CS229 – Machine Learning

The Dusk of Survey Data and the Dawn of Aerial Imagery in Economics: Cost-efficient housing price learning in the developing world

Alexandr Lenk, Matias Cersosimo, Rodrigo Naumann[1]

1. Motivation and Related Work

Urban landscape in developing countries is under constant economic and physical transformation. A case study of a particular interest is Dar-es-Salaam, the economic and financial center of Tanzania. The population of Dar has been increasing at a rate roughly constant of more than 5% in the last 30 years, currently reaching 6,300 million. Recently, the city has acquired better infrastructure in terms of public transportation, especially in the form of the Bus Rapid Transit (BRT) and new road amenities in order to handle the increasing flow of people. This has led to existing neighborhoods being reshaped both geo-spatially and socially and as a result neighborhood valuations have been also changing. A common proxy for neighborhood value is housing pricing, in the form of property values and rents.

We believe rent prediction represents an even more valuable tool for the study of neighborhood value heterogeneity as it proxies both for the quality of houses and neighborhoods as well as incorporates benefits in terms of easy road transportation and access to jobs. A common approach in Economics for understanding the determinants of rent is the collection of household survey data at multiple points in time. This technique, however, can be expensive both in terms of money and time and prone to measurement error due to the inherently subjective nature of the instrument. Aerial imagery, on the other hand, could cover a much wider part of the city in a significantly less amount of time and potentially capture intangible features that would otherwise not be contained in a survey.

Development economists and researchers have previously used satellite imagery to get proxies for quality. For instance, Henderson et. al (2012) uses pixel overnight intensity as a proxy for GDP of a given region. In another article, Marx et. al (2016) extracts roof luminosity to account for quality of dwelling structure and Neal Jean (2016) uses satellites images as the inputs of a Convolutional Neural Network to measure poverty.

Our approach consists in using Inception V3 network to extract Deep Features from drone images of Dar-es-Salaam, closely following the work of Bency et. al (2017). We then feed those Deep Features into a prediction algorithm (Regularized Linear Regression and Multi-Layer Perceptron) to predict continuous rent. We compare the prediction performance of our model with a benchmark model based on survey data only as well as one based on raw pixels only (this is, without the Deep Feature extraction phase). We find that all models perform similar in terms of root-mean-squared error, which shows that drone imagery represents a viable cost-efficient alternative to traditional data collection methods. In particular, we interpret the results of the Deep-Feature model as a lower

[1]The three team members have contributed equally to the development of this project.

bound on its performance as we were limited by training size and computational power.

2. Data

Our raw data consists of the following:

Survey Data collected in 2017 by the World Bank from 1700 households in various parts of Dar-es-Salaam. The survey data contains information on monthly rent, household size, the size of dwelling, roof and building material, house amenities such as bathrooms and toilets, as well as neighborhood amenities (availability of water, electricity and street lights).

The image data is broken down into 21 blocks (each one a separate .tif file) corresponding to different neighborhoods of Dar. Each one of these images has a resolution of around 35000×35000 pixels, and consists of three layers (RGB). Each of these pixels represents 7×7 cm of space in the real world. We are using the *rasterio* module in Python to access the geo-tags of each pixel. With each pixel being geo-referenced, we are able to match each household appearing in our survey data to its corresponding pixel values from the image.

Figure 1: Drone Image Example

For each identified household, we draw radii of length 10m and length 40m. The idea is that the lower radius will allow to learn more detailed information on the housing structure itself and the immediate surroundings, whereas the larger radius should allow us to learn more about the neighborhood in terms of shape of street, road quality or occupation density. Overall, we are able to merge 498 households[2], which is our final sample size on which we perform sample splitting for training, validation and testing in proportion of 70%-20%-10%.

3. Transfer Learning & Feature Extraction

First, we perform a multi-level classification task on house wealth. We divide the houses in our train and dev sets into three categories according to rent:

- Low-quality, cheap house:

$$\text{Rent} \leq 30,000 \text{ TSH} - 13 \text{ USD}$$

- Medium-quality, medium-rent house:

$$\text{Rent} \in (50,000, 80,000) \text{ TSH} - (21, 35) \text{ USD}$$

- High-quality, expensive house:

$$\text{Rent} \geq 100,000 \text{ TSH} - 45 \text{ USD}$$

Notice that we drop houses with rents of intermediate values. This is because we want to keep sharp contrasts between the three classes in order to be able to perform the classification more precisely. For the *SoftMax* classification we use *Inception V3*, a pre-trained network on the famous *Imagenet* dataset that has been trained on more than 1 million images belonging to 1,000 classes of objects. For that, we used the *Keras* functional API available in the *Tensorflow* toolboox. While the initial convolutional layers of the network identify more general image features such as lines and shapes, the later layers are trained to identify specific objects for a given task. Thus, we opted to retrain the final block of convolutional filters and fully connected layers using our dataset

[2]It is true that standard machine learning techniques require much larger datasets. However, Economics has to usually deal with small dataset problems. Adapting the techniques to work with less observations is an active area of research in Economics.

while keeping the rest of the layers fixed to the values learnt from *Imagenet*. The fully connected layer is modified to generate K different features which act as input to the final classifier. In order to inhibit over-fitting on training data, we utilize data augmentation, by flipping and rotating the images. The number of features K is our main hyper-parameter, which has been chosen over a 3-step grid ranging from 8 to 35 features. We find that the optimal performance on the dev set is reached with a feature number of 11 features, which results in a classification accuracy of 65%. This is both for the 10m and 40m-radii. Hence, we extract 11 deep features for each radii, which results in a total of 22 deep features. We additionally add a variable on the distance from the nearest road that we calculate based on the Euclidean distance between a given house and the nearest road. Distance to roads is an important predictor for rent since road access is related to job location.

4. Rent Prediction Exercises

In the second step, we do the prediction of continuous rent. We use a Regularized Linear Regression and a Multi-Layer Perceptron. We perform separate rent prediction on 4 types of inputs:

- The Deep Features we extracted previously (size of 22).
- Raw Pixels from 10-m radius image (size of 193551).
- Raw Pixels from 40-m radius image (size of 3048195). Note that the pixels from the 10-m image are necessarily included here.
- Survey Data on dwelling and neighborhood characteristics (size of 15 quantitative + qualitative features).

Each time, we train our model on the full train sample (349 images), and we choose the best hyper-parameters depending on root-mean-squared error on the validation set (99 images). Finally, after having chosen the optimal model, we retrain the model on those optimal hyper-parameters on the train and validation set combined and report root-mean-squared error on the test set (50 images). The formula for the RMSE is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{rent}_{i,r,t}^{(i)} - rent_{i,r,t}^{(i)}\right)^2}$$

At this stage, our hyperparameters are the L2-regularization penalty for the Linear Regression and Multi-Layer Perceptron. Additionally, for the multi-layer perceptron, we select the optimal number of hidden layers and number of hidden neurons per hidden layer. By optimal hyperparameters, we mean hyperparameters that achieve "good" bias/variance trade-off. What does that mean for our sample? The mean rent in the full sample is 50,000 TSH. Thus, we believe that an acceptable bias would be equal to 10% of the mean rent. This is, whenever possible, we set the penalty term such that RMSE equals around 5,000 TSH. Unfortunately, this was possible only for the models with raw pixels as inputs. For the deep features and survey data, we were not able to set such a term to obtain an RMSE of 5,000 on the validation set unless that this would result in an unacceptably high error on the validation set. Hence, for the latter cases we were trying to get RMSE of around 50,000 TSH instead and then obtaining the lowest RMSE on the validation set as possible so as to avoid over-fitting. Similar considerations were applied to the MLP network architecture.[3]

Table 1: Optimal Penalty

	α_{LR}	α_{MLP}
Survey Data	1	1
Pixels 10m	5	5
Pixels 40m	40	0
Deep Features	0	0

[3]We test the 10 different architecture: A single layers with either 5,10,50,100,250 hidden neurons or 2 layers with (5,5), (10,5), (50,5), (100,10), (250,10) hidden neurons. Notice that we could not choose a number of hidden neurons beyond our train set size of 349.

Table 2: Optimal Architecture - MLP

	# layers	# neurons
Survey Data	1	5
Pixels 10m	1	5
Pixels 40m	1	50
Deep Features	1	5

We first tried to train the model using stochastic gradient descent, but unfortunately this was too computationally expensive so instead we chose the Broyden-Fletcher-Goldfarb-Shanno algorithm, which is a Newton-style optimization algorithm that uses an approximation of the Hessian rather than its exact form, but performs much faster. The trade-off is that it is more sensitive to starting values, but after having tried various random seeds, we only get quantitatively negligible differences. Also, to keep the optimization strategy symmetric and computationally feasible, we perform the linear regression as a neural network with a single hidden layer and a single hidden neuron and we apply the identity function as activation. This approach is equivalent to a linear regression, but we recognize that the closed-form solution is more stable.

5. Prediction Results and Conclusions

We present below the plots for the RMSE for the four types of inputs, each plot representing performance of for each of the two prediction algorithms.

Figure 2: RMSE - LR

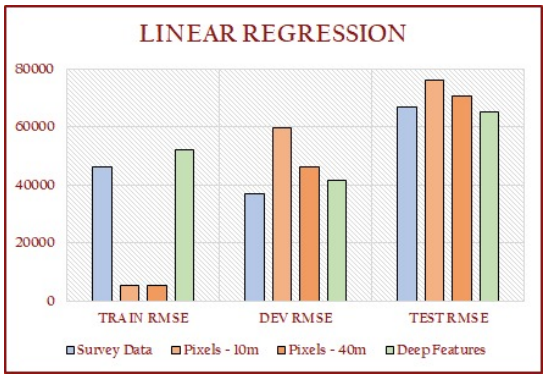
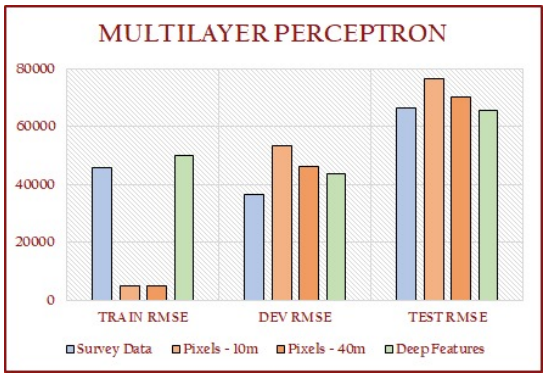


Figure 3: RMSE - MLP



First, we observe that the two sets of plots look very similar. This is not surprising as in the continuous case, the usual activation function for the last hidden layer in a MLP is RELU defined by $\max(W_a + b, 0)$. Since, we could only test rather shallow network architectures with either 1 or 2 hidden layers, there was no space to use non-linear activation functions. Hence, it is not surprising to expect similar performance of linear regression and RELU since the only difference in the activation function is the elimination of negative predictions, but this does not occur in our case so the two should be basically equivalent.

Notice that RMSE is comparable in the train and dev set for the survey and deep feature models, but high. This is evidence of high bias and low over-fit. For the 10m and 40m, we have evidence of low bias and high overfit, although as we had explained, these were the only two models we were able to achieve acceptable bias on the train set in proportion to mean rent. What matters ultimately, however, is the performance on the test set. There, we observe that all types of inputs perform similarly with a RMSE of around 70,000 TSH (30 USD).

The results imply the following conclusions: First, we have established that the deep feature model based on drone images performs the same as the recorded survey data but does so in much more cost-efficient way.[4] Surveys of this nature take

[4]We have additionally run a model where we combine survey data and the 22 deep features, and we observe no significant changes in RMSE which implies that the deep features contain at least as much information as the survey data

months of planning and working in the field, and cost around 200,000 USD per round whereas flying a drone for 1 week to achieve a similar city coverage costs around 500 USD. Clearly, both the financial and time cost of drone imagery is much lower without sacrificing performance. We recognize, however, that the survey data in the case of Dar has been collected to answer transportation policy questions separate from rent predictions, hence we recognize the comparison is not fully valid here as the type of questions the two data sets are asking are not the same and survey data maybe be the only option for certain types of questions. On the other hand, once the research question is well-defined, drone images might still be a feasible more cost-efficient alternative.

Second, the absolute performance of all models is very poor. For the pixel and Deep Feature models however, we interpret such performance as a lower bound. The power of deep learning models comes from the training of an extremely large data set and benefiting from a dense network structure. Due to data availability and computational power limitations, we were unable to conduct such more complex analysis. If we were able to perform those, we would hypothesize that the performance of the Deep Feature model should be much better compared to the survey and raw pixel models as we are capturing tangible and intangible feature of both the house and the neighborhood in an efficient and parsimonious way. In addition, we believe that the Deep Feature model should perform best as it would be based on a pre-trained model for image recognition and as such he deep features would represent a more disciplined input to the prediction model compared to the raw pixels applied to an MLP directly.

6. Future Work

Our finding that survey and Deep Feature models lead to the same performance is promising. For policy use, however, using a Deep Feature model would only make sense if we are able to achieve much lower RMSE. To do so, we aim to do the following:

- Leverage weak-supervision to enlarge our label set of housing rent based on images only without the need to get rent from surveys. We should note that 80% of our recorded survey rent was only estimated rent as the occupants were either home-owners or living with relatives, only 20% were actual renters. Hence, for the weak-supervision and creation of labeling function we would mainly focus on the 20% subsample to get more labelling precision.

- Perform further Data Augmentation to expand our Training Data.

- Incorporate information from bigger radii around the house to get more information on neighborhood characteristics.

- The World Bank has conducted a survey in 2019 and has shown interest in our work, so we hope that we can move forward in the aforementioned directions

7. References

- Bency, Archith J., et al. "Beyond spatial auto-regressive models: Predicting housing prices with satellite imagery." 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2017.

- Henderson, J. Vernon, Adam Storeygard, and David N. Weil. "Measuring economic growth from outer space." American Economic Review 102.2 (2012): 994-1028.

- Jean, Neal, et al. "Combining satellite imagery and machine learning to predict poverty." Science 353.6301 (2016): 790-794.

- Law, Stephen, Brooks Paige, and Chris Russell. "Take a look around: using street view and satellite images to estimate house prices." ACM Transactions on Intelligent Systems and Technology (TIST) 10.5 (2019): 54.

- Marx, Benjamin, Thomas M. Stoker, and Tavneet Suri. "There is no free house: Ethnic patronage in a Kenyan slum." American Economic Journal: Applied Economics (2015).Bency, Archith J., et al. "Beyond spatial auto-regressive models: Predicting housing prices with satellite imagery." 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2017.

- Python packages and libraries such as *rasterio*, *TensorFlow*, *keras*, *scikit-learn* and typical libraries such as *numpy*, *pandas*, etc.

- All the computationally intensive code-running was done in Google Cloud Platform.

Link to our code

Google Drive:

`https://drive.google.com/open?id=1wBB8p2Hkxk5l_nZEz2jan5H6iQ7mT3OL`

Dropbox:

`https://www.dropbox.com/s/znyfzpqg5q3khmt/Code.zip?dl=0`