers include a large number of parameters. Thus, this network is able to learn deeper representations of features with fewer parameters relative to Alex Net while it is much faster than VGG.

In transfer learning, we reuse the convolutional layer part for feature extraction and train the classification part with our dataset. We don't have to update the weights in the layers of the feature extraction part so the model can be trained with less computational resources and reduce training time. The preprocessed images of tunnel surface are input to the (the base layers) of the Inception v3 model.

Here, the image data is converted into feature vectors. We make the last two layers of the model trainable to train it on the crack dataset. After training the model will be able to classify the images.

Output: After training and validation, we get the target class labels as images with crack presence and images with crack absence.

IV. Result and Discussion

A. Result:

Upon evaluation, the crack detection model achieved high accuracy and f1 score. The increase in accuracy while training was visualized using Tensor board and after 10 epochs of training, an excellent validation accuracy of 0.99 was achieved. Since the dataset is balanced, accuracy can be considered as the evaluation metric for our binary classifier model. At the same time loss decreased to a minimum of 0.01. To rule out the doubt of overfitting in our model, the F1 score of the model was also evaluated. The achieved F1 score was 0.98 which shows a good trade-off between precision and recall. The high accuracy and f1 score can be attributed to the use of a balanced dataset with high quality images, as well as the use of the pre-trained neural network Inception v3.

Figure 3: Epoch Accuracy

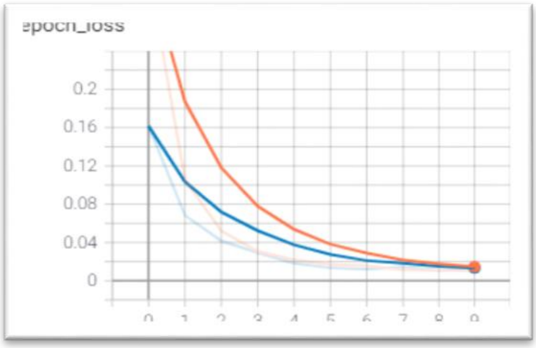

Figure 4: Epoch Loss

B. Future Scope:

The future scope or work of our project is to use our secure crack detection model in real time using IoT (Internet of Things) [4][3]. The model may be integrated to a moving IoT camera which has the ability to capture tunnel surface images as it moves along the tunnel, and then the camera can send the captured images real time to our web application which will encrypt it and save it in cloud. The saved image can be decrypted and input to the crack detection model inside the cloud platform. The crack in the image can thus be found real time for future analysis and for repairs.

V. Conclusion

Cracks impact the structural strength of any concrete structure like buildings or subway tunnels. It is necessary that these cracks be detected and repaired early and quickly so that the structures' strength does not deteriorate. The secure tunnel crack detection model using machine learning that we developed is able to find the cracks on the tunnel concrete surface images and achieve high accuracy for it. For enhanced security, we perform homomorphic encryption over the input image to make the data in transit secure. The presence of crack in the input image is detected using the crack detection model. Transfer learning technology with Inception v3 which is a pre trained neural network was used for this purpose. This network was selected so as to reduce the cost of training a new deep learning model and also to achieve high accuracy.

VI. Reference

[[1] Liu, Z., Suandi, S. A., Ohashi, T., & Ejima, T. (2002). Tunnel crack detection and classification system based on image processing. Machine Vision Applications in Industrial Inspection X. doi:10.1117/12.460191

[2] Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J., Zou, L., & Shen, X. (2020). Image-based concrete crack detection in tunnels using deep fully convolutional networks. Construction and Building Materials, 234, 117367. doi: 10.1016/j.conbuildmat.2019.117367

[3] Doulamis, A., Doulamis, N., Protopapadakis, E., & Voulodimos, A. (2018). Combined Convolutional Neural Networks and Fuzzy Spectral Clustering for Real Time Crack Detection in Tunnels. 2018 25th IEEE International Conference on Image Processing (ICIP). doi:10.1109/icip.2018.8451758

[4] Q. Song et al., "Real-Time Tunnel Crack Analysis System via Deep Learning," in IEEE Access, vol. 7, pp. 64186-64197, 2019, doi: 10.1109/ACCESS.2019.2916330.

[5] Xiaoling Xia, Cui Xu and Bing Nan, "Inception-v3 for flower classification," 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, 2017, pp. 783-787, doi: 10.1109/ICIVC.2017.7984661.

[6] Prateek Prasanna, Kristin Dana, Nenad Gucunski, Basily Basily, "Computer-vision based crack detection and analysis," Proc. SPIE 8345, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2012, 834542 (6 April 2012); https://doi.org/10.1117/12.915384

[7] Kalubandi, V. K. P., Vaddi, H., Ramineni, V., & Loganathan, A. (2016). A novel image encryption algorithm using AES and visual cryptography. 2016 2nd International Conference on Next Generation Computing Technologies (NGCT). doi:10.1109/ngct.2016.7877521

[8] Homomorphic Image Encryption Sachin Rana1, Om Jadhav1 , Shivam Rajput 1, Pranjal Bhansali1, Varshapriya Jyotinagar2 1B.Tech, Dept. of Computer Engineering, Veermata Jijabai Technological Institute, Mumbai, India. 2Faculty of Computer Engineering, Veermata Jijabai Technological Institute, Mumbai, India.

[9] Alnajjar, Fady & Jassmi, Hamad & Gochoo, Munkhjargal & Khan, Wasif & Serhani, Mohamed. (2021). Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. Sensors. 21. 1688. 10.3390/s21051688.

[10]Protopapadakis, E., Voulodimos, A., Doulamis, A., Doulamis, N., & Stathaki, T. (2019). Automatic crack detection for tunnel inspection using deep learning and heuristic image post-processing. Applied Intelligence. doi:10.1007/s10489-018-01396-y