VISUAL RATING SYSTEM

INTRODUCTION

The conditional assessment of bridges has to steps:

1. Visual Inspection
2. Structural Design Check

The visual inspection process is characterized by the capturing of photographs of the defects in the bridges and making a manual compilation of these photographs in correspondence with the members it is found in. The process also involves the visual ratification of the bridge upon inspection by the expert. Both these ideas are used in coalescence to provide a final assessment on the bridge. This approach could possibly be prone to errors due to the lack of a systematic execution of the process and the excessive time consumption due to its requirement of manual inputs which could lead to missing images, wrongful correspondence of the image to the member etc..

This project aims to minimize these errors by reducing the time it takes to complete the visual inspection process. The photos of each member can be taken in a systematic manner and each defect can be accounted for with correspondence to the member in which it is located. An artificial neural network is used to identify these defects and record them to the member. Each member would be accounted for in this process and the final condition rating of the bridges could be given based on the number of defects in these members through a severity check. Also, it would provide a holistic understanding to the end user. Since the visual inspection process is a very important part of the conditional assessment of the bridges wherein it plays the role of providing a basic understanding of the bridge by eliminating the role of logistics for every person concerned, the representation of it to the experts must be detail-oriented and a database is essential.

DESIGN CODE FORMAT

1. INPUT OF BRIDGE SPECIFICATIONS:

- The various members of the bridge should be entered into the program along with their numbers. (ex: beam1, beam2, column1, girder1, stiffener1, abutment1, foundation).
- Their structural importance can be provided with a factor between 0 and 1 which will account for the member in the final severity check.
- Once the database of the various structural and non-structural members is recorded, the location of the bridge (soil type), the span and other extraneous but important details (usage of bridge, unique structural indeterminacies) should be recorded.
- This could be done using class data types using loops for the input of the members in string format to enable creation of that array, which would be the member.

2. UPDATION OF IMAGES:

- The images corresponding to the members should be placed in a folder with file names as required by the program, which could be named matching it to the member it is located in.
- So once the image arrays are formed by the program for the members in question, the defects are predicted the artificial neural network model. For this we can employ, OpenCV or Tensorflow in Python to do image recognition using datasets for structural defects (https://www.kaggle.com/aniruddhsharma/structural-defects-network-concrete-crack-images/version/1)
- Basic image recognition could also be used to convert image into array using numpy and PIL to match and identify defects.
- Once the model is trained to do image analysis of the defects, the most probable defect will be recorded within the specific member in the database. The number of defects in a member will vary and should be accommodated in the code.
- The severity of each defect can be accommodated by assigning it a value between 0 and 1.
- A database containing the images of each defect in each member which can be accessed can also be created.

3. SEVERITY CHECK:

- A number grade out of 10 should be given, holistically to the bridge considering the defects, its number, the severity factor of the defects inputted and the importance factor of the member.
- The factors can be multiplied along with the number of defects to provide a rating based on threshold values above which the bridge could be categorised as safe, unsafe and to be decided using structural check or further analysis.

- The experts would be able to review the entire database with the rating and the bridge's various details to get a better understanding of the bridge as well as the determination of further requirements.

VISUAL RATING SYSTEM CODE DETAILS

CLASSES:

1. MEMBERS:

It is a class which represents the element in the bridge which stores data of name, importance factor, images of defects in a list in a list for each element, and the defects for each element in the same manner and the defect rating for the element.

2. FUNCTION (load_images_from_folder):

Taking the input of a main folder path which contains subdirectories (ex. Beam1, Beam2) which in turn contains images of defects of the respective element.

3. FUNCTION (defect_identification):

It takes the class object of members which has the details of each member stored as an instance of the class which is taken as input here and the images of the element are analysed by the CNN image classification AI Model after converting to a numpy array after which the decimal values of predictions are converted using np.argmax to '0' meaning first class of defects (i.e Corrosion) and '1' meaning 2^{nd} class of defects (i.e Cracks) and it is stored in indexes array. Then for each image there is a print of the defect that is detected by the model. The defect class of all images are all appended to another single list in iterations and returned into a list.

4. FUNCTION(defect_input):

Takes input of the class object of the elements and inputs the defects into each element by checking the defects with the returned list from the function defect_identification. A suitable defect rating is inputted to the element after checking the number of defects present in the respective element. (The importance factor and the defect rating could be computed with each other in summation with all the elements to give a FINAL RATING for the bridge.)

MAIN CODE:

- The class object of members is created for each element containing its details.
- The Image Classification model is loaded, which has to be present in the same directory as the Visual Rating System script ("img_class.model")
- The classes that the defects of each element belong to is returned to the defect list from function "defect_identification".
- The function "defect_input" is called to modify the class object to append the defects present in the element.
- The details of the elements are printed after the defects have been identified and stored.

IMAGE CLASSIFICATION CNN DETAILS

- We define keras sequential model from Python tensorflow to carry out the image classification.
- The directory of the datasets of classes of defects are inputted into training objects with 20% validation which are Corrosion and Spalling.
- The training of the 2 classes is done using the layers as seen in the code.
- Data augmentation training where the images repositioned to train the model more and to reduce overfitting is done.
- Training is done for 140 epochs with "adam" optimizer and Loss function Sparse Categorical Crossentropy.
- The classification model can be improved.

HOW TO USE

- Use Python IDLE to view the python scripts.
- DO NOT run the AI classification model script as it will train the model again and give different results. It has already been trained and saved as "img_class.model" which is used by the Visual Rating system Python script
- There will be a test folder in the downloads which will contain example images of elements of our bridge "beam1" and "beam2". For our experiment we will use these images assuming that they are of the bridge elements.
- The program will ask to define the number of elements for which we will enter 2.
- Then the path to the images of the first element "beam1" must be entered (for example: D:/test/beam1/
- Then the details of that element: name, importance factor etc…
- Then the path to the images of 2nd element (for example: D:/test/beam2/)
- Then entering of details of the 2nd element must be done.
- The program will then analyse these images, identify what defects they have and store them in the respective elements' details as defects and give it a defect rating based on the number of defects present in it.
- Then the details of each element will be printed along with the defects and the defect rating

FUTURE CHANGES:

- Should adopt a computation for the importance factor of the beam and the defect rating and sum it for all the elements to give a final rating for the bridge (inclusive of all its elements) itself.
- The image classification model must be trained with more datasets than only "corrosion" and "cracks" and must be made more accurate by using more convolution, dense layers to analyse the image.
- Could use tkinter package from Python to make a Graphical User Interface to make it easier for people to use the program.

REFERENCES

- https://www.mdpi.com/2076-3417/10/3/972/htm
- https://nnfs.io/
- googleimagesdownload is dead. Long live bingimagesdownload… | by Yuichi Fujiki | Medium
- Image classification | TensorFlow Core
- tf.keras.preprocessing.image_dataset_from_directory
- Python - Getting file directory as user input - Stack Overflow