**Using aerial parcel imagery to predict single family home sale prices**

Russell Richie  
MindCORE and Cognitive Science

drrichie@sas.upenn.edu

University of Pennsylvania

**Introduction**

Scalable, accurate, and unbiased valuation of land is essential for tax assessment, predicting (re)development potential, and more. In developed urban areas, vacant land sales – the clearest signal of land value – can be rare, necessitating statistical techniques for separating improved property sales (e.g., single family homes, SFH’s) into improvement values and land values.

For example, Kolbe et al. (2019) infer a land value gradient as follows. First, linearly regress (*y*) neighboring properties’ differences in sale prices onto (*X*) the difference in their improved characteristics, to estimate coefficients for the latter. Second, subtract predicted improved values from sales prices to infer residual land values. Third, interpolate between transacted properties’ residual land values, to infer land values for untransacted properties. I have applied this method to >50K recent Philadelphia SFH sales collected from the Office of Property Assessment, and explain >75% of variation in out of sample vacant land sales.

To my knowledge, work with such techniques has only exploited structured information about a property (e.g., number of bedrooms). However, unstructured information, e.g., aerial footage of Philadelphia, could contain useful information about a property and its improvements, like (1) building materials, quality, and style, (2) presence of decks, pools, gardens, and other improvements, and (3) parcel shape. Such information is only partially encoded in OPA’s structured data. For example, although the structured data have a categorical ‘parcel shape’ variable, it only has levels for irregular, grossly irregular, triangular, rectangular, and right of way (a long narrow parcel of land). Of course, ‘irregular’ and ‘grossly irregular’ could describe many different shapes, for example:

A aerial view of a house

Description automatically generated

It stands to reason that certain shapes, potentially more irregular ones like those above, that may make it more difficult to maximize use of the land, sell for less.

At the same time, previous work has suggested that neural networks that process raw remote sensing data, like streetview and aerial/satellite images of neighborhoods (Law et al., 2019) and ground-level images of building interiors and exteriors (Nouriani & Lemke, 2022), outperform models based on structured data alone, when predicting property prices, i.e., the transaction price of land bundled with all improvements affixed to it. This project extends that work, by focusing on what information *aerial imagery* can provide specifically about a *parcel and the improvements on it*, with an eye towards how this information can improve specifically land valuation.

To extend the Kolbe et al. (2019) method to process raw, unstructured images along with structured data, we could employ twin neural networks, as illustrated below. In each (twinned) network, a convolutional neural network would process the aerial image of the parcel, and end in a single node representing the ‘Aerial Quality Index’. This node would then linearly combine with structured data to predict the property’s price. Finally, we could subtract the first twinned network’s predicted price from the other twinned network’s price.

A diagram of a property

Description automatically generated

However, given the complexity of this architecture, it is worth first checking whether the aerial imagery adds any predictive value over the structured data, in predicting a *single* property’s price (rather than the difference between neighboring properties, which is necessary for separating property value into land value and improvement value). Thus, using aerial imagery and structured information about parcels in Philadelphia, I compared three models:

1. Structured-Alone: A linear model based on the structured information alone, which includes many continuous and categorical variables about the building, and fixed effects for neighborhood.
2. Image-Alone: A convolutional neural network based on the aerial image alone (orange subgraph of the figure above).
3. Structured+Image: A neural network combining the structured data and aerial image. (blue subgraph of the figure above)

To foreshadow, I find that Structured-Alone is a very strong baseline, predicting 76% of out of sample variance. Image-Alone performed surprisingly (to me) well, explaining 42% of out of sample variance. However, Structured+Image performed no better than Structured-Alone, never explaining more than 75% out of sample variance, suggesting that the aerial images contain no information that is not already contained in the structured data. All materials and code are available upon request by emailing the author.

**Method**

*Data*

I collected a publicly available dataset of single family home sales from Philadelphia’s Office of Property of Assessment[[1]](#footnote-1). The following describes my data filtering pipeline[[2]](#footnote-2):

1. I retained transactions from 2020 to 2023 (inclusive)
2. Filtered out transactions under $10,000 (a heuristic to eliminate non-arms-length sales)
3. Filtered out various zoning categories that don’t contain single family homes (e.g., vacant, or multi-family).
4. Filtered out transactions where the ratio between the OPA assessed market value, and the actual sales price, was below .2 or above 5. This is another heuristic to eliminate non-arms-length sales, or otherwise filter out transactions with questionable recorded sales prices.

This yields 59,522 arms-length single family home sales. The distribution of sales prices is below.

A graph of a graph

Description automatically generated

As can be seen, sales price is distributed approximately log-normally, so I log-transform price before model-fitting.

The following table contains the structured variables collected by OPA that I used for modeling[[3]](#footnote-3).

|  |  |
| --- | --- |
| **Categorical variables** | **Numerical variables** |
| basements | depth |
| central\_air | fireplaces |
| exterior\_condition | frontage |
| garage\_type | garage\_spaces |
| interior\_condition | number\_of\_bedrooms |
| other\_building | number\_of\_bathrooms |
| parcel\_shape | number\_of\_rooms |
| type\_heater | number\_stories |
| year\_built\_estimate | total\_livable\_area |
| building\_code\_description\_new | total\_livable\_area\_squared |
| zoning | year\_built |
| sale\_month | year\_built\_squared |
| neighborhood | sale\_year |

For categorical variables such as zoning category, certain levels of these categorical variables occur too infrequently (<50 times) for a model to reliably estimate their effect. We lump all such levels of a given variable into a new category 'other'. Building code descriptions are written as short phrases. We convert these to a term-document matrix which can be used in modeling. Columns in this matrix which refer to infrequent tokens (occurring < 20 times) are dropped. For numerical variables, we replace all missing values with the median value for that variable. We also create a new variable indicating whether a value was missing for that variable (e.g., depth\_is\_na would be true any time a home was missing a measurement of its depth). For variables where a value of 0 is impossible or incredibly improbable (total livable area, year built, number of stories, number of rooms), we replace 0's with the median value. All categorical variables were one-hot-encoded for modeling.

This sales data contains only latitude and longitude of each parcel, so I also obtained a publicly available dataset of Philadelphia parcel vector data[[4]](#footnote-4). I used this to cut aerial image rasters[[5]](#footnote-5) for each parcel in my filtered SFH sales data, to obtain a single image for each parcel, containing all and only the parts of that parcel visible from overhead.

The appendix contains aerial images of the parcels for the 10 most and 10 least expensive properties. Aside from parcel size and maybe amount of ‘green space’ on the parcel, it’s unclear to me what properties of the aerial images capture price-relevant information (but I’m not an expert appraiser or aerial imagery consumer).

One complication is that the aerial images of Philadelphia are divided into tiles, and a small number (~5%) of parcels overlap multiple tiles. It is possible to obtain the pieces of each tile containing a parcel, and mosaic these, but owing to time constraints, I have not done that. As far as I can see, the main selection bias introduced by skipping such parcels, is that it omits larger parcels.

The following figure illustrates that there is some variation in parcel sizes, although the dimensions cluster tightly in a few ways: one group is rectangular with height>width, another is rectangular with width>height, and another is roughly square.

A graph of a graph

Description automatically generated with medium confidence

Because parcel images are unequal in size, I pad or crop all images to 500 by 500. As seen in the above figure, <10% of images are larger than that in either dimension.

I also drop the 4th channel in the images, which contains near infrared, because for my CNN I utilize the pre-trained VGG16, which requires 3-channel inputs[[6]](#footnote-6). Finally, I normalize the 8-bit integer image arrays by 255.

I split all transactions into a training set (70%) and a test set (30%). This split is “spatially missing-at-random”, i.e., the training and test sets both contain properties from all parts of the city. I focused on this manner of split because I am most interested in the use case of assessments. Future work can evaluate models (more stringently) by holding out one or more neighborhoods for the test set (as in Law et al., 2019, i.a.).

*Models*

The Structured-Alone model is a linear model of all structured variables. Image-Alone and Structured+Image utilize a convolutional neural network with the following layers:

1. Average Pooling2d, with pool\_size=(2,2)[[7]](#footnote-7)
2. Mask 0 values. This masks all values in the image array that are outside the parcel.
3. Batch normalize. (This and the previous step were my attempt to efficiently standardize images while ignoring values outside the parcel.)
4. VGG16
5. Flatten
6. Dense(50, relu)
7. Dropout(.5)
8. Dense(1, linear)

In Image-Alone, this linear output node is the prediction of price. In Structured+Image, the output node of the image network is concatenated with the structured data, and all then feed to a single node predicting price.

In training the image models, I first freeze all weights of VGG16, and train with Adam until validation does not improve for three epochs, using the default learning rate. I then unfreeze all weights and fine-tune with a lower learning rate (1e-5). Image models are trained in batches of 32 samples.

**Results and discussion**

|  |  |
| --- | --- |
| **Model** | **Test set performance (R-squared)** |
| Structured-Alone | 76% |
| Image-Alone | 42% |
| Structured+Image | 75% |

The table above shows performance. Structured-Alone is a very strong baseline, and Image-Alone performed surprisingly well. But Structured+Image performed no better than Structured-Alone, suggesting that the aerial images contain no information that is not already contained in the structured data[[8]](#footnote-8).

It is perhaps not terribly surprising that the image adds no information beyond the structured data. The structured data are highly comprehensive – including the one-hot-encoded variables, there are 293 variables. Just looking impressionistically at the images, as a naïve consumer of aerial imagery, it is often hard to know what relevant information is there. This is why I was surprised that the Image-Alone model performed as well as it did (R-squared=42%). This performance is especially impressive compared to the aerial-only model in Law et al (2019), which never achieves more than 15% out of sample R-squared in their spatially missing-at-random test in a dataset of London home sales. I suspect my image-alone model outperforms theirs because my model fits only on the image for the parcel, whereas Law et al.’s model fits on a fixed size image of the parcel *and a large, zoomed out view of the surrounding neighborhood, without any indication to the model of the boundary between parcel and neighborhood*. I suspect this means that their model has a difficult time finding and exploiting the relevant features of the parcel (cf. location amenities of the neighborhood) -- other tests of mine show that parcel/improvement features account for 60% of the variance in property prices in Philadelphia, and fixed effects for neighborhoods only explain another 15%. Below I consider how an image-based model could potentially combine the strengths of my parcel-centric approach, with the detailed location amenity information provided by aerial imagery of the surrounding neighborhood.

In any case, the surprising strength of my Image-Alone model suggests it may actually be useful in cases where mass appraisal is necessary, but an appraisal entity doesn’t have the (human) resources to collect the extensive amount of structured information the current data set has.

*Next steps*

There are a couple things I could try to improve the Image-Alone and Structured+Image models.

Other directions involve figuring out what information in the structured data is redundantly captured in the aerial images (besides parcel area, I suspect the parcel shapes correlate with location, which in turn correlates strongly with price). One way to do that is train a ‘multi-task’ network that predicts each dimension of the structured data. There is potentially value in building a model that can automate extraction of that information. (I think there may be existing commercial solutions for this, actually). I would need to do more literature review to see what has already been published on this.

I’m also considering trying to fit a network on the aerial image of the parcel and the surrounding neighborhood, but with some additional layer or other architectural element that allows the boundary of the parcel to be incorporated. In BERT, you can use segment ID's and the [SEP] token to tell the network what parts of an input belong to a particular phrase, sentence, or document, which is useful for tasks like relation extraction, document similarity, or textual entailment. This may help the network know where the parcel is, and where the neighborhood is, and thereby separately learn the influence of each, and even their interaction (e.g., the value of a garage is lower in neighborhoods with the MFL visible nearby). It could be interesting to compare this to approaches that just train on aerial imagery of parcel+neighborhood (eg Law et al 2019) without any information provided to the model about where the parcel begins and ends.

This approach is similar to that of Bency et al (2017). They trained CNN’s on various zoom levels of neighborhoods, concatenate the features extracted from all zoom levels, and feed these to non-linear regressors. They find that their image-alone models explain 80% of the variance in London housing sale prices. Their highest zoom level includes the transacted parcel and only the immediately surrounding neighborhood (0.003 sq. kms) and so is perhaps similar to my parcel-only images. However, for applications like land valuation where it is essential to separate property values into parcel (improvement) and neighborhood (land/location) characteristics, it may be preferable to pursue the approach I described above.

**Conclusion**

Property and land valuation are critical tasks for public and private actors, from assessment offices, to developers, to ordinary renters and homebuyers. Here I explored the utility of aerial imagery of a parcel, processed with convolutional neural networks, to predict property values. While a model of the image by itself achieved nontrivial out of sample accuracy (42% R-squared), the image added no predictive power (75%) over structured data alone (76%). Future work can attempt to improve modeling parcel imagery, combine parcel and neighborhood imagery more intelligently, or explore what structured data is in the aerial image.

References

Kolbe, J., Schulz, R., Wersing, M., & Werwatz, A. (2019). Land value appraisal using statistical methods. *Zeitschrift für Immobilienökonomie*, *5*, 131-154.

Law, S., Paige, B., & Russell, C. (2019). Take a look around: using street view and satellite images to estimate house prices. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(5), 1-19.

Liu, Z., Wu, J., Fu, L., Majeed, Y., Feng, Y., Li, R., & Cui, Y. (2019). Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. *IEEE access, 8*, 2327-2336.

Nouriani, A., & Lemke, L. (2022). Vision-based housing price estimation using interior, exterior & satellite images. *Intelligent Systems with Applications*, *14*, 200081.

Appendix

Most expensive parcels

1001537772.npy

A aerial view of a building

Description automatically generated

1001333276.npy

A close-up of a roof

Description automatically generated

1001173925.npy

A high angle view of a building

Description automatically generated

1001079182.npy

A close-up of a building

Description automatically generated

1001173905.npy

A height meter with a rooftop

Description automatically generated with medium confidence

1001173904.npy

A high angle view of a building

Description automatically generated

1001173839.npy

A bird's eye view of a building

Description automatically generated

1001615720.npy

A bird's eye view of a roof

Description automatically generated

1001234143.npy

A aerial view of a house

Description automatically generated

1001163013.npy

A aerial view of a house

Description automatically generated

Least expensive parcels

1001168303.npy

A rectangular object with numbers on the side

Description automatically generated

1001468909.npy

A close-up of a metal object

Description automatically generated

1001299469.npy

A close-up of a person

Description automatically generated

1001477078.npy

A tall white brick wall with a window

Description automatically generated with medium confidence

1001305632.npy

A bird's eye view of a building

Description automatically generated

1001262099.npy

A view of a building from above

Description automatically generated

1001084024.npy

A close-up of a rectangular object

Description automatically generated

1001503399.npy

A close-up of a scale

Description automatically generated

1001452020.npy

A white rectangular object with black background

Description automatically generated

1001318454.npy

A close-up of a white rectangular object

Description automatically generated

1. <https://opendataphilly.org/datasets/philadelphia-properties-and-assessment-history/> [↑](#footnote-ref-1)
2. I also deduplicated transactions occurring at the same parcel, at different times, because I think this caused problems downstream with merging various datasets, but actually I don’t think this should be a problem. I could modify this component, but it shouldn’t alter model performance much. [↑](#footnote-ref-2)
3. A data dictionary for these variables can be found here: <https://metadata.phila.gov/#home/datasetdetails/5543865f20583086178c4ee5/representationdetails/55d624fdad35c7e854cb21a4/> [↑](#footnote-ref-3)
4. <https://www.pasda.psu.edu/uci/DataSummary.aspx?dataset=462> [↑](#footnote-ref-4)
5. <https://www.pasda.psu.edu/uci/DataSummary.aspx?dataset=2052> [↑](#footnote-ref-5)
6. Following Liu et al (2019), I tried adapting VGG16 for a 4th channel by initializing a 4th channel in the first convolutional layer with the average of the first convolutional layer across all three channels. In contrast to Liu et al, this did not improve performance over 3-channel models. [↑](#footnote-ref-6)
7. I also tested downsampling by factors of 4 and 1 (no downsampling). I am finding that omitting downsampling altogether (R2=47% as of writing, while the model continues to train) outperforms downsampling by a factor of 2 (42% in image-alone) and 4 (35% in image-alone). I have yet to test omission of downsampling in the structured+image model. [↑](#footnote-ref-7)
8. Since simultaneously finding weights on structured and unstructured data could be challenging, I also tried first regressing price onto structured variables, obtaining residuals, and then training Image-Alone models on those residuals. These models never achieved test set R-squareds above 0 (but did start to overfit on training data). [↑](#footnote-ref-8)