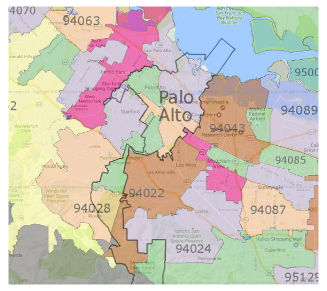
# Siamese Neural Network for Identifying Duplicates in Real Estate Databases

Acknowledgements: dataset - MLS Listings, Inc.
Advisors: **Rishi Bedi**, **Olivier Moindrot –** Stanford University

Sergey Ermolin



## Motivation and Business Use-Case

- Real Estate databases often contain duplicate entries, i.e. the same house/lot appears more than once with different attributes. For example, a house located in Milpitas would often be listed in both East Bay and South Bay databases. The content of these entries could be different to appeal to different demographics of each area.
- When data feeds from EastBay and SouthBay databases are merged, it results in two duplicate listings which violates database integrity. The purpose of this project is to provide means to identify and flag these duplicates for future removal.
- Contrary to what one might think, property's street address by itself is not enough to identify the duplicate entries - it is often misspelled or even misrepresented to make the house appear to belong to a more desirable city (eg: Almaden vs San Jose or Antelope vs Sacramento). One of the most reliable indicators of duplicate listings are jpeg images uploaded by agents.

## Problem Statement

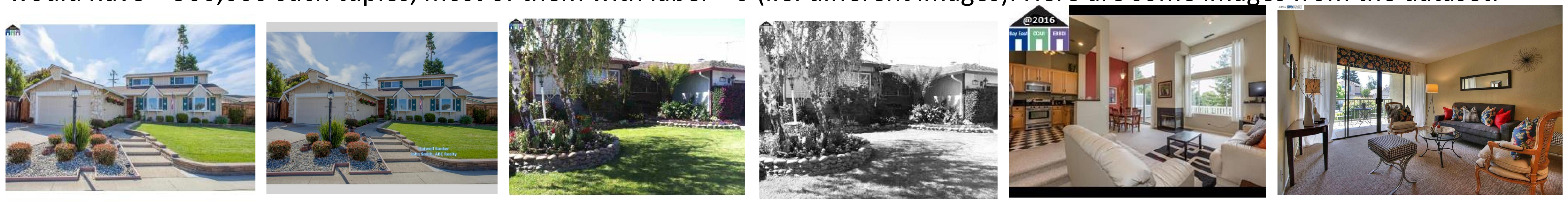
- Simple binary-comparison of images rarely works since the same images from identical listings may differ by size, color enhancements, and added advertisement watermarking
- At least two standard approaches exist for image similarity identification: Siamese (or triplet) networks and semantic segmentation. After consulting with several advisors, we decided to implement a modified Siamese architecture described in [1] because the expected scope was most befitting the time and resources constraints of a 1-person course project.
- Success metrics. From [1] (as well as from CS231N homework), 1-Nearest Neighbor method provides ~25% success rate for this task, while a Siamese network can reach above 70% accuracy (as high as 92%). This accuracy level is deemed to be sufficiently high to identify duplicate database entries when used in conjunction with other relational database attributes.

## References (partial)

[1] Koch, G., Zemel, R. S., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition (2015).
[2] Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. (2005).

## Dataset – courtesy of MLS Listings, Inc, Sunnyvale, CA

MLS Listings, a local real estate database hosting company, provided several datasets of house images (100, 1000, 10,000 images), both unique and "duplicates". The images included both external and internal views of the houses. Obviously, the datasets had more different images than similar ones, so some data augmentation was necessary. I post-processed each dataset, creating a .csv file of pair-wise C(N,2) tuples <filename_Image1, filename_Image2, label>, where 'label' was a binary 0/1 value. A dataset of 1000 images would have ~ 500,000 such tuples, most of them with label = 0 (i.e. different images). Here are some images from the dataset:



## Methods/Algorithms/Models

- We used a Siamese architecture. Basic approach to this problem was first outlined in [2], but we used a more recent modified Siamese architecture described in [1].
- Conceptually, we have to branches (A/B) of identical CNN with the same weights whose outputs are compared via a difference norm and then fed to a sigmoid function for logistic differentiation. See Fig 1 and 2.

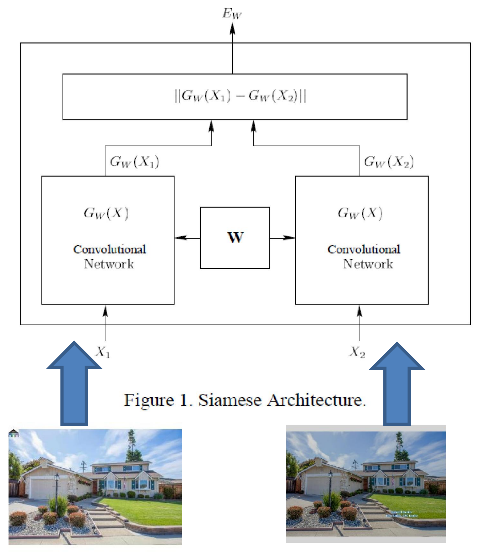


Figure 1. Siamese Architecture.

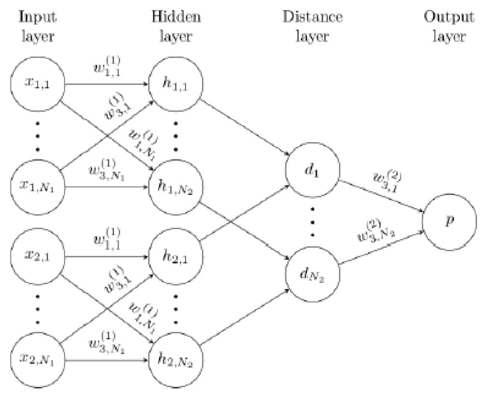


*Figure 2* A simple 2 hidden layer siamese network for binary classification with logistic prediction $p$. The structure of the network is replicated across the top and bottom sections to form twin networks, with shared weight matrices at each layer.

- The modified architecture described in [1] included an additional fully-connected layer [d1…dn] which combined 4096-wide feature outputs of previous layer into a single-value output, essentially implementing the formula

$$\left(\sum_j \alpha_j |\mathbf{h}^{(j)}_{1,L-1} - \mathbf{h}^{(j)}_{2,L-1}|\right)$$

- Conceptually, we have two branches of identical VGG-16 CNNs with the same weights whose outputs are compared via a norm and then fed to a sigmoid function for logistic regression See Fig 1 and 2.

## Experiments and Results.

- We adapted Olivier Moindrot tensorflow VGG-16 transfer learning tutorial for this work.. In Tensorflow, we actually instantiate one VGG-16 branch instead of two as in Fig 1. We have N pairs of images, but we feed 2N images into the input. Before the last fully connected layer D, we take an L1 vector norm of abs([1..N] - [N+1..2N]) and then feed the result to layer D. We use a sigmoid loss function.
- We sanity-checked our network implementation by feeding it a very unbalanced dataset (99% of training pairs and 75% of validation pairs were different) and observed that network returned approximately the same accuracy.
- For real training, we preserved the weights in all but two last layers of VGG model and retrained the entire model.
- On a more balanced dataset of ~ 500 images with 0/1 label ratio of 50% , we were able to achieve validation accuracy of 63% (training accuracy was 75%), indicating that training was taking place.
- We feel that on a larger dataset we will be able to train the model even better and achieve higher validation accuracy.

## Conclusion and future work

We feel that modified Siamese CNN architecture is useful for identifying duplicate entries in real-estate database. Our future work will focus on using 'precision' and 'recall' metrics along with plain accuracy metrics. We will use larger datasets, different L_norms and optimization methods