### **House Price Estimation using Both Visual and Textual Data**

### 

### 

]

**Problem Statement**

This project aims to estimate house price using both visual and textual data.

**Methodology**

**Dataset**

We had a total of 2140 images and 535 rows.We have removed the outliers by keeping the price value between 50k to 900k and the new dataset has 453 rows. Then we have divided the data set into 80% : 20% ratio.

**Regression**

In the project we have implemented the following neural network model, CNN and a combined model using Keras functional API for House Price Estimation using following activation function and optimizers:-

**Activation function** : relu

**Optimizer** : adam

**Kernel size for the CNN** : 4\*4, 3\*3, 2\*2

**Experimental Results and Analysis**

In this project we tried different activation functions such has relu, sigmoid & tanh in all the models.

**Combined Model**

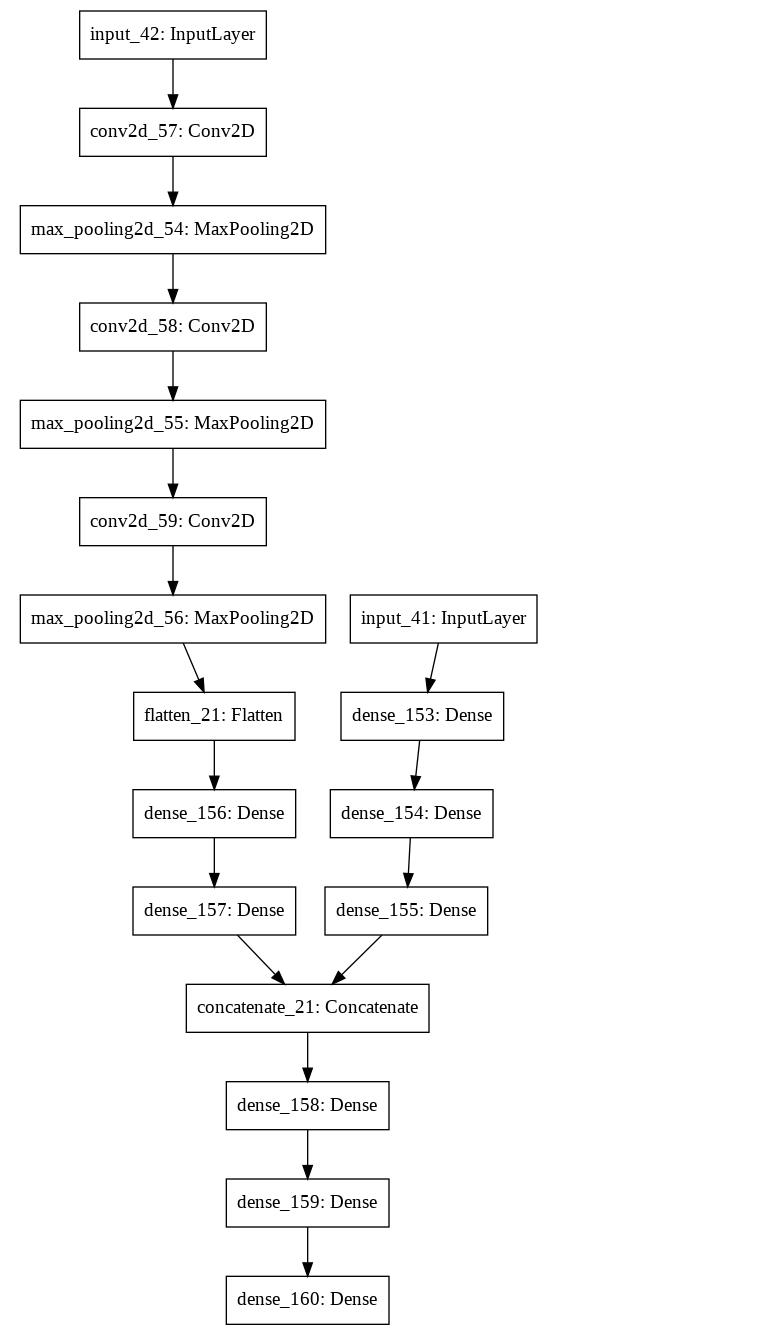
We have 3 hidden layers with 8 neurons, 4 neurons and 1 neuron in this model and have combined the output of the two models namely:

1. **Fully Connected Neural Network**

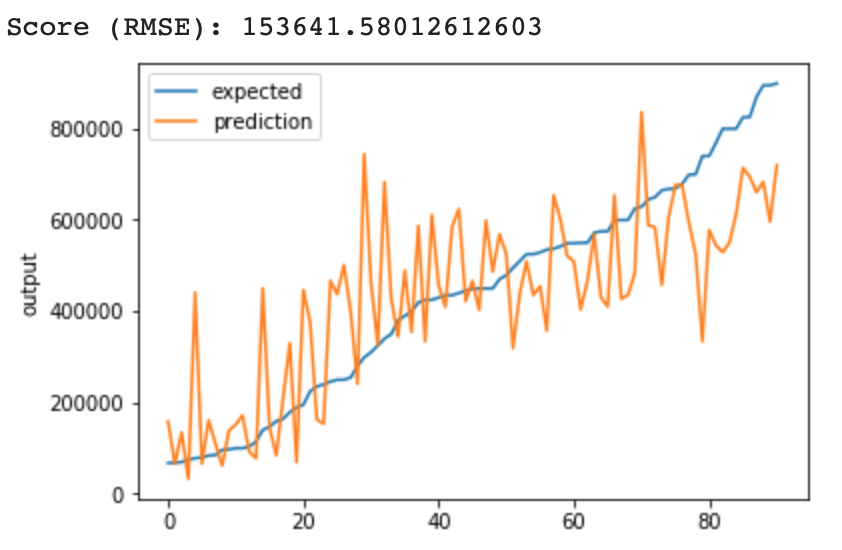
* We first used 3 hidden layers.
* First hidden layer we use 64 neurons, second hidden layer we used 32 neurons & in the third hidden layer we used 5 neurons.

1. **Convolutional Neural Network**

* We have used 6 hidden layers with 3 convolutions and 3 max pooling layers.
* In this model, we have images with 128 row, 128 columns and 3 channels .
* We used 3 convolutional layers with 64 ,32 ,32 neurons respectively, 3 max pooling layers with pool size of 2\*2 for CNN model.
* We have used kernel size of 4\*4, 3\*3 ,2\*2.



**Results:**



**Task Division and Project Reflection**

**Task Division**

**Preprocessing**

- Loading data [Combined]

- Numerical data pre-processing [Combined]

- Categorical data one hot encoding [Combined]

**Train & Test**

Model implementation Combined model/ NN /CNN ​​[Combinedi]

Visualization / parameters test ​​[Combined]

**Project Reflection**

**Challenges**

The various challenges that we realized with this dataset while training and testing neural networks are:

* Removing outliers.
* Combining data frame of text and image and then removing outliers.
* We were facing issues of NaN values when we were removing the outliers

**Solution**

* **Model Implementation details & observations**  
  1) **​Keras Functional API**​​:
* ● **Multiple input and one output:** Model consists of three sub models NN and CNN, the combined output of both is fed to another NN model. And the prediction is made.

2) ​**Compilation ( model.compile() )**​​:

* ● **Optimizer**​​ -​ ​It is the most popular ​**Optimization**​​ algorithms used in optimizing a  
  **Neural Network**​​. For instance, update and tune the Model's parameters in a direction so that we can minimize the Loss function. Optimizers we used:  
  ■ Adam
* **● Loss -** ​​A loss function. This is the objective that the model will try to minimize.
  + ○ For regression model - ‘mean\_squared\_error’

3) ​**Training ( model.fit() ):**

* ● **Validation data**​​-​tuple ​(x\_val, y\_val)​ on which to evaluate the loss and any model metrics at the end of each epoch.
* ● **Verbose -** value used 2.
* **● Epochs -** Number of epochs to train model,value used 800
* **● Batch\_size -** Samples per gradient update,value used 32(default)