SURFACE CRACK DETECTION OF BUILDINGS AND STRUCTURES USING DEEP LEARNING WITH CONVOLUTIONAL NEURAL NETWORKS AND DECISION TREES

Venkata Ramana Kaneti
Assistant Professor, Department of CSE
VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India – 500090.

Abstract: The growth in population has led to a corresponding increase in construction projects and structures. Consequently, there is a growing need for thorough inspections, especially with regard to concrete surfaces. Surface cracks in concrete are considered a significant defect in civil structures and pose potential risks to human life and society. These cracks can signal a range of issues, from minor damage to more serious structural problems, a lack of durability, or even aesthetic concerns. Therefore, it is crucial to identify and address these cracks. Building inspections are carried out to assess the strength and structural integrity of a building. As previously mentioned, the detection of surface cracks is a pivotal aspect of building inspection, as it plays a key role in determining the overall health of the structure. To achieve this, the proposed method employs Convolutional Neural Networks (CNNs) and Decision Trees. These technologies aid in the identification of surface cracks by analyzing images. A Kaggle dataset is used for this purpose, and data augmentation techniques are applied to improve the accuracy of crack detection. The dataset is appropriately divided into training, testing, and validation sets. The validation process involves testing a collection of images using the model developed. The primary objective of this proposed system is to automate the identification of surface cracks, thereby reducing reliance on human inspectors, decreasing the time required for inspections, and ultimately mitigating the escalating costs in the construction industry.

Keywords: Keywords: building; surface crack detection; convolutional neural networks; image analysis; image segmentation

1. INTRODUCTION

The construction industry has emerged as a vital component of India's economy, contributing to nearly 8% of the country's GDP and providing employment to more than 49 million people. Valued at over $260 billion and experiencing a growth rate of approximately 5.6% annually, the Indian construction sector is on track to become the world's third-largest construction market by 2025. As India's population continues to grow and development projects multiply, there is an unprecedented demand for faster construction and higher-quality buildings. Ensuring the strength and integrity of these structures is a pivotal aspect of this endeavor, which is achieved through rigorous building inspections.

One of the fundamental concerns during these inspections is the detection of surface cracks. These cracks are often indicative of foundational issues, which can be attributed to structural failures, design flaws, or subpar construction practices. Common causes of surface cracking include inadequate vibration during concrete placement, improper concrete cover, concrete expansion and contraction due to temperature variations, and a high water-cement ratio for workability, among others.

To address this critical concern, the proposed system harnesses the power of machine learning algorithms to identify the presence of surface cracks on building exteriors. This innovative system enables us to assess the condition of a building's surface, thereby providing insights into the overall health of the structure. By scrutinizing the quality of the foundational concrete, we can proactively prevent unwanted incidents such as structurally weak buildings, structural failures, or even building collapses.

2. LITERATURE REVIEW

2.1 Ultrasonic Pulse Velocity (UPV)

An ultrasonic pulse velocity (UPV) test is a non-destructive, on-site technique employed to assess the quality of both concrete and natural rock materials. This test involves evaluating the strength and integrity of concrete structures or natural rock formations by measuring the speed of an ultrasonic pulse passing through them. By transmitting an ultrasonic pulse through the concrete or rock under examination and recording the time it takes for the pulse to traverse the material, we can effectively detect the presence of cracks or voids. Higher pulse velocities typically indicate a higher quality and structural integrity of the material, whereas slower velocities may suggest the presence of cracks or voids within the concrete.

The Indian government recommends this test in certain instances to certify and verify the construction of residential buildings. The Ultrasonic Pulse Velocity test serves several valuable purposes, including:

a. Evaluating Material Quality and Homogeneity: It aids in the assessment of the quality and uniformity of concrete materials.
b. Strength Prediction: It can provide valuable insights into the anticipated strength of the concrete.
c. Dynamic Modulus of Elasticity: This test is instrumental in estimating the dynamic modulus of elasticity of concrete, which is crucial for understanding its behavior under various loads.
d. Crack Depth Estimation: UPV can be used to estimate the depth of cracks present in concrete structures.
e. Internal Flaw Detection: It is effective in identifying internal flaws, cracks, honeycombing, and poor patches within the material.

It's important to note that while ultrasonic pulse velocity testing is an informative indicator of material quality and integrity, other complementary tests, including destructive testing, should be carried out to gain a comprehensive understanding of the structural and mechanical properties of the material.

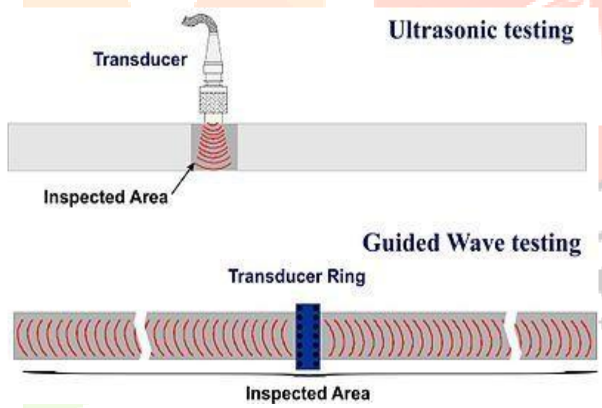

Figure 1. Existing Systems

2.1 Guided Wave Testing

Guided wave testing (GWT) is a non-destructive evaluation technique that harnesses acoustic waves to travel along elongated structures, guided by their boundaries. This unique approach allows these waves to traverse extensive distances with minimal energy loss. Currently, GWT is a widely embraced methodology for inspecting and assessing a variety of engineering structures. In some instances, it can enable the inspection of hundreds of meters from a single testing location. While it is also referred to as guided wave ultrasonic testing (GWUT), ultrasonic guided waves (UGWs), or long-range ultrasonic testing (LRUT), it should be noted that GWT fundamentally differs from conventional ultrasonic testing. Figure 1 provides a visual depiction of how waves are transmitted in both systems.

2.3 Advantages of the Existing Systems

- The depth of the crack can be determined.
- Detection of internal or external concrete loss
- Long-range inspection is possible, i.e., the potential to achieve hundreds of meters of inspection range.

2.4 Disadvantages

- Unskilled people cannot use the devices in order to detect cracks, i.e., intensive training is required.
- The equipment can be quite expensive.
- The sensitivity and precision of these types of measurements are quite weak since the density of cracking must reach comparatively high values before it can be detected.

2.5 Proposed System

The central aim of this proposed method is the utilization of Machine Learning algorithms to categorize images depicting building surfaces into two distinct classes: "Surfaces with cracks" and "Surfaces without cracks." What makes this system particularly noteworthy is its capacity to execute this classification task without the necessity for costly equipment, setting it apart from established methodologies. This approach incorporates the deployment of three discrete Convolutional Neural Network (CNN) models, judiciously integrated to enhance the accuracy of discerning cracks on these surfaces. The training process encompasses the extraction of diverse features from the images, thereby facilitating the system's proficiency in the precise categorization of these surfaces.

2.6 Objectives

The main objectives of the proposed model are:

- To detect whether the given surface image is affected by a crack or not is the mainobjective.
- To train the system effectively using an ensemble version of ResNet50, Inception v3,and 4 Layer Architecture CNN.
- To enable the constructor/building incharge to take appropriate measures to repair the surface based on the result of the system. This in turn ensures the strength of the building and reduces the risk of the building collapsing, building breakage, etc.

3. METHODOLOGY AND ARCHITECTURE

A curated collection of images, encompassing both those featuring cracks and those devoid of any, is subjected to a series of essential preprocessing steps. These images are integral to the training process of the various models employed. Subsequently, these models are collectively integrated into an ensemble Convolutional Neural Network (CNN), functioning as a unified entity. Images slated for classification are submitted to this composite entity, which ultimately furnishes predictions regarding the presence or absence of cracks on the display.

The initial step involves the meticulous preprocessing of the training dataset. This dataset serves as the foundation for model training. An ensemble model, a composite of the three distinct models, is meticulously preserved for subsequent predictive tasks. The training dataset is then used to ascertain the model's accuracy. Following this assessment, the dataset undergoes further preprocessing, culminating in the classification process that leverages the trained model. These outcomes are then meticulously compared to gauge the rate at which the model accurately classifies the presence of cracks.

The preserved ensemble model is made available for download, thus enabling its application in predicting the presence of cracks in other images. This predictive procedure is initiated by providing an input in the form of an image folder, which subsequently undergoes preprocessing. The downloaded model is then set into action to ascertain whether a surface crack is present or not. Finally, the outcomes are presented on the system interface. The information displayed on the system serves as the basis for informed decision-making pertaining to the building. Figure 2 offers a visual representation of the overarching architecture of the proposed system.

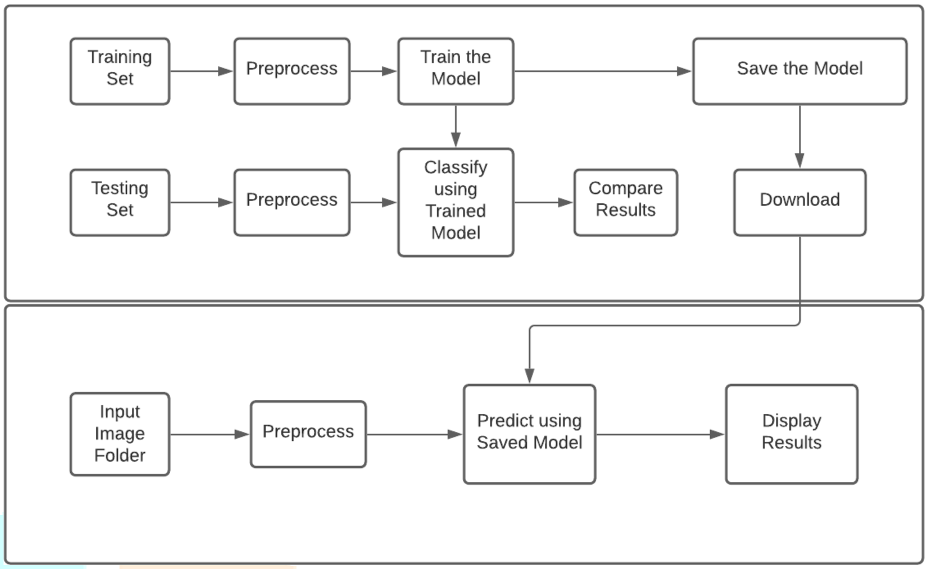

Figure 2. Architecture Diagram for Surface Crack Detection

This work divided into following modules:

1. Loading of Images
2. Preprocessing
3. Training the Models
4. Apply Ensemble CNN
5. Prediction on Test Images

3.1 Loading of Images

- An adequate amount of negative and positive images must be taken.
- The quality of the image must be of at least 144p resolution.
- There should be minimal to no objects in the background.

Although it is not a requirement, the most important thing is that images are all the same resolution. This ensures consistency in both the training and testing phases and finally affects the prediction.

3.2 Preprocessing

- The images must be converted to RGB channels.
- There must be resizing done to 227 x 227 pixels.

The aim of this process is to improve the image data (features) by suppressing unwanted distortions and enhancement of some important image features so that our models can benefit from this improved data to work on. Computers are able to perform computations on numbers and are unable to interpret images in the way that we do. We have to somehow convert the images to numbers for the computer to understand. The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing.

3.3 Training the Models

- There are three models; specifically: ResNet50, Inception v3, and 4 Layer Architecture CNN.
- These models must be trained using the training images present in the dataset.

The dataset used to train the models is taken from Kaggle [2]. In machine learning, training data is the data you use to train a machine learning algorithm or model. Training data requires some human involvement to analyze or process the data for machine learning use. How people are involved depends on the type of machine learning algorithms you are using and the type of problem that they are intended to solve.

3.4 Ensembling the Models

- All three models are then ensemble together to get a collective output.
- To ensemble the models the average precision is taken on the outputs of the model.
- This is known as the CNN-based ensemble model.

A neural network ensemble is a learning paradigm where many neural networks are jointly used to solve a problem. In general terms, an ensemble can be considered a learning technique where many models are joined to solve a problem. This is done because an ensemble tends to perform better than singles improving the generalization ability. The way an ensemble can be carried out doesn't have any limit. The same is true for the number and the types of models considered. The habit to keep in mind is to choose components with low bias and high variance.

3.5 Predicting Test Images

Within this module, the preserved model is deployed to conduct predictions on test images, culminating in the derivation of the ultimate prediction. Following the initial four modules, the final module is engaged to effect predictions on these images. The results yield either a positive or negative outcome. A negative result signifies the absence of a surface crack in the image under assessment, while a positive result indicates the presence of a surface crack. In this context, "prediction" alludes to the output generated by an algorithm, following its training on historical data, and its subsequent application to new data, as it forecasts the likelihood of a specific outcome.

Machine learning models have begun to permeate critical domains such as healthcare, legal systems, and the financial sector. Therefore, comprehending the decision-making process of these models and ensuring its alignment with ethical requirements and legal regulations has become imperative.

It is vital to adhere to the prescribed sequence. This entails sequentially loading images, conducting preprocessing, training the models, applying Ensemble CNN techniques, and ultimately, performing predictions on the test images. The initial step involves loading images captured with an appropriate camera or equivalent device to ensure image resolution adheres to recommended standards. Subsequently, various image transformations are executed as part of the preprocessing measures. The subsequent phase revolves around model training and ensemble integration. Finally, leveraging the ensemble model, predictions are executed on the images to ascertain the presence of surface cracks.

4. ALGORITHM AND MODELS USED

4.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of neural network inspired by the organization and functioning of biological neural networks. In CNNs, individual units, known as neurons, are organized into layers. Neurons within each layer are connected to neurons in the next layer, forming a network for data transmission. Each neuron conducts a simple mathematical computation and shares its output with connected neurons. CNNs have gained prominence in image analysis and interpretation, particularly in medical imaging, thanks to breakthroughs in neural network techniques like dropout and rectified linear units, as well as increased computing power, notably through GPUs. These advancements have enabled CNNs to excel in complex image recognition tasks, including those with numerous object classes, as seen in competitions like ImageNet and COCO. CNNs consist of multiple layers of artificial neurons, which function as mathematical units. When presented with an image, each layer produces activation maps that highlight significant image features. Neurons within the network process pixel values, adjusting them based on their weights and applying activation functions.

The initial layer typically identifies fundamental features like edges in various orientations (horizontal, vertical, diagonal). Subsequent layers extract more complex features such as corners and combinations of edges. As the network progresses deeper, it detects higher-level features, including objects and even specific objects like faces. The core operation in CNNs is "convolution," where pixel values are multiplied by weights and summed. CNNs typically comprise several convolution layers along with other components. The final layer, the classification layer, takes the output of the last convolution layer and generates confidence scores between 0 and 1 to indicate the likelihood of the input image belonging to a particular class (e.g., cat, dog, or horse).

Training CNNs is a significant challenge, involving the adjustment of neuron weights to extract the correct features from images. During training, a large dataset of labeled images is provided, and the network processes these images with initially random weights. The network then compares its output to the correct labels and makes adjustments to its neuron weights, a process known as backpropagation. Backpropagation optimizes weight tuning and gradually improves the network's classification accuracy over several epochs. Each run through the entire training dataset is referred to as an "epoch." As training progresses through multiple epochs, the network's weight adjustments become smaller, and it eventually converges to its best performance. Following training, the network's accuracy is assessed using a test dataset, which comprises labeled images not encountered during training. This evaluation measures how well the network generalizes to unseen data. If a CNN performs well on the training data but poorly on the test data, it is considered to be "overfit," often due to insufficient training data variety or excessive training epochs.

The success of CNNs can be attributed, in large part, to the availability of extensive image datasets developed in recent years. Notably, ImageNet, with over 14 million labeled images, and specialized datasets like MNIST, a collection of 70,000 handwritten digit images, have played a crucial role in advancing CNN technology.

4.2 ResNet 50

ResNet, short for Residual Networks, stands as a foundational neural network architecture widely employed in various computer vision applications. In 2015, this model claimed victory in the ImageNet challenge, marking a pivotal moment in deep learning. What set ResNet apart was its ability to effectively train exceptionally deep neural networks, exceeding 150 layers. Prior to ResNet, the challenge of vanishing gradients hindered the training of such deep networks.

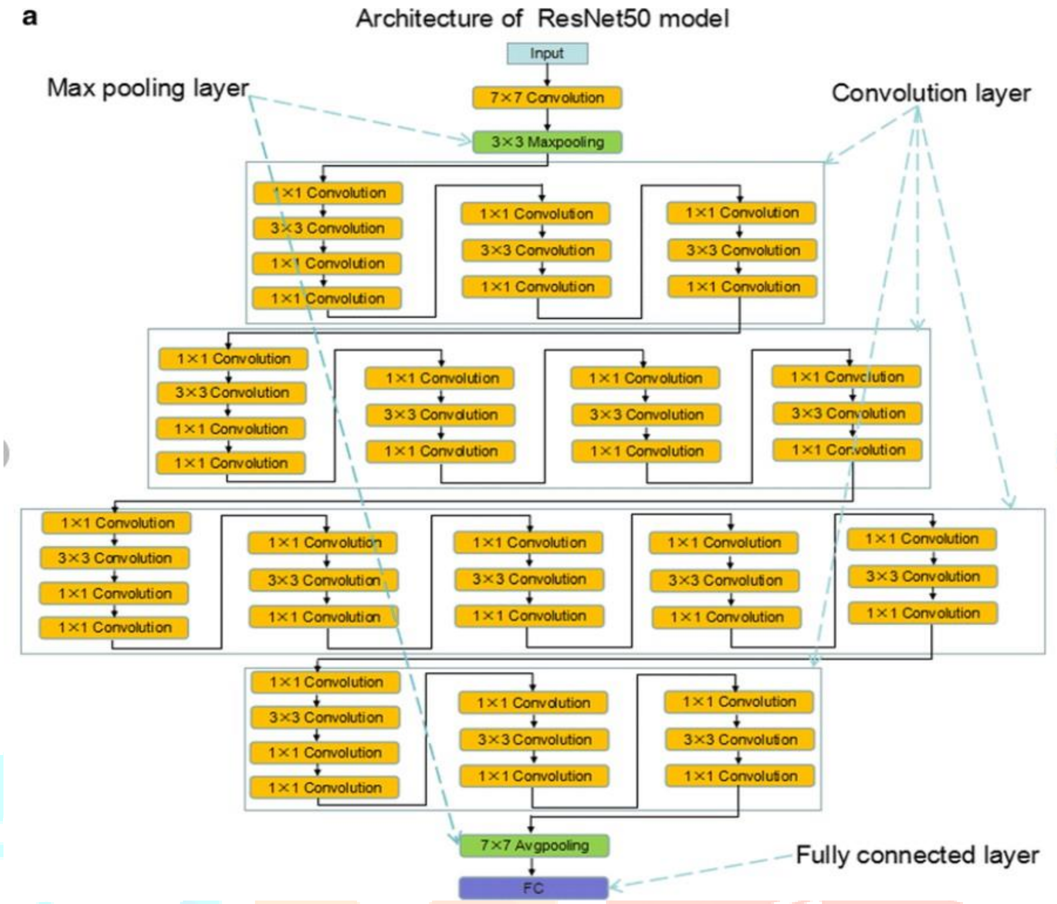

Figure 3. ResNet50 Architecture

The ResNet-50 model is a notable example and is structured into five stages, each consisting of a convolution block and an identity block. Each convolution block incorporates three convolution layers, and the identity block also consists of three convolution layers. In total, ResNet-50 boasts over 23 million trainable parameters. To overcome the vanishing gradient issue, ResNet makes use of skip connections, which involve adding the output of an earlier layer to a later layer. This architectural innovation is crucial for mitigating the vanishing gradient problem.

The implementation of ResNet models is facilitated by deep learning frameworks like Keras, which allows for the loading of pretrained ResNet-50 models. ResNet, an abbreviation for Residual Network, represents a specific category of neural networks introduced in the landmark paper "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in 2015. The ResNet models achieved remarkable success, as evidenced by their performance in various competitions and applications:

- Secured 1st place in the ILSVRC 2015 classification competition with a top-5 error rate of 3.57% (as an ensemble model).
- Achieved 1st place in ILSVRC and COCO 2015 competitions in ImageNet Detection, ImageNet localization, Coco detection, and Coco segmentation.
- Replaced VGG-16 layers in Faster R-CNN with ResNet-101, leading to relative improvements of 28%.
- Demonstrated the ability to train efficient networks with 100 and even 1000 layers.

In deep neural networks, it is common to add additional layers to improve accuracy and performance when solving complex problems. These added layers progressively learn more intricate features. For instance, in image recognition, the initial layers might detect edges, followed by layers identifying textures and eventually layers recognizing objects. However, conventional convolutional neural networks exhibit a limit to their depth, as illustrated in Figure 3.

ResNet addresses the vanishing gradient challenge by incorporating skip connections. These connections create an alternate shortcut pathway for gradients to flow through, ensuring that the model can learn identity functions. This guarantees that higher layers perform at least as well as lower layers, if not better.

Keras, an open-source neural network library written in Python, is a versatile tool that can be seamlessly integrated with popular deep learning frameworks like TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Keras is designed to facilitate rapid experimentation with deep neural networks, making it a preferred choice for many researchers and practitioners.

4.3 Inception v3

Inception v3 is a convolutional neural network architecture derived from the Inception family, which introduces various enhancements such as Label Smoothing, Factorized 7 x 7 convolutions, and the integration of an auxiliary classifier to convey label information deeper into the network. It also incorporates batch normalization for side head layers. Pretrained Inception v3 models can be loaded using Keras, and the general architecture is illustrated in Figure 4.

Inception v3 primarily aims to optimize computational efficiency by refining the earlier Inception architectures. This concept was put forward in the 2015 paper titled "Rethinking the Inception Architecture for Computer Vision."

In comparison to VGGNet, Inception Networks (specifically GoogLeNet/Inception v1) have demonstrated superior computational efficiency. This pertains to both the number of network parameters and the economical costs, including memory and other resources. When adapting an Inception Network for different applications, it's essential to ensure that the network's computational advantages are not compromised. To facilitate the adaptation of an Inception network, Inception v3 incorporates several techniques for network optimization. These techniques encompass factorized convolutions, regularization, dimension reduction, and parallelized computations.

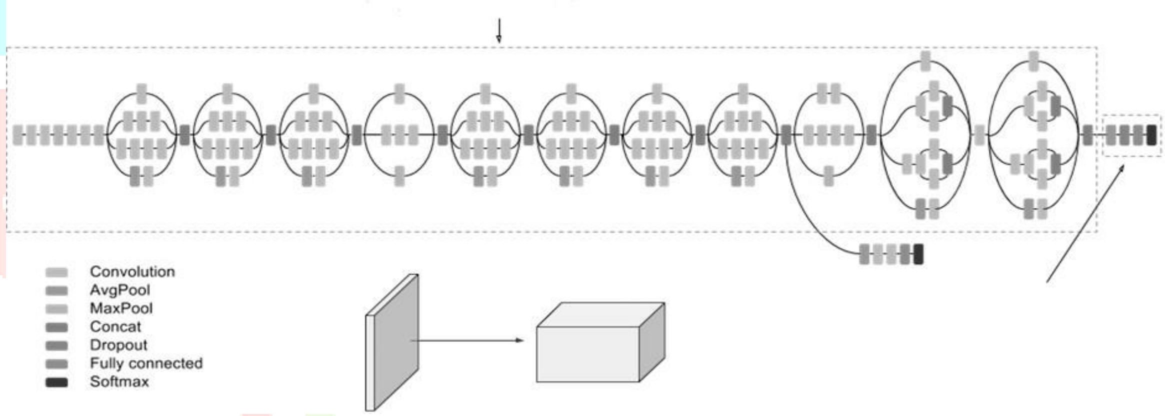

Figure 4. Inception v3 Architecture

4.4 Four Convolution Layers CNN Model

The initial layer of the model features an input layer with an input shape of (227, 227, 3), representing the dimensions of the image (height, width, and channels). Following this, we have the Convolution layer, often referred to as the feature extractor layer. This layer is responsible for extracting image features.

In the Convolution layer, a portion of the image is connected to perform a convolution operation, which involves calculating the dot product between the receptive field (a local region of the input image with the same size as the filter) and the filter. The result of this operation is a single value in the output volume. The filter is then slid over the next receptive field of the input image with a certain stride, and the operation is repeated. This process continues until the entire image has been processed. The output from the Convolution layer serves as input for the subsequent layer.

Next, the output from the Convolution layer is passed to the max pooling layer. The purpose of the pooling layer is to reduce the spatial dimensions of the input image after convolution. This is particularly useful when sandwiched between two convolution layers. Applying fully connected layers (FC) directly after a Convolution layer without pooling would be computationally expensive, which is typically not desirable. Therefore, max pooling is employed as a means to reduce the spatial dimensions of the input image. In this model, the Convolution and pooling layers are repeated four times, resulting in a four-layer CNN architecture.

4.5 Arithmetic Mean Ensemble Method and Algorithm

The proposed system utilizes the Arithmetic Mean Ensemble Method to merge the models mentioned earlier. Ensemble methods are strategies that involve building multiple models and then combining their outputs to achieve more accurate results. In many machine learning competitions, ensemble methods have consistently outperformed single models.

In the simple arithmetic mean ensemble method, for each instance in the test dataset, the average predictions are calculated. This approach often mitigates overfitting and leads to a smoother regression model. The following pseudocode illustrates this averaging method:

```
final_predictions = []
for row_number in range(len(predictions)):
    final_predictions.append(mean(predictions[row_number, ]))
```

Ensemble methods offer increased flexibility and can scale with the amount of available training data. However, one drawback of this flexibility is that they learn through stochastic training algorithms, which makes them sensitive to the specifics of the training data. Consequently, they may yield different sets of weights and predictions in each training iteration.

In summary, ensembles of models involve the practice of combining predictions from multiple statistical models to produce a final prediction. This approach introduces diversity in the model's representational capacity, analogous to seeking opinions from multiple experts. Ensembles are a common enhancement for traditional machine learning models, such as upgrading decision trees to random forests. However, forming ensembles of deep models, which have lengthy training times, from scratch is often impractical.

4.6 Kaggle Dataset

Kaggle serves as an online community catering to data scientists and machine learning enthusiasts. This platform offers users the ability to discover and share datasets, create and experiment with models within a web-based data science environment, collaborate with fellow data scientists and machine learning engineers, and partake in data science competitions to tackle various challenges.

Kaggle is a valuable resource for both current and aspiring data scientists, providing access to a wealth of resources. It boasts a collection of 19,000 public datasets and 200,000 public notebooks, making it an ideal starting point for those looking to begin or contribute to projects, hone their skills, and expand their data science portfolios. Kaggle further offers a user-friendly Jupyter Notebooks environment that requires no setup, provides customization, and grants access to free GPUs. Additionally, it hosts a vast repository of community-published data and code.

Beyond this, Kaggle provides micro-courses covering topics like machine learning, Python, deep learning, and more. These courses can typically be completed in 3-7 hours, making them a convenient learning resource. Users can also engage in competitions designed to solve real-world data science problems, offering a chance to put their skills to the test and gain insights from a community of over 3 million data scientists.

Kaggle welcomes data scientists of all levels, whether beginners eager to learn, contribute, and build their skills, advanced practitioners seeking competition challenges, or those falling in between. The platform enjoys high regard within its community, with users relishing the competition opportunities and the chance to continue their learning journeys. Many success stories highlight individuals who began as novices in competitions and evolved into highly skilled practitioners.

The dataset in use comprises images depicting various concrete surfaces, some with cracks (positive category) and some without (negative category). It features a total of 10,000 images, with 5,000 in each class, all having dimensions of 227 x 227 pixels and RGB channels. This dataset stems from 458 high-resolution images (4032x3024 pixels), following the methodology proposed by Zhang et al. in 2016. Notably, the high-resolution images exhibit considerable variability in surface finish and illumination conditions. It's worth mentioning that no data augmentation techniques like random rotation, flipping, or tilting were applied to this dataset.

The images featured in Figure 5 represent the negative category (crack-free), while those in Figure 6 depict the positive category (with cracks). These images adhere to the standards outlined in Chapter 2, ensuring an adequate number of both negative and positive images, a minimum image resolution of 144p, and minimal to no background objects.

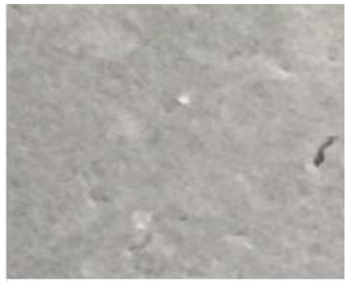

Figure 5. Image with No Surface Crack

Figure 6. Image with Surface Crack

5. RESULTS AND DISCUSSIONS

The system initially displays an initial screen prompting the user to enter a file consisting of images. The user must click on the "Choose Files" to be able to select the respective file consisting of the images to be classified. After this selection is done, the user must click on the "Submit" button. After which, the trained model will be applied in order to make the respective prediction of whether there is a presence of surface cracks or not. The results displayed on the screen will be in the form as depicted in Figure 7.

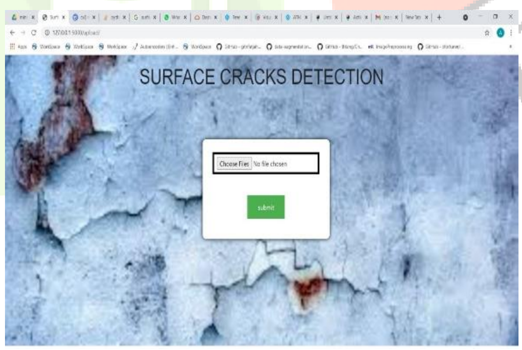

Figure 7. Initial Screen

The following image, that is, Figure 8 shows the selection of the folder in order to upload the files to the system. The folder consists of the images that are to be predicted as either having no surface crack or having a surface crack.

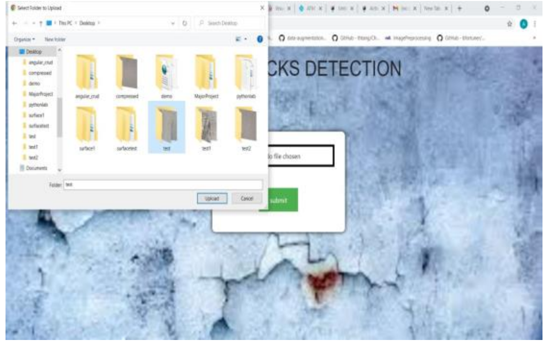

Figure 8. Uploading the Image Folder

Figure 9 shows the screen after the folder has been selected. Now the user must click on the green "submit" button in order to proceed with the prediction process.

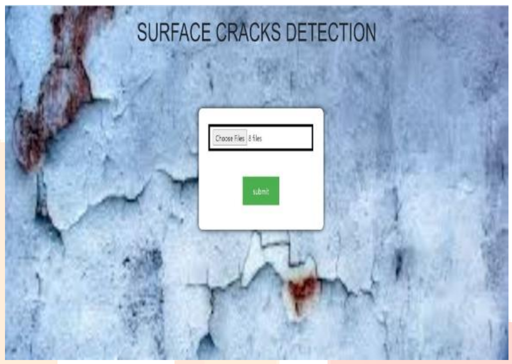

Figure 9. Uploaded Image Folder

RESULTS

Filename	Crack Detection
18001.jpg	Negative
18001_1.jpg	Positive
18002.jpg	Negative
18002_1.jpg	Positive
IMG_20210507_200120 (1).jpg	Positive
IMG_20210507_200127 (1).jpg	Positive
IMG_20210507_2001381.jpg	Negative
IMG_20210507_2001481.jpg	Negative

Figure 10. Results Displayed on Scree

Figure 10 shows the screen displayed after the prediction is completed. A table with two columns (File name and Prediction) is displayed with the results row-wise. A negative result indicates that there is no crack in the uploaded image. Whereas, a positive result indicates that there is a crack in the uploaded image.

5.1 Individual Model Evaluation

ResNet 50

The amount of 'No Surface Crack' images that got classified correctly is 1000 and those images that got classified as Surface Crack is 0. The number of 'Surface Crack' images that got classified incorrectly are 7 and those that got classified correctly is 993.

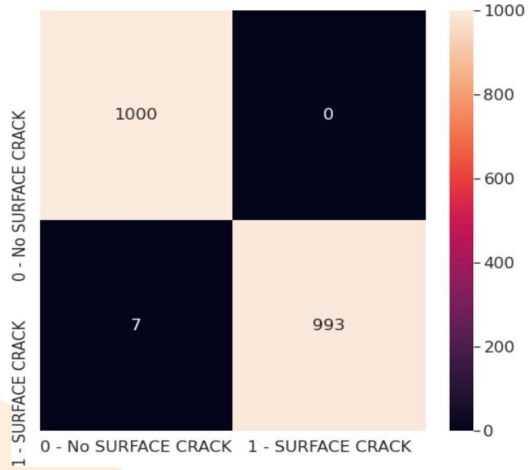

Figure 11. Confusion Matrix

Classification Report on Test Images

	precision	recall	f1-score	support
0	0.99	1.00	1.00	1000
1	1.00	0.99	1.00	1000
accuracy			1.00	2000
macro avg	1.00	1.00	1.00	2000
weighted avg	1.00	1.00	1.00	2000

Figure 12. Classification Report

Inception V3

The amount of 'No Surface Crack' images that got classified correctly is 1000 and those images that got classified as Surface Crack is 0. The number of 'Surface Crack' images that got classified incorrectly are 14 and those that got classified correctly is 986.

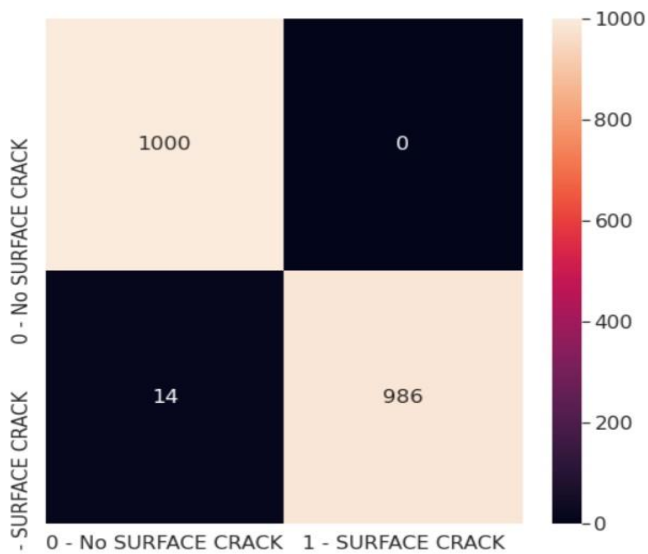

Figure 13. Confusion Matrix

Classification Report on Test Images

	precision	recall	f1-score	support
0	0.99	1.00	0.99	1000
1	1.00	0.99	0.99	1000
accuracy			0.99	2000
macro avg	0.99	0.99	0.99	2000
weighted avg	0.99	0.99	0.99	2000

Figure 14. Classification Report

4 Layer Architecture CNN

The amount of 'No Surface Crack' images that got classified correctly is 996 and those images that got classified as Surface Crack is 4. The number of 'Surface Crack' images that got classified incorrectly are 9 and those that got classified correctly is 991.

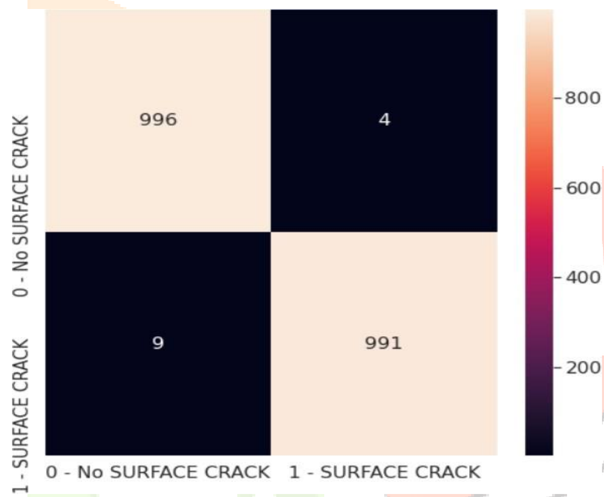

Figure 15. Confusion Matrix

Classification Report on Test Images

	precision	recall	f1-score	support
0	0.99	1.00	0.99	1000
1	1.00	0.99	0.99	1000
accuracy			0.99	2000
macro avg	0.99	0.99	0.99	2000
weighted avg	0.99	0.99	0.99	2000

Figure 16. Classification Report

5.2 Overall Model Evaluation

The amount of 'No Surface Crack' images that got classified correctly is 1000 and those images that got classified as Surface Crack is 0. The number of 'Surface Crack' images that got classified incorrectly are 9 and those that got classified correctly is 991.

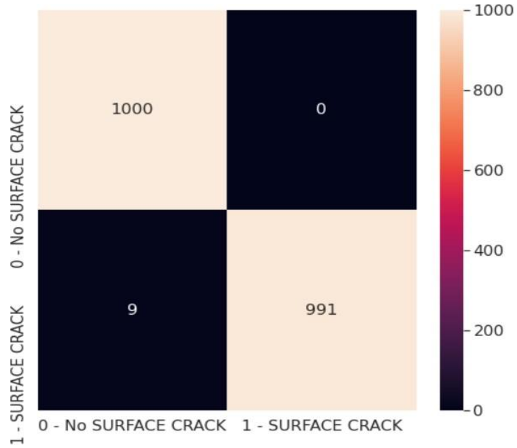

Figure 17. Confusion Matrix

Classification Report on Test Images

	precision	recall	f1-score	support
0	0.99	1.00	1.00	1000
1	1.00	0.99	1.00	1000
accuracy			1.00	2000
macro avg	1.00	1.00	1.00	2000
weighted avg	1.00	1.00	1.00	2000

Figure 18. Classification Report

6. CONCLUSIONS AND SCOPE FOR FUTURE ENHANCEMENTS

Detecting cracks in buildings is crucial for assessing their structural health, but it often involves time-consuming, error-prone manual work and costly equipment. Our proposed system addresses these challenges by leveraging machine learning, utilizing ResNet 50, Inception v3, and a 4-layer architecture to predict the presence of surface cracks in images.

This system significantly reduces time and costs in crack detection, benefiting the construction industry and overall efficiency. It not only serves the industry but also ensures building safety during inspections, reducing the risk of collapses.

The system can be integrated into a mobile app and drones, allowing for efficient crack location scanning. Future enhancements may include identifying crack types, depth, and material compatibility. Additionally, data encryption and authentication can protect user data integrity and prevent unauthorized access.

REFERENCES

[1] G. Hidalgo. (2017, April 24). Openpose. Retrieved from https://github.com/CMU-Perceptual-Computing-Lab/openpose/tree/master/models

[2] A. Pandian. (2020, May). Surface Crack Detection. Retrieved from https://www.kaggle.com/arunrk7/surface-crack-detection

[3] D. Gooch, et al. (2003). Crack Detection. Retrieved from https://www.sciencedirect.com/topics/materials-science/crack-detection

[4] Ronacher, et al. (2010, April 16). Flask. Retrieved from https://palletsprojects.com/p/flask/

[5] P. Dwivedi. (2019, Jan 4). Understanding and Coding a ResNet in Keras. Retrieved from

https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d3 3

[6] C. Szegedy, et al. (2016). Inception-v3. Retrieved from https://paperswithcode.com/paper/rethinking-the-inception-architecture-for

[7] Golubev, et al. (2004, January 1). Open Source Computer Vision Library.

Retrieved from https://github.com/opencv/opencv

[8] G. Anbarjafari. (2014, September 29). Digital Image Processing. Retrieved from https://sisu.ut.ee/imageprocessing/book/1

[9] A. Rosebrock. (2016, March 28). Measuring size of objects in an image with OpenCV. Retrieved from

https://www.pyimagesearch.com/2016/03/28/measuring-size-of-objects-in-an-image-with-opencv/

[10] A. Rosebrock. (2014, January 20). Basic Image Manipulations in Python and OpenCV: Resizing (scaling), Rotating, and Cropping. Retrieved from https://www.pyimagesearch.com/2014/01/20/basic-image-manipulations-in-python-and-o pencv-resizing-scaling-rotating-and-cropping/