Using Images as Covariates: Measuring Curb Appeal with Deep Learning*

Ardyn Nordstrom[1], Morgan Nordstrom[2], and Matthew D. Webb[3]

[1] *Carleton University, SPPA*

[2] *Carleton University, Department of Economics*

[3] *Carleton University, Department of Economics*

September 12, 2024

Abstract

This paper details an innovative methodology to integrate image data into traditional econometric models. Motivated by forecasting sales prices of houses, we use deep learning to extract "information" from images. Forecasts from a neural network trained on the encoded data from classifiers results in improved out-of-sample predictive power. We show that image-based forecasts increase the accuracy of hedonic forecasts when encoded features are regarded as additional covariates. The study exemplifies the benefits of interdisciplinary methodologies, merging machine learning and econometrics to harness untapped data sources for more accurate forecasting.

Keywords: Deep learning, machine learning, convolutional neural networks, images, housing

JEL: C45, C89, R3

*Thanks to audience members at the 17th International Conference on Computational and Financial Econometrics, the Canadian Economics Association Annual Conference, and Carleton University. Webb thanks the Social Sciences and Humanities Research Council of Canada (SSHRC grant 435-2021-0396) for financial support. We thank chatGPT for valuable coding and tikz assistance.

1 Introduction

The presence of unobservables is one of the most fundamental challenges in empirical economics research. Deep learning methods can help researchers across disciplines capture some of these unobservables by accessing information embedded in images, videos, and other types of unstructured data. By combining multiple layers of processing, deep learning makes it possible to incorporate this data in standard econometric approaches. These methods have been widely adopted in other disciplines like medicine (e.g., Ker et al., 2017) and biology (e.g., Moen et al., 2019) where image-based data is more common. However, new sources of unstructured data such as satellite images, traffic photos, and photos from social media are now more widely available, presenting new opportunities for disciplines like economics. This raises the question: how can this type of data be used to answer economic questions? In this paper, we attempt to answer this by showing how employing convolutional neural networks can convert unstructured data in images into usable information for economic research.

Given the importance of visual features of housing in purchasing decisions, we assess whether image data can improve housing price predictions that come from standard hedonic analyses. Hedonic pricing models offer a relatively straightforward approach to estimate housing prices using observable characteristics about a house and its location and are widely used in economics. However, the standard hedonic pricing model can overlook important but unobservable attributes that influence how houses are priced.

Consider two hypothetical houses that are similar in terms of their bedrooms, bathrooms, square footage, lot size, features, and neighborhood characteristics. This type of information is traditionally captured in structured real estate data. While two houses may be similar across these features, many other characteristics influence the perceived worth of each property. These kinds of differences are observable when potential buyers look at photos of, or visit, the property but are not captured in structured housing data.

To capture some of these unobservables we use deep learning methods, a subset of machine learning, which combine multiple layers of neural networks to process data. Unlike machine learning methods that economists are more familiar with, deep learning methods do not rely on input data to have a specific representation or structure (Goodfellow et al., 2016). Instead, deep learning can be used to detect the representations that exist within this unstructured data. In the case of images, each layer of a convolutional neural network identifies increasingly comprehensive patterns within the pixels that make up the image until later layers can identify objects and other features in the data (LeCun et al., 2015; Goodfellow et al., 2016).

When convolutional neural networks are used to identify features in unstructured data, these are typically referred to as "architectures". Many pre-trained architectures have been used in the literature. For example, the recent Ludwig and Mullainathan (2024) paper uses "ResNet50," a widely-used architecture that was

trained on the ImageNet dataset containing over ten million photos (He et al., 2016). However, there are many other trained architectures, which are each able to identify different features within a photo. For applications that exclusively use pre-trained architectures to identify features in images, these parts of the architectures are often referred to as "encoders". While most applications to date (e.g. Ludwig and Mullainathan, 2024; Law et al., 2019; Yeh et al., 2020) have used one encoder to identify the features in images, we use multiple encoders. In a recent working paper, Compiani et al. (2023) uses multiple encoders to measure the similarity of photos of consumer products to measure price elasticities, however, this analysis focuses on comparing single encoders separately. To our knowledge, we are the first to use multiple encoders simultaneously. We find the use of multiple encoders can improve the out-of-sample predictive power of econometric models that use image data.

Our work builds on the seminal work by Yeh et al. (2020), who use deep learning based on the ResNet18 encoder (a predecessor to ResNet50) to predict wealth levels of areas using images captured by satellites. Yeh et al. (2020) show that deep learning can be used to measure common economic indicators from publicly available satellite images. Since this data is frequently missing in the African contexts they focus on, this work highlights the utility of using deep learning to capture traditionally unobserved data. Others have since demonstrated how using the unobservables detected using deep learning can improve economic predictions and evaluation research. The use of satellite imagery data is now widely used in economics research to use information about ground conditions from satellite imagery (Donaldson and Storeygard, 2016; Takahashi and Todo, 2017; Asher and Novosad, 2020). For example, a recent related working paper by Han et al. (2024) shows how satellite imagery data can be used to detect tree cover loss in Toronto. They use the information from these images to estimate the impact that tree cover loss has on housing prices, however, they do not rely on deep learning to identify features in the satellite images.

Using a similar approach to Yeh et al. (2020), a recent working paper by Huang et al. (2021) uses the ResNet50 architecture to estimate the quality of housing in areas in Kenya that had recently been exposed to a cash transfer program. They find that using satellite images alone can measure the impact of this kind of intervention. Ludwig and Mullainathan (2024) apply this approach to a different context to use deep learning in questions related to criminal justice. While they develop a framework for how deep learning can be used to identify new hypotheses from unstructured data, they apply their procedure to make predictions about judges' sentencing decisions. By using the encoded features (again using ResNet50) from mugshot images, along with structured information about a defendant, they can explain how a defendant's appearance influences a judge's sentencing decisions. These applications both demonstrate how the features from single encoders can make image data useful in economic research.

Given the importance of visual aspects of properties in determining real estate values, housing has

been used as an application in the computer science literature to test how deep learning can be used to conduct visual content analysis and scene understanding. Law et al. (2019) develop and train two separate architectures to identify features in Google Street View images and satellite images, respectively, in London, UK. These features are used to predict a property's desirability, which they use to predict housing prices. Like in our study, they find that information about a location's desirability can improve the accuracy of predictive house price models. However, unlike Law et al. (2019), we show that these gains can be achieved using multiple well-established, pre-trained encoders that economists are becoming more familiar with, using only a single exterior photo from a property. We also extend the work of Zhao et al. (2019), who used deep learning models to process interior and exterior photos of houses from listings in Australia to come up with a measure of how aesthetically pleasing the photos were. This measure was then used as an input in a model that used multiple machine learning methods to predict housing prices.

This discussion provides an outline for how our current work differs from and contributes to the existing literature. The first contribution is related to our methodology. Existing research that uses deep learning models has so far primarily focused on using one or two prediction methods. In Ludwig and Mullainathan (2024), predictions come from a convolutional neural network. In Law et al. (2019), predictions come from both ordinary least squares and a convoluted model that augments a linear model with the predictions of desirability from a neural network. This was based on work by Peterson and Flanagan (2009), Ahmed and Moustafa (2016), which compare the results of predictions from standard linear hedonic price models to predictions from neural networks. We take a broader approach to compare the predictions from each of the models considered in the literature: ordinary least squares, neural networks, and a two-stage "convoluted" model that first uses a neural network to make predictions that are subsequently used as an input in ordinary least squares. In addition to using multiple methods, to our knowledge, we are also the first to test each type of model using multiple feature encoders, which we find has important implications for how image data is used most effectively.

The second contribution is related to how novel methods can be used to capture unobservables in unstructured data. This is a rapidly growing part of the economics literature. For example, Gorodnichenko et al. (2023), Curti and Kazinnik (2023), and Alexopoulos et al. (2023) have used information from vocal tone, video, and text data to capture unobservables in press conferences from the Federal Reserve, respectively. Similar to how we are trying to incorporate the visual characteristics of a property that contain information about its value, these papers recognize that tone and body language contain information about the message being communicated.[1] The current paper is an essential step to expand the growing literature that shows

1. Specifically, Alexopoulos et al. (2023) uses video, audio, and text from the Federal Reserve Chair's meetings to assess the emotions of the meetings. They find that both stock prices and volatility respond to the Fed Chair's emotional indices.

how this kind of novel data can be used to in standard econometric analyses (e.g. Lehrer and Xie, 2017; Wu and Brynjolfsson, 2015; Compiani et al., 2023; Donaldson and Storeygard, 2016).

We show that information contained in photos of home exteriors can capture previously unobservable information about what affects housing prices. The biggest improvements in prediction accuracy come from using these features from multiple encoders. In the best case, this ultimately improves predictions about house prices by 11.6%. In the following section, we describe the data. Section 3 describes the three distinct methods we use to incorporate information from these images, and then use this to estimate how information about a home's features can predict housing prices. Section 4 summarizes the findings from these three models, and Section 5 concludes with a discussion of the implications for future research.

2 Data

Our property data comes from the Multiple Listing Service (MLS) administered and maintained by the Toronto Regional Real Estate Board. As the MLS is intended to provide Realtors with all the reference information a home buyer might need to compare potential purchases, the list of unique characteristics detailed for each home in this dataset is extensive. Each property details the housing type, number of bedrooms, bathroom and living spaces, the quality of the basement and garage finishings, the presence and type of any HVAC systems, and the list and sale price and dates. For the remainder of this paper, we describe this collection of variables as the property's "attributes."

We started with a sample of 646,171 properties listed and sold through the MLS between January 2000 and December 2020. Most of the City of Toronto is zoned for detached homes intended for single-family occupancy, and consequently, the housing stock in this dataset is primarily single-family detached homes. We deliberately restricted our sample to only these detached homes, excluding apartments, rowhouses, and multiplexes. We further restricted our sample to only include properties sold between December 2018 and February 2020. This time period was chosen to avoid any price shocks caused by the onset of the COVID-19 pandemic, while simultaneously restricting to a relatively short period of time. This left 8,638 properties with available data on household attributes.

Properties in our sample have a relatively wide range of sale prices, with the average selling price being roughly $1.48M (adjusted for inflation to January 2020; logarithmically equal to 14.07). The broad spectrum

Gorodnichenko et al. (2023) analyze Fed Chair's vocal tone during (Federal Open Market Committee) FOMC meetings and find that stock prices increase when the tone is positive. Curti and Kazinnik (2023) analyze videos of FOMC meetings to assess facial expressions and find that negative expressions are priced into financial markets.

of sale values, from a minimum of less than $400,000 to over $13M, suggests our restricted sample of sales still includes a diverse range of properties. The sale price varies by the amenities present on the property. The average home in our dataset has 3.34 bedrooms and 3.06 bathrooms, and as we would expect from the hedonic pricing literature, homes with more bedrooms and bathrooms tend to sell for a higher price.

Two-story homes are the most prevalent, making up 45% of the properties, closely followed by bungalows, accounting for 33%. Other features, like the presence of a finished or walk-out basement (75% of properties), attached garage (48%), central air conditioning (83%) or a basement apartment (11%), were all positively correlated with the sale price, which is also expected from the hedonic literature. We include all these attributes as control variables in some of our analyses below.

We geocoded each property based on its address to its neighborhood and postal code, which we use for spatial clustering in some of our analyses below. Using the address of each property, we also searched for the historic listing for each property on a publicly-accessible sale database. From the listing, we saved the primary image. The images on the publicly accessible sale database are the same images that were included with the property when it was originally listed on the MLS. In most property listings, the primary photo is a "curb appeal" photo highlighting the front of the house. Of the 8,638 properties from the attributes sample, we manually excluded any properties where the primary photo was of the interior or overhead view of the house, or properties without photos. This left a final sample of 7,277 addresses with both image and property characteristic data, which we use for all of the analyses described below.

3 Methods

Our methodology involves first using six well-known encoders to identify features within images. These encoded features are then used to predict housing prices using three types of models: OLS, neural networks, or a hybrid of the two that uses the output from a neural network as an input in OLS. Each model is estimated using each of six neural network architectures, as well as pairwise combinations of each architecture (for example, all features from the ResNet50 encoder are combined with all the features from the VGG16 encoder), and a "tout ensemble" model that uses all of the features from each encoder. This approach allows us to determine whether additional information can be obtained by using more than one encoder to classify an image.

3.1 Turning Images Into Data

An encoder is a neural network that transforms an input image into a set of features (an encoding). This is achieved by passing the input through several layers of the network, where each layer captures and represents

different aspects of the data. Different encoder models trained on the same input dataset may yield different encodings for the same input image because of the architectural differences in the network layers. These structural variations lead to different feature representations, making some models more suitable for specific types of tasks or data than others. For our purposes, we combine and compare the encodings from four contemporary models: ResNet50 (He et al., 2016), VGG16 (Simonyan and Zisserman, 2014), Inception Network (Szegedy et al., 2015), and MobileNet (Howard et al., 2017).

ResNet50 was a significant innovation in convolutional neural network (CNN) encoder design, introducing a novel solution in the form of residual connections (or "skip" connections). These connections allow gradients to flow through the network directly by skipping one or more layers. This architectural innovation has enabled the construction of much deeper networks than were previously feasible without a corresponding decrease in performance due to training difficulties. This encoder architecture — and the feature map predicted by its convolutional layers when the "Home Alone" house was provided as an input image — is illustrated in Figure 1.

Like ResNet50, VGG16, Inception Network (also known as GoogleNet), and MobileNet, each uses its own novel architecture to encode an input image and each architecture's unique approach to processing and extracting features results in different encodings. We used versions of these encoders which had been pre-trained on the ImageNet dataset. ImageNet is a vast collection of over 14 million images from 1,000 different classes, and each encoder is capable of classifying images into each of those classes and giving a confidence score for that classification (He et al., 2016; Simonyan and Zisserman, 2014; Szegedy et al., 2015; Howard et al., 2017).

In addition to encoding the entire photo using pre-trained encoders, we also attempted to encode the individual *features* in each image through panoptic segmentation. Panoptic segmentation is a progressive methodology in computer vision that combines semantic and instance segmentation to categorize every pixel in an image into a class (e.g., "tree", "car", "building"), and distinguish individual instances of objects within the same class.

We made use of two commonly used, pre-trained OneFormer panoptic encoders: COCO (Common Objects in Context) (Lin et al., 2014) and Ade20K (Zhou et al., 2019). The COCO encoder is trained on an extensive range of images featuring complex everyday scenes and provides a rich set of contexts for encoding the features in our property photos. It encompasses diverse object categories, both "stuff" (background features like sky or grass) and "things" (countable objects like signs or vehicles) that provide rich nuance about the features in each property photo. Similarly, the Ade20K is also pre-trained on a large number of images spanning diverse settings and object classes. It is part of the MIT Scene Parsing Benchmark and is used extensively in computer vision and panoptic encoding tasks. Using both of these panoptic encoders allows

us to encode different features for the same property. For example, the COCO dataset has a larger number of classes, but the Ade20k dataset was better able to identify snow. Combining the encodings from both datasets and both the counts and proportion of each feature yields 450 encoded feature variables. Figure 2 illustrates the image-level and panoptic feature encoding of two images of familiar houses from popular culture.

3.2 How Image Data are Used to Predict House Prices

In this paper, we explore several methods for predicting house prices to test whether photos of houses can meaningfully improve predictions of price compared to predictions based only on observable attributes from real estate listings. To determine the most effective strategy, we compare the findings when we use different models (OLS, convolutional neural network (CNN), or a combination of OLS with a preliminary CNN step) and different inputs (images, attributes, or both). For image processing, we employ encoders to convert the data in each image into encoded features as described in section 3.1.

The combination of different models and input types leads to ten unique ways to include attributes and images in models to predict housing prices, which are presented in Figure 3. Models A and B use only the attributes that are available in MLS in either an OLS model (Model A) or a neural network (Model B). Models C and D use only the encoded images in either an OLS model (Model C) or a neural network (Model D). Models E and F use a combination of both images and attributes by directly inputting the attributes and encoded images into an OLS model (Model E) or a neural network (Model F). Since each encoder generates dozens or, in many cases, thousand of features for each image, Models C and E are estimated using penalized OLS (lasso).

Models G through J all use OLS in the final prediction of housing prices. However, each of these models include an additional step that uses a neural network to predict prices based on the attributes or the encoded images. These first-round predictions are then used as an input in an OLS model to come up with a final house price prediction. This intermediate step is similar to the two-step approaches used by Law et al. (2019) and Zhao et al. (2019), and is used to refine the price prediction by incorporating the neural network's output and the listing attributes into a traditional regression framework. Like the earlier models, these two-step models vary based on which inputs are used for each step. The final house price predictions in Models G and H are based on an OLS regression that only include the predicted price from a neural network. The distinction between Models G and H is that in Model G, this predicted price is based on a neural network that includes just the encoded images, while Model H includes the predicted price based on both the encoded images and the attributes. Models I and J again included the predictions from a first-step neural network

that was based on encoded images (Model I) or encoded images and the attributes (Model J), but include the attributes directly in the final OLS regression as well.

This multi-step approach allows us to combine a neural network's ability to handle a great deal of predictors along with the interpretability of OLS. In this way, Models I and J offer an interpretable modification to traditional hedonic pricing models that incorporate data from images. An advantage of this strategy is that it does not require LASSO or other penalized regression methods for variable selection. In our sample, the number of regressors (including the predicted price from the neural network and the MLS attributes) is significantly smaller than the number of observations.

Each of these models is estimated using different sets of encoded features from ResNet50, VGG16, Inception Network, and MobileNet, COCO, and Ade20k. We also estimated each of these models using combinations of features from each pair of encoders (e.g. ResNet50 and VGG16, ResNet50 and Inception Network, ..., COCO and Ade20k, etc.). We also estimate each model using all encoded features, which included the features from each of the six individual encoders. This combination of all encoded features is referred to the "tout ensemble" or "tout" model.

3.3 Assessing the Information Content of the Images

For each of the ten models described above, we consider two different strategies to assess how much information is contained within the image and how the various models described in Section 3.2 capture that information.

The first approach is to simply regress the actual sales price on the predicted sales price from the model. The original sample of 7,277 houses was split in half to form a training and testing set for the image encoding step. To assess the information content in these images, we therefore use only the portion left for image testing, and split this remaining sample in two to form the analysis training and testing sets to assess the performance of the models that use the encoded features. Our main analysis considers two main questions.

Our first question is do images capture useful information about houses? To answer this question, we analyze the relationship between predicted and forecasted house prices for a sample of 1,844 homes in the analysis training sample ("in-sample"(. The model used is simply:

$$\text{price}_h = \alpha + \beta \widetilde{\text{price}_h^{att}} + \gamma MLS_h. \tag{1}$$

Here price_h is the actual sales price, in logs, $\widetilde{\text{price}_h^{att}}$ is the forecasted sales price (also in logs), and MLS_h is the vector of MLS listing attributes. For this analysis, we must restrict our attention to the price forecasts that came from the neural networks used as the first stage in models G through J as the other models include

the hundreds or thousands of individual encoded features, which do not have a meaningful interpretation independently. This is also the primary methodological innovation in the paper, as we show there are many ways in which to use both sets of inputs.

The second question we ask is: can data from images improve our housing price predictions? To answer this question, we look at the out-of-sample forecast accuracy of the ten models and to assess whether adding the image data improves the model's accuracy. We do this by comparing the Mean Squared Errors (MSEs) for each model type, with each set of inputs for all houses in the analysis testing sample. These out-of-sample MSEs are calculated using the following formula:

$$MSE_{m,a} = \frac{1}{N}\sum_{h=1}^{N}(\text{price}_h - \widetilde{\text{price}}_h^{m,a}). \tag{2}$$

Here m indexes a model, and a indexes the set of attributes.[2] We assess out-of-sample forecasts using both the testing set from the testing/training split described above, which divides the analysis sample in half. However, we also consider the out-of-sample performance using 5-fold cross-validation, which allows for a larger set of data to train the model with.

4 Results with an Application to Housing Price Predictions

In this section we discuss the results of the analysis described above.

4.1 Do Images Capture Useful Information about Houses?

Table 1 describes the in-sample results from estimating Equation (1). We present the coefficient associated with the predicted price, $\widetilde{\text{price}}_h^{att}$, for the models that use OLS as the final step to predict house prices (Models G through J). For each model we consider the in-sample relationship that the predicted prices have on actual sale prices, when the CNN-predicted prices were based on the features from each of the single encoders (RNF, CC+CT, AC+AT, IC, VGG, MN) as well as the ensemble model, which incorporates features all encoded features from the single encoders and the pairwise combination of each encoder.[3]

Panel 1 presents the relationship between predicted prices and actual sale prices when the prices were only based on the features identified by the encoder (Model G). We find that for all encoders considered,

2. We additionally conducted an out-of-sample forecasting exercise in results not shown.

3. These estimates are also available for each model using pair-wise combinations of the single encoders, however, the number of encoder combinations makes this impractical to present graphically or in a table. The single-encoder and tout encoder versions give a representative description of the relationship observes using the other combinations.

the predicted prices were significant in explaining the sale price of a home. Importantly, we show that the predicted prices from the CNNs contain useful information even when the predicted prices were generated using only the features identified by the encoders and not the MLS features. However, the differences between the encoders is considerable. The R^2 can be as low as 0.02 or as high as 0.262. The highest R^2 comes from the "tout ensemble" model, which uses all of the encoded features.

Panel 2 shows the estimates from Model H, which adds the MLS attributes to the neural networks when predicting $\tilde{p}$. A similar pattern emerges here, where there is considerable variation in both the R^2 and the coefficients, across the encoders. Again, the model with the highest R^2 is the "tout ensemble" model which uses the features from each single encoder and each pairwise combinations of encoders. Interestingly, the coefficient is higher for several of the individual encoders, but their R^2 values are considerably lower. The addition of the attributes to the networks also increases the model fit and the coefficients compared to the estimates based on the image data alone.

The third panel presents the in-sample estimates from Model I, which changes how the MLS attributes are used. Rather than including them as inputs into the neural network used to predict $\tilde{p}$, they are used as regressors in the OLS estimates. Under this specification, the coefficients on predicted prices decrease significantly. This is to be expected, as there are now many informative covariates in the OLS regressions. However, the coefficients are all still highly statistically significant even after including the MLS attributes. This indicates that the encoded features in the images contain meaningful information beyond what is captured in the MLS data alone. Another major change is that the R^2 values are much higher, which is intuitive given the new covariates. Again, like in Panel 1 the highest R^2 and largest coefficient belong to the "tout ensemble" model.

Finally, the fourth panel presents the in-sample estimates from Model J, which uses the MLS attributes twice: once as a set of inputs for the neural network, and again as a set of OLS regressors. The highest overall R^2 are obtained from this approach, specifically from the "tout ensemble" model. Overall, the change in R^2 from Model I, where there were no attributes in the CNN-prediction step, is minimal. However, the variation in coefficients is even more extreme. No longer are all the coefficients positive and statistically significant. Now, some of the coefficients are negative and some of those are even statistically significant.

Taken together these results suggest that the images do indeed contain information that is relevant to the sales price. Moreover, the use of multiple encoders results in superior estimates of the forecasted prices for all specifications. This improvement is largest when the attributes are not included in the final OLS step. Finally, although adding MLS attributes greatly improves the model fit, the images in isolation appear to have non-trivial explanatory power.

4.2 Can Data from Images Improve our Housing Price Predictions?

To answer this question, we consider whether adding data from the images improves the out-of-sample performance of the models in Figure 3. We compare the prediction accuracy of each model to the accuracy observed when only the attributes are included in the out-of-sample predictions.

Figure 4 describes the out-of-sample mean squared error from each of the models in Figure 3. We present the findings using both the testing set from a 50/50 training-testing set split of the analysis data (grey markers) and 5-fold cross-validation (blue markers). Across the different types of models, the minimum MSE for the models that use OLS only (Models C and E), neural networks only (Models D and F), or OLS with a preliminary CNN step (Models G through J) are 0.0418, 0.1259, and 0.0358, respectively, from the non-CV test set and 0.0361, 0.1796, and 0.0359 when the test set comes from using 5-fold cross-validation. For all cases, the lowest possible MSEs come from the model that uses both images and attributes.

When each model type (penalized OLS, neural network, and OLS with a CNN step) are estimated without input from images, the out-of-sample MSE is 0.0403, 0.381, and 0.0405, respectively. These are indicated by the red lines in Figure 4. This means that adding the images only improved out-of-sample performance for the two-step model, which uses the output from a first step CNN as an input in the final OLS model, and only when both images and attributes are used as inputs. When the encoded image features are used directly in either a penalized OLS model or a neural network, this does not improve the out-of-sample prediction accuracy over what is obtained from the attributes alone.

Figure 4 also shows that in half of all models, the best performing models use the "tout ensemble" encoded features, which include the encoded features from each encoder. These are indicated using a darker square in Figure 4. Here we see that their own, the neural networks have quite high mean squared error rates, as shown in Panel (a). The worst convoluted (with or without attributes) model has a better MSE than the best neural network. The neural network with only MLS attributes and no image data has an MSE of 0.381. When we consider neural networks based only on the image data the mean of the MSEs is 0.665, with considerable dispersion between the models. The best MSE from an image-only neural network comes from the "tout ensemble," which is the only encoder that out performs the attribute-only model (Model B). Interestingly, when we add MLS attributes to the image data the best MSE improves considerably, and the "tout ensemble" model is no longer the winner. Here, the winning model is the pair of MobileNet and Inception encoders. The "tout ensemble" MSE is still respectable at 0.2755, compared to a mean of 0.5857 and a minimum of 0.1259. Conversely, the worst-performing model has a far worse MSE compared to neural networks with no MLS attributes, 5.8715 versus 1.7296.

There is considerably more variation in performance of the NN models across the different encoders when

cross-validation is used, however, the conclusions remain consistent if we consider the findings that use 5-fold cross-validation: the "tout" model performs better than any single or pair of encoders, adding attributes (generally) improves performance, and using only a NN to estimate prices based on encoded features leads to less precise measures than other models that use OLS. This corresponds well to the results in Table 1 where some of the worst models have negative coefficients when the MLS attributes are combined with the image data in forming the neural network forecasted price (Panel 4, Model J).

Given the extreme MSE values that are obtained when the NN-based models are used, Panel B of Figure 4 excludes Models D and F to better visualize the performance of the other models. Here it is obvious that the MSEs from the convoluted model without attributes in either the neural network or the OLS stage are better than those from the neural network alone. This is particularly telling as these models only have an intercept and a slope on $\tilde{p}$ for the forecasting stage. The mean MSE from the train/test split is 0.1995, compared to the mean neural network MSE of 0.6649. Additionally, if we add MLS attributes to $\tilde{p}$ the MSEs get smaller. This is true both with and without attributes in the regression stage. These differences are small but noticeable, from means of 0.1995 to 0.1684 for images only, and from means of 0.0400 to 0.0392 for images and attributes (for the train/test split).

In half of these cases, we see that the best-performing MSEs are associated with the "tout ensemble" model under the train/test split approach. When 5-fold CV is used, the tout ensemble model is the best performer in 5 of the 6 models in Figure 3 that include images. Here, the best performing MSE for the two-step models comes from the "tout ensemble" model when the MLS attributes are included in the regression but omitted from the neural network. The smallest MSE is 0.0359 when attributes are used only once, versus 0.0443 when used twice. When the train/test split is used, the minimum MSE comes from the two-stage model using attributes twice, with an MSE of 0.0358. This is the best-performing model of any permutation considered, and the change in MSE amounts to a 12% improvement in out-of-sample prediction accuracy, compared to when only attributes are used (when the out-of-sample MSE was 0.0405).

There are two main takeaways from this analysis. The first is that using many encoders results in the best MSEs most of the time. When we use cross-validation, the tout ensemble model performed best in six of the eight models that include images (Models C through J). When we use the train/test split, the tout ensemble model performed best in four of the eight models.

The second is that using the image and attribute data together seems to minimize the MSE. This is visually evident when you compare the left (image-only) and right (images and attributes) graphs in Panel B of Figure 4. The MSEs are consistently lower and less varied after attributes are added. For the penalized OLS strategy, using both resulted in the lowest MSE. The MSE was also lowest for neural networks when using both sets of data as well. A similar result holds for the convoluted models, but as mentioned, the best

result comes from using the MLS data only in the regression stage.

Our intuition on why multiplier encoders work better than a single encoder is as follows. Normally when we think about categorizing data we expect that alternative classifications may be highly correlated with one another. For example, recording educational attainment by either years of education, or as the highest level of degree or diploma. Similarly, something like sentiment analysis might result in one person regarding a statement as "good" while another person might regard it as "positive". In these cases, the additional categorizations will not add much value as the classifications are likely to be strongly correlated with each other.

However, in the case of image encoders the classification categories are largely non-overlapping. This is perhaps surprising, but the number of 'things' in the world is obviously unfathomably large so it wouldn't be feasible to assign a probability to each of them. Thus, one encoder might have 'palace' as a category and another has 'church'. Similarly, one encoder might have 'aviary' while another has 'orchard'. Thus if we combine multiple encoders together we get different assessments about the likelihood that the images belong to certain categories. Seemingly important for our dataset is that none of the encoders that we used contain a category for 'house'. If they all did, we might be back in an environment where each encoder returned essentially the same information – that all images were houses.

5 Conclusions

These findings demonstrate how deep learning can be used to capture information from images that is unobservable in traditional data. We apply this to housing, where visual details are particularly relevant since these details influence buyers' perception of a property. However, these subtle visual details are not captured by the characteristics included in traditional housing data.

Using deep learning we can improve out-of-sample house price predictions using one image of the front of a house. These results are based on an analysis of single-family homes, so the findings may not be directly applicable to other kinds of real estate. However, we believe this paper's main contribution is related to the finding that deep learning and, specifically, the use of multiple encoders can make image data useful in standard econometric analyses.

An important limitation of the current work is that we are only focused on predicting house prices. We do not attempt to identify any exogenous variations or make any inferences about the causal relationship between the content of these images and housing prices. Other work by Huang et al. (2021) has demonstrated how images from satellites can be used as outcomes in causal analyses, however, we are focused on a different question. Instead, we use data in images as covariates.

This contributes to a growing area in the economics literature that shows how deep learning can be used to capture information that had previously been unobservable in traditional data sources. Unlike previous work, which has primarily focused on the use of one encoder at a time, we show that by using several encoders that use deep learning we can capture more information than would be possible using a single encoder to detect features in image-based data. In our application, using multiple encoders improves the predictive power of most of the models we estimate.

Different encoders, such as Inception Network, ResNet50, MobileNet, and the others used here, are designed with different architectures and are trained on different datasets (Szegedy et al., 2015; He et al., 2016; Howard et al., 2017). For example, Inception Net's architecture is designed to accommodate visual information that varies in scale and complexity, while VGG architecture is particularly well suited to detect differences in texture and detail (Szegedy et al., 2015; Kumar et al., 2022). This means that by including multiple encoders, we can capture more information from these images. In the best case in our setting, this leads to large increases in in-sample model fit and a 12% improvement in out-of-sample prediction accuracy compared to the same method using only attributes. Given the growing interest in using images and other unstructured data in economics, we expect these methods will be beneficial to a broad set of applications.

References

Ahmed, Eman, and Mohamed Moustafa (2016) 'House price estimation from visual and textual features.' *arXiv preprint arXiv:1609.08399*

Alexopoulos, Michelle, Xinfen Han, Oleksiy Kryvtsov, and Xu Zhang (2023) 'More than words: Fed chairs' communication during congressional testimonies.' *Journal of Monetary Economics*

Asher, Sam, and Paul Novosad (2020) 'Rural roads and local economic development.' *American economic review* 110(3), 797–823

Compiani, Giovanni, Ilya Morozov, and Stephan Seiler (2023) 'Demand estimation with text and image data.' *SSRN Working Papers*

Curti, Filippo, and Sophia Kazinnik (2023) 'Let's face it: Quantifying the impact of nonverbal communication in fomc press conferences.' *Journal of Monetary Economics* 139, 110–126

Donaldson, Dave, and Adam Storeygard (2016) 'The view from above: Applications of satellite data in economics.' *Journal of Economic Perspectives* 30(4), 171–198

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016) *Deep learning* (MIT press)

Gorodnichenko, Yuriy, Tho Pham, and Oleksandr Talavera (2023) 'The voice of monetary policy.' *American Economic Review* 113(2), 548–84

Han, Lu, Stephan Heblich, Christopher Timmins, and Yanos Zylberberg (2024) 'Cool cities: The value of urban trees.' Technical Report, National Bureau of Economic Research

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016) 'Deep residual learning for image recognition.' In 'Proceedings of the IEEE conference on computer vision and pattern recognition' pp. 770–778

Howard, Andrew G, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam (2017) 'Mobilenets: Efficient convolutional neural networks for mobile vision applications.' *arXiv preprint arXiv:1704.04861*

Huang, Luna Yue, Solomon M Hsiang, and Marco Gonzalez-Navarro (2021) 'Using satellite imagery and deep learning to evaluate the impact of anti-poverty programs.' Technical Report, National Bureau of Economic Research

Ker, Justin, Lipo Wang, Jai Rao, and Tchoyoson Lim (2017) 'Deep learning applications in medical image analysis.' *Ieee Access* 6, 9375–9389

Kumar, Sanjeev, Suneet K Gupta, Manjit Kaur, and Umesh Gupta (2022) 'Vi-net: A hybrid deep convolutional neural network using vgg and inception v3 model for copy-move forgery classification.' *Journal of Visual Communication and Image Representation* 89, 103644

Law, Stephen, Brooks Paige, and Chris Russell (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

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015) 'Deep learning.' *nature* 521(7553), 436–444

Lehrer, Steven, and Tian Xie (2017) 'Box Office Buzz: Does Social Media Data Steal the Show from Model Uncertainty When Forecasting for Hollywood?' *The Review of Economics and Statistics* 99(5), 749–755

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014) 'Microsoft coco: Common objects in context.' In 'Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13' Springer pp. 740–755

Ludwig, Jens, and Sendhil Mullainathan (2024) 'Machine Learning as a Tool for Hypothesis Generation*.' *The Quarterly Journal of Economics* p. qjad055

Moen, Erick, Dylan Bannon, Takamasa Kudo, William Graf, Markus Covert, and David Van Valen (2019) 'Deep learning for cellular image analysis.' *Nature methods* 16(12), 1233–1246

Peterson, Steven, and Albert Flanagan (2009) 'Neural network hedonic pricing models in mass real estate appraisal.' *Journal of real estate research* 31(2), 147–164

Simonyan, Karen, and Andrew Zisserman (2014) 'Very deep convolutional networks for large-scale image recognition.' *arXiv preprint arXiv:1409.1556*

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015) 'Going deeper with convolutions.' In 'Proceedings of the IEEE conference on computer vision and pattern recognition' pp. 1–9

Takahashi, Ryo, and Yasuyuki Todo (2017) 'Coffee certification and forest quality: evidence from a wild coffee forest in ethiopia.' *World Development* 92, 158–166

Wu, Lynn, and Erik Brynjolfsson (2015) 'The future of prediction: How google searches foreshadow housing prices and sales.' In 'Economic analysis of the digital economy' (University of Chicago Press) pp. 89–118

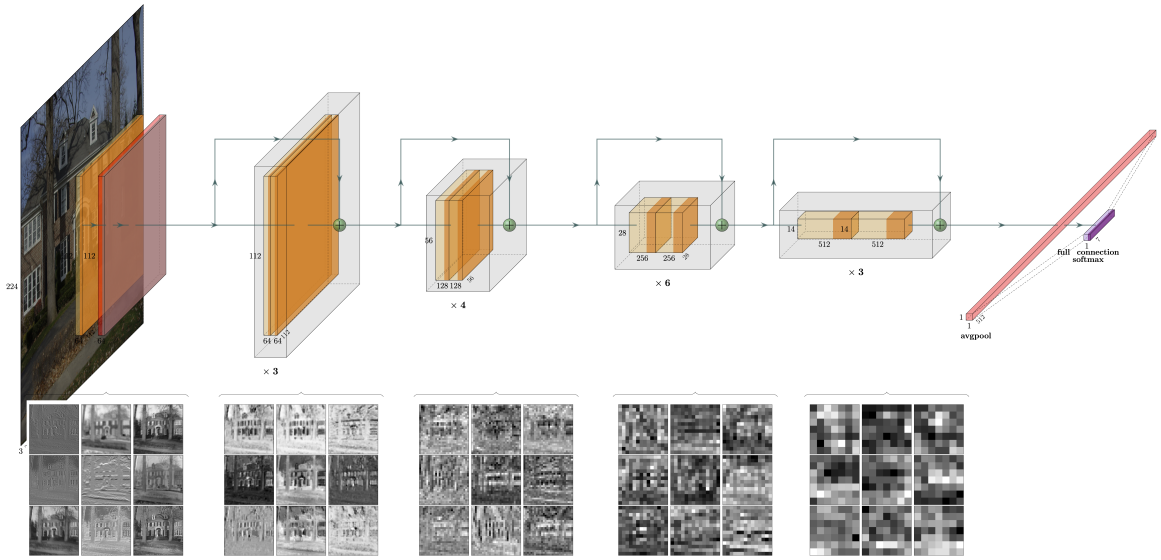
Yeh, Christopher, Anthony Perez, Anne Driscoll, George Azzari, Zhongyi Tang, David Lobell, Stefano Ermon, and Marshall Burke (2020) 'Using publicly available satellite imagery and deep learning to understand economic well-being in africa.' *Nature communications* 11(1), 2583

Zhao, Yun, Girija Chetty, and Dat Tran (2019) 'Deep learning with xgboost for real estate appraisal.' In '2019 IEEE symposium series on computational intelligence (SSCI)' IEEE pp. 1396–1401

Zhou, Bolei, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba (2019) 'Semantic understanding of scenes through the ade20k dataset.' *International Journal of Computer Vision* 127, 302–321

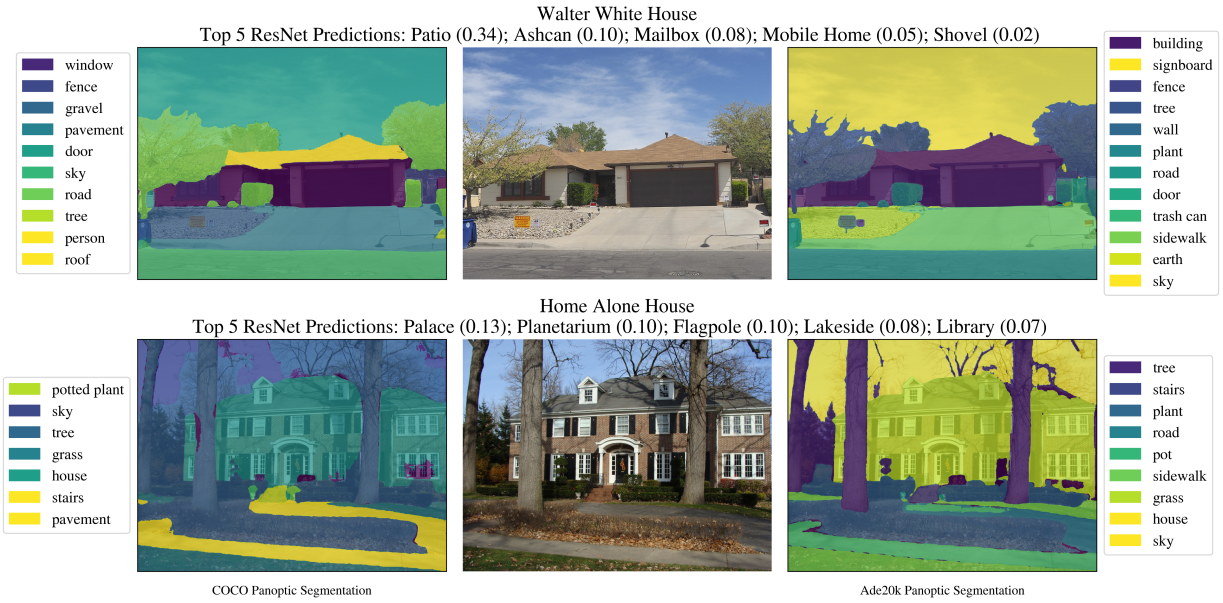
A Tables and Figures

Figure 1: ResNet50 Architecture with Convolution-Layer Predictions



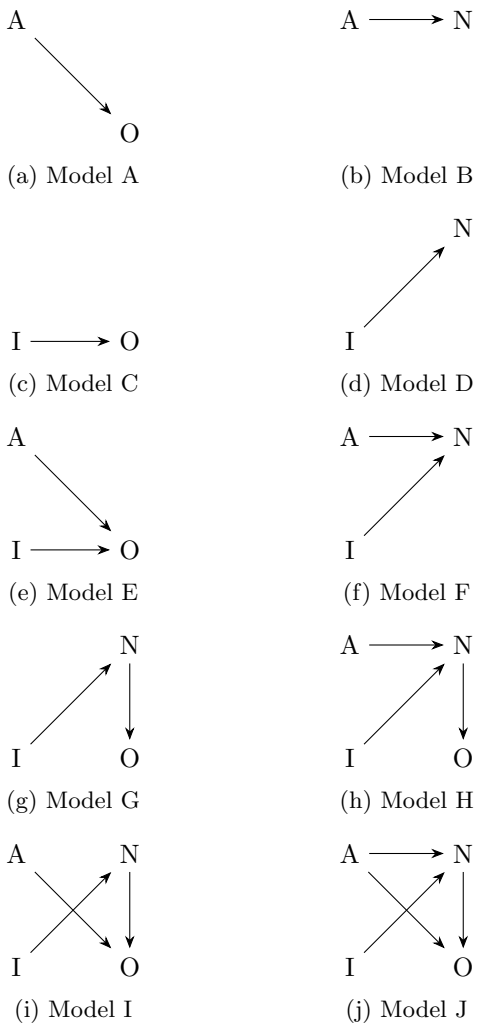
Note: This figures illustrates the steps in a convolutional neural network (CNN) encoder. The connections within the CNN allow gradients to flow through the network directly by skipping one or more layers. These layers are demonstrated using the image of the house from the film "Home Alone".

Figure 2: Examples of Encoded Images and Features Within Images



Note: These diagrams illustrate the panoptic feature encoding of two images of familiar houses from popular culture.

Figure 3: 10 Ways to Include Attributes and Images



(a) Model A (b) Model B

(c) Model C (d) Model D

(e) Model E (f) Model F

(g) Model G (h) Model H

(i) Model I (j) Model J

Legend:

A - attributes

I - (encoded) images

N - neural network

O - OLS model

Note: This figure describes the different models used to test the performance of different approaches to incorporate images in predictions of house prices. We use different estimation methods (OLS, convolutional neural network (CNN), or a combination of OLS with a preliminary CNN step) and different inputs (images, attributes, or both). Note that the images are always encoded before they are fed into the neural network or ordinary least squares (OLS) model. This means that each model is implemented 32 times, once for each single encoder, pairwise combination of encoders, or the "tout ensemble" combination, which uses all encoded features. These encoders yield dozens, and in some cases, thousands, of features for each image. Therefore, Models C and E are estimated using penalized OLS, specifically lasso.

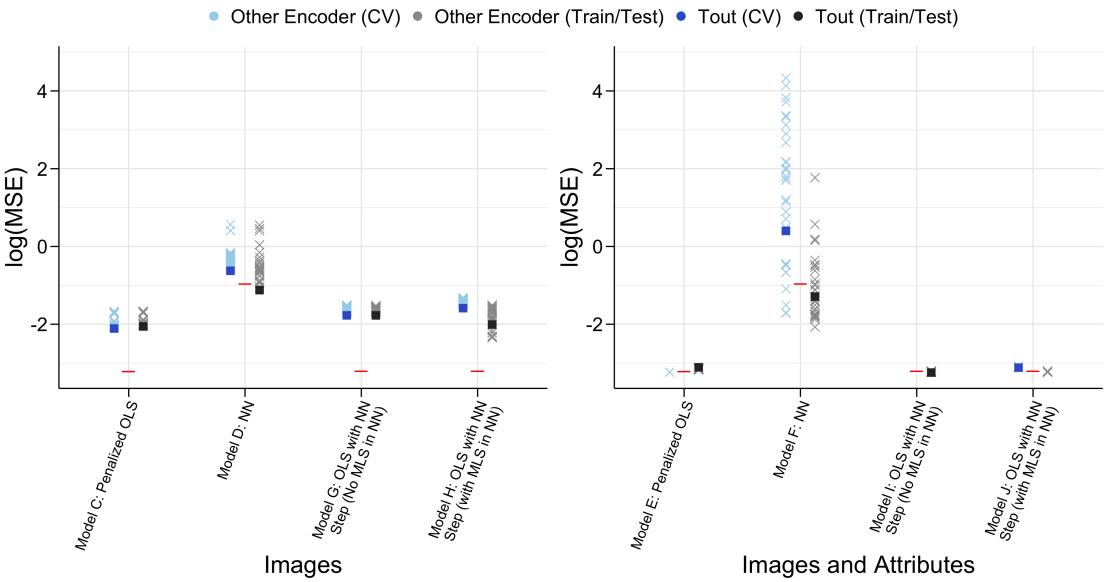
Table 1: In-Sample Price Predictions

	Feature Encoder						
	(1) RNF	(2) CC+CT	(3) AC+AT	(4) IC	(5) VGG	(6) MN	(7) Tout
Panel 1—Model G: No Attributes in Model, No Attributes in CNN Step							
$\tilde{p}_{model}$	0.223***	0.0894***	0.129***	0.199***	0.0526***	0.196***	0.447***
	(0.0238)	(0.0206)	(0.0217)	(0.0234)	(0.00917)	(0.0274)	(0.0445)
Attributes	-	-	-	-	-	-	-
Observations	1,844	1,844	1,844	1,844	1,844	1,844	1,844
R-squared	0.112	0.023	0.046	0.110	0.021	0.090	0.262
Panel 2—Model H: No Attributes in Model, Attributes in CNN Step							
$\tilde{p}_{model}$	0.998***	0.514***	0.966***	0.876***	0.0901***	0.514***	0.639***
	(0.0665)	(0.0817)	(0.0765)	(0.0635)	(0.0177)	(0.0970)	(0.0342)
Attributes	-	-	-	-	-	-	-
Observations	1,844	1,844	1,844	1,844	1,844	1,844	1,844
R-squared	0.339	0.164	0.256	0.285	0.024	0.042	0.424
Panel 3—Model I: Attributes in Model, No Attributes in CNN Step							
$p(\tilde{noatt})_{model}$	0.0309***	0.00423	0.0133**	0.0249***	0.00925**	0.0278***	0.0684***
	(0.00735)	(0.00649)	(0.00620)	(0.00729)	(0.00362)	(0.00788)	(0.0144)
Attributes	✓	✓	✓	✓	✓	✓	✓
Observations	1,844	1,844	1,844	1,844	1,844	1,844	1,844
R-squared	0.872	0.871	0.871	0.872	0.871	0.872	0.875
Panel 4—Model J: Attributes in Model, Attributes in CNN Step							
$p(\tilde{w/att})_{model}$	0.0867***	0.0758***	0.111***	0.0263	-0.0158**	-0.0405	0.118***
	(0.0282)	(0.0187)	(0.0293)	(0.0331)	(0.00758)	(0.0345)	(0.0165)
Attributes	✓	✓	✓	✓	✓	✓	✓
Observations	1,844	1,844	1,844	1,844	1,844	1,844	1,844
R-squared	0.872	0.873	0.873	0.871	0.871	0.871	0.879

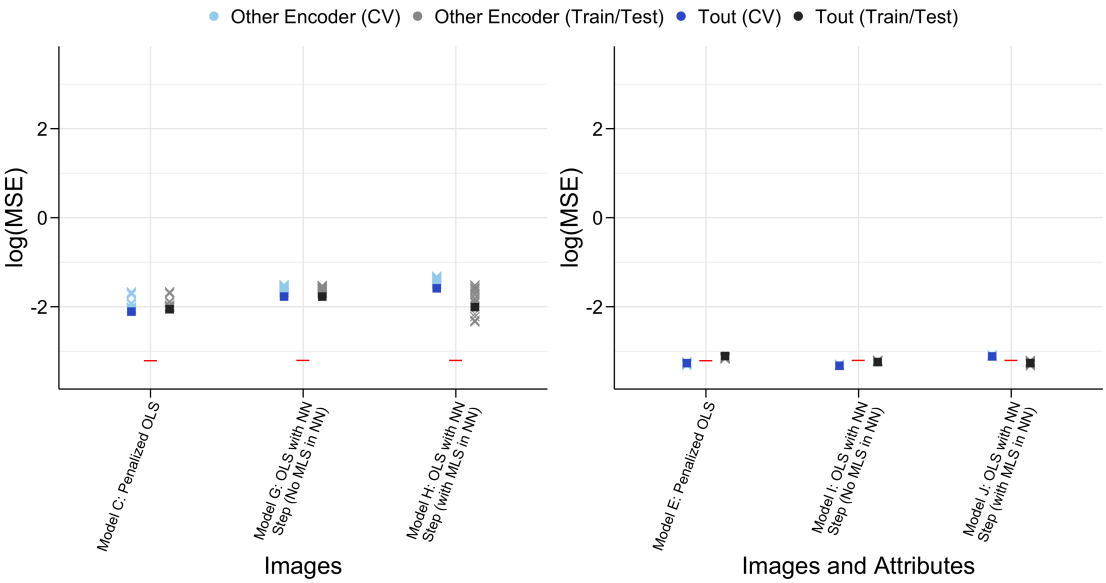
Note: Robust standard errors in parentheses. Clustered at the 3-digit postal code. *** p<0.01, ** p<0.05, * p<0.1. This table presents the in-sample estimates of the value of $p(\tilde{w/att})_{model}$ for the models that use OLS as the final step to predict house prices (Models G through J). This describes the relationship that the predicted prices from the first-stage CNN have on actual sale prices. The CNN-predicted prices are based on the features from each of the single encoders (RNF, CC+CT, AC+AT, IC, VGG, MN) as well as the "tout ensemble" model, which incorporates features all encoded features from all encoders.

Figure 4: Model Performance - With and Without NN Models

(a) MSE Performance - All Models



(b) MSE Performance - No NN



Note: These figures present the out-of-sample mean squared errors (MSEs) for each model, using different types of encoders to process the images. MSEs have been logged to show the variation across models. The grey markers reflect the out-of-sample MSEs from a testing set, which was roughly half the size of the overall analysis sample. The blue markers reflect the out-of-sample MSEs from cross-validation, which allowed for a larger sample size for training. The red markers show the relevant MSE for each estimation method, using only attributes about the house. This figure demonstrates that, in general, the addition of data from images only decreases out-of-sample MSE when a neural network is used as a first step to estimate a sales price (in the case of Models E, I, and J). Moreover, the "tout ensemble" model, which uses the encoded features from all encoders considered in this analysis outperforms other encoders the majority of the time, regardless of which model is used to include these features.