**4. Experimental Results and Discussions**

**Table 3. Performance Metrics in Multi-Class Classification**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Formulae** | **Macro Formulae** | **Description** |
| Accuracy |  |  |  |
| True Negative Rate |  |  | Shows how well the algorithm performs in predicting that an image is not of a certain class |
| Precision |  |  |  |
| Recall Score |  |  | Shows how well the algorithm performs in predicting that an image is of a certain class |
| F1 Macro |  |  | The F1 score is a widely employed metric within the parametric family of F-measures, specifically for the parameter value β=1. |

**4.1 Performance Metrics**

Post prediction, the image being predicted takes one of the four definitions. True Positive (TP), False Positive (FP), True Negative (TN), True Positive. The general performance of the model is obtained using these definitions in algorithms called metrics described in **Table 3**. These are very important to assess the efficacy of the models in the prediction of the target values.

The dataset is balanced as we have similar count of each soil class **Table 2.** Hence, the accuracy is a good metric to evaluate the general performance of the model. Being a multi-class classification problem, for other metrics except accuracy, the metrics are calculated first for each class and then an average is taken as the total report. The Accuracy is good if there is no concern for class distribution thus, it was avoided due to the imbalance in the amount of data across the three classes. For the other metrics, the macro average approach was taken in this study (Grandini et al., 2020). The true negative rate tells us how sensitive our model is to wrongly predicting a class as another class. The True positive rate on the contrary tells us how sensitive our model is in correctly predicting classes. A high value would mean that our model can accurately put most of the images in their correct class. The F1 score is generally used to get a balance of both worlds and is usually used in classification problems where we care about how well or model behaves in classifying TPs and TNs.

**4.2 Analysis**

Towards obtaining the best parameters for the segmentation technique, the correctness of the resulting image was compared with the ground truth using the IoU score after denoising and an average of 0.7 IoU score was obtained for the best set of parameters (*kernel size* of (31, 31), *division* value of 0.4, *Tarea* value of 12, *Tlength* of 43 and *Tradius* of 18. In deciding the parameters to be tuned, few test runs were done to get an idea of the best performing range of values for different kernel size and values for tuning were selected based on this intuition.

**Table 2**. Different parameters of CCELD and the Skele-Marker tuned for each kernel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel | Division Value | Area | Length | Radius |
| (10, 10) | 0.4, 0.5, 0.7 | 6, 8 | 21, 23, 25, 27, 29 | 5, 6, 7, 8, 9 |
| (12, 12) | 0.4, 0.5, 0.7 | 6, 8 | 21, 23, 25, 27, 29 | 5, 6, 7, 8, 9 |
| (14, 14) | 0.4, 0.5, 0.7 | 6, 8 | 21, 23, 25, 27, 29 | 5, 6, 7, 8, 9 |
| (16, 16) | 0.4, 0.5, 0.7 | 6, 8 | 21, 23, 25, 27, 29 | 5, 6, 7, 8, 9 |
| (18, 18) | 0.4, 0.5, 0.7 | 6, 8 | 21, 23, 25, 27, 29 | 5, 6, 7, 8, 9 |
| (20, 20) | 0.4, 0.5, 0.7 | 6, 8 | 21, 23, 25, 27, 29 | 5, 6, 7, 8, 9 |
| (22, 22) | 0.4, 0.5, 0.7 | 6, 8 | 21, 23, 25, 27, 29 | 5, 6, 7, 8, 9 |
| (24, 24) | 0.4, 0.5, 0.7 | 6, 8 | 21, 23, 25, 27, 29 | 5, 6, 7, 8, 9 |
| (25, 25) | 0.85, 1.0, 0.4, 0.7, 1.2 | 10, 12, 14, 16, 18, 8 | 30, 32, 34, 36, 38, 25, 28, 31, 37, 40, 43 | 10, 11, 12, 13, 14, 6, 8, 16, 18, 20 |
| (28, 28) | 0.4, 0.7, 1.2 | 8, 10, 12 | 25, 28, 31, 34, 37, 40, 43 | 6, 8, 10, 12, 14, 16, 18, 20 |
| (31, 31) | 0.4, 0.7, 1.2 | 8, 10, 12 | 25, 28, 31, 34, 37, 40, 43 | 6, 8, 10, 12, 14, 16, 18, 20 |
| (34, 34) | 1.5, 2.0, 0.4, 0.7, 1.2 | 8, 10, 12 | 25, 27, 29, 31, 33, 28, 34, 37, 40, 43 | 8, 10, 12, 6, 14, 16, 18, 20 |
| (37, 37) | 0.7, 1.2, 0.4 | 8, 10, 12 | 37, 40, 43, 25, 28, 31, 34 | 14, 16, 18, 20, 6, 8, 10, 12 |

**4.3 Sensitivity analysis of Image Preprocessing Parameters**

From [Fig 11](#fig11div), the MIOU as well as MF1 scores seem to be more sensitive to the division parameter which makes sense as it’s the first attempt towards denoising by CCELD before denoising by Skele Marker. One interpretation is that with a good threshold, the noise present in the original binary image after CCELD will be less. Hence, moderate values of Skele-marker will be required to denoise thus values lower than the benchmark from (Hamish Dow et al., 2023) is required. Interactive analysis is performed in [4.3](#interanalysis) to evaluate the influence of the parameters on the output when combined together.

|  |  |
| --- | --- |
| (a) | |
| (b) | **(c)** |
| (d) | **(e)** |

**Figure** **9:** Typical results of Mean IoU score (left) and Mean F1 score (right) vs **(a)** Kernel size **(b)** Division **(c)** Tlength **(d)** Tradius **(e)** Tarea

**4.4** **Interactive influence of all Parameters**

In the domain of image preprocessing, the fine-tuning of various parameters is instrumental in achieving optimal results. These parameters hold the power to profoundly influence the quality and characteristics of processed images. Yet, assessing their individual impact alone often offers an incomplete picture of the overall outcomes. To gain a deeper understanding of how these parameters collectively shape the processed images, an interactive analysis becomes indispensable. In this analysis, the comprehensive effects of the parameters on the preprocessed image are revealed.

This analysis is broken down into 3 based on the magnitude of kernel size;

**1. Small Kernel:** This includes kernels with values ranging from (10,10) to (19, 19)

**2. Medium Kernels:** This includes kernels with values ranging from (19,19) to (25,25)

**3. Large Kernels:** This includes kernels with values ranging from (25,25) to (40,40)

|  |  |
| --- | --- |
| (i) | (i) |
| (ii) | **(ii)** |
| (iii)  (a) | **(iii)**  **(b)** |

**Figure 10:** Planar approximations of **a(i-iii)** CCELD parameters and **b(i-iii)** Skele-Marker parameters influence on MIOU score across the different Kernel segments.

The results of CCELD parameter influence on MIOU are approximated with a linear plane and in the small kernel region, the problem of underfitting is observed as the plane is sloped downwards towards smaller kernel size values, coming to larger kernels, a strict negative correlation between MIOU and the division parameter is immediately noticed. It is important to point that the linear approximation might not be the best and should not be interpreted that much lower division parameter values will yield better results as values of same kernel with similar division values from the medium kernel plot are close indicating a plateau.

The results of the Skele-Marker parameters influence on MIOU shows that across all kernel segments, a larger length indicates better performance. At a given length, there’s a variation of MIOU scores and in each group similar area and radius values exist. This indicates that these parameters themselves don’t significantly influence the results.

From the above results the general deductions are drawn;

1. A small division is a good way to start denoising.

2. For moderate crack widths, a large kernel (between 25,25 and 40,40) is recommended.

3. Large Tlength values are encouraged 55 is recommended from (Hamish Dow et al., 2023) and a value of 43 is used in this study.

4. Small area values (between 6 and 8) will suffice and radius values between (10 and 14) will suffice.

**4.5 MobileNet Training and Performance Comparisons**

**Table 4** Performance of different Algorithms on the test data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | MIOU | MF1 | MP | MTPR |
| Image Preprocessing + ADABoost | **0.70** |  |  |  |
| MobileNet |  |  |  |  |

Transfer learning was employed and weights from Imagenet was used with the VGG19 as the encoder which was found to be the best encoder by (Liu & Wang, 2022). On the first 2 epochs, the encoder weights were frozen. By freezing the encoder for the first few epochs, the encoder’s weights are kept relatively stable since they are already well-learned from ImageNet and the features learnt by the encoder from ImageNet are generally useful for a wide range of computer vision tasks. This is a way to prevent the encoder from adapting too quickly to the specific task, which could potentially result in the loss of valuable knowledge contained in the ImageNet weights. By doing so, the decoder is then allowed to adapt and fine-tune to the specific segmentation task.

After this initial learning by the decoder, the encoder weight was made trainable. The model is immediately evaluated for the Mean IOU score of the validation dataset for each epoch. During training, a checkpoint was created to save the best model weights after some epochs such that those weights could be reloaded and training can be resumed. Data augmentation of vertical flipping, horizontal flipping, width shifting, height shifting and zooming was also employed and the model results with and without data augmentation are recorded. Both cases were run for 50 epochs which was enough to get good performance since transfer learning was employed and the validation set yielded better results without data augmentation. The best model was saved after which the weights were loaded and used in making predictions.

The UNet model had the best performance from the training without augmentation and achieved a MIOU score of 0.64 and MF1 score of 0.78. [Fig 12](#fig14a) shows the training and validation loss as well as the validation MIOU scores at different epochs with and without data augmentation.

|  |  |
| --- | --- |
| (a) | (b) |
| (c) | |

**Figure 12:** **(a)** Training and validation loss for 50 epochs without Data Augmentation **(b)** Training and validation loss for 50 epochs with Data Augmentation **(c)** MIOU on validation set for model trained with and without Data Augmentation.

SHOW CLASSIFICATION REPORT TABLES FOR THE DIFFERENT MODELS

**5. Conclusion and Future Works**

In this study, a novel technique for crack detection in images is introduced, namely “Crack Contrast Enhancement and Local Deviation (CCELD)”. This technique leverage image preprocessing towards obtaining efficient crack segmentation. The CCELD method utilizes the principle that the darkest regions in an image containing cracks can be used for effective identification. By applying convolution and algorithms containing thresholding and denoising, CCELD augmented with the Skele-Marker method successfully highlights crack regions, allowing for accurate detection. However, further experimentation and fine-tuning may be needed to further optimize the size of the mean kernel and other parameters and enhance the thresholding process for a broader range of image conditions. The range of parameter values is usually dependent on crack width.

CCELD augmented with the Skele-Marker denoising technique performed slightly better than the UNet model with an MIOU score of 0.06 and F1 score of 0.02 more than that of the UNet Model which indicates a slightly better performance. However, the CNN models are amazing and with tons of training data they will perform very well but will also require high computational resources which aren’t available to everyone. This study proposes a promising step towards accessible, efficient and automated crack detection for a developing region like Nigeria. Towards a practical implementation, a [web app](https://crack-detection-using-cnns-zd5cust8t9yfertfcammnj.streamlit.app/) that implements video and crack segmentation was created for the end users.

For future improvements, the following will be explored:

**1. Dataset Expansion:** Expansion of the dataset with a wider variety of crack images, including different crack types, sizes, and orientations, will be done to enhance the techniques’ generalization capabilities.

**2. Fine-Tuning:** More extensive experimentation to fine-tune the parameters of CCELD for optimal performance under various conditions will be conducted.

**3. Image wise Analysis:** Did the general parameters perform best for each image? If not, what happened? Can ideal parameter values be determined on an image level using ML? These questions will be answered going forward.

**4. Severity Classification:** Severity classification will be incorporated into the crack detection process. This addition would provide valuable insights into the extent of damage, enabling more informed maintenance decisions.

In conclusion, the presented techniques offer innovative approaches to crack detection that show promise in image preprocessing. By addressing the suggestions for improvement and conducting further validation, these methods have the potential to contribute significantly to the field of image-based crack detection and pave the way for safer and more efficient infrastructure maintenance.

**Acknowledgements**

We would like to extend our gratitude to several individuals whose contributions were integral to the completion of this study. Firstly, we express our sincere appreciation to Engr Prof. Nnolim for his guidance and support throughout the development and implementation of the novel preprocessing crack detection technique – Crack Contrast Enhancement and Local Deviation (CCELD). His expertise and insights greatly enriched the methodology and results of this research.

We are also thankful to AEDJAC Lab for their assistance in providing a comfortable and equipped working environment. Their contributions in providing these resources enabled the proficient completion of the project.

Without the invaluable support and collaboration from these individuals and organizations, this study would not have been accomplished. Their dedication and expertise have contributed immensely to the success of this research endeavor.

**Declaration of Generative AI and AI-assisted technologies in the writing process**

During the preparation of this work the author(s) used ChatGPT in order to implement few functions in python code. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

**References**

*Udoh,* *(7) Logistics Regression using Gradient Descent | LinkedIn*. (n.d.). Retrieved October 15, 2023, from https://www.linkedin.com/pulse/logistics-regression-using-gradient-descent-udoh-chigozie/?trackingId=j8o2caXZSC2NSnetUCkrTw%3D%3D

Carrasco, M., Araya-Letelier, G., Velázquez, R., & Visconti, P. (2021). Image-Based Automated Width Measurement of Surface Cracking. *Sensors (Basel, Switzerland)*, *21*(22), 7534. https://doi.org/10.3390/s21227534

Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks: Deep learning-based crack damage detection using CNNs. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378. https://doi.org/10.1111/mice.12263

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 6. https://doi.org/10.1186/s12864-019-6413-7

Fujita, Y., Shimada, K., Ichihara, M., & Hamamoto, Y. (2017). A method based on machine learning using hand-crafted features for crack detection from asphalt pavement surface images. *Thirteenth International Conference on Quality Control by Artificial Vision 2017*, *10338*, 117–124. https://doi.org/10.1117/12.2264075

Ganesan, P., & Sajiv, G. (2017). A comprehensive study of edge detection for image processing applications. *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 1–6. https://doi.org/10.1109/ICIIECS.2017.8275968

Gao, Q., Wang, Y., Li, J., Sheng, K., & Liu, C. (2020). An Enhanced Percolation Method for Automatic Detection of Cracks in Concrete Bridges. *Advances in Civil Engineering*, *2020*, e8896176. https://doi.org/10.1155/2020/8896176

Golding, V. P., Gharineiat, Z., Munawar, H. S., & Ullah, F. (2022). Crack Detection in Concrete Structures Using Deep Learning. *Sustainability*, *14*(13), Article 13. https://doi.org/10.3390/su14138117

*Google Colaboratory*. (n.d.). Retrieved November 12, 2023, from https://colab.research.google.com/github/d2l-ai/d2l-tvm-colab/blob/master/chapter\_gpu\_schedules/arch.ipynb#scrollTo=B7\_DogYoCPgJ

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: An Overview* (arXiv:2008.05756). arXiv. https://doi.org/10.48550/arXiv.2008.05756

Inazumi, S., Intui, S., Jotisankasa, A., Chaiprakaikeow, S., & Kojima, K. (2020). Artificial intelligence system for supporting soil classification. *Results in Engineering*, *8*, 100188. https://doi.org/10.1016/j.rineng.2020.100188

Joshi, D., Singh, T. P., & Sharma, G. (2022). Automatic surface crack detection using segmentation-based deep-learning approach. *Engineering Fracture Mechanics*, *268*, 108467. https://doi.org/10.1016/j.engfracmech.2022.108467

Kheradmandi, N., & Mehranfar, V. (2022). A critical review and comparative study on image segmentation-based techniques for pavement crack detection. *Construction and Building Materials*, *321*, 126162. https://doi.org/10.1016/j.conbuildmat.2021.126162

Liu, F., & Wang, L. (2022). UNet-based model for crack detection integrating visual explanations. *Construction and Building Materials*, *322*, 126265. https://doi.org/10.1016/j.conbuildmat.2021.126265

Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258. https://doi.org/10.1016/j.array.2022.100258

Nguyen, A., Nguyen, C., Gharehbaghi, V., Perera, R., Brown, J., Yu, Y., & Kalbkhani, H. (2022). A computationally efficient crack detection approach based on deep learning assisted by stockwell transform and linear discriminant analysis. *Structures*, *45*, 1962–1970. https://doi.org/10.1016/j.istruc.2022.09.107

Nnolim, U. A. (2020). Fully adaptive segmentation of cracks on concrete surfaces. *Computers & Electrical Engineering*, *83*, 106561. https://doi.org/10.1016/j.compeleceng.2020.106561

Perez, L., & Wang, J. (2017, December 13). *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. arXiv.Org. https://arxiv.org/abs/1712.04621v1

Poynton, C. (2005). *Frequently Asked Questions about Color*.

Prewitt operator. (2023). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Prewitt\_operator&oldid=1165475587

Semma, A., Lazrak, S., Hannad, Y., Boukhani, M., & El Kettani, Y. (2021). WRITER IDENTIFICATION: THE EFFECT OF IMAGE RESIZING ON CNN PERFORMANCE. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVI-4/W5-2021*, 501–507. https://doi.org/10.5194/isprs-archives-XLVI-4-W5-2021-501-2021

Skeleton-based noise removal algorithm for binary concrete crack image segmentation. (2023). *Automation in Construction*, *151*, 104867. https://doi.org/10.1016/j.autcon.2023.104867

Sobel operator. (2023). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Sobel\_operator&oldid=1174035253

Talab, A. M. A., Huang, Z., Xi, F., & HaiMing, L. (2016). Detection crack in image using Otsu method and multiple filtering in image processing techniques. *Optik*, *127*(3), 1030–1033. https://doi.org/10.1016/j.ijleo.2015.09.147

Taluja, C., & Thakur, R. (2019, March 28). *An Intelligent Machine Learning Model for Soil Image Classification*.

V G, N., & S., H. (2010). Study and comparison of various image edge detection techniques used in quality inspection and evaluation of agricultural and food products by computer vision. *Int Agric Biol Eng J.*, *4*. https://doi.org/10.3965/j.issn.1934-6344.2011.02.083-090

*What is Canny edge detection?* (n.d.). Retrieved October 8, 2023, from https://www.educative.io/answers/what-is-canny-edge-detection

Zeeshan, M., Adnan, S. M., Ahmad, W., & Khan, F. Z. (2021). Structural Crack Detection and Classification using Deep Convolutional Neural Network. *Pakistan Journal of Engineering and Technology*, *4*(4), 50–56. https://doi.org/10.51846/vol4iss4pp50-56

Zhao, M., Shi, P., Xu, X., Xu, X., Liu, W., & Yang, H. (2022). Improving the Accuracy of an R-CNN-Based Crack Identification System Using Different Preprocessing Algorithms. *Sensors*, *22*(18), Article 18. https://doi.org/10.3390/s22187089