Project 2-2

Subject:

House Price Prediction with Image and Descriptive Data

(Using Input Fusion Method)

Authors:

Zahra MohammadBeigi

Student no. : 97222079

Ghazal Rafiei

Student no. : 97222044

Getting Inference From Data:

The house price dataset which we are using includes not only numerical and categorical data, but image data as well .
Numeric/continuous values, such as n_city, bed, bath, sqft, ...
Categorical values, including street and city.
Image data, including house images.

	image_id	street	city	n_city	bed	bath	sqft	price
0	0	1317 Van Buren Avenue	Salton City, CA	317	3	2.0	1560	201900
1	1	124 C Street W	Brawley, CA	48	3	2.0	713	228500
2	2	2304 Clark Road	Imperial, CA	152	3	1.0	800	273950
3	3	755 Brawley Avenue	Brawley, CA	48	3	1.0	1082	350000
4	4	2207 R Carrillo Court	Calexico, CA	55	4	3.0	2547	385100

A sample:

Preprocessing Numeric and Categorical data:

Let's have a look at first:

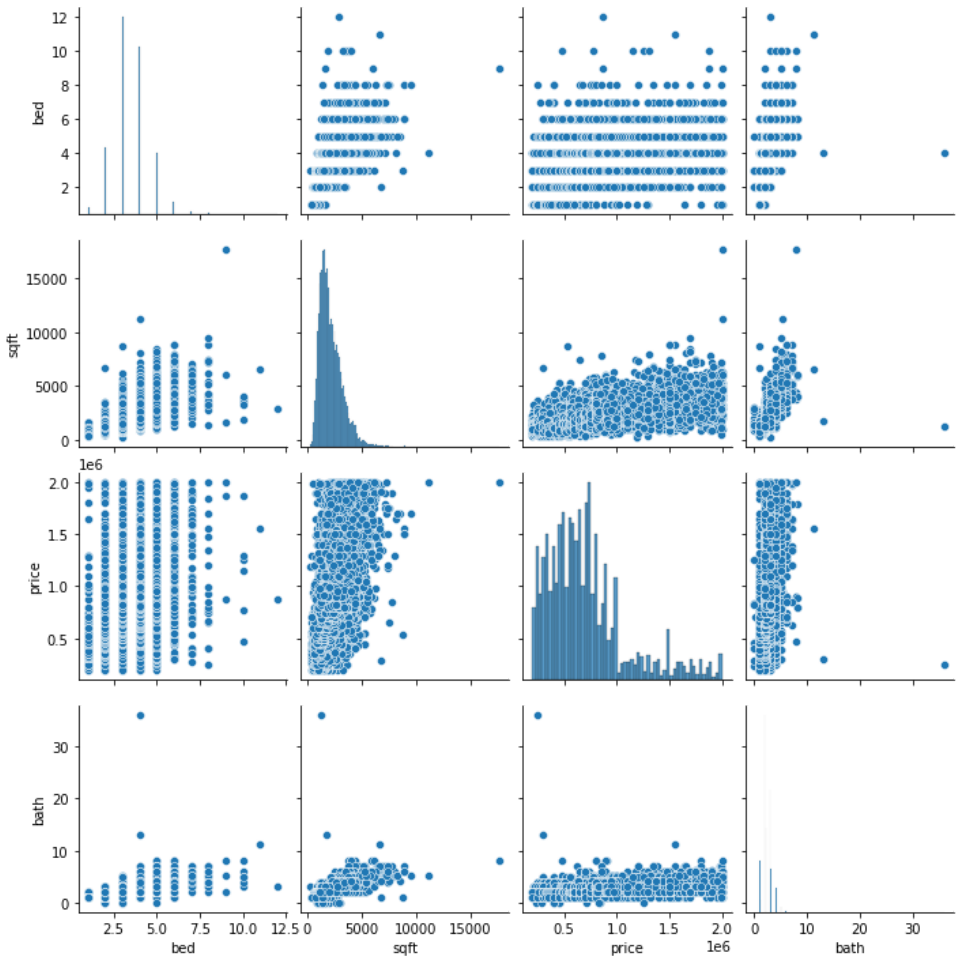
	count	mean	std	min	25%	50%	75%	max
image_id	15474.0	7736.500000	4467.103368	0.0	3868.25	7736.5	11604.75	15473.0
n_city	15474.0	216.597518	112.372985	0.0	119.00	222.5	315.00	414.0
bed	15474.0	3.506398	1.034838	1.0	3.00	3.0	4.00	12.0
bath	15474.0	2.453251	0.958742	0.0	2.00	2.1	3.00	36.0
sqft	15474.0	2173.913209	1025.339617	280.0	1426.00	1951.0	2737.75	17667.0
price	15474.0	703120.937508	376976.154421	195000.0	445000.00	639000.0	834975.00	2000000.0

```
1 df.isnull().sum()
```

```
image_id    0
street      0
city        0
n_city      0
bed         0
bath        0
sqft        0
price       0
dtype: int64
```

As you can see all of the columns are non-null we don't need to retrieve null values.

To see which parameters are more correlated with each other, we draw a pairplot:

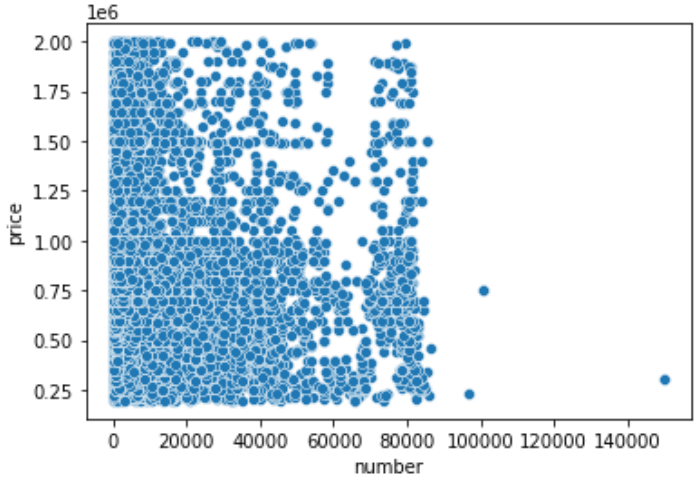


From our 15474 data we have 12401 unique streets and 415 unique cities.
The left image below is the ten first attributes of the street column.and the right one is the attributes which their street name is 'Dixie Street' and 'Lakeview Drive'(these street names have been chosen arbitrarily).
According to these we can conclude there are different street numbers with the same street name.

```
0    1317 Van Buren Avenue
1            124 C Street W
2          2304 Clark Road
3       755 Brawley Avenue
4    2207 R Carrillo Court
5       755 Brawley Avenue
6      1100 CAMILIA Street
7      803 Chaparral Court
8      803 Chaparral Court
9          2306 Lark Court
Name: street, dtype: object
```

```
2822 Dixie Street
2822 Dixie Street
7108 Lakeview Drive
6788 Lakeview Drive
39900 Lakeview Drive
18195 Lakeview Drive
40218 Lakeview Drive
40218 Lakeview Drive
```

We split numbers and street names and store them in two different columns and we'll check out whether there is correlation between street numbers and prices.



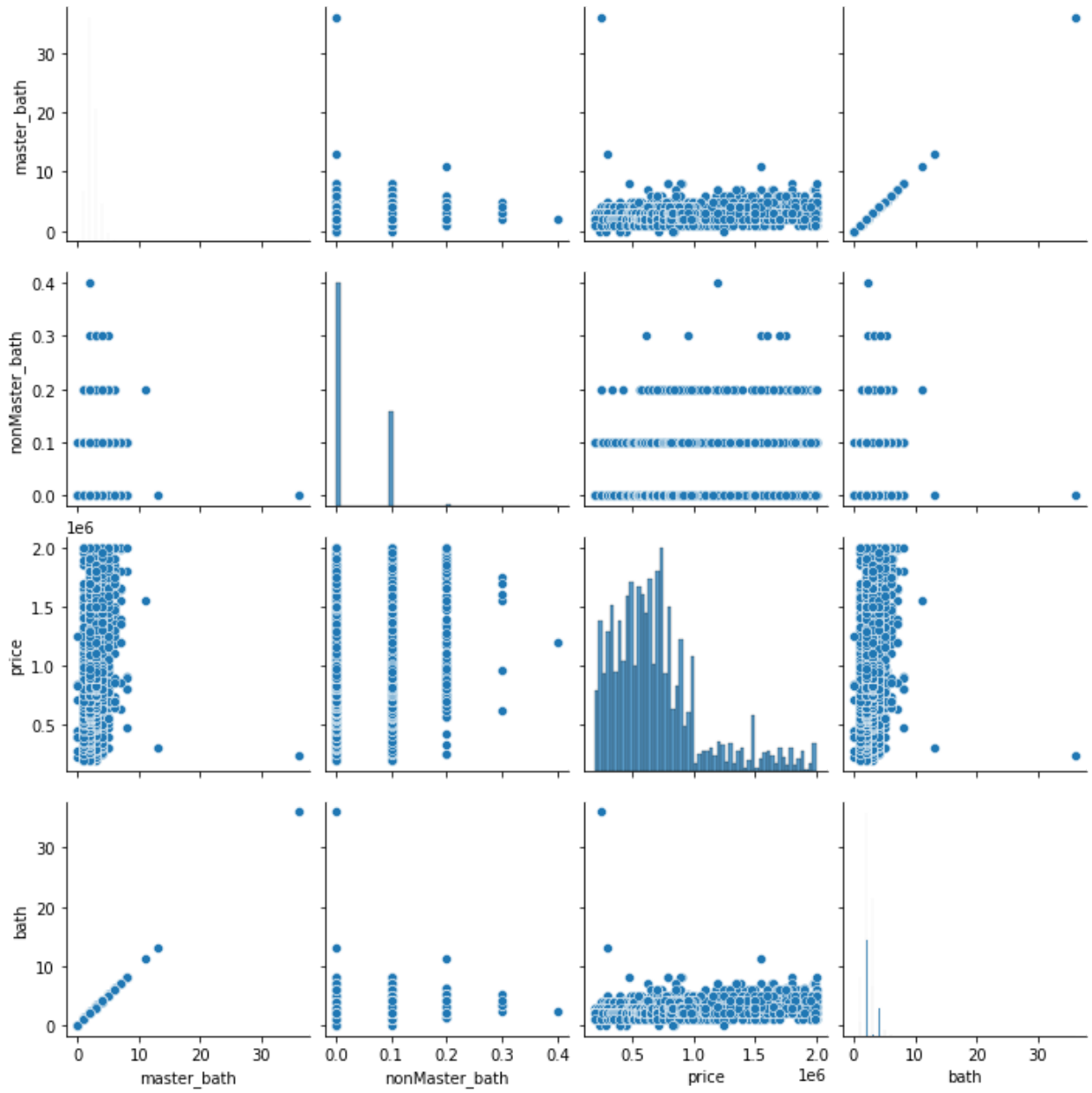
Because there are so many different cities and street names and dimensions of data frames would be so large, we don't consider them for now. But adding numbers to the data frame adds only one column so we can consider that.Also there can be a correlation due to the plot.

There might be a pattern. We will see this later in MLP models.

We split each number in the column bath to its integer and decimal parts and stored them in 'master_bath' and 'nonMaster_bath' columns respectively.

	master_bath	nonMaster_bath
0	2.0	0.0
1	2.0	0.0
2	1.0	0.0
3	1.0	0.0
4	3.0	0.0

Also we drew the pair plot below and we can conclude due to it that it won't make much difference but we'll see later.



MODELS:

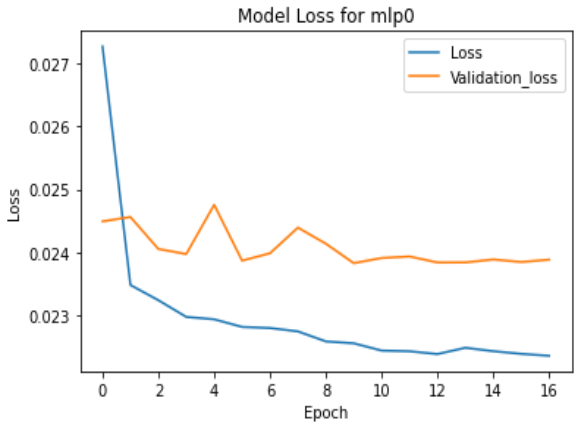
For reducing the time spent in reading data and also in training our model we chose only 3000 of our data randomly.

Maybe if we could use all 15000 images, we would gain more accurate results.

MLP(Multilayer Perceptron) for Descriptive Data:

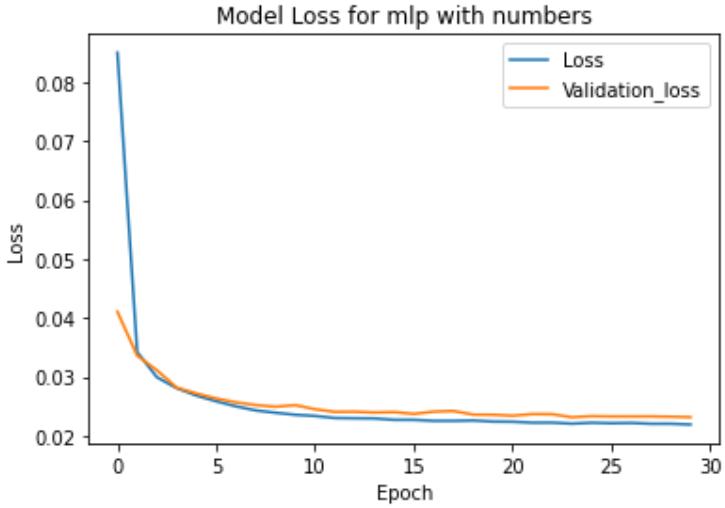
1. With The Default Columns

```
Epoch 1/30
225/225 - 0s - loss: 0.0273 - val_loss: 0.0245
Epoch 2/30
225/225 - 0s - loss: 0.0235 - val_loss: 0.0246
Epoch 3/30
225/225 - 0s - loss: 0.0232 - val_loss: 0.0241
Epoch 4/30
225/225 - 0s - loss: 0.0230 - val_loss: 0.0240
Epoch 5/30
225/225 - 0s - loss: 0.0229 - val_loss: 0.0248
Epoch 6/30
225/225 - 0s - loss: 0.0228 - val_loss: 0.0239
Epoch 7/30
225/225 - 0s - loss: 0.0228 - val_loss: 0.0240
Epoch 8/30
225/225 - 0s - loss: 0.0227 - val_loss: 0.0244
Epoch 9/30
225/225 - 0s - loss: 0.0226 - val_loss: 0.0241
Epoch 10/30
225/225 - 0s - loss: 0.0226 - val_loss: 0.0238
Epoch 11/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0239
Epoch 12/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0239
Epoch 13/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0238
Epoch 14/30
225/225 - 0s - loss: 0.0225 - val_loss: 0.0238
Epoch 15/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0239
Epoch 16/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0238
Epoch 17/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0239
Epoch 00017: early stopping
```



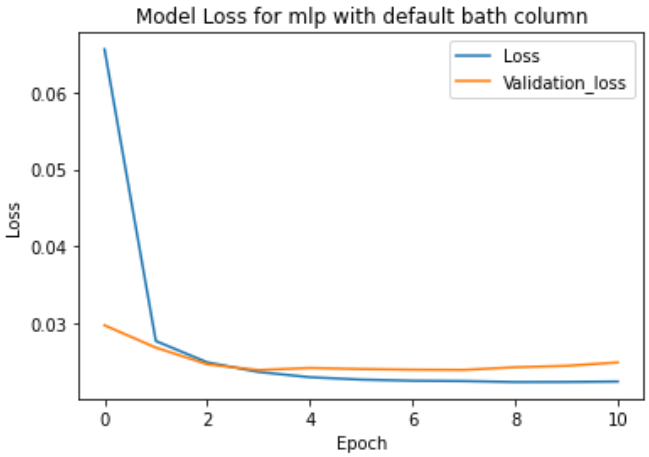
1. **Street numbers column included:**

```
Epoch 1/30
225/225 - 0s - loss: 0.0850 - val_loss: 0.0411
Epoch 2/30
225/225 - 0s - loss: 0.0342 - val_loss: 0.0336
Epoch 3/30
225/225 - 0s - loss: 0.0300 - val_loss: 0.0311
Epoch 4/30
225/225 - 0s - loss: 0.0281 - val_loss: 0.0282
Epoch 5/30
225/225 - 0s - loss: 0.0268 - val_loss: 0.0272
Epoch 6/30
225/225 - 0s - loss: 0.0259 - val_loss: 0.0264
Epoch 7/30
225/225 - 0s - loss: 0.0250 - val_loss: 0.0257
Epoch 8/30
225/225 - 0s - loss: 0.0243 - val_loss: 0.0252
Epoch 9/30
225/225 - 0s - loss: 0.0239 - val_loss: 0.0250
Epoch 10/30
225/225 - 0s - loss: 0.0236 - val_loss: 0.0252
Epoch 11/30
225/225 - 0s - loss: 0.0234 - val_loss: 0.0245
Epoch 12/30
225/225 - 0s - loss: 0.0230 - val_loss: 0.0241
Epoch 13/30
225/225 - 0s - loss: 0.0230 - val_loss: 0.0241
Epoch 14/30
225/225 - 0s - loss: 0.0230 - val_loss: 0.0240
Epoch 15/30
225/225 - 0s - loss: 0.0227 - val_loss: 0.0240
Epoch 16/30
225/225 - 0s - loss: 0.0227 - val_loss: 0.0237
Epoch 17/30
225/225 - 0s - loss: 0.0226 - val_loss: 0.0241
Epoch 18/30
225/225 - 0s - loss: 0.0226 - val_loss: 0.0242
Epoch 19/30
225/225 - 0s - loss: 0.0226 - val_loss: 0.0236
Epoch 20/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0236
Epoch 21/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0234
Epoch 22/30
225/225 - 0s - loss: 0.0222 - val_loss: 0.0237
Epoch 23/30
225/225 - 0s - loss: 0.0223 - val_loss: 0.0237
Epoch 24/30
225/225 - 0s - loss: 0.0221 - val_loss: 0.0232
Epoch 25/30
225/225 - 0s - loss: 0.0222 - val_loss: 0.0234
Epoch 26/30
225/225 - 0s - loss: 0.0222 - val_loss: 0.0233
Epoch 27/30
225/225 - 0s - loss: 0.0222 - val_loss: 0.0233
Epoch 28/30
225/225 - 0s - loss: 0.0221 - val_loss: 0.0233
Epoch 29/30
225/225 - 0s - loss: 0.0221 - val_loss: 0.0233
Epoch 30/30
225/225 - 0s - loss: 0.0219 - val_loss: 0.0232
```



3. 'master_bath' and 'nonMaster_bath' Instead of Column Bath:

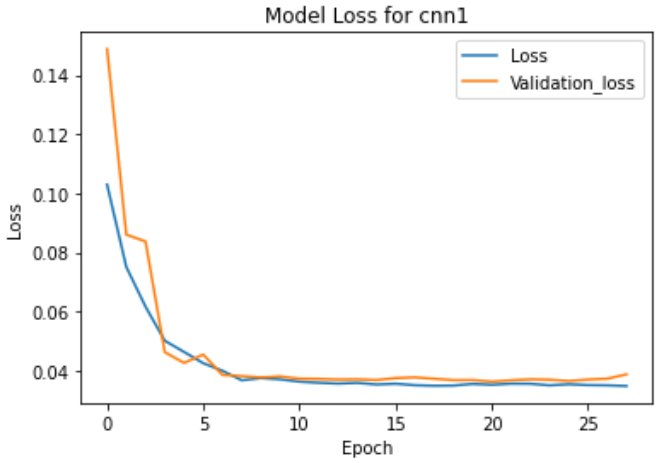
```
Epoch 1/30
225/225 - 0s - loss: 0.0656 - val_loss: 0.0297
Epoch 2/30
225/225 - 0s - loss: 0.0276 - val_loss: 0.0268
Epoch 3/30
225/225 - 0s - loss: 0.0249 - val_loss: 0.0246
Epoch 4/30
225/225 - 0s - loss: 0.0236 - val_loss: 0.0239
Epoch 5/30
225/225 - 0s - loss: 0.0229 - val_loss: 0.0241
Epoch 6/30
225/225 - 0s - loss: 0.0226 - val_loss: 0.0240
Epoch 7/30
225/225 - 0s - loss: 0.0225 - val_loss: 0.0239
Epoch 8/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0239
Epoch 9/30
225/225 - 0s - loss: 0.0223 - val_loss: 0.0242
Epoch 10/30
225/225 - 0s - loss: 0.0223 - val_loss: 0.0244
Epoch 11/30
225/225 - 0s - loss: 0.0224 - val_loss: 0.0249
Epoch 00011: early stopping
```



CNN(Convolutional Neural Network) for Image Data:

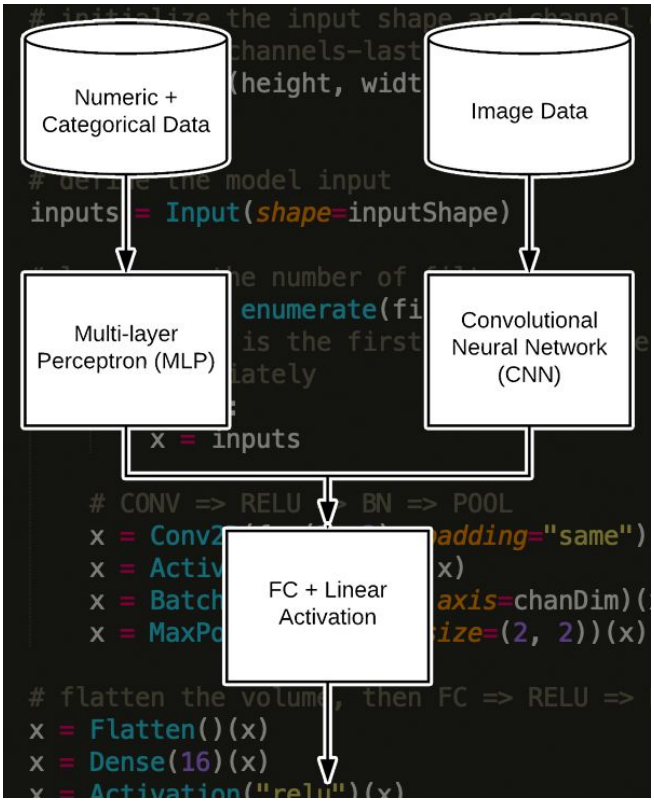
```
Epoch 1/50
WARNING:tensorflow:Callbacks method `on_train_ba
36/36 - 3s - loss: 0.1030 - val_loss: 0.1488
Epoch 2/50
36/36 - 2s - loss: 0.0751 - val_loss: 0.0861
Epoch 3/50
36/36 - 3s - loss: 0.0617 - val_loss: 0.0838
Epoch 4/50
36/36 - 3s - loss: 0.0502 - val_loss: 0.0464
Epoch 5/50
36/36 - 3s - loss: 0.0464 - val_loss: 0.0427
Epoch 6/50
36/36 - 3s - loss: 0.0426 - val_loss: 0.0455
Epoch 7/50
36/36 - 3s - loss: 0.0400 - val_loss: 0.0386
Epoch 8/50
36/36 - 2s - loss: 0.0368 - val_loss: 0.0383
Epoch 9/50
36/36 - 2s - loss: 0.0376 - val_loss: 0.0378
Epoch 10/50
36/36 - 2s - loss: 0.0372 - val_loss: 0.0381
Epoch 11/50
36/36 - 2s - loss: 0.0364 - val_loss: 0.0374
Epoch 12/50
36/36 - 2s - loss: 0.0360 - val_loss: 0.0373
Epoch 13/50
36/36 - 2s - loss: 0.0357 - val_loss: 0.0371
Epoch 14/50
36/36 - 2s - loss: 0.0360 - val_loss: 0.0372

Epoch 15/50
36/36 - 2s - loss: 0.0354 - val_loss: 0.0369
Epoch 16/50
36/36 - 2s - loss: 0.0357 - val_loss: 0.0376
Epoch 17/50
36/36 - 2s - loss: 0.0352 - val_loss: 0.0378
Epoch 18/50
36/36 - 2s - loss: 0.0350 - val_loss: 0.0374
Epoch 19/50
36/36 - 2s - loss: 0.0350 - val_loss: 0.0369
Epoch 20/50
36/36 - 2s - loss: 0.0356 - val_loss: 0.0369
Epoch 21/50
36/36 - 2s - loss: 0.0353 - val_loss: 0.0364
Epoch 22/50
36/36 - 2s - loss: 0.0357 - val_loss: 0.0369
Epoch 23/50
36/36 - 2s - loss: 0.0357 - val_loss: 0.0371
Epoch 24/50
36/36 - 2s - loss: 0.0351 - val_loss: 0.0370
Epoch 25/50
36/36 - 2s - loss: 0.0355 - val_loss: 0.0366
Epoch 26/50
36/36 - 2s - loss: 0.0352 - val_loss: 0.0371
Epoch 27/50
36/36 - 2s - loss: 0.0351 - val_loss: 0.0374
Epoch 28/50
36/36 - 2s - loss: 0.0349 - val_loss: 0.0388
Epoch 00028: early stopping
<tensorflow.python.keras.callbacks.History at 0x7f16593ad7f0>
```



FUSION MODELS:

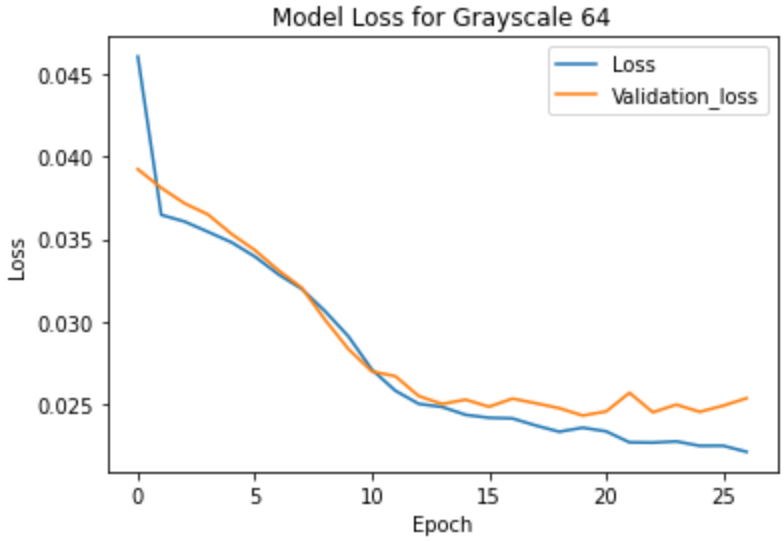
In fusion model we concatenate our mlp and cnn as it's shown below:



1. Grayscale 64*64

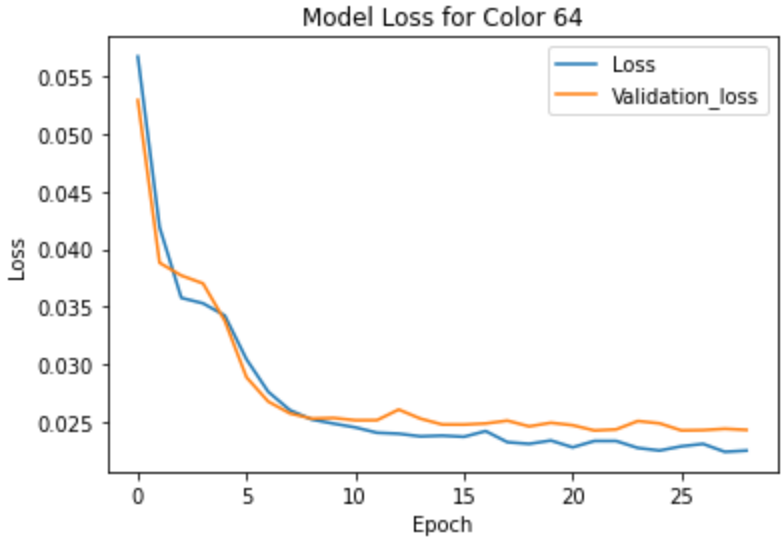
```
Epoch 1/100
WARNING:tensorflow:Callbacks method `on_train
36/36 - 3s - loss: 0.0461 - val_loss: 0.0392
Epoch 2/100
36/36 - 2s - loss: 0.0365 - val_loss: 0.0381
Epoch 3/100
36/36 - 2s - loss: 0.0361 - val_loss: 0.0372
Epoch 4/100
36/36 - 2s - loss: 0.0354 - val_loss: 0.0365
Epoch 5/100
36/36 - 3s - loss: 0.0348 - val_loss: 0.0353
Epoch 6/100
36/36 - 3s - loss: 0.0339 - val_loss: 0.0343
Epoch 7/100
36/36 - 3s - loss: 0.0329 - val_loss: 0.0331
Epoch 8/100
36/36 - 2s - loss: 0.0320 - val_loss: 0.0321
Epoch 9/100
36/36 - 2s - loss: 0.0306 - val_loss: 0.0301
Epoch 10/100
36/36 - 2s - loss: 0.0291 - val_loss: 0.0283
Epoch 11/100
36/36 - 2s - loss: 0.0271 - val_loss: 0.0270
Epoch 12/100
36/36 - 2s - loss: 0.0258 - val_loss: 0.0267
Epoch 13/100
36/36 - 2s - loss: 0.0250 - val_loss: 0.0255
Epoch 14/100
36/36 - 2s - loss: 0.0248 - val_loss: 0.0250
```

```
Epoch 16/100
36/36 - 2s - loss: 0.0237 - val_loss: 0.0247
Epoch 17/100
36/36 - 2s - loss: 0.0242 - val_loss: 0.0248
Epoch 18/100
36/36 - 2s - loss: 0.0232 - val_loss: 0.0251
Epoch 19/100
36/36 - 2s - loss: 0.0231 - val_loss: 0.0246
Epoch 20/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0249
Epoch 21/100
36/36 - 2s - loss: 0.0228 - val_loss: 0.0247
Epoch 22/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0242
Epoch 23/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0243
Epoch 24/100
36/36 - 2s - loss: 0.0227 - val_loss: 0.0250
Epoch 25/100
36/36 - 2s - loss: 0.0225 - val_loss: 0.0248
Epoch 26/100
36/36 - 2s - loss: 0.0228 - val_loss: 0.0242
Epoch 27/100
36/36 - 2s - loss: 0.0231 - val_loss: 0.0243
Epoch 28/100
36/36 - 2s - loss: 0.0224 - val_loss: 0.0244
Epoch 29/100
36/36 - 2s - loss: 0.0225 - val_loss: 0.0243
Epoch 00029: early stopping
```



2. Color 64*64

```
Epoch 1/100
WARNING:tensorflow:Callbacks method `on_train
36/36 - 3s - loss: 0.0567 - val_loss: 0.0530
Epoch 2/100
36/36 - 2s - loss: 0.0419 - val_loss: 0.0388
Epoch 3/100
36/36 - 2s - loss: 0.0358 - val_loss: 0.0377
Epoch 4/100
36/36 - 2s - loss: 0.0353 - val_loss: 0.0370
Epoch 5/100
36/36 - 2s - loss: 0.0342 - val_loss: 0.0338
Epoch 6/100
36/36 - 2s - loss: 0.0304 - val_loss: 0.0288
Epoch 7/100
36/36 - 2s - loss: 0.0276 - val_loss: 0.0267
Epoch 8/100
36/36 - 2s - loss: 0.0260 - val_loss: 0.0257
Epoch 9/100
36/36 - 2s - loss: 0.0252 - val_loss: 0.0253
Epoch 10/100
36/36 - 2s - loss: 0.0248 - val_loss: 0.0253
Epoch 11/100
36/36 - 2s - loss: 0.0245 - val_loss: 0.0251
Epoch 12/100
36/36 - 2s - loss: 0.0240 - val_loss: 0.0251
Epoch 13/100
36/36 - 2s - loss: 0.0239 - val_loss: 0.0260
Epoch 14/100
36/36 - 2s - loss: 0.0237 - val_loss: 0.0252
Epoch 15/100
36/36 - 2s - loss: 0.0244 - val_loss: 0.0253
Epoch 16/100
36/36 - 2s - loss: 0.0242 - val_loss: 0.0248
Epoch 17/100
36/36 - 2s - loss: 0.0241 - val_loss: 0.0253
Epoch 18/100
36/36 - 2s - loss: 0.0237 - val_loss: 0.0251
Epoch 19/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0248
Epoch 20/100
36/36 - 2s - loss: 0.0236 - val_loss: 0.0243
Epoch 21/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0246
Epoch 22/100
36/36 - 2s - loss: 0.0227 - val_loss: 0.0257
Epoch 23/100
36/36 - 2s - loss: 0.0227 - val_loss: 0.0245
Epoch 24/100
36/36 - 2s - loss: 0.0227 - val_loss: 0.0250
Epoch 25/100
36/36 - 2s - loss: 0.0225 - val_loss: 0.0245
Epoch 26/100
36/36 - 2s - loss: 0.0225 - val_loss: 0.0249
Epoch 27/100
36/36 - 2s - loss: 0.0221 - val_loss: 0.0254
Epoch 00027: early stopping
```

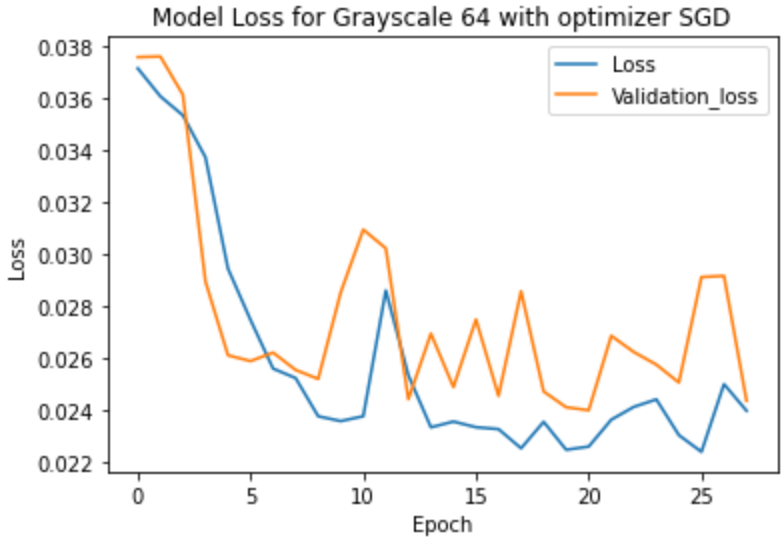


From this section, we find the second model better than the other, so we will use Model Color 64*64 with loss 0.021 and validation loss 0.025 for next sections.

Fusion Models with Different Optimizer

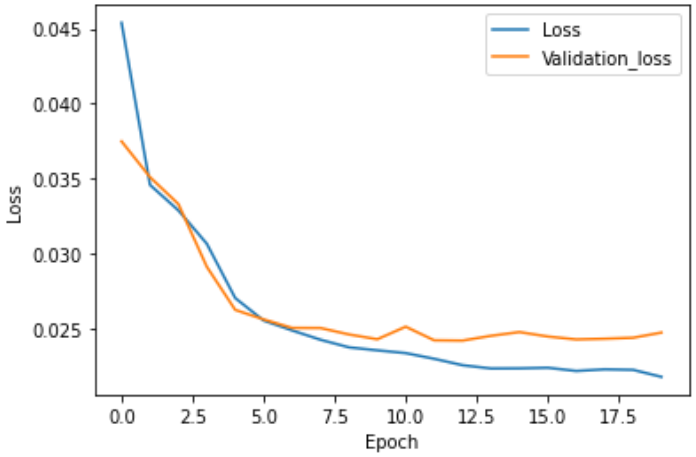
1. SGD

```
Epoch 1/100
WARNING:tensorflow:Callbacks method `on_train_
36/36 - 3s - loss: 0.0371 - val_loss: 0.0376
Epoch 2/100
36/36 - 2s - loss: 0.0360 - val_loss: 0.0376
Epoch 3/100
36/36 - 2s - loss: 0.0353 - val_loss: 0.0361
Epoch 4/100
36/36 - 2s - loss: 0.0337 - val_loss: 0.0289
Epoch 5/100
36/36 - 2s - loss: 0.0294 - val_loss: 0.0261
Epoch 6/100
36/36 - 2s - loss: 0.0275 - val_loss: 0.0259
Epoch 7/100
36/36 - 2s - loss: 0.0256 - val_loss: 0.0262
Epoch 8/100
36/36 - 2s - loss: 0.0252 - val_loss: 0.0255
Epoch 9/100
36/36 - 2s - loss: 0.0238 - val_loss: 0.0252
Epoch 10/100
36/36 - 2s - loss: 0.0236 - val_loss: 0.0285
Epoch 11/100
36/36 - 2s - loss: 0.0238 - val_loss: 0.0309
Epoch 12/100
36/36 - 3s - loss: 0.0286 - val_loss: 0.0302
Epoch 13/100
36/36 - 3s - loss: 0.0253 - val_loss: 0.0244
Epoch 14/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0269
Epoch 15/100
36/36 - 2s - loss: 0.0236 - val_loss: 0.0249
Epoch 16/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0275
Epoch 17/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0245
Epoch 18/100
36/36 - 2s - loss: 0.0225 - val_loss: 0.0286
Epoch 19/100
36/36 - 2s - loss: 0.0235 - val_loss: 0.0247
Epoch 20/100
36/36 - 2s - loss: 0.0225 - val_loss: 0.0241
Epoch 21/100
36/36 - 2s - loss: 0.0226 - val_loss: 0.0240
Epoch 22/100
36/36 - 2s - loss: 0.0236 - val_loss: 0.0268
Epoch 23/100
36/36 - 2s - loss: 0.0241 - val_loss: 0.0262
Epoch 24/100
36/36 - 2s - loss: 0.0244 - val_loss: 0.0257
Epoch 25/100
36/36 - 2s - loss: 0.0230 - val_loss: 0.0251
Epoch 26/100
36/36 - 2s - loss: 0.0224 - val_loss: 0.0291
Epoch 27/100
36/36 - 2s - loss: 0.0250 - val_loss: 0.0291
Epoch 28/100
36/36 - 2s - loss: 0.0240 - val_loss: 0.0243
Epoch 00028: early stopping
```



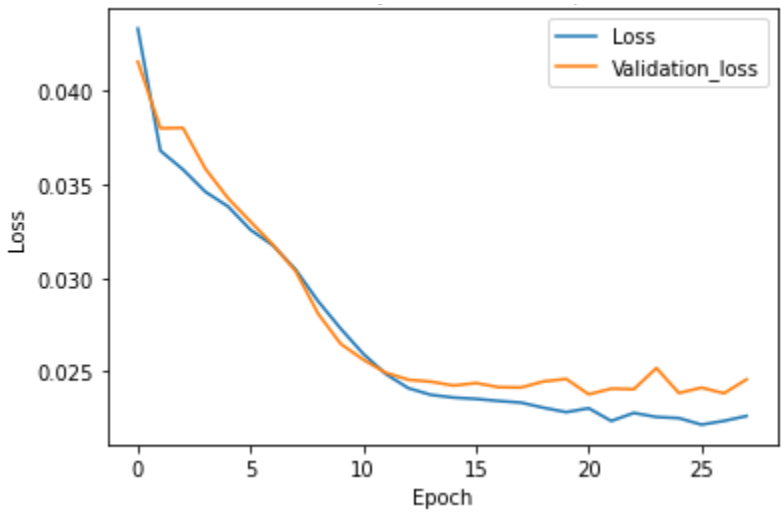
3. User Defined Adam with Learning Rate 0.001 , Decay = 0.001/200

```
Epoch 1/100
WARNING:tensorflow:Callbacks method `on_train_
36/36 - 3s - loss: 0.0454 - val_loss: 0.0375
Epoch 2/100
36/36 - 2s - loss: 0.0346 - val_loss: 0.0351
Epoch 3/100
36/36 - 2s - loss: 0.0329 - val_loss: 0.0333
Epoch 4/100
36/36 - 2s - loss: 0.0307 - val_loss: 0.0291
Epoch 5/100
36/36 - 2s - loss: 0.0270 - val_loss: 0.0262
Epoch 6/100
36/36 - 2s - loss: 0.0255 - val_loss: 0.0256
Epoch 7/100
36/36 - 2s - loss: 0.0249 - val_loss: 0.0250
Epoch 8/100
36/36 - 2s - loss: 0.0242 - val_loss: 0.0250
Epoch 9/100
36/36 - 2s - loss: 0.0237 - val_loss: 0.0246
Epoch 10/100
36/36 - 2s - loss: 0.0235 - val_loss: 0.0243
Epoch 11/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0251
Epoch 12/100
36/36 - 2s - loss: 0.0230 - val_loss: 0.0242
Epoch 13/100
36/36 - 2s - loss: 0.0225 - val_loss: 0.0242
Epoch 14/100
36/36 - 2s - loss: 0.0223 - val_loss: 0.0245
Epoch 15/100
36/36 - 2s - loss: 0.0223 - val_loss: 0.0248
Epoch 16/100
36/36 - 2s - loss: 0.0224 - val_loss: 0.0245
Epoch 17/100
36/36 - 2s - loss: 0.0222 - val_loss: 0.0243
Epoch 18/100
36/36 - 2s - loss: 0.0223 - val_loss: 0.0243
Epoch 19/100
36/36 - 2s - loss: 0.0222 - val_loss: 0.0244
Epoch 20/100
36/36 - 2s - loss: 0.0218 - val_loss: 0.0247
Epoch 00020: early stopping
```



4. Default Adam

```
Epoch 1/100
WARNING:tensorflow:Callbacks method `on_train_
36/36 - 3s - loss: 0.0433 - val_loss: 0.0415
Epoch 2/100
36/36 - 2s - loss: 0.0368 - val_loss: 0.0380
Epoch 3/100
36/36 - 2s - loss: 0.0358 - val_loss: 0.0380
Epoch 4/100
36/36 - 2s - loss: 0.0346 - val_loss: 0.0358
Epoch 5/100
36/36 - 2s - loss: 0.0338 - val_loss: 0.0343
Epoch 6/100
36/36 - 2s - loss: 0.0326 - val_loss: 0.0330
Epoch 7/100
36/36 - 2s - loss: 0.0317 - val_loss: 0.0318
Epoch 8/100
36/36 - 2s - loss: 0.0304 - val_loss: 0.0304
Epoch 9/100
36/36 - 2s - loss: 0.0288 - val_loss: 0.0281
Epoch 10/100
36/36 - 2s - loss: 0.0273 - val_loss: 0.0265
Epoch 11/100
36/36 - 2s - loss: 0.0260 - val_loss: 0.0256
Epoch 12/100
36/36 - 2s - loss: 0.0249 - val_loss: 0.0249
Epoch 13/100
36/36 - 2s - loss: 0.0241 - val_loss: 0.0246
Epoch 14/100
36/36 - 2s - loss: 0.0238 - val_loss: 0.0245
Epoch 15/100
36/36 - 2s - loss: 0.0236 - val_loss: 0.0243
Epoch 16/100
36/36 - 2s - loss: 0.0235 - val_loss: 0.0244
Epoch 17/100
36/36 - 2s - loss: 0.0234 - val_loss: 0.0242
Epoch 18/100
36/36 - 2s - loss: 0.0233 - val_loss: 0.0242
Epoch 19/100
36/36 - 2s - loss: 0.0231 - val_loss: 0.0245
Epoch 20/100
36/36 - 2s - loss: 0.0228 - val_loss: 0.0246
Epoch 21/100
36/36 - 2s - loss: 0.0230 - val_loss: 0.0238
Epoch 22/100
36/36 - 2s - loss: 0.0224 - val_loss: 0.0241
Epoch 23/100
36/36 - 2s - loss: 0.0228 - val_loss: 0.0241
Epoch 24/100
36/36 - 2s - loss: 0.0226 - val_loss: 0.0252
Epoch 25/100
36/36 - 2s - loss: 0.0225 - val_loss: 0.0239
Epoch 26/100
36/36 - 2s - loss: 0.0222 - val_loss: 0.0241
Epoch 27/100
36/36 - 2s - loss: 0.0224 - val_loss: 0.0238
Epoch 28/100
36/36 - 2s - loss: 0.0226 - val_loss: 0.0246
Epoch 00028: early stopping
```



The default model has the least loss.

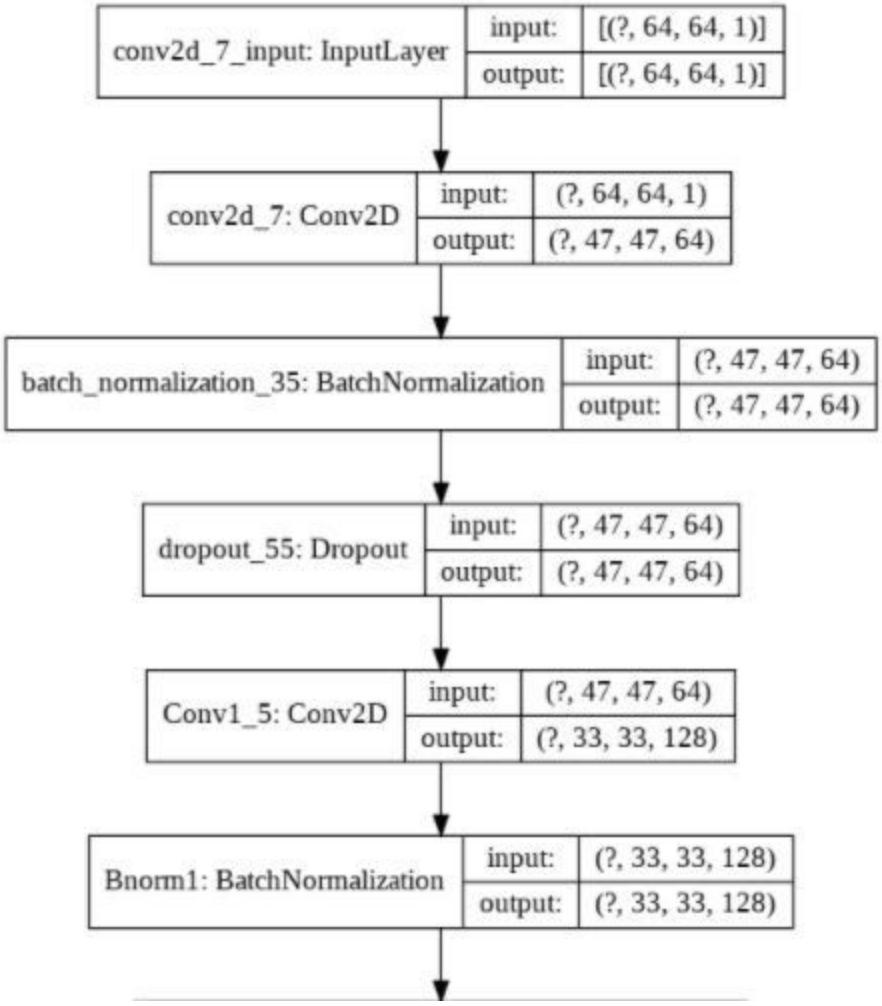
Final Model Summary:

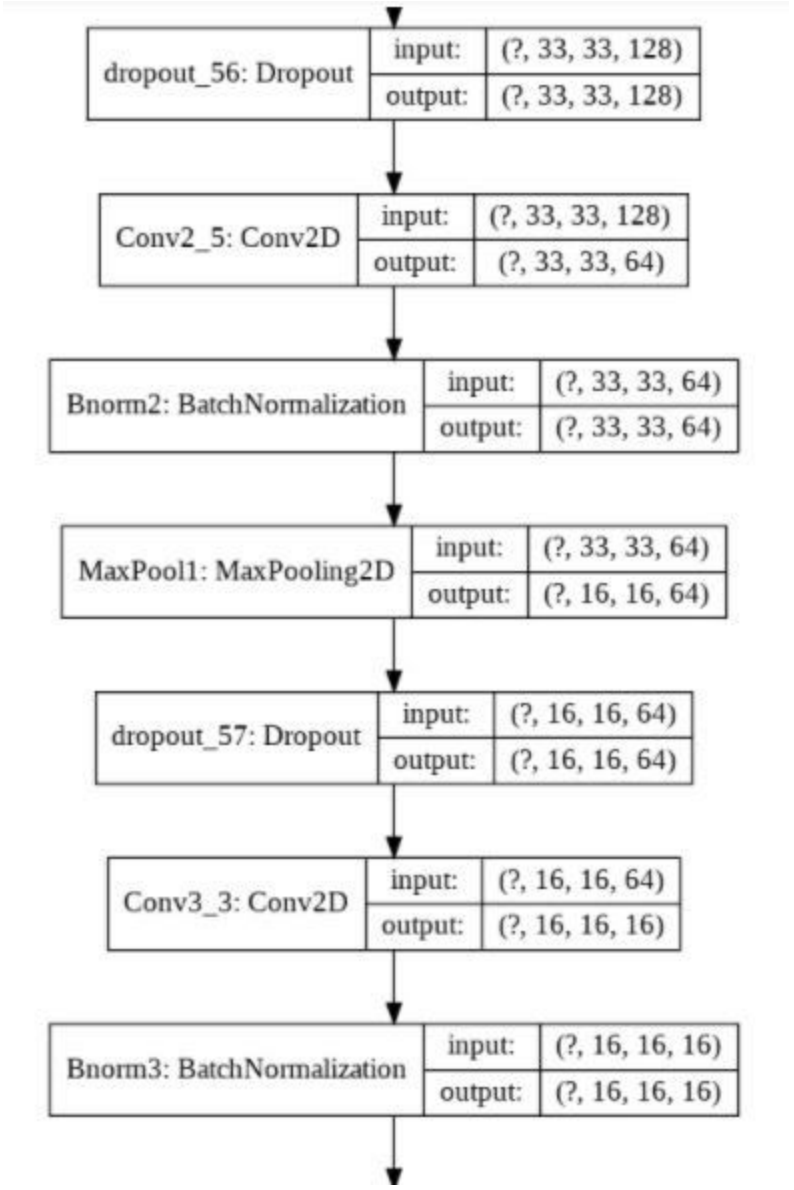
```
Model: "functional_13"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
conv2d_7_input (InputLayer)  [(None, 64, 64, 1)]       0
_________________________________________________________________________________________________
conv2d_7 (Conv2D)            (None, 47, 47, 64)        20800       conv2d_7_input[0][0]
_________________________________________________________________________________________________
batch_normalization_35 (BatchNo (None, 47, 47, 64)     256         conv2d_7[0][0]
_________________________________________________________________________________________________
dropout_55 (Dropout)         (None, 47, 47, 64)        0           batch_normalization_35[0][0]
_________________________________________________________________________________________________
Conv1_5 (Conv2D)             (None, 33, 33, 128)       1843328     dropout_55[0][0]
_________________________________________________________________________________________________
Bnorm1 (BatchNormalization)  (None, 33, 33, 128)       512         Conv1_5[0][0]
_________________________________________________________________________________________________
dropout_56 (Dropout)         (None, 33, 33, 128)       0           Bnorm1[0][0]
_________________________________________________________________________________________________
Conv2_5 (Conv2D)             (None, 33, 33, 64)        663616      dropout_56[0][0]
_________________________________________________________________________________________________
Bnorm2 (BatchNormalization)  (None, 33, 33, 64)        256         Conv2_5[0][0]
_________________________________________________________________________________________________
MaxPool1 (MaxPooling2D)      (None, 16, 16, 64)        0           Bnorm2[0][0]
_________________________________________________________________________________________________
dropout_57 (Dropout)         (None, 16, 16, 64)        0           MaxPool1[0][0]
_________________________________________________________________________________________________
Conv3_3 (Conv2D)             (None, 16, 16, 16)        50192       dropout_57[0][0]
_________________________________________________________________________________________________
Bnorm3 (BatchNormalization)  (None, 16, 16, 16)        64          Conv3_3[0][0]
_________________________________________________________________________________________________
Conv4_3 (Conv2D)             (None, 16, 16, 32)        18464       Bnorm3[0][0]
_________________________________________________________________________________________________
Bnorm4 (BatchNormalization)  (None, 16, 16, 32)        128         Conv4_3[0][0]
_________________________________________________________________________________________________
dropout_58 (Dropout)         (None, 16, 16, 32)        0           Bnorm4[0][0]
_________________________________________________________________________________________________
AvgPool1 (AveragePooling2D)  (None, 8, 8, 32)          0           dropout_58[0][0]
_________________________________________________________________________________________________
Conv5_1 (Conv2D)             (None, 6, 6, 128)         36992       AvgPool1[0][0]
_________________________________________________________________________________________________
```

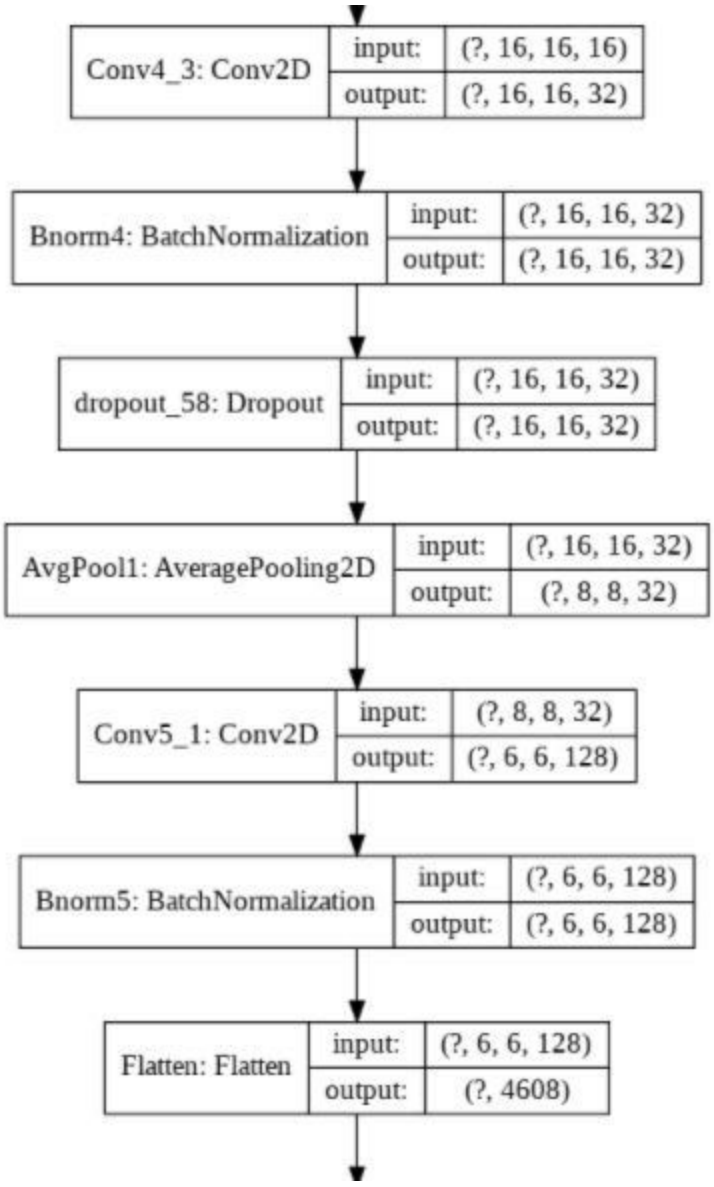
Flatten (Flatten)	(None, 4608)	0	Bnorm5[0][0]
dense_100 (Dense)	(None, 64)	294976	Flatten[0][0]
dropout_59 (Dropout)	(None, 64)	0	dense_100[0][0]
batch_normalization_36 (BatchNo	(None, 64)	256	dropout_59[0][0]
dense_101 (Dense)	(None, 32)	2080	batch_normalization_36[0][0]
dropout_60 (Dropout)	(None, 32)	0	dense_101[0][0]
batch_normalization_37 (BatchNo	(None, 32)	128	dropout_60[0][0]
dense_102 (Dense)	(None, 15)	495	batch_normalization_37[0][0]
dropout_61 (Dropout)	(None, 15)	0	dense_102[0][0]
dense_97_input (InputLayer)	[(None, 5)]	0	
batch_normalization_38 (BatchNo	(None, 15)	60	dropout_61[0][0]
dense_97 (Dense)	(None, 8)	48	dense_97_input[0][0]
dense_103 (Dense)	(None, 6)	96	batch_normalization_38[0][0]
dense_98 (Dense)	(None, 4)	36	dense_97[0][0]
batch_normalization_39 (BatchNo	(None, 6)	24	dense_103[0][0]
dense_99 (Dense)	(None, 1)	5	dense_98[0][0]
preds (Dense)	(None, 1)	7	batch_normalization_39[0][0]
concatenate_6 (Concatenate)	(None, 2)	0	dense_99[0][0] preds[0][0]
dense_104 (Dense)	(None, 32)	96	concatenate_6[0][0]
dropout_62 (Dropout)	(None, 32)	0	dense_104[0][0]
dense_105 (Dense)	(None, 16)	528	dropout_62[0][0]
dense_106 (Dense)	(None, 8)	136	dense_105[0][0]
dense_107 (Dense)	(None, 1)	9	dense_106[0][0]

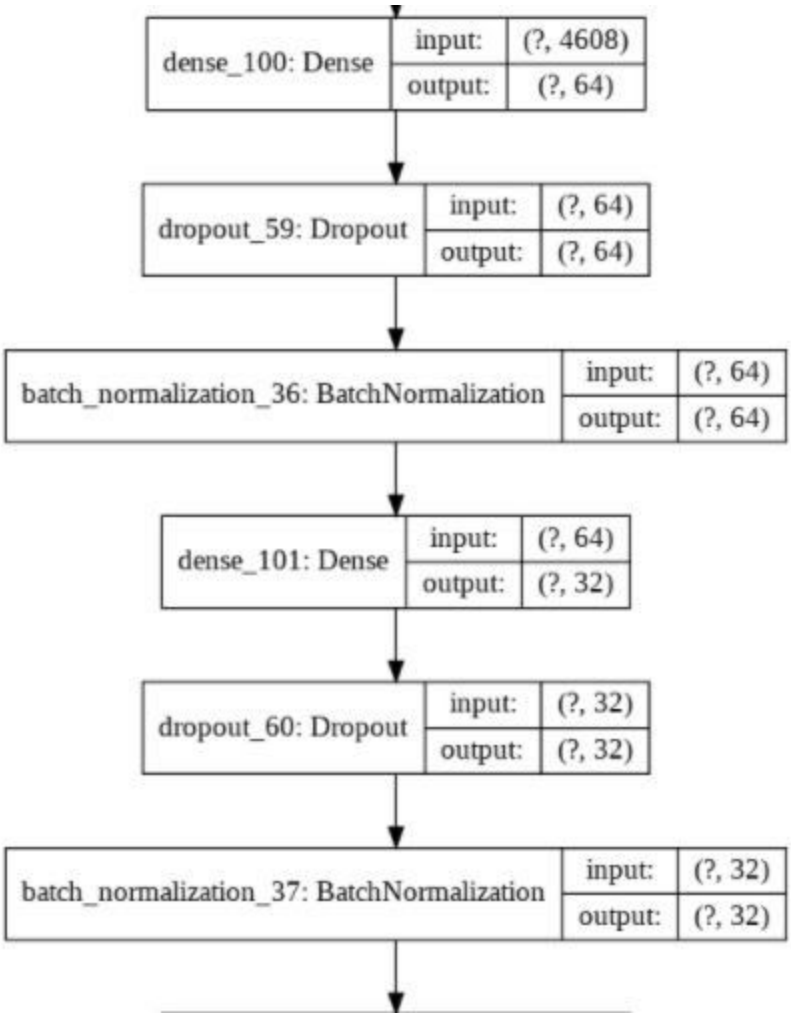
Total params: 2,934,100
Trainable params: 2,933,002
Non-trainable params: 1,098

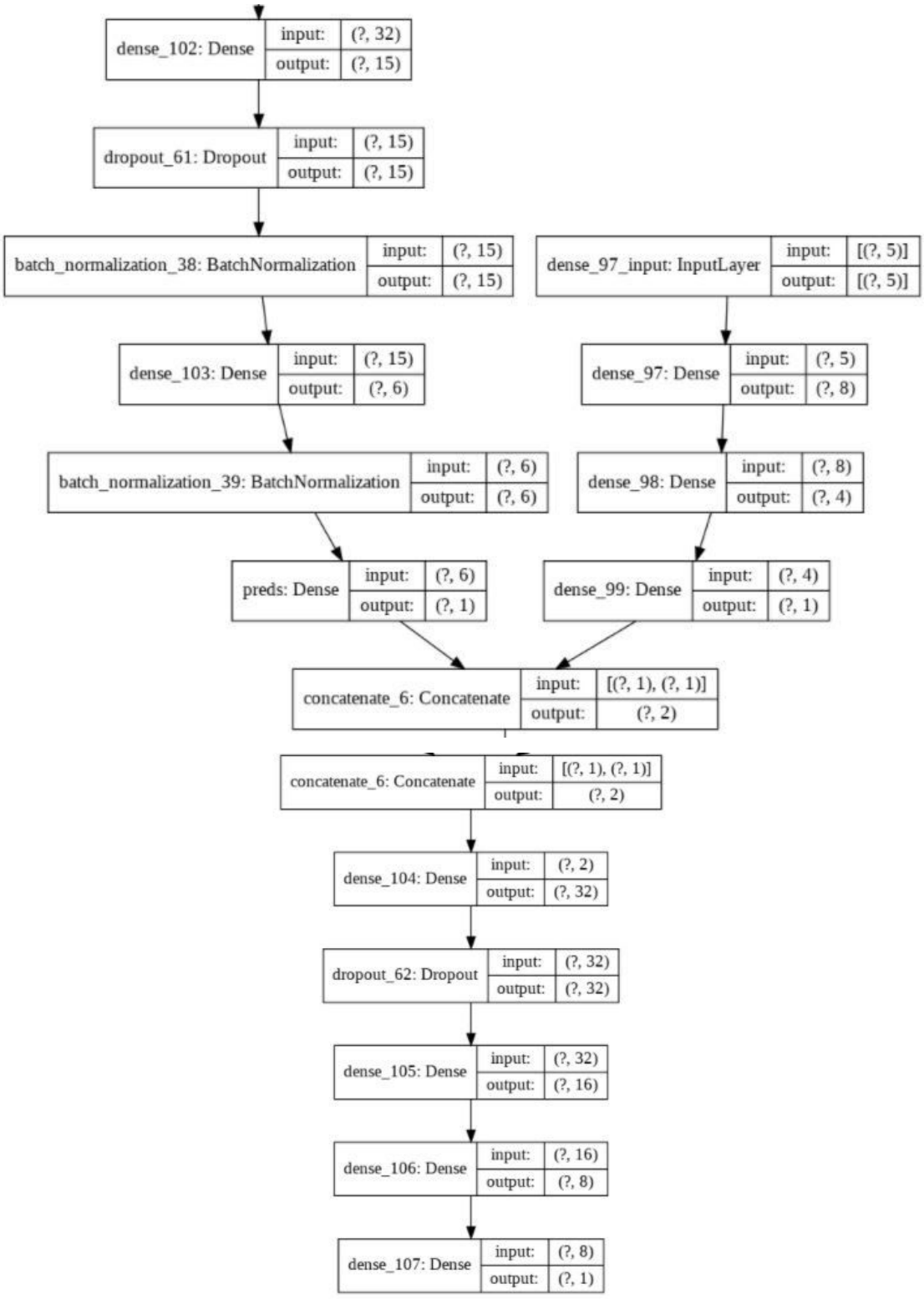
Final Model Plot:




dropout_56: Dropout
input: (?, 33, 33, 128)
output: (?, 33, 33, 128)
Conv2_5: Conv2D
input: (?, 33, 33, 128)
output: (?, 33, 33, 64)
Bnorm2: BatchNormalization
input: (?, 33, 33, 64)
output: (?, 33, 33, 64)
MaxPool1: MaxPooling2D
input: (?, 33, 33, 64)
output: (?, 16, 16, 64)
dropout_57: Dropout
input: (?, 16, 16, 64)
output: (?, 16, 16, 64)
Conv3_3: Conv2D
input: (?, 16, 16, 64)
output: (?, 16, 16, 16)
Bnorm3: BatchNormalization
input: (?, 16, 16, 16)
output: (?, 16, 16, 16)


Conv4_3: Conv2D
input: (?, 16, 16, 16)
output: (?, 16, 16, 32)
Bnorm4: BatchNormalization
input: (?, 16, 16, 32)
output: (?, 16, 16, 32)
dropout_58: Dropout
input: (?, 16, 16, 32)
output: (?, 16, 16, 32)
AvgPool1: AveragePooling2D
input: (?, 16, 16, 32)
output: (?, 8, 8, 32)
Conv5_1: Conv2D
input: (?, 8, 8, 32)
output: (?, 6, 6, 128)
Bnorm5: BatchNormalization
input: (?, 6, 6, 128)
output: (?, 6, 6, 128)
Flatten: Flatten
input: (?, 6, 6, 128)
output: (?, 4608)


dense_100: Dense
input: (?, 4608)
output: (?, 64)
dropout_59: Dropout
input: (?, 64)
output: (?, 64)
batch_normalization_36: BatchNormalization
input: (?, 64)
output: (?, 64)
dense_101: Dense
input: (?, 64)
output: (?, 32)
dropout_60: Dropout
input: (?, 32)
output: (?, 32)
batch_normalization_37: BatchNormalization
input: (?, 32)
output: (?, 32)


dense_102: Dense
input: (?, 32)
output: (?, 15)
dropout_61: Dropout
input: (?, 15)
output: (?, 15)
batch_normalization_38: BatchNormalization
input: (?, 15)
output: (?, 15)
dense_97_input: InputLayer
input: [(?, 5)]
output: [(?, 5)]
dense_103: Dense
input: (?, 15)
output: (?, 6)
dense_97: Dense
input: (?, 5)
output: (?, 8)
batch_normalization_39: BatchNormalization
input: (?, 6)
output: (?, 6)
dense_98: Dense
input: (?, 8)
output: (?, 4)
preds: Dense
input: (?, 6)
output: (?, 1)
dense_99: Dense
input: (?, 4)
output: (?, 1)
concatenate_6: Concatenate
input: [(?, 1), (?, 1)]
output: (?, 2)
concatenate_6: Concatenate
input: [(?, 1), (?, 1)]
output: (?, 2)
dense_104: Dense
input: (?, 2)
output: (?, 32)
dropout_62: Dropout
input: (?, 32)
output: (?, 32)
dense_105: Dense
input: (?, 32)
output: (?, 16)
dense_106: Dense
input: (?, 16)
output: (?, 8)
dense_107: Dense
input: (?, 8)
output: (?, 1)

Predictions:

	Actual Price	Predicted Price	Difference
1	1395000	816214.0	578786.0
2	345000	740135.0	-395135.0
3	749000	1261676.0	-512676.0
4	299000	766433.0	-467433.0
5	1888888	739600.0	1149288.0
6	355000	754063.0	-399063.0
7	898000	847020.0	50980.0
8	1049000	919889.0	129111.0
9	360000	867426.0	-507426.0
10	499900	739642.0	-239742.0

Standard Deviations:

On Predicted Price : 201759.8

On Actual Price : 404902.0

Loss on Validation Set: 0.065

In conclusion, we could only use descriptive data because we do not gain so much better results by using images, also it takes a long time to be learned.