ERSA- Earthquake Resistant Structure Analysis

Google Developer Students Club- JSS Science and Technology University, Mysuru, India

Abhilash Hathwar

Bhuvan Chandra Koduru

Srivatsa Joshi

Manusreshta B L

What is ERSA?

ERSA stands for Earthquake Resistant Structure Analysis. The main aim of ERSA is to determine the structural integrity of buildings, and hence provide means to increase it and take further action.

Goals

Lately, we have been seeing news of earthquakes and their devastation. Wouldn't it be helpful for everyone if they could have an understanding of the structural integrity of their buildings and possible ways to retrofit them? This is what ERSA aims to achieve. Using data provided by the user, we can make an analysis about the stability of their building and methods to improve it. Doing so would greatly aid society, especially people in earthquake prone areas. This project is based on the seismic zone classification of geographic locations in India, hence is most accurate for Indian regions. Such a functionality has not been implemented before, especially for countries like India. This app, which is easy to use, would greatly aid the people of India.

Tech Stack used

Front end: React.js with APIs such as

Back end: Python (flask framework), along with Tensorflow and Keras.

Functionalities

Soft stories:

The term “soft-story” refers to one level of a building that is significantly more flexible or weak in lateral load resistance than the stories above it and the floors or the foundation below it. This condition can occur in any of the conventional construction types and is typically associated with large openings in the walls or an exceptionally tall story height in comparison to the adjacent stories. These soft stories can present a very serious risk in the event of an earthquake, both in human safety and financial liability.

Detecting soft stories can help in retrofitting the building and reducing risks during earthquakes. This is done in our web app:

By uploading an image of the building:

If the user wishes, they can upload a picture of their building, and a Convolutional neural network (CNN) is used to classify images of buildings as soft stories or not.

The other functionality of our web app is the earthquake stability prediction.

Given important features of the building such as geographic location, soil type, materials, importance of the building etc. we can make a general estimate about the damage that could be caused to the building. The value used to make the estimate is the “horizontal seismic acceleration coefficient”, Ah.

Once a building is classified as a soft story, a list of retrofitting measures are provided to the user, from which they can choose the method which is most applicable to their building.

If a building is not a soft story, measures to stay safe during earthquakes are provided, to increase earthquake awareness.

Working

Soft story classification

The idea used behind classifying soft story images is a Convolutional Neural Network (CNN). CNNs are used to detect features in images, such as boundaries, horizontal features, vertical features etc. CNNs are also used to reduce the size of images without compromising on the information of the image, hence making it easier to train on larger datasets. CNNs operate on the differences in the RGB (Red Green Blue) values among neighbouring pixels.

Dataset used:

Images have been collected from Google Chrome, using a Chrome extension using which we downloaded images. We also found a dataset with about 1500 images.

Augmentation:

Usually, the image data received may not be of sufficient size, and may not consider human errors such as cropping and skewing of images. Hence, we have augmented the image dataset by inducing slight rotations, skewing and horizontal mirroring. From this, we obtained a total dataset size of around 6000 images.

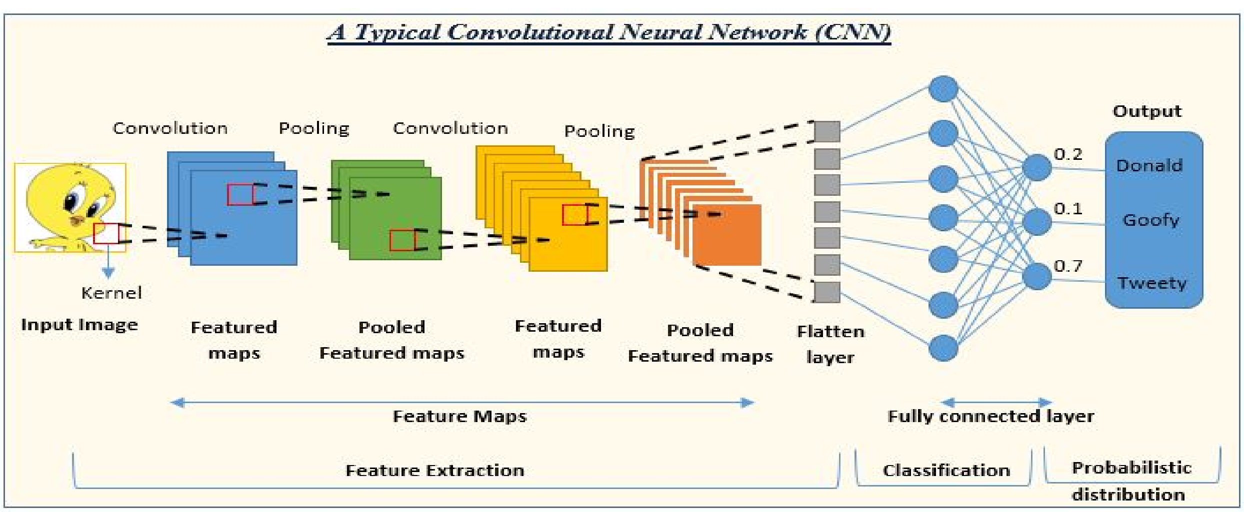
Data preprocessing:

The RGB values, which lie in the range of 0-255 are divided by 255 to reduce the values to a range of 0-1, which greatly aids the training process of the neural network.

Also, each image is resized to 256x256 pixels to maintain uniformity across all images so they can be used as input to the neural network, which expects a certain "shape" as its input.

Splitting the data:

The dataset has been split into training (70%), validation (20%) and testing (10%) sets in order to prevent overfitting.



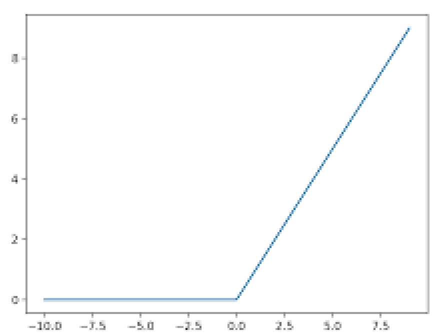
Pooling is the process of reducing the size of images without compromising on the quality of the image. The input image is then “flattened” to make a vector of one column and this vector is passed as the input to the neural network.

Architecture used:

```
model2 = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation="relu", input_shape=(256, 256, 3), kernel_initializer='he_uniform'),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Conv2D(32, (3, 3), activation="relu",kernel_initializer='he_uniform'),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Conv2D(16, (3, 3), activation="relu",kernel_initializer='he_uniform'),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation="relu",kernel_initializer='he_uniform'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(1, activation="sigmoid",kernel_initializer='glorot_uniform')
])
```

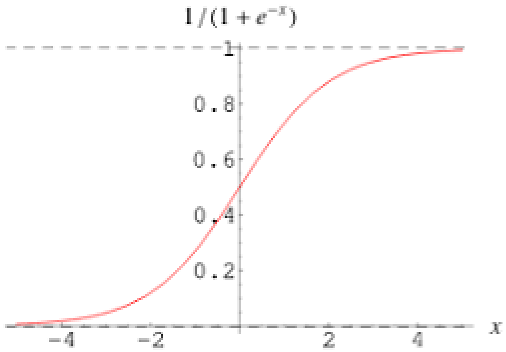
Activation functions used:

1. Apart from the output layer, all layers have ReLu as the activation function.



   Which is as shown above.

2. The output layer has a Sigmoid activation function which is used to predict the probabilities of each class, i.e, soft story or not. It is used to predict a value lying between 0 and 1.



Filters:

Across the various convolutional layers, 16 or 32 filters are learnt for the images which help identify the features of soft story or sturdy buildings.

Pooling:

To reduce the size of each image and aid computation, max pooling of 2x2 is applied, which effectively halves the image without losing the information of the image.

Initialisers:

Different activation functions have different initialisers for each neuron, which improve the accuracy of predictions.

He uniform initialiser:

It is used mainly for the ReLu activation function and selects a value from a given range of values in a uniform distribution.

- Xavier Normal initialization

$$W \sim N(0, Var(W))$$
$$Var(W) = \sqrt{\frac{2}{n_{in} + n_{out}}}$$

- He Normal initialization

$$W \sim N(0, Var(W))$$
$$Var(W) = \sqrt{\frac{2}{n_{in}}}$$

- Xavier Uniform initialization

$$W \sim U(-\sqrt{\frac{6}{n_{in} + n_{out}}}, +\sqrt{\frac{6}{n_{in} + n_{out}}})$$

- He Uniform initialization

$$W \sim U(-\sqrt{\frac{6}{n_{in}}}, +\sqrt{\frac{6}{n_{in}}})$$

Glorot uniform initialiser: (Also called the Uniform Xavier initialiser)

It is used to initialise neurons with the Sigmoid activation function.

Xavier initialization

Uniform Xavier initialization: draw each weight, w, from a random uniform distribution in [-x,x] for $x = \sqrt{\frac{6}{inputs + outputs}}$

Normal Xavier initialization: draw each weight, w, from a normal distribution with a mean of 0, and a standard deviation $\sigma = \sqrt{\frac{2}{inputs + outputs}}$

Results:

As shown below, the training set accuracy is close to 99.9%, with the validation set accuracy being close to 75%. The test set accuracy was around 67%.

```
Epoch 1/20
127/127 [==============================] - 52s 400ms/step - loss: 2.3142 - accuracy: 0.5544 - val_loss: 0.6790 - val_accuracy: 0.6102
Epoch 2/20
127/127 [==============================] - 55s 427ms/step - loss: 0.5728 - accuracy: 0.7170 - val_loss: 0.6124 - val_accuracy: 0.6649
Epoch 3/20
127/127 [==============================] - 58s 456ms/step - loss: 0.4134 - accuracy: 0.8250 - val_loss: 0.5846 - val_accuracy: 0.6866
Epoch 4/20
127/127 [==============================] - 57s 445ms/step - loss: 0.2412 - accuracy: 0.9090 - val_loss: 0.5938 - val_accuracy: 0.7109
Epoch 5/20
127/127 [==============================] - 59s 463ms/step - loss: 0.1369 - accuracy: 0.9545 - val_loss: 0.6323 - val_accuracy: 0.7474
Epoch 6/20
127/127 [==============================] - 60s 472ms/step - loss: 0.0687 - accuracy: 0.9801 - val_loss: 0.8356 - val_accuracy: 0.7274
Epoch 7/20
127/127 [==============================] - 65s 507ms/step - loss: 0.0361 - accuracy: 0.9904 - val_loss: 0.7602 - val_accuracy: 0.7361
Epoch 8/20
127/127 [==============================] - 67s 520ms/step - loss: 0.0320 - accuracy: 0.9931 - val_loss: 0.8785 - val_accuracy: 0.7335
Epoch 9/20
127/127 [==============================] - 62s 482ms/step - loss: 0.0312 - accuracy: 0.9926 - val_loss: 0.8435 - val_accuracy: 0.7465
Epoch 10/20
127/127 [==============================] - 64s 499ms/step - loss: 0.0247 - accuracy: 0.9943 - val_loss: 1.1209 - val_accuracy: 0.7387
Epoch 11/20
127/127 [==============================] - 68s 533ms/step - loss: 0.0233 - accuracy: 0.9953 - val_loss: 0.9769 - val_accuracy: 0.7483
Epoch 12/20
127/127 [==============================] - 67s 524ms/step - loss: 0.0267 - accuracy: 0.9946 - val_loss: 0.9747 - val_accuracy: 0.7283
Epoch 13/20
...
Epoch 19/20
127/127 [==============================] - 73s 573ms/step - loss: 0.0096 - accuracy: 0.9993 - val_loss: 1.2312 - val_accuracy: 0.7431
Epoch 20/20
127/127 [==============================] - 69s 539ms/step - loss: 0.0104 - accuracy: 0.9968 - val_loss: 1.1174 - val_accuracy: 0.7405
```

Ah calculation:

The value of the horizontal seismic coefficient

$$V_B = A_h W$$

$$A_h = \frac{Z}{2} \cdot \frac{S_a}{g} \cdot \frac{I}{R}$$

Z=Zone Factor

S_s/g = Spectral Acceleration taken from Response Spectrum

I= Importance Factor

R=Ductility / Over-Strength Reduction Factor

Table 2 Zone Factor, *Z*
(*Clause* 6.4.2)

Seismic Zone	II	III	IV	V
Seismic Intensity	Low	Moderate	Severe	Very Severe
Z	0.10	0.16	0.24	0.36

And according to the soil types (rocky, medium or hard), the value of (Sa/g) can be calculated.

The importance factor of a building depends on its use. Important service and community buildings such as schools and hospitals have an importance factor of 1.5, while the others have a factor of 1.

The response reduction factor has been assumed to be 3, since most concrete buildings have a response reduction factor of 3.

The effect of earthquakes on a building depends on the value of the seismic coefficient obtained:

1. Ah <0.1g: The influence of earthquakes on such buildings is less.
2. Ah <0.4g: Moderate influence of earthquakes on the building
3. Ah>0.4g: High influence of earthquakes on the building can be observed.

Once the impact is calculated, retrofitting methods can be suggested to the user, as follows. The user can pick any method which applies to them.