Damage Assessment In Building Using Python And Deep Learning

K. Chaethra Sri[1]*, Smt. B. Ajitha M.Tech, (Ph.D)[2]

[1]*M. Tech (Computer Aided Structural Engineering), Civil Engineering Department, JNTUA College of Engineering, Ananthapuramu, India, Email: Chaethrasri25@gmail.com
[2]Associate Professor in Civil Engineering Department, JNTUA College of Engineering, Ananthapuramu, India, Email: ajitha123.civil@jntua.ac.in, ORCID No: 0000-0003-4715-8131

***Corresponding Author: -** K. Chaethra Sri
*M. Tech (Computer Aided Structural Engineering), Civil Engineering Department, JNTUA College of Engineering, Ananthapuramu, India, Email: Chaethrasri25@gmail.com

ABSTRACT
The appearance of cracks and other distortions caused by building movement can be visually unattractive and disconcerting for occupants and if left untreated they can affect the integrity, safety and stability of the structure. Accuracy is crucial for disaster response and recovery efforts. Inspection of surface cracks humanly is laboursive and danger. So, here we compare whether the proposed model can assess the damage accurately like human or even better than human using python and deep learning. In recent years, deep learning techniques have shown great potential in automating damage assessment tasks. In this, we propose convolutional neural networks and recurrent neural network. Firstly, we compile the dataset of crack images with varying degrees of damage. The dataset is pre-processed to normalize the images and augment the dataset to improve model generalization. Experimental results on the compiled dataset demonstrate the effectiveness of the proposed approach for damage assessment in buildings. This shows >90% of accurate prediction. The developed Python-based deep learning model shows promising accuracy and efficiency in identifying different types of damage in buildings.

KEY WORDS: Python, Deep Learning, Crack Detection, Convolutional Neural Networks.

I. INTRODUCTION
1.1 GENERAL
Natural disasters such as earthquakes, hurricanes and floods can cause significant damage to buildings, posing threats to human safety and property. Sometimes, settlements due to own weight over time may lead to cracks in structures. Damage assessment has always been done by visual inspection by human experts, which can be labour-intensive and arbitrary. Data capture is the initial step in damage assessment using Python and Deep Learning (DL). This can be done by gathering sensor or picture data from a variety of sources including HD cameras, satellites, drones and information providing sources. These data sources record the exterior or interior features of the structure such as damage indicating fissures, deformations or structural anomalies. Many libraries and frameworks are available in Python that does tasks like image scaling, and contrast enhancing. Python enables us to examine massive volumes of data and generate predictions about the scope and severity of damage by using DL models. Deep Learning (DL) algorithms have shown considerable potential in image classification. DL models can quickly and accurately estimate the type and severity of damage by evaluating photos and sensor data, which can help emergency response teams to make wise judgments. We shall examine the application of Python and DL to damage assessment in buildings in this paper. Readers will have a better idea of how DL can be used to solve practical issues in disaster response and civil engineering.

II. LITERATURE SURVEY
[1] Yildirim, U., Kim, J., & Park, J. (2020). Deep learning-based damage assessment in buildings using remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing, 159, 154170. doe: 10.1016/j.isprsjprs.2019.12.002. The article likely presents a methodology that involves training CNNs using labelled data, such as annotated remote sensing images that indicate the location and extent of building damage. The trained CNNs can be used automatically to classify new remote sensing images as damaged or undamaged. The authors may have evaluated the performance of their proposed approach using various metrics, such as accuracy, precision, recall, and F1-score and compare their results with existing methods or benchmarks. This may also discuss the potential applications and implications of using deep learning for earthquake damage assessment, recovery efforts, informing urban planning policy-making and understanding of the impacts of earthquakes on built environments.

[2] Yu, C., Wang, L., Yang, W., & Zhou, G. (2020). Building damage assessment using deep learning from post-earthquake remote sensing images. ISPRS International Journal of Geo Information, 9(6), 348. doi:10.3390/ijgi9060348. The article focuses on the use of deep learning techniques for assessing building damage caused by earthquakes using remote sensing images. Remote sensing images, which are typically acquired using satellite or aerial sensors, can provide valuable information for post-earthquake damage assessment. The authors propose a deep learning-based approach for automatically detecting and assessing building damage in post-earthquake remote sensing images. The article may also discuss the potential applications and implications of using deep learning.

[3] D. Dais, İ.E. Bal, E. Smyrou, V. Sarhosis, Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning, AutomationConstruction**.125(2021), pp.103606**. In this study the feasibility of DL techniques for crack detection on images from masonry walls is investigated. In order to address the lack of data in the literature, a dataset with photos of complex backgrounds, various crack types and sizes is taken. Different DL networks are considered and by leveraging the effect of transfer learning crack detection on masonry surfaces is performed both on patch and pixel level. The performance of U-net-Mobile Net trained on the masonry dataset deteriorates, i.e. F1 scores declines from 79.6% to 74.7%, when tested on concrete images but not as drastically as reported in the literature when networks trained on concrete images were consequently tested on masonry photos.

[4]. Pasupuleti Revathi, B. Ajitha., Prediction Of Compressive Strength of Metakaolin Blended With Concrete Using Ann (2023), Volume 26 Issue 01, 2023 ISSN: 1005-3026 Https://Dbdxxb.Cn/ Original Research Paper. In M60 grade concrete with various ages, a notable improvement in mechanical qualities was seen after the addition of Metakaolin. If 100% replacement of the steel slag aggregate is desired without exceeding the concrete's workability limit, the recommended Metakaolin dosage is 15%. Predicting the strength attributes is done by Artificial Neural Networks (ANN). The three layers of an ANN are the hidden layer, input, and output. 45 samples will be utilized as training and testing data sets for the ANN model during development. After neural network training, by taking into account the factors influencing the properties of concrete's compressive strength is forecasted using an Artificial Neural Network model (ANN) created in MATLAB. A significant correlation between predicted and measured values is indicated by the obtained R value of 0.99, which is almost equivalent to 1. Based on influencing factors, multi-layered feed-forward network models offer quick predictions. To avoid multiple mixes, which is more cost-effective, civil engineers can benefit from these kinds of computational issues.

III. METHODOLOGY

3.1 Python

Python is a versatile programming language widely used in various applications, including crack detection. Python offers a range of libraries and tools suitable for crack detection tasks. Python's simplicity and readability make it an ideal choice for beginners and experts alike in crack detection. Python's NumPy library facilitates efficient handling and manipulation of image data for crack detection purposes. By training a machine learning model on labelled crack images, Python can automatically detect cracks in unseen images Python's pandas library enables efficient data manipulation and handling for pre-processing tasks. Using Python, crack detection systems can handle large datasets and extract valuable features for accurate crack identification.

3.2 Deep Learning (DL)

Deep Learning is the methodology of machine learning. As this learning is based on deep neural network, it allows neural networks to simulate human like decision making. DL model can provide fast and accurate predictions of damage, which can aid in emergency responses. By leveraging the capabilities of deep learning models, Python allows us to analyse large amounts of data, such as images or sensor readings, and make predictions. Deep learning involves data acquisition. By collecting images or outsource the sensor data from various sources. Python provides a wide range of libraries and frameworks, such as pandas, NumPy, OpenCV or scikit image, that facilitate image pre-processing tasks, such as image resizing, or contrast enhancement. By the end of this paper, readers will have a better understanding of how DL can be applied to real-world problems in civil engineering and disaster response.

3.3 DEEP LEARNING ARCHITECTURES

1) CNN (Convolutional Neural Network)

CNN is the supervised deep learning algorithm we use. In this type of neural network, the layers are trained in a robust manner. CNN has input layer, hidden layers and output layers.
Maxpooling is the discretisation process allows to reduce and resize the dimension of image from the hidden layers. CNNs when we think of a neural network, we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution.

2) RNN (Recurrent Neural Network)

In feed forward neural network the information flows through input layer, hidden layer and output layer. While this process is going on the issues are occurred in feed forward network i.e. not handling the sequential data, considering only current input, not memorizing the previous inputs. These issues recurrent neural networks are introduced. RNNs works on the principle of saving the output of a layer and feeding this back to input of the layer in order to predict the output of the layer. Sequence input analyses the image and classifies it as cracked or normal. Sequence output gives the predicted image.

3) Backpropagation

So technically speaking, Backpropagation is used to calculate the gradient error of the network with modifiable weights. Backpropagation through time (BPTT) happens in RNNs in a manner similar to train feed forward neural network with backpropagation. BPTT for RNNs becomes difficult due to the problem called vanishing gradient. So, here comes RNNs to address the problem with memory cells instead of neurons. This gives the accurate desired results.

4) ResNet50

ResNet (Residual Network) is used to train multiple layers and still achieves compelling results. During the process of training layers, the problem of vanishing gradient occurs. ResNet solves this problem by updating the weights and biases effectively to the initial layers, using this, the gradients can flow directly through skip connection from output layers to input layers.

3.4 WORKING STEPS OF CNNS

a) CONVOLUTIONAL OPERATION

The convolution operation is the first component of our strategy. working of CNN’s is as follows, An RGB image is nothing but a matrix of pixel values having three planes whereas a grayscale image is the same but it has a single plane

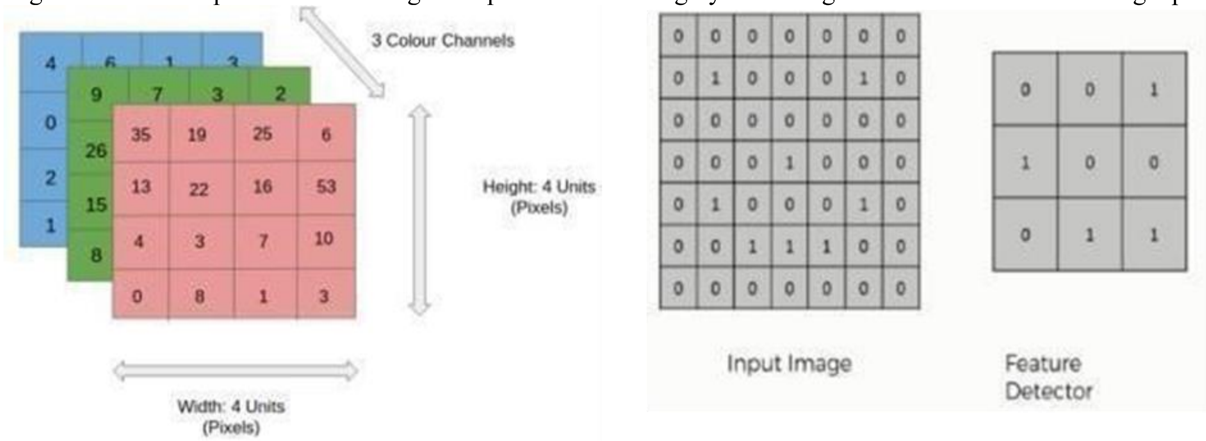
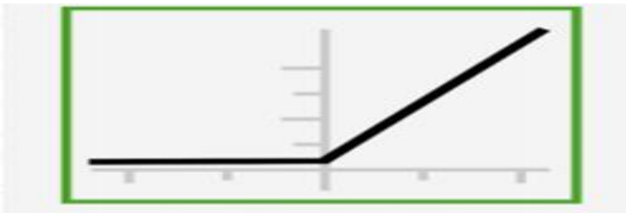


Fig 1: RGB to Grey Scale

b) ReLU LAYER (Rectified Linear Unit)

The Rectified Linear Unit or Relook will be used in the second portion of this process. We will discuss Relook layers and examine the role of linearity in Convolution Neural Networks. For CNN's, it’s preferable to use non-negative activation functions, and a neuron that uses it is called Rectified Linear Unit (ReLU).



$$f(x) = \max(0,x)$$

Fig 2: Linearity Rectifier Function

c) LAYERING POOLS & FLATTENING

We'll discuss pooling in this section and learn exactly how it typically operates. But max pooling will be the central concept in this situation. This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional Neural Networks.

d) FULL CONNECTION

By understanding this, you'll be able to imagine how Convolutional Neural Networks work more clearly and how the "neurons" that are ultimately created and learn how to classify photos. Finally, we'll put everything in perspective and provide a brief summary of the idea in the section. where SoftMax layer and cross entropy is a variant of resnet replaced by 1*1 convolution layer and is addressed.

3.5 SYSTEM SPECIFICATIONS AND SOFTWARES USED

Table.1: System Specifications

S. No	S/W Specifications:		H/W Specifications	
1.	Operating System	Windows 10	Processor	I5/Intel Processor
2.	Server-side Script	Python 3.6	RAM	8GB (min)
3.	IDE	PyCharm	Hard Disk	128 GB
4.	Libraries Used	NumPy, pandas, IO, OS, TensorFlow, Keras.		

SOFTWARES USED

PyCharm, XAMPP (X-operating system, Apache, MySQL, Php, Perl), MySQL (My Structured Query Language)

Note: XAMPP Apache and MySQL should be kept on during the whole process.

V. WEB APPLICATION STEPS

Step 1:

- Open XAMPP software and start the MySQL action.
- Minimize the xampp without stopping the MySQL action and open the MySQL.
- Check the login credentials in the table i.e. Name, email, password and phone no:.

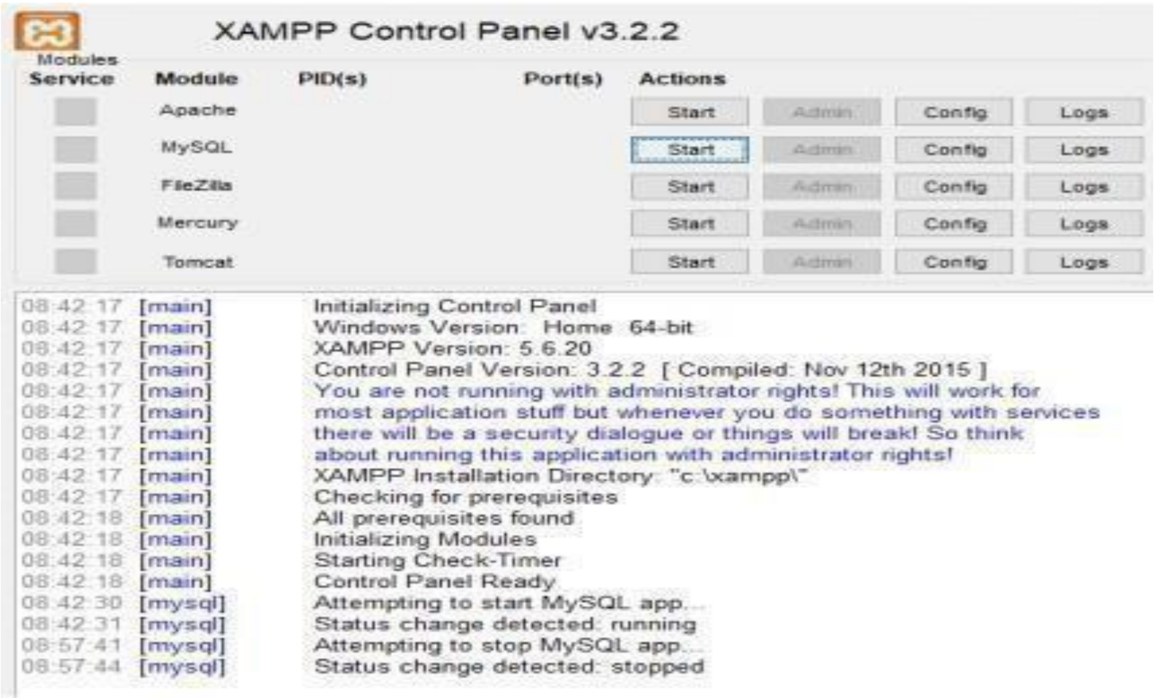


Fig 3: XAMPP control panel

Step 2:

- Now minimize the MySQL software and open the PyCharm where the code is built.
- Enter the command line **pythonapp.py**, to run the program.
- This will generate the link where it leads to the test the further damage assessment.
- Now tap on the link it will redirect you to your website. Now enter the log in details.

Fig 4: Generated Link

Fig 5: Log in Site

STEP 3:
- As soon as you login, you get to upload the image.
- Here firstly we test the cracked images by choosing the cracked image from the folder.
- We upload the image and tap predict then the image is predicted as cracked.

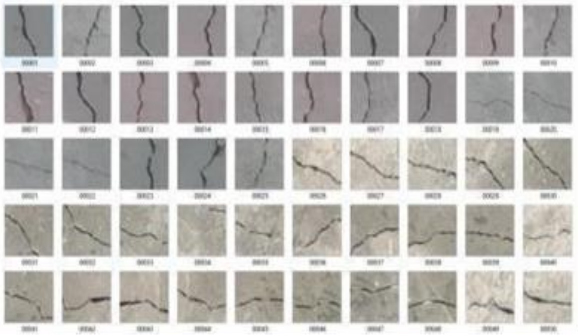


Fig 6: Cracked Images

Fig 7: Image Predicted as Cracked

STEP :4
- Here we test the non-cracked images by choosing the non-cracked image from the folder.
- After choosing the image tap on the predict then image is detected as Normal.

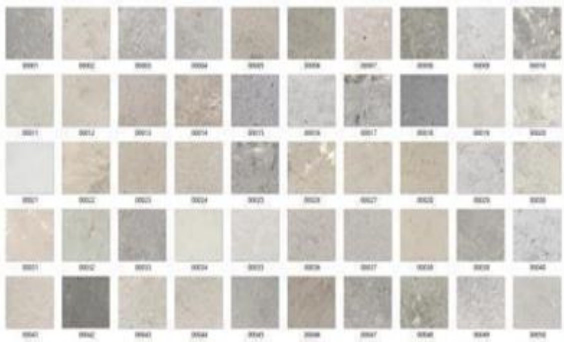


Fig 8: Non-Cracked Images

Fig 9: Image Predicted as Normal

VI. RESULT AND DISCUSSIONS

The result is generated through the code. The x-axis has epochs and the y-axis has accuracy the code above is for my CNN model and I want to plot the accuracy for it, and the output is to be plotted using matplotlib. This program builds the Deep Learning Model for Binary Classification. The data is split into three sets: Training set, Validation set, Test set. The below graph plots the accuracy over the number of epochs. Visualizing data is one of the best ways to humanize data to make it easy to understand and get the relevant trends from it Matplot is the most used curves to understand the progress of Neural Networks is an Accuracy curve. We get the accuracy of >90% in the assessing of the given data. A more important curve is the one with both training and validation accuracy. It records training metrics for each epoch. There was a marginal change in performance because of increasing the training from 10to 20 epochs (within ±0.4%). Previously published pretrained models had similar results. These were marginally better than the current study, which we achieved >90.4% for accuracy. The pretrained model results outperformed CNN models that were trained from scratch. CNN models trained from scratch, only achieved an accuracy of around 90%. This was expected, as pretrained models have weights generated from millions of images, while models trained from scratch have only thousands. Surprisingly the grayscale CNN models produced the same results as the RGB models. The 20-epoch grayscale model was nearly identical. The grayscale results were hard to compare due to the lack of studies on the grayscale. So, CNNs are used for crack detection. This suggests that CNN models created with binarized images detects the images of crack detection. The grayscale and RGB models produced similar results, suggesting that colour is not critical in crack detection. The model was created using pretrained weights further investigation for comparing models "trained from scratch" could confirm the results.

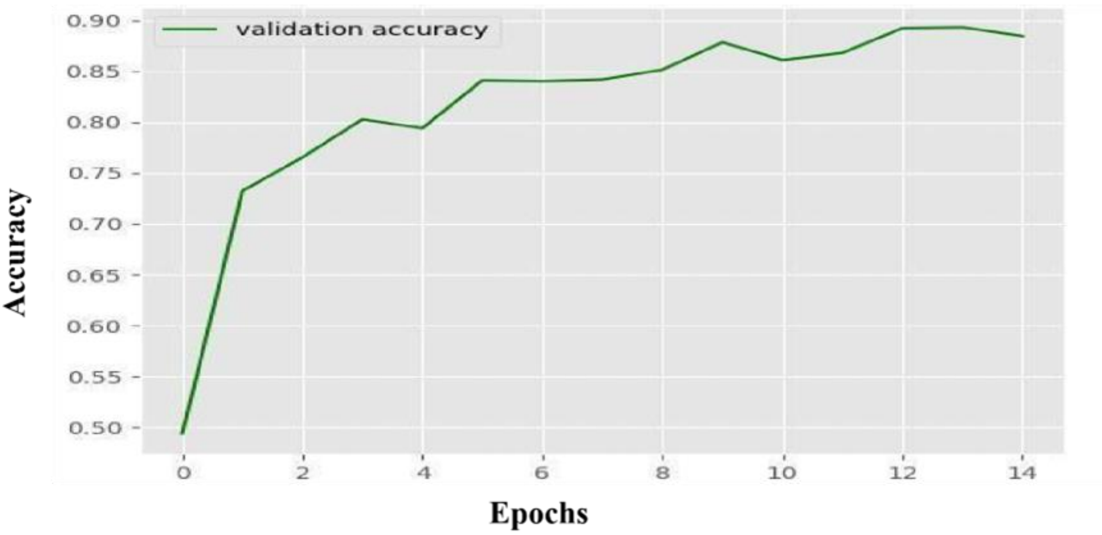


Fig 10: Validation Accuracy Graph

VII. CONCLUSION

- A strong solution for damage assessment in buildings can be provided by using python and deep learning techniques.
- By leveraging CNNs we can effectively detect subtle cracks Python libraries like tensor flow can be used for deep learning tasks as well as training and deploying CNN models Continues Monitoring and maintaining the system gives reliable results i.e; accurate validation with in real world scenarios.
- This network designed to achieve good prediction results. It is possible to develop an automated and effective system for determining the type of damage in buildings and gives good prediction results. Such as cracked and normal.
- By gathering and pre-processing pertinent data, creating and training a deep learning model, and deploying the system in a production environment with a user-friendly interface.
- To guarantee the system's accuracy and dependability in real-world circumstances, it is crucial to continuously monitor and enhance its performance.
- The suggested system has the potential to enhance disaster response and recovery efforts, assisting decision-making processes and permitting swift and efficient responses to lessen the impact of disasters with more study and development in this area.

VIII. REFERENCES

[1] C.Koch, K.Georgieva, V.Kasireddy, B.Akinci and P.Fieguth,"A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure,"Adv.Eng.Inform.,vol.29,no.2, pp. 196–210, Apr. 2015.

[2] Gao Y., Mosalam K.M. Deep transfer learning for image-based structural damage recognition. Comput. Aid. Civ. Infrastruct. Eng. 2018; 33:748–768. doi: 10.1111/mice.12363. [CrossRef] [Google Scholar]

[3] A. Mohan and, S. Poobal, 'Crack detection using image processing: Acritical review and analysis," Alexandria Eng. J., vol. 57, no. 2, pp. 787–798, Jun. 2018.

[4] A. Ahmadi, S. Khalesi, and M. Bagheri, "Automatic road crack detection and classification using image processing techniques, machine learning and integrated models in urban areas: A novel image binarization technique," J. Ind. Syst. Eng., vol. 11, pp. 85–97, Sep. 2018.

[5] Cha Y.J., Choi W., Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Computer. Aided Civ. Infrastructure. Eng. 2017; 32:361–378. doi: 10.1111/mice.12263. [CrossRef] [Google Scholar]

[6] Sony S., Dunphy K., Sadhu A., Capretz M. A Systematic Review of Convolutional Neural Network Based Structural Condition Assessment Techniques. Eng. Struct. 2021; 226:111347. doi: 10.1016/j.engstruct.2020.111347. [CrossRef] [Google Scholar]

[7] Talab, A.M.A.; Huang, Z.; Xi, F.; Hai Ming, L. Detection crack in image using Otsu method and multiple filtering in image processing techniques. Optik 2016, 127, 1030–1033. [Google Scholar] [CrossRef]

[8] Dung, C.V. Autonomous concrete crack detection using deep fully convolutional neural network. Autom. Constr. 2019, 99, 52–58. [Google Scholar] [CrossRef]

[9] Özgenel, Ç.F.; Sorguç, A.G. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In Proceedings of the International Symposium on Automation and Robotics in Construction, Berlin, Germany, 20–25 July 2018. [Google Scholar]

[10] Munawar, H.S.; Hammad, A.W.A.; Haddad, A.; Soares, C.A.P.; Waller, S.T. Image-based crack detection methods: A review. Infrastructures 2021, 6, 115. [Google Scholar] [CrossRef]

[11] Bui, H.M.; Lech, M.; Cheng, E.; Neville, K.; Burnett, I.S. Using Grayscale Images for Object Recognition with Convolution-Recursive Neural Network; IEEE: Piscataway, NJ, USA, 2016. [Google Scholar]