# Visual Score Generation and Comparison Metrics for Advertising

Kun Qian
UC Davis
kunqian@ucdavis.edu

Yu Li
UC Davis
yooli@ucdavis.edu

Xincheng Lei
UC Davis
xclei@ucdavis.edu

Kevin Jesse
UC Davis
krjesse@ucdavis.edu

## Abstract

*Pictures of goods and services are ubiquitous in various areas of commerce from homes on AirBnB, cars on AutoTrader, and homes on Zillow. Similarly, all of these commerce sites have items for sale that are priced according to various qualities. Estimating an item's worth and public perception a priori give vendors an opportunity to enhance the advertisement thus increasing revenue. Determining this value requires a comprehensive understanding of multiple images and taking consideration of various feature weighting i.e are the bedrooms of the apartment more important than the front yard? We propose an architecture that can effectively evaluate these images of an apartment or home and provide accurate estimations of the nightly rate. We validate our model on the AirBnB dataset. The AirBnB dataset is composed of various home styles, images, features, and nightly prices. With our results, we hope that companies like AirBnB will follow a similar paradigm for instant feedback on the products they are about to post, make adjustments according to our model results, and improve their advertising and customer reach.*

## 1. Introduction

More than 600 billion dollars are spent on advertising just this year and companies like WPP and Omnicom Group charge premiums to generate reports on product advertisement performance. Often, these product advertisement reports lack detailed evaluations on the images presented to the customer. Moreover, in an environments where users do not have access to such expensive services, such as homeowners, there is no service that helps guide the user to improving how their product or home is perceived to the customers. If customers of platforms like AirBnB have access to metrics that gauge how their product is received, they can iterate on their advertisement until it is just right. Intuitively with price estimation, customers can better understand what features, lighting, poses, etc are contributing to the negative scoring. Furthermore customers can see where there home ranks with similar homes and determine if their advertisement is properly executed and meeting market expectations.

**Figure 1: This AirBnB home** is detected as a high quality home but is available for a lower mid-tier price.

Airbnb is a online marketplace for short term rentals, accessible via its websites and mobile apps. Members can use the service to arrange or offer lodging, primarily homestays, or tourism experiences. The website contains more than 4 million properties, each with detailed description and aorund 20 images, located in 191+ countries and districts. We propose to analyze Airbnb's dataset considering the quantity of images on their website is large enough, which is required for training model. Moreover, the their images are all publicly available and with same format and size. We designed a model that captures price correlated features in both foreground and background such as quality of the furniture and paintings on the walls. The model effectively capture stylistic characteristics that make a home desirable and this is validated in our price estimation metric. Our model is validated on our new AirBnB dataset which is composed of various images of homes with their associated features such as nightly pricing, reviews, etc. We find this dataset to simulate realistic image computation challenges for hugely popular platforms with high traffic image data. Our deep learning architecture is based off several well known networks, like ResNet, with small modifica-

tions to enhance our performance on the aforementioned dataset. These architectures are highly adaptable to various domains other than home rentals like AutoTrader and Amazon.

We will make our network architecture and dataset available and public.

## 2. Related Work

Price estimation has been broadly applied in various fields. To our knowledge there has been little published work on advertisement feedback leveraging computer vision. Computer vision has been used to predict the quality of items in other markets like the quality of meat [2]. Matthias Zeppelzauer et al[11] propose a method to predict the building age of the houses by HOG and SIFT. Previous automatic advertisement evaluation was done on user interaction with ads from a user experience perspective [10]. House price estimation with neural models has been done with Google Map input parameters such as local amenities, real estate evaluations school location, etc [9]. However, these features have little value to other domains like the rental market and lack the true understanding of the predicted user experience. Our model captures this information along without leveraging additional boosting information such as the map and user posting.

AirBnB posting predictions has been done before by Junyao Wang et al[3] by using regression models to predict the price according to location and reviews. History price was also used to predict the average house price in an specific area[1]. It is also reasonable to predict the price by rating the reviews on the AirBnB[5]. These quantitative features might represent a fraction of the value but do not provide any insight to the customer for advertisement improvement. Our work evaluates the home as a whole by the uploaded pictures and evaluates the value by commonly desired features.

### 2.1. ResNet

Since neural network has been widely and deeply explored in the recent years, the importance of the number of hidden layers becomes widely recognized and has been confirmed by experiments in various fields. In the domain of computer vision, After the success of AlexNet[4](8 layers), more and more deeper network frameworks, like VGG[6](19 layers) and GoogleNet[8](22 layers) has been proposed as well. However, with increasing the number of hidden layers, the training difficulty as well as the degradation problem, which indicates that with the network depth increasing, accuracy gets saturated and then degrades rapidly, has been exposed.

The deep residual learning framework, ResNet[1] has been proposed to elegantly deal with that problem, which explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. According to the Fig.2, we can find that, with ResNet, rather than hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. The deepest ResNet constructs more than 1000 layers while the 55-layers and 152-layers ResNet are more welcome to apply considering computation efficiency against task requirement. In this paper, we also explores the 50-layers and 152-layers ResNet.

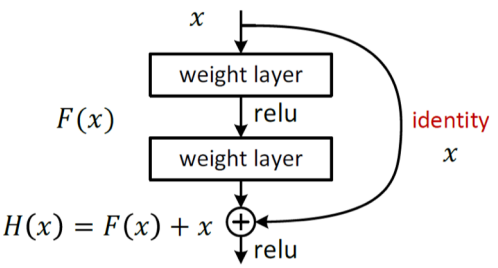


**Figure 2:** Residual learning: a building block [1]

## 3. AirBnB Dataset

AirBnB is a popular online marketplace and hospitality service which members of the service can arrange lodging, homestays, or tourism experiences. Members submit listings of their homes with pictures and price their home to their choice. Airbnb members often over and under estimate their homes rental worth. When customers see comparable homes for less, members with overestimated values often have low rental frequencies. Similarly, potential earnings are lost when the homeowner under estimates the homes worth. By identifying miss estimations 1, we are able to provide evaluation insights prior to the homes listing and notifying the homeowner that the listing should have a higher price. Once a listing is posted, we can use the prior insights to notify customers of good deals.

### 3.1. Dataset Information

We gather listing information from insideairbnb[1] which provides us with raw airbnb listing data. This data includes the listing ID, descriptions, number of past rentals, summaries, location, price, and more. This data includes host specific information and interesting description techniques that we hope to use in a multimodal future evaluation. Most importantly we extract the price and the listing url. This url is used to retrieve the listing photos posted on the AirBnB webpage.

[1] http://insideairbnb.com

We centralized our data collection to the densest city of AirBnB rentals available, New York City. We filtered our criteria to entire homes which is 50.2% of the New York City market. The remaining 49.8% are either private or shared bedrooms. Due the the inconsistency of the latter we chose only entire home listings. Moreover we filtered our listings to single bedroom flats with similar amenities. We excluded exclusive low availability homes because that will factor into the value and not be captured in the image data. This left us with about half of the 50k listings.

We noticed that many users upload miscellaneous photos for their listings: things to do, pictures of nearby attractions, etc. By restricting the listings available photos between 5-11 we reduce the probability that the photos will have miscellaneous data. If a homeowner posts 5 photos, it is highly unlikely that a large portion of them are unrelated. While there exists a possibility of miscellaneous photos, we have not encountered. Moreover, the variable number of photos in each listing is an interesting twist to existing computer vision datasets, but mimics real life requirements platforms such as AirBnB face.

### 3.2. Data Collection

We built and published a data collection tool[2] to extract AirBnB listings. The listing photos are extracted by simulating real user traffic with selenium[3] webdriver and chromium[4]. Simulated users go to the listing, click through the photos, and extract photo hosting urls. The hosted images are then downloaded from these extracted urls. The code is fully parallel and the tool is robust for large data sets. Our dataset included 1,135 homes and 9637 photos. Typical data collection tools would take up to a week, but ours runs in just a few hours for a single data dump (30k-50k homes). We ran our collection on a Ubuntu 16.04 LTS machine with a 48x thread Xeon CPU @2.2 GHz. The thread heavy CPU provided us with 48 unique simulated users capable of traversing the data collection algorithm.

### 3.3. Dataset Analysis

In order to classify homes into low, mid, and high tier by the higher level features detected in our deep architectures, we had to find reasonable price points that split our data properly. We found three logical class divisions at $0-$125, $125-$175, and $175+ nightly rental rate. The distribution of the dataset can be observed in Figure 4.

[2]https://github.com/kevinjesse/AirBnB
[3]https://www.seleniumhq.org
[4]https://www.chromium.org

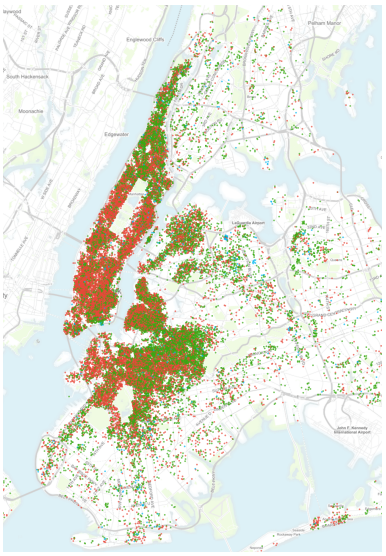

**Figure 3: New York City Listing Densities** capture the full homes (red) and single/shared bedrooms (green) available.

| Number of Homes Split | | | |
|---|---|---|---|
| Class | Train | Test | Total |
| Low Tier (Class 1) | 182 | 35 | 217 |
| Mid Tier (Class 2) | 485 | 111 | 596 |
| High Tier (Class 3) | 268 | 54 | 322 |
| All Classes | 935 | 200 | 1135 |

**Table 1:** The distribution of homes amongst the training/test splits according to the three defined classes.

| Number of Photos Split | | | |
|---|---|---|---|
| Class | Train | Test | Total |
| Low Tier (Class 1) | 1460 | 388 | 1848 |
| Mid Tier (Class 2) | 4027 | 1012 | 5039 |
| High Tier (Class 3) | 2233 | 517 | 2750 |
| All Classes | 7720 | 1917 | 9637 |

**Table 2:** The distribution of photos amongst the training/test splits according to the three defined classes.

The dataset can be loaded seamlessly with the Torch dataloader used to load imagenet and CIFAR datasets. This makes the dataset easy to run and all requirements for running the architecture on the dataset can be found online[5].

### 3.4. Multiple-input Images

To explain the target variations, we may have auxiliary inputs to make the network receive more features and data about the main input. Multiple-input Multiple-output is an efficient way to make the network more precise and fine-tuned. However, we need more loss functions to supervise

[5]https://github.com/kevinjesse/airbnb_deep

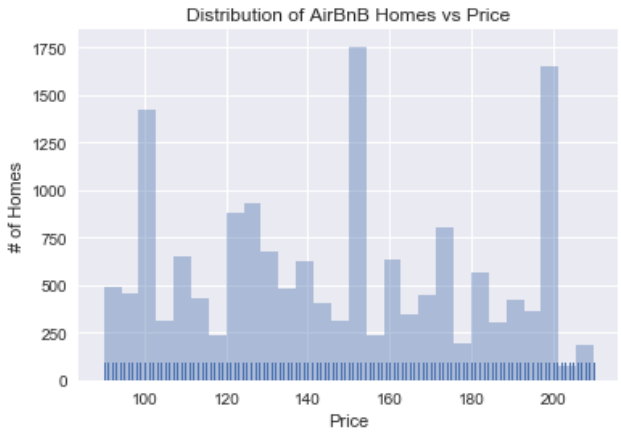


**Figure 4: Distribution of the AirBnB DataSet vs Price** We chose low, mid, and high tier class boundaries by the distribution of the dataset. Notice the influx of prices around simple numbers of $100, $150, and $200. Notice the three clear classes around these prices. Our boundaries are determined by the distribution of data around $125 and $175.

the networks. Auxiliary loss function usually is used earlier in the model to accelerate the convergence process. It is also widely applied in computer vision. Yu Sun et al proposed a multi-input method for large scale flower grading[7]. They use three images of one same flower from a different angle such that the model knows more information about the object. Another good way to catch more input features is to concatenate the input images on the channel and train them in a single-input model. We only have one loss function and it is more like basic convolutional neural networks. We can choose different architecture based on the input layout and complexity we want. We use input images concatenation and one cross entropy loss function in this paper.

## 3.5. Data Normalization

To normalize the number of photos, we can set a fixed number of photos as the input of our networks. One way to get a reasonable result is setting one image as an input and get a classification or regression for each image. Observe below the many ways we can model the variable image input.

### 3.5.1 Mean Classification

We can get an average rating based on all the photos that a house has and we can set different weights for different kinds of images. For example, we may place more considerable weight for an image which contains more higher level features in the house and set a smaller weight lower level features. We evaluate this proposal and found that individual loss evaluation on single photo classification introduces noise.

**Figure 5: Difficult Example to Discriminate** This home is high quality, but some images have a low scores associated with them. The majority of these pictures were classified correctly. We choose to use maximum voting schema for the cross entropy loss.

### 3.5.2 Observations in Classifications

In the dataset, we have some pictures that only contains a drawing on the wall or just a plant at the corner of the house, which will add noise into the rating result. As shown in Figure.4, we can see the decoration of the house is excellent and expect the price to be high, and other images only contain a sofa and the pillow on the bed, which should decrease the score of the house. For example, shown in Figure.4, we can conclude from all the photos that this house is high quality because there is a good rug, comfortable lights and the furniture looks expensive. However, the image of the refrigerator and the image of the sink would not get too high score since there are very few subjects in a single image.
We anticipate, room and pose normalization across a fixed set of photos should greatly improve the regression and classification, but we declined this initial approach because it is not representative of similar platforms. Moreover we figure that low level information might play a bigger role than higher level features because most kitchens, bathrooms, and bedrooms have similar features. This is not the case as we will show in a later section.

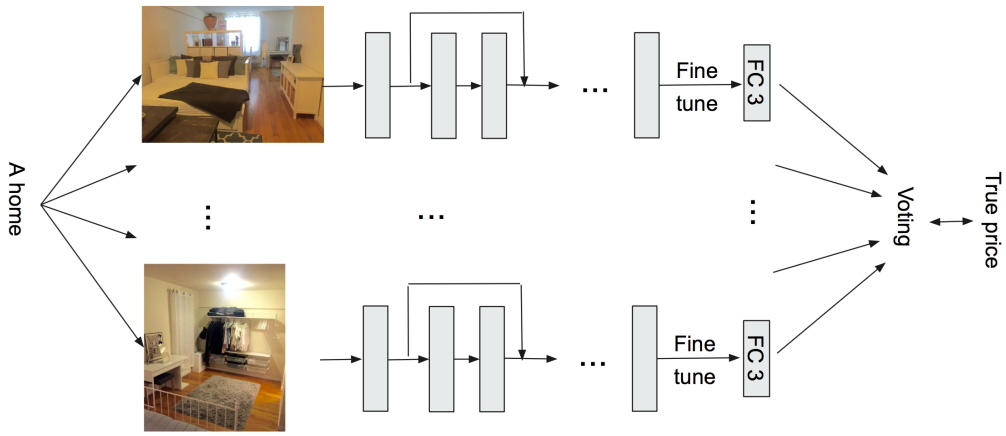


**Figure 6: Neural architecture network using pretrained ResNet152 with a majority vote cross entropy loss.** The network is fine tuned after the voting has occured across all batches. Standard residual backpropagation on the batch is performed with the majority voting cross-entrophy loss.

# 4. Experiment

## 4.1. features from different layers

In CNN, the higher level layers have more global meaning of the images such as the objects detected and the themes of the image. On the other hand, lower level layers capture the local features such as texture, brightness and resolution. Which one is more related to the price of the home? We have done an experiment on this question.

In a pre-trained VGG-16 network for imagenet, the last layer in the second and fourth blocks are extracted and the feature maps are used as an input for another predictive model for the price. The outputs on these two layers are 56*56*128 and 14*14*512. The size of feature maps are too large for our dataset ( 1000 homes). So we use the max pool in each channel reducing it to vectors with size 128 and 512. They are feeded into multiple models including K nearest neighbors, support vector machine, logistic regression and gradient boosting decision tree. We get the best performance of predicting home price from K nearest neighbor. With the similar spirit to majority voting, each image will have a class labels. And the label for the home is the voting of images belonging to it.

The performance of predictions based on $block2\_conv2$ and $block4\_conv3$ are in figure 7. It reflects that the images of the same class does not have simple similarity on the feature maps either in low or high levels. The support vector machine and gradient boosting decision tree are also applied to predict the prices. However, since we have many features versus only 1000 prices points, the overfiting is too strong to get even worse results. An average inside the channel is also tested and also leads to worse result.

## 4.2. Feature Extraction

We conducted our experiments based on a pre-trained neural network, ResNet with 152 hidden layers, trained with ImageNet dataset for feature extraction and tuned the model with Airbnb's dataset

## 4.3. Feature Extraction

We conducted our experiments based on a pre-trained neural network, ResNet with 152 hidden layers, trained with ImageNet dataset for feature extraction and tuned the model with Airbnb's dataset

# 5. Improvements To Model

## 5.1. Preprocessing

Except for image normalization we mentioned before, there are still some preprocessing for cleaning up dataset we could conduct.

**Image Classification** In the dataset, all the images including building-related images or unrelated images(e.g. nearby tourist interests), outdoor view or indoor view, overview of a room or close-up for certain objects are all mixed up. When we did the image normalization, we did not know what kind of image we choose since there is no order for different kind of images. In that way, the comparison of different rooms is not consistent. It is a reasonable method to deal with the problem that training a classifier to label all the image with taggers like indoor, outdoor, objects etc. so that we could get specific feature extractor for different kind of images

**Advanced Image normalization**Since images from different house came from different owners, the photoing condition(resolution, brightness, contract etc.) can be various.

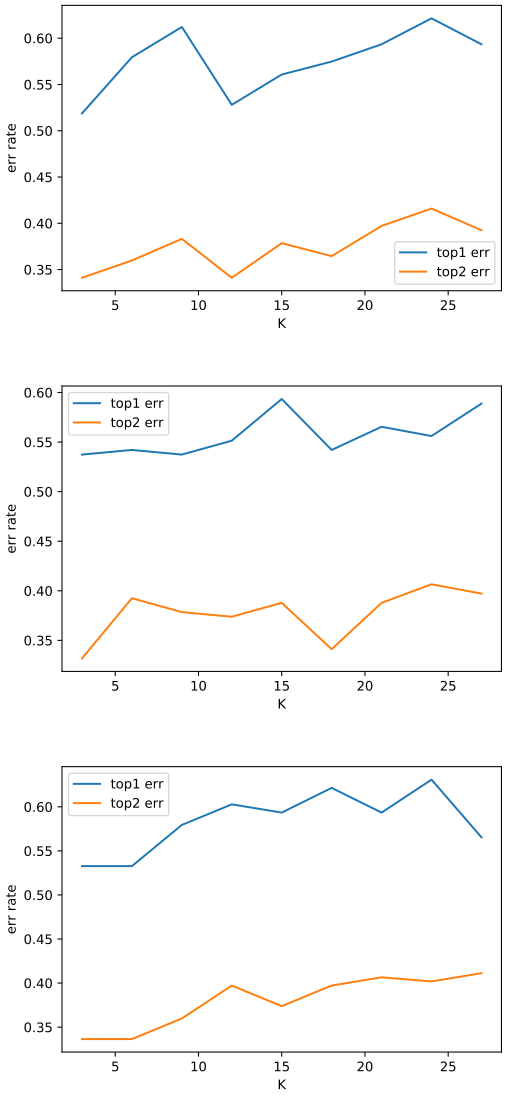


**Figure 7: Performance based on feature maps extracted from different layers of the VGG-16 network** top1 and top2 error rate on the prediction based on the max pool from the feature maps in the layer $block2\_conv2$ (top) and $block4\_conv3$ (medium). Also, we tried average inside each channel in the layer $block1\_conv2$ (bottom). We applied a K nearest neighbor on the feature maps to predict the price classes. The performance is as bad as random prediction.

Though the network might actually ignore such features, but normalizing them would more or less lead to some improvements.

**Image Concatenation** The input in our framework is a group of images which is not acceptable by regular neural network. RNN is a choice for group of input, but that is especially efficient for ordered sequences. Since there is

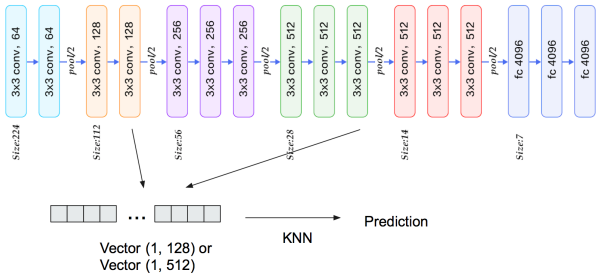


**Figure 8: VGG-16 with low level feature extraction.** Abalation study to see if low level features contributed to classification more than the higher level features i.e. quality of photo, lighting, texture. Results of the this study showed that higher level features performed better.

no order among images, RNN seems not necessary. One option is to concatenate images along channel dimension. This method can not only deal with the input dimension, but also make it possible to extract features from the combination of multiple images.

## 5.2. Output Setup

We use 885 houses to train our model, which includes 166 low-tier price houses, 476 mid-tier price houses, and 243 high-tier price houses. For the test process, we use 250 houses to test our model, which includes 51 low-tier price houses, 120 mid-tier houses, and 79 high-tier houses. However, when we tested our model, we got a higher percentage of the mid-tier houses than the input from the dataset. The proportion of the mid-tier is far more than the other classes. We think the reason is that we only set 3 classes for our model and the classification is only based on a continuous score predicted by the image, which causes some houses of the low-tier and high-tier are wrongly classified into the mid-tier. Specifically, wrongly classified houses in the mid-tier could be from the low-tier and the high-tier, while only houses in the mid-tier could be classified wrongly into the low-tier. For that reason, the percentage in the mid-tier is higher than the other classes.

To solve that problem, we could set more classes of prices, and it will reduce the false-classified percentage of the mid-tier price since each class in the middle will get prediction wrongly from both side of it. The other way is to add more features as multi-input in the model and estimate the price not only based on the picture of the house but also the review or the surroundings. In this way, we can decrease the weight of a linear rate and avoid such distribution.

## 5.3. Attention Mechanism

As mentioned in data normalization part, people have different interest in different kind of features of a room. Different features also have different impacts on rental price. Based

on the images type classification discussed above, assigning weight to each image would emphasize the different interest for different kind of images. Then an extra attention layer might help for find an optimum assignment strategy and distributing corresponding weights.

### 5.4. Dataset Expanding

We collected the data on October 2018, and it is feasible to expand our dataset to a broader timeline, for example, we can train our model using the price of houses on AirBnB of a whole year. This can help us normalize the dataset because the price will change for the holidays and some city policies. If we can work a dataset that is from a complete year, we can lower the noise from different times and periods. Our crawler is still working, and we may try to train the model using data from the different time of the year in the future, and we can even get some more exciting results.

### 5.5. Combination with Other Information

In our framework, we predicted the price barely based on the images that owners provided. However, empirically, the price is depending on complex factors such as location, check-in time, surrounding facilities, hidden devices or services which are not able to extract based only on images. That is also why the Airbnb requires more detailed illustration about the room other than images. We used barely images since we focused on exploring new technique for computer visions, but in actual price prediction, it is obviously benefit to concatenate such factors with image features.

## 6. Conclusion

In this paper, we introduce a new computer vision dataset of AirBnB homes. Moreover, we contribute a data mining tool that can be adjusted for various existing media platforms i.e hotels.com, tripadvisor, turo, etc. We perform a complete evaluation of various architectures with various modifications to account for our variable sized image portfolio
With limited data, we applied pre-trained neural network ResNet and VGG fitted for the imagenet dataset. For the ResNet we did a fine tuning and train a new layer for the classification task. For VGG we have done experiments on feature extractions from different layers and feeded into different predictive models. The performance can be boosted with larger dataset and better pre-trained networks specifically for aesthetic classification tasks. After all, with enough data and computation resources, an end-to-end training can be easily done since the prices are free annotations from the website.

## References


[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] P. Jackman, D.-W. Sun, and P. Allen. Recent advances in the use of computer vision technology in the quality assessment of fresh meats. *Trends in Food Science Technology*, 22(4):185 – 197, 2011.

[3] C. X. Junyao Wang, Hsin Lu. Predicting listing price on Airbnb dataset. "`https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a052.pdf`", 2017.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[5] Y. Li, Q. Pan, T. Yang, and L. Guo. Reasonable price recommendation on airbnb using multi-scale clustering. In *2016 35th Chinese Control Conference (CCC)*, pages 7038–7041, July 2016.

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] Y. Sun, L. Zhu, G. Wang, and F. Zhao. Multi-input convolutional neural network for flower grading. *J. Electrical and Computer Engineering*, 2017:9240407:1–9240407:8, 2017.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[9] A. Varma, A. Sarma, S. Doshi, and R. Nair. House price prediction using machine learning and neural networks. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1936–1939, April 2018.

[10] B. Wang, J. Wang, L. Duan, Q. Tian, H. Lu, and W. Gao. Interactive web video advertising with context analysis and search. In *2010 20th International Conference on Pattern Recognition*, pages 3252–3255, Aug 2010.

[11] M. Zeppelzauer, M. Despotovic, M. Sakeena, D. Koch, and M. Döller. Automatic prediction of building age from photographs. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ICMR '18, pages 126–134, New York, NY, USA, 2018. ACM.