2020 International Conference on Computational Performance Evaluation (ComPE),
North-Eastern Hill University, Shillong, Meghalaya, India. July 2–4, 2020

Inspection of Concrete Structures by a Computer Vision Technique and an Unmanned Aerial Vehicle

Alokita Karmokar
Department of Electronics and Telecommunication,
Rajiv Gandhi Institute of Technology,
Mumbai, Maharashtra-400053
karmokaralokita@gmail.com

Nikhil Jani
Department of Electronics and Telecommunication,
Rajiv Gandhi Institute of Technology,
Mumbai, Maharashtra-400053
nikhiljani.38@gmail.com

Anushka Kalla
Department of Electronics and Telecommunication,
Rajiv Gandhi Institute of Technology,
Mumbai, Maharashtra-400053
anushka4kalla@gmail.com

Harsh Harlalka
Department of Electronics and Telecommunication,
Rajiv Gandhi Institute of Technology,
Mumbai, Maharashtra-400053
harlalka.harsh36@gmail.com

Poonam Sonar
Department of Electronics and Telecommunication,
Rajiv Gandhi Institute of Technology,
Mumbai, Maharashtra-400053
poonam.sonar@mctrgit.ac.in

***Abstract*—We have proposed a visual inspection technique for concrete structures using deep learning and a hardware ecosystem, an Unmanned Aerial Vehicle (UAV). The UAV is a quadcopter that can fly to unreachable sections of a site which consists of a camera that captures images of the concrete surfaces via a mobile device and feed the real time images in the CNN model. The images taken from such remote locations may contain different types of surfaces, shadowed regions and surfaces with holes. The cracks are properly detected by the CNN 'AlexNet' algorithm and masking with sliding window technique in such conditions due to variation in the image data set. The experimental results were simulated on a standard online data set of 40,000 images of Mendeley Data which is freely available and 3000 images have been chosen from the entire data set for this method. The classes have been divided into 2 categories of 'crack' and 'no crack' for the proposed method's data set. There are 1050 training images and 450 testing images for each category. Experimental results were achieved on Google Colab cloud service using Python Tensorflow API (Application Programming Interface). The proposed 'AlexNet' CNN algorithm achieves 98.4 % accuracy and the model is deployed to a masking technique with sliding window to detect cracks in a 3008x2000 pixel resolution image by breaking the image into 227x227 pixel resolution image patches. The experimental results have proved that the proposed method handles noisy background such as cracks with shadows and stains, cracks on rusty and rough surfaces and minor dimension cracks with good efficiency.**

***Index Terms*—Machine Learning, Deep Learning, Computer Vision, Convolutional Neural Network, AlexNet, Unmanned Aerial Vehicle(UAV), Image Processing, Drone, Crack Detection, Construction**

I. Introduction

Construction workers and engineers have to work in hazardous conditions in order to maintain various structures which put their lives at stake. To carry out a non-destructive inspection and to avoid any mishaps due to dilapidated conditions in residential, commercial buildings, bridges, highways and roads, this method proposes a computer vision technique in which we use a Convolutional Neural Network (CNN) called "AlexNet" with an Unmanned Aerial Vehicle (UAV). Drone simply alludes to any airborne vehicle that's unmanned, that is, the pilot does not sit inside the vehicle itself. UAV's have many applications which incorporate Review, Mapping, Agriculture, Emergency Administrations, Mining, Surveying, Development and Search & Rescue. A UAV will save time and it will help in detecting cracks non-destructively. Crack inspections of structures have been done visually but huge and tall towers and buildings have some areas which are inaccessible to humans and hence provide us with limitations for crack inspections. The study provides a brief idea about the upcoming technology which is used for investigating cracks in places that are inaccessible to humans. It combines Computer Vision and a UAV. The results are of good accuracy and hence suitable for crack detection without any manual inspection assisting automation in civil engineering. The purpose of this method is to reduce accidents during construction work and to improve the already existing crack image processing and detection techniques.

Manual crack detection needs trained reviewers and is subjective from an individual to another individual. To make the detection uniform numerous image processing techniques were developed to identify concrete cracks. Cracks can be detected from simple image processing techniques which includes threshold and histogram concepts [1] [2]. Further for enhancing the detection procedures like, Fast Haar transform (FHT), Fast Fourier transform (FFT), Sobel, and Canny edge detectors were utilized [3] [4]. These image processing techniques give good results but lack confidence in the outputs when faced with different background noises such as the presence of shadows, stains, lights, rust, etc. To overcome the disadvantages of image processing techniques the researchers adopted some machine learning algorithms. Artificial Neural Networks (ANN's) and Support Vector Machine (SVM) [5] [6] were adopted for detection of cracks and other dilapidated conditions in concrete structures in which concrete crack features are first extracted using image processing techniques mentioned above and then put through the machine learning algorithms. As a result these algorithms rely heavily on extracted crack features and are affected by false feature extraction. Hence a Convolutional Neural Network has been suggested to avoid false concrete crack feature extractions. CNN's tend to learn features in images with comparatively lesser parameters and computations because of less number of connections between the neurons [7].

The organization of paper is described as follows. Section II explains the methodology of software and hardware used for the proposed method. In Section III, the AlexNet architecture , its parameters, the data set, the masking technique and the final results of the CNN model have been elaborated. Section IV explains all the hardware calculations and firmware used for the UAV while section V is the conclusion.

II. Methodology

A. The proposed Classifier for Concrete Crack Detection.

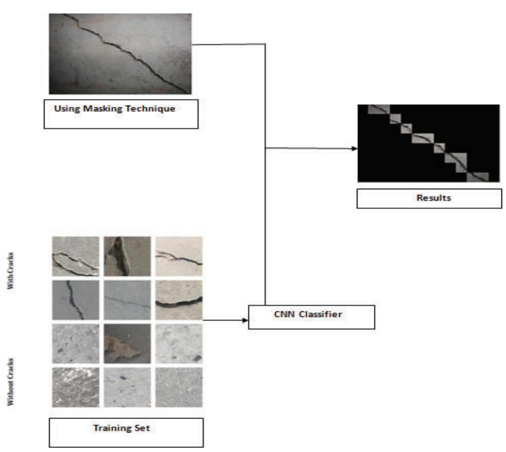

Figure 1: Flow diagram of the proposed Classifier for Inspection of Concrete Structures using Deep Learning.

The images are put through a convolutional neural network named "AlexNet" after the designer Alex Krizhevsky [8]. The flow diagram of the convolutional neural network is shown in Figure 1. The image data set is segregated into two sections namely "Training Data Set" through which the network will get trained as it includes both the data and the outcomes and compute the cost using the appropriate loss function [7]. The second part is named "Testing Data Set" which will help to estimate the model's properties as it includes only the input data and not the corresponding outputs. There is no dependency of test data on the training data and hence if after testing it follows the same probability distribution as the training data the model is said to have minimal over fitting. Then the real time image from the UAV is processed by the model and also goes through masking and sliding window technique to give the final detected results.

B. The proposed UAV model.

The hardware system has the ground station which consists of a laptop/PC which will obtain various flight parameters of the drone, primarily its longitude, latitude and altitude for easy control with the help of the open source flight controller "Ardupilot's Mission Planner". The same flight controller will be used to burn the firmware for pixhawk 2.4.8 and also will be used to set waypoints for an autonomous flight of the drone. Alongside that, there is a transmitter remote control "FSi6" which consists of needed toggle switches to maneuver the drone. The drone consists of a PPM trans receiver named "FS-iA6". The transmitter transmits the signal to the transceiver unit on the drone which is interfaced directly to Pixhawk 2.4.8 controller which is the Brain of the drone/quad-copter. According to command, the controller varies the speed of the four motors via the four ESC's(Electronic Speed Controller) to make the changes to Yaw, Pitch & Roll (the three axis in which the drone can be moved in angular direction)and the usual 3- dimensions of X, Y & Z axes. The controller takes in 5V from a 3300mAh Li-Po battery to powers it. The ESC's are used between the controller and motor to set the direction of motors (clockwise for two opposite motors and anti-clockwise for other two) and also for dynamic breaking and swift speeding. Another important component is the (GPS + Compass) module which actually contacts with the satellites in real time and helps us locate or move the drone in all three dimensions (longitude, latitude & altitude). The camera is attached at the bottom of the UAV. Fig 2 shows the Block Diagram of our proposed UAV.

III. The AlexNet Architecture

The proposed method utilizes the famous CNN algorithm called 'AlexNet' [8]. The only change in this algorithm is that the number of outputs is changed to 2 instead of the 1000 classes in the original paper. The two classes are 'cracks' and 'no cracks'. Figure 3 shows the parameters used for the proposed method. Table I presents the detailed specifications of the CNN.

A. Cost Function and Adam (Adaptive Moment Estimation) learning rate optimizer.

A cost function is a quantified output that tells what quantity deviated the model is from the particular values to be predicted. Typically these are calculated as the squares difference between the presently calculable probability and

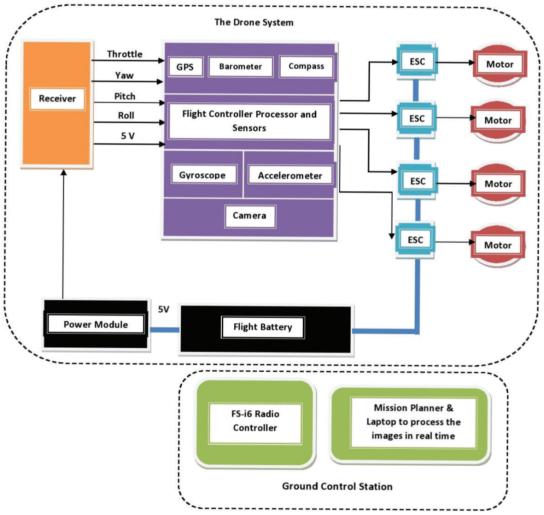

Figure 2: Block Diagram of the proposed UAV.

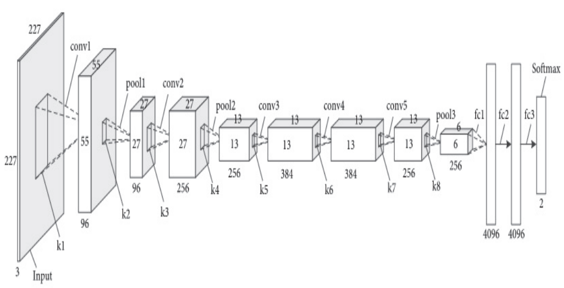

Figure 3: Overall architecture of CNN where conv = convolution layer; pool = pooling layer; fc=fully connected layer; K = kernel of each operation. [7]

the correct probability. The performance of a classification model is measured by its Cross-entropy which is a probability value between zero and one. Cross-entropy loss will increase because the expected probability diverges from the particular label. For our application as there are two categories the formula used is:

Table I: Specifications of the AlexNet. [7]

Layer	Kernel Size	Stride	Pad	Num Output
conv1	11	4	0	96
pool1	3	2	0	96
conv2	5	1	2	256
pool2	3	2	0	256
conv3	3	1	1	384
conv4	3	1	1	384
conv5	3	1	1	256
pool3	3	2	0	256
Fc1	-	-	-	4096
Fc2	-	-	-	4096
Fc3	-	-	-	2
Softmax	-	-	-	2

$$cross\ entropy = -(y(log(p) + (1-y)log(1-p)) \quad (1)$$

where y is binary indicator (0 or 1) if class label is the correct classification for observation & p is the predicted probability observation o is of the same class. This cost function is then fed back to the network using the concept of back propagation.

The back propagation is done with the help of gradient. Negative of Gradients in calculus are used to find the minima of a function. The Adam (Adaptive Moment Estimation) [9] optimizer will help to constantly redefine the weights and biases to bring the predicted values to the actual wanted values. The Adam optimizer is the most advanced and the newest optimizer in practice as it is very fast and efficient. It is based on the AdaGrad and RMSprop algorithm's adaptive learning rates and the momentum algorithm. The RMSprop equation is enhanced in the Adam optimizer by multiplying it with the momentum. The learning rates are adaptive in certain intervals of epoch cycle so that the weights converge faster. The momentum added in the equation will again help us to reach the global minima at a faster rate. The equation for the Adam optimizer is given below from the paper.

$$\Delta w_i(t) = \frac{-\eta}{\sqrt{G_i(t)} + \epsilon} M_i(t) \quad (2)$$

where

$$M_i(t) = \alpha M_i(t-1) + (1-\alpha)\frac{\partial L}{\partial w_i}(t) \quad (3)$$

here in equation 2, $\Delta w_i(t)$ is the change in the weight with index i at time t, eta is the step size, $G_i(t)$ is the gradient computed at that time, ϵ is some small number to make sure the denominator is not zero, $\frac{\partial L}{\partial w_i}(t)$ is the derivative of the loss function with respect to the weight with index i. $G_i(t)$ is given by the formula $G_i(t) = G_i(t-1) + (\frac{\partial L}{\partial w_i}(t))^2$. As t keeps on increasing the value of $G_i(t)$ keeps on increasing, and as it is in the denominator of equation 2, the weights will start decreasing and converge into a proper value at a faster rate. $M_i(t)$ is the momentum which is expressed in equation 3. α is a hyperparameter which is set to 0.9 mostly & $M_i(t-1)$is the momentum computed in the previous step. The CNN repeats the above mentioned method repeatedly till equation 2 converges. Throughout the training of CNN , the quantity of used images in every update is the batch size. The complete update of a batch size is an iteration, and the complete update of the entire data set is an epoch cycle.

B. Data Set

The proposed method utilizes the images from a standard online data set of 40,000 images, "Mendeley Data-Concrete Crack Images for Classification by Çağlar Fırat Özgenel from Engineering, Middle East Technical University" [10]. 3000 images have been chosen from the entire data set for this paper. The classes of 'crack' and 'no crack' have been divided into 1050 training images and 450 testing images each. The batch size selected is 64 for 1500 epoch cycles. Table II

Table II: Segregation of the Data Set.

Image Class	Number of Training Images	Number of Testing Images
Crack Images	1050	450
Non-Crack Images	1050	450

shows the segregation of the data set. The iterations and masking technique were performed on Google Colab's IPython Notebook which provides 12 Gb of GPU for 12 hours at a stretch using the very well documented python API named Tensorflow. Tensorflow is a free and open source library used for data science. The library is very efficient and provides us with it's built-in layer functions for neural networks.

C. *Masking and Derived Measurements of Test.*

The CNN classifier proposed in this paper is tested on new images to verify its robustness.Masking technique is employed here to showcase areas with positive cracks within the image. As shown in Figure 4, a large image with 3008×2000 pixel resolutions is scanned by breaking the image into sizes of 227×227 pixel resolutions. Each broken pixel is passed through the model and the no crack portions get masked by black pixels. The entire image is then stored showing only the cracks in the image.

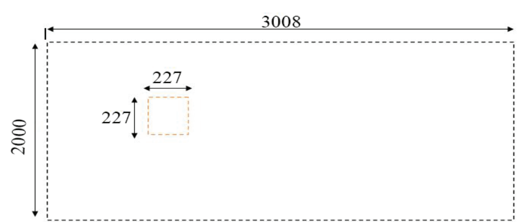

Figure 4: Process of breaking a high resolution 3008x2000 pixels image into 227x227 pixels for Crack Detection.

The regions with actual cracks in the new camera images are defined as positive regions and those with no cracks have been labeled as negative regions.The positive and negative regions that are detected and classified correctly by the CNN model are known as true-positive (TP) and true-negative (TN) regions, else false-positive (FP) and false-negative (FN) regions, severally. The accuracy came out to be 98.4 %. The confusion matrix is shown in Figure 5. Also other derived measurements of test results such as precision, sensitivity, specificity and the F1-score have been listed in Table III of below. Figure 6 shows testing results in various real life situations. The false-positive and false-negative regions have been highlighted too. The pictures are examined for broken surfaces, shadowed, rough, and rusty surface, are used for testing. So these pictures prove that the CNN model is robust in real world for various conditions.

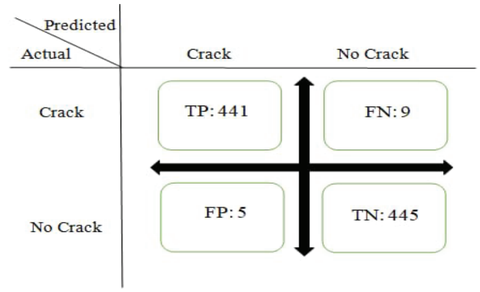

Figure 5: Confusion matrix showcasing TP=441, TN=445, FP=5 & FN=9 by our proposed method.

Table III: Test Results of the proposed method.

Parameter	Formula	Result (%)
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	98.40
Precision / Sensitivity	$\frac{TP}{TP+FP}$	98.87
Recall	$\frac{TP}{TP+FN}$	98.00
Specificity	$\frac{TN}{FP+TN}$	98.89
F1 Score	$2 * \frac{Precision*Recall}{Precision+Recall}$	98.43

IV. Hardware & Software Components for the UAV.

A. *Hardware.*

For our proposed method we have used a Quadcopter for the UAV. The reason being, it is sufficient to lift the amount of weight we want to carry including the camera and all the modules. Hexacopter could have been used but then it would increase the cost to benefit ratio.

1) Quadcopter Frame.: We are using an X configuration for the Drone. For our UAV we have chosen the cheapest and one of the strongest materials available that is glass-fiber.Fiberglass has good reinforcing property with a Young's Modulus in the range from 72 GPa to 84.7 GPa and is incredibly light-weight. We are using a Q-450 Frame for our proposed method.

2) Motors, ESC's and Propellers: Motor: A brushless DC motor (BLDC) has a rotor in the form of a permanent stator in the form of poly-phase armature windings. It differs from conventional DC motor in such a way that it doesn't contain brushes and the communication is done using electronic drive using stator wings. BLDC motors are interfaced to the controller using (Electronic Speed Controllers) ESC's which will control the speed of motors. The command is give by the remote control to the receiver that is connected on-board [11].

Motor calculations [11]: The thrust to weight relationship given below should be satisfied for the motors to provide proper lift capability.

$$Ratio = \frac{Thrust}{Weight} = \frac{ma}{mg} = \frac{a}{g} \tag{4}$$

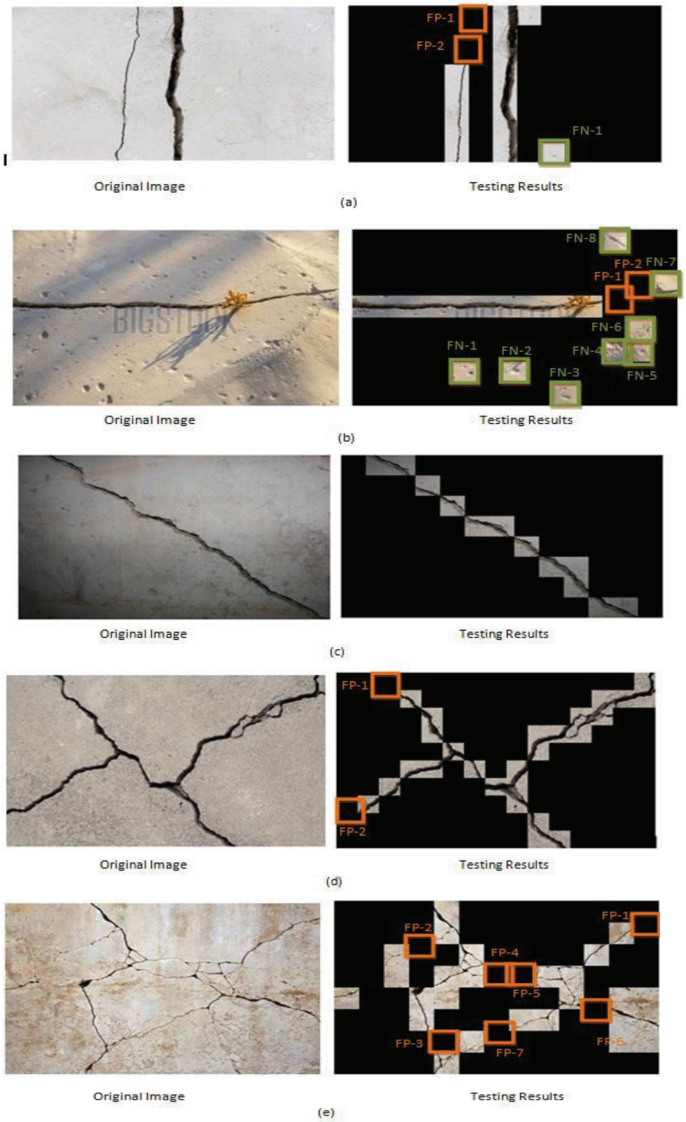

Figure 6: Damaged surfaces: (a) different thicknesses (minor and major cracks); (b) surface with holes and shadows; (c) shadowed surface; (d) rough surface; (e) rusty surface

for the quadcopter to lift off the above mentioned ratio should be greater than 1. In this case, we have assumed that,

$$TotalThrust = 2 * (Total\ weight\ of\ Quadcopter) \quad (5)$$

$$Thrust\ provided\ by\ each\ motor\ = \frac{Total\ Thrust}{4} \quad (6)$$

ESC(Electronic Speed Controller): As the Micro Controller provides Low Voltage and Low Current signal, and it isn't enough to drive the motors. The ESC's do the job of supplying the desired quantity of voltage and current to power the motors directly via the power distribution board to the motors.

ESC Calculation [11]:

$$ESC\ rating = [1.2to\ 1.5] * (max.\ ampere\ rating\ of\ motor) \quad (7)$$

Propellers:It is a type of fan that can convert rotational motion into thrust. Their classification is done in terms of diameter and pitch, represented as the product of its diameter and pitch. For example: 10*4.5 where 10 inches is the diameter and the 4.5 inches is the pitch.

3) Battery: A new type called as "Li-Po" (Lithium Polymer) batteries have been used. They provide longer run times compared to traditional Li-ion batteries.

Battery Calculation [11]:

$Max.\ Current\ withdrawal\ by\ motors = no.\ of\ motors * maximum\ current\ withdrawal\ by\ single\ motor\ (assumed\ to\ be\ 15A)$

$$= 4 * 15 = 60A \quad (8)$$

Based on the above calculation and our need to extend the flight time we have selected 11.1 V(3 cells), 3300mAh battery, 1000KV motors, 30 A ESC's & 10 inches propellers.

4) Flight Controller Pixhawk: A controller is necessary to control the motors for movement in all the three linear and three rotational axes. It is considered as the brain of the UAV. Pixhwak 2.4.8 has been used as it provides the highest stability, and a backup processor among its peers.

Detailed specifications about the flight controller can be found here [12].

5) Remote control and Transceiver: Flysky's Fsi6 is choosen as our transmitter and Fs-iA6 is the receiver. Both consists of 6 channels and in receiver these channels can be extended to 10 channels if an upgraded version of transmitter firmware is used.

6) GPS module with Compass: A GPS module is required to get UAV in a fixed position. That is, the drone in perfect longitude, latitude and altitude. The compass is required to control the direction of the UAV. We have used Ublox 7M GPS module.

7) Camera to shoot real-time crack images.: A camera is required to capture real-time crack images. The live feed is obtained by connecting a Wi-Fi camera with a laptop's network. The camera that we have been used is EKEN-H9R Wi-Fi Camera. An application named EZ-iCam can be used to click the images on a mobile device. The captured images are stored on cloud platforms and also images can be stored in the SD card provided with the camera.

The UAV implementation for our proposed method is shown in Figure 7.

B. ArduPilot Firmware.

It is an open source autopilot software. It can control any vehicle system due to its advanced features. Some of the vehicles it controls are airplanes, multirotors, boats and even submarines. The software as is an open source project is installed worldwide and has been tested for its ability and proven to be very good for hobbyists and project teams across the globe. It provides basic programming for a UAV free of cost and is editable according to the users application.

Figure 7: UAV implementation of the proposed method.

1) The Firmware: The Firmware from and Ardupilot is used to program the flight controller (PixHawk 2.4.8). All the voltage commands and steps are included in the program of the firmware. The firmware that was burnt onto the controller was according to our specifications of UAV.

2) Calibration and Failsafes of UAV components using Mission Planner.: The Accelerometer, Compass, GPS, ESC's, Radio Controller can all be calibrated and set to their desired threshold values using Mission Planner software. Calibration is necessary in order to reduce uncertainty in any measurement of the components [12]. Failsafe mechanisms have been used to ensure easy recovery of drone and mishaps in case of any component failures. Our proposed method uses two failsafe mechanisms namely "Return to Launch" which makes the UAV return to it's launch location if triggered and "Land" mode will make the UAV land itself on the site where its vertical projection would appear.

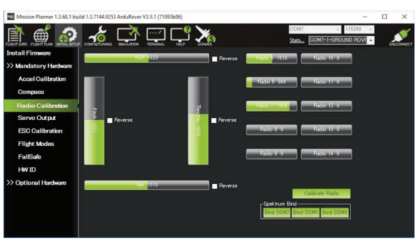

Figure 8: Calibrating the Radio Controller using Mission Planner. [12]

3) Flight Modes.: The Documentation has a total of 20 flight modes of which 10 modes are used extensively.Different types of stabilization modes for smooth flights ,autonomously flight plans and safety options are included in these modes. The radio controller is used to select the modes by the transmitter switches present on the device [12]. In our proposed method the Loiter and Altitude Hold modes have been used mainly to keep the UAV stable during flight. The Loiter mode keeps the drone stable in 3 axis that is it maintains the current location, heading and altitude. The Altitude Hold mode is used to maintain it's altitude which helps us in stabilizing the UAV to it's desired height.

V. Conclusion

This method proposed a computer vision technique called "Convolutional Neural Network" and integrated it with a hardware system "Unmanned Aerial Vehicle" for visual inspection of concrete cracks. When the images are captured from remote locations by the UAV, it might contain several types of background noises. These are handled very well by the 'AlexNet' CNN algorithm used in our proposed method, which results in 98.4%.The experimental results also prove its efficiency in different background noises to be very robust due to variety in the data set. The technique of 3008x2000 resolution pixel images being masked by a 227x227 resolution pixel patches prove to be very efficient. The novelty of the proposed method is that minor dimensions of cracks, cracks with stains, cracks on rusty and rough surfaces can be detected with very good efficiency and accuracy as proven experimentally. The same can be used to detect cracks in pipes and scaling of concrete structures. In future, the proposed method can be trained on a huge data set with different abnormalities and variations in cracks of different structures. Also increasing the wireless capabilities of the camera installed on the UAV will help to capture and send the images over longer distances and evaluate the images in real time.

References

[1] H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *2009 17th European Signal Processing Conference*. IEEE, 2009, pp. 622–626.

[2] K. Kirschke and S. Velinsky, "Histogram-based approach for automated pavement-crack sensing," *Journal of Transportation Engineering*, vol. 118, no. 5, pp. 700–710, 1992.

[3] B. Santhi, G. Krishnamurthy, S. Siddharth, and P. Ramakrishnan, "Automatic detection of cracks in pavements using edge detection operator," *Journal of Theoretical and Applied Information Technology*, vol. 36, no. 2, pp. 199–205, 2012.

[4] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *Journal of Computing in Civil Engineering*, vol. 17, no. 4, pp. 255–263, 2003.

[5] S.-W. Liu, J. H. Huang, J.-C. Sung, and C. Lee, "Detection of cracks using neural networks and computational mechanics," *Computer methods in applied mechanics and engineering*, vol. 191, no. 25-26, pp. 2831–2845, 2002.

[6] M. S. Kaseko, Z.-P. Lo, and S. G. Ritchie, "Comparison of traditional and neural classifiers for pavement-crack detection," *Journal of transportation engineering*, vol. 120, no. 4, pp. 552–569, 1994.

[7] S. Li and X. Zhao, "Image-based concrete crack detection using convolutional neural network and exhaustive search technique," *Advances in Civil Engineering*, vol. 2019, 2019.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] Ç. F. Özgenel and A. G. Sorguç, "Performance comparison of pretrained convolutional neural networks on crack detection in buildings," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 35. IAARC Publications, 2018, pp. 1–8.

[11] A. Javir, K. Pawar, S. Dhudum, N. Patale, and S. Patil, "Design, analysis and fabrication of quadcopter," *Journal of Advance Research in Mechanical & Civil Engineering (ISSN: 2208-2379)*, vol. 2, no. 3, pp. 16–27, 2015.

[12] "Open source drone software. versatile, trusted, open. ardupilot." https://ardupilot.org/, (Accessed on 08/11/2020).