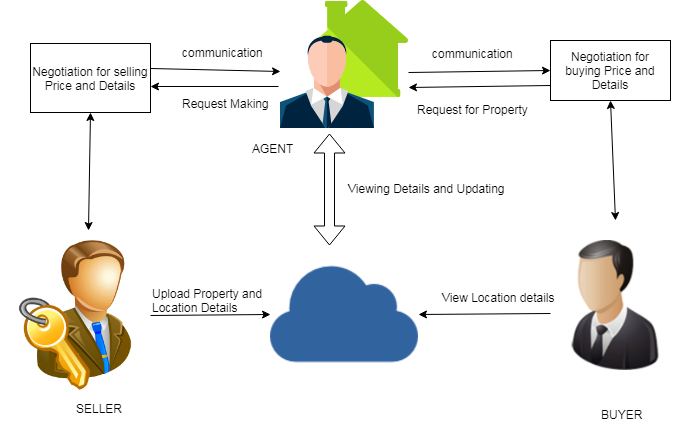
**IMAGE BASED APPRAISAL OF REAL ESTATE PROPERTIES**

**ABSTRACT**

Real estate appraisal, which is the process of estimating the price for real estate properties, is crucial for both buys and sellers as the basis for negotiation and transaction. Traditionally, the repeat sales model has been widely adopted to estimate real estate price. However, it depends the design and calculation of a complex economic related index, which is challenging to estimate accurately. Today, real estate brokers provide easy access to detailed online information on real estate properties to their clients. We are interested in estimating the real estate price from these large amounts of easily accessed data. In particular, we analyze the prediction power of online house pictures, which is one of the key factors for online users to make a potential visiting decision. The development of robust computer vision algorithms makes the analysis of visual content possible. In this work, we employ a Recurrent Neural Network (RNN) to predict real estate price using the state-of-the-art visual features. The experimental results indicate that our model outperforms several of other state-of-the-art baseline algorithms in terms of both mean absolute error (MAE) and mean absolute percentage error (MAPE).

**ARCHITECTURE:**



**MODULES:**

In this project there are four modules present as listed in the below

* Property Addition
* Adding Location Details
* Price Negotiation
* Geometrical Analysis

**MODULES DESCRIPTION:**

1. **PROPERTY ADDITION**

The property addition is the main initiative module for the project. Once authorized user login into the system, they can perform their activity as per their wish. In this module, User must have interested in selling the property which they own. The Property details such as Location, Address, and Facilities that the households are need to add to the cloud where everything that seller uploads can viewable to buyer and agent.

1. **ADDING LOCATION DETAILS**

In this module user that is seller need to upload the details of their location as well as their neighboring facility location such as schools, colleges and medical etc., In previous modules also user need to add the location that are into the raw typed format but here in this module we can upload the location details in maps and map formats. Spotting these locations can be very handy for agents or users to get to know about the details of property and neighboring details.

1. **PRICE NEGOTIATION**

This module is mainly designed for buyers and agents. Firstly, buyer sends the request to agents along with the cost of expectations and other query details about property. Once agents view the request from the buyer, Agent can decide the price according to the merit of location and both the buyer and seller. This module designed like chat. Dual way communication can be accomplished among the various users.

1. **GEOMETRICAL ANALYSIS**

The Geometrical analysis of given data set is done by charts. Here in this project there are two graphs have been plot between numbers of locations versus city. The pie chart and line charts are established in this project in order to analysis the data effectively.

**EXISTING SYSTEM:**

Current research from both estate industry and academia has reached the conclusion that real estate value is closely related to property infrastructure, traffic, online user reviews and so on. Generally speaking, there are several different types of appraisal values. In particular, we are interested in the market value, which refers to the trade price in a competitive walrasian auction setting Traditionally, both real estate industry professionals and researchers have relied on a number of factors, such as economic index, house age, history trade and neighborhood environment and so on to estimate the price. Indeed, these factors have been proved to be related to the house price, which is quite difficult to estimate and sensitive to many different human activities. The current algorithms are 1). Regression Models and 2). Deep Walk. Regression model has been employed to analyze real estate price index. Recently, the results in Fu et al. show that sparse regularization can obtain better performance in real estate ranking. Thus, we choose to use LASSO which is an l1-constrained regression model, as one of our baseline algorithms. Deep Walk is another way of employing random walks for unsupervised feature learning of graphs. The main approach is inspired by distributed word representation learning. In using Deep Walk, we also use \_-neighborhood graph with the same settings with the graph we built for generating sequences for B-LSTM. The learned features are also fed into a LASSO model for learning the regression weights. Indeed, deep walk can be thought as a simpler version of our algorithm, where only the graph structures are employed to learn features. Our framework can employ both the graph structure and other features, i.e. visual attributes, for building regression model.

**DISADVANTAGE:**

The existing system is quite difficult to estimate and sensitive to many different human activities. There are lot of difficult works have been done with the existing systems to measure the number of factors such as economic index, house age, history trade and neighborhood environment.

Current research from both estate industry and academia has reached the conclusion that real estate value is closely related to property infrastructure, traffic online user Reviews and so on.

**PROPOSED SYSTEM**:

We intend to employ the pictures for the task of real estate price estimation. We want to know whether visual features, which are a reflection of a real estate property, can help estimate the real estate price. Intuitively, if visual features can characterize a property in a way similar to human beings, we should be able to quantify the house features using those visual responses. Meanwhile, real estate properties are closely related to the neighborhood. In this work, we develop algorithms which only rely on 1) the neighbor information and 2) the attributes from pictures to estimate real estate property price To preserve the local relation among properties we employ a novel approach, which employs random walks to generate house sequences. In building the random walk graph, only the locations of houses are utilized. In this way, the problem of real estate appraisal has been transformed into a sequence learning problem. Recurrent Neural Network (RNN) is particularly designed to solve sequence related problems. Recently, RNNs have been successfully applied to challenging tasks including machine translation, image captioning, and speech recognition. Inspired by the success of RNN, we deploy RNN to learn regression models on the transformed problem. The main contributions of our work are as follows: To the best of our knowledge, we are the first to quantify the impact of visual content on real estate price estimation. We attribute the possibility of our work to the newly designed computer vision algorithms, in particular Convolutional Neural Networks (CNNs). We employ random walks to generate house sequences according to the locations of each house. In this way, we are able to transform the problem into a novel sequence prediction problem, which is able to preserve the relation among houses. We employ the novel Recurrent Neural Networks (RNNs) to predict real estate properties and achieve accurate results.

**ADVANTAGE:**

A picture is worth a thousand words. One advantage with images and videos is that they act like universal languages. For the given house pictures, people can easily have an overall feeling of the house, e.g. what is the overall construction style, how the neighboring environment looks like. These high-level attributes are difficult to be quantitatively described

Map Based Location information are most commonly effective than the viewing in raw details. The most accurate details can be viewed in simple steps

The proposed algorithms are very effective than the existing algorithms such as LASSO and Deep Walk.

**ALGORITHMS:**

**DIGITAL IMAGE PROCESSING:**

Digital image processing is the use of computer [algorithms](https://en.wikipedia.org/wiki/Algorithm) to perform [image processing](https://en.wikipedia.org/wiki/Image_processing) on [digital images](https://en.wikipedia.org/wiki/Digital_image). As a subcategory or field of [digital signal processing](https://en.wikipedia.org/wiki/Digital_signal_processing), digital image processing has many advantages over [analog image processing](https://en.wikipedia.org/wiki/Analog_image_processing). It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of [multidimensional systems](https://en.wikipedia.org/wiki/Multidimensional_systems).

**Digital image transformations**

**Filtering**

Digital filters are used to blur and sharpen digital images. Filtering can be performed in the spatial domain by convolution with specifically designed kernels (filter array), or in the frequency (Fourier) domain by masking specific frequency regions. The following examples show both methods:

|  |  |  |
| --- | --- | --- |
| Filter type | Kernel or mask | Example |
| Original Image | {\displaystyle {\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}}} | [Affine Transformation Original Checkerboard.jpg](https://en.wikipedia.org/wiki/File:Affine_Transformation_Original_Checkerboard.jpg) |
| [Spatial Lowpass](https://en.wikipedia.org/wiki/Lowpass) | {\displaystyle {\frac {1}{9}}\times {\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}}} | [Spatial Mean Filter Checkerboard.png](https://en.wikipedia.org/wiki/File:Spatial_Mean_Filter_Checkerboard.png) |
| [Spatial Highpass](https://en.wikipedia.org/wiki/Edge_detection) | {\displaystyle {\begin{bmatrix}0&-1&0\\-1&4&-1\\0&-1&0\end{bmatrix}}} | [Spatial Laplacian Filter Checkerboard.png](https://en.wikipedia.org/wiki/File:Spatial_Laplacian_Filter_Checkerboard.png) |
| [Fourier Representation](https://en.wikipedia.org/wiki/Fast_Fourier_transform) | Pseudo-code:  image = checkerboard  F = Fourier Transform of image  Show Image: log(1+Absolute Value(F)) | [Fourier Space Checkerboard.png](https://en.wikipedia.org/wiki/File:Fourier_Space_Checkerboard.png) |
| Fourier Lowpass | [Lowpass Butterworth Checkerboard.png](https://en.wikipedia.org/wiki/File:Lowpass_Butterworth_Checkerboard.png) | [Lowpass FFT Filtered checkerboard.png](https://en.wikipedia.org/wiki/File:Lowpass_FFT_Filtered_checkerboard.png) |
| Fourier Highpass | [Highpass Butterworth Checkerboard.png](https://en.wikipedia.org/wiki/File:Highpass_Butterworth_Checkerboard.png) | [Highpass FFT Filtered checkerboard.png](https://en.wikipedia.org/wiki/File:Highpass_FFT_Filtered_checkerboard.png) |

**IMAGE PADDING IN FOURIER DOMAIN FILTERING**

Images are typically padded before being transformed to the Fourier space, the highpass filtered images below illustrate the consequences of different padding techniques:

|  |  |
| --- | --- |
| Zero padded | Repeated edge padded |
| [Highpass FFT Filtered checkerboard.png](https://en.wikipedia.org/wiki/File:Highpass_FFT_Filtered_checkerboard.png) | [Highpass FFT Replicate.png](https://en.wikipedia.org/wiki/File:Highpass_FFT_Replicate.png) |

Notice that the highpass filter shows extra edges when zero padded compared to the repeated edge padding.

**REQUIREMENT ANALYSIS**

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

**REQUIREMENT SPECIFICATION**

**Functional Requirements**

* Graphical User interface with the User.

**Software Requirements**

For developing the application the following are the Software Requirements:

1. Python
2. Django
3. MySql
4. MySqlclient
5. WampServer 2.4

**Operating Systems supported**

1. Windows 7
2. Windows XP
3. Windows 8

**Technologies and Languages used to Develop**

1. Python

**Debugger and Emulator**

* Any Browser (Particularly Chrome)

**Hardware Requirements**

For developing the application the following are the Hardware Requirements:

* Processor: Pentium IV or higher
* RAM: 256 MB
* Space on Hard Disk: minimum 512MB

CONCLUSION:

In this work, we propose a novel framework for real estate appraisal. In particular, the proposed framework is able to take both the location and the visual attributes into consideration. The evaluation of the proposed model on two selected cities suggests the effectiveness and flexibility of the model. Indeed, our work has also offered new approaches of applying deep neural networks on graph structured data. We hope our model can not only give insights on real estate appraisal, but also can inspire others on employing deep neural networks on graph structured data.