**Image recognition Project**

**Steps of AI model building:**

1. Load both folders (healthy & defect) into server.
2. Append all images into 2 arrays (input data & labels)
3. Exploratory data analysis for small image representation of a larger images
4. **Data pre-processing**

* Create dataset matrix with the standard shape
* Convert images into arrays matrix and dump those arrays matrix into dataset matrix
* Normalize the images to bring image intensity pixel intensity into a range that is more familiar to sense
* Split the dataset into training & validation set
* Encode categorical variables into binary variables

1. **Model building**

* I used sequential model of keras (python library)
* I used this ConvNet architecture

INPUT -> [CONV -> RELU] -> POOL -> [CONV -> RELU] \*2 -> POOL -> [CONV -> RELU]\*3 -> POOL -> [FC -> RELU] -> SoftMax (binary output)

These are following layers used in ConvNet model:

* **Separableconv2d:** It performs a depthwise spatial convolution followed by a pointwise convolution which mixes together the resulting output channels.
* **Averagepooling2d**: It extracts features so smoothly & gain more information about image than max pooling.
* **Relu layer**: It is used to increase non-linearity in the network. It removes negative values from an activation map by setting them to zero.
* **Dropout**: It ignore randomly selected neurons during training.
* **Fully connected layer**: It’s a last layer of ConvoNet, where the input from the other layers is flattened and sent so as the transform the output into the number of classes as desired by the network.
* **Softmax**: Softmax is implemented through a fully connected layer just before the output layer. It is used to compute the probabilities for the classes.

1. **GridSearchCV**

After building model, parameter optimization is very necessary because the accuracy of model depends on parameters like optimizer, learning rate, batch size & epochs. So for this I used GridSearchCV to select best optimizer & learning rate value.

1. **Model Optimization**

There are many optimizer such as 'SGD', 'RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax' & 'Nadam'. So I used ‘Adam’ optimizer to compile the model because it gave highest accuracy than other optimizers.

1. **Flask app**

I used flask web framework to create web application. It has 2 pages.

1st page is image uploading page and 2nd page shows the result (healthy, cracked).

I deployed this application to my local machine. User can upload one image at a time and click to submit button and can see the result (healthy & cracked).

Name – Rajkumar

Designation-Associate Data Analyst

Company-LogisticsNow

University-IIT Bombay