Prediction Of Real Estate Price

Ali Batuhan ÜNDAR
aliundar@hacettepe.edu.tr

Muhammed İkbal ARSLAN
muhammed-arslan@hacettepe.edu.tr

Muhammed Enes KOÇAK
muhammed.kocak@hacettepe.edu.tr

Abstract

Nowadays, it is obvious that some advanced and new techniques are necessary to solve the problems such as uncertainty and unfairness of the methods used in determining the house prices. Our project has emerged to solve this problem and is about predicting house prices. Improved machine language methods on the implementation of this project strategy are explained. These methods include 3 "shallow" and 2 "deep" learning algorithms. We used set of continuous data like size and room count alongside with some image data. For this purpose we tested shallow learning algorithms, such as K-Nearest Neighbour(KNN), Linear regression and Support Vector Machine(SVM) and deep learning algorithms such as Convolutional Neural Network(CNN) and Artificial Neural Network(ANN). Additionally, we tested our own collected and optimized data with these models. The results of the modeling and comparison analysis are presented such that they can be used as an helper to the understanding the specific results on used models.

1. Introduction

The need for estimation of house prices, which led to the emergence of our project, is one of the main problems that machine learning deals with and defines. The estimation made after this need may be complicated by the continuous change in the factors affecting the prices. However, such an estimate is needed for individuals and groups such as sellers, buyers, tenants and agents. There are many usage area such as it helps tenants and buyers when making a decision about renting a house.

In addition, it helps sellers and tenants decide on the best possible prices. When performing the classification process, which is one of the main focuses of the project, it will help us to catch a fixed price scale and will prevent unnecessary money flow paid to the intermediaries as it will estimate the price by making a classification according to the luxury situation of the houses. Additionally, measures will be taken against deception and fraud, which has become a sensitive issue in Turkey, we are pleased to be easy.

At the last point of today's technology and machine learning, although we have unlimited access to our great data, we need to understand that processing these data is a problem in itself.

• With the help of the methods and researches that come with machine learning, we tried to overcome these problems as much as we could.

• Our models allow us to perform accurate estimates while balancing the average expectation based on data from almost infinite data repositories. We can say that this is our main goal in our project.

Our project is to create a kind of shallow and deep machine learning models which can predict housing prices by using the variables obtained from the collected house properties. For this purpose, we prepare our data using the data obtained from real estate sites. However, there are some problems that you may encounter during the collection of your own data. If such data includes visual content, such problems multiply. That's exactly what we've been facing in the project. The data we collected had to go through some intensive process steps before being used. We also had to check if our data was balanced in itself. We had to test whether our data had sufficient length or feature size, so we had to create a general solution. Our data contains an average of 5 thousand house data, while the average 10 images per house. As expected, there were too many image data to process and prepare for the model.

After giving a brief information about the data we created, we will continue with the providing information about shallow models and deep models in related titles.

2. Related Work

When we searched for the related works about predicting house prices, we saw that some works have the same research area like us, such as image classification and regression-based implementations for continuous data. We mainly narrated the producable and extensible code basely Caffe framework and AlexNet CNN architecture on multi featured images such as [9], [5].

In the following lines we will contribute about how it performs with the data which we collect. Additionally, we find linear support vector regression solutions closely to ours such as [2]. But when we apply this regression model to our dataset we take 60% test accuracy and which is the lowest value if we compared to other implementations. Also this [7] article talks about how interior and exterior photos of houses impact the price prediction made by computer.

While we research the kNN effect, founded [6] is about how the surrounding neighborhood affects the price of a real estate. Not exactly the same thing but helpful. We saw that neighborhood as important as hyper parameters. For this reason we read some recently made studies such as [4] that affects the house prices and how much they affect. Very helpful when doing fine tuning. While analyzing the parameters, we realized that some locations is weightly affect the stabil results. In this way, [8] helped us a lot. But it only uses location and image data to make prediction. Meaning it make predictions mostly based on images and they are claimed their method is pretty accurate.

In addition to these articles, we found [3] and this is about the Support Vector Regression(SVR) and Neural Network implementation for house price estimation. They say the SVR model was trained and tested with the extracted features according to the relationship between the number of features and the accuracy of the estimation model. In the light of this information, we tested that SVR model on our dataset too.

3. The Approach

Our approach to realizing our project to estimate home prices was to improve the data set and learn the reason for the low accuracy score. We had to understand whether this problem originated from the models or from the data set. Then, as we mentioned before, we tested the ”shallow”, ”deep” and ”hybrid” learning algorithm models and made improvements. However, we wanted to reflect the train accuracy we received to the tests. For this purpose, we experimented on neural network models and finally we decided on one combined hybrid model. After talking about the models, we will talk about the structure of the neural network we have created.

We mentioned that the model accuracy failure with the data which we collected and these failures was about some weird features. We overcame this problem with changing the importance of some houses and adding threshold value for eliminate them.

3.1. Data

First, we did some analysis of data. To determine the usable data for our model, we investigated the effects of certain criteria on price. For example, after investigations, we realized the location of the houses is quite effective on the price. But we had to make them continuous data to use location data. To overcome this problem, we chose to use location data by enumerating. Especially for shallow models, we have analyzed that some features alone will not be sufficient.

In line with the analysis, we determined the features that could be useful for each model and created special datasets for the models. This gave us the advantage of avoiding the problems that will cause time loss before we start designing models.

3.2. Shallow Models

The K-Nearest Neighbor (KNN) model, which we first realized on house prices, was not very pleasing. But as a beginning, we realized that it wouldn't be right to just settle with that. We could predict what caused these results. We already selected our features and after that we looked to the houses itself. We knew that there were some houses have 11 bathrooms or 10 bedrooms, there are huge ones. So, we filtered some of them. As a result of filtering, our KNN accuracy increased to 60–70% like that we expected.

After that, when we tested the data we collected with Linear Regression and Support Vector Machine models, the results were not as bad as like beginning of KNN model. Linear regression gives us a bit stable accuracy like around 70% with the filtered data. Additionally, we tried the model with the houses from Ankara. We saw that the dissension between the prices of houses from different locations are effects the accuracy negatively, in Turkey actually.

While creating the Support Vector Machine(SVM) model, we used the number of square meters, rooms and bathrooms counts. The building age was not used as a parameter because it affected our results badly. Especially the data from “Ankara”. Additionally, we applied a linear regression model and the most effective feature on regression was bathroom number.

3.3. Deep Models

After we confirmed that continuous data, somewhat, works we moved on to deep models for image classification.
We first attempted to use the Convolutional Neural Network (CNN), the most common model used to obtain results from image data. We built our model on VGG16[1] architecture. However, unlike standard VGG16, we've adapted the fully

[1] https://en.wikipedia.org/wiki/VGG

connected layer to ourselves. We thought it would be logical for our model to return a quality score according to the images given. In this way, we have aimed to make the data of our image continuous and determinant.

We categorized the image data according to the prices of the house they belong to. So every image has a quality score. Then we tried to get results by training our model according to class labels. However, the results were not as good as we expected.
In a paper[1] we saw it is possible to get better results using just fully connected layers so we wanted to give it a shot. So we thought about to try to get results using an Artificial Neural Network (ANN).

In order to use image information in ANN, we have transformed the pixel values in the image into a vector and we have classified each picture as a quality score, as we did in CNN. Then we tried to get better results from CNN by trying various optimization methods. However, the only difference we could get in ANN was that it was faster than CNN. Other than that, the accuracy rates were almost no different. But not all things about accuracy. Compared to CNN, ANN learns more consistently even though it learns very little.

3.4. Hybrid Model

We really to believe that our problem can be solved using shallow models. However, since our data includes images, we have limited options here. Therefore, we decided to use a hybrid deep model in which we can use the room images. Basically, our approach here is to train one CNN model for each of the three different rooms of the house using image data. Then we want to connect the outputs of these networks to the ANN by combining them with the continuous data we have. However, when we were training the image data in ANN, we found that we didn't achieve a different accuracy from CNN, and we decided to train the image data in ANN. Because we received faster results in ANN. After this change, our new model has transformed into an ANN model where the results of three ANN are merged together with the continuous data we have.

4. Experimental Results

This section contains 3 different results part.

- Shallow models, which shows us which continuous features we should use in our last models. Also shows how much continuous attributes affects the prediction.
- Deep models, which shows us how should we build our image recognition models. Also shows how much image data will be helping/how effective it will be.
- Combined hybrid model. This will be show us how effective the combination of two models above.

Firstly the shallow models or continuous data testing. We tried four algorithms on our continuous data: KNN, Linear Regression, SVM/SVR and lastly a simple neural network. Without any filtering or thresholding all of the algorithms above gave poor result. That's when we started to do some processing on our data.

First and foremost, location data was a huge issue. Data was, obviously, in string format. We wanted to change this data to numerical value so that we can use it in our models and increase accuracy. For that reason we just enumerated them. We took all the distinct locations and gave them an integer number. But that hardly increased our accuracy. So we think that maybe we can enumerate places in some order. Like prices. So first we ordered places in the order of average prices and then enumerated them. That increased our KNN scores a bit.

Secondly we tried to normalize our data. That reduced KNN score a lot but somewhat improved our linear regression results.

Looking at the regression coefficients we tried to decide what the algorithms believed important and do some feature selection based on that. From that result we deducted that algorithm sometimes weighted some parameters a lot sometimes it did not. For example square meter of a house sometimes took large coefficients, while sometime it took smaller numbers. From that we saw that locations have great impact on how parameters are selected. For example in city ”Izmir” we saw that house prices increased in parallel with number of rooms while in ”Ankara” it increased more with size of house. That's why we decided to filter for a single city. That raised our lowest results on KNN and linear regression to 40% - 50% .

That was another issue with the data. Even with the filter it was still unbalanced. We shuffled data before each prediction and the accuracy varied greatly. While out lowest score was about 40% our highest score was 93% on KNN and 81% on linear regression. We thought about it a bit and decided that perhaps our location enumeration is flawed.

Issue is that while our enumeration increased linearly the prices vary much more complexly. There is no linear relation between neighbouring(places that have enumeration index close to each other) places. That was perhaps because our data was so little in size, 5.000ish data raw, about 2k after location filtering.

Lastly, training of an neural network on continuous data. As we noticed previously the problem we wanted to solve is linear. For that reason we initially tried a ANN with no hidden layer. That gave is really poor results, very close to 0% accuracy. Even on very high epoch rates we couldn't improve it. To be sure we tried wider and larger models with no improvements.
What we saw there, during training was, literal chaos. Loss

was going down but accuracy during batch trains was all over the place. Meaning learning either is super slow or not happening at all.
Let's think about it for a second. Such scenario only occurs when data is inconsistent. Meaning either some of our features contradicts each other or our dataset is just not fit to be learned on. At this point we were, again, at a crossroad. We need to fix our data. For this reasons we scraped another set of data, which we believed more consistent.

After a lot of testing on new dataset we arrived to a conclusion. Results were same. And we were at loss(get it). We searched around for days until we figured out the problem. We were looking at wrong metric. That was the thing caused us a lot of headaches. Of course that would make sense. There is no way accuracy is a good metric for the regression problem.

For that reason we switched over to the "R squared"/"Coefficient of determination" metric. For the first time in weeks our neural network model showed promise.
But not much. Now that we compared our old dataset with the new one we collected the results are obvious.

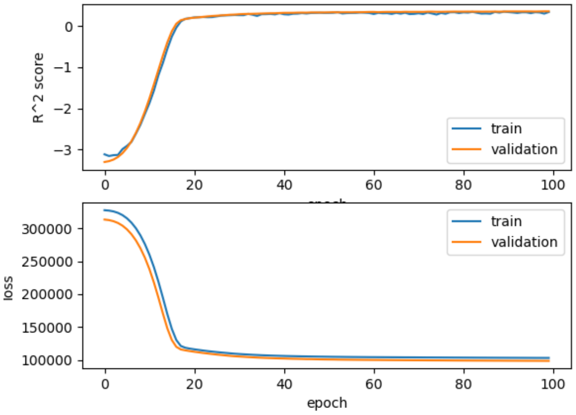

Figure 1. Freshly scraped data collected only from Ankara. Accuracy not really visible here but it reaches up to 36%.

From the graphs above we deducted that our dataset is actually not sufficient when it comes to price prediction. But sadly we are not have a luxury to change dataset this late into project. But do not worry things will change soon enough.

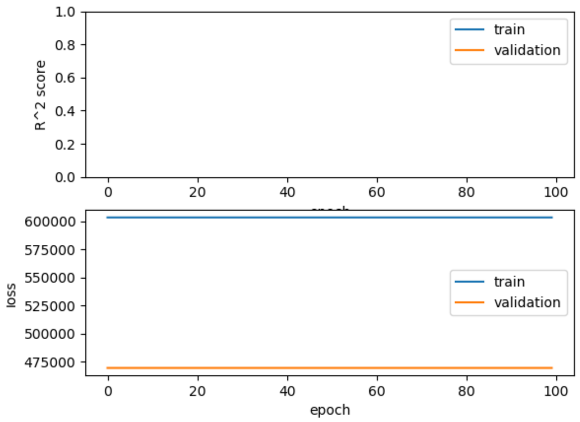

Figure 2. Our old dataset trained on same network. There is no accuracy shown here since the r score is negative (-1.43) here.

With that out of the way, next section is about image classification.
As we mentioned we started with CNN models first. But we had a learning problem with it
After a bit research, from a paper[1] we learned it is possible to predict prices from images just by using ANN. We gave it a shot and showed us a similar results as the CNN, sometimes even better.
The results aren't great but in the end better than nothing.

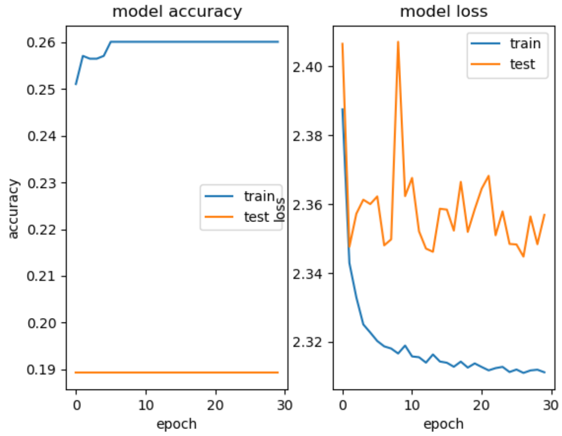

Figure 3. CNN model trained on our VGG16. Even after 30 epochs and 10 hours there is only slight increase in accuracy.

We also ,for comparison, get ourselves some other dataset [1] from internet. This is done in order to see where exactly is our problem. As a results we saw lower accuracy scores on Neural networks compared to our dataset as you

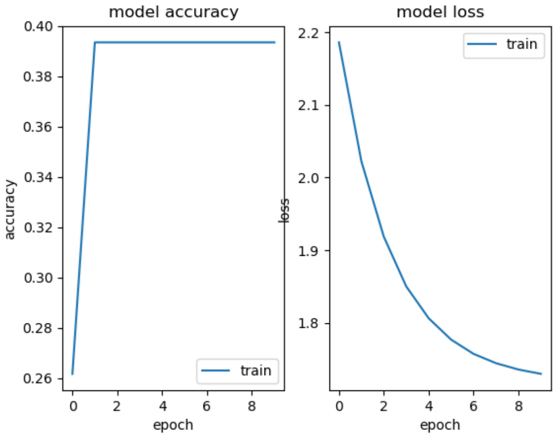

Figure 4. Graph shows sudden increase in accuracy and slow decrease in loss.

can see below

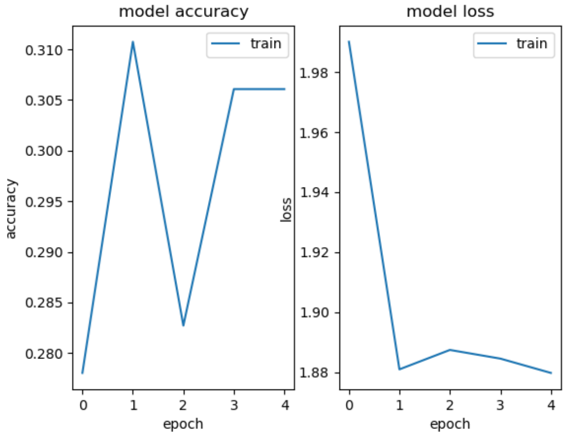

Figure 5. A ready-to-use dataset trained on our classification model.

The problem with this results is not with data being bad. It's data not being compatible with our model.

With this done, let's look at our combined results.
In neural network, individually, models above didn't gave us any good results. But when combined together we started to see some very good results.

This is acquired after lots of tweaking on model. Somehow we couldn't get any good results using the parameters we found on previous tests of models. We will be talking about them shortly.
But can we get this results consistently?
The answer is sadly no. Problem is, while the accuracy on the hybrid model is pretty high, but it still relies on the image classification done on the ANN parts. And accuracy of these layers, as we saw on previous section, is kind of low in comparison.

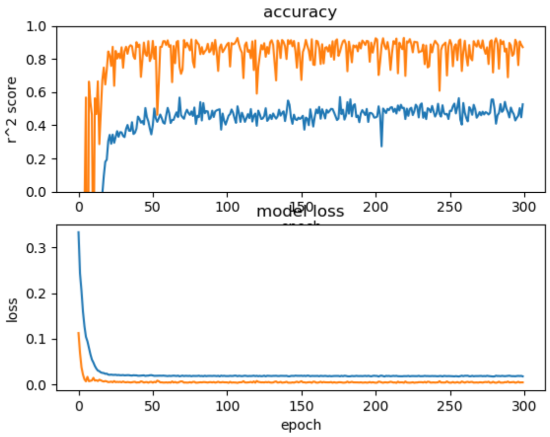

Figure 6. Graph shows 83% accuracy on validation data.

There are a lot of improvements and tweaks could have been done/can be done in future and before that let's talk about our findings.

4.1. Effects of Activation

The effects of activation functions on image classification parts are very subtle. We tried lots of different activations with not much of change.
The regression parts, however, affected by it a lot.

- Relu was an obvious choice when it comes to regression. While it worked well on most parts, when we were working on just continuous data(first section) it didn't performed well. Especially our first collected dataset and the dataset from internet wasn't really fond of it.
- Tanh worked wonders on our first collected dataset. There is an obvious gap compared to the relu. We are not sure why but it was our go to function when it comes to testing of shallow neural network
- For fun, we gave linear function as activation. It, didn't performed well. Not because accuracies were low, but they were inconsistent compared to other two. But we achieved some high scores time to time.

For comparisons sake here is few example results of models trained with different activation functions.

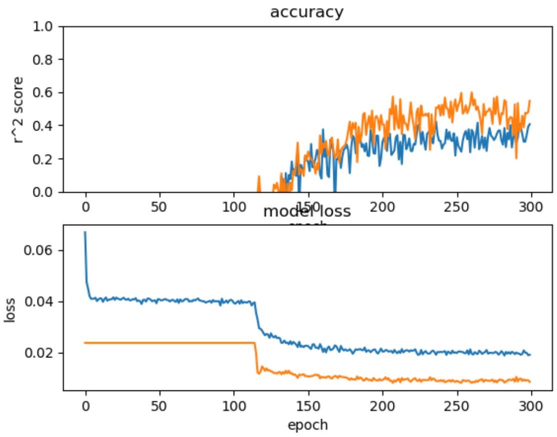

Figure 7. ReLU activation function. It converged eventually but pretty late. Also accuracy was lower and we are not sure if it will reach the accuracy of the tanh in later epochs. Also loss shows some kind of step. Perhaps local minimas were problem for relu?

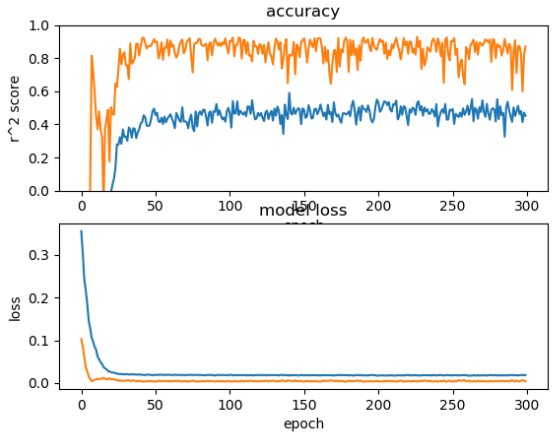

Figure 8. Tanh activation function. It was the best activation function for our dataset. As you can see it converged pretty early on and with higher accuracy.

Keep in mind that the epoch count is really high (300) so when I say converge faster, that means it converged in reasonable time.

4.2. Effects of batching

The batching, again, had no effect on image classification when it comes to ANN. For CNN we cannot say for sure since our machines cannot handle batch size higher than 1 (memory issues with large dataset).

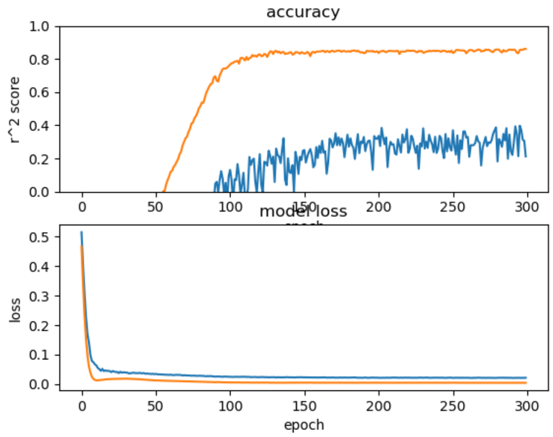

Figure 9. Sigmoid for comparison. Since it is not a classification problem of course it is not performing well(training accuracy). But it is still a surprise to see for it to reach a higher validation accuracy than the relu and very smoothly too. Sadly it also converged very later on.

The regression parts loved high batches. It allowed models converge faster and even more accurately.

We believe this may caused by lowered step sizes. Models were more accurate when epochs have low step counts. Perhaps that was the reason for accuracy increase.

We found that average accuracy of 16-32 worked very well for our dataset. Here is a comparison of hybrid layer with different batch sizes

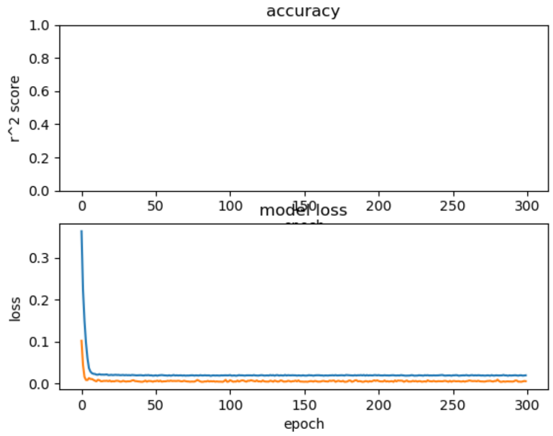

Figure 10. Batch size of 2

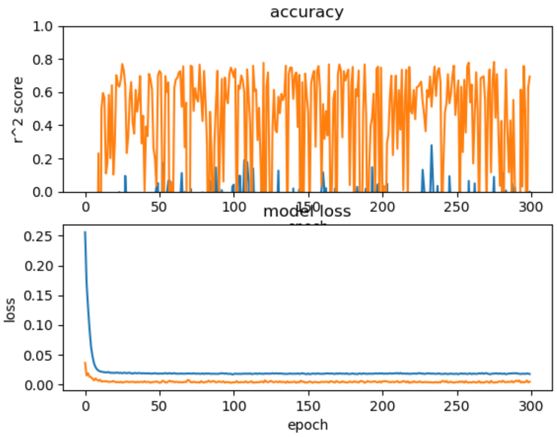

Figure 11. Batch size of 4

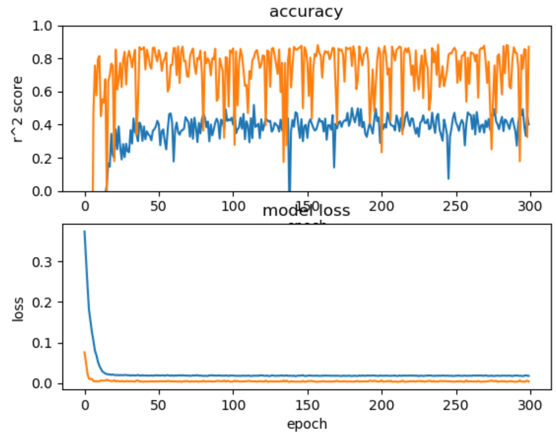

Figure 12. Batch size of 8

After that there is no distinct change. Meaning there is an certain threshold value.

4.3. Some sample predictions

These are some predictions for image classification.

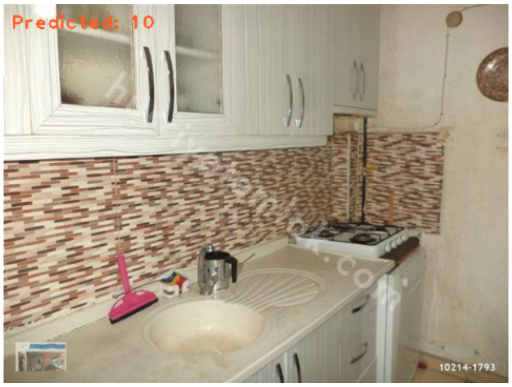

Figure 13. CNN classification Result for Kitchen. A bad example for the kitchen picture is one of the worst examples of which the predicted is 10 despite the class 1 it belongs to.

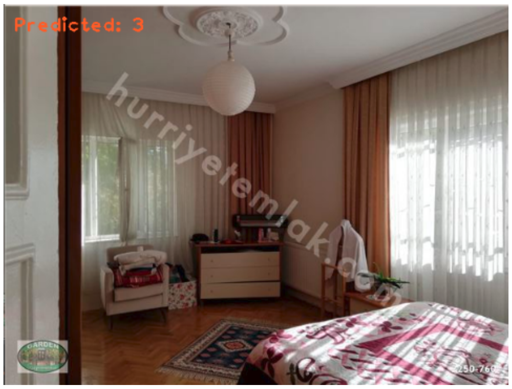

Figure 14. CNN classification Result for Bedroom. One of the examples he predicted in the near class, the bedroom results were not so bad. The class to which it belongs is 5 and the estimated to 3.

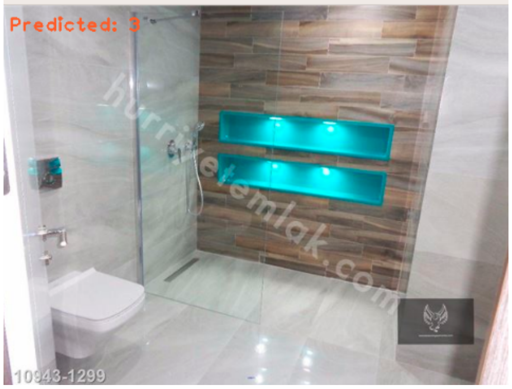

Figure 15. CNN classification Result for Bathroom. As for bathroom images, the results are mostly wrong. An example where the full 4 class predicts below.

The pictures we have are not very clean as seen. So we give up the CNN model which is not as successful as we want. This is the result that makes us choose the ANN structure, which we get similar results.

5. Conclusions

With this article, we share our experiences with the models and experiences we have trained by combining both the visual and textual features that we have gathered for the estimation of house prices. The models we create can be used for future studies, and perhaps the hybrid model that we have designed can be improved to provide better results. Thanks to the experiments, it has been shown that both visual and textual information can be used in model training to provide better estimation accuracy than textual features.

Well results were achieved before using image data. We were able to get high predicted score results even using only continuous data. We had set out to combine image data and continuous data to develop and make predictions of the hybrid model that we had previously considered.

We designed the model we wanted to build as a result of our research. Our model consisted of a CNN and a neural network. Of course, in the process of creating this model, as we have explained, we can say that the shallow and deep models we use are guiding us. We believe that the optimum accuracy values depend on the number of features and stability, the content of the images and the optimal values of the CNN model outputs. We can easily say that we believe that the experimental work we have done until the end of the study from our creation of datasets helped us to educated ourselves and will light to the future works.

References

[1] E. Ahmed and M. Moustafa. House price estimation from visual and textual features. *arXiv preprint arXiv:1609.08399*, 2016.

[2] A. J. M. R. I. B. Alejandro Baldominos, Iván Blanco and C. Afonso. Identifying real estate opportunities using machine learning. pages 11–15, 2018.

[3] M. N. M. Eman H. Ahmed. House price estimation from visual and textual features. 2016.

[4] Y. Gu. What are the most important factors that influence the changes in london real estate prices? how to quantify them? 2018.

[5] H. X. Y. W. K. W. Li Yu, Chenlu Jiao. Prediction on housing price based on deep learning. 12(2):90–98, 2018.

[6] B. L. Marco De Nadai. The economic value of neighborhoods: Predicting real estate prices from the urban environment. 2018.

[7] S. B. Omid Poursaeed, Tomas Matera. Vision-based real estate price estimation. 2017.

[8] L. C. J. L. Quanzeng You, Ran Pang. Image based appraisal of real estate properties. 2016.

[9] T.-J. Y. J. E. Vivienne Sze, Yu-Hsin Chen. Efficient processing of deep neural networks:a tutorial and survey. pages 1–23, 2017. `https://arxiv.org/pdf/1703.09039.pdf`.