Deep Convolutional Neural Network for Real Time Visual Inspection of Structural Audit

Valmik B. Nikam, Nilkamal More, Aditya Shenoy, Rahul Bhoir, Kalyani Borkar, Rani Deore

Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai, India
vbnikam@it.vjti.ac.in

Abstract: Structural audit requires manual expertise of surveyors that can be subjective and erroneous. To solve this problem, researchers are studying digital image processing and deep learning techniques for crack detection. The digital image processing techniques are specialized only to the domain it is designed for and deep learning techniques have only been applied to a single class of structural faults, that is a crack, but there are other structural faults that are part of the visual inspection of the structural audit. There is a need for a single generalized model, which can classify more than one structural defect during the visual inspection of structural audit. The proposed system classifies cracks, dampness and paint peel-off using a single generalized model as part of visual inspection while doing structural audit of infrastructures. The contribution of this work is 95% overall accuracy, 95% average precision, 95.25% average recall, and 95% average F1-score. Usage of a non-complex model has reduced the time taken for training the model and getting predictions from the model. This paper performs multi label classification, which gives more information to the structural auditors than just crack information.

1. Introduction

Construction of different infrastructures is a necessity for human life as it brings value to the society, fulfils human needs, and develops the geographic area around it. As the building gets old, the risk of living/working in the building increases. The structure and walls of old buildings show cracks, paint peel off, and dampness. The building needs to be examined after 30 years. The process is called structural audit. For structural audit, the expert has to go to the location and do visual inspection of the building, do different chemical tests, and so forth. While performing visual inspection for detection of cracks, dampness, paint peel off, the process totally depends on the surveyor's experience and knowledge. Sometimes there may be human errors.

The literature states digital image processing techniques for crack detection. Image processing techniques cannot automatically learn complex features and thus need to be hardcoded and is specialized for only one domain. Papers, which are working in the same domain, are detecting only cracks in their work. However, there are other structural defects in the infrastructure that can be detected visually including the dampness of the wall and peeling off paint.

There is a need for a single generalized model, which can classify more than one structural defect during the visual inspection of structural audit. The proposed system classifies cracks, dampness and paint peel-off using a single generalized model as part of visual inspection while doing structural audit of buildings.

The remainder of this paper is structured as follows: the Literature Review section offers a background of the work done in this area; the Proposed Model portion describes the classification models of four classes of the wall in the structural audit; and in the Results and Analysis section, this research paper presents the results of the model. The final sections conclude the proposed work and add scope of research for future works in the related fields.

2. Literature Review

S. Kang et al. [1] state that there are several techniques used for crack detection. Laser method is time consuming, the acceleration sensor method has low accuracy. So a camera based method that uses image processing is used which gave an accuracy of 89.33%. J. Vora et al. [2] research on binarization, a RGB image undergoes binarization and is converted to a grayscale image and then to a binary image setting a threshold value. User's parameters (sensitivity and window size) in binarization sometimes causes the crack to remain undetected. Therefore, two new user's parameter are set P_w and P_l. Using these parameters two binary images are obtained which undergo an edge detection algorithm. R. K. Meghana et al. [3] state that images are captured using UAV, then converted to grayscale. Void detection on concrete is done using MATLAB. Image stitching is carried out to merge the images to get a clear view of crack. A binary image is formed and then thresholding is carried out to remove skeleton pixels. X. Liu et al. [4] propose that for enhancing the noisy image multiscale enhancement method is used. MESG algorithm is used. SOBEL filter is used for obtaining binary image. Morphological processing is done to remove noise and join parts of cracks. W. Li et al. [5] present method to identify deformation, crack and surface damage normal edge detection is not enough, and therefore CANNY edge detection algorithm is used. Using this, the paper got the relative length and width, the centre ratio. M. Lin et al. [6] present that an original image is pre-processed using a band pass filter method to reduce the noise in the image and then an adaptive line detector is used due to changing region size and direction. Then the HMRF-EM algorithm is used to increase the accuracy to label the data. Start and ends of labelled data are clustered on relative positions to get true ends. L. Peng et al. [7] states a traditional method for defect detection on the wind turbine, which is very costly, has low efficiency and takes lots of time. A new method applies

Wiener filter on the image to remove the motion blur. An adaptive median filtering method is used for removing the noise from the image. To regain the image quality, enhancement algorithms are used. L. Calderón et al. [8] state that edge detection is applied on the image to reduce pixels. Not every dark pixel is a crack so the pixels which is dark but not crack is removed by passing through a modified line kernel.

Y. Wang et al. [9] present a research to capture random size inputs by using Full Convolution Network (FCN). This paper also captures temporal features of subsequent frames by using Convolutional Long Short Term Memory (Conv-LSTM). As this paper is using FCN, the output generated is a pixel level segmentation of videos. H. Shi et al. [10] research on the object detection over traffic camera videos. The paper compares two deep learning methods namely Region-based Full Convolutional Network (R-FCN) and Faster Region-based Convolutional Neural Network (R-FCN). The results of this paper shows that RFCN performs better than Faster RCNN. P. Babayan et al. [11] compare three deep learning techniques for vehicle and pedestrian recognition, namely RCNN, YOLO, and SSD. Faster R-CNN performs well in terms of AUC and mAP. The processing time of Faster R-CNN is higher than YOLO. B. Hou et al. [12] state a method of object detection in high resolution remote sensing videos. This paper considers two architectures, YOLO and Faster R-CNN. Faster R-CNN was selected as it produces a better result for smaller size objects which are present in remote sensing video. K. Mouri et al. [13] focus on using R-FCN for selecting goods (objects) in stock in the logistics industry automatically. After object detection, the paper's methodology uses Grow cut algorithm for segmentation. This segmentation refines the bounding box produced by R-FCN to the real shape of the object using background and foreground seeding method. B. Tian et al. [14] state object detection methods in video dataset using deep learning methods for traceability. Once trained, the model is deployed online and objects are detected and traced using Gaussian background modelling and other non-parametric modelling. The results show that this deep learning method is suitable for downloaded videos as well as real time video analysis. M. Gasmallah et al. [15] state a methodology for Predictive Object Detection (POD) using deep learning techniques. The architecture used in the paper uses YOLOv2 (You Only Look Once version 2) and LSTM (Long Short Term Memory). The YOLO architecture has time benefits as it processes the image only once. The LSTM architecture is used to incorporate temporal features of previous frames in the video. A. Anjum et al. [16] propose a method for video stream analysis using cloud computing for object detection and classification. This paper states that saving the input video data on the cloud, decoding videos, and moving operator function code to the computing nodes on the cloud can give scalability and time benefits. The nodes on the cloud have GPU cores, thereby collectively improving the time required for analysis of videos.

N. Truong et al. [17] propose a Super Resolution (SR) reconstruction and a marker detection model based on deep learning for drone landing. In SR reconstruction, the low-resolution image frame size of 320 × 240 pixel is captured which is given to the module for detecting the landing area. The inputted image is given to CNN with DCSCN produced high-resolution image. C. Wang et al. [18] state an approach for map construction of the unfamiliar surroundings and using it to localize. The model includes Deep Reinforcement Learning (DRL), Reinforcement Learning (RL), Markov Decision Processes (MDP), and Partially Observable Markov Decision Processes (POMDP), which helps in navigation in complex environments. Recurrent Deterministic Policy Gradient (RDPG) and fast RDPG algorithm helps in navigation. S. Hassan et al. [19] present deep learning based YOLO model for detecting UAV. YOLO model takes input image, which is resized and divided into grids. If the object falls on the grid centre, then the duty of the grid is to detect an object, forecast B's bounding boxes, calculate bounding box x, y, z, h confident, predict conditional probability, determine bounding box confidence score, otherwise considered as an undefined body. YOLOv2 is 92.10% precise, and YOLOv3 is 95.20%, which is enhanced but YOLOv3 took longer training time. A. Zeggada et al. [20] state a multi-label classification model for UAV images. In this model, CNN features extracted from tile using GoogLeNet. For a multi-labelling issue, the paper proposed a multi-label layer. A. Mahadik et al. [21] state structural audit method for building. It is necessary and important to carry out the structural audit specialists and act instantly through advice provided in the audit report to ensure safety of the building.

N. Yusof et al. [22] propose a method to classify and detect pavement crack using deep CNN. This study states a deep convolution neural network (DCNN) that is capable to sense the crack robustly while dealing with a complicated background image. The network is trained using stochastic gradient (SGD) training algorithm. Exploring how the size of input image affects deep CNN with respect to detection and classification performance, different sizes, and grid scale are adopted in architecture. S. Liang et al. [23] identify the problem in organizing crack images in health checking of structures and hence propose a novel algorithm for concrete crack detection. In this algorithm, obtained attributes are used as input to train a SVM classifier. This algorithm had a better capability to identify cracks with noticeable contours. Hessain matrix based linear filtering approach is used to increase the crack area and regulate the threshold gap to get the crack binarization segmentation result. L. Zhang et al. [24] state that using the Internet of Things (IoT) technology for collecting video data is easy. The methodology used in this paper is CNN. This method could successfully solve the problems of the high risk factor in domestic and low fracture analysis. M. David Jenkins et al. [25] represent a DFCN network to achieve the pixel-wise classification of road and pavement cracks. This paper presents an algorithm for semantic segmentation of road and pavement surface cracks using a U-Net. Dataset used is openly accessible CrackForest. F. Nagata et al. [26] state application of DCNN for vision-based fault examination. The DCNNs are trained using the man-made images and then tested through classification trials. The designed DCNNs are trained using the generated images and then evaluated through classification experiments. X. Wang al. [27] presents a research on the segment of roadway crack images into variable scales of nets and then the image is cut into nets. CNN is to check for cracks. The model only keeps the nets that contain crack so that the bones of crack is conserved. S. Gibb et al. [28] propose a method that utilizes Genetic Algorithm for CNN model improvement. This method evolved several parameters

that dictate the structure of CNN, including: the number of conv layers, filters size, and the number of filters. Developing CNN structures produces high-performance networks that have higher sorting accuracy than the existing network when tested on images containing cracks. Nilkamal More et al. [29] states the use of Support Vector Machine (SVM) for the classification of satellite images and for identifying green space areas.

3. Proposed Solution

The proposed system classifies cracks, dampness and paint peel-off using a single generalized model as part of visual inspection while doing structural audit of buildings. As the structural audit domain is not explored much before, two model architecture were experimented including Transfer Learning (ResNet-50) and a custom Convolutional Neural Network, Effective Defect Classifier (EDC).

3.1 Transfer Learning

Training a model from scratch with a small dataset does not always give good accuracy. So one could try to increase the size of the dataset by using available similar datasets on the Internet, but as this domain is not explored previously, there was no similar dataset available. Alternatively, one can use someone else's model which is already trained on a very large dataset. Transfer learning (TL) is a research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. This paper tried using the pre-trained ResNet-50 model (Figure 1) for the problem. This model is trained on the CIFAR-10 dataset. The dataset is made of animals, vehicles, fruits, insects, man-made objects, and so forth.

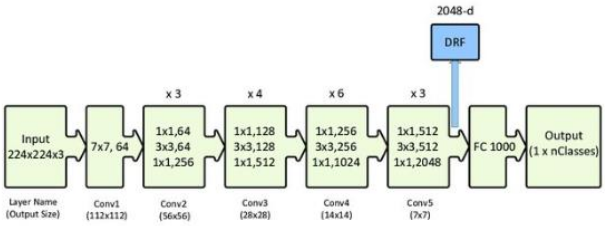

Fig. 1. *ResNet-50 Architecture*

3.2 Effective Defect Classifier (EDC) Architecture.

The classification problem involves detection of complex features in the image data. Hence, a custom Convolutional Neural Network (CNN) was tried. CNN is a type of Deep Learning technique in which the model tries to capture different features, shapes, edges, and so forth in the input image. A typical architecture of CNN includes convolution layers at the beginning and fully connected dense layers at the end. The convolutional layers extract features from the input and the dense layers combine these features to learn to recognize the classes in the dataset.

For the architecture, hyper parameters like the number of convolutional layers, the number of filters in each convolutional layer and number of dense layers were tuned. The final architecture is illustrated in Figure 2, which consists of four convolutional layers with 16 filters each and a kernel size of 3*3. Max Pooling of 2*2 is used after each convolutional layer. One hidden dense layer is added before the output layer containing 32 nodes. The input layer matches the dimensions of the images in the dataset, that is (256, 256, 3) where 3 is the number of channels in JPG images. The output layer consists of four nodes as the dataset has four classes. The dataset consists of 36,000 images (32,000 training images and 4,000 validating images).

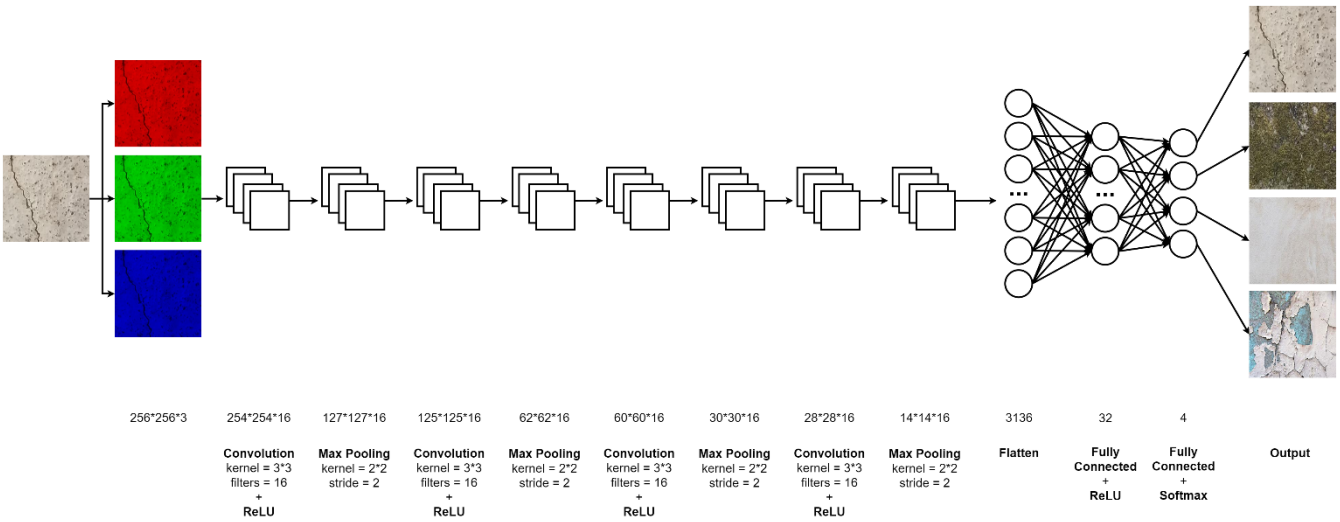

Fig. 2. *Effective Defect Classifier (EDC) Architecture*

4. Result and Analysis

This section discusses the characteristics of the custom dataset and the pre-processing algorithm applied to the dataset before passing it to the model. This section also analyses the results obtained using ResNet-50 and Effective Defect Classifier (EDC) Architecture based on various metrics.

4.1 Dataset

This paper is using a custom dataset, as the dataset for all structural defects has not been created before. The dataset consists of four classes: Crack, Dampness, Paint Peel Off, and the Plain wall. The images were captured in the Veermata Jijabai Technological Institute (VJTI) campus, Mumbai. These images are cropped and then each cropped sample is resized to 256*256 pixels. Using RGB images easily captures many important features of the classes can be instead of using grayscale images. The dataset consists of 36,000 images in total out of which 32,000 images were used for training and 4,000 were used for validating. Table 1 shows the count of raw cropped samples for each class. Then pre-processing techniques were applied to the raw images to make it ready for model training and validation.

Table 1 Information about Raw Dataset

Class	Raw cropped Samples
Crack	705
Dampness	630
Paint Peel Off	650
Plain Wall	1095
Total	3080

Each of the four classes contains an unequal amount of cropped images. So random 625 samples are picked for each class from the cropped samples. Each class now has an equal number of samples of 625 images per class (total 2,500 images), which makes the dataset not skewed to any class. The four classes are not orientation dependent that is even after flipping, rotating orthogonally, or transposing on any diagonal, the class would not change. For example, a crack if rotated by 90 degrees would still remain a crack. Thus seven non-distorting transformations (it does not add any synthetic pixels) are applied to the images. These seven non-distorting transformations are: rotate the image by 90 degrees, rotate image by 180 degrees, rotate the image by 270 degrees, flip the image horizontally, flip the image vertically, transpose image, transverse the image.

These original images and their seven non distorted transformation images are saved. By this step, the dataset size is scaled up by a factor of eight resulting to a total of 8*625 = 5000 images per class (total 20,000 images). These images are then split into training and validating images in an 80:20 ratio. So 4000 images per class (total 16,000 images) are used for training purposes while 1000 images per class (total 4,000 images) are used for validation. In the process of training, distorting transformations are applied so that the model does not overfit and generalizes on unseen output. Two random distorting transformations are generated per training image. Total training images per class then results to 4000*2 = 8000 images per class (total 32,000 images). Both training images and validate images are then saved in the train folder and validate folders respectively.

4.2 Experimental Analysis

This section, experiments and analyses different parameters for the model, which will improve the accuracy of the model. This section then discuss the metrics of the experiment results.

4.2.1 Optimal Training and Validation Split of Dataset: As a custom dataset is generated for this paper, various split points were experimented to split the dataset into training and validating images. Six split points were tried in this experiment: 65:35, 70:30, 75:25, 80:20, 85:15, and 90:10. Figures 3 to 14 have plotted the accuracy versus epochs and loss versus epochs for each of these model experiments.

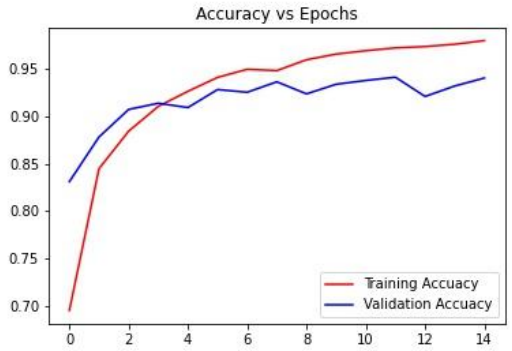

Fig. 3. *Accuracy vs Epochs 65:35 split*

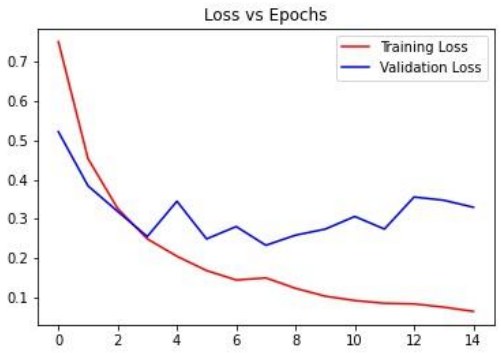

Fig. 4. *Loss vs Epochs 65:35 split*

Figures 3 and 4 show that the intersection between training accuracy (or loss) and validation accuracy (or loss) is at epoch 3. This early intersection indicates that due to lower training data, the model has started to overfit since epoch 3. This overfitting indication is clearer in Figure 4 where the validation loss is gradually increasing as the number of epoch's increases.

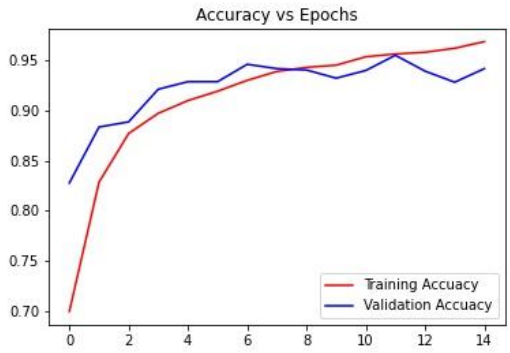

Fig. 5. *Accuracy vs Epochs 70:30 split*

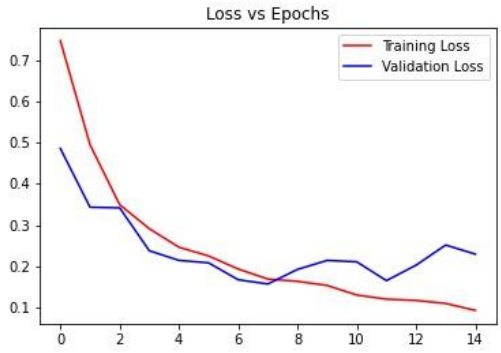

Fig. 6. *Loss vs Epochs 70:30 split*

In the 70:30 split case, the intersecting point is shifted towards right at epoch 8. This shift is because the model is getting more training images this time and is thus learning better than the 65:35 case. But similar to the 65:35 case, the model is showing overfitting indications at epoch 8 and onwards, which can again be seen by the gradual increase in the slope of validation loss in Figure 6.

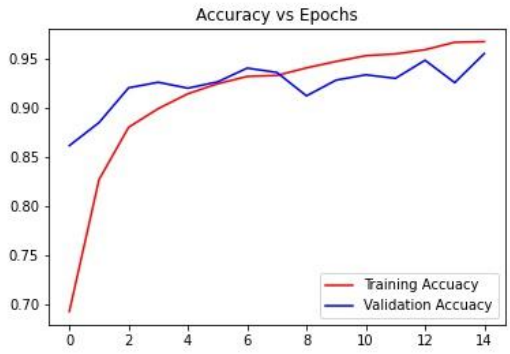

Fig. 7. *Accuracy vs Epochs 75:25 split*

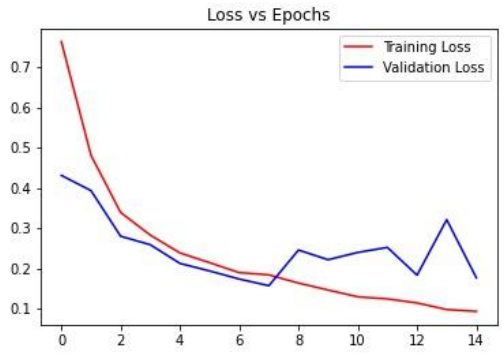

Fig. 8. *Loss vs Epochs 75:25 split*

Next, the 75:25 split was tried. This experiment showed very less overfitting, as compared to the above two cases, which is shown in Figures 7 and 8. In Figure 8, the validation loss almost has a zero slope even after the intersection point, which is at epoch 7, which indicates that this model is showing signs of a well-fitted model. Thus, 75:25 split was one of the candidates for the final model.

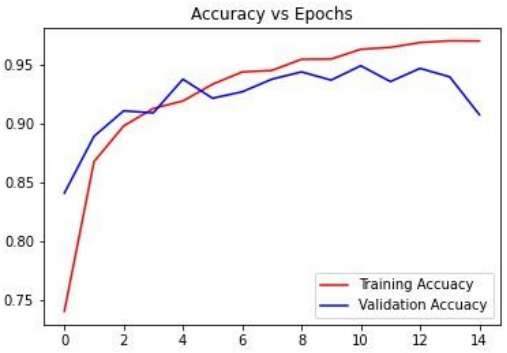

Fig. 9. *Accuracy vs Epochs 80:20 split*

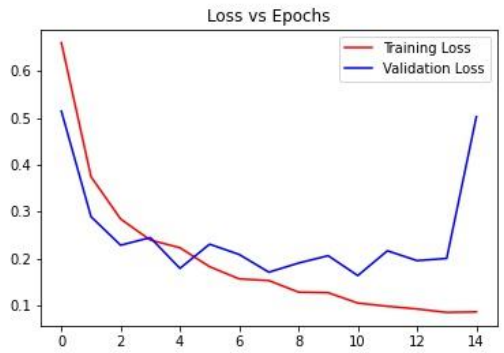

Fig. 10. *Loss vs Epochs 80:20 split*

When the 80:20 split case was tried, the Figures 9 and 10 indicated similar signs as of the 75:25 split. The validation loss in Figure 10 has almost zero slope after the intersection point (at epoch 3). Thus, this model has also shown signs of a well-fitted model. Thus, the 80:20 split was also one of the final candidates for the final model.

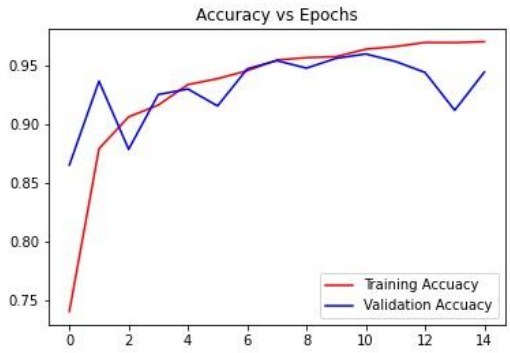

Fig. 11. *Accuracy vs Epochs 85:15 split*

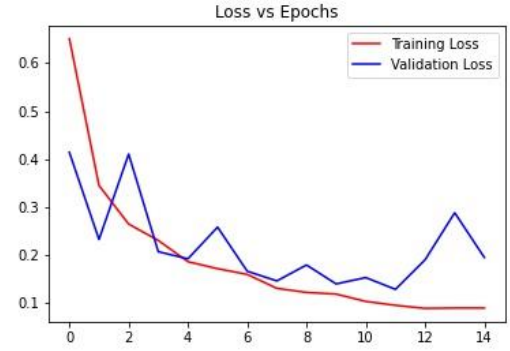

Fig. 12. *Loss vs Epochs 85:15 split*

Figures 11 and 12 show the result of the 85:15 split case. The model is showing signs of overfitting at epoch 13 and further where even the validation accuracy in Figure 11 is decreasing along with an increase in the validation loss in Figure 12. This model is over fitted. The probable reason for this overfitting is due to the higher proportion of training dataset as compared to the validation dataset. This reason is also applicable to the next case of 90:10 split, as the training proportion is much higher than the validation in the 90:10 case, graphs have a huge gap between the training and validation accuracy in Figure 13 and training and validation loss in Figure 14

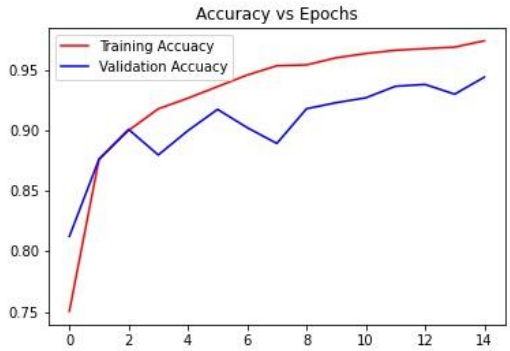

Fig. 13. *Accuracy vs Epochs 90:10 split*

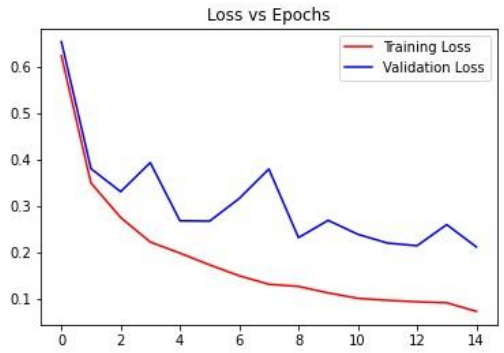

Fig. 14. *Loss vs Epochs 90:10 split*

The results would have come similar using any one of two final candidates that is 75:25 and 80:20. This paper went with 80:20 because the part of the zero slope in the 80:20 case is smoother than the 75:25 case.

4.2.2 Transfer Learning: The ResNet-50 model was trained for 30 epochs by using the 'conv5_block3_2_relu' layer output as the convolution layer and with one dense layer. Training the model gave a training accuracy of 70% but the validation accuracy of 80% (Figures 15 and 16). However, on testing this model on unseen images, the model predicted every image as paint peel-off resulting in a testing accuracy of 25%. The requirement for transfer learning to give good results is that the problem should be similar to the problem on which the model is trained. Hence, the problem is not similar to the ResNet-50

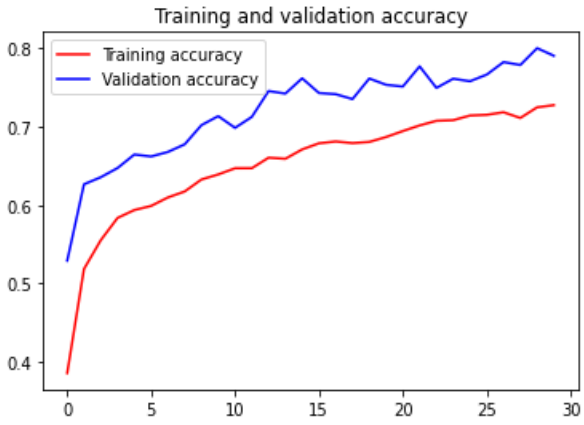

Fig. 15. *Accuracy vs Epochs -Transfer Learning*

4.2.3 Effective Defect Classifier (EDC):
Training time: The training time history is shown in Appendix 1. This table shows that as the model consists of a reasonably lower number of layers, the average time required for model training (77.6 seconds) is low too.

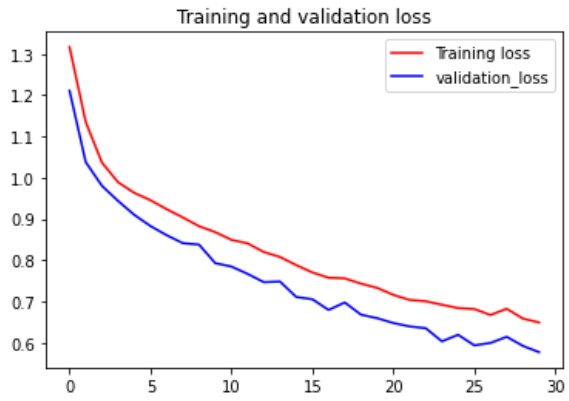

Fig. 16. *Loss vs Epochs -Transfer Learning*

Accuracy and Loss: Training the model for 15 epochs gave the following results for accuracy and loss. Figure 10 shows that the loss of the model is almost steady between epochs 2 and 14, and it will overfit if the number of epochs increases. Hence, the model is well fitted. The validation accuracy in this case is 95%, which is a great increase from earlier attempts. See Appendix 2 for the table of results.

Confusion matrix: The confusion matrix, shown in Table 2, shows that the values of true positive are very good. The classes of Crack, Dampness, Plain Wall, and Paint Peel Off are abbreviated as the letters C, D, N, and P respectively. Out of 1000 test images per class, approximately 950 images (95%) per class are identified correctly. Tables 3 to 6 mention class wise confusion matrices for each of the four classes. Here TP, FP, FN, and TN refer to True Positives, False Positives, False Negatives, and True Negatives respectively.

Table 2 Confusion Matrix

Predicted→ Actual ↓	C	D	N	P	Total
C	932	27	13	28	1000
D	16	966	8	10	1000
N	8	10	957	25	1000
P	24	11	19	946	1000

Table 3 Confusion Matrix of Crack Class

Predicted→ Actual ↓	C	Not C
C	TP = 932	FN = 68
Not C	FP = 48	TN = 2952

Table 4 Confusion Matrix of Dampness Class

Predicted→ Actual ↓	D	Not D
D	TP = 966	FN = 34
Not D	FP = 48	TN = 2952

Table 5 Confusion Matrix of Plain Wall Class

Predicted→ Actual ↓	N	Not N
N	TP = 957	FN = 43
Not N	FP = 40	TN = 2960

Table 6 Confusion Matrix of Paint Peel Off Class

Predicted→ Actual ↓	P	Not P
P	TP = 946	FN = 54
Not P	FP = 43	TN = 2957

Precision, Recall, and F1-Score: Precision measures the relative number of FP as compared to the number of TP. Higher precision indicates that the amount of negatives predicted as positive is low. Similarly, Recall measures the relative number of FN as compared to the number of TP. Higher recall indicates that the amount of positives falsely classified as negative is low. Ideally, both precision and recall should be high. F1-score unifies both of these scores. The calculation of these values for Crack class is shown below. Similarly, other classes' metrics have been calculated and all the values are filled in the Table 7.

Precision, Recall and F1-score for crack class:

Precision = TP / TP+FP = 932 / 932+48 = 0.95
Recall = TP / TP+FN = 932 / 932+68 = 0.93
F1 Score = 2 * (Recall * Precision) / (Recall + Precision)
= 2 * (0.93*0.95) / (0.93+0.95) = 0.94

Table 7 Class wise Metrics

	Precision	Recall	F1-score	Total
Crack	0.95	0.93	0.94	1000
Dampness	0.95	0.97	0.96	1000
Plain Wall	0.96	0.96	0.96	1000
Paint Peel Off	0.94	0.95	0.94	1000

Comparison between available models and Effective Defect Classifier model: Available models are binary classifiers and they only classify between a crack and non-crack. The EDC model is a multi-label classifier and it classifies crack, dampness, the plain wall and paint peel-off with an accuracy of 95%.

5. Conclusion and Future Scope

This paper has generated a dataset with four classes (Crack, Dampness, Paint peel-off, Plain wall) and each class has 9000 images. Testing this dataset using the ResNet-50 model gave a low validation accuracy. Then a custom CNN model, Effective Defect Classifier (EDC), gave a validation accuracy of 95%, which can classify more classes as compared to the existing models. Precision for crack, dampness, plain wall and paint peel-off, is 95%, 95%, 96% and 94%. Recall is 93%, 97%, 96%, 95%. F1-score is 94%, 96%, 96%, 94%.

Current dataset consists of four classes, which can be increased by identifying more varieties of defects. Current dataset has 9000 images per class. By doing more surveys, one can increase the count of images and by using a good camera; one can improve the image quality. Pre-processing

techniques can be applied on the dataset to get more features from images.

6. Acknowledgment

Veermata Jijabai Technological Institute supports this research. Our special thanks to Professor V. B. Nikam who provided guidance and helps us to get insights and expertise that greatly assisted throughout the research. We thank Assistant Professor Nilkamal P. More for assistance with deep learning technologies that helps us elevate this research further.

7. References

[1] S. Kang, C. Chun, S. Shim, S. Ryu and J. Baek, "Real Time Image Processing System for Detecting Infrastructure Damage: Crack," 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2019, pp. 1-3.

[2] J. Vora, M. Patel, S. Tanwar and S. Tyagi, "Image Processing Based Analysis of Cracks on Vertical Walls," 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, 2018, pp. 1-5.

[3] R. K. Meghana, S. Apoorva, Mohana and Y. Chitkara, "Inspection, Identification and Repair Monitoring of Cracked Concrete structure –An application of Image processing," 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 1151-1154.

[4] X. Liu, Y. Ai and S. Scherer, "Robust image-based crack detection in concrete structure using multi-scale enhancement and visual features," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, 2017, pp. 2304-2308.

[5]W. Li, M. Zhang, Z. Shen, W. Hu and P. Li, "Track Crack Detection Method in Complex Environment," 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2018, pp. 356-359.

[6] M. Lin, R. Zhou, Q. Yan and X. Xu, "Automatic Pavement Crack Detection Using HMRF-EM Algorithm," 2019 International Conference on Computer, Information and Telecommunication Systems (CITS), Beijing, China, 2019, pp. 1-5.

[7] L. Peng and J. Liu, "Detection and analysis of large-scale WT blade surface cracks based on UAV-taken images," in IET Image Processing, vol. 12, no. 11, pp. 2059-2064, 11 2018.

[8] L. S. Calderón and J. Bairán, "Crack detection in concrete elements from RGB pictures using modified line detection Kernels," 2017 Intelligent Systems Conference (IntelliSys), London, 2017, pp. 799-805.

[9] Y. Wang, Y. Lyu, Y. Cao and M. Y. Yang, "Deep Learning for Semantic Segmentation of UAV Videos," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, 2019, pp. 2459-2462.

[10] H. Shi, Z. Liu, Y. Fan, X. Wang and T. Huang, "Effective object detection from traffic camera videos," 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, 2017, pp. 1-5.

[11] P. V. Babayan, M. D. Ershov and D. Y. Erokhin, "Neural Network-Based Vehicle and Pedestrian Detection for Video Analysis System," 2019 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2019, pp. 1-5.

[12] B. Hou, J. Li, X. Zhang, S. Wang and L. Jiao, "Object Detection and Trcacking Based on Convolutional Neural Networks for High-Resolution Optical Remote Sensing Video," IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019, pp. 5433-5436.

[13] K. Mouri, H. Lu, J. K. Tan and H. Kim, "Object Detection on Video Images Based on R-FCN and GrowCut Algorithm," 2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT), Busan, 2018, pp. 1-4.

[14] B. Tian, L. Li, Y. Qu and L. Yan, "Video Object Detection for Tractability with Deep Learning Method," 2017 Fifth International Conference on Advanced Cloud and Big Data (CBD), Shanghai, 2017, pp. 397-401.

[15] M. H. Gasmallah and F. Zulkernine, "Video Predictive Object Detector," 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, 2018, pp. 365-371.

[16] A. Anjum, T. Abdullah, M. F. Tariq, Y. Baltaci and N. Antonopoulos, "Video Stream Analysis in Clouds: An Object Detection and Classification Framework for High Performance Video Analytics," in IEEE Transactions on Cloud Computing, vol. 7, no. 4, pp. 1152-1167, 1 Oct.-Dec. 2019.

[17] N. Q. Truong, P. H. Nguyen, S. H. Nam and K. R. Park, "Deep Learning-Based Super-Resolution Reconstruction and Marker Detection for Drone Landing," in IEEE Access, vol. 7, pp. 61639-61655, 2019.

[18] C. Wang, J. Wang, X. Zhang and X. Zhang, "Autonomous navigation of UAV in large-scale unknown complex environment with deep reinforcement learning," 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, 2017, pp. 858-862.

[19] S. A. Hassan, T. Rahim and S. Y. Shin, "Real-time UAV Detection based on Deep Learning Network," 2019

International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea (South), 2019, pp. 630-632.

[20] A. Zeggada, F. Melgani and Y. Bazi, "A Deep Learning Approach to UAV Image Multilabeling," in IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 5, pp. 694-698, May 2017.

[21] A.B. Mahadik and M.H. Jaiswal, "Structural Audit of Buildings," 2014 International Journal of Civil Engineering Research, 2014, ISSN 2278-3652 Volume 5, Number 4 (2014), pp. 411-416

[22] N. A. M. Yusof, M. K. Osman, M. H. M. Noor, A. Ibrahim, N. M. Tahir and N. M. Yusof, "Crack Detection and Classification in Asphalt Pavement Images using Deep Convolution Neural Network," 2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2018, pp. 227-232.

[23]S. Liang, X. Jianchun and Z. Xun, "An Extraction and Classification Algorithm for Concrete Cracks Based on Machine Vision," in IEEE Access, vol. 6, pp. 45051-45061, 2018.

[24] L. Zhang, G. Zhou, Y. Han, H. Lin and Y. Wu, "Application of Internet of Things Technology and Convolutional Neural Network Model in Bridge Crack Detection," in IEEE Access, vol. 6, pp. 39442-39451, 2018.

[25] M. David Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy and G. Morison, "A Deep Convolutional Neural Network for Semantic Pixel-Wise Segmentation of Road and Pavement Surface Cracks," 2018 26th European Signal Processing Conference (EUSIPCO), Rome, 2018, pp. 2120-2124.

[26] F. Nagata, K. Tokuno, K. Watanabe and M. K. Habib, "Design Application of Deep Convolutional Neural Network for Vision-Based Defect Inspection," 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 2018, pp. 1705-1710.

[27] X. Wang and Z. Hu, "Grid-based pavement crack analysis using deep learning," 2017 4th International Conference on Transportation Information and Safety (ICTIS), Banff, AB, 2017, pp. 917-924

[28] S. Gibb, H. M. La and S. Louis, "A Genetic Algorithm for Convolutional Network Structure Optimization for Concrete Crack Detection," 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, 2018, pp. 1-8.

[29] Nilkamal More, V. B. Nikam & Biplab Banerjee (2020): Machine learning on high performance computing for urban greenspace change detection: satellite image data fusion approach, International Journal of Image and Data Fusion, DOI: 10.1080/19479832.2020.1749142

[30] Nikam, V. B., and B. B. Meshram. "Modeling rainfall prediction using data mining method: A Bayesian approach." 2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation. IEEE, 2013.

8. Appendix

Appendix 1 Training Time History

Epochs	Training Time (sec)	Epochs	Training Time (sec)
1	80	9	77
2	80	10	77
3	78	11	77
4	78	12	77
5	78	13	77
6	77	14	77
7	77	15	77
8	77	-	-
Avg. Training Time per Epoch		77.6 seconds	
Total Training Time For 15 Epochs		19 minutes 24 seconds	

Appendix 2 Accuracy and Loss

Epochs	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss
1	70.36	71.95	0.71	0.70
2	85.69	91.30	0.39	0.35
3	89.21	92.90	0.29	0.22
4	90.88	90.68	0.25	0.30
5	92.09	92.37	0.21	0.22
6	92.60	91.15	0.19	0.28
7	93.79	93.15	0.17	0.21
8	93.74	93.48	0.17	0.23
9	95.06	94.67	0.13	0.17
10	95.12	94.35	0.13	0.24
11	95.51	93.48	0.12	0.27
12	96.16	94.23	0.10	0.20
13	96.57	95.15	0.09	0.23
14	96.41	94.92	0.10	0.24
15	97.12	95.07	0.08	0.26