DeepCity: Using Images of Cities to Predict Work-from-home Shocks

Diego Jasson Arjun Ramani Benjamin Wittenbrink
Stanford University
{djasson | aramani3 | witten }@stanford.edu

Abstract

COVID-19 reshaped American housing markets by causing workers with the ability to work-from-home to leave city centers for the suburbs. Economists typically study the drivers of such changes using tabular economic data, like population density. But this misses out on much of urban life that has not been tabulated, such as the presence of skyscrapers or upscale amenities. We propose this context can be visually learned. Using a dataset of over 60,000 images of American cities from Google Street View, we predict post-Covid housing market performance using a variety of neural methods and show three main results. First, a pretrained vision transformer fine-tuned on our data, is able to predict an ordinal post-Covid housing performance with 32% accuracy, which is substantially better than other vision models (for context, randomly picking labels achieves 5% accuracy). Second, saliency maps suggest that our models captures key urban features like skylines, the distance to large buildings, and sidewalk corners. Third, an ensemble model that combines the scores from our vision transformer with tabular economic data outperforms either approach individually, achieving an accuracy of 51%. Our results suggest urban imagery contain unique information relevant to how the pandemic affected housing markets.

1. Introduction

The pandemic caused large-scale outflow of people and economic activity away from city centers in America's largest metro areas. The scale of this outflow was not random. Cities with pricey housing and high densities of skilled tech workers saw larger outflows because such workers had the ability to work remotely. This in turn led to a relative reduction in property values for homes close to dense city centers and a corresponding increase in property further away [18]. Additionally, warmer climates and areas with more space performed exceptionally well [8].

We hypothesize that such characteristics can be visually learned. For instance, skyscrapers indicate office workers and upscale amenities, such as Whole Foods or Equinox gyms, are more frequent in areas with lots of high-skilled workers. To this end, in this paper we develop models to predict the extent of the COVID shock of a city from images of a city's streets. Our exact specification uses Google Street View image data as an input to various models including logistic regression and a two-layer convolutional net – our two baselines – as well neural models like the Vision Transformer (ViT), DenseNet, ResNet, and Virtual Geometry Group (VGG). With these models, we output a class prediction representing a binned range of percent changes in Zillow's home value index.

Our deeper, image-based models perform substantially better than our baseline models. However, all models appear able to distinguish between different neighborhoods to classify changes in home prices. As an additional experiment, we construct an ensemble model to quantify the additional predictive power of image data over traditional tabular economic data. Our results suggest that computer vision methods are able to select features that predict urban housing market dynamics but are hard to measure in conventional economic data.

2. Related Work

2.1. Impact of COVID on Agglomeration

A long-standing literature shows how so-called "agglomeration economies" contribute to city formation. Cities benefit from deeper labour pools which reduces search costs for firms. Ideas flow more easily leading to more firm formation. Firms also face lower coordination costs with their suppliers and buyers. Such dynamics are especially the case for high-skill workers in industries like technology and finance. Indeed Diamond (2016) [4] finds a growing pattern of high-skilled workers concentrating in a few cities over the late 1900s. That in turn led to both higher housing prices, the development of dense commercial districts featuring skyscrapers and the growth of consumption amenities like restaurants and leisure activities like Equinox gyms.

Work-from-home (WFH) may change these dynamics by reducing the cost of coordinating over distance. That has led to large and persistent outflows from city centers in

America's priciest cities [18], especially in those with many WFH-compatible jobs like tech. Other literature shows that worker productivity is surprisingly high under work-from-home [3, 9], which provides further evidence that outflows from city centers should persist. These papers inform our hypothesis that the characteristics of cities that had large outflows are visually learnable.

To predict home price changes, economists will typically run multivariate regressions of the form $y_i = f(X_i)$ where y_i represents the change in home prices and X_i a matrix of characteristics for a zip code i. In our setting X includes population density, the share of WFH-compatible jobs, distance from the central business district and number of business establishments. However, there are many relevant zip code characteristics that are not available in tabular format. To our knowledge there are few widely accessible measures of, say, the number of skyscrapers across cities, or the degree of gentrification as measured by upscale exercise gyms.

Our hypothesis is that rich image data may contain additional information on top of existing tabular characteristics. If true, we should achieve better performance with an ensemble model combining image and tabular data than either model individually.

2.2. Computer Vision and Economics

There is a growing literature applying advances in and techniques from computer vision to economic fields. For example, Glaeser et al. [10] focus on predicting home values from images of homes to identify architecturally distinct features and find aesthetics and appearance play a small but statistically measurable role in property pricing. Naik et al. and Donaldson et al. use computer vision models to document urban change within American cities [17] [6]. Both show that neighborhoods with high education levels and high population density experience improvements in physical appearance. Conversely better physical appearance predicts improvements in economic outcomes.

One component of such physical appearance is architectural style. That is the topic of a recent paper by Lindethal et al. who test the impact of architectural style on property values using various ML techniques [16]. Specifically they classify images using a large deep CNN pre-trained om ImageNet called Inception-v3 and fine-tune it on human-labeled data. They find that rare architectural styles have the biggest impact of price showing that visual information that is unlikely to be captured in standard datasets is an important component of the property market.

Arietta et al. [1] focus on using visual elements from street-level images of American to predict non-visual statistics, such as the crime rate, housing prices and population density. The authors obtain the best results in predicting crime, and compare these results to the predictions from Mechanical Turk workers and find significant increases in accuracy of around 33%. However, the model performance appears to be somewhat city-specific with significantly worse across-city performance.

Recently, Xu et al. [24] study the extent to which subjective and objective measures of street quality explain property values. Subjective assessments come from surveys and objective assessments come from features extracted from street-view imagery. The paper's key finding is that subjective assessments add substantial predictive power to ML models on top of image data. This suggests that though image data is quite successful in predicting variation in property values there is much information it does not contain. The implication for our study is that the tabular economic data may not be fully captured by the image data.

3. Methods

3.1. Models

3.1.1 Baseline

We utilize two sources for our baseline. First, we compare our preferred models with naive, simple alternatives. First we use a logistic regression on non-image data. And second we use a basic two layer convolutional network where each layer corresponds to a block of batch normalization (BN), 3×3 convolution, ReLU, and 2×2 max pooling. This is then inputted to a dropout layer and a final fully connected layer.

Second, we run a multinomial logit using only our tabular data of economic indicators and compare this with our preferred image-based models. We anticipate that this baseline model will perform better than the computer vision specification alone. However, we are principally interested in combining the models on the image and tabular data to form an ensemble method.

3.1.2 Vision Transformer (ViT)

Our main model is a Vision Transformer (ViT), first introduced by [7] as an adaption of the seminal Transformer architecture by [21] from natural language processing. We provide a visual overview of this model in Figure 1 using an illustration from [7].

The standard Transformer is designed for text data and thus accepts a one-dimensional input of word token embeddings. To adapt a Transformer to image data, the ViT is modified to take in a transformed two-dimensional input. In particular, for a three-dimensional image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ where H and W represent the height and width respectively and C the number of channels, ViT flattens it into a sequence of image patches $\boldsymbol{x}_p \in \mathbb{R}^{(HW/P^2) \times (P^2 C)}$ such that P represents the desired resolution of each image patch.

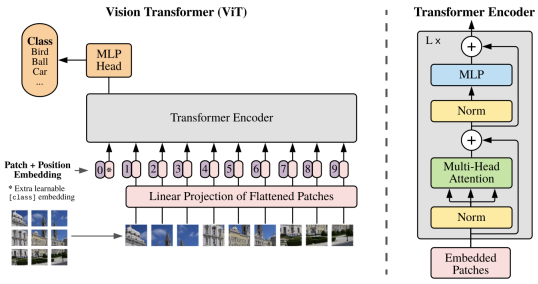

Figure 1. Vision Transformer (ViT) Architecture [7]

Consequently, $N = HW/P^2$. These patches are then flattened using a linear projection into $N \times D$ space to obtain a feature extraction $\boldsymbol{z}_0$:

$$\boldsymbol{z}_0 = [\boldsymbol{x}_{\text{class}}; \boldsymbol{x}_p^1\boldsymbol{E}; \ldots; \boldsymbol{x}_p^N\boldsymbol{E}] + \boldsymbol{E}_{pos} \quad (1)$$

for $\boldsymbol{E} \in \mathbb{R}^{(P^2C)\times D}$ and $\boldsymbol{E}_{pos} \in \mathbb{R}^{(N+1)\times D}$, a matrix of positional encodings. We denote $\boldsymbol{z}_0^0 = \boldsymbol{x}_{\text{class}}$.

This value is then taken as input to the model encoder, consisting of alternating layers of multi-headed self-attention (MSA) and multi-layer perceptron (MLP) blocks, which contain two layers and a Gaussian Error Linear Unit (GELU) activation function each (see Figure A.1). Note

$$GELU(x) = xP(X \leq x) = x\Phi(x) \approx x\sigma(1.702x) \quad (2)$$

where $\Phi(x)$ represents the standard Gaussian CDF [12]. The MSA layer is defined as in [21] using scaled dot product attention, that is for queries $\boldsymbol{q}$, keys $\boldsymbol{k}$, and values $\boldsymbol{v}$:

$$\begin{aligned} [\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}] &= \boldsymbol{z}\boldsymbol{U}_{qkv} & \boldsymbol{U}_{qkv} &\in \mathbb{R}^{D\times 3D_k} \\ \boldsymbol{A} &= \text{softmax}(\frac{\boldsymbol{q}\boldsymbol{k}^T}{\sqrt{D_k}}) & \boldsymbol{A} &\in \mathbb{R}^{N\times N} \\ SA(\boldsymbol{z}) &= \boldsymbol{A}\boldsymbol{v} \end{aligned} \quad (3)$$

where $D_k = D/k$. Intuitively, $\boldsymbol{A}$ captures the pair-wise similarity between elements. Then MSA with k attention heads is just a concatenation of these k outputs linearly projected back into D by a matrix $\boldsymbol{U}_{msa}^{kD_k\times D}$. Before each block, a layernorm (LN) is applied and residual connections are introduced after each block [2, 23]. Thus, for each layer $l \in 1, \ldots L$, the model calculates

$$\begin{aligned} \boldsymbol{z}'_l &= \text{MSA}(\text{LN}(\boldsymbol{z}_{l-1})) + \boldsymbol{z}_{l-1} \\ \boldsymbol{z}_l &= \text{MLP}(\text{LN}(\boldsymbol{z}'_l)) + \boldsymbol{z}'_l. \end{aligned}$$

Finally, to calculate the image representation, a LN is applied to the last class token: $y = \text{LN}(z_L^0)$.

One major benefit of this transformer architecture is that it has significantly less image-specific inductive bias than CNN models. This is because a convolutional layer is local and dependent on the neighborhood structure within the two-dimensional $H \times W$ space (though we have seen that with deep CNNs the overall receptive can be approximately global). In contrast, since MSA is fully connected, these self-attention layers are fully global.

3.1.3 ResNet

In addition to ViT, we use a variety of convolutional methods. The first is ResNet, which presents a residual learning framework to allow deeper neural networks [11]. The main challenge for deep neural networks in computer vision prior to the introduction of ResNet was not overfitting but instead that they were difficult to successfully optimize. To address this, ResNet introduces residual "shortcut" connections, as illustrated in Figure 2.

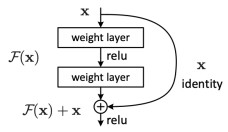

Figure 2. An example of a residual connection [11]

Formally, these connections allow the model to bypass the non-linear transformations by providing an identity function, that is on the l-th layer, we have

$$\boldsymbol{x}_l = F_l(\boldsymbol{x}_{l-1}) + \boldsymbol{x}_{l-1}.$$

If we consider H_l to be the underlying mapping from $\boldsymbol{x}_l$ to $\boldsymbol{x}_{l-1}$, then we are effectively trying to learn the residual, i.e. $F_l(\boldsymbol{x}_{x-1}) = H_l(\boldsymbol{x}_{l-1}) - \boldsymbol{x}_{x-1}$. This is hypothesized to address the issue as solvers oftentimes have issue with many consecutive nonlinear layers. For an indepth discussion of the network architecture used, see [11].

3.1.4 DenseNet

We also use a DenseNet, a model whose signature contribution is that each layer in the network is not only connected to the one before it but to all previous layers [13]. As a result, all of the feature maps from the previous layer are used as in inputs for a given layer. Specifically, the l-th layer in the network is defined as

$$\boldsymbol{x}_l = H_l([\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{l-1}]), \quad (4)$$

where H_l represents a composition of operations. This dense connectivity has led to the model being referred to as a DenseNet.

In their implementation, the authors define H_l as the sequence: batch normalization (BN), ReLU, and a 3×3 convolution. Placed in between these dense blocks are transition layers which correspond to the composition of BN, a

1×1 convolution, and a 2×2 average pooling layer. Generally, these transition layers are also used to downsample the image with a compression factor. Overall, this architecture addresses the vanishing-gradient problem common in deep neural networks and also "strengthens feature propagation" [13].

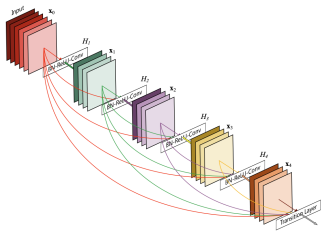

Figure 3. DenseNet architecture [13]

3.1.5 Visual Geometry Group (VGG)

Additionally, we include the Visual Geometry Group CNN, whose main contribution to the computer vision literature – aside from strong performance – revolves around depth and the convolutional filter size [20]. In particular, VGG demonstrated that a neural architecture with relatively small filters that is sufficiently deep actually has superior performance to shallower models using large filters. In particular, with sufficient depth these small filters have an identical receptive field on the original image as the large filters but require fewer parameters. For example, note that a stack of three 3×3 layers have the same receptive field as one 7×7 convolutional layer; however, they only require $3(3C)^2 = 27C^2$ weights as opposed to $(7C)^2 = 49C^2$, which is 81% larger [20].

3.1.6 Additional CNNs

We include two additional CNN-based models, AlexNet and SqueezeNet, during our testing. However, we do not use them on our full specification, and consequently, their description is omitted here in favor of brevity. For more information about these models, see [15] for AlexNet, and [14] for SqueezeNet.

3.1.7 Ensemble

In this model, we combine the image and the tabular data. We do so by taking the the logits from the best performing neural model (the ViT) and appending it to the tabular data before running a multinomial logit. This effectively increases the number of regressors from 4 to 24 because we have one logit for each class label. This enables us to evaluate what residual information is contained in the images that cannot be explained by traditional economic variables and vice versa. Thus, in this component of our technical approach, we are effectively performing an ablation study, where we remove one source of information at a time to uncover the marginal value of each.

3.2. Evaluation Method

We formulate our question as both a classification and a regression task. For classification, we convert our target into an ordinal variable by binning it into twenty classes. This allows us to use multi-class cross entropy loss with mean reduction defined as follows:

$$\begin{aligned} l(x, y) &= \sum_{n=1}^{N} \frac{1}{\sum_{n=1}^{N} w_{y_n}} l_n \\ l_n &= -w_{y_n} \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C} \exp(x_{n,c})} \end{aligned} \tag{5}$$

Here, x is the input, y is the target, w is the weight, N represents the batch size and C the number of classes. We define class weights w_c according to the inverse distribution of classes in our training sample. This ensures meaningful predictions, particularly given the unbalanced nature of our labels.

For the regression task, we utilize a mean-squared error loss (squared L2 norm) defined as follows:

$$l(x, y) = \frac{1}{N} \sum_{n=1}^{N} (x_n - y_n)^2 \tag{6}$$

Here, x is the input, y the target, and N the batch size.

4. Dataset and Features

4.1. Data

We use the University of Central Florida (UCF) Center for Research in Computer Vision's Google Street View dataset [25]. The UCF dataset contains approximately 62,000 images from the Google Street View API and covers the downtown and neighboring districts of three American cities: New York City (specifically Manhattan); Orlando, FL; and Pittsburgh, PA. The spatial coverage of these images is visualized at the zip code level in Figure 4.

Each image has latitude and longitude coordinates and is tagged with its neighbors. For each location in the dataset, five images, each from different viewpoints, are included: four oriented in accord to the cardinal directions and one to the sky[1]. Figure 5 presents an example of these images.

We train all of our models on images taken from the north perspective. We then formulate two separate validation samples, one composed of a subset of these north

[1]There are actually six images for each location as the north view is duplicated.

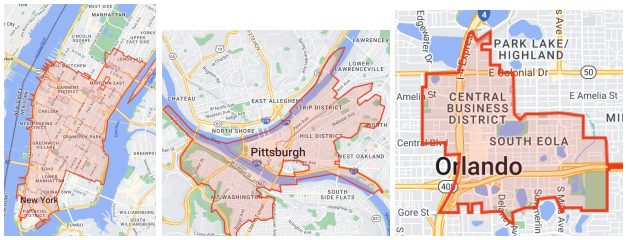

Figure 4. Image Coverage Map of our Dataset

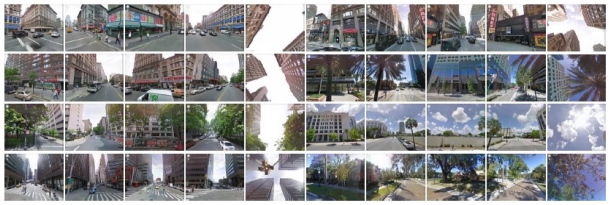

Figure 5. Example images from our dataset [25]

images (hereon "North") viewpoint and one from the east (hereon "East"). We chose not to train on a sample composed of all of the cardinal directions, in order to prevent the model from simply memorizing different images. This is particularly important as the data contains images that are geographically proximate. So it is likely that a validation image from the same viewpoint would be very similar to data the model was trained on. This concern is supported when we compare the validation accuracy from the North and East sample in the results section – accuracy is far higher on the North dataset than the East dataset.

Our target variable is the growth rate in Zillow's home price index [26] between February 2020 and 2022. The variation in this variable across locations is largely dependent on the differential impact of Covid across geographies. Indeed we find a 81% correlation between the post-Covid growth rate and a the difference in growth rates post-Covid vs pre-Covid. This suggests that the variation in housing prices does not follow pre-Covid trends. In addition, we collect traditional economic predictors of home price changes like population density, distance from city center, the share of teleworkable jobs and the number of business establishments. All economic data is collected from the US Census Bureau [19] except for the share of teleworkable jobs which is collected from [5].

4.2. Preprocessing

4.2.1 Images

Before an image is input to the model for training or validation, we apply various transformations. For ViT we use its feature extractor which resizes the image to 224×224 using a bilinear resampling method and then normalizes the pixel values across the RGB channels, each with mean 0.5 and standard deviation 0.5.[2] For all of our CNN models, we cross-validated the performance obtained using the ViT feature extractor versus a manual combination of transforms incorporating randomness – namely `RandomResizedCrop` and `RandomHorizontalFlip` – and found the feature extractor lead to superior performance.[3] Hence, for all models we use the same feature extractor to resample the image to be of size 224×224 and normalize its pixel values.

4.2.2 Target Interpolation

Although the economic indicators that we collect, including the Zillow home value index, are aggregated at the zip code level, the actual images are at a sub-zip code level, as they have latitude and longitude coordinates. To address this, we took an image's coordinates and found the zip code with the minimum Euclidean distance from the image with respect to the zip code's centroid. Our initial dataset consisted of 10,343 locations from 38 zip codes. The distribution of locations over these zip codes is quite skewed – the three most frequent zip codes have 2,631, 1,324 and 681 locations respectively. This raises the concern that the models would learn the image features of specific zip codes rather than the underlying relationship to predict home price changes in general. In response, we smooth out our labels using the following approach, as outlined in [1].

The core of this approach is interpolating a smooth function over latitude and longitude coordinates after fixing the known values of zip code centroids. Our interpolation method takes a weighted sum of radial basis functions to learn the value of any missing points as described in [22]. Here, our function is the inverse multiquadric function

$$f(r) = \frac{1}{\sqrt{1 + (\epsilon r)^2}} \tag{7}$$

where r represents the Euclidean distance between locations. In addition, ϵ is a parameter to control falloff, which we set equal to 2 in accord with [1].

We visualise the distribution of labels after our interpolation below. The distribution is roughly trimodal after interpolation and becomes more balanced with more weight on larger values after interpolating. One reason for this is there are many coordinates towards the edge of large zip codes. Such points would initially be assigned the value associated with a distant centroid. After interpolation, their values would be weighted more towards other nearby zip codes, which more accurately captures the market dynamics of housing.

Upon obtaining the interpolated values, in order to specify a classification problem, we discretize the target variable

[2] Accessed through Huggingface.

[3] See `pytorch` for this implementation.

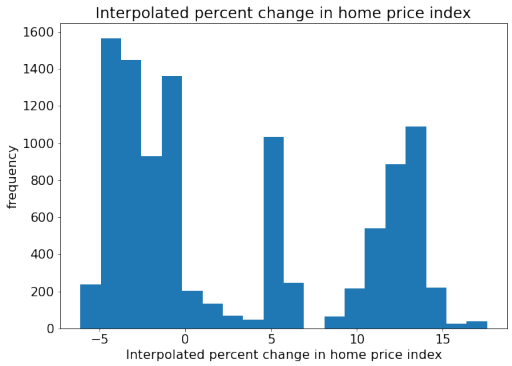

Figure 6. The distribution of home price changes after interpolation

into buckets of one percentage point. For example, a target value of 1.5 % would be placed into the bin $[1, 2]$ and given an associated class label.

5. Experiments/Results/Discussion

5.1. Experimental Details

We run our experiments using the AWS setup provided by the CS 231N course, which means a EC2 G4dn.xlarge instance (4 vCPUs running one NVIDIA 4 GPU with 16 GB of memory). We developed our own Github repository and made use of `pytorch` and Huggingface model implementations. We trained all models with a batch size of 32 and fine-tuned for 10 epochs. We used `AdamW` to set the learning rate with a initial values of `lr=1e-4` and `eps=1e-8`. The learning rate was reduced after hyperparameter tuning to enable more stable updates. We were concerned about overfitting in our image models, and thus chose `AdamW` for our optimizer as it uses weight decay. As the majority of our methods are pretrained, the only hyperparameter tuning we did was for the learning rate as well as for the filter sizes in the baseline convolutional model.

To evaluate our model performance, we report the overall accuracy $\boldsymbol{A}$ for all models. In addition, for the ensemble model we include the precision $\boldsymbol{P}_j$ and the recall $\boldsymbol{R}_j$. These are defined as

$$\mathbf{P}_j = \frac{TP_j}{TP_j + FP_j}, \quad \mathbf{R}_j = \frac{TP_j}{TP_j + FN_j},$$

where j represents a class, and TP_j represents the number of true positives, TN_j true negatives, FP_j false positives, and FN_j false negatives, all for class j.

5.2. Results

5.2.1 Image Results

We first run all of our models on a smaller subset of images with approximately 1,000 unique locations for 10 epochs. This sample results in a binned target variable of seven classes. Note, this is also the sample we tune our hyperparameters on. We chose to perform this analysis on the smaller sample in order to conserve compute and our AWS credits and we additionally hypothesized that the performance on this smaller sample should be representative of the overall performance. The validation accuracy results for the models are displayed in Table 1.

Table 1. Comparison of our models (small sample)

Model	**Validation Accuracy (%)**	
	North	**East**
Baselines		
Random Chance	14.28	14.28
Logistic Regression	51.82	28.27
Two-Layer CNN	71.59	31.40
Image Models		
AlexNet	65.51	30.52
DenseNet	88.07	39.39
ResNet	87.16	38.62
SqueezeNet	52.49	25.68
VGG	79.23	43.32
ViT	**89.02**	**48.92**

As can be seen from Table 1, our top performing models are ViT, DenseNet, ResNet, and VGG, all of which significantly outperform our baseline models. This is true across both validation samples (North and East). Suprisingly, the simple two-layer CNN actually performed better than AlexNet and SqueezeNet. Although we performed parameter search, this suggests that likely some input parameter to these models was suboptimal. Consequently, we do not consider them anymore.

Notably, the performance on the East sample is substantially lower – less than half – of the performance on the North sample. This confirms our concerns that there is likely great similarity between the images from the same viewpoint given their geographic proximity, and consequently, the models are not learning a legitimate mapping from images to home values but are simply "memorizing" the data. This is further supported by poor out-of-sample performance of the North model, omitted for brevity.

We take the four best performing models as well as our baselines and run them on the full sample of data. The full sample of data contains almost 10,000 images and yields a

20 class binning of the target variable. The results on this sample are shown in Table 2.

Table 2. Comparison of our models (full sample)

Model	Validation Accuracy (%)	
	North	East
Baselines		
Random Chance	5	5
Logistic Regression	50.45	14.3
Two-Layer CNN	73.5	15.23
Image Models		
DenseNet	74.78	27.46
ResNet	77.34	29.38
VGG	74.53	26.93
ViT	**90.56**	**32.59**

Interestingly, the top level validation accuracy in this table among both the North and the East samples decreased across almost all models. This is likely explained by the introduction of significantly more classes – in particular, adding the remaining images expanded the scale of the underlying continuous target value substantially. Nonetheless, while the small sample did not perfectly resemble the full sample, the overall results are encouragingly similar.

In Figure 7, we report a confusion matrix using the ViT predictions on the validation set. We normalize the cells by their row – that is, true class – and so, a cell of 1 indicates 100% of the predictions in that row were of that class. The darker areas along the diagonal suggest that our model does successfully learn about the relationship between the city images and the target variable of home prices during the COVID-19 pandemic, as classes are most often classified as themselves or misclassified in a neighboring class. Since our class variable is ordinal, misclassifications that are proximate to the true class are still more "accurate" than those that are far.

5.2.2 Tabular and Ensemble Results

As a comparison for our image-based results, we also run a multinomial logit regression on our tabular economic data of the form $y_i = f(X_i)$ where y_i is one of 20 bins for the home price index percent change and X_i is a vector of four variables: population density, share of workers who can WFH, number of business establishments, and distance from the central business district. We use the 2-D smoothing method to construct interpolated values for all image locations in our dataset. We split our dataset into a training set with two-thirds of observations and a validation set with one-third of observations and achieve a validation accuracy

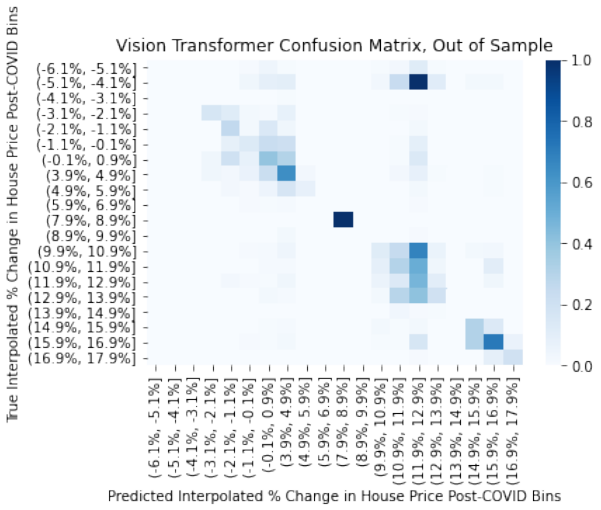

Figure 7. The confusion matrix for the Vision Transformer on East input data. The bins correspond to ranges of interpolated percent change in house prices post-COVID

of 45%. This interestingly outperforms the results from our ViT of 33%. Our ensemble model which combines both the output from our transformer with our tabular economic data outperforms both and achieves 51% accuracy.

These results suggest that image data is not everything – there is a lot of information contained in the tabular economic data relevant to our problem that cannot be learned from our vision models. Still, and perhaps our most interesting result, image data does contain unique information not found in the tabular data. That is shown by the ensemble outperforming the other two models.

Table 3. Ensemble Model Results

Model	Validation Accuracy (%)		
	Weighted avg Precision	Weighted avg Recall	Raw Accuracy
Model Logit	31	45	45
ViT	33	33	33
ViT + Logit	**43**	**51**	**51**

5.3. Saliency Maps

To better understand how our model generates predictions from our input images, we generate saliency maps for a few example inputs using our trained ResNet. Saliency map input images were resized, center cropped, and normalized with the same transformation as validation set images. To generate the saliency map, we perform backprop on the

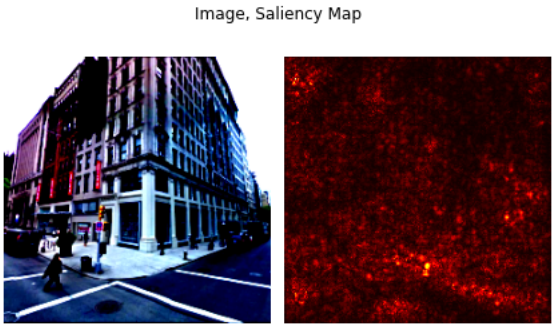

Figure 8. A image and saliency map pair for a street corner in SoHo, New York City

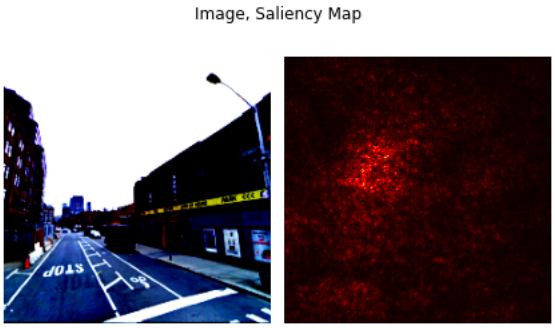

Figure 9. A image and saliency map pair for a street on the lower West Side, New York City

output from the model and plot the absolute magnitude of the gradient.

Our salience maps reveal that the model seems to gauge the height of buildings and takes particular interests in skylines. In 9, for example, we see the model focuses the space above buildings. This pattern of salience suggests that skylines and building height are an important factor in describing the economic agglomeration that occurs in cities. As discussed earlier, taller buildings tend to correlate with office jobs and high skilled labor. Generating a measure of building height and skyline characteristics would be very difficult using traditional econometric techniques. Our salience maps therefore validate that deep learning for this vision task provides valuable information about cities.

In 9, in particular, we focus on the skyline seen in the distance, since it provides spatial information about the photographed west side neighborhood in New York City's proximity to downtown's business district.

If our perspective, however, does not include skyline, then the model must focus on other parts of the image. In 8, we see how a building blocking the skyline in the center of the image pushes the model to focus on the corner of the sidewalk and the skyline visible in the center right and upper left of the image.

5.3.1 Regression

Our question has a more natural formulation as a regression problem, as our outcome variable is continuous. However, we encountered significant difficulty with this approach. Most models struggled to learn and yielded predictions with very little separation. In particular, the predictions were almost always extremely tightly clustered around the mean. Below we discuss some hypotheses for this behavior. L2 loss is a much more unstable and more difficult quantity to optimize than cross entropy loss, which was used for the classification. This makes intuitive sense: the precise values of the predictions do not matter in the classification, so long as the correct class receives the highest score. In contrast, the magnitude of the score is the very quantity we are trying to accurately predict in L2 loss. This makes the L2 loss formulated in the regression problem significantly less robust. Consequently, it seems likely that we have insufficient data and computational power to properly learn the continuous mapping from images to our targets.

Another possibility would be that, rather than truly predicting the change in home price post-COVID shock, our model learns to classify images to some zipcode and then simply bins the image appropriately. If this were the case, we would expect far worse performance in the regression setting. Further, the large decrease in validation accuracy when working with East inputs as opposed to North inputs suggests that we may have overfit some of our models. However, the ordinal approach that we took does not suggest that this is fully the case. In our confusion matrix, we can see our errors down the diagonal, which suggests the model is in fact learning. Since we have large variation in the type of images, however, it may be the case that certain chunks of our data that do not contain useful features (for example, images under bridges, images with major instances of occlusion, etc) are simply classified to a bin, while other images allow for meaningful classification of price change.

Additionally, the position of the sun, how the pavement looks, different standards for scaffolding, bollards, and parking lines may provide the model with significant information about what bin the image belongs to without truly providing much information about the underlying economic features. Some of these features in images, such as the position of the sun, dramatically change with different camera positions, whereas road infrastructure and street signs change across cities. To overcome these pitfalls, future work should seek to employ more data from more metro areas with greater variation in price change and different

learning techniques that may yield better regression outcomes.

6. Conclusion/Future Work

We employ a novel, computer vision approach to understand how the Covid pandemic affected housing market dynamics. By utilizing computer vision techniques, our models are able to improve the prediction of house price changes compared to just using tabular economic data. This shows the value of information contained inside the street view images of the cities.

We find that an efficient vision transformer produces the best results, with 89.02% accuracy validation accuracy facing north (which is likely overfit because our training data faced north, too) and 32.59% accuracy with images that face east (a more robust result). By comparison a multinomial logit trained on tabular economic data achieved 45% accuracy on held-out data. Our ensemble model which combines both the output from our transformer with our tabular economic data achieves 51% accuracy.

One reason the vision transformer worked well was that the transformer was pre-trained. As a result, we were able to extract features more easily from our images. Our models, however, were unable to produce good results in the continuous setting, suggesting future work should seek to improve the model architectures and draw from richer data so that continuous predictions can be made. In addition, future research should consider expanding the sample of cities to be more inclusive of the greater United States to better capture national trends.

References

[1] Sean M Arietta, Alexei A Efros, Ravi Ramamoorthi, and Maneesh Agrawala. City forensics: Using visual elements to predict non-visual city attributes. *IEEE transactions on visualization and computer graphics*, 20(12):2624–2633, 2014. 2, 5

[2] Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*, 2018. 3

[3] Nicholas Bloom, James Liang, John Roberts, and Zhichun Jenny Ying. Does working from home work? evidence from a chinese experiment. *The Quarterly Journal of Economics*, 130(1):165–218, 2015. 2

[4] Rebecca Diamond. The determinants and welfare implications of us workers' diverging location choices by skill: 1980-2000. *American Economic Review*, 106(3):479–524, 2016. 1

[5] Jonathan I Dingel and Brent Neiman. How many jobs can be done at home? *Journal of Public Economics*, 189:104235, 2020. 5

[6] Dave Donaldson and Adam Storeygard. The view from above: Applications of satellite data in economics. *Journal of Economic Perspectives*, 30(4):171–98, 2016. 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[8] The Economist. How the pandemic has changed american homebuyers' preferences. 2022. 1

[9] Natalia Emanuel and Emma Harrington. "working" remotely? *Selection, Treatment, and Market Provision of Remote Work , Harvard Job Market Paper*, 2021. 2

[10] Edward L Glaeser, Michael Scott Kincaid, and Nikhil Naik. Computer vision and real estate: Do looks matter and do incentives determine looks. Technical report, National Bureau of Economic Research, 2018. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[12] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3, 4

[14] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 4

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 4

[16] Thies Lindenthal and Erik B Johnson. Machine learning, architectural styles and property values. *The Journal of Real Estate Finance and Economics*, pages 1–32, 2021. 2

[17] Nikhil Naik, Scott Duke Kominers, Ramesh Raskar, Edward L Glaeser, and César A Hidalgo. Computer vision uncovers predictors of physical urban change. *Proceedings of the National Academy of Sciences*, 114(29):7571–7576, 2017. 2

[18] Arjun Ramani and Nicholas Bloom. *National Bureau of Economic Research*, 2021. 1, 2

[19] Steven Ruggles, Sarah Flood, Sophia Foster, and Ronald Goeken. Jose pacas, megan schouweiler, and matthew sobek. 2021.". *IPUMS USA: Version*, 11, 2021. 5

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3

[22] JG Wang and GRs Liu. On the optimal shape parameters of radial basis functions used for 2-d meshless methods. *Computer methods in applied mechanics and engineering*, 191(23-24):2611–2630, 2002. 5

[23] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019. 3

[24] Xiang Xu, Waishan Qiu, Wenjing Li, Xun Liu, Ziye Zhang, Xiaojiang Li, and Dan Luo. Associations between street-view perceptions and housing prices: Subjective vs. objective measures using computer vision and machine learning techniques. *Remote Sensing*, 14(4):891, 2022. 2

[25] Amir Roshan Zamir and Mubarak Shah. Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1546–1558, 2014. 4, 5

[26] Zillow. Housing data. 2022. 5

A. Appendix

A.1. Packages: Citations

1. `torch` and `torchvision` for pretrained models of AlexNet, DenseNet, ResNet, SqueezeNet, and VGG as well as various utility libraries throughout the pipeline
2. Huggingface `transformers` library for pretrianed model of ViT and the feature extractor
3. `sklearn` for various metric utilities
4. `numpy` and `pandas` for various data manipulation
5. `seaborn` and `matplotlib` for plotting software
6. `rclone` for making the transition to AWS seamless

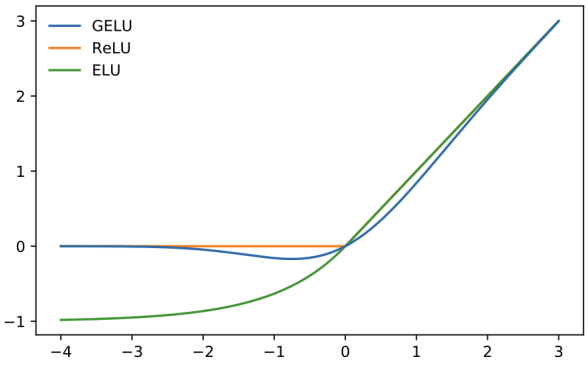

Figure A.1. The GELU with $\mu = 0, \sigma = 1$ compared with ReLU and ELU

A.2. Contributions

1. Diego Jasson: I helped generate the idea for the project, evaluate data sets, download and ingest data, and validate results. In particular, I managed the model visualization and qualitative assessment of model performance, including generating saliency maps and confusion matrices. Along with the rest of the team, I helped generate tables/figures write the paper.
2. Arjun Ramani: I helped frame our problem statement, ingested and cleaned our tabular data, ran our initial baselines, and built/ran our ensemble model. Additionally, I pre-processed our labels with our interpolation techniques. Along with the rest of the team, I helped generate our tables/figures and write the paper.
3. Benjamin Wittenbrink: I contributed to the initial literature review and data collection. In addition, I implemented the majority of the neural models and maintained the data and model pipeline. Like the rest of

my teammates, I helped write the sections of this report, but most heavily in the models, data, and results sections. I also set up and configured the AWS environment.

A.3. Acknowledgements

(a) NA.