CS231N Project: Visual Price Estimation for Real Estate

Sam Lowe
Stanford University
samlowe@stanford.edu

Svetak Sundhar
Stanford University
svetak@stanford.edu

Abstract

Motivated by the widespread adoption of deep learning techniques across a variety of industries, we explore the potential for convolutional neural networks operating on visual features to assist in the valuation of real estate properties. After a brief review of existing approaches to this problem, we propose a network architecture capable of utilizing multimodal features to predict the price of a house based on images and textual descriptors of the property. In preprocessing our data, we generated tiled images combining external views of a home with photos of a sample bedroom, bathroom, and kitchen from the property. To quantify the potential gains in fusing this modality with the typical descriptors of real estate, such as the number of bedrooms and bathrooms, we experimented with simple models operating solely on the visual or textual features as a baseline. Observing that neither was sufficient in isolation, we built a multimodal network on top of a pretrained GoogLeNet architecture, and after a wide hyperparameter search, achieved a mean average error in price prediction across our test set of $1635.45. To better understand the performance of our final model, we also performed a brief ablation study and generated saliency maps for several datapoints. It is our hope that our explorations of the price estimation problem in particular can garner insights to better understand other questions in the industry.

1. Introduction

Real estate valuation is a costly and time-consuming process that typically requires a domain expert, an appraiser, to physically visit a particular property in order to make their assessment of its value. However, real estate prices are valuable data for many stakeholders in a given community, and creating a low-cost alternative to the usual process for appraisal has the ability to provide substantial benefits across the board for these various groups. For homeowners, real estate valuations can deliver insight into the appreciation, or depreciation, of what is, for many, their largest asset. Urban planners and local governments could use the data across communities to target services and bolster struggling neighborhoods. Real estate agencies and investors could also benefit by using pricing data to guide investment and development. Additionally, agencies can benefit from better understanding which features of a house strongly influence property price. Motivated by the myriad impacts that can result from an alternative valuation process, we are interested in exploring the potential for convolutional neural networks (CNNs) to craft visual features based on images of properties to estimate their prices. We believe that using CNNs to automatically generate features from images can work in tandem with textual information about a property (such as number of bedrooms and bathrooms) to ultimately provide an assessment of property value.

1.1. Problem Statement

The task of generating real estate valuations is fundamentally a regression problem. That is, we want to take as input a set of features describing a particular property and apply some continuous function to it that will return a real-valued, nonnegative estimate of the property's value, ranging from the tens of thousands up to multiple millions of dollars. The particular formation of the problem under study takes as input a combination of visual and textual features, where the visual features consist of a set of images depicting various internal and external parts of a property and the textual features include fundamental descriptors of the home, including the number of bedrooms and bathrooms and the area in which it is located. It is our belief that a neural model will perform the best as the continuous function approximator mentioned above. Particularly, we will construct a model that uses convolutional layers to extract features from our images and combine the generated features with a representation of our textual data before passing it through a series of fully-connected layers that will generate the final valuation estimate.

2. Related Work

2.1. Textual Networks for Price Estimation

The topic of real estate price estimation is a relatively popular one in the literature, but many approaches focus only on textual descriptors of properties, often in combination with general economic measures. Both Sun [13] and Li & Chu [10] explored the applicability of neural models to the problem of valuation, using economic descriptors of a property and its local area as inputs to their models. Sun designed a simple feed-forward neural network for the task and utilized a genetic algorithm to iteratively compute the optimal initialization for the network, while Li & Chu attempted to compare the performance of a standard feed-forward network with a radial basis function network, but found no conclusive results favoring one over the other. Similar studies at predicting price based on textual descriptors have attempted to find the best set of input features or the best type of model for the task [1] [11]. In contrast to these approaches, we believe that visual descriptors of a property are a critical component in estimating its price and must be considered in tandem with textual and economic features to produce accurate results.

2.2. Neural Networks for Investment Strategy

Given that one potential application for a price-estimation model is in guiding real estate investment, we also explored related work in applying deep networks to investment strategy. Heidari & Rafatirad [4] used a transfer learning approach to fine tune a bidirectional attention model aimed at the task of rent prediction. This data could then be leveraged to compare mortgage prices against rent potential to guide home buyers towards safer investments. While our model does not explicitly identify potential investment opportunities, a well-trained price estimator could be used to identify properties that are over- or under-valued on the market, but we consider that a side effect of solving the problem at hand and not a core motivation guiding our research.

2.3. Valuation from Visual Features

Several previous projects have explored the ability of visual data to provide a meaningful signal for real estate valuation purposes. Poursaeed et al. attempted to use crowd-sourced labels reflecting the level of luxury depicted in images from various room types to train a convolutional network to approximate the perceived luxury of a property as an additional datapoint for neural network-based valuation [12]. This essentially reduced the problem of training the CNN to a multi-class classification problem before concatenating the resulting luxury level with other relevant metadata for regression. Law et al. augmented traditional appraisal metrics, like age and size, with images of both the property itself and satellite imagery of the surrounding area [8]. In their work, a CNN model was used to extract general features, and as such, was trained as part of their end-to-end architecture, rather than separately, as in [12]. The authors explored two variations on the post-feature extraction stage of their price-estimation architecture, as they were interested in comparing the accuracy that could be achieved by a "black-box" multi-layer perceptron with the interpretability provided by a linear model, which they used to learn proxy variables for the visual desirability of neighborhoods that could generalize across different learning and modeling tasks. Ahmed & Moustafa also explored the combination of textual and visual features to estimate house prices in several communities across California using neural models, but their work relied on combining hand-crafted visual features drawn from representative images of different room types with the textual data for use with a fully-connected model, rather than automate the feature extraction process with CNNs [7]. Other works that have applied visual features to the price estimation problem are more tangentially related to our work, such as the random walk-based RNN proposed by You et al. that uses sequential data to encode neighborhood relations and the CNN model of Elnagar & Thomas that aims to identify damage in property photos [15] [3]. Based on these related works, we believe that our development of an end-to-end fully multimodal model is novel for this particular problem.

3. Dataset

In this work, we will use the dataset proposed by [7], which consists of data for 535 different properties across the State of California. The dataset represents a combination of textual and visual features for each property, with the textual data consisting of the number of bedrooms, the number of bathrooms, the area of the house in square feet, and the zipcode of the property. The visual data for each property consists of four images, one taken from in front of the house, and one each of a bedroom, bathroom, and kitchen. The prices for the properties represented in the data set ranged from $22,000 to $5,858,000.

3.1. Pre-Processing

We split the dataset into a training set of size 428 and a test set of 107, representing an 80/20 split of the original dataset, and further reserved 15% of our training set for validation purposes. To preprocess our data, we began by naming each column and dropping all NA/NAN values, as we believed that was the best way to handle the NA values (approximating would not have been beneficial in our use case). We then encoded the zipcode as a one-hot vector over the zipcodes present in the dataset, split features and labels from the data, and used a MinMax scaler to transform our values into the range [0,1]. Before the final scal-

Figure 1. Sample tiled and normalized image from our dataset.

ing of our property prices, we also log-scaled all the price labels to better handle the wide range of prices observed in our dataset. For the images, we created a collage of the 4 images per property, each sized at 112 x 112, to have a final output image of 224 x 224. We then divide our images by 255 as commonly done in Computer Vision literature, to describe the 0-255 RGB pixel range within a 0.0-1.0 range. A sample of one of our tiled, normalized images is given in Figure 1. For our multimodal network, we further normalized our images using the statistics from the dataset used to pre-train the GoogLeNet model to ensure it could extract the proper features. Note that our preprocessing was loosely based off of the work done in the repository from [7], as we believed this was an efficient means of going about cleaning and preprocessing our dataset.

4. Method

4.1. Transfer Learning

Our ultimate aim in this project is designing a model architecture that can operate over both modalities of data that we believe are relevant to solving the valuation problem: textual and visual. However, the constrained size of our dataset presents a challenge, particularly in regards to training the convolutional portion of the network. State-of-the-art convolutional neural networks are often very deep, with dozens of network layers and millions of trainable weights, and an attempt to train such a model on our compact dataset would be far more likely to result in extreme overfitting rather than recovering any semantically meaningful interpretation of the data. Our data-poor training regime thus necessitates transfer learning. Transfer learning is a process by which a model that has been pre-trained for a particular task, oftentimes the ImageNet object recognition challenge [2], is re-purposed for a new learning objective. Transfer learning is a common practice across computer vision research due to the computational expense of training a deep CNN model from scratch and is generally a tractable approach given the generalizability of the features learned in early layers of such models (like filters identifying lines and edges). There are two main approaches to transfer learning: fine-tuning an entire model for the new task, or freezing the weights in the model for use as a fixed feature extractor. Given our dataset size, we believed that backpropagating through the entire network was more likely to hamper our model's performance rather than provide any gain in accuracy, so we opted for the later approach. Next, we will describe the model chosen for this transfer task: GoogLeNet.

4.2. GoogLeNet

GoogLeNet, the winner of the 2014 ImageNet challenge, represents two main advances over the types of CNN models that came before it: it went deeper with its convolutional layers, allowing it to learn better hierarchical features, and it did away with the fully-connected layers that contained the majority of parameters in earlier models, enabling a greater level of computational efficiency [14]. Both of these features make GoogLeNet an ideal choice for our task, with the strong hierarchical features enabling better generalization to our problem setting and the computational efficiency allowing for rapid prototyping and iteration. The ability of GoogLeNet to go deeper than previous convolutional networks was facilitated by the development of a strong local network topology that could be stacked repeatedly, the Inception module. The Inception module was motivated by the fact that relevant features in an image often occupy receptive areas of differing sizes, so rather than define a single, fixed-size convolution at a given layer in the network, the Inception module computes convolutions of various filter sizes in parallel and then concatenates the output along the channel dimension. However, these parallel computations are expensive to compute, and stacking this operation would result in ever-expanding channel dimensions, so the Inception module also includes 1x1 "bottleneck" layers which preserve the spatial dimensions of the data while reducing the channel dimension. Other novel features of the GoogLeNet architecture are the average pooling layer at the end of the Inception layers that transforms a (H x W x C) input to a length-C vector for classification (enabling variable-sized input) and the auxiliary classification networks at various depths to promote gradient flow through the network.

4.3. Architecture

Our core idea is to take a pre-trained GoogLeNet model with a newly initialized fully-connected layer and concatenate its output to the textual descriptors of a property as a complete representation for regression purposes. The net-

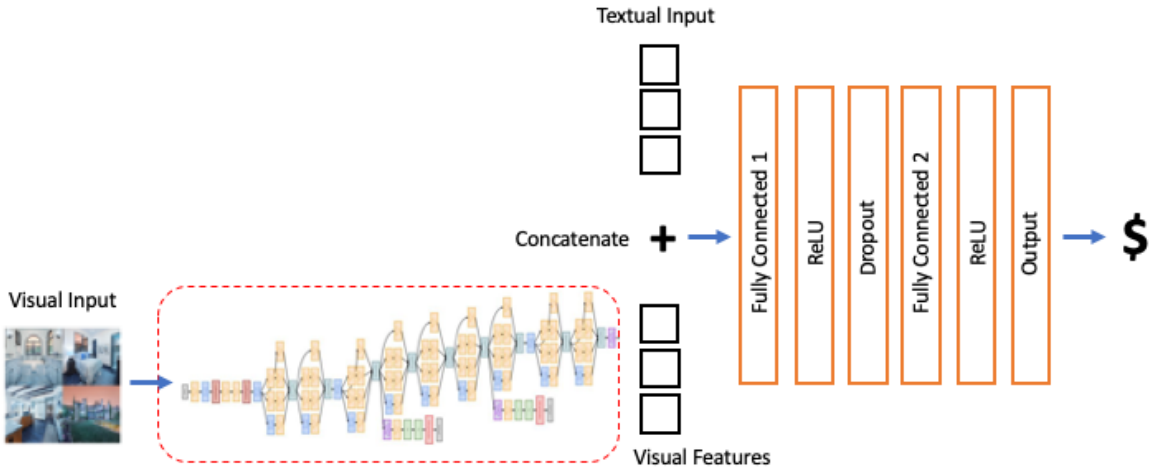

Figure 2. Our full architecture, with GoogLeNet - in red - serving as a fixed feature extractor over our visual input. GoogLeNet figure taken from CS231N course notes [9].

work that we have built on top of this concatenated feature consists of a dense layer followed by a ReLU activation and dropout, a second dense layer followed by a ReLU, and an output layer with a single output representing the final predicted price. To train the model, we only backpropagate our error - the mean absolute error between predicted and true price - through the fully-connected network and the final layer of the GoogLeNet model, retaining the weights in the convolutional layers. We believe this approach will be able to extract rich features from our visual data to improve the predicted prices over a textual-only approach. Our full architecture is displayed in Figure 2.

4.4. Baselines

Given that we are interested in exploring the potential gains in price prediction accuracy when augmenting textual descriptors of real estate with images of the property, a natural point of comparison is the price prediction accuracy that can be achieved using only textual data. As a baseline, we implemented a simple fully-connected neural model using Keras that takes as input our processed textual features and returns a price estimate. The model consists of three fully-connected layers, with hidden dimensions of size 64 and 32 and an output dimension of 1. We used ReLU activations between our hidden layers and followed the training regime described in section 5.5.

To explore the level of information about a property encoded in its visual depictions, we decided that another valuable point of comparison would be a simple convolutional network with exclusively visual features. Our model consists of three convolutional layers with filter sizes of 3x3 and total filter counts of 16, 64, and 128, respectively. We perform batch normalization and max pooling with a filter size of 2x2 after each convolutional layer. The output of the final max pool operation is flattened and passed through a series of fully-connected layers with ReLU activations, hidden dimensions of 8 and 4, and a final output of size 1.

4.5. Learning Algorithm

We trained all of our models using the same training procedure, which we will describe in this section. We used batch gradient descent as our optimization method, in which a loss value is calculated over a training batch and then backpropagated through the network to find the contribution of each trainable parameter to the calculated loss. The loss function we used to measure the error from our predicted outputs was mean absolute error (MAE) loss, as this was the common loss function we observed prediction models using in this problem space [10]. For a training batch of size N, the MAE loss takes the form:

$$MAE = \frac{1}{N}\sum_{1}^{N}|y_i - y_i^*|$$

where y_i is our predicted value and y_i^* is the ground truth. We then backpropagate this value to find the gradients with

respect to each of our parameters, and use Adam optimization to take a step in the negative direction of our gradient to (hopefully) reduce the loss over the next training batch. Adam is a per-parameter adaptive learning rate method that calculates a bias-corrected running average of the first and second moments of the gradient m_t and v_t and then updates the parameters w of our model using the formula:

$$w_{t+1} = w_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon}$$

where η is our learning rate, a tunable hyperparameter [6]. We trained all of our models for 25 epochs (complete passes through the training data).

5. Experiments

We will now describe our experiments with our baseline and multimodal models. Our training and evaluation code was adapted from the CS231N course assignments [9].

5.1. Saliency Map Visualization

Seeking to better understand the types of features the pretrained GoogLeNet model was attuned to detecting, we were interested in developing saliency maps for some of our images. Saliency maps are representations of the pixels in a given image that most influence the output of a convolutional model. However, because we were using the pretrained model as a fixed feature extractor in our multimodal architecture and thus had to disable gradient flow through that portion of the network as a result, we were not able to generate saliency maps on our final model. Because the weights in the convolution layers were frozen, though, it seemed to be a reasonable approach to generate saliency maps using only the pretrained GoogLeNet model, as it could still provide insight into the features that were identified throughout the Inception module layers. Furthermore, to better characterize the contributions of the individual room types in our tiled images, we generated saliency maps for component images to disentangle the activations across the collage. The process of generating a saliency map involves backpropagating the gradient of the model's output back onto the original input image. By taking the absolute value of those gradients, we are able to observe the pixels that have the greatest effect on the model's output. Note that we used the implementation from [5] to generate our saliency maps. In generating our saliency maps, we observed that the model was able to identify "important" regions of the image quite well, which is likely a side effect of the broad categories of objects that comprise the original ImageNet dataset. Figures 3 and 4 display two sets of original images and their corresponding saliency maps. Notice the saliency map corresponding to the image of the kitchen in Figure 3 finds most of the image important (the table, the oven, and the cabinets), but not the distant room on the top right of the image. This makes sense as an effect of the "restaurant" category in the original dataset, meaning that our model is primed to extract features representing relevant portions of our kitchen images. Notice also that the saliency map corresponding to the image of the bathroom in Figure 4 finds the counter top important. This is valuable behavior for our final valuation model, as features such as the bathroom countertop will likely have a greater influence on the price of the property over the bathroom floor or shower curtains. Taken holistically, our saliency maps confirm the applicability of transfer learning from an ImageNet-trained GoogLeNet model to our price estimation task.

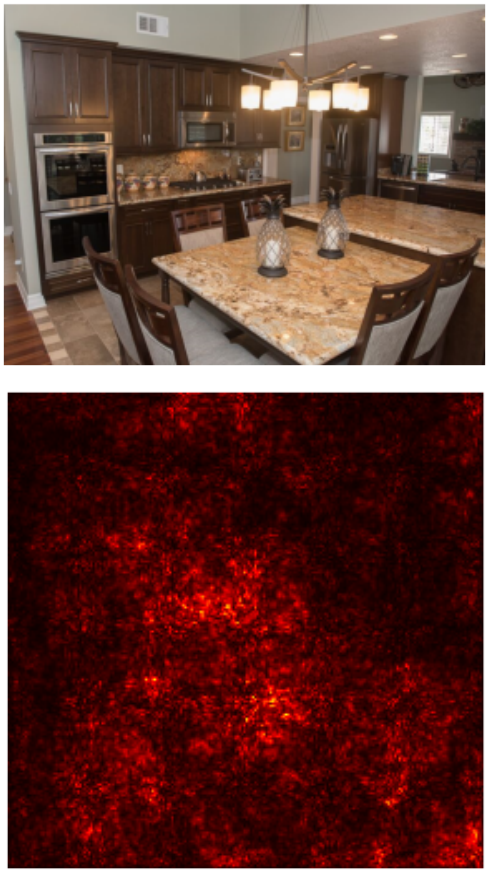

Figure 3. Image of kitchen and corresponding saliency map.

5.2. Hyperparameter Search

In order to discover the strongest possible model that could be constructed with the architecture described in section 5.3, we performed a broad search over the set of hyperparameters we believed to be most impactful to the performance of our multimodal model. The hyperparameters and their corresponding domains are as follows: size of feature output from GoogLeNet (16, 64, 256), hidden units in the first dense layer (32, 128, 512), hidden units in the

Figure 4. Image of bathroom countertop and corresponding saliency map.

second dense layer (16, 64, 256), dropout probability (0, 0.25, 0.5, 0.75), batch size (2, 4, 8, 16), and learning rate (0.0001, 0.001, 0.01, 0.1). We trained models with each of the 1,728 possible configurations of these hyperparameters for 25 epochs, and selected the best configuration based on the final validation error. The best configuration of hyperparameters was 16 visual features, 128 hidden units in dense layer 1, 16 units in dense layer 2, a dropout probability of 0.75, a batch size of 2, and a learning rate of 0.001.

5.3. Ablation Study

After training our final multimodal model with the hyperparameters discovered via the search process described above, we wanted to better quantify the impact of each of the modalities on the final price estimation error. We formulated this question as an ablation study: if we disable the part of the network corresponding to either vision or text, how is the final estimation impacted? We implemented this ablation study as a modification of our test accuracy functionality. We provided additional boolean flags to the function that would disable either the visual input or convolutional features by replacing them with a identically-sized zero vector before concatenation. This has the effect of disabling all the related neurons throughout the rest of the network and should provide insight into the contribution of each modality to the final price estimation. The outcomes of this study are included in our table of results.

Method	Mean Absolute Error
Baseline Text Model	$595,545.15
Baseline Visual Model	$595,543.84
Multimodal Model	$1635.45
Multimodal Model (Text-Only)	$3515.97
Multimodal Model (Vision-Only)	$1481.64

Table 1. Results from our experiments, given as the mean absolute error in price prediction on our test set.

5.4. Results

The results from the experiments with our baseline textual and visual models, multimodal model, and ablation study are given in the form of the average price prediction error on our test set in Table 1.

5.5. Discussion

Our results demonstrate a remarkable performance gain when fusing the two modalities in our dataset. Clearly the features that we analyzed via our saliency maps were a powerful asset when applied to the valuation problem. Based on the similar performance metrics seen across the two baseline models, we hypothesize that learning this task from scratch with such a small dataset is not a tractable problem, leaving both models in similar local optima and justifying our application of transfer learning to price estimation.

One discovery we made when training our models was the importance of early stopping. We originally trained all our models with 100 epochs, but realized that was far too many passes through a dataset as small as ours, and led our networks to bad local optima. For example, after 100 epochs, our multimodal model would output a constant result regardless of input. By reducing our training epochs to 25, we were able to avoid overfitting on our training set and retained diverse outputs across our test set.

Our ablation study results are interesting, but ultimately inconclusive due to the small size of our dataset. The results do seem to support the hypothesis we formed based on our baseline results that the problem is a difficult one to learn from scratch with our dataset, with the text-only network showing highly degraded performance. It is difficult to make a strong conclusion about the vision-only ablation study. Given that it showed marginally increased performance over the multimodal model, it could be an indication that the GoogLeNet features are doing much of the heavy lifting, but it could also simply be an artifact of a particular

test/train split over a small dataset. A more useful conclusion can be drawn when comparing the vision-only ablation study with the text-only study and seems to support our belief that the visual features of a real estate property are one of the key signals when appraising its value.

6. Conclusion and Future Work

Ultimately, the proposed multimodal model yielded better performance than our baseline text and visual models, as well as our text only multimodal model. It did not perform as well as our multimodal model with only visual data passed into it. We cannot extrapolate as strong of a conclusion as we might have hoped given the size of our dataset, however the multimodal approach seems to be a promising direction for this problem space. The most valuable potential avenue for the future of this work would be the construction of a larger dataset, which could greatly reduce the stochasticity in the training of our networks, allowing us to draw stronger, more general conclusions. A larger dataset would also enable the training of a deep convolutional network from scratch, which would allow the model to learn features more specific to this task, and enable us to generate saliency maps over the multimodal model to better understand features of a house that strongly correlate with property price. These observations could perhaps lead to understanding deeper questions in the industry, such as whether a property is under distress to sell. Clearly, the potential impacts of this work encourage future explorations of this problem space, which would be greatly enabled by a more data-rich regime.

7. Contributions

In pursuit of this final report the contributions of each author are as follows:

Sam Lowe: Multimodal model design, hyperparameter search, ablation study, paper writing

Svetak Sundhar: Baseline model design, data preprocessing, saliency maps, paper writing.

We sincerely thank Fei-Fei Li, Ranjay Krishna, Danfei Xu, and all the CS231N course staff for exposing us to many of these ideas and for a great quarter!

References

[1] Vincenza Chiarazzo, Leonardo Caggiani, Mario Marinelli, and Michele Ottomanelli. A neural network based model for real estate price estimation considering environmental quality of property location. *Transportation Research Procedia*, 3:810–817, 2014. 17th Meeting of the EURO Working Group on Transportation, EWGT2014, 2-4 July 2014, Sevilla, Spain. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[3] Samaa Elnagar and Manoj A Thomas. Real estate image-based appraisal using mask region based convolutional networks. 2019. 2

[4] Maryam Heidari and Setareh Rafatirad. Semantic convolutional neural network model for safe business investment by using bert. In *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 1–6, 2020. 2

[5] Irfan Khalid. Saliency map for visualizing deep learning model using pytorch. 5

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 5

[7] Stephen Law, Brooks Paige, and Chris Russell. House price estimation from visual and textual features. *International Joint Conference on Computational Intelligence*, 2016. 2, 3

[8] Stephen Law, Brooks Paige, and Chris Russell. Take a look around: Using street view and satellite images to estimate house prices. *ACM Transactions on Intelligent Systems and Technology*, 10, 2019. 2

[9] Fei-Fei Li, Ranjay Krishna, and Danfei Xu. Cs231n course website and notes, 2021. 4, 5

[10] Li Li and Kai-Hsuan Chu. Prediction of real estate price variation based on economic parameters. In *2017 International Conference on Applied System Innovation (ICASI)*, pages 87–90, 2017. 2, 4

[11] Jian-Guo Liu, Xiao-Li Zhang, and Wei-Ping Wu. Application of fuzzy neural network for real estate prediction. In Jun Wang, Zhang Yi, Jacek M. Zurada, Bao-Liang Lu, and Hujun Yin, editors, *Advances in Neural Networks - ISNN 2006*, pages 1187–1191, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 2

[12] Omid Poursaeed, Tomas Matera, and Serge Belongie. Vision-based real estate price estimation. *Machine Vision and Applications*, 29(4):667–676, 2018. 2

[13] Yan Sun. Real estate evaluation model based on genetic algorithm optimized neural network. *Data Science Journal*, 18(1), 2019. 2

[14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. 3

[15] Quanzeng You, Ran Pang, Liangliang Cao, and Jiebo Luo. Image-based appraisal of real estate properties. *IEEE Transactions on Multimedia*, 19(12):2751–2759, 2017. 2