Improving Real-Estate Price Predictions with Images

CS 542: Final Project Report

Hannah Catabia catabia@bu.edu
Mona Jalal jalal@bu.edu
Ramesh Krishnan S. Pallavoor rameshkp@bu.edu
Nithin Varma nvarma@bu.edu

December 12, 2017

1 Project Task

Our task was to improve real estate price predictions for Foxy Real Estate company. Specifically, we used image data to augment the numerical and categorical data [1] (i.e., zip code, square footage) that the company currently uses for price prediction.

2 Related Work

Real estate price prediction is a difficult problem, which is evidenced by the sheer number of companies competing with each other in designing proprietary algorithms for the task. According to a USNews article that appeared in 2015 [Mea15], there is a large difference in the prices estimated by several leading companies for the same listings. The article discusses the example of a neighborhood in Miami where the houses vary a lot with regard to condition. According to the article, the prices predicted by various companies such as Zillow, Eppraisal, Redfin, ForSaleByOwner.com, Chase, etc., lie in the range of $220,000 - $470,000.

Some real estate price prediction work has appeared in the last two years that studies the effect of augmenting images with the metadata available for the listings. In [YPCL17], the authors cluster the neighborhoods using a random walk and use recurrent neural networks (RNNs) to evaluate the prices of houses from the same neighborhood. Bappy et al. [BBSR17] consider the problem of automating image annotation with labels like bedroom, bathroom, kitchen etc. As part of their work, they released a comprehensive annotated dataset called REI (Real Estate Images) that can serve as a benchmark dataset for house image classification tasks. In [PMB17], the authors compiled two different datasets, one for room type detection and one for luxury-level detection. The authors additionally explored the relation between luxury level and property prices. After detecting the room type and luxury level of each room, they added it to metadata from Zillow to improve price estimation via a dense neural network. They achieved 2.1% less median error rate compared to that of Zestimate. Upchurch et al. [BUSB15] studied the related problem of material detection from images. Their algorithms are able to do pixel-wise segmentation for the images based on the material types in them, as well as classify *patches* of images into 23 different categories. They also released an annotated dataset MINC-2500 to serve as a benchmark for material detection.

3 Approach

The first step in our approach was to preprocess all of our image data using a preexisting neural network architecture, with weights fine-tuned on house images. We then combined the preprocessed image data with the metadata for each listing and fed it to a fully-connected neural network (FCN) that predicts home prices. All of our neural networks were produced using the Keras [Cho15] API with TensorFlow [ABC+16] as backend.

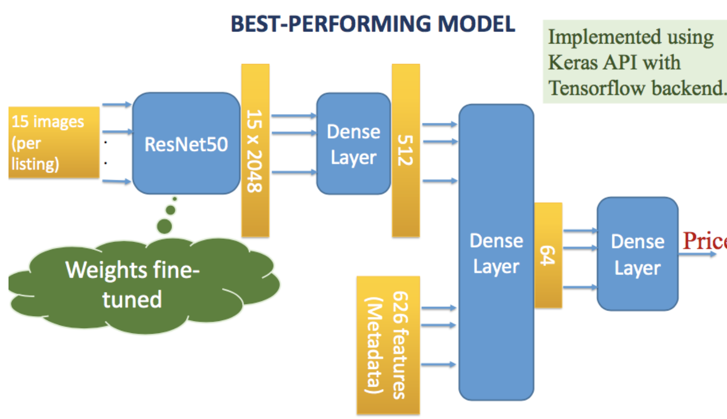


Figure 1: The neural network model we used to predict home prices

For preprocessing, we used the Keras model of ResNet50 [HZRS16], loaded with weights pre-trained on ImageNet [RDS+15]. We then fine-tuned this neural network with real estate images shared with us by the authors of [PMB17]. We preprocessed each image by passing it through the fine-tuned neural network and extracting the last flat layer before category prediction, and obtained a vector with 2,048 features.

Since each real estate listing had 15 images associated with it, we concatenated the preprocessed image outputs and fed them into a FCN layer that outputs a vector of 512 features. We paired this output with the metadata from

[1] In this report, we will refer to the numerical and categorical data associated with each listing as "metadata."

each real estate listing, and fed them into another FCN layer with a 64-feature output. One final FCN layer predicted the price. An illustration of our architecture is given in Figure 1.

4 Datasets, Metrics, and Training Process

In this section, we briefly describe the details of the datasets that we used to train and test our neural networks. We also discuss the different metrics that we used to evaluate the performance of our network.

4.1 Foxy dataset

The dataset provided by Foxy includes 54,760 listings, 34,829 of which has a complete set of 15 images associated with them. We decided to use a subset of these listings to train our neural network. Figures 2a-2c provide a graphical summary of this data.

4.2 Preprocessing metadata

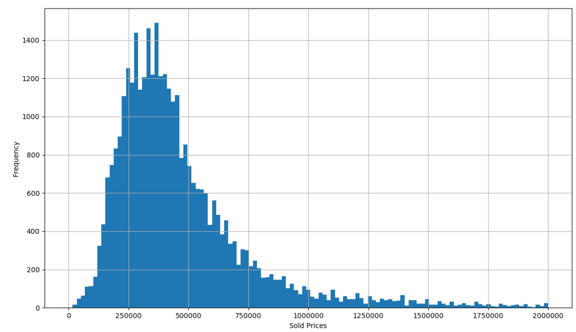
We used five attributes of numerical data from the Foxy CSV file: square footage, building age, parking lot size, number of bedrooms, and number of bathrooms. We also used zip code data in one-hot encoding format and concatenated it with the numerical data, resulting in a vector of length 626 for each listing. We would like to note that we did not normalize the numerical data because we found our model performed better without normalization. We think this is because square footage has a large impact on sales price, and foregoing normalization made this feature have a greater impact.

4.3 Dataset and metrics for fine-tuning

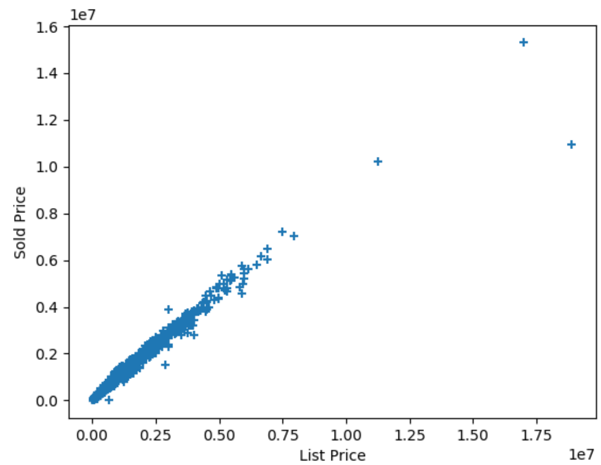
We fine-tuned ResNet50 using a room type categorization dataset shared with us by the authors of [PMB17]. This dataset consists of 149,994 training set images, and 8,545 images, split into seven categories as follows: exterior, interior, bathroom, bedroom, dining room, and kitchen.

After loading ResNet50 with weights pre-trained on ImageNet, we replaced the final layer with our own FCN layer with softmax activation for 7 categories. Then we froze the first 11 convolutional layers of ResNet50, chose categorical cross entropy for loss and stochastic gradient descent (SGD) for optimization, and set the learning rate (lr) to 0.0001. We trained on 75 images per batch, with 500 steps per epoch (about 25% of the training set was used in each epoch). For validation, we also used batches of 75 images over 25 steps (about 22% of the testing set was used in each epoch).

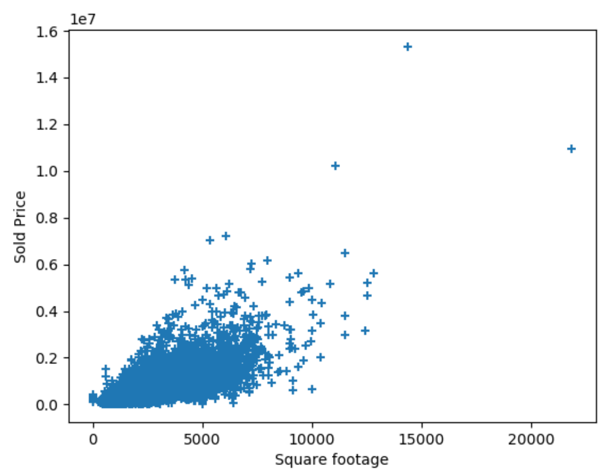
After 25 epochs of training, we noticed the diminishing returns on the accuracy. To save time, we decided to stop fine-tuning at this point. We used our fine-tuned model with the highest accuracy for preprocessing. This was our model from the 24th epoch, which had a training loss of 0.23, a test loss of 0.30, a training accuracy of 0.92, and a test accuracy of 0.90.



(a) Histogram of sold price



(b) Sold price plotted against List Price



(c) Sold price plotted against Square Footage

Figure 2: Data Visualization

4.4 Training process and price prediction metrics

We split the Foxy dataset into training (80%), validation (10%), and test (10%) sets. We used the holdout validation method and the RMSProp optimizer, and set the training batch size to 32. For the loss function, we used the mean

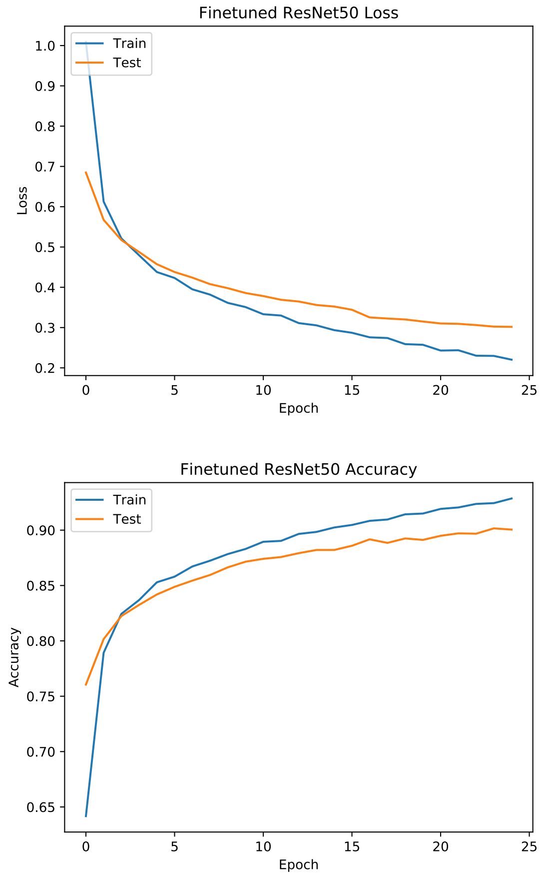


Figure 3: Loss and accuracy plots from fine-tuning ResNet50 with room type dataset from [PMB17]

squared error (MSE) between the predicted prices and actual sale prices. We also tracked mean absolute error (MAE) and mean absolute percentage error (MAPE) at each epoch. If there are m listings, such that the actual selling price and the predicted price of the i^{th} listing are denoted by a_i and p_i respectively, the aforementioned quantities are,

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(p_i - a_i)^2$$

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}|p_i - a_i|$$

$$\text{MAPE} = \frac{100}{m}\sum_{i=1}^{m}\frac{|p_i - a_i|}{a_i}.$$

We also used an early-stopping approach to avoid overfitting on the training data, which stops the training if the best validation MAPE did not change in previous 5 consecutive epochs.

5 Evaluation

We tested four basic versions of our model to see which ones predicted price most accurately: (a) using metadata only, (b) using images only, (c) using metadata and images preprocessed with ResNet50, and (d) using metadata and images preprocessed with fine-tuned ResNet50.

5.1 Metadata or Images only

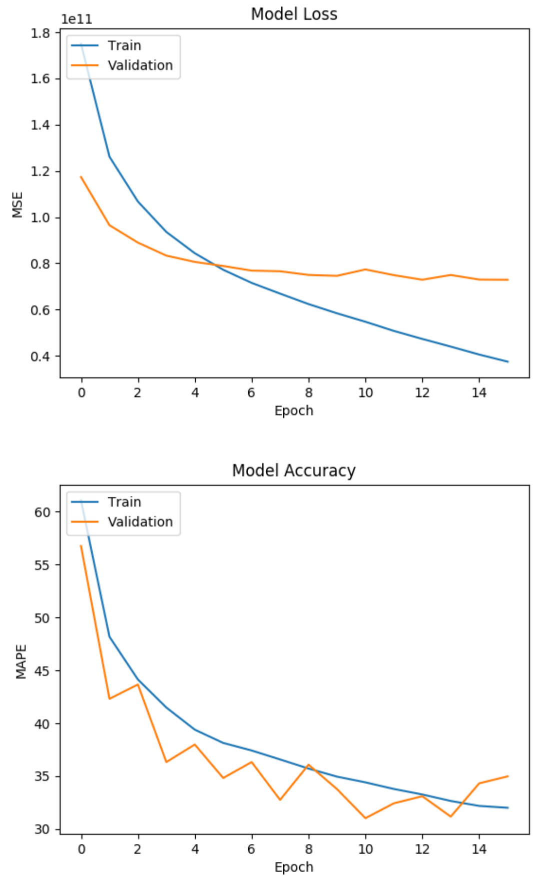


Figure 4: Loss and accuracy plots for price prediction using images only.

When working with either only images or only metadata, we found that the images made better price predictions. When we trained a neural network only on metadata, the MAPE was 40.35%, the MAE was $156,087. On the other hand, when we trained a neural network on just image data, we achieved a MAPE of 33.23% and a MAE of $141,977.

5.2 Metadata and Images Together

Models using the combination of images and metadata performed the best. When we used ResNet50 to preprocess without fine-tuning it, our MAPE was 29.02% and our MAE was $127,556. However, fine-tuning ResNet50 led to our best results: a MAPE of 27.75% and a MAE of $124,641. It should be noted that these errors are nonetheless quite large and leave lots of room for improvement (such as incorporating the materials detected in each property to the metadata). However, they are not terribly far from some other work done in real estate price prediction; for instance, [YPCL17] created a model that predicted prices in Rochester, New York with a MAPE of 22.69%.

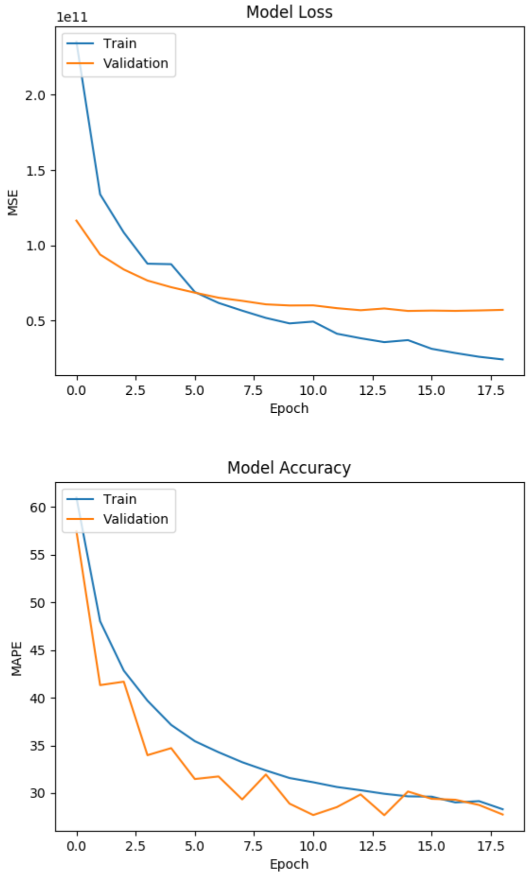


Figure 5: Loss and accuracy plots from from predicting prices with metadata and fine-tuned images

Foxy provided us with another metric to measure our success: to see if we could beat current industry best standards in margin of error, which is predicting within 10% of the sale price, 70-75% of the time. By this metric, we certainly failed: we were only able to predict within 30% of the sales price 70% of the time (See Figure 6 for a breakdown of our error margins).

Absolute percentage error	Fraction of data
≤ 5%	13%
≤10%	26%
≤ 15%	39%
≤ 20%	50%
≤ 25%	61%
≤ 30%	70%

Figure 6: Margins of error of our best-performing model

5.3 List Price as a Feature

There is one numerical feature in the Foxy dataset we did not use in the models presented thus far: list price. This is the price that the seller lists with the real estate listing, hoping to get an offer of matching value. If we were to predict sales price by just restating the list price, our predictions would have been much more accurate than what our neural network predicted: on average, the list price deviates from the sales price by only 6%.

It did not seem fair to us to use listing price as a feature, as this is "borrowing" from human intuition about sales price prediction. However, there could be justification for doing so: perhaps someone trying to decide how much money to offer on a house could use list price as a data point. Indeed, some researchers have included it in their models; for instance, [PMB17] achieved a median error rate of 5.8% when using list price as a feature in conjunction with images and other metadata.

As an experiment, we added list price (not normalized) to the data fed to our neural network, to see how this model would perform. Our predictions were, on average, within 3.2% of the sales price–a definite improvement on the 6% margin of error from using list price alone.

6 Conclusion and Future Work

Our best performing model for real estate price prediction used both images and metadata, showing that image data can be a useful asset in this field. First, we preprocessed Foxy dataset images through ResNet50, which was fine-tuned using a room detection dataset from [PMB17]. Next, we passed image data and metadata through a FCN utilizing 3 dense layers, which outputs a price prediction. In the end, our neural network was not able to beat humans at prediction, as list price still predicted sales price far better than our model. However, our model was able to improve on human intuition when it used list price as a feature. Further work may be done to improve both types of models. One concrete idea for future work is to use the output of the last layer of the materials detection neural network by Upchurch et al. [BUSB15] along with the data that we are currently using. A possible advantage of this direction would be that we would be able to incorporate information about the various kinds of materials in houses (e.g., formica vs. granite) and learn their influence on house price. Another future path is incorporating the luxury-level dataset (once it gets released) from [PMB17] to our framework and train our model using it. This way we would have a new metadata as luxury level based on average luxury level of all the images from one property listing.

7 Roles

All the team members put in at least 30-40 hours of work on various aspects of the project. The following table summarizes the contributions of each member.

Task	Lead	Files	#Lines
CSV data cleaning	Hannah	LEcleaning.py	50
Finetuning	Hannah	luxtune.py	100
Preprocess images through untuned and fine-tuned ResNet	Nithin	image-preprocessing-through-resnet.py, csv-processor.py	100
Metadata only price prediction	Nithin	meta-data-only-price-prediction.py	60
Trying different neural network architecture and finding the best one	Ramesh	images-only-price-prediction.py, metadata-image-price-prediction.py	100 (each)
Summarizing results and performance plots	Ramesh		
Poster	Nithin		
Report	Hannah Mona		
Sys Admin, setting up and using material segmentation framework, acquring and cleaning REI, room type dataset and material (MINC) dataset	Mona	additional help with coding and Keras, troubleshooting, and related study	

8 GitHub Link

The URL to our GitHub repository is given below. Our codes and the trained models are available there.
`https://github.com/nithvarma/ML-Project-at-BU---Real-Estate-Price-Prediction`.

References

[ABC+16] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[BBSR17] Jawadul H. Bappy, Joseph R. Barr, Narayanan Srinivasan, and Amit K. Roy-Chowdhury. Real estate image classification. In *2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, Santa Rosa, CA, USA, March 24-31, 2017*, pages 373–381, 2017.

[BUSB15] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3479–3487, 2015.

[Cho15] François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[Mea15] Teresa Mears. 7 online tools to help you estimate your home's value. 2015.

[PMB17] Omid Poursaeed, Tomas Matera, and Serge J. Belongie. Vision-based real estate price estimation. *CoRR*, abs/1707.05489, 2017.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, December 2015.

[YPCL17] Quanzeng You, Ran Pang, Liangliang Cao, and Jiebo Luo. Image based appraisal of real estate properties. *IEEE Trans. Multimedia*, PP(99), 2017.