CAPSTONE PROJECT PHASE-II REPORT

on

**Underwater Crack Detection**

Submitted By

**Amogh Wyawahare(1032180681)**

**Vaishnavi Tiwari(1032180651)**

Under the Guidance of

**Prof. Vrushali Pagire**



**School of Electronics & Communication Engineering**

**Dr. Vishwanath Karad**

**MIT WORLD PEACE UNIVERSITY, PUNE.**

**[2021-2022]**

Text

Description automatically generated

**SCHOOL OF**

**ELECTRONICS & COMMUNICATIONENGINEERING**

Academic Year 2021\_-2022\_

**CERTIFICATE**

###### This is to certify that B. Tech Capstone Project Report entitled

**UNDERWATER CRACK DETECTION**

Work has been carried out successfully by

###### **Vaishnavi Tiwari (1032180651)**

**Amogh Wyawahare (1032180681)**

###### during the academic year 2021-22 towards the partial fulfilment of degree of **Bachelor of Technology in Electronics & Communication Engineering** as per the guidelines prescribed by Dr. Vishwanath Karad MIT World Peace University, Pune.

Project Guide Batch Coordinator Project Coordinator Head of School Prof. Vrushali Pagire Dr. Apurva Naik Dr. Alka Barhatte Dr. Vinaya Gohokar

Date:

Place: Pune

Text

Description automatically generated

**DECLARATION**

We the undersigned, declare that the work carried under Capstone Project Phase-II entitled **“**Underwater Crack Detection”represents our idea in our own words. We have adequately cited and referenced the original sources where other ideas or words have been included. We also declare that We have adhered to all principles of academic honesty and integrity and have not misprinted or fabricated or falsified any ideas/data/fact/source in my/our submission. We understand that any violation of the above will be cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or whom proper permission has not been taken when needed.

Date: 26th may 2022

Place: MIT WPU Pune

|  |  |  |
| --- | --- | --- |
| **PRN Number** | **Name of student** | **Signature with date** |
| 1032180681 | Amogh wyawahare | **sign A**  Date: 26th may 2022 |
| 1032180651 | Vaishnavi Tiwari | **sign v**  Date: 26th may 2022 |

Date: 26th may 2022

Place: MIT WPU Pune

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Abstract** | | 6 |
| **List of Figures** | | 5 |
| **Abbreviations** | |  |
| **1.** | **Introduction** | 7 |
| **2.** | **Review of Literature** | 8 |
| 2.1 | Literature Review |  |
| 2.2 | Aim and Objectives of project |  |
| **3.** | **System Development** | 9 |
| 3.1 | System block diagram |  |
| 3.2 | System Specifications |  |
| 3.3 | Challenges Faced/Complexities involved |  |
| **4.** | **System Implementation** | 11 |
| 4.1 | System Design and Description |  |
| 4.2 | Flow chart/ Algorithm implemented |  |
| **5.** | **Results and Analysis** | 18 |
| 5.1 | Results of Implementations |  |
| 5.2 | Analysis of Results |  |
| **6.** | Conclusion | 20 |
| **7.** | **References** | 21 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| Fig 1 | System block diagram | | 9 |
| Fig 2 | 2-layer CNN Model | | 11 |
| Fig 3 | Architecture of 2-layered CNN model | | 12 |
| Fig 4 | Architecture of VGG16 | | 13 |
| Fig 5 | Architecture map of VGG-16 | | 14 |
| Fig 6 | | Architecture of 5-layered CNN model | 14 |
| Fig 7 | Steps of Program | | 15 |
| Fig 8 | Data Annotation | | 17 |
| Fig 9 | Classified images with and without cracks | | 18 |
| Fig 10 | Crack Detection | | 19 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Table 1 | System specification | 5 |
| Table 2 | Result | 11 |

**ABSTRACT**

# Cracks on concrete surfaces are common because of the influence of internal and external variables (such as temperature change, foundation deformation, shrinkage, and so on). Cracks, a typical civil engineering flaw, not only harm the health of structures but also cause additional issues. As a result, fracture identification is a critical component of structural health monitoring. The old detection approach is based on human vision, which is expensive yet inefficient, and the detection results are based on subjective human assessment. To address these issues, scientists have proposed a variety of approaches for automatically detecting concrete surface faults. Many machine learning techniques (such as deep learning) are already being utilized to recognize objects with acceptable results, thanks to advancements in processing capability and picture collecting equipment. Deep learning can automatically extract high-level semantic information from source photos, resulting in a new way for detecting cracks in concrete surfaces. The convolutional neural network (CNN), which has a high capacity for learning depth characteristics directly from training data, enhances the efficiency and accuracy of identifying concrete surface faults significantly. It has also been stressed in picture categorization and object detection. Many deep learning-based algorithms for detecting pavement cracks, concrete cracks, concrete insect holes, and other faults have been created by researchers. Excessive training parameters and complex network architecture are common issues with CNN-based fracture detection systems. Object detection techniques are being researched further to overcome these issues.

# CHAPTER 1

# INTRODUCTION

With the growing number of aging infrastructure across the world, there is a high demand for a more effective inspection method to assess its conditions. Routine assessment of structural conditions is a necessity to ensure the safety and operation of critical infrastructure. However, the current practice to detect structural damages, such as cracks, depends on human visual observation methods, which are prone to efficiency, cost, and safety concerns. In this Project, we present an automated detection method, which is based on convolutional neural network models and a non-overlapping window-based approach, to detect crack/non-crack conditions of concrete structures from images. To this end, we construct a data set of crack/non-crack concrete structures, comprising training patches, validation patches, and test patches. Our approach provides over 99% accuracy and over 87% precision in detecting the cracks for most of the convolutional neural network models. We also show that our approach outperforms existing models in literature in terms of accuracy and inference time. Our evaluation shows that deeper convolutional neural network models have higher detection accuracies; however, they also require more parameters and have higher inference time. We believe that this study would act as a benchmark for real-time, automated crack detection for condition assessment of infrastructure. With the rapid development of water conservancy projects, to meet the needs of power generation, shipping and irrigation, lots of hydropower stations are built. The dam is an important hydraulic structure of hydropower stations. Cracks caused by structural deformation, earthquakes, water flow impact and other factors have potential safety hazards for the normal operation of dam. Regular crack detection plays a crucial part in the maintenance and operation of existing dams. The crack detection of traditional human-based visual inspection is inefficient, subjective, and time-consuming. Therefore, automatic and efficient crack detection is highly essential for a structural health assessment of dams. We have demonstrated that a properly trained convolutional neural network (CNN) is able to accurately detect a crack on surface.

**CHAPTER 2**

**2.1 LITERATURE REVIEW**

The main challenges in detecting cracks from images (or videos) are that the features must be invariant to scale, translation, noise, lighting conditions, and shadows. The manual feature extraction methods, which use handcrafted features, are suitable for a specific case and often fail to perform well when tested in real-world conditions. Recently, DL algorithms, such as the CNNs, have shown promising results for real-world applications.30 With the availability of large amounts of training data, the CNN-based architectures usually outperform their shallow counterparts due to generalization of features through hierarchical learning of features at different abstract levels [1]. Two methods can be used to reduce or remove the noise: the wavelet transform technique to remove Gaussian noise, and the median filter to process the water scattering noise [2]. The recognition accuracy of the algorithm can be further improved by adding more data samples. By increasing the data samples and fine-tuning the super-parameters, the recognition accuracy of the algorithm can be further improved, effectively reducing the interference of underwater suspended materials and identifying various crack types[2]. In the process of network training, in order to improve efficiency and better save computing resources and time, this paper adopts the training strategy of freezing certain layers. The entire training process is divided into two stages. In the first stage, only the backbone network structure is trained; in the second stage, the overall network structure is trained. In the training process, the Cosine Annealing learning rate strategy is adopted, and the hyperparameters are optimized according to the genetic algorithm[3]. In pre-processing step, blurness of the image is removed, using Wiener filter. Cracks in poor contrast and non-uniform illumination images are not easy to identify and may yield erroneous results[4]. In underwater dam [Pengfeishi et al., 2017], it is difficult to detect and classify the cracks. Hence, solar images are used to detect and classify crack into tiny, medium and large using tensor voting method. [Salari and Ouyang, 2016] stated that images not only contains a road segment, but also includes other complicated background components. Crack in the pavement images with complicated background components such as trees, houses, etc. are also detected and classified using SVM, fractal thresholding and radon transform[4]. One interesting fact is that although shallow CNN and VGG16 have their own advantages and disadvantages, the average performance of shallow CNN is better[5].

**2.2 AIM AND OBJECTIVES OF PROJECT**

* Underwater crack detection
* Quantification analysis of crack
* Measurement of crack

**CHAPTER 3**

**SYSTEM DEVELOPMENT**

**3.1 SYSTEM BLOCK DIAGRAM**

As required by all the image detection/classification models, a dataset was required. To build a dataset images of cracks were collected and downloaded from internet. There are 40000 images with 20000 images having and crack and 20000 images without crack. This dataset is then split into 70% for training and 30% for testing. Further training data is split into 80% for training and 20% for validation. Using training data and validation data, Neural network is trained and features are extracted. Then sigmoid classifier is used to classify images with crack.

Collecting crack images

Building databank

Training set

Test set

Validation set

Training a CNN

Crack detection

|  |  |
| --- | --- |
|  | CNN classifier  Fig 1 System block diagram  Figure 1 shows a flow chart of using a CNN to detect cracks. It includes three steps: building crack database, training the CNN, and testing the trained CNN classifier. To train a CNN, a large amount of raw images are taken from concrete surface. The collected raw images are cropped into smaller images, and then, cropped small images are manually classified into images with and without cracks. After that, training set and validation set are selected randomly from the database and imported into a CNN for training and validation. The training process generates a CNN classifier that is capable of classifying images into images with and without cracks. Using the trained CNN classifier and an exhaustive search with a sliding window, cracks can be separated from images accordingly.  **3.2 SYSTEM SPECIFICATION** |
|  | We employed a 2-Layer convolutional network, a 5-Layer convolutional network, and the VGG-16 training model. The data for the crack detection dataset is separated into two classes: positive for images with cracks and negative for images without cracks. For data annotation, we utilized the Label img software. For training data, 22400 and 5600 photos were taken for testing and validation, respectively. The model was tested with 12000 photos.  Table 1: system specification   |  |  | | --- | --- | | **Training model used** | 2-Layered convolutional network, 5-Layered convolutional network, VGG-16 | | **Number of classes** | 2 | | **Class names** | Positive, Negative | | **Training dataset** | 22400 images for testing ,5600 images for validation | | **Testing dataset** | 12000 images | | **Data annotation** | Label img |  * 1. **CHALLENGES FACED** * Not enough training data: The biggest problem we faced for crack detection was a lack of data. It took a lot of work to obtain the data. Data gathering and processing for underwater object detection learning presents additional obstacles. * Over fitting: Overfitting is a data science notion that arises when a statistical model fits perfectly against its training data. When this occurs, the algorithm is unable to perform successfully against unseen data, therefore negating its objective. The ability to generalize a model to new data is ultimately what allows us to use machine learning algorithms to make predictions and classify data on a daily basis. * Non representative data: Several unfavorable conditions, including as light scattering and absorption by water and the presence of suspended particles, degrade image quality. The background operates as a foggy entity that distorts perspective and reduces shapes and colors, making augmentation and restoration approaches necessary yet unsuitable for underwater vision. |

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

**4.1 METHODOLOGY**

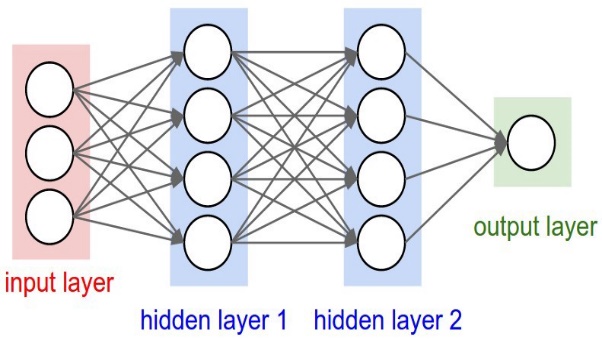
A conventional convolutional neural network usually includes an input layer, a convolutional layer, a pooling layer (typically, it is shown as the form of max pooling layer or the average pooling layer), a fully connected layer, a SoftMax layer, a dropout layer, and an output layer .

For CNN structure, convolutional layer mostly changes the number of channels, and pooling layer will play a role in reducing the image size by down sampling the feature maps or reduce their dimensionality, without losing any significant information or any background information. To train the modified model, we have collected images, including some of the datasets downloaded from Kaggle. The environment of underwater structures, such as concrete piers, is more complex than that of the superstructure. It has the characteristics of rapid flow, turbid water quality, low visibility, strong corrosivity, and large sediment concentration on the surface of the structure. Then, deciding the training model. To save training time and generalize the training process, in the down sampling, part of the parameters in convolutional layers are initialized from pretrained models weights. Dividing dataset into training, validation and testing dataset. Then, Pre-training the dataset and Training model. Detecting cracks on the surface of dams images with or without cracks.

We have used several models to develop our proposed system.

1. **2-Layer CNN Model**

* We have first added a convolutional 2D layer with 16 filters, a kernel of 3x3, the input size as our image dimensions, 120x120x3, and the activation as ReLU.
* After that, we have added a max pooling layer that halves the image dimension, so after this layer, the output will be 59x59x3.
* We have stacked 2 of these layers together, with the other layer having 32 filters.
* Finally, we have done average pooling of the output of the CNN layers, fed it into a fully-connected layer, and then to a sigmoid layer for binary classification.



|  |
| --- |
|  |

Fig2: 2-layer CNN Model

**Model Architecture:**

**Table

Description automatically generated**

Fig3: Architecture of 2-layered CNN model

1. **VGG-16**

* VGG16 is a convolution neural net (CNN ) architecture which was used to win ILSVR(Imagenet) competition in 2014. It is considered to be one of the excellent vision model architecture till date.
* The input to the network is image of dimensions *(224, 224, 3).*
* The first two layers have *64* channels of *3\*3* filter size and same padding. Then after a max pool layer of stride *(2, 2)*, two layers which have convolution layers of 256 filter size and filter size *(3, 3),* This followed by a max pooling layer of stride *(2, 2)* which is same as previous layer.
* Then there are *2* convolution layers of filter size *(3, 3)* and *256* filter. After that there are *2* sets of *3* convolution layer and a max pool layer. Each have *512* filters of *(3, 3)* size with same padding

**Diagram

Description automatically generated**

Fig4: Architecture of VGG16

**Diagram

Description automatically generated**

Fig5: Architecture map of VGG-16

1. **5-Layer CNN Model**

* We have added a convolutional 2D layer with 16 filters, a kernel of 3x3, the input size as our image dimensions, 200x200x3, and the activation as ReLU.
* After that, we have added a max pooling layer that halves the image dimension, so after this layer, the output will be 99x99x3.
* We have stacked 5 of these layers together, with the other layer having 32 and 64 filters.
* Finally, we have done average pooling of the output of the CNN layers, fed it into a fully-connected layer, and then to a sigmoid layer for binary classification.

**Model Architecture:**

Table

Description automatically generated

Fig6: Model architecture of 5-layered CNN model

1. **YOLO**

YOLO is an abbreviation for the term ‘You Only Look Once’. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images. YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects. This means that prediction in the entire image is done in a single algorithm run. The CNN is used to predict various class probabilities and bounding boxes simultaneously.

**Yolo v3 Architecture**

Inspired by ResNet and FPN (Feature-Pyramid Network) architectures, YOLO-V3 feature extractor, called Darknet-53 (it has 52 convolutions) contains skip connections (like ResNet) and 3 prediction heads (like FPN) — each processing the image at a different spatial compression.

Like its predecessor, Yolo-V3 boasts good performance over a wide range of input resolutions. In GluonCV’s model zoo you can find several checkpoints: each for a different input resolutions, but in fact the network parameters stored in those checkpoints are identical. Tested with input resolution 608x608 on COCO-2017 validation set, Yolo-V3 scored 37 mAP (mean Average Precision). This score is identical to GluonCV’s trained version of Faster-RCNN-**ResNet50**, (a faster-RCNN architecture that uses ResNet-50 as its backbone) but 17 times faster. In that model zoo the only detectors fast enough to compete with Yolo-V3 (Mobilenet-SSD architectures) scored mAP of 30 and below.

YOLO algorithm works using the following three techniques:

* Residual blocks
* Bounding box regression
* Intersection Over Union (IOU)

Building Database

Building Network Model

Model Training

Phased Model Training

Overall Model Testing

Embedded Processor Testing

Fig7: Steps of Program

**5. Data Annotation**

Image annotation is defined as the task of annotating an image with labels, typically involving human-powered work and in some cases, computer-assisted help. Labels are predetermined by a machine learning engineer and are chosen to give the computer vision model information about what is shown in the image. The process of labeling images also helps machine learning engineers hone in on important factors that determine the overall precision and accuracy of their model. Example considerations include possible naming and categorization issues, how to represent occluded objects, how to deal with parts of the image that are unrecognizable, etc.

From the example image below, a person applies a series of labels by applying bounding boxes to the relevant objects, thereby annotating the image. In this case cracks are marked manually with orange color. This process is then repeated and depending on the business use case and project, the quantity of labels on each image can vary. Some projects will require only one label to represent the content of an entire image (e.g., image classification). Other projects could require multiple objects to be tagged within a single image, each with a different label (e.g., bounding boxes).

Bounding boxes are the most commonly used type of annotation in computer vision. Bounding boxes are rectangular boxes used to define the location of the target object.

They can be determined by the 𝑥 and 𝑦 axis coordinates in the upper-left corner and the 𝑥 and 𝑦 axis coordinates in the lower-right corner of the rectangle.

A picture containing text, monitor, electronics, screenshot

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Fig8: Data Annotation

**Model Initialization:**

To improve efficiency and conserve computational resources and time during network training, it employs the training approach of freezing specific layers. There are two parts to the training procedure. Only the backbone network structure is trained in the first step; the complete network structure is trained in the second stage. The Cosine Annealing learning rate technique is used throughout the training process, and the hyperparameters are optimised using the genetic algorithm.

**Evaluation Metrics of Accuracy:**

Crack detection based on deep learning is quantitatively measured by objective evaluation metrics, which can measure many aspects of the quality of a restoration algorithm. There are many objective evaluation metrics commonly used in object detection, such as intersection over union (IoU), precision, recall and mean average precision (mAP). IOU is the ratio of the intersection and union between the bounding box predicted by the model and the real bounding box, which is also called the Jaccard index.

mAP is a common index used to evaluate the accuracy of algorithms in the field of object detection. In this paper, the objective evaluation index mAP is used for calculation, as shown in Equation, where AP is the average precision. Taking recall as the horizontal axis and precision as the vertical axis, the P-R curve can be obtained, and the AP value can then be calculated. Simply, this averages the precision values on the P-R curve. The definition of AP is shown in Equation.

mAP=1/|QR|∑q=1QRAP(q)

AP=∫10p(r)dr

The construction of the P-R curve is drawn by the precision and the recall. The precision refers to the number of correct recognitions of all samples predicted to be positive. The recall reflects the missed detection rate of the model. Precision and recall are defined in Equations respectively. True positive (TP) indicates that the detection category is positive and predicted to be positive, while false positive (FP) indicates that the detection category is negative and predicted to be negative. False negative (FN) indicates that the detection category is positive and predicted to be negative, and P is the number of positive samples in the testing set. The precision and recall are independent of each other. High precision means that the false detection rate is low, which can lead to a high missed detection rate.

In addition to the mAP, the model size and computational complexity FLOPs are used to evaluate the model compression algorithm. The model’s size is closely related to its parameters, which can be used to measure the simplification of the YOLOv4 model. FLOPs reflect the calculation amount of the algorithm. The unit of FLOPs is GMacs, which is short for Giga multiply–accumulation operations per second. It represents the floating-point operations per second, which can reflect the algorithm’s calculation performance.

Precision=TP/TP+FP

Recall=TP/TP+FN=TP/P

**CHAPTER 5**

**5.1 RESULT OF IMPLEMENTATION**

The loss value indicates how well or poorly a model performs after each optimization iteration. 0.05201, 0.03082, and 0.00623 are the test losses for 2- Layered CNN, 5- Layered CNN, and VGG16, respectively. An accuracy metric is used to interpretably measure the algorithm's performance. The accuracy of 2- Layered CNN, 5- Layered CNN, and VGG16 is 98.26%, 99.37%, and 99.74%, respectively.

Table 2: Result

|  |  |  |
| --- | --- | --- |
| **Model** | **Test Loss** | **Test Accuracy** |
| **2- Layered CNN** | **0.05201** | **98.26%** |
| **5- Layered CNN** | **0.03082** | **99.37%** |
| **VGG16** | **0.00623** | **99.74%** |

**5.2 ANALYSIS OF RESULT**

The results reinforce the established fact that deeper networks (such as VGG) provide better classification results. The result also indicates that despite 2 LAYER CNN being comparatively shallower network, it provides better results than expectation in detecting cracks of concrete structures. There is very less difference between accuracy of VGG-16 and 5-Layer CNN Model. According to the result VGG 16 is the most accurate model. Images with and without crack are classified successfully. Total 500 images were taken for training and 57 for testing were taken. Cracks are detected successfully using YOLO V3.

**A close up of a white surface

Description automatically generated with low confidenceA close up of a white surface

Description automatically generated with low confidence**

**A close up of a white surface

Description automatically generated with low confidence**

Fig9: Classified images with and without cracks

**A picture containing graphical user interface

Description automatically generatedMap

Description automatically generated with medium confidence**

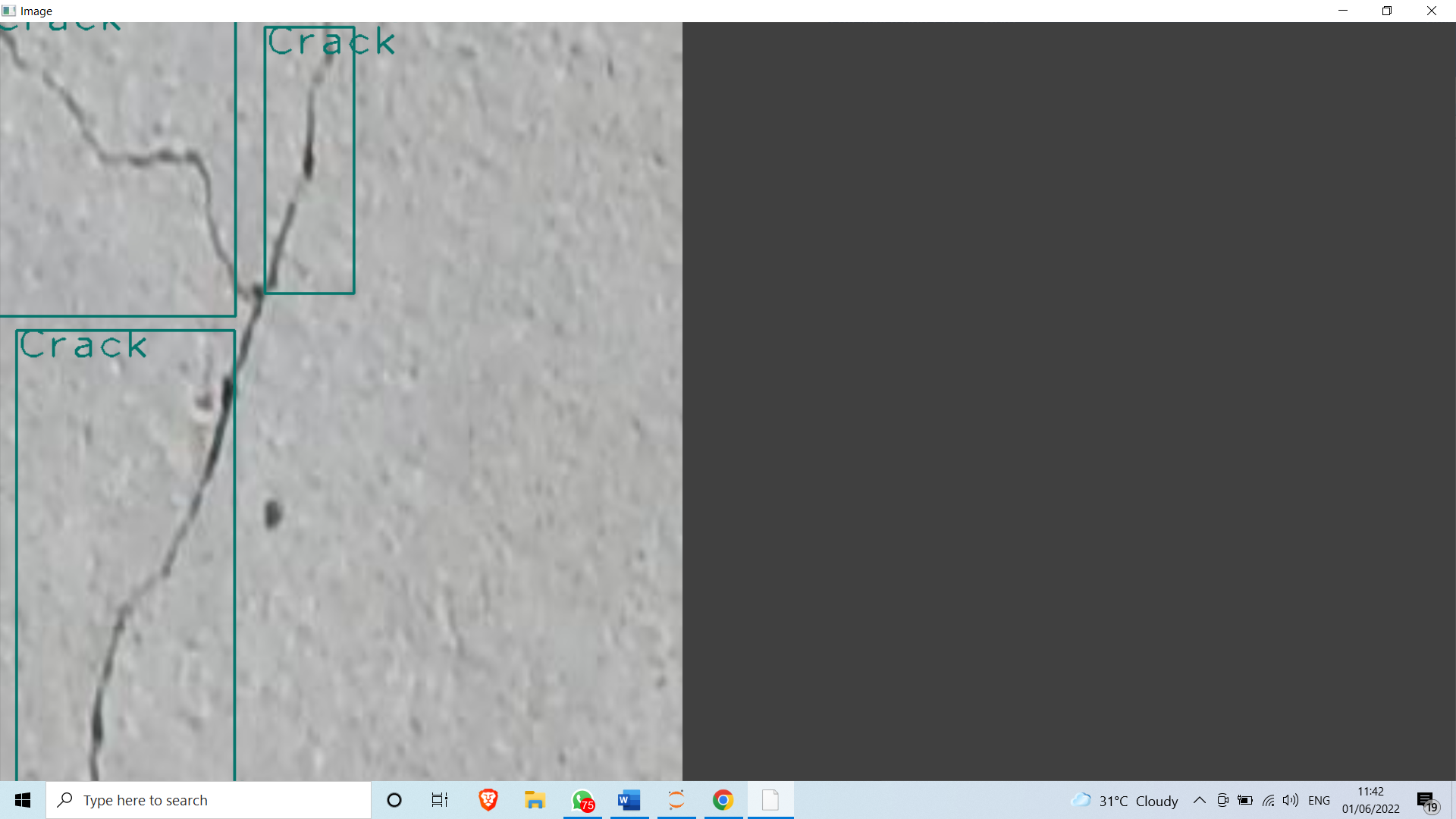
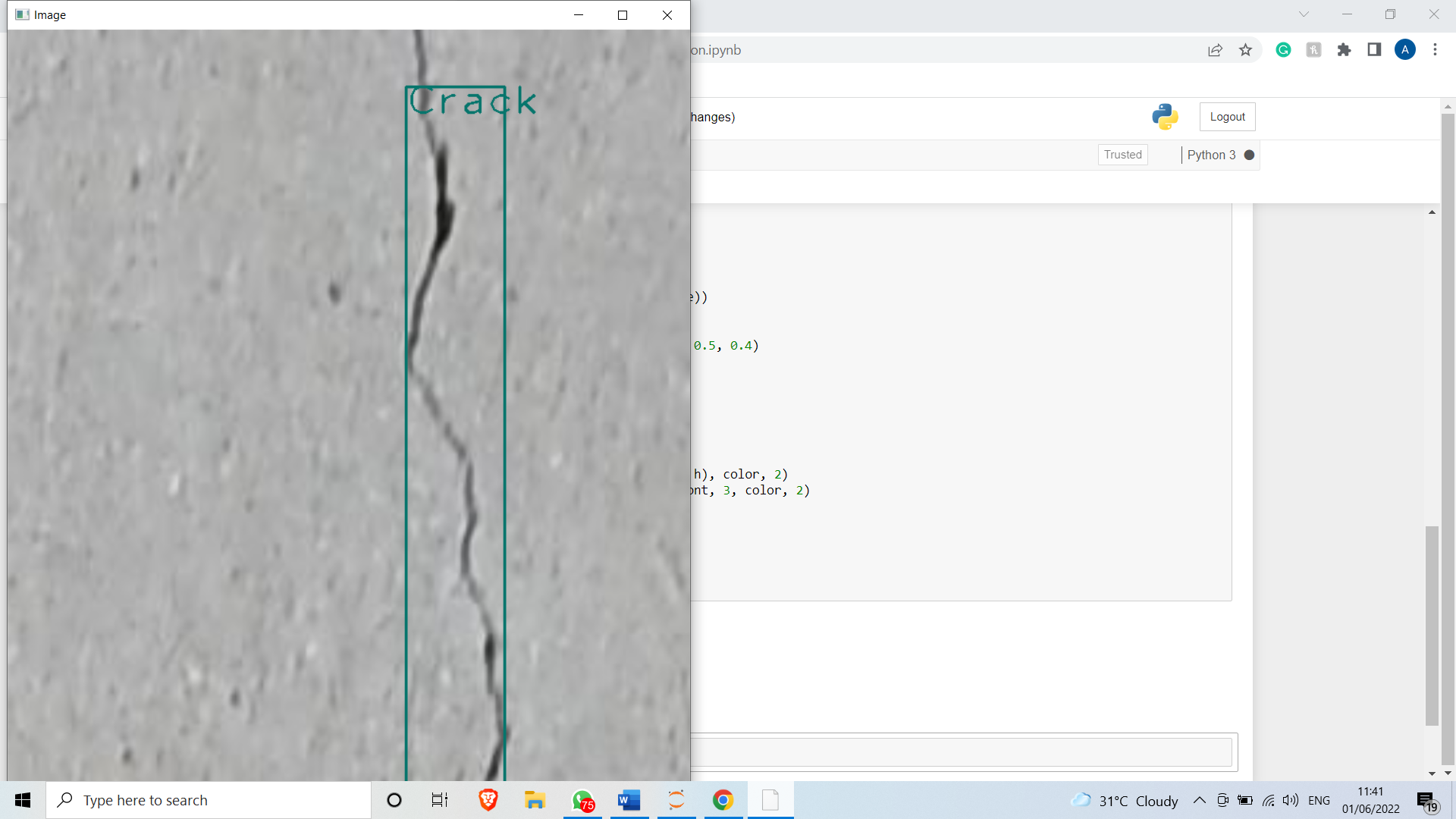


Fig10: Crack Detection

Confusion matrix for the model trained is shown below in table 3. Precision of the model is 0.91, Recall is 0.79, F1 score was calculated to be 0.76 and accuracy is 0.73.

Table 2: Confusion matrix

|  |  |  |
| --- | --- | --- |
|  | Predicted false | Predicted True |
| Actual False | **0** | **4** |
| Actual True | **11** | **42** |

**CHAPTER 6**

**CONCLUSION**

Detecting cracks on concrete surfaces is crucial for the inspection and management of infrastructure as they indicate the possibility of underlying structural damage due to defect or aging and may affect the safety, durability, and serviceability of infrastructure. Automatic crack detection models are imperative to address the drawbacks of manual inspections such as labor and time-intensity, high cost, and safety. Existing methods that use window-based scanning use larger window size, resulting coarsely identified patches of cracks, with less precise localization of cracks in smaller images. In addition, window-based methods use multiple scans (at least 50% overlap in patches) to detect the cracks that appear on the edges of the scanning windows. In this article, we presented an automated crack detection approach using CNN and non-overlapping windows to detect crack/non-crack conditions of concrete structures from images. The size, accuracy, and processing speed of the model can meet the requirements of accurate real-time object detection, which can provide support for the development of mobile monitoring system. As a result, it can achieve real-time automatic vision-based crack detection on concrete surface without other equipment. Although the improved YOLOv3 model shows good performance, there is still a long way to go before it is suitable for engineering applications. First, in the implementation of the improved method, there are many artificially adjusted hyperparameters derived from the training and verification set. Many experiments need to be conducted to explore the influence of these hyperparameters on the performance of the model. Second, a real-time mobile crack detection system should be developed to monitor the concrete surface cracks for timely repair and protection. Lastly, we will collect more types of defect images to expand the database, such that the proposed method has greater accuracy and robustness.

**REFERENCES**

1. Rao, Aravinda S; Nguyen, Tuan; Palaniswami, Marimuthu; Ngo, Tuan (2020). Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure. Structural Health Monitoring, (), 147592172096544–. doi:10.1177/1475921720965445
2. Qi, Zhilong; Zhang, Jinyue; Liu, Donghai (). [American Society of Civil Engineers Construction Research Congress 2020 - Tempe, Arizona (March 8â10, 2020)] Construction Research Congress 2020 - A CNN-Based Method for Concreate Crack Detection in Underwater Environments. , (), 566–575. doi:10.1061/9780784482865.060
3. **MDPI and ACS Style**

Yao, G.; Sun, Y.; Wong, M.; Lv, X. A Real-Time Detection Method for Concrete Surface Cracks Based on Improved YOLOv4. *Symmetry* **2021**, *13*, 1716. <https://doi.org/10.3390/sym13091716>

1. Review and Analysis of Crack Detection and Classification Techniques based on Crack Types Sheerin Sitara. N.1 , Kavitha. S.2 , Raghuraman. G.2 1 PG Scholar, Department of Computer Science and Engineering, SSN College of Engineering, Chennai, India. 2 Faculty of Computer Science and Engineering, SSN College of Engineering, Chennai, India.
2. Shi, J., Yin, W., Du, Y., Folkesson, J. (2019) Automated Underwater Pipeline Damage Detection using Neural Nets
3. Aditya Agarwal , Tushar Malani , Gaurav Rawal , Navjeet Anand, Manonmani S, 2020, Underwater Fish Detection, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 04 (April 2020)
4. D. K. Rout, B. N. Subudhi, T. Veerakumar and S. Chaudhury, "Prominent Object Detection in Underwater Environment using a Dual-feature Framework," Global Oceans 2020: Singapore – U.S. Gulf Coast, 2020, pp. 1-5, doi: 10.1109/IEEECONF38699.2020.9389401.
5. H. M. Manik, "Underwater acoustic signal processing for detection and quantification of fish," Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, 2011, pp. 1-3, doi: 10.1109/ICEEI.2011.6021534.
6. T. Akgül, N. Çalik and B. U. Töreyın, "Deep Learning-Based Fish Detection in Turbid Underwater Images," 2020 28th Signal Processing and Communications Applications Conference (SIU), 2020, pp. 1-4, doi: 10.1109/SIU49456.2020.9302150.
7. J. H. Christensen, L. V. Mogensen, R. Galeazzi and J. C. Andersen, "Detection, Localization and Classification of Fish and Fish Species in Poor Conditions using Convolutional Neural Networks," 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), 2018, pp. 1-6, doi: 10.1109/AUV.2018.8729798.
8. Saini and M. Biswas, "Object Detection in Underwater Image by Detecting Edges using Adaptive Thresholding," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 628-632, doi: 10.1109/ICOEI.2019.8862794.
9. S. K. Mahapatra, S. K. Mohapatra, S. Mahapatra and S. K. Tripathy, "A Proposed Multithreading Fuzzy C-Mean Algorithm for Detecting Underwater Fishes," 2016 2nd International Conference on Computational Intelligence and Networks (CINE), 2016, pp. 102-105, doi: 10.1109/CINE.2016.25.
10. G. Xu et al., "Detection of Bluefin Tuna by Cascade Classifier and Deep Learning for Monitoring Fish Resources," Global Oceans 2020: Singapore – U.S. Gulf Coast, 2020, pp. 1-4, doi: 10.1109/IEEECONF38699.2020.9389012.
11. Shi, Pengfei & Fan, Xin & Ni, Jianjun & Wang, Gengren. (2016). A detection and classification approach for underwater dam cracks. Structural Health Monitoring. 15. 10.1177/1475921716651039.
12. L. Mucolli, S. Krupinski, F. Maurelli, S. A. Mehdi and S. Mazhar, "Detecting cracks in underwater concrete structures: an unsupervised learning approach based on local feature clustering," OCEANS 2019 MTS/IEEE SEATTLE, 2019, pp. 1-8, doi: 10.23919/OCEANS40490.2019.8962401.
13. Chanqing Zhang et al 2021 Meas. Sci. Technol. **32** 125107
14. Hirai, Hiroyasu & Ishii, Kazuo. (2019). Development of Dam Inspection Underwater Robot. Journal of Robotics, Networking and Artificial Life. 6. 18. 10.2991/jrnal.k.190531.004.
15. V. Vijayan, C. M. Joy and S. S, "A Survey on Surface Crack Detection in Concretes using Traditional, Image Processing, Machine Learning, and Deep Learning Techniques," 2021 International Conference on Communication, Control and Information Sciences (ICCISc), 2021, pp. 1-6, doi: 10.1109/ICCISc52257.2021.9484914.
16. Jung, H. K. and Park, G. (2019) ‘Rapid and non-invasive surface crack detection for pressed-panel products based on online image processing’, Structural Health Monitoring, 18(5–6), pp. 1928–1942. doi: 10.1177/1475921718811157.