Concrete Surface Crack Detection Classification Using Deep Learning

CS3244 Project Report
By: ML Singapore! 12
Ang Bo Yuan (A0219906X), Loo Guan Yee (A0223941J),
Michelle Yong (A0221827H), Tan Weiu Cheng (A0219804A)
Lecturers: Bryan Low Kian Hsiang, Daren Ler Shan Wen

Abstract

The concrete cracks surface acts as a visual warning for inadequate structural support, damaged infrastructure, and poor material composition. The detection of such warnings in concrete buildings is crucial for buildings maintenance. If left untreated, it can significantly compromise the structural integrity of the buildings. The need for time and cost-effective methods of building condition surveys is growing especially with the increase in high-rise buildings and expanding transportation networks. This study aims to create and select an optimal concrete surface crack classification model with Deep Learning to alleviate the current labour consuming approach toward concrete monitoring.

Introduction

The condition of the concrete building structures deteriorates over time due to several factors like ageing, weathering, and human activities. External cracks are indications of imminent structural failures and require immediate attention to rectify the issues. Failure to identify is potentially fatal as seen from past incidents such as Nicoll Highway Collapse in 2004 and the PIE Viaduct collapse in 2017.

The inspection and compliance of concrete integrity fall under the supervision of the BCA (Building Construction Authority) in Singapore. BCA conducts regular audits and inspections on buildings. However, the current method used for inspecting surface constructions requires human intervention and is limited to accessible structural surfaces within human reach. Hence, the surveyors could not check hard to reach areas.

Singapore used an average of 11.5 million cubic metres of concrete annually over the last five years (MND 2021). The building and transportation sector contributes to the majority of Singapore's concrete consumption. The Housing Development Board (HDB) has transitioned its building method from using bricks to precasts. The precast technology is an efficient method for producing concrete-filled moulds, which are then stacked up and stored in large chambers for hardening. The precast technology has significantly increased construction productivity (HDB 2022) and reduced construction time. Approximately 70% of HDB's Build To Order (BTO) projects are built using concrete. Given the extensive use of concrete by HDB, the model would be helpful in assessing the precasts before construction, as well as the condition of existing HDB buildings.

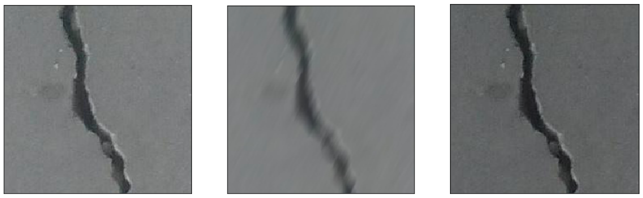
In the Land Transport Authority (LTA) 2040 master plan, more underground stations will be built around Singapore in the future to improve connectivity (LTA 2016). This increases labour demand to maintain the growing transport network, placing a strain on the existing workers. Hence, the utilization of drones to scan and detect cracks in MRT tunnels alleviate the labour strain in the future. However, the drones may produce unsteady images during air maneuver.

Our group has shortlisted 3 state-of-the-art deep learning convolution models: AlexNet, VGG-16 and ResNet-50. These models will be trained with a holistic set of surface crack data, accounting for different image occurrences and our drone implementation. The models will be used to predict another set of unseen images, depicting possible real-life scenarios applicable to Singapore's context. We aim to choose the most robust deep convolutional neural network model based on its prediction performance for the unseen images.

Data sets

AlexNet, VGG-16 and ResNet-50 will be trained and validated with the dataset available from the digital USU (Utah State University) open-source data (Dorafshan, Thomas, and Maguire 2018). This data set collects cracked and non-cracked images of concrete bridge decks, walls and pavements in the vicinity of the USU Campus. The dataset contains 58,000 images that sufficiently cover a various number of conditions such as obstructions, shadows, surface roughness, texture, edges, holes and background debris.

We will be evaluating the performance of the models in an unseen context. The unseen dataset is the METU (Middle East Technical University) image dataset (Özgenel and Sorguç 2018). The images in the dataset are collected around the METU campus. The dataset is divided into negative and positive crack images classes. Each class has 20,000 images. A visual inspection of the images shows the variation in the surface finish and illumination conditions. Similarly, blurring and light intensity adjustments will be applied to the images to investigate the performance of the models in the Singapore application context i.e. crack classification by drones manoeuvring in dimly lit MRT tunnels.



(a) Original image (b) Blurred image (c) Darkened image

Figure 1: Image Variations

Approach

Comparison of different CNN models

To select a high accuracy and efficient model for crack detection, the following three CNN models are trained and tested with the pre-processed data sets. By comparing the (i) $recall/accuracy$, and the (ii) computational time, we can study the performance of these existing models on crack classification.

AlexNet The AlexNet architecture consists of a total of eight layers, five of which are the convolutional layers and three fully-connected (FC) layers (Wei 2020). The convolution layers include multiple feature maps. Each feature map consists of an arrayed neurons and neurons from the same feature map share weights called the convolution kernels. The kernels weaken the connection between the network layers. When conducting convolution operations, kernels convolve with the upper neurons at a certain stride; bias is added and outputted in the process (Li and Zhao 2019).

The model is trained in the ImageNet database, and modified with only 2 classes, which are images with and without cracks (Li and Zhao 2019). AlexNet uses Rectified Linear Units (ReLU) instead of the $\tanh$ function to reduce training time. ReLU is a non-linear activation function which is $f(x) = \max(0, x)$ with a much faster gradient descent training time because the gradients of the ReLU are zero or one. If the training samples give positive input to ReLU, learning will happen in that neuron. This results in faster calculations and convergence speed (Li and Zhao 2019).

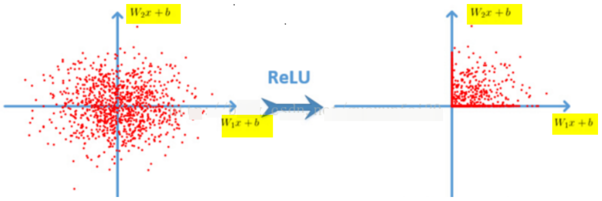
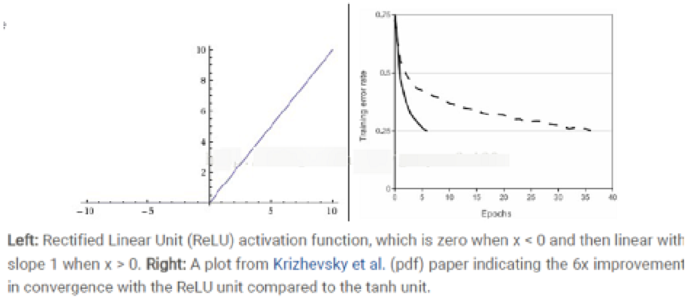


Figure 2: ReLU function: $f(x) = \max(0, x)$; source: Varshney (2020)

Multi-GPU training is utilized by the model to place half of its neurons on one GPU and the other half on another GPU. This method of training allows for bigger models while cutting training time significantly. Label-preserving transformation generates image translations and horizontal reflections of the data set, which increased the training set by a factor of 2048 (Wei 2020). The larger and more varied data trains the model more vigorously for more accurate results.



Left: Rectified Linear Unit (ReLU) activation function, which is zero when x < 0 and then linear with slope 1 when x > 0. **Right:** A plot from Krizhevsky et al. (pdf) paper indicating the 6x improvement in convergence with the ReLU unit compared to the tanh unit.

Figure 3: Number of iterations for a four-layer convolutional network to reach 25% training error in $\tanh$ and ReLU; source: Varshney (2020)

Furthermore, the Overlapping Pooling technique groups neighbouring neurons and overlap them to reduce the error caused by overfitting (Krizhevsky, Sutskever, and Hinton 2012). Another neuron dropout technique consists of "switching off" neurons with a predetermined probability (e.g. 50%). For every iteration, a different sample of the model's parameters forces each neuron to be more robust when used with other random neurons (Wei 2020). The dropout technique randomly removes neurons with a given dropout probability when weights are updated. The full connection layers functions as logical inference. The softmax layer estimate a possibility for the cracked and un-cracked class so it is essential for classification (Li and Zhao 2019).

The motivation for using AlexNet is to be able to train a model using a large varied data set with faster training time yet mitigates overfitting the model at the same time (Wei 2020). AlexNet uses the ReLU function for faster gradient descent training time with larger data mini-batch (Varshney 2020). The data augmentation method implemented and overlapping pool helps to avoid overfitting. (Li and Zhao 2019) This allows us to generalise well to real-life image cracks that are being influenced by many more variables as opposed to training with smaller data sets.

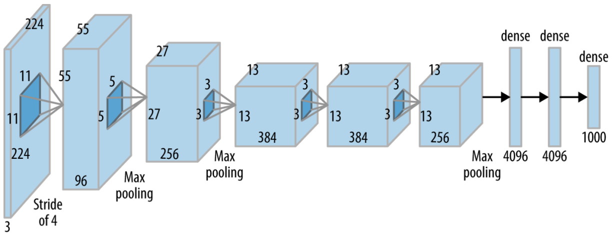


Figure 4: Architecture of AlexNet; source: Anwar (2022)

VGG-16 The general layout of VGG networks starts with spatial pooling, which consists of 5 max-pooling layers following several convolutional layers. The first set of convolutional layers starts with the same dimension as its input (224 × 224) and 16 channels. For each subsequent set of convolutional layers after a max-pooling layer, the dimensions are halved and the number of channels is doubled until it reaches 512 channels. This is achieved by having the kernel size and stride of max-pooling layers fixed at 2. The kernel size of the convolutional layers is fixed at 3 × 3 and the padding is fixed at 1.

The rationale of increasing the number of feature channels by a factor of 2 while reducing its dimensions, helps to extract low-level features (e.g. pixel-level details) and transform them into higher-level features (e.g. edges, texture). The spatial pooling stage enables the second part of the neural network to infer from higher-level data.

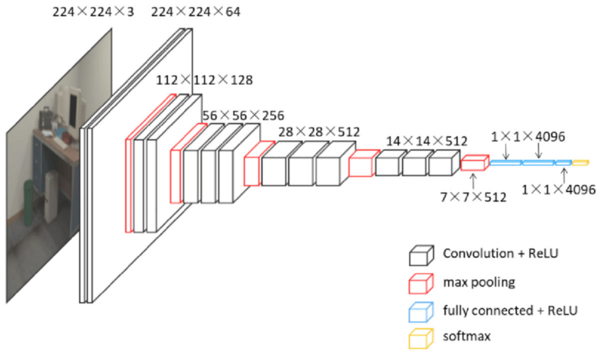


Figure 5: Architecture of VGG-16; source: Anwar (2022)

After which, the output of the spatial pooling stage is squashed and passed into FC layers: the first two having 4096 nodes each, while the last one has 1000 nodes. Since our problem only requires a binary classification, an additional 3 FC layers of 64, 64, and 2 nodes each were added. For nonlinearity, ReLU is used for the convolutional layers and the FC layers. The final layer is a soft-max layer used to turn the outputs of the final layer into a probability distribution representing the probability distribution of the input's classification, or the model's confidence in its classification.

Our group used the VGG-16 model to leverage its spatial pooling. Convolutional layers in conjunction with pooling layers provide the model with the ability to learn translational invariance (Biscione and Bowers 2020). Through the use of convolution, the model can relate multiple pixels, which helps with learning the features.

As mentioned in the original paper, the use of multiple convolutional layers per block before a MaxPool increased its non-linearity, as more ReLU layers were incorporated before pooling, making the decision function more discriminative (Simonyan and Zisserman 2014). The choice of having a small 3 × 3 kernel and stride of 1 also made sure that each pixel contributed to the next layer more while not over-increasing the number of parameters. This was especially important as some of the cracks were only a few pixels wide, the model must be able to capture these features effectively.

ResNet-50 Since the AlexNet introduction in 2012, Subsequent deep learning neural networks designed have added more layers. The additional layers greatly improved the image recognition tasks of deep Neural Network (Girshick 2015). Initially, vanishing or exploding gradient is a major issue when more layers are added. The issue is solved with normalised initialisation and intermediate normalisation (Ioffe and Szegedy 2015) layers are implemented to allow the layers to start converging for stochastic gradient descent with back-propagation (LeCun et al. 1989).

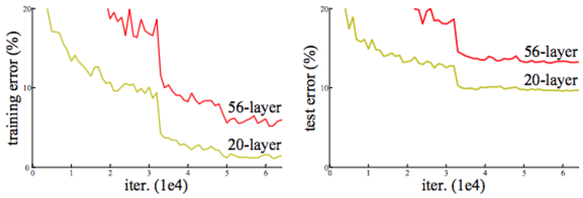


Figure 6: The training error increases when the number of layers increases (He et al. 2015)

However, a degradation problem was noticed when the deeper networks started to converge. When the network depth increases, the accuracy first becomes saturated and then experienced degradation (He et al. 2015). The degradation is not caused by overfitting contrary to the common literature but caused by the addition of more layers to the deep model (He et al. 2015).

Introduced by the Microsoft Research team in 2015, ResNet (Residual Network) is a deep convolutional neural network to tackle the accuracy degradation faced by deeper layers. ResNet has obtained 1st place in the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2015 for its ensemble model, ImageNet detection and ImageNet localization.

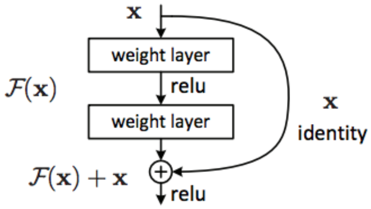


Figure 7: A Residual building block (He et al. 2015)

Our group has selected ResNet-50 to train our Utah training dataset. ResNet-50 is a convolutional neural network that is 50 layers deep, consisting of 48 convolution layers, 1 MaxPool and 1 Average Pool layer. ResNet-50 consists of multiple residual blocks. It has incorporated the concept of skip connection, which allows the model to learn an identity function that ensures that the top layer will perform at least as good as or better than the lower layers. From figure 2, there is a direct connection that skips some layers in between. This connection is the skip connection and is illustrated with a residual block example. $\mathbf{x}$ is denoted as the inputs to the first of these layers. $\mathcal{F}(\cdot)$ is denoted as the activation function.

For each input vector, its corresponding output vector, denoted by $\mathbf{y}$, is denoted by the following equation:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{w_i\}) + \mathbf{x} \tag{1}$$

representing the residual learning function to be learned. In figure 7 which has 2 layers, $\mathcal{F}$ is equal to $W_2\sigma(W_1\mathbf{x})$. The σ is denoted as ReLU and the bias is omitted for expression.

However, the equation assumes that the dimensions of $\mathcal{F}$ and $\mathbf{x}$ are equal. In the case where the dimension of $\mathcal{F}$ and $\mathbf{x}$ are not equal, a linear project W_s by the shortcut connections could be performed to match the dimension. Note that the function $\mathcal{F}(\mathbf{x}, \{w_i\})$ can represent multiple convolutional layers.

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{w_i\}) + W_s\mathbf{x} \tag{2}$$

With the skip connection, the problem of vanishing gradient in deep neural networks is overcome, allowing an alternate shortcut path for the gradient to pass through. The skip connection helps the additional layers in a deep network to learn the identity function. Hence, if the output equals inputs, performance will not degrade even with extra layers. The use of skip connection adds the output from an earlier layer to a later layer, mitigating the vanishing gradient problem presence in multi-layer training. The residual blocks improve the efficiency of CNN with more neural layers while minimizing error. As a result, the skip connections make it possible to train much deeper networks and these additional layers help solve complex problems more efficiently, as the different layers could be trained for varying tasks to get highly accurate results, producing results substantially better than the earlier models.

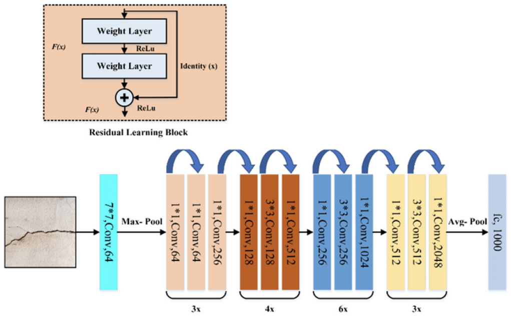


Figure 8: Architecture of ResNet-50 (Ali et al. 2021)

Pre-processing and Training

The digital USU data consists of mostly negative data: 8484 positives and 47,608 negatives. We have augmented the positive examples by modifying the image's brightness, doubling the positive examples numbers to 16,968.

Furthermore, we have opted for stratified sampling for data splitting to prevent the skewness of data and ensure that the training model has sufficient positive training examples. On top of that, 20% of the training data was used to validate our model's performance.

The dimensions of our training images have 256 × 256 do not match our models' input size, we would need to perform one of the following pre-processings:

1. Resize the image to fit the dimensions (224 × 224)
2. Crop image into several chunks of the desired dimension

Although the second option is more ideal as it preserves maximal pixel-level features, cropping the image into 4 parts (to capture the whole image) would make our training time increase by 4 folds. Hence, the first option of resizing the data was selected to improve our models' time efficiency. Our image pre-processing was done using `ImageDataGenerator`.

20% of the training data was used to validate our model's performance. To increase our training speed, we trained our model using mini-batches of size 64. The models were trained with a maximum of 100 epochs with an early stop if the training loss is no longer decreasing. Through experiments, we found that the AlexNet model trains for 3 epochs before it stops, while VGG-16 trains for 9 epochs and the ResNet-50 trains for 12 epochs.

Testing

The $accuracy$, $recall$, $precision$ and F_1 score based on the model performance on the unseen METU images were calculated using the following formulae:

$$accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{3}$$

$$precision = \frac{T_P}{T_P + F_P} \tag{4}$$

$$recall = \frac{T_P}{T_P + F_N} \tag{5}$$

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

The cost of false negatives is not tolerable due to massive monetary loss and potential loss of lives. Therefore, $recall$ is the metric of focus. We will choose the model with the highest $recall$ score such that its false negative rate is kept below the threshold of 5% which is the human error rate.

We used Google Colab Pro (Intel® Xeon Haswell 2 cores, 24GB RAM, Nvidia Tesla P100 GPU) to train and infer the images. We also tracked the time needed for the model needed to infer 40,000 images. With a sufficiently large images size, we believed that this could approximate a real-time prediction if the model is implemented.

Results

Recall and accuracy scores Comparing across the 3 trials, AlexNet outputs the best $recall$ score consistently and kept the false-negative rate less than the human error threshold of 5%. However, its prediction $accuracy$ and $precision$ are also the lowest. When implemented, the AlexNet model would be the best in minimising the number of misclassified cracked surfaces, but a large number of false positives defeat the model's objective. As the AlexNet model is less than 50% accurate across all 3 data sets, the aim of the model saving crack surveillance cost would not be met.

Model	Unmodified				Blurred				Brightness-Adjusted			
	Accr	Prec	Recall	F_1	Accr	Prec	Recall	F_1	Accr	Prec	Recall	F_1
AlexNet	0.490	0.495	0.963	0.654	0.486	0.493	0.957	0.651	0.486	0.493	0.945	0.648
VGG-16	0.813	0.972	0.644	0.775	0.665	0.993	0.333	0.499	0.824	0.935	0.700	0.798
ResNet-50	0.795	0.882	0.680	0.768	0.687	0.781	0.518	0.623	0.766	0.772	0.755	0.763

Table 1: Performance of models across different types of images (Accr: Accuracy; Prec: Precision)

A possible reason behind the low accuracy of the AlexNet model is due to the absence of pre-trained weights from "imagenets" in the decision function to make the model discriminative. In comparison, VGG-16 and ResNet-50 incorporated pre-trained weights and performed transfer learning through the convolutional layers to the training images, yielding much higher accuracy and precision scores.

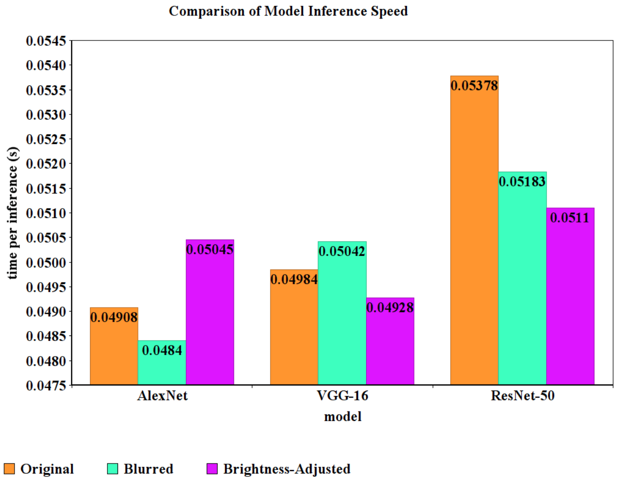


Figure 9: Bar chart of inference speed

Computational time The number of epochs corresponds to the amount of time needed to train the model. The AlexNet model, VGG-16 and ResNet-50 were trained for 3 epochs, 9 epochs and 12 epochs respectively. Hence, the time efficiency of AlexNet is highest followed by VGG-16 and ResNet-50. Across the 3 trials, the comparison of inference speed is generally consistent with the epochs relationship. However, the AlexNet model has a longer inference time with the brightness-adjusted images.

Conclusion

The novelty of our model stems from the training process, where a variety of augmentations were considered and used in our data. The augmentations have 2 benefits. Firstly, the trained models are more robust towards classifying cracks in a wide range of situations. This makes our model more advantageous compared to other training methodologies in the field of crack classification. Although the results of the study were lower than the actual accuracy rates stated in their respective research papers, the findings in this report are valuable as they depict the realistic performances of these models.

Secondly, by considering the augmentations and accounting for more cases, the training process could overcome the issue of under-fitting caused by the lack of open-source data sets available. The image augmentations mimic the shakiness of drones and pictures taken in the dark. This reduces the bias inherited by only training clear pictures taken in board daylight, which was seen in many similar research papers on crack classification literature.

In the context of Singapore's application, the AlexNet model is the most applicable model. It takes the shortest amount of time to infer new images compared to the other two models and has the highest $recall$. However, the model needs to improve on its $accuracy$ and $precision$ to have similar $accuracy$ and $precision$ values as VGG-16 and ResNet-50 models. Based on the Table Inspection, the addition of pre-trained weights from "imagenets" in the decision function yields better $accuracy$ and $precision$. If AlexNet is trained with the pre-trained weights, the model will possibly score well for $accuracy$ and $precision$ while maintaining a low false negative score.

The drones application can be complemented with Google API (Mandal, Uong, and Adu-Gyamfi 2019). The trained AlexNet model is hosted on online cloud services such as Google Clouds Services or Amazon Web Services. Google API can scan and identify cracks of buildings or infrastructures with satellite imaging. If the Google satellite images are high resolution enough, the captured images can be broken down into smaller images and used for predictions. The result will then be sent back to the BCA for analysis.

We hope that this report helps to identify the most suitable deep learning algorithm for Singapore's context. As the models were trained with data from other countries, the collection of local concrete images would greatly optimise the deep learning models.

All in all, we believe that deep learning bridges the gap between Singapore's Smart Nation vision and the built sector. The AlexNet model may help Singapore to improve the built environmental landscape, creating a smarter and safer environment for Singapore residents to live in.

Reflection

During the research process, we met and overcame the following challenges:

1. The open-source data set is imbalanced in the positive and negative distributions, hindering our model training process significantly. Random Sampling does not ensure the even representation of positive and negative images, causing the model inference to be wrong. We switched to stratified sampling; Then, we augmented the data by

playing with image resizing and obtained more positive images.

2. With a large set of images, we were limited by the processing power of Google Colab. Testing the different model parameters became more difficult and time-consuming. Hence, we have subscribed to the Google Colab Pro to obtain less disruptive training time. We also exploited the power of "parallel computing" by splitting the training data role evenly among the members to minimise the downtime.

This project is tough, yet rewarding. It marks the beginning of our venture into deep learning. We have gained valuable insights into state-of-the-art deep learning models for image classification in the deep-learning field and realised that there are still ongoing innovations in the deep-learning models to create a model that is robust and highly accurate.

Data Availability

To support the findings of this report, all data were available from the corresponding author upon request.

Web Links

Github Repository:
https://github.com/DavidTan0527/CS3244-Project
Google Drive with data used:
https://tinyurl.com/2357j2u4

Work Distribution

Ang Bo Yuan implemented the AlexNet model. Loo Guan Yee created the code for the VGG-16 and ResNet Model and wrote on the ResNet model technique. Michelle Yong performed the cleaning of the data, report proofreading and Novelty write-up. Lastly, Tan Weiu Cheng created functions to blur/modify the image and create plots and equations in LaTex.

References

Ali, L.; Alnajjar, F. S.; Jassmi, H. A.; and Serhani, M. A. 2021. Figure 6. the architecture of resnet-50 model.

Anwar, A. 2022. Difference between alexnet, vggnet, ResNet and inception.

Biscione, V.; and Bowers, J. 2020. Learning translation invariance in cnns.

Boesch, G. 2022. Deep residual networks (ResNet, RESNET50) - guide in 2022.

Dorafshan, S.; Thomas, R. J.; and Maguire, M. 2018. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, 186: 1031–1045.

Dwivedi, P. 2019. Understanding and coding a ResNet in Keras.

Girshick, R. 2015. Fast R-CNN.

HDB. 2022. Precast Technology.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition.

Hoang, N.-D. 2018. Image processing-based recognition of wall defects using machine learning approaches and steerable filters.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4): 541–551.

Li, S.; and Zhao, X. 2019. Image-based concrete crack detection using convolutional neural network and exhaustive search technique. *Advances in Civil Engineering*, 2019: 1–12.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation.

LTA. 2016. Land Transport master plan 2040.

Maguire, M.; Dorafshan, S.; and Thomas, R. J. ???? SDNET2018: A concrete crack image dataset for Machine Learning Applications.

Mandal, V.; Uong, L.; and Adu-Gyamfi, Y. 2019. Automated road crack detection using deep convolutional neural networks. *2018 IEEE International Conference on Big Data (Big Data)*.

MND. 2021. Written Answer by Ministry of National Development on consumption of concrete in Singapore for the last five years.

Ren, Y.; Huang, J.; Hong, Z.; Lu, W.; Yin, J.; Zou, L.; and Shen, X. 2020. Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Construction and Building Materials*, 234: 1–12.

Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition.

Team, G. L. 2022. Introduction to resnet or residual network.

Varshney, P. 2020. Alexnet Architecture: A complete guide.

Wei, J. 2020. Alexnet: The architecture that challenged CNNS.

Yasahan, W.; Yue, a. Y. Z., Zhang; Zhao, L.; Xian, S.; and Guo. 2019. Towards Scale-Aware Rotated Object Detection in Aerial Imagery. *IEEE explore*, 7: 173855–173865.

Özgenel, F. 2019. Concrete crack images for classification.

Özgenel, F.; and Sorguç, A. G. 2018. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. *2018 Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*, 693–700.