CE6051: Machine Learning in Civil Engineering

Term Project

**IMAGE DETECTION & CLASSIFICATION**

**OF BRICKS AND REBARS FROM CONSTRUCTION DEBRIS**

**Submitted by**

Vani Kundarapu (CE19B061)

Bhumireddy Chandana (CE19b045)

Sowbarnika S (CE19B024)

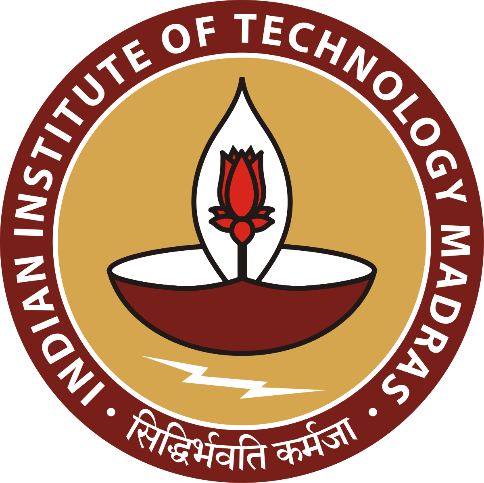
Charulatha K (CE19B032)

Rahul Mani Krishna Metta (CE19B016)

Keerthana (ED19B014)

**Submitted to**

Prof. Benny Raphael

****

Department of Civil Engineering Indian Institute of Technology Madras

**CONTENTS**

1. Objective

1. Data Classification
2. Data Augmentation
3. Python Libraries
   1. Keras
   2. Tensorflow
   3. Tensorboard
   4. NumPy
   5. Pandas
   6. Matplotlib
   7. Seaborn
   8. Pathlib
4. Convolution Neural Networks
   1. Convolution Layer
   2. Activation Layer
   3. Pooling Layer
   4. Flattening Layer
   5. Fully Connected Layers
5. Decision making Tools
   1. Sigmoid Layer
   2. Loss Function
   3. Optimizer
   4. Epoch
6. Results

**1. Objective**

The overall objective of the study is to identify recyclable or reusable objects from the construction debris. For this particular study rebars and bricks are to be detected by the neural network model. The algorithm works to automatically identify whether the given image has the presence of Brick or Rebar and classify it. The model will undergo different stages of processing and would be able to predict the required output based on the inputs and training stages.

    The steps involved in achieving the objectives are as follows:

* To build a CNN model which could classify images into Bricks and Rebars.
* To study the parameters related to the model and analyse them to improve the model.

**2.Data Classification**

The available 336 images are classified such that 80% images (267) are used for training, 10% images (33) for validating and 10% images (35) for testing.

**3. Data Augmentation**

To increase the accuracy of the data, large number of images are required. Data Augmentation is done to expand our data set by duplicating the existing images. Therefore, the available 336 images are Augmented and total number of images now becomes 1320. 125 Rebar images are increased to 492 images and 211 brick images are increased to 828 images. To obtain this increased number of datasets from the input data, we adopted some changes to our existing dataset. Rotating images to 45°, width shift of range 0.2, height shift of range 0.2, zoom range of 0.2, shear range of 0.2 and horizontal flip are the changes that are done to the given datasets. These Data Augmentation is done using Image Data Generator library, which is a part of Keras library.

Some techniques used for Data Augmentation are:

* **Flip:** Images can be flipped horizontally and vertically. Horizontal flip is used which rotates the image by 180°. horizontal\_flip=True is used.

*Original image Flipped image*

* **Rotation:** rotation\_range=45, rotates the image by 45°. Key thing to note about this operation is that image dimensions may not be preserved after rotation.



*Original image 45° rotated image*

* **Zoom range:** zoom\_range=0.2 means zoom-in and zoom-out the image by 20%.

*Original image Zoomed image*

* **Shear range:** Shear means that the image will be distorted along an axis, mostly to create or rectify the perception angles. It's usually used to augment images so that computers can see how humans see things from different angles. shear\_range=0.2 shear the image by 20%

*Original image Sheared image*

**4. Python Libraries**

**4.1. Keras:** It is a user-friendly high level deep learning API used for building Neural Networks. Keras is written in Python and is used to construct simple neural network.

**4.2 Tensorflow:** Google created this library to make it simple to implement machine learning and deep learning algorithms. For the purpose of simplifying the calculation of numerous mathematical experiments, computational algebra and optimization approaches are combined.

**4.3. Tensorboard:** Tensorboard is a visualisation tool that is part of Tensorflow. Analysing Data Flow Graphs is done using it. The learning process is managed by a module referred to as Hyperparameters.

**4.4. NumPy:** NumPy stands for Numerical Python. It is an open-source software and fundamental package used for scientific computation with Python. NumPy is used majorly to work with arrays and in the domains of linear algebra, Fourier transform, and Matrices.

**4.5. Pandas:** Pandas is a Python library that is used as a tool for data analysis. PanelData is the acronym for Pandas. Data may be imported from a variety of file types using Pandas, which also supports a number of data manipulation operations like merging, reshaping, selecting, and data cleaning.

**4.6. Matplotlib:** Matplotlib is a Python package that can be used to generate 2D graphs and plots using Python scripts. It provides a module called pyplot that simplifies graphs by offering features for controlling line styles, font characteristics, formatting axes, and so on. It supports a wide range of graphs and plots, including histograms, bar charts, power spectra, error charts, and many more. It is used in conjunction with NumPy to give an environment that is a viable open-source alternative to MatLab. It may also be used with graphics toolkits such as PyQt and wxPython.

**4.7. Seaborn:** Seaborn is a library that uses Matplotlib underneath to plot graphs,used for visualizing random distributions

**4.8. Pathlib:** Working with files while treating file paths as objects is simple with the help of the Pathlib library. By specifying the path, we first build Raw directory, which is then called by utilising this pathlib function.

**5.Convolution Neural Network (CNN) Model**

Combining and integrating two functions results in convolution, which demonstrates how one function changes the shape of the other. Four convolutional layers, four pooling layers, two fully connected layers, and one sigmoid layer make up the CNN model employed in this study. ReLU is the activation function employed, and dropout is also utilised.All of the components mentioned are explained in detail as follows:

**5.1. Convolution Layer:**

CNN distinguishes itself from other neural networks because of its convolution layer. It is the initial layer and is responsible for extracting features from the input images. Convolutions also offer a significant benefit in picture pre-processing, as pixels close to each other are considerably more co-related to each other for image recognition.

An image matrix and a filter or kernel are the two inputs that convolution requires in order to function. The matrix is multiplied in such a way that the kernel superimposes itself on the 3\*3, 5\*5, and 7\*7 image pixels. Once the matrix has been multiplied, the resulting resultant matrix is simply summed to obtain the convolved matrix. Convolution of the image with various filter types can accomplish a number of tasks, including edge recognition, sharpening, and blurring, among others.

**5.2. Activation Layer (ReLU):**

For a non-linear operation, ReLU stands for Rectified Linear Unit. There are numerous activation functions available, such as the sigmoid function (**f(x) = (1 + e-x)-1**) and the tanh function (f(x) = tanh(x)), however the most generally employed is **ReLU (f (x) = max (0,x))**. The primary benefit of ReLU is that its gradient is not saturated, which substantially speeds the convergence of stochastic gradient descent as compared to sigmoid / tanh functions. ReLU is performed to each pixel or feature in the image matrix, and all negative pixels that may result from convolution with a filter with negative inputs are transformed to 0, while all other pixels remain unchanged.

**5.3. Pooling Layer:**

The pooling layer is used when there is a lot of duplication and additional computation in the output feature matrix following the convolution layer because of the large number of input neurons. Subsampling the input image using pooling layers lowers the number of parameters, memory use, and computational load without compromising any of the image's detail. Although there are other pooling options, max-pooling has been chosen in this model

**5.4. Flattening:**

A flatten layer is used between the convolutional layer and the fully connected layer to turn the two-dimensional feature matrix into a vector that can be input to a fully connected neural network.

**5.5. Fully Connected Layer:**

The output of the flattening layer is used as an input by the fully connected layer at the end of CNN. There may be several FCNs, but there is only one in this study. To perform classification based on the features retrieved by earlier CNN layers, a fully connected layer is required. This layer is a typical artificial neural network layer that connects every neuron to every other neuron in the layer next to it.

**6. Decision Making Tools**

**6.1. Sigmoid Layer:**

The sigmoid ((1+e-x)-1) layer is used for classifying of the input data. A sigmoid layer applies a sigmoid function to the input, resulting in output that is limited to the interval (0,1). As the classification is binary, the sigmoid function has been used to finally output if the given input image is brick or rebar.

**6.2. Loss Function:**

A loss function is simply a way to assess how effectively the algorithm is modelling the dataset. Binary cross-entropy is the chosen loss function. This loss function was chosen because it may be used to implement a yes-or-no decision. To determine the final loss, binary cross-entropy simply averages the true value's distance from the prediction class.

The function is defined as: L(y, ŷ) =−1/𝑁 \* Σ (𝑦∗log (ŷ𝑖) + (1−𝑦) ∗log (1−ŷ𝑖)), where ŷ is the predicted value.

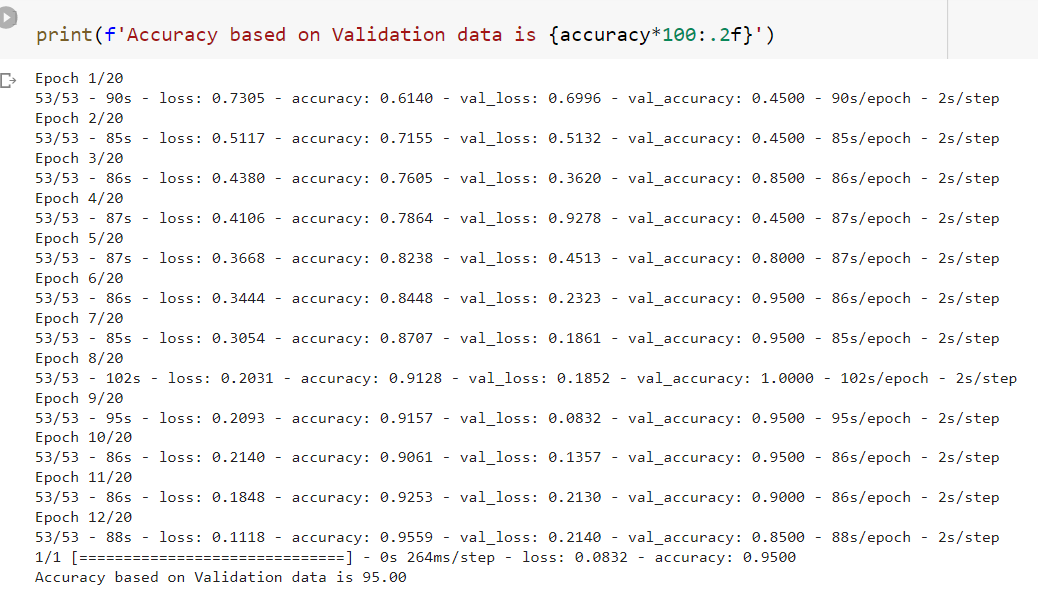
**6.3. Optimizer:**

Optimizers aid neural networks in their efforts to reduce the loss function. Adam was picked as the optimizer in the research. Adam is an abbreviation for adaptive moment estimation. Instead of a simple average, Adam uses an exponential moving average of the gradients to scale the learning rate. It maintains an exponentially declining average of previous gradients.

**6.4. Epoch:**

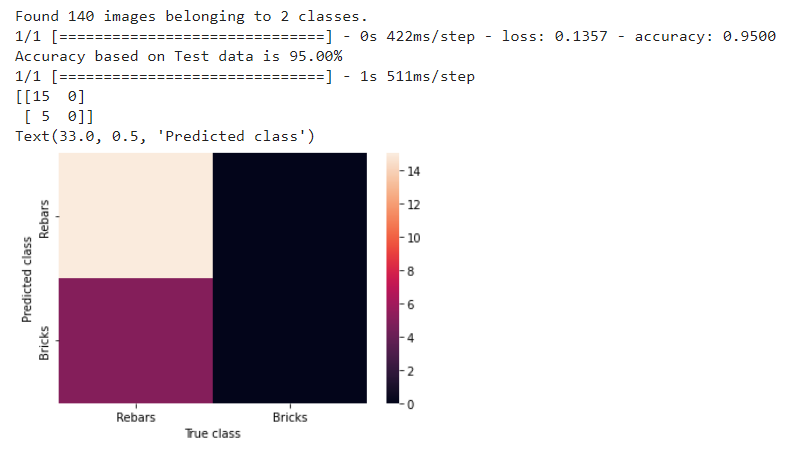
Epoch refers to a single forward and backward transit through the neural network of an entire dataset. The dataset is split up into smaller batches because providing one epoch is a very large dataset. The weight of the neural network is changed more frequently as the number of epochs rises, and the curve transitions from underfitting to ideal to overfitting.

In this project, an inbuilt Keras function called as Callbacks was employed to determine the appropriate number of epochs. Callbacks are used to monitor the loss function because once the loss function reaches a certain value specified by the user, the model should cease fitting the data, thereby saving time and preventing overfitting.



1. **Results:**

The results of the testing can be understood from the following confusion matrix



**9. Accuracy:**

Accuracy of the model based on test data is **95.00 %**

**Contribution:**

Kundarapu Vani-CE19B061: **Model Development**

Bhumireddy Chandana-CE19B045: **Data Augmentation and Test Code**

Sowbarnika S-CE19B024: **Report Preparation**

Charulatha K-CE19B032: **Report Preparation**

Rahul Mani Krishna Metta-CE19B016: **Data Classification and Data Augmentation**

Keerthana-ED19B014: **Data Classification**

**References:**

* <https://neptune.ai/blog/image-processing-python>
* <https://www.youtube.com/watch?v=ccdssX4rIh8>
* <https://www.youtube.com/watch?v=We05FyZY-KY&t=352s>