AUSTROADS BRIDGE CONFERENCE 2022

Peer reviewed paper

Hosted by the Department for Infrastructure and Transport, South Australia

Bridge CNN Defect Prediction Models Using Existing Image Data

L Kemp, Mrs, Council for Scientific and Industrial Research (CSIR), University of Pretoria

MP Roux, Mr, Council for Scientific and Industrial Research (CSIR)

WJvdM Steyn, Prof, University of Pretoria

Abstract

In South Africa, it is a requirement for all road agencies to conduct principal visual inspections of all bridge structures every five years. Smaller municipalities do not always have the necessary funds available for principal bridge inspections, resulting in either bridge inspections not being executed, or inspections being done by unqualified people.

This paper intends to investigate the possibility of using existing bridge inventory and inspection image data to develop Convolutional Neural Network (CNN) models to predict and classify bridge defects autonomously. This research aims to improve the quality of bridge inspections and condition ratings assigned to defects to be more consistent and not reliant on human subjectivity. These models could ultimately be used for quality control in a Bridge Management System (BMS).

The CSIR STRUMAN BMS contains inspection and inventory images captured during principal visual bridge inspections. As a proof-of-concept, bridge roadway joints were considered. 600 images of bridge roadway joints captured in the system were classified according to Defect and No Defect datasets. Different CNN classification models were developed to predict whether an image of a bridge roadway joint contained a defect or not. The image datasets were used to train, validate, and test the performance of the CNN models. The performance of the CNN models was evaluated using a Confusion Matrix and Classification report to select the best-performing model. In conclusion, the selected model was evaluated when introduced to new unseen images.

The best performing CNN model utilised transfer learning and data augmentation to predict with 95% accuracy from images if a bridge roadway joint had a defect and with 65% accuracy if the bridge roadway joint had no defect.

Keywords: CNN Models, Bridge Inspections, Defect Prediction, BMS

1. Introduction

The South African road network has a total length of 750 000 kilometres, the longest road network in Africa and the tenth longest in the world. The road network is valued at more than US$ 125.6 billion [1]. It is thus critical for all road agencies to protect and maintain this asset and all road-related structures.

Principal visual inspections for all bridge structures are required every five years as prescribed in TMH19 [2] [3] . Bridge and senior bridge inspectors are highly qualified and experienced persons. For such individuals to inspect all bridge and major culvert structures in a defined region, is a time-consuming and costly exercise. Smaller municipalities and metros do not always have adequate budget allocation for principal bridge inspections every 5 years. This result in either bridge inspections not being executed, or inspections being done by unqualified people.

The Council for Scientific and Industrial Research (CSIR) has been investigating the use of Fourth Industrial Revolution (4IR) technologies aiming to improve the visual bridge inspection methodology in South Africa [4]. To date, a proof-of-concept study has been conducted to incorporate the use of an Unmanned Arial Vehicle (UAV) to capture bridge image data. The captured images were processed to create point cloud models. Accredited bridge inspectors attempted to identify defects and complete inspection sheets using only the point cloud models and captured images as a proposed new inspection methodology. The study concluded that visual bridge inspections can be performed using only point cloud models and images, but there are limitations. The new inspection methodology can be used as a screening process for inspectors to determine if a structure requires further onsite inspection or not [5]. As part of the study, the cost and time components of the new inspection methodology versus traditional TMH19 inspections were compared. The cost and time components related to the new inspection methodology did not prove to have any significant benefits.

The time- and cost-saving aspect of the new inspection methodology will depend on limiting the human aspect of inspections. The objective of this paper is to determine if it is possible to detect and classify bridge defects autonomously from existing image data by applying deep learning and computer vision techniques. The CSIR STRUMAN Bridge Management System (BMS) contains inspection and inventory images captured during principal visual bridge inspections. As a proof of concept, bridge roadway joints were considered. Different Convolutional Neural Network (CNN) classification models were developed to predict whether an image of a bridge roadway joint contained a defect or not.

This research aims to improve the quality of bridge inspections and condition ratings assigned to defects to be more consistent and not reliant on human subjectivity. These models could ultimately be used for quality control in a BMS.

2. Background

An international survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures has been conducted in 2009 in the USA. The study noted that among the possible techniques for inspecting civil infrastructure, the use of optical instrumentation relying on image processing is less time-consuming and an inexpensive alternative to current (traditional) monitoring methods [6].

Several image processing techniques, including enhancement, noise removal, registration, edge detection, line detection, morphological functions, colour analysis, texture detection, wavelet transform, segmentation, clustering and pattern recognition, are key pieces that should be merged to solve this problem [6].

The rapid evolvement of technology creates the possibility to achieve what was previously considered a limitation. A more recent study conducted in 2020 in the US, focused on streamlined bridge inspection systems, utilising UAVs and machine learning applications [7].

The study proposed advanced data analytics tools to automatically [7]:

1. Identify type, extent, growth, and 3D location of defects using computer vision techniques,
2. Generate a 3D point-cloud model and segment structural elements using human-in-the-loop machine learning, and
3. Establish a geo-referenced element-wise as-built bridge information model to document and visualize damage information.

As demonstrated in the USA study, most image processing approaches are limited to detecting only one type of defect at a time. The US study presented and evaluated the steps and algorithms that are necessary for detecting various changes simultaneously through digital image processing and the introduction of machine learning [7]. A schematic illustration of the proposed automated bridge inspection system is shown in Figure 1.

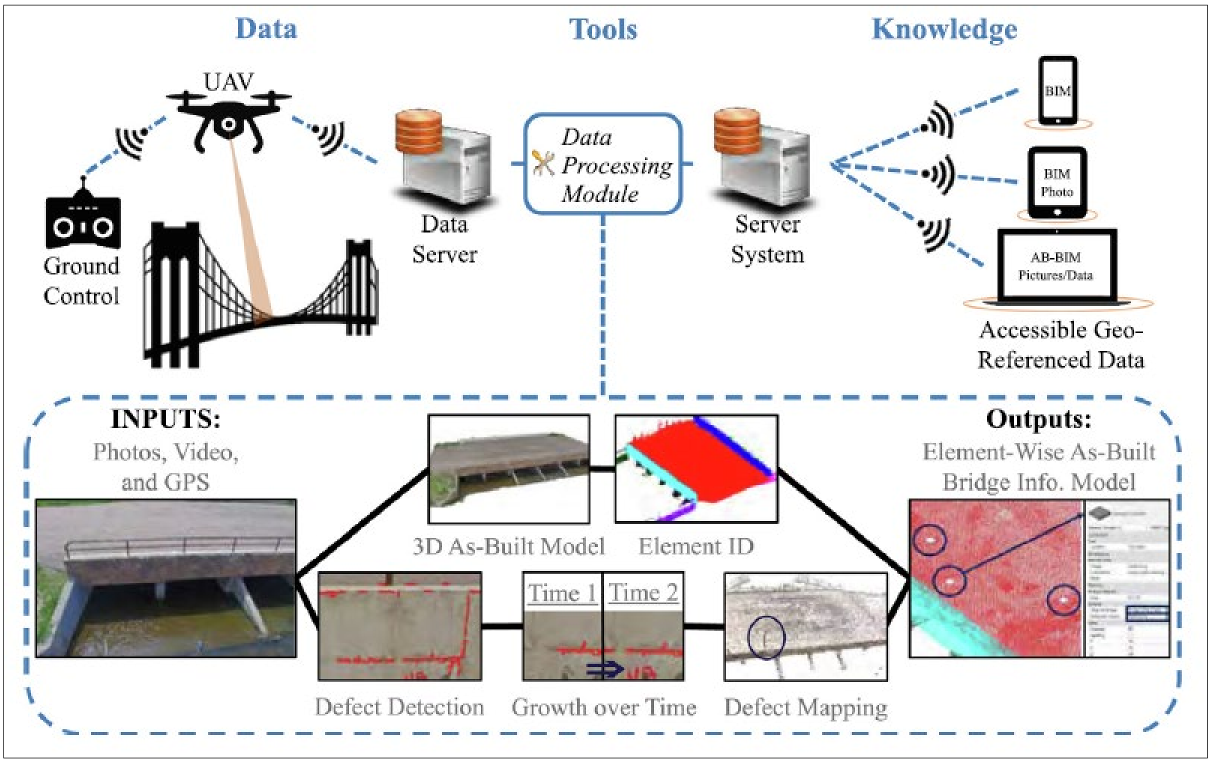

Figure 1: Proposed automated bridge inspection system [7]

The advancement of deep learning, as a branch of machine learning, and a CNN developed with more hidden layers and a more complex network structure, has more powerful feature learning and feature expression abilities than traditional machine learning methods [8].

CNN is a multi-layer artificial neural network specially designed to handle two-dimensional (image) input data. The use of CNN allows for learning and extracting relevant features while eliminating the need for a complex modelling process. The use of CNN models for image classification, object detection, attitude estimation and image segmentation, has delivered good results and progress in this field [8].

The accuracy of the CNN architecture highly depends upon three factors namely, large scale database, high-end computational unit and the network depth. The requirement of training large databases is solved due to availability of public databases. Transfer learning can be used to fine-tune the pre-trained network parameters, obtained from training large databases, for an image classification task. To improve the recognition accuracy further toward a human vision system, researchers proposed deeper CNN architectures and developed the VGG16 architecture for object recognition tasks [9].

For image classification tasks, it is necessary to expand the insufficient training image samples through various data augmentation methods. To avoid overfitting a large amount of labelled data is required to train CNN models. In the case of insufficient training data, regularization technologies are commonly used to prevent overfitting, such as Dropout, Batch Normalization and data augmentation. Data augmentation refers to the process of creating new similar samples of the training set through employing random crop, horizontal flip, rotation, shifting, colour jittering, addition of noise and Principal Component Analysis (PCA) jittering [10].

A study conducted by Utah State University investigated the feasibility and application of deep learning CNN for UAV-assisted structural inspections of concrete decks. The training dataset consisted of lab-made bridge deck images with cracks. The study concluded that it is feasible to apply deep learning CNNs in autonomous civil structural inspections with comparable results to human inspectors. The results indicated that the fully trained dataset had a validation accuracy of 94.7% and a validation accuracy of 97.7% when using transfer learning [11].

3. Methodology

To determine if existing inspection and inventory image data could be used to identify defects autonomously, deep learning models were developed to categorise images and predict if a bridge element had a defect or not. The bridge element selected for this study was bridge roadway joints. Different CNN models were developed and evaluated to optimise the performance and ultimately select the best-suited model.

CNN is a class of deep neural networks to analyse visual imagery. CNN allows for extraction of higher representation content of images. Unlike conventional image recognition where image features are defined manually, a CNN uses raw pixel data of images, trains the model, and extracts the features automatically for more accurate classification.

The CSIR STRUMAN BMS inspection and inventory image data were used to compile images of roadway joints of different types and sizes of bridges in South Africa. The images were categorised according to Defect and No Defect classes. A total of 600 images were used, 400 images belonging to the Defect class and 200 images to the No Defect class. Examples of images from the two classes are shown in Figure 2 and Figure 3.

Figure 2: Examples of images in the Defect class

Figure 3: Examples of images in the No Defect class

The CNN classification models were developed using the Spider programming environment written in Python and built-in libraries such as Keras and Sklearn were utilised. These libraries contain efficient tools used in machine learning and statistical modelling, including classification.

The data was split into three subsets used for training, validation and testing. 80% of the data were used for training and 20% used for testing. The training set, containing representative data of Defect and No Defect roadway joint images, was further split into 80% training and 20% validation datasets. The test dataset was not included in any training or validation stages and only introduced in the final testing stage to determine the performance of the model when introduced to new unseen data.

It is important for the Defect and No Defect classes to be balanced and have the same number of images during training, to avoid any bias when predicting a class. Data augmentation, a computer vision technique, was used to extend and balance the No Defect dataset. Images from the No Defect class were flipped, shifted and rotated at different angles to create more images, equal to the number of images in the Defect class.

The first model developed, referred to as the baseline model, was a simple CNN model. The model consisted of two convulsion layers, two pooling layers and a dense output layer. Since this is 'n binary classification problem with only two classes, activation functions ReLU was defined for the intermediate layers, Sigmoid for the output layer and Binary Cross-entropy as the Loss function. The Loss function evaluates how well the specific algorithm models the given data. The model was trained and validated with the respective datasets for 50 epochs and the performance of the model was evaluated with a Confusion Matrix and Classification report to summarise the prediction results. The baseline model developed is shown in Figure 4.

```
#Define Baseline Model

Baseline_model = keras.Sequential([
            keras.layers.Conv2D(16, kernel_size=(3,3), activation='relu',input_shape=(224,224,3)),
            keras.layers.MaxPool2D(),
            keras.layers.Conv2D(32, kernel_size=(3,3), activation='relu'),
            keras.layers.MaxPool2D(),
            keras.layers.Flatten(),
            keras.layers.Dense(1,activation='sigmoid')])

Baseline_model.compile(optimizer=keras.optimizers.Adam(),
              loss='binary_crossentropy',
              metrics= ['accuracy'])

history = Baseline_model.fit(train_data, epochs=50,validation_data=val_data, shuffle=True)

#Evaluate Results

predictions = []
labels =  []

for x, y in val_data:
  Y_pred = (Baseline_model.predict(x) > 0.5).astype("int32")
  actual = y.numpy()
  predictions = np.concatenate([predictions,Y_pred[:,-1]], axis= 0)
  labels = np.concatenate([labels,actual], axis= 0)


print(confusion_matrix(labels, predictions))
print(classification_report(labels, predictions, target_names = ['Defect (Class 0)','No Defect (Class 1)']))
```

Figure 4: CNN Baseline Model

The second model developed was a more complex CNN model with deeper layers. The model consisted of three convulsion layers, three pooling layers and a dense output layer. ‘Dropout’ and ‘Early Stopping’ were introduced to avoid the model from overfitting on the training data, saving the weights of the model when validation loss was at a minimum. The same activation and Loss functions were defined as specified in the baseline model. The model trained for 12 epochs before early stopping was called, as the validation accuracy did not improve for 5 consecutive epochs. The performance of the model was evaluated with a Confusion Matrix and Classification report. The more complex CNN model developed is shown in Figure 5.

```
#Define more complex CNN Model

CNN_model = keras.Sequential([
            keras.layers.Conv2D(32, kernel_size=(3,3), activation='relu',input_shape=(224,224,3)),
            keras.layers.MaxPool2D(),
            keras.layers.Conv2D(64, kernel_size=(3,3), activation='relu'),
            keras.layers.MaxPool2D(),
            keras.layers.Conv2D(125, kernel_size=(3,3), activation='relu'),
            keras.layers.MaxPool2D(),
            keras.layers.Flatten(),
            keras.layers.Dropout(0.5),
            keras.layers.Dense(1,activation='sigmoid')])

CNN_model.compile(optimizer=keras.optimizers.Adam(),
              loss='binary_crossentropy',
              metrics= ['accuracy'])

es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=5)
mc = ModelCheckpoint('best_model.h5', monitor='val_accuracy', mode='max', verbose=1, save_best_only=True)

history = CNN_model.fit(train_data, epochs=50,validation_data=val_data, shuffle=True, callbacks=[es,mc])
saved_model = keras.models.load_model('best_model.h5')

#Evaluate Results

predictions = []
labels =  []

for x, y in val_data:
  Y_pred = (saved_model.predict(x) > 0.5).astype("int32")
  actual = y.numpy()
  predictions = np.concatenate([predictions,Y_pred[:,-1]], axis= 0)
  labels = np.concatenate([labels,actual], axis= 0)


print(confusion_matrix(labels, predictions))
print(classification_report(labels, predictions, target_names = ['Defect (Class 0)','No Defect (Class 1)']))
```

Figure 5: More complex CNN model

For the third model, transfer learning was introduced. Transfer learning utilises pre-trained weights of a previous model developed with large datasets and more classes. Transfer learning can be used by freezing the early convolutional layers of the network and only specifying and training the last few layers and output layer, to reduce the computation time and to make predictions based on the requirements of the current dataset. The transfer learning model used was the VGG16 model developed with the ImageNet dataset. ‘Dropout’ and ‘Early Stopping’ were included, and the same activation and Loss functions were defined as specified in the baseline model. The model trained for 10 epochs before early stopping was called. The performance of the model was evaluated with a Confusion Matrix and Classification report. The transfer learning VGG16 model developed is shown in Figure 6.

```
VGG16_model = tf.keras.applications.VGG16(input_shape = (224, 224, 3), include_top = False, weights = "imagenet")
VGG16_model.trainable = False

TL_model = tf.keras.Sequential([VGG16_model,
                                keras.layers.Flatten(),
                                keras.layers.Dense(100,activation='relu'),
                                keras.layers.Dropout(0.5),
                                keras.layers.BatchNormalization(),
                                keras.layers.Dense(1,activation='sigmoid')])


TL_model.compile(optimizer=keras.optimizers.Adam(),
              loss='binary_crossentropy',
              metrics= ['accuracy'])

es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=5)
mc = ModelCheckpoint('best_model.h5', monitor='val_accuracy', mode='max', verbose=1, save_best_only=True)

history = TL_model.fit(train_data, epochs=50,validation_data=val_data, shuffle=True, callbacks=[es,mc])
saved_model = keras.models.load_model('best_model.h5')

#Evaluate Results

predictions = []
labels =  []

for x, y in val_data:
  Y_pred = (saved_model.predict(x) > 0.5).astype("int32")
  actual = y.numpy()
  predictions = np.concatenate([predictions,Y_pred[:,-1]], axis= 0)
  labels = np.concatenate([labels,actual], axis= 0)


print(confusion_matrix(labels, predictions))
print(classification_report(labels, predictions, target_names = ['Defect (Class 0)','No Defect (Class 1)']))
```

Figure 6: Transfer Learning VGG16 model

For the final model, an even larger dataset was created using more data augmentations to increase the number of images for each class to 3 600 images. Each image in the Defect class was augmented eight times and the images in the No Defect class 17 times to ensure the datasets were balanced. The VGG16 transfer learning model developed was used and retrained with the larger dataset. The model trained for nine epochs before early stopping was called. The performance of the model was evaluated with a Confusion Matrix and Classification report.

Finally, the test dataset was fitted to the VGG16 transfer learning model, trained with the larger dataset, to predict the classes of the unseen images. The predicted class of each image was compared to the actual class to evaluate the final performance of the model.

4. Results

The performance of each CNN model was evaluated through a Confusion Matrix, Classification report and by considering the training and validation accuracy and loss. The evaluation metrics of the different models were compared to select the best-suited model. The best model was then used to predict the classes of the test dataset.

Training and validation accuracy

The accuracy of the training and validation datasets was recorded for each epoch in the training and validation stage. The accuracy of the models was calculated based on the model's ability to predict the correct class of an image. The training and validation accuracies for the different models for each epoch are shown in Figure 7.

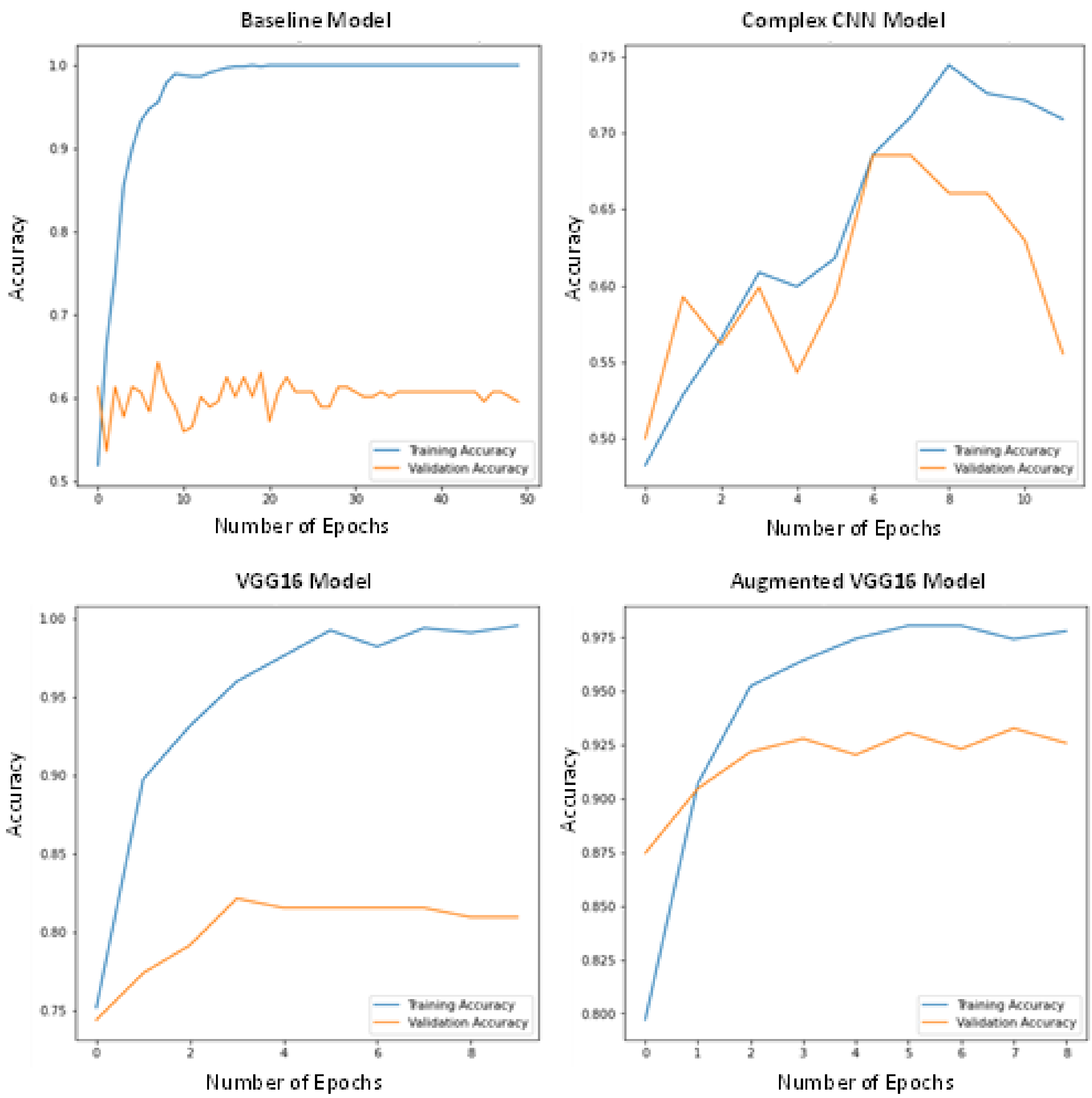

Figure 7: CNN model training and validation accuracy

Training and validation loss

The Loss function compares each of the predicted probabilities of a class, to the actual class output, which is 0 or 1 for Binary Cross-entropy, during the training and validation stage. It then calculates the score that penalises the probabilities based on the distance from the expected value. The loss is thus a measure of how close or far the actual value is from the predicted value.

The loss was recorded for each epoch in the training and validation stage. The training and validation loss for the different models for each epoch is shown in Figure 8.

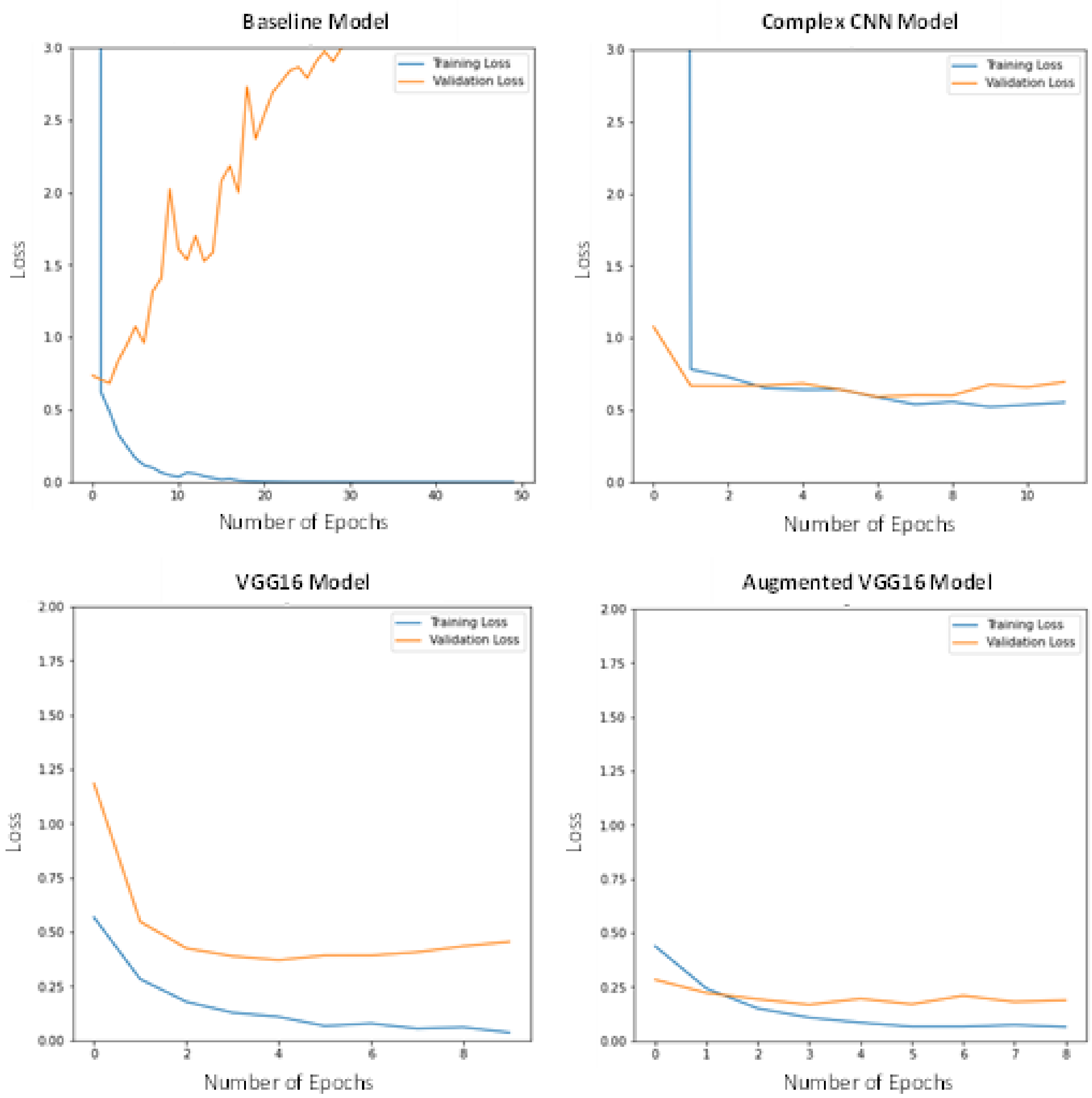

Figure 8: CNN model training and validation loss

Confusion Matrix

A Confusion Matrix is a summary of the model predictions and provides insight into the accuracy and precision of predicted classes versus the actual class. After the model had been trained, the validation dataset was fitted to the model and the predicted and actual classes were recorded for each image. A Confusion Matrix for each of the validation predictions and actual classes were constructed for each model. The Confusion Matrix for each model is shown in Table 1.

Table 1: Confusion Matrix of CNN models

Baseline Model			**Complex CNN Model**		
	Predicted Defect	Predicted No Defect		Predicted Defect	Predicted No Defect
Actual Class Defect	**55**	**28**	Actual Class Defect	**27**	**56**
Actual Class No Defect	**40**	**45**	Actual Class No Defect	**5**	**80**
VGG16 Model			**Augmented VGG16 Model**		
	Predicted Defect	Predicted No Defect		Predicted Defect	Predicted No Defect
Actual Class Defect	**65**	**18**	Actual Class Defect	**670**	**63**
Actual Class No Defect	**12**	**73**	Actual Class No Defect	**36**	**697**

Classification report

A Classification report is used to measure the quality of the predictions from a classification algorithm. The report summarises the predictions in terms of precision, recall and F1-score. The metrics of a classification report use the predictions made for each class. Precision is the percentage of the correct predictions, recall is the fraction of the correctly identified classes and the F1-score is the weighted harmonic mean of precision and recall.

A Classification report was generated for each of the models, based on the predictions made on the validation dataset. A summary of the results for each model and class is presented in Figure 9.

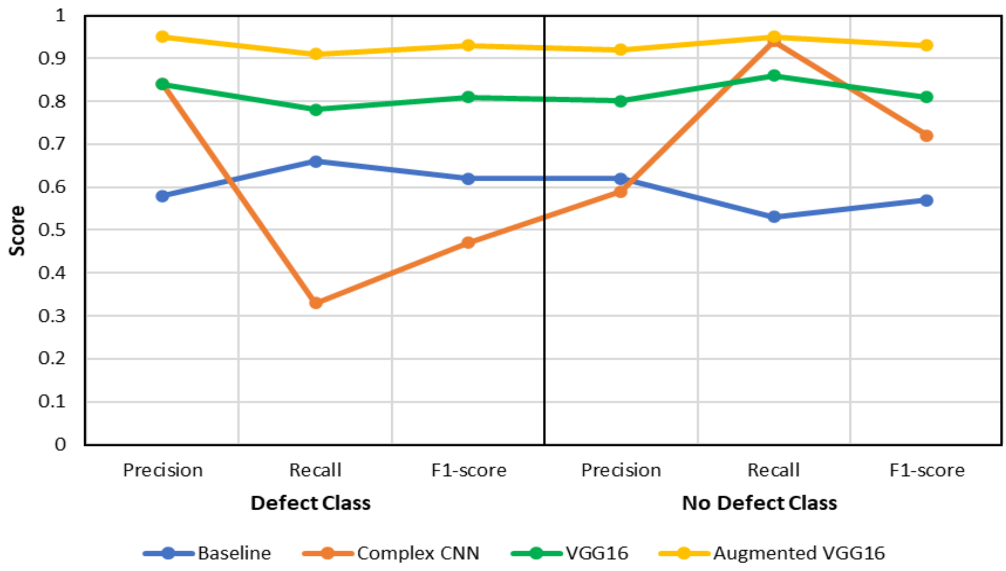

Figure 9: Classification report results

Test dataset

As a final measure of the model's performance, the test dataset was fitted to the VGG16 model, and trained with the larger augmented dataset. This evaluated the model's ability to predict the correct class of unseen images.

The test dataset had 80 Defect images and 40 No Defect images. A Confusion Matrix and Classification report were generated to evaluate the results as shown in Table 2 and Table 3.

Table 2: Confusion Matrix for the test dataset

	Predicted Defect	Predicted No Defect
Actual Class Defect	**76**	**4**
Actual Class No Defect	**14**	**26**

Table 3: Classification Report for the test dataset

Model	**Precision**		**Recall**		**F1-score**	
	Defect	No Defect	Defect	No Defect	Defect	No Defect
VGG16 Augmented	0.95	0.65	0.84	0.87	0.89	0.74

5. Discussion of results

The cost and time-saving aspects of the new inspection methodology will depend on limiting human involvement in inspections. Autonomous defect detection for bridge inspections will reduce the time spent on inspections and improve the quality and constancy of bridge inspections. This study focussed on using existing inspection and inventory image data to predict and classify bridge roadway joint defects autonomously. To compare the different CNN models the performance of each model was evaluated.

The first model developed was the Baseline model which had a maximum validation accuracy of 64%. Evaluating the validation loss, the baseline model was overfitting on the training data as the validation loss continued to increase for each epoch. These results were confirmed in the Confusion Matrix and Classification report.

For the second model, an additional convolution layer was added to increase the complexity. The Complex CNN model had a maximum validation accuracy of 68%. The addition of a Drop Out layer and Early Stopping prevented the model from overfitting as indicated in the validation loss. Considering the Confusion Matrix and Classification report, the model performed better than the Baseline model in predicting the No Defect class but performed worse in predicting the Defect class.

The third model utilised the pretrained weights of the VGG16 transfer learning model. The model had a validation accuracy of 82%. The Drop Out layer and Early Stopping prevented the model from overfitting, as confirmed in the validation loss results. The Confusion Matrix and Classification report indicated the performance of the model is balanced in predicting either the No Defect or Defect class.

For the final model, the validation and training datasets were augmented to increase the number of images. The VGG16 model was retrained with the larger dataset and had the best validation accuracy of 93% and lowest validation loss compared to the other models. The Confusion Matrix and Classification report indicated the model predicting the classes of the validation dataset with 95% precision for the Defect class and 92% for the No Defect class. The Augmented VGG16 model was selected as the best performing model for predicting whether a bridge roadway joint image included a defect or not.

To evaluate the model performance when introduced to unseen images, the test dataset containing 80 images of bridge roadway joints with defects and 40 images of non-defected bridge roadway joints, were fitted to the Augmented VGG16 model. The model classified 95% of the Defect images correctly and 65% of the No Defect images correctly.

Evaluating the different model performances indicates the positive effect techniques such as Drop Out, Early Stopping and data augmentation have on the performance of the model. The model validation accuracy increased from 64% to 93%. The need for larger datasets with more representative images is evident in the performance of the model when introduced to new unseen images.

6. Conclusions

The new inspection methodology will have cost and time-saving benefits if human involvement could be limited. It is possible to detect and classify bridge defects autonomously, using existing image data and applying deep learning and computer vision techniques, if large datasets are available.

Different CNN models were developed to predict whether an image of a bridge roadway joint included a defect or not. The best performing model, VGG16 Augmented model, could predict with 95% accuracy if an image included a defect and with 65% accuracy if an image did not include a defect.

The model can only predict the class of an image based on the data used during training. The original dataset was relatively small to develop an accurate CNN model. Transfer learning and data augmentation improved the performance, but more representative images are required to increase the accuracy of the model when predicting classes of unseen data, especially for the No Defect class.

The VGG16 Augmented CNN model developed could be improved by increasing the dataset with more representative images. Different transfer learning models could be explored to better fit the requirements of the dataset. The individual layers in the CNN of the VGG16 Augmented model could be investigated to ensure the correct features are extracted from the image and could be adjusted accordingly.

The model could be further developed by introducing more classes for different rated defects. Experienced bridge inspectors could assist in defining these classes, ensuring consistent defect ratings. The model could then be used to predict the ratings of defects and not be dependent on human subjectivity.

Different CNN models could be developed for each bridge element and incorporated into the new inspection methodology to identify and predict defect ratings based on the images captured using a UAV, advancing into complete autonomous defect detection and rating.

7. References

[1] N. Treasury, “Provincial Budget and Expenditure Review: 2015/16 - 2022/23,” National Treasury, Pretoria, 2021.

[2] COTO, “TMH 19 Manual for the visual assessment of road structures. Part A: Road structure management information,” The South African National Roads Agency SOC Limited, Pretoria, 2020a.

[3] COTO, “TMH 19 Manual for the visual assessment of road structures. Part B: Visual assessment guide,” The South African National Roads Agency SOC Limited, Pretoria, 2020b.

[4] L. Kemp, M. P. Roux, M. Kemp and R. Kock, “Application of Drones and Image Processing for Bridge Inspections,” in *Proceedings of the 39th Southern African Transport Conference, 5-8 July 2021*, Pretoria, 2021.

[5] L. Kemp, W. Steyn and M. Roux, “Enhanced Methodology for Visual Bridge Inspections in South Africa,” *SAICE Journal,* 2022.

[6] M. R. Jahanshahi, J. S. Kelly, S. F. Masri and S. Gaurav, “A survey and evaluation of promising approaches for automatic image-based,” *Structure and Infrastructure Engineering,* vol. 5, no. 6, pp. 455-486, 12 August 2009.

[7] B. J. Perry, Y. Guo, R. Atadero and J. W. van de Lindt, “Streamlined bridge inspection system utilizing unmanned aerial vehicles (UAVs) and machine learning,” *Measurement,* vol. 164, November 2020.

[8] A. A. M. Al-Saffar, H. Tao and M. A. Talab, “Review of Deep Convolution Neural Network in Image Classification,” in *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications, 23-14 October*, Indonesia, 2017.

[9] M. Shaha and M. Pawar, “Transfer learning for image classification,” in *2nd International Conference on Electronics, Communication and Aerospace Technology. 29-31 March*, India, 2018.

[10] J. Shijie, W. Ping, J. Peiyi and H. Siping, “Research on Data Augmentation for Image Classification Based on Convolution Neural Networks,” in *Chinese Automation Congress, 20-22 October 2017*, China, 2017.

[11] S. Dorafshan, R. Thomas, C. Coopmans and M. Maguire, “Deep Learning Neural Networks for UAS-Assisted Structural Inspections: Feasibility and Application,” in *International Conference on Unmanned Aircraft Systems, 12-15 June 2018*, Dallas, TX, USA, 2018.

[12] S. Dorafshan and M. Maguire, “Bridge inspection: human performance, unmanned aerial systems and automation,” *Journal of Civil Health Monitoring,* vol. 8, pp. 443-476, 21 May 2018.

Author contacts

Lana Kemp, Council for Scientific and Industrial Research (CSIR), Senior Civil Engineer/ Researcher

lkemp@csir.co.za , linkedin.com/in/lana-ronnenbergh-132196126

Michael Roux, Council for Scientific and Industrial Research (CSIR), Principal Researcher

mproux@csir.co.za

Wynand Steyn, University of Pretoria, Head of Department Civil Engineering

wynand.steyn@up.ac.za