Multimedia Networks for House Price Prediction

Colin Gaffney
Stanford University
cgaffney@stanford.edu

Juan Carlos Sarmiento
Stanford University
jcs10@stanford.edu

Abstract

Image-based classification using convolutional neural networks (CNN's) has seen considerable success in the past few years. This paper presents a model for valuing houses using a combination of images, text descriptions, and economic data, which are gathered from government websites and Zillow, the popular online real-estate market. Using three separate inputs the model combines a CNN based on ResNet, a recurrent neural network (RNN), and a fully-connected network to predict the prices of houses, which is formulated as a classification problem by discretizing the prices into buckets.

Our results include a baseline model and our final implementation of the Multimedia Model discussed above and trained on various subsets of the inputs. Our experiments test different loss functions, individualized class weightings, and bucketing distributions. Our quantitative and qualitative results indicate that the full model achieves significantly improved results over the baseline and independent components.

1. Introduction

Online markets have been booming in recent years. Companies like Amazon, Airbnb, and Zillow are now offering platforms for supply and demand aggregation for each of their sectors. Real estate transactions are central for the economy and having a precise pricing tool will give buyers and sellers more transparency. Some of the applications of the project we have in mind are identifying possible investments, a spam checker for the website, or as a pricing tool for real estate agents.

Although most traditional approaches to housing price prediction use exclusively economic data, we seek to build a model that trains on the same input as a human user of Zillow sees. We will explore how the pictures and text-based house descriptions in online listings, in addition to their economic features, are predictive of their prices. Specifically, the input to our model is a house's image, text description, and associated economic data, which we combine with a CNN, RNN, and fully-connected network to classify the house into one of M price buckets.

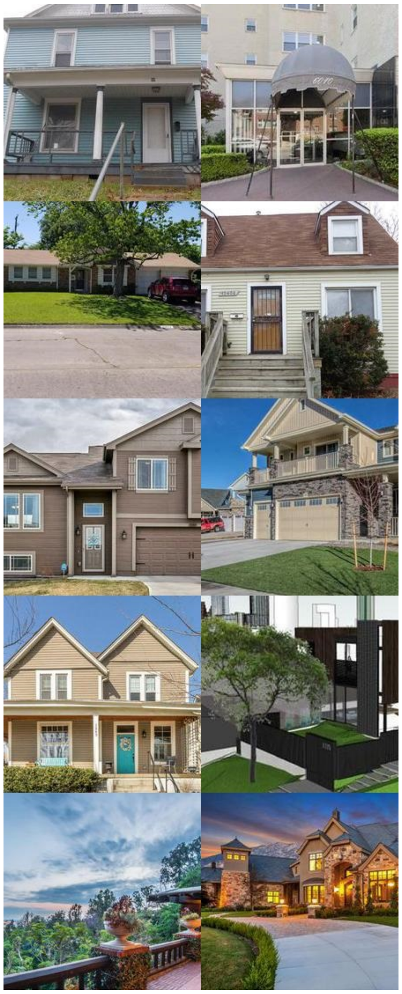

Figure 1. Sample images from the Zillow dataset. The images are displayed such that the prices of the houses increase in the downward direction.

Compared to existing works, we will be working with a dataset that is an order of magnitude larger and more representative of houses across the entire United States which is drawn from Zillow. Furthermore, while most existing works focus on a specific medium of data as input (typically economic data), our model incorporates contributions from diverse data media, including images, economic data, and natural language. We also believe that working with data from a commonly used website like Zillow allows our model to have broader applicability to real-world prediction tasks.

1.1. Related Work

House price prediction is a classic task that has been studied by many researchers, primarily on a relatively small regional level (1) (2). Most previous work on the subject of predicting house prices uses only economic features to predict house prices. Such approaches most frequently use linear models for regression rather than classification. For example, C. H. Nagaraja et al. (3) present a classic autoregressive model for house price prediction. One recent paper (4) saw its best results using such exotic algorithms as RIPPER, an optimized propositional rule learner. However, more recent approaches have used neural models. One such model that also incorporates images is presented by (5). Their approach is as follows: first, they preprocess the images by running them through GoogLeNet and storing the output of the last layer. Then, they feed that data into an RNN in an order defined by random walks of houses based on proximity. This approach more explicitly incorporates relevant pricing information from nearby houses.

As we will describe below, our model incorporates three independent components which use different neural architectures. Our primary component is a convolutional neural network (CNN), which has seen considerable success in visual recognition tasks. Past research in computer vision has seen perhaps its greatest success in classification tasks, which prompted our decision to formulate our problem similarly. Successful architectures are traditionally evaluated on the ImageNet (6). Two of the first deep convolutional architectures successfully applied to the ImageNet task were ”AlexNet” (7) and ”VGG” (8). However these models were soon outperformed by significantly deeper and more efficient networks, such as ”ResNet,” (9) ”GoogLeNet,” (10) ”Xception,” (11) and others (12). In addition to their depth, these models provide a number of key innovations, including residual connections, which combat the vanishing gradient problem for deep networks, and ”inception modules,” which improve the efficiency of very deep CNN's. The above models with weight matrices pre-trained on the ImageNet task can be applied to other tasks with less data, such as our own.

Since our Multimedia Model also incorporates textual data, we also consider recent advancements in the field of language modeling, which primarily relies on the family of recurrent neural networks (RNN's). Significant attention has been devoted to developing dense representations for one-hot word vectors (13) (14). These word vector representations enable more efficient training and transfer learning, which we make use of in this project.

1.2. Data

The data we worked with consists of 3 types of inputs gathered from Zillows website (15) and governmental entities, including the IRS (16) and Federal Housing Finance Agency (FHFA) (17). We incorporate more than 50,000 houses from the 100 most populous cities in the US. This data split into a training/validation set of 45,000 observations and a test set of 5,000 observations. As shown in Fig. 1, the images are highly diverse; some are actual houses, others are computer generated, and some are just landscapes of the houses' views. A small, but not insignificant proportion of the dataset contains interior shots of houses. From Zillow's website we also gather the text descriptions of houses that were written by the user posting the house for sale.

Additionally, we gather macroeconomic, ZIP-code level data from government websites, including the Bureau of Labor Statistics and the US Census. In particular, this data includes tax information and housing price indexes. To summarize, our model receives pictures from the houses, the listing's description (natural language), and economic information on a ZIP code level.

2. Methods

2.1. Data Preprocessing

For each epoch of training, we augment the image data by introducing a centered random crop of the image as well as a horizontal flip and a rotation of up to 20 degrees. In addition to providing an augmented dataset, this also allows us to standardize image sizes. While most images in the dataset are 300x400, some are larger or smaller.

The houses were bucketed according to their price for the classification task. We believe bucketing to formulate our problem as a classification one makes sense because pricing houses is a subjective task and giving a price range (buckets) rather than a single value (regression) is more realistic. At first, we examined a bucketing scheme which spaced the bucket endpoints linearly; however this resulted in an ill-shaped distribution as is shown in Fig. 2. This is a result of the fact that most houses are clustered around a low, reasonable price, but the distribution of prices has a long tail for more expensive and rarer houses, which may range up to hundreds of millions of dollars in value. We take two steps to combat this long-tailed distribution.

First, we cap our second-largest bucket at a price of $10,000,000, and any houses with a higher value fall into the last bucket. Second, we distribute the bucket fenceposts

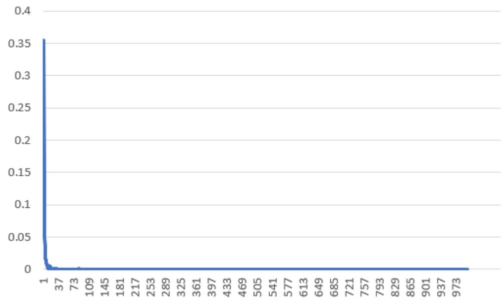

Figure 2. Distribution of training observations using evenly spaced price buckets.

in a geometric fashion, which result in a more normal distribution as shown in Fig. 3. We theorized that these changes would allow for more useful predictions (since all house prices do not fall into the same bucket) and easier training progress (since there are fewer classes with only a few training examples), which was later found to be true.

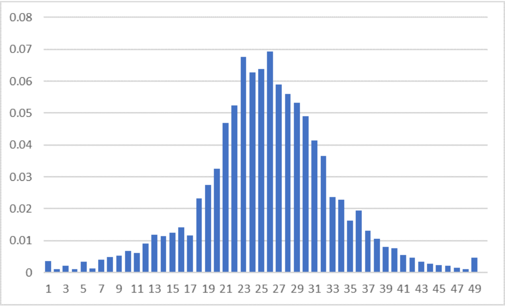

Figure 3. Distribution of training observations under geometrically-spaced price buckets.

2.2. Evaluation Metrics

We judge our model primarily on top-k accuracy, which counts a prediction as correct if one of the top k choices of the model is the correct class. Specifically, we use top-5 accuracy. We consider this metric to be a more realistic test of the model's predictive power since housing valuations may be somewhat subjective and may vary arbitrarily in some cases.

Concerning qualitative evaluations, we perform a qualitative error analysis on certain observations. For example, if the predicted class was very far from the actual class, we examine what features of the image misled the classifier and tried to adjust the model to reduce them in the future. We also employ saliency maps of selected images to examine what regions of the images matter to the model. We will elaborate on these evaluations in the Experiments section of the paper.

2.3. Baseline

The baseline approach to a house price classification problem uses two features, beds and baths, to predict the correct price class for a given example. For this task, we employ a multinomial logistic regression model. This baseline is intended to be used as a reference point for the more complex models presented below.

Data	Accuracy
Training	9.4%
Validation	9.1%
Test	8.9%

Table 1. Results for the baseline model

The results for the baseline model are displayed in Table 1. As expected, the simple logistic regression model performs very poorly, but outperforms a random prediction, which gives an accuracy of about 7% for 50 buckets and a geometric bucketing distribution. Obviously, this model's data is significantly limited, since it is confined to only two key features. Furthermore the model itself lacks the expressive power to make more accurate predictions. However, designing a baseline gives us a suitable reference point with which to compare our more advanced neural model.

2.4. Multimedia Model

Our primary model architecture consists of a neural network that consists of three separate modules: (1) a CNN for image inputs, (2) an LSTM RNN for natural language inputs, and (3) a moderately deep fully connected network for economic data. The outputs of these modules are then concatenated and fed through a final softmax layer which normalizes the predicted scores for each class into a probability distribution. A visual representation of the model is included in Fig. 4. This model is modular in design and any combination of (1), (2), and (3) can be employed in isolation. We experiment with individual components of the model as well as the three components in conjunction.

The CNN component, module (1), used in the model is the ResNet model from He et al. As noted above, this model architecture has shown strong performance on standard tasks such as ImageNet, and is frequently used in transfer learning applications. Briefly, ResNet consists of 50 convolutional layers of varying size applied in succession. Additionally, the architecture incorporates residual connections where the output of layer denoted $H(x) = F(x)$ is modified as $H(x) = F(x) + x$. This gives the network the ”choice” to backpropagate the gradient through an identity matrix, potentially allowing for improved gradient flow to lower network layers.

For natural language inputs in module (2), we initialize word vectors using GloVe embeddings from Pennington et

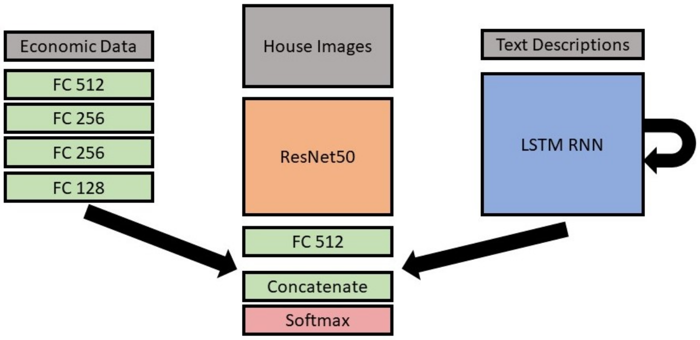

Figure 4. A visual representation of the titular ”Multimedia Model.”

al. We chose to use the embeddings trained on Twitter data. We believe the house descriptions written by users will be closer in nature to tweets than Wikipedia articles or Common Crawl. Two hidden layers are used in the RNN, and timesteps are marked by individual dense word vectors. A single output vector is produced at the last timestep of the RNN, and the intermediate outputs are discarded.

For training purposes, crossentropy loss, which is displayed below, is used as a cost function. The Adam optimization algorithm (18) is employed.

$$L^{(i)} = -\sum_{j=1}^{M} y_j^{(i)} log(\hat{y}_j^{(i)})$$

The hidden layer sizes and quantities shown in Fig. 4 represent an empirical ”optimum” that was manually adjusted over many trials. Although we had neither the time nor resources to run a more thorough hyperparameter search, we found that the indicated number of layers and sizes performed best. In particular, note that we do not have any hidden layers in between the concatenation operation and the softmax transformation.

3. Experiments

4. Multimedia Model

As discussed above, the inherent modularity of the proposed neural architecture makes training individual models on a specific subset of inputs quite easy. In addition to training the model with the combination of image, economic, and textual inputs, we also trained separate models on each input individually. The results are presented in Fig. 5. We believe that isolated image inputs allow the model to achieve higher performance primarily because available economic data is mostly on a ZIP code level, which does not differentiate between individual houses enough.

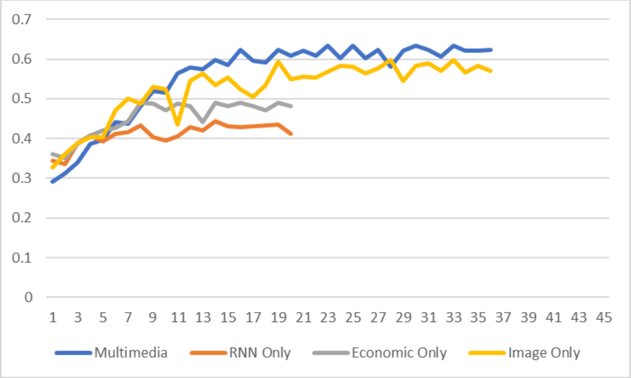

Figure 5. Progression of top-k validation accuracy as a function of training time for individual model components.

4.1. Regularization Penalties

The first results of our model showed us that the model was over-predicting the middle buckets. To remedy this we designed a regularization term that penalizes the model from predicting the middle buckets. This penalty is added to our primary loss function, cross-entropy loss. The function is presented as

$$L_2^{(i)} = L^{(i)} + \frac{\alpha}{|\hat{y}_i - \frac{M}{2} + \epsilon|}$$

where L is our previous softmax loss, ϵ is a small constant, M is the number of classes, α is a tunable hyperparameter, and $\hat{y}_i$ and y_i are the predicted and true labels for the i-th example. The results for different values of α appear in Figure 6.

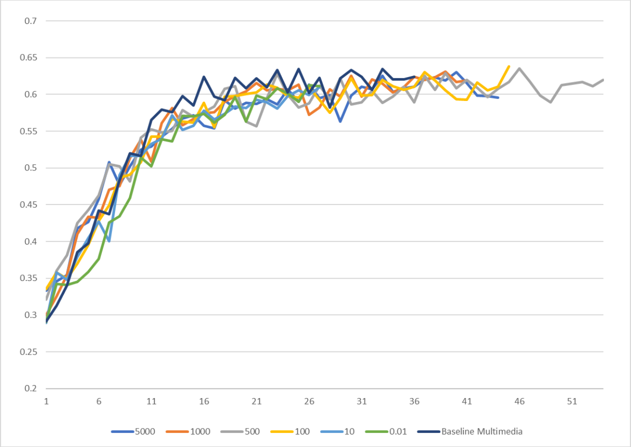

Figure 6. Top-k validation accuracy as a function of training time corresponding to different values of the hyperparameter α in the distance-based regularization penalty. For comparison, results from a model trained without the penalty are also shown.

From the figure, we can see how the model with the previous loss performed better for a number of epochs. Also, after testing on a range for α that went from 0.01 to 5000 we can conclude the extra term on the loss makes little difference. We also added a regularization term penalizing $-Var(\hat{y}_{1:k})$, where k is the batch size. Similarly, this had no obvious effect on accuracy rates.

4.2. Class Weighting

Another proposed solution for the bias towards predicting middle classes was to provide a specific weight for each of the M classes. The weighting is defined as follows:

$$W_i = \frac{1}{\beta * p_j}$$

where p_j is the empirical proportion of training data points belonging to that j-th class and β is a tunable hyperparameter. The results are shown in Figure 7.

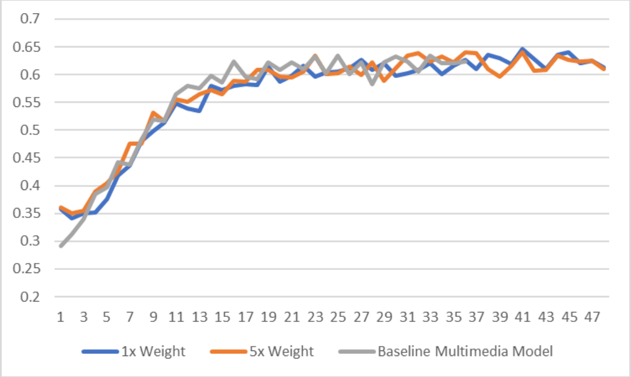

Figure 7. Top-k validation accuracy as a function of training time for different values of the hyperparameter β in the class weighting function. Results from the full model trained without class weighting is also presented.

As we can see, the weighting's impact on the accuracy of our model is difficult to discern, but accuracy did improve by a few percentage points.

4.3. Bucketing

We have tried numerous bucketing schemes, including the obviously poor linear spacing method described earlier and the geometrically spaced method that most of our experiments incorporate. However, we also tried a uniformly spaced bucketing scheme where each class contained roughly equal numbers of training observations. The empirical distribution from the training set is shown in Fig. 8.

Unfortunately, we observed no improvement in accuracy after training our model with these new buckets. We thus conclude addressing the class imbalance does not improve the model's performance.

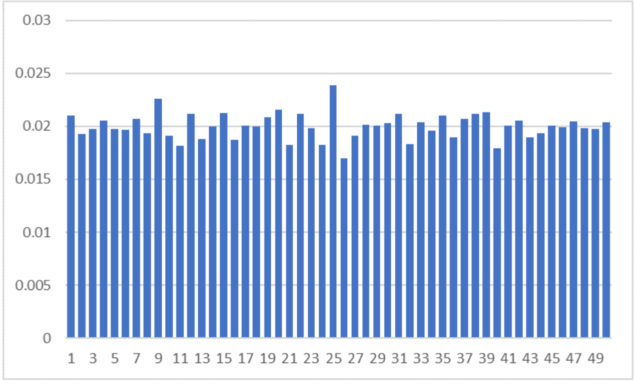

Figure 8. A histogram displaying the distribution of training set observations under the uniform bucketing scheme.

4.4. Saliency Visualization

After training our model, we generate saliency maps (19) for a number of random images. Figures 9 and 10 contain some examples.

From both figures we can see how the model captures not only the general outline of houses, but also tends to lend particular importance to windows and doors. Conversely, note that the model learns to ignore the sky and grass, although we occasionally observe the saliency maps highlighting an extraneous piece of sky. The images from Fig. 9 illustrate a key limitation of the current model, however, showing a great deal of uninterpretable noise in the image on the bottom left. Since there are relatively few indoor scenes in the dataset, the model fails to interpret them correctly.

4.5. Qualitative Error Analysis

In this section we present a qualitative error analysis from a number of validation set observations whose predictions deviate from their labels by the largest absolute value. As observed earlier in the saliency maps, the model pays considerable attention to windows and doors in the images. For the first image in Figure 11, we note that the building is an apartment with many windows and doors, and the model may thus be confused into a predicting a higher value than it should. The error associated with the second image is easily interpretable; the image does not contain a picture of any house at all. In general, this is a minor problem with our dataset, since there is a non-negligible portion of the images that are indoor scenes or do not show a standard front view of the house.

The third picture is worth much more than the model predicted, but we can offer a similar analysis as before, in that there is only one door, and practically no windows, which may confuse the model into predicting a lower class. Furthermore, it is possible that the unusual colors of the house make it harder for the model to classify. Finally, the bottom

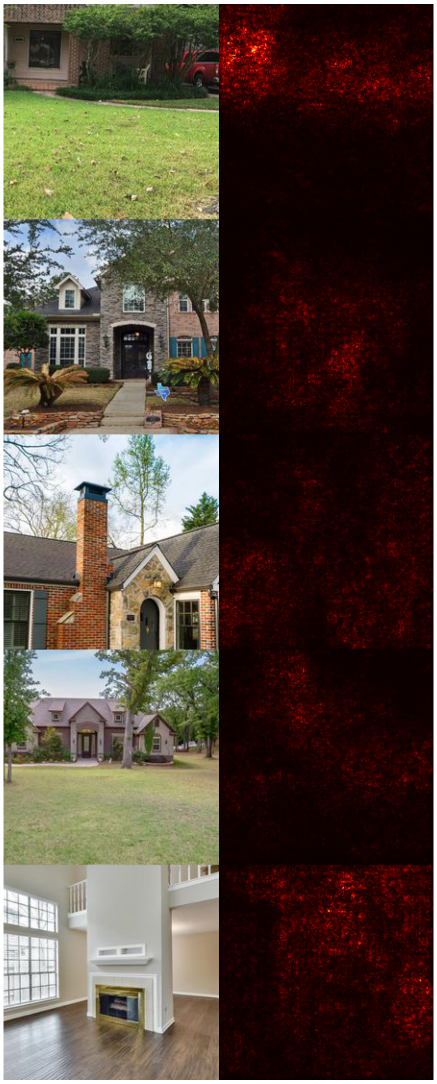

Figure 9. Sample saliency maps for randomly selected images.

image may be difficult to classify because there is a white-painted tree obscuring the image of the house. We have previously observed that the model is successfully able to ignore vegetation in the images, but the unusual coloration of this image probably blends with the house and distorts the model's prediction.

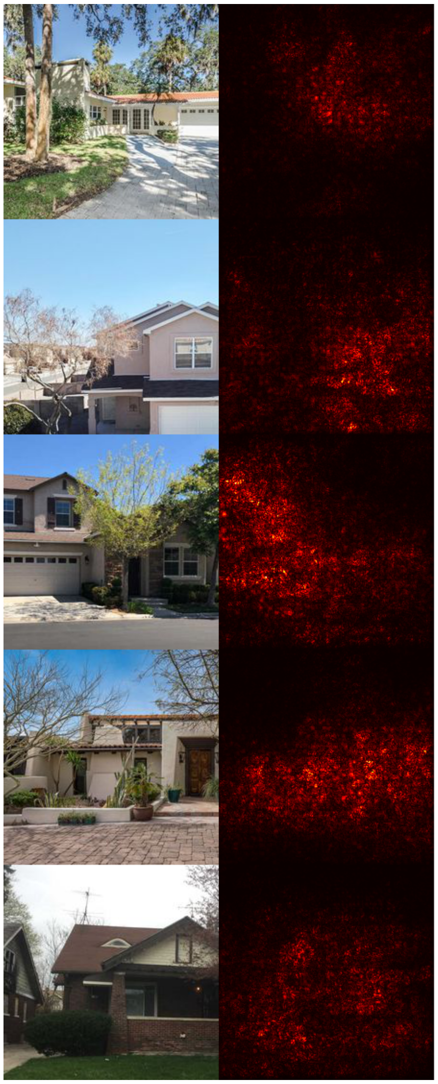

Figure 10. Sample saliency maps for randomly selected images.

4.6. Comparison of Results

Finally, we present a summary of our results from multiple experiments. In general, few of our experiments resulted in any improvement over the primary model, referred to above as the "Multimedia Model." The addition of customized regularization penalties and experiments with various bucketing methods did not yield improved results; however, we saw that weighting the classes according to the em-

Dataset	Baseline	Multimedia	Class Weighted	Multimedia (Top-k)	Class Weighted (Top-k)
Training	9.4	18.3	**20.2**	**69.6**	69.4%
Validation	9.1	16.5	**17.6**	63.4	**64.7%**
Test	8.9	16.0	**17.3**	62.9	**64.6%**

Table 2. A presentation of accuracy results from the most successful experiments. Top-k accuracy results are presented for k = 5.

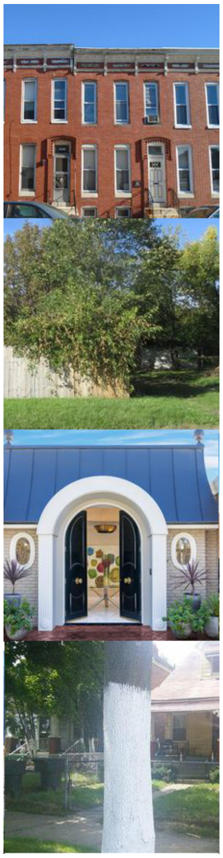

Figure 11. Examples of incorrectly predicted images. The predicted labels deviate from their true labels by -24, -23, 23, and -23, respectively. A negative value denotes that the model predicted a higher value for the house than is actually the case.

pirical distribution of prices in the training set gave measurably better performance. The results demonstrate the complexity of the problem in that advanced models have difficulty achieving high accuracy rates. It is likely that the Zillow dataset is simply too varied and is not standardized enough to support deep learning with less than 50,000 training observations.

Figure 12 illustrates the distribution of predictions we got from our model for the validation set. These predictions are the result of the best experiment we performed. Ideally, the distribution should look Gaussian and resemble Figure 3.

We also present a confusion matrix of test results for the best performing model in Fig. 13. The figure demonstrates a generally satisfactory trend of results, but there is much more variance around the correct class than is ideal. The white diagonal coloring shows how a lot of the predictions fall on or are close to the diagonal that represents correct classifications.

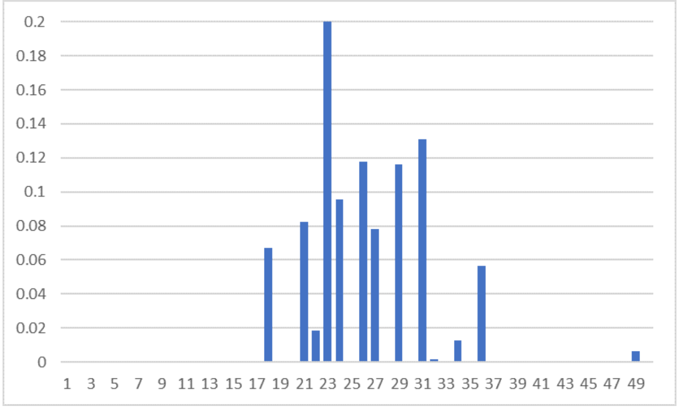

Figure 12. Distribution of predictions for the validation set based on our best experiment, the Multimedia model with class weights.

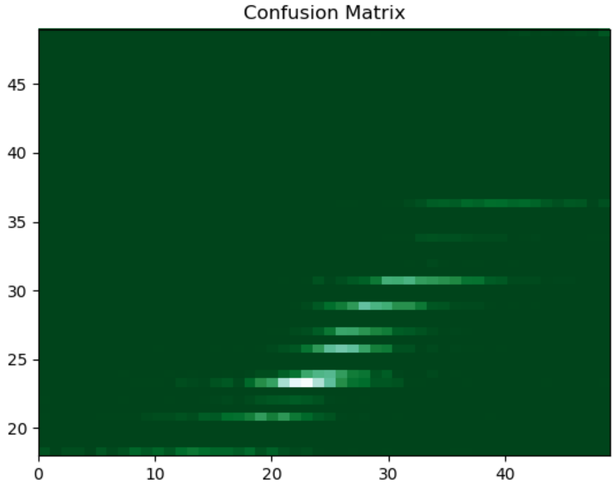

Figure 13. A confusion matrix for test results on the class-weighted Multimedia Model.

5. Conclusion

In this paper we have presented a deep neural model for the prediction of housing prices, a problem that is formulated as a multiclass classification problem. We have also presented a novel dataset drawn from the popular online real-estate market, Zillow. The data used by our model draws from three different media, including images of houses, ZIP code-level economic data, and user-generated text-based descriptions of the houses.

Our experiments show that our prediction task is a difficult one due to the small size of our dataset and its natural diversity. Most attempts to address class imbalance problems in our dataset were unsuccessful, but supplying different weights to each class based on the training set's empirical distribution improved performance.

Perhaps the most obvious next step is to increase the size of the dataset. We may also see significant performance gains by gathering a wider array of economic features to complement the images and text descriptions. We are also intrigued by the idea of using the same data and a generative adversarial network (GAN) to generate images of houses corresponding to each bucket, thus producing a range of differently-priced house images.

6. References

1. Limsombunchai, Visit. ”House price prediction: hedonic price model vs. artificial neural network.” New Zealand Agricultural and Resource Economics Society Conference. 2004.

2. Selim, Hasan. ”Determinants of house prices in Turkey: Hedonic regression versus artificial neural network.” Expert Systems with Applications 36.2 (2009): 2843-2852.

3. Nagaraja, Chaitra H., Lawrence D. Brown, and Linda H. Zhao. ”An autoregressive approach to house price modeling.” The Annals of Applied Statistics (2011): 124-149.

4. Park, Byeonghwa, and Jae Kwon Bae. ”Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data.” Expert Systems with Applications 42.6 (2015): 2928-2934.

5. You, Quanzeng, et al. ”Image-Based Appraisal of Real Estate Properties.” IEEE Transactions on Multimedia 19.12 (2017): 2751-2759.

6. Russakovsky, Olga, et al. ”Imagenet large scale visual recognition challenge.” International Journal of Computer Vision 115.3 (2015): 211-252.

7. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

8. Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

9. He, Kaiming, et al. ”Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

10. Szegedy, Christian, et al. ”Going deeper with convolutions.” Cvpr, 2015.

11. Chollet, Franois. ”Xception: Deep learning with depthwise separable convolutions.” arXiv preprint (2016).

12. Howard, Andrew G., et al. ”Mobilenets: Efficient convolutional neural networks for mobile vision applications.” arXiv preprint arXiv:1704.04861 (2017).

13. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. ICLR Workshop, 2013.

14. Pennington, Jeffrey, Richard Socher, and Christopher Manning. ”Glove: Global vectors for word representation.” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

15. Zillow. https://www.zillow.com/

16. IRS Statistics of Income Division. https://www.irs.gov/statistics

17. Federal Housing Finance Agency (FHFA). https://www.fhfa.gov/

18. Kingma, Diederik P., and Jimmy Ba. ”Adam: A method for stochastic optimization.” arXiv preprint arXiv:1412.6980 (2014).

19. Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. ”Deep inside convolutional networks: Visualising image classification models and saliency maps.” arXiv preprint arXiv:1312.6034 (2013).