**Intelligent Visual Inspection on Live Building Video Stream Data for Structural Audit**

V. B. Nikam, Nilkamal More, Aditya Shenoy, Rahul Bhoir, K. Borkar, R. Deore, Abhay Bambole

Veermata Jijabai Technological Institute, Mumbai, India

vbnikam@it.vjti.ac.in, neelkamalsurve@somaiya.edu

**Abstract:** Structural audit requires manual expertise of structural experts that can be subjective and erroneous. Deep learning techniques are 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 high performance model which can classify more than one structural defects in the live visual inspection data while doing structural audit. The proposed system classifies cracks, dampness and paint peel-off as a part of visual inspection while doing structural audit of infrastructures. Usage of a non-complex model has reduced the time taken for training the model (20 minutes) and getting predictions (9ms per prediction).

**Keywords:** Convolutional neural network, strcuctual crack, dampness, live video stream, paint peel-off, structural audit, classification, deep learning.

**1 Introduction**

Construction of different infrastructures is a necessity for human life as it brings value to the society, fulfills human needs, and also 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 researchers worked in the same domain have classified for cracks in their research work. There are other structural defects in the buildings that can be detected visually including dampness of wall and peeling off of paint. Existing crack detection models are using complex models, with a high number of hidden layers and hidden nodes, which increases the compute overhead and thus reduces the performance of the system. This lower performance makes it difficult to use these models to be deployed in a live stream classification environment unless a costly hardware setup is used for model prediction. There is a need for a high performance model which can classify multiple structural defects in the live visual inspection data while doing structural audit. The multiple label classification in real time video streaming have been addressed in this paper. The proposed system classifies cracks, dampness and paint peel-off in live streaming video as a part of visual inspection at the time of structural audit. This paper is planned as: the Literature Review section provides a critical survey of literature and related research work done in this area; the Proposed Model section describes the proposed multi-class classification model for the structural audit; and the Results and Analysis section demonstrates the results of the custom designed model. The Conclusion and Future Scope section discusses scope of research and future work which can be carried out in the related field.

**2 Literature Review**

Y. Wang et al. [1] 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. [2] research on the object detection on 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. [3] 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. [4] state a method of object detection in high resolution remote sensing videos. This paper considered two architectures, YOLO and Faster R-CNN. Faster R-CNN was selected as it produces better result for smaller sized objects which are present in remote sensing video. K. Mouri et al. [5] 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 real shape of the object using background and foreground seeding method. B. Tian et al. [6] 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. [7] 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. [8] 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. [9] propose a Super Resolution (SR) reconstruction and marker detection model based on deep learning for drone landing. In SR reconstruction, the low resolution image frame 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. [10] state an approach for map construction of the unfamiliar surrounding and using it to localize. The model includes Deep Reinforcement Learning (DRL), Reinforcement Learning (RL), Markov Decision Processes (MDP), 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. [11] 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 center, then the duty of the grid is to detect object, forecast B’s bounding boxes, calculate bounding box x, y, z, h confident, predict conditional probability, determine bounding box confidence score, otherwise undefined body. YOLOv2 is 92.10% precise and YOLOv3 is 95.20% which is enchanced, but YOLOv3 took longer training time. A. Zeggada et al. [12] state a multi-label classification model for UAV images. In this model, CNN features extracted from tile using GoogLeNet. For multi-labeling issue, the paper proposed mutlilabel layer. A. Mahadik et al. [13] state structural audit method for building. It is necessary and important to carry out the structural audit specialists and act instantly through advices provided in the audit report to ensure safety of the building.

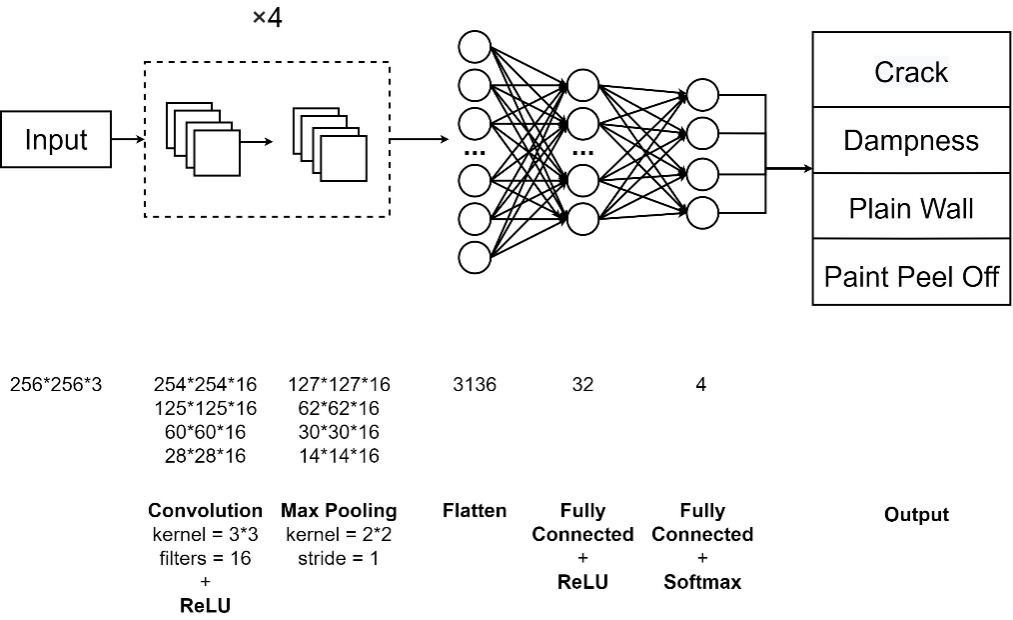
N. Yusof et al. [14] propose a method to classify and detect pavement crack using deep CNN. This study states a deep convolution neural network (DCNN) that is capable to sense the crack robustly while dealing with complicated background image. The network is trained using stochastic gradient (SGD) training 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. [15] 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 threshold gap to get the crack binarization segementation result. L. Zhang et al. [16] 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 high risk factor in domestic and low fracture analysis. M. David Jenkins et al. [17] represent a DFCN network to achieve 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. [18] 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. [19] research on the segment of roadway crack images into variable scales of nets and then 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. [20] 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 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.

**3 Proposed Solution**

The proposed system classifies cracks, dampness and paint peel-off in live streaming video as part of visual inspection during structural audit. As the structural audit domain is not explored much before, this paper is using a custom Convolutional Neural Network (CNN) for intelligent visual inspection.

**3.1 Custom CNN Architecture**

Wall defect classification problem involves detection of complex features in the image data. For the architecture, hyper-parameters, like convolutional layer count, filter size in each convolutional layer and dense layer count, were tuned. The final architecture obtained after various trials and errors is illustrated in Figure 1.

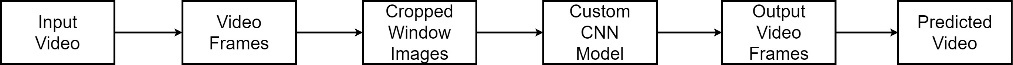


**Fig. 1.**Custom CNN Architecture

This CNN model consists of four layers of convolution with 16 filters each and a kernel size of 3\*3. The model performs Max Pooling of 2\*2 after each convolutional layer. The convolution layers follow a hidden Dense layer before the output layer containing 32 nodes. The input layer matches the dimensions of the images in our 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 consists of four classes. The dataset consists of 36,000 images (32,000 training images and 4,000 validating images).

**3.2 Sliding Window Mechanism**

As the CNN model works on image input, there should be a system to extract frames from images and then use the model for predictions. Figure 2 shows the block diagram of this system.



**Fig. 2.**Block Diagram for Sliding Window System

This system splits the input video into frames. Then each frame is cropped in a window to match the input dimension of the model. The model predicts the classes and the system then generates the predicted video.

This paper uses a sliding window mechanism to generate the predictions by the custom CNN model and visualize the prediction output. This sliding window mechanism is described in Algorithm 1.

|  |
| --- |
| **Algorithm 1.** Sliding Window |
| /\*Convolves upon the frames of the input video and uses the model to do prediction and generates an output video for visualizing the model predictions\*/  **input:** input\_video  **output:** output\_video  **begin**  /\*Loop until the live stream stops or the end of the file has reached in case of non-live video\*/  **while** input\_video.has\_more\_frames() **do**  /\*Read the next frame\*/  frame = input\_video.read\_frame()  /\*Round down frame dimensions to make them multiples of 256\*/  width, height = frame.get\_dimensions()  new\_width, new\_height = floor(width / 256) \* 256, floor(height / 256) \* 256  resized\_frame = frame.resize(new\_width, new\_height)  /\*Create a result frame which is double the shorter dimension of the resized\_frame and same longer dimension\*/  **if** new\_width < new\_height **do**  result\_frame = create\_empty\_image(2 \* new\_width, new\_height)  **else do**  result\_frame = create\_empty\_image(new\_width, 2 \* new\_height)  **end**  /\*Paste the resized frame at top left at (0, 0) coordinate of the result frame\*/  result\_frame.paste(resized\_frame, (0, 0))  /\*Loop for all rows and columnsof the sliding window\*/  **for** y = 0 **to** new\_height **incrementing** 256 **do**  **for** y = 0 **to** new\_width **incrementing** 256 **do**  /\*Crop the image in the current sliding window coordinate (x, y) of size (256, 256)\*/  cropped\_image = resized\_frame.crop(x, y, 256, 256)  /\*Pass the cropped image to the model for prediction\*/  prediction = model.predict(cropped\_image)  /\*Paste the color coded visual grid according to prediction in the corresponding (x, y) coordinate in empty part of the result frame\*/  result\_frame.paste\_corresponding\_visual(x, y, prediction)  **end**  **end**  /\*Add the result frame to the output video\*/  output\_video.add(result\_frame)  **end**  **end**  **return** output\_video |

The algorithm:1 works on live video input as well as video input stored on disk. The program loops until all the frames are processed. For each frame, the dimensions of the image are rounded down to values which are multiples of 256 as the dimensions of the custom CNN model are 256\*256. A new result image is created. The program then pastes the resized image at the top or the left side of the result image depending on the orientation of the image and the remaining half of the image is for the model predictions generated by the program.

The frames are divided into a 256\*256 grid. These grid images are passed to the model to predict what class the window belongs to. The color coded prediction output is pasted to the corresponding part of the remaining half of the result image. Once the entire grid is processed, the frame is completely processed and the result image can be saved as a frame in the output video.

**4 Result and Analysis**

This section discusses the characteristics of the custom dataset used. This section also analyses the performance results of the custom CNN model.

**4.1 Training Time**

Training time is an important metric as this paper is looking for a high performance model. The training time history is shown in Table 1.

The table shows that the average time required for model training (77.6 seconds) is low. This low average training time is because the model consists of a reasonably lower number of hidden layers and hidden nodes in these layers.

**Table 1.**Training Time History

|  |  |  |  |
| --- | --- | --- | --- |
| **Epochs** | **Training Time (sec)** | **Epoch** | **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 | - | - |
| **Total Training Time for 15 Epochs** | | 19 minutes 24 seconds | |

**4.2 Output Format**

The output format is color coded and shown adjacent to the input video for better visualization. Figures 3-6 shows some frames of the output videos that are generated on test videos.

|  |  |
| --- | --- |
| **https://lh6.googleusercontent.com/cEDCX50gpLVo6DxLswirby3sYnI95CWXxnROXMgP_3IjxvMG5eOFA5UN-PgO9h9gnNmKmtze1uVOOUkrkNLZQA6efpyj_kCcmwVaILYJhmvD0feccaNnmsnNzqq6FjJ-KDqp1qd1** | **https://lh4.googleusercontent.com/1zwHrYYdpEMeEQIKaIwQjLm0zXuUZb6Jp8fnaMS5-HsstlqgboJ2NFp22SkIB16EDUwZyBIvIOQI5kPCqeLnrXB0HfHwgHOrFk_7gXrj5HEh4aHP0tsShIPtFbDwSfZfSggaZCpy** |
| **Fig. 3.**Output video frame 1 | **Fig. 4.**Output video frame 2 |

|  |  |
| --- | --- |
| **https://lh3.googleusercontent.com/FogaJdKrWNFihNfXcspbhgIp2Ho9vhEJdyBjuZpNiJxBud4ziGbTCeeKk0Z4D4bm8iDp7gQPbLnLxf3Uow5UiaKP4MVUHC93vw6UUGYkdDJEg4qfEEGuzYhx_zXqkTDK9s152Glq** | **https://lh4.googleusercontent.com/aUeVBURUPuAx8TRZRy0A9N0fp5faTV9SUDYIFsKe9HN7Lqxtmtn8uLzZteWh2j54eMB-h_ZjZgQLFaN-dforkRuKhs3B5PpiFNULqfSLvQQfVzKhjNZHGE7nK0buSrNB_bu6MsNK** |
| **Fig. 5.** Output video frame 3 | **Fig. 6.**Output video frame 4 |

The frames show the regions where the model predicts that a certain class belongs in the corresponding region in the input video. Here the color code used is: red for crack, green for dampness, yellow for paint peel off, and blue for plain wall.

**4.3 Prediction Time**

When the program was run on a NVIDIA GeForce 940MX Graphic Card powered laptop, prediction took 9ms per (256\*256) image and 10 such predictions are made per frame for 720p videos leading to 90ms per frame. The program skips frames if the incoming rate of frames is higher, but this can be easily solved if high end hardware is used instead.

**5 Conclusion and Future Scope**

This paper has used a dataset with four classes (Crack, Dampness, Paint peel-off, Plain wall) and each class has 9,000 images. The custom CNN model was able to train in 20 minutes which can classify more classes as compared to the existing models. The model is able to do a single prediction in 9ms on a commodity hardware.

Current dataset consists of four classes which can be increased by identifying more varieties of defects. Current dataset has 9,000 images per class. Doing more surveys can increase the count of images. Using a good camera can improve the image quality. Using Principal Component Analysis can reduce the features of the data. This reduction in features will further reduce the training and prediction time of the model.

**References**

1. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. 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
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. 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
20. 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.
21. 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
22. 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.