Wan, Wayne Xinwei; Lindenthal, Thies

Working Paper

Towards accountability in machine learning applications: A system-testing approach

ZEW Discussion Papers, No. 22-001

Provided in Cooperation with:
ZEW - Leibniz Centre for European Economic Research

Suggested Citation: Wan, Wayne Xinwei; Lindenthal, Thies (2022) : Towards accountability in machine learning applications: A system-testing approach, ZEW Discussion Papers, No. 22-001, ZEW - Leibniz-Zentrum für Europäische Wirtschaftsforschung, Mannheim

This Version is available at:
https://hdl.handle.net/10419/250385

// NO.22-001 | 01/2022

DISCUSSION PAPER

// WAYNE XINWEI WAN AND THIES LINDENTHAL

Towards Accountability in Machine Learning Applications: A System-Testing Approach

Leibniz Association

ZEW

Towards Accountability in Machine Learning Applications: A System-Testing Approach

Wayne Xinwei Wan*, Thies Lindenthal†

January 4, 2022

Abstract

A rapidly expanding universe of technology-focused startups is trying to change and improve the way real estate markets operate. The undisputed predictive power of machine learning (ML) models often plays a crucial role in the 'disruption' of traditional processes. However, an accountability gap prevails: How do the models arrive at their predictions? Do they do what we hope they do – or are corners cut?

Training ML models is a software development process at heart. We suggest to follow a dedicated software testing framework and to verify that the ML model performs as intended. Illustratively, we augment two ML image classifiers with a system testing procedure based on local interpretable model-agnostic explanation (LIME) techniques. Analyzing the classifications sheds light on some of the factors that determine the behavior of the systems.

Keywords: machine learning, accountability gap, computer vision, real estate, urban studies

JEL Codes: C52, R30

*Corresponding Author. Department of Land Economy, The University of Cambridge. Email: xw357@cam.ac.uk

†Department of Land Economy, The University of Cambridge. Email: htl24@cam.ac.uk.

Acknowledgements: This publication is the result of a project sponsored within the scope of the SEEK research programme which was carried out in cooperation between the University of Cambridge and ZEW – Leibniz-Zentrum für Europäische Wirtschaftsforschung GmbH Mannheim, Germany. We thank Peter Buchmann for setting up the infrastructure and providing excellent technical support.

1 Introduction

This paper does not develop any narrowly defined machine learning (ML) wizardry but addresses a fundamental problem of complex prediction systems: How can we verify that a system is performing in the way it is meant to perform? Can we be sure that its outcomes are not spurious or biased?

The triumph of ML applications has only started but has revolutionized commerce, personal interactions, entertainment, medicine, government services, state supervision – and research, already (Simester *et al.*, 2020). In real estate and urban studies, a rapidly expanding literature explores the potential of ML algorithms, introducing novel measurements of the physical environments or using these estimates to improve the traditional real estate valuation and urban planning processes (Glaeser *et al.*, 2018; Johnson *et al.*, 2020; Karimi *et al.*, 2019; Lindenthal & Johnson, 2021; Liu *et al.*, 2017; Rossetti *et al.*, 2019; Schmidt & Lindenthal, 2020; Shen & Ross, 2020). These studies, again and again, demonstrate the undisputed power of ML-systems as prediction machines. Still, it remains difficult for researchers to establish causality or for end-users to understand the internal workings of any models. An "accountability gap" (Adadi & Berrada, 2018) remains: *How* do the models arrive at their prediction results? Can we trust them not to bend rules or to cut corners?

This accountability gap holds back the deployment of ML-enabled systems in real-life situations (Ibrahim *et al.*, 2020; Krause, 2019). If system engineers cannot observe the inner workings of the models, how can they guarantee reliable outcomes? Further, the accountability gap also leads to obvious dangers: Flaws in prediction machines are not easily discernible by classic cross-validation approaches (Ribeiro *et al.*, 2016). More importantly, the opacity of the ML models also gives rise to the legal and ethical concerns for its real-life applications (Mullainathan & Obermeyer, 2017). For instance, anecdotal evidence reports that some ML engines for recruitment have exerted biases again the female applicants (Dastin, 2018). Traditional ML model validation metrics such as the magnitude of prediction errors or F_1-scores can evaluate the models' predictive performance, but they provide limited insights for addressing the accountability gap.

Training ML models is a software development process at heart. We posit that ML system

developers therefore should follow best practices and industry-standards in software testing. Particularly, the *system testing* stage of software test regimes is essential: It verifies whether an integrated system performs the exact function as required in the initial design (Ammann & Offutt, 2016). For ML applications, this system testing stage can help to close the accountability gap and to improve the trustworthiness of the resulting models. After all, thorough system testing has verified that the system is not veering off into dangerous terrain but stays on the pre-defined path. System testing should be conducted before evaluating the model's prediction accuracy, which can be considered as the acceptance testing stage in the software testing framework. Accuracy is not meaningful without verification. Alternatively, the system testing stage could be implemented as a set of constraints imposed on the model during the training phase.

Interpreting the mechanism of ML models in the system testing stage is fundamentally challenging due to the trade-off between the model's interpretability and flexibility (James *et al.*, 2013), especially for the deep learning models (LeCun *et al.*, 2015). Earlier methods include visualizing intermediate activation layers of the input data or the filters in the model (Zeiler & Fergus, 2014), but understanding the visualizations from these methods are difficult for the end-users. In recent years, several up-to-date model interpretation algorithms have been developed, which attempt to reduce the complexity by providing an individual explanation that solely justifies the prediction result for one specific instance (Lei *et al.*, 2018; Lundberg & Lee, 2017; Selvaraju *et al.*, 2017; Ribeiro *et al.*, 2016). This kind of model interpretation technique is also referred to as the local model explanation (Adadi & Berrada, 2018; Molnar, 2019). However, most of the current local interpretation tools are qualitative and require human inspection for each observation. Thus, these tools for model verification do not easily scale up with a large sample.

Examples are often more informative than a long treatise. In this paper, we develop system-testing stages for two ML-enabled use cases: 1) a building vintage classifier and 2) an automatic valuation model (AVM) for residential real estate. Both use cases leverage street-level images of residential real estate as inputs, which are easy to motivate and have been employed in research recently (Law *et al.*, 2019; Lindenthal & Johnson, 2021). In this paper, we do not have an intrinsic

interest in the actual predictions these machines produce but focus on model verification tests: First, we form expectations on relevant information in the images that the models *should* pick up and irrelevant aspects to be ignored, ideally. Then we identify the areas of the input images that are most relevant for the ML models using a local model interpretation algorithm. The final step tests whether the models meet the expectations.

Specifically, we combine expert domain knowledge and an ensemble of ML applications, including computer-vision object detection models, image classifiers, and local interpretable model-agnostic explanation algorithms. We start by asking architects which aspects of a façade they focus on when trying to assess the vintage of a house. They advise that doors, windows, roof shape, building materials, proportions, and ratios, or the existence of garages and driveways are most informative when assessing a house's vintage, while trees or cars should be ignored. Estate agents, however, appreciate additional cues such as trees or (expensive) cars when assessing the value of a home. To them, they are indicators of externalities and a neighborhood's affluence. We expect that well-behaved ML models will arrive at similar priorities when analyzing images of houses.

Next, we augment off-the-shelf image classifiers that have been re-trained to detect architectural styles of English homes (replicating Lindenthal & Johnson, 2021) or the quintile of price regression residuals.[1] Widely available object detection models can reliably locate doors, windows, overall façades, trees, and cars, among many other object categories. Further, we implement the local interpretable model-agnostic explanation algorithm (LIME) – one of the popular local model interpretation tools – to find the areas in the input images that are most relevant for the predictions.

Our results reveal that the ML classifiers lay their eyes on the right things. The vintage classifier indeed focuses on windows and doors and puts less weight on information from the trees and cars. Classifiers that assess home value, however, have a slightly different way of reading the images.

[1] A computer-vision based classifier is selected as an illustration due to its popularity in real estate and urban studies (Naik *et al.*, 2016), although our approach also extends to other ML classifiers, e.g. in text-mining (Ambrose *et al.*, 2020; Fan *et al.*, forthcoming; Shen & Ross, 2020).

First, the computer vision capabilities of AVM we investigate can capture otherwise unobservable hedonic housing characteristics and thereby improve the model's predictive power. To do so, the classifier embedded in the AVM pays attention to informative objects like doors and windows, just like the vintage classifier before. More interestingly, however, the AVM will rely more on indirect cues of value such as cars, just as we hoped. In sum, we can confirm that our ML black box examples do not venture far away from what a true expert would do – that peace of mind is the true contribution of this paper.[2] Overall, we do not aim for internal or external validity in a typically applied economics sense. Instead, we promote a conceptual terminology and offer a proof of concept – an approach often found in engineering or computer science papers.

In addition, we extend the existing qualitative model-interpretation techniques to a formal quantitative test. Methodology-wise, this helps to scale up the model interpretation analyses for a large sample size, which is essential for most of the applications in real estate and urban studies. In summary, our proposed method extends to other ML models and, due to the essence of closing the accountability gap, this study has important implications for ML applications in real estate and urban studies, as well as in other subjects beyond.

Due to the nature of local model interpretation, the design of system testing is specific to the functional requirement of the models. Therefore, we also discuss a few factors that need to be considered when designing a system test. Firstly, we find that, with a larger size of training samples, the system testing scores saturate earlier than the prediction accuracy scores. Secondly, the appropriate size of the selected interpretable information is essential for reaching a reliable system testing score. Lastly, the quality of the input training data (i.e. the noise level) also greatly impacts the model's performance in the system testing.

The rest of the paper is structured as follows. Section 2 reviews the literature that applies ML models, particularly the image-based ML classifiers, in real estate and urban studies. Section 3 describes our proposed system testing approach and suggest a model verification test. Section 4 illustrates the suggested system testing stage with two concrete examples of ML systems, and the

[2]Depending on the use case, additional tests might be advisable for a full due diligence, of course.

system testing results are presented in Section 5. Section 6 discusses some practical factors that impact the performance of the system tests. Section 7 concludes.

2 Literature Review

Over the past few years, the application of ML techniques in evaluating human perceptions of housing quality and urban environment has caught increasing attention in the literature (Aubry *et al.*, 2019; Koch *et al.*, 2019). For example, the studies on the aesthetic value of residential properties demonstrate how the predictive power of ML models improves the traditional real estate valuation process. Before the wide application of ML methods, some literature has provided clues that the aesthetics of different architectural styles carry impacts on the market transaction price (Buitelaar & Schilder, 2017; Coulson & McMillen, 2008; Francke & van de Minne, 2017). The exteriors of a building also introduce externalities that can spill over to the market price of surrounding buildings (Ahlfeldt & Mastro, 2012; Lindenthal, 2017). Nevertheless, a majority of these prior studies using traditional approaches, such as the human assessment by experts or surveys, to measure the subjective perceptions towards the physical housing features (Freybote *et al.*, 2016). The human assessment is normally costly and time-consuming, and it is also threatened by the limited sample size and large bias from unobserved factors. Some other studies use indirect measurements of the building styles, such as the zoning of conserved buildings or the introduction of redevelopment projects, to achieve cleaner institutional settings for the evaluation (Ahlfeldt *et al.*, 2017). Unfortunately, few of these approaches can scale up well.

Emerging literature aims to address these challenges by applying deep learning techniques to classify human perceptions towards housing quality. Utilizing the rich building-level images from Google Street View, Glaeser *et al.* (2018) find that the improvement in building appearance is associated with higher home values in Boston, while the appearance of foreclosed properties depreciates significantly. In the UK, the architectural style is found to be a significant determinant for resale prices, but it has a limited impact on the primary market (Lindenthal & Johnson, 2021).

Law *et al.* (2019) show that street image and satellite image data can capture visual urban qualities and improve the estimation of house prices. By identifying the uniqueness of building vintages relative to the surrounding neighborhoods, Schmidt & Lindenthal (2020) document that the behavior of rounding price is linked to the liquidity and uniqueness of assets. Johnson *et al.* (2020) also confirm the price premium from a property's curb appeal, and they discover that the premium is more pronounced during market downturns. Using ML algorithms to quantify levels of semantic uniqueness in property descriptions, Shen & Ross (2020) find that the uniqueness of properties leads to higher prices and longer listing periods.

Apart from the strength in assessing the quality of individual houses, recent literature has also demonstrated the power of ML methods in urban studies. It illustrates that the ML methods can effectively measure many previously unobserved features of the urban environments (Ibrahim *et al.*, 2020). On the social dimension, several studies reveal that demographics in the neighborhood, including income, race, education, and voting patterns, are associated with and are predictable by the physical appearance and perceived safety of the urban environment (Gebru *et al.*, 2017; Glaeser *et al.*, 2018; Naik *et al.*, 2016, 2017). On the dimension of physical planning, greenery and street-facing windows are found to positively attribute to the perception of safety (De Nadai *et al.*, 2016). Rossetti *et al.* (2019) quantify the quality and human perception of public spaces in urban environments. Zhang *et al.* (2018) propose a framework to represent the locale of street scenes, while Liu *et al.* (2017) evaluate the maintenance quality of building façade and the sense of continuity along the streets in Beijing.

While deploying ML models is getting more popular in the research of housing and urban environment, a vital concern remains: If the end-users do not understand and trust an ML model, they will not use it (Ribeiro *et al.*, 2016). This is especially true when we aim to move one step forward, from the models' predictions to real-life decisions and policymaking (Ibrahim *et al.*, 2020). Unfortunately, it is textbook knowledge that there exists a trade-off between a model's flexibility (i.e. prediction accuracy) and the model's interpretability (James *et al.*, 2013). For the deep learning models that we develop, the nature of complexity hinders end-users from interpreting

them, and the end-users normally just consider the models as "black boxes". As a result, this becomes an essential barrier for promoting policy insights from the models' prediction results.

To bridge this accountability gap, several model interpretation methodologies have been introduced (Adadi & Berrada, 2018; Lei *et al.*, 2018; Lundberg & Lee, 2017; Selvaraju *et al.*, 2017; Ribeiro *et al.*, 2016) in recent years. The intuition behind this is that, although the global decision function of a classification model is very complex and hard to interpretable, justifying the prediction for a specific instance is feasible with fidelity. For instance, Ribeiro *et al.* (2016) propose a novel local interpretable model-agnostic explanation (LIME) system to explain ML classification models with human-understandable representations by approximating the model locally with sparse linear explanations.[3] The image to be classified is firstly divided into several interpretable components (i.e., contiguous super-pixels). Then, by randomly turning some of the super-pixels off, the algorithm generates a set of pseudo instances "near" the original image and tests how the classification result of each pseudo instance deviates from the classification of the original image. Finally, by learning a locally weighted linear model on the pseudo instances, the super-pixels with the highest positive impact on the initial classification are selected as the explanation of the model. More recently, Lundberg & Lee (2017) proposed the Shapley Additive Explanations (SHAP) for the local interpretation of models. Instead of using linear regression as in LIME, this method weights each feature using the Shapley value originated from game theory. It is worth noting that these methodologies are qualitative, and each explanation is only valid for one local instance (i.e., each specific image). One exception is the recent study by Krause (2019), which uses the model-diagnostic method to extract the marginal contribution of each period toward observed prices in the data and then constructs a house price index, accordingly.

In addition to the challenge of low interpretability, the unopened "black box" may also lead to undesirable classifiers, in which the model's flaws are very difficult to be identified if we merely check the accuracy of prediction. Ribeiro *et al.* (2016) demonstrate this modeling issue with an experiment classifying the photos of wolves and huskies (see Appendix Figure A2). They in-

[3] Appendix Figure A1 visualizes the intuition of LIME using a classification function in a 2-D panel. See Ribeiro *et al.* (2016) for detailed discussions.

tentionally choose pictures of wolves that have snow in the background, and pictures of huskies without snow, as the training samples. As a result, the trained classifier predicts "wolves" based on the snow in the background rather than the animals. However, this flaw in the model cannot be easily identified by just reviewing the prediction accuracy, especially if all the images of wolves in the validation test have snow in the background.

As a result, we argue that the standard metrics testing the prediction accuracy of an ML model—such as the F_1 score and the Herfindahl index— are not sufficient to evaluate the model's performance. Lindenthal & Johnson (2021) also discuss this potential misclassification issue, specifically in the context of housing quality and urban environments. They find that the automatically collected street images contain irrelevant information like trees and vehicles for classifying building vintages. Images with larger areas showing the buildings will result in higher classification accuracy. However, there is still no direct evidence showing whether the model has included irrelevant information in the predictions.

To close these research gaps, we extend the qualitative local model-explanation method in the prior literature to a formalized quantitative evaluation method in this study, denoted as a *model verification test*. In addition, using a concept similar to the testing procedures in software development (Ammann & Offutt, 2016), we propose an extended testing framework for the general ML models in real estate and urban studies. Given our model verification test is not restricted to the types of classification problems or the exact ML algorithms chosen, we use a model with deep networks for image classification as our illustrative example, which is one of the most popular ML applications for researches in housing quality and urban environment. Also, we choose to implement one of the commonly used local model interpretation techniques—the LIME algorithm proposed by Ribeiro *et al.* (2016)—in the paper, while the other model local interpretation methodologies like SHAP also apply for our testing approach.

3 System Testing and Model Verification Tests

This section first proposes an abstract framework of system testing that improves the interpretability and due diligence of ML models. Subsequently, it describes a concrete system testing implementation based on a quantitative model verification test suitable for ML models using image data.

3.1 System Testing

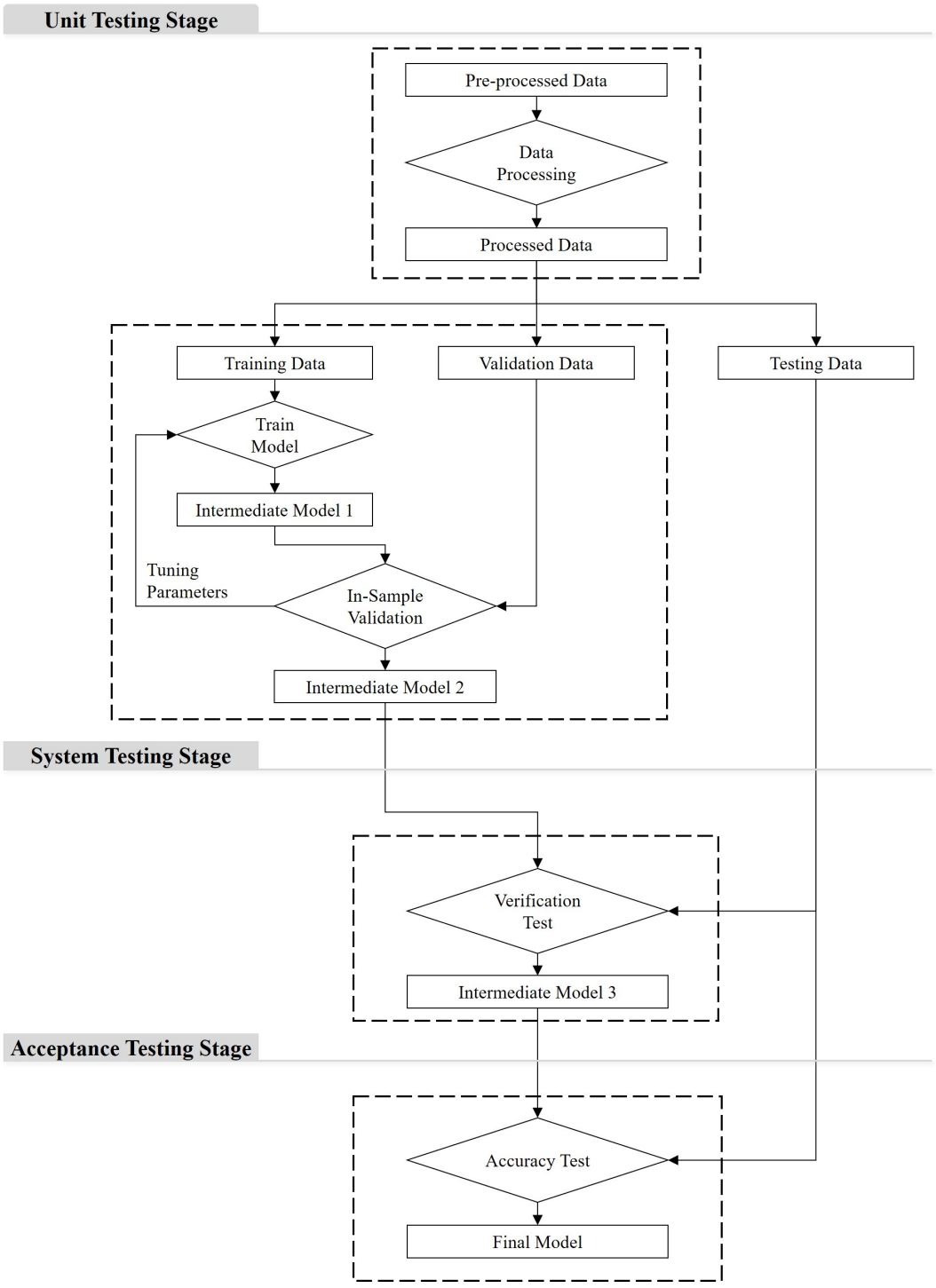
A dedicated testing framework has been widely adopted in software engineering (Ammann & Offutt, 2016): To ensure that functionality and performance meet the pre-specified requirements, the newly developed software usually needs to pass four major stages of testing before deployment. The *unit testing* stage examines the functionality of individual components of the software; the *integration testing* stage examines whether the individual units are well combined; the *system testing* stage checks whether the integrated system meets all end-to-end specifications. In the last step, then the *acceptance testing* stage assesses the performance of the system to ensure acceptance by end-users.

The processes of developing ML models can also be generalized using the same concepts from software development. Data collection and pre-processing can be considered as separate functional components that could be verified with empirical unit tests. Depending on the complexity of the research questions, one or more ML models might be trained, and each trained model can again be considered as a separate functional unit, which can be tested by means of in-sample cross-validation. The frequently-used ML model accuracy tests assess whether the performance (i.e. prediction accuracy) of the system meets the end user's requirements, similar to an acceptance test.

Missing, however, is a system-testing stage to ensure that the integrated system of the functional units performs as intended and that the predictions are obtained reliably.[4] Following best

[4]More accurately, there are more than 50 types of system testing for various purposes. Investigating whether the model performs the exact function as required is more similar to our proposed concept of functional testing, which is a critical one among these system tests. See Ammann & Offutt (2016) for detailed discussions.

Figure 1: The Flow Chart of Machine Learning Process with the Proposed System Testing



Notes: The figure plots the flow chart of our proposed framework for applying machine learning models in real estate and urban studies. Using the concept similar to the hierarchies of system testings, we consider data collection and each trained models as individual units. For simplicity, only one trained model is plotted in this flow chart, and the integration testing between the units is ignored. The classic accuracy test is considered as the acceptance testing of the model. Our proposed model verification test is considered as a system testing to investigate whether the model serves the initial functional purpose as expected, and it is conducted before the acceptance testing.

practice, we argue that system tests of any ML system should be successfully passed before even starting to verify the model's predictive performance.

3.2 Model Verification Test for (some) CV Models

While system testing is a conceptual stage to verify that ML models comply with the design requirements, specific testing routines, so-called *model verification test*, are needed to conduct this stage. Next, we design a novel model verification test and demonstrate how to implement system testing for one of the major applications of ML models in real estate and urban studies – ML image classifiers (Glaeser *et al.*, 2018; Johnson *et al.*, 2020; Liu *et al.*, 2017; Rossetti *et al.*, 2019; Schmidt & Lindenthal, 2020; Lindenthal & Johnson, 2021).

Obviously, a concrete implementation of a system test depends on the specific task at hand, the data used and the type of models trained. Here, we suggest not peak into the inner workings of ML systems directly but instead to analyze whether model outcomes can be traced back to input characteristics. To do so, we leverage relatively new local model interpretation techniques that reveal the elements or aspects of input data that are most decisive for a model's output. We then determine how much of this crucial information overlaps with desirable information, based on human expert input.[5]

More specifically, our model verification test suits the situation when we classify images j into different categories (e.g., classify building into architectural styles, uniqueness levels, price ranges, etc.), and we want to understand whether the ML models have used information of object i (e.g., building façades) for the classification. The model verification test returns two quantitative testing outputs—the verification test score and ratio, which both measure to what extent that the ML model uses information i for the classification results.

Firstly, we calculate the model verification test score ($TestScore_{ij}$) for each object type i in

[5]A fundamentally different alternative, a so-called global model interpretation approach, investigates the functionality of individual components "inside" ML models, e.g., the neurons in a deep convolution neural network model (Olah *et al.*, 2020). However, due to the complexity in global interpretation, these tools normally require strong assumptions on the explainable functionalities (Zeiler & Fergus, 2014). To use another neurological metaphor: The global model interpretation approaches are comparable to MRI scans that try to detect and explain patterns of activity in human brains.

the image j. This score represents how much the model predicts based on the information from object type i in this image j. Technically, we first extract object i in an image j, using the off-the-shelf ML object detection tools. For a specific object i, we denote the associated area in the image as $ObjectArea_{ij}$. There may exist multiple objects in this object type, and we denote the set of all these objects as I. Next, we analyze which areas in the images (i.e., the super-pixels) contribute the most to the model's classification decision, using local interpretation tools such as the LIME algorithm. We call the area of the image j that best explaining the classification result "the interpretable area", which is denoted as $InterpretArea_j$. The verification test score is then calculated as follow:

$$TestScore_{ij} = \frac{(\bigcup_{i \in I} ObjectArea_{ij}) \bigcap InterpretArea_j}{InterpretArea_j}. \tag{1}$$

We find the overlaps between the interpretable areas and the detected object areas of type i. Since the sizes of interpretable areas are not equal in every image, we normalize the overlapped area by the size of the interpretable area. In other words, the test score equals the proportion of the interpretable areas that originate from the object type i. The interpretation is that, if the test score is higher, the model utilizes more information from this object type for its predictions. Therefore, this also marks an advancement in the previous ML interpretation methodology: While the image segments returned by previous methods help to interpret how the model works qualitatively, we propose a quantitative approach to formalize the model verification test.

Secondly, we calculate an additional testing output, named as the verification test ratio. We introduce this alternative testing output because the detected object areas in each image may not be equal. It is noteworthy that if the model just randomly selects any image segments to make predictions, the probability that it captures information from objects in type i should equal the total area of objects in type i divided by the total area of the image. To address this issue, we construct a benchmark for the verification test score of object i in image j:

$$Benchmark_{ij} = \frac{\bigcup_{i \in I} ObjectArea_{ij}}{ImageArea_j}. \tag{2}$$

If the test score is higher than the corresponding benchmark score, it indicates that the model is intentionally capturing information from this object type to reach the prediction results. Accordingly, we calculate the ratio of the verification test score to the benchmark score to further address the different scales of benchmark scores across object types:

$$TestRatio_{ij} = \frac{TestScore_{ij}}{Benchmark_{ij}}. \tag{3}$$

The interpretation is that, if the ratio is larger than one, it implies that the model intentionally extracts information from this type of object for the classification. In contrast, a ratio lower than one implies that the model considers the information from this object irrelevant for the classification.

4 Two Examples

To illustrate how the proposed model verification test helps to interpret and verify the outcomes of ML models, we use two common applications of ML in real estate and urban studies as examples. The first model classifies images of houses, ensembles, or streetscapes into pre-defined categories, all based on street-level image data (Glaeser *et al.*, 2018; Johnson *et al.*, 2020; Liu *et al.*, 2017; Rossetti *et al.*, 2019; Schmidt & Lindenthal, 2020; Lindenthal & Johnson, 2021). The second example is a basic AVM that combines a traditional hedonic model with a computer vision approach, again classifying building image data by property value (similar to, e.g., Ahmed & Moustafa, 2016; Law *et al.*, 2019).

4.1 Example 1: Architectural Style Classification

In the first example, we replicate Lindenthal & Johnson (2021) and classify the architectural styles of residential buildings in Cambridge, UK. The guiding principle for our classifier is to emulate human experts' classifications as closely as possible. This implies it should focus on the same aspects that architects or realtors would pay attention to. The model should...

- ... focus on the façade of the house and ignore the background, sky, gardens, yards, people, or streets;
- ... inspect the brickwork, which again is a good indicator for vintage in Cambridge.
- ... pay attention to doors and windows, as their sizes, locations and styles are correlated with the house's vintage;
- ... ignore trees and cars as much as possible;

While we do not wish to discard subconsciously picked up cues or patterns, our tests can only reflect clear rules. A subjective statement such as "this façade somehow reminds me of a Georgian house I once lived in" might be true but cannot be translated into a test requirement.

4.1.1 The Vintage Classifier

We first replicate the ML vintage classifier in Lindenthal & Johnson (2021) and re-use their extensive image dataset of around 25,000 building images from Cambridge (UK). These images have been collected from Google Street View and classified into architectural styles by architects.[6] To achieve both a clearer differentiation and a more balanced sample size for each category of the architecture style, we classify the samples into seven styles, including the *Georgian*, *Early Victorian* (denoted as *Victorian* for short), *Late Victorian/Edwardian* (denoted as *Edwardian* for short), *Interwar*, *Postwar*, *Contemporary*, and *Revival* style. Appendix C summarizes the definitions and key features of these architectural styles in the UK.

We conduct stratified sampling by architectural styles to construct the dataset for the model's training, in-sample validation, and out-of-sample testing. In total, there are 2,791 buildings selected in our out-of-sample testing dataset. We first use the Inception computer vision models (Szegedy *et al.*, 2016) to obtain 2048 element strong feature vectors for each image. Then we train a deep convolutional neural network model to classify buildings into seven architecture styles,

[6]This sub-sample is far larger than the required sample size to reach the model's saturated training accuracy, but more images are manually tagged with the architectural styles to enable the out-of-sample validation of the model's prediction accuracy. See Lindenthal & Johnson (2021) for a detailed discussion.

which gives us the classification model that we intend to analyze further. The technical details of training the vintage classifier are provided in Appendix D. The out-of-sample classification performance is of secondary interest to us but cross-validation results can be found in Appendix E.

The standard metrics of model prediction accuracy, e.g. precision, recall, F_1 scores, or the Herfindahl index of the scores for each vintage (HHI) indicate that our simple model performs at a satisfactory level. To better understand the way the model classifies the images, we apply the LIME model explanation algorithm following Ribeiro *et al.* (2016) to all images in our test sample.[7] For each image, the algorithm detects the image segments that were most relevant for the model to classify the house into the most probable style. These areas are called *super-pixels*. We limit the number of super-pixel groups the algorithm can identify to a maximum of five.

Finally, our model verification tests all relate to classes of objects that should be in the focus of the classifiers – or not. We automatically detect façades, doors, windows, trees, or cars on all the 2,791 images in the testing sample set. To give an example, Figure 2 presents a selection of objects detected in one randomly selected image.[8] A rectangular mask is drawn tightly around each identified object, and a numerical score is returned denoting the confidence in the detection result. Admittedly, many objects do not have perfectly rectangular shapes. Still, we consider the area within the mask as the relevant image area associated with the specific object. A visual check confirms that our object detection model effectively identifies all objects of interest in our sample reliably.

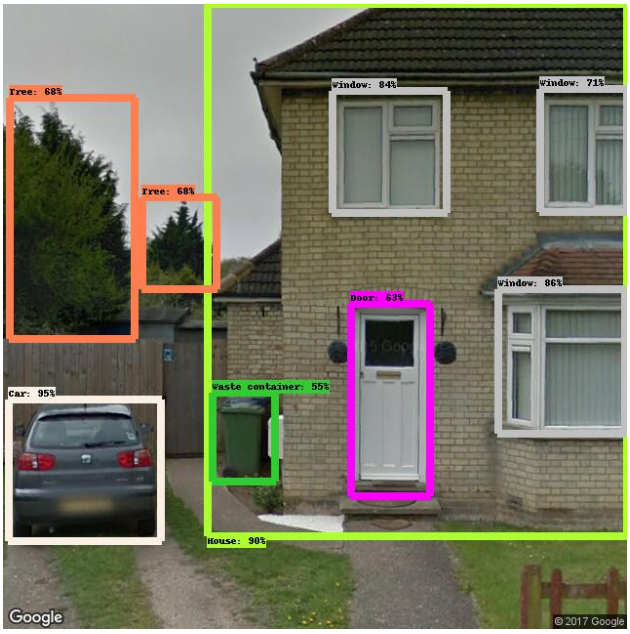
4.2 Example 2: AVM Residual Value Classification

The second example is a simple hedonic valuation model that is augmented by an additional ML building image classifier. The goal is to extract additional information from the residuals of a hedonic regression by classifying the homes into residual value quintiles using images. Other than the difference in classification categories, the classifier is identical to the previous example.

[7] The package is available at: `https://github.com/marcotcr/lime`

[8] We use the latest version of the Inception/ResNet object detection model: `https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md`.

Figure 2: Object Detection in Images



Notes: This figure provides an example of object detection results in our testing image samples. Each detected object is enclosed by a box, labeled with the classified object type. The corresponding percentage next to the label indicates the confidence in the detection results.

Again, we analyse whether the super-pixels coincide with doors, windows, or other objects. In addition, we investigate how the super-pixels will differ when the classification task changes. We expect some overlap in super-pixels from the architectural style classification and a residual value classification. Architectural styles have been documented as important determinants of residential property price in the UK (e.g. Law *et al.*, 2019; Lindenthal & Johnson, 2021) and the residual value classifier is likely to focus on the same objects as the architectural style classifier. Adding style information to the hedonic regression stage, however, should reduce the reliance on architectural features and lead to more different super-pixels.

We form several hypotheses of the model verification testing results for this example:

- There is information in the residuals from the hedonic price model that can be extracted by the ML image classifier. Adding the residual value classifier will improve the accuracy of the overall AVM.

- Residual value classifiers will categorize buildings more accurately when residuals stem

from relatively parsimonious hedonic models. For models with only a few hedonic variables, there are simply more unobserved factors that might be picked up in the images.

- The super-pixels from the residual value classifier will partially overlap with the superpixels from the architectural style classification in 4.1. However, the overlap will be smaller when architectural style is added as a hedonic variable in the initial hedonic regression.

- The residual value classifier will disproportionally rely on information from windows and doors because windows and doors are critical determinants of building styles.

- Positive externalities from greenery could influence property values. We expect trees to matter more for residual values than for architectural styles.

- Cars and property value are both related to a household's wealth. We, therefore, expect information on nearby cars not to be ignored when trying to assess a house's value.

4.2.1 AVM Data and Methodology

In this example, we combine the same image data in the first example with the transaction data. We collect the residential property transactions in Cambridge, England between January 1995 and October 2018 from the UK Land Registry. The Land Registry records the date of transaction, the price paid, street address, a classification of the property type (flat, detached, semi-detached, or terraced house), the estate type (freehold or leasehold), and an indicator for newly built properties. We exclude the leasehold properties and flats in our sample. This ends up with 26,841 transactions of 15,855 buildings. Appendix Table B1 presents the summary statistics of the transaction data. We also measure each building's floor areas using maps for the UK Ordnance Survey and estimate the building's volume by combining the building outlines with digital elevation models from the Environment Agency (2015), following the method by (Lindenthal, 2018). In addition, we also measure the distance from each residential property to the city center (i.e., the Great St. Mary's Church). Lastly, we match the address of the buildings with 69 unique district codes in

Cambridge.[9]

Our first hedonic price model without controls for the architectural styles (Model 1) is specified as follow:

$$log(Price_{it}) = \alpha_0 + X_{it}'\beta + \varphi_t + \omega_i + \epsilon_{it}. \quad (4)$$

$log(Price_{it})$ is the logarithmic form of the transaction price for property i at time t. X_{it} is a set of hedonic housing features, including the logarithmic form of distance to the city center, the floor area, and the volume of the house; and a dummy variable indicating the new constructions. φ_t and ω_i denote the year and district fixed effects, respectively. ϵ_{it} is the error term. Our second the hedonic model (Model 2) is modified from Equation (4) by adding an additional set of dummy variables denoting the predicted architectural styles from our ML model in Section 3. The coefficients of the two hedonic models are estimated using the full sample of transactions, excluding the 2,791 buildings in the out-of-sample prediction set. Appendix Table B2 reports the hedonic model estimation results.

Then, we train the ML models to classify the residuals of the two hedonic models in terms of five quintiles[10], using the sample of 13,448 building images that can match with the most recent transaction record (excluding the 2,791 images in the predict sample set). Like our ML model for classifying architectural style in Section 3, we randomly sampled 80% of these images as the training data and use the rest 20% as the in-sample validation data. The other model settings are the same as the architectural style model. Lastly, we use the ML classifiers to predict the hedonic price residuals (i.e., which quintiles it belongs to) of the sampled images in the prediction set.

[9]We classify the districts according the Lower Super Output Areas (LSOA) in the UK, which typically has 1,000-3,000 residents and 400-1,200 households of comparable economic and socio-demographic characteristics.

[10]Images can also be trained to directly predict the continuous residuals of the hedonic price models. Here we transform to a classification problem to be comparable with results of the architectural style classifications

5 Results

5.1 Example 1: Vintage Classification Based on Street-level Imagery

Figure 3 illustrates the interpretation results of an *Interwar*-style house, which is correctly predicted by the model with a prediction score of 0.4119. The verification test reveals that the model has mainly considered the brickworks in the front door and the design of windows on the first floor for this conclusion. Nevertheless, the model has relatively low confidence in this prediction result, with the HHI equal to 0.3117. The top second and third predicted styles are *Revival* and *Postwar*, and the corresponding prediction scores are 0.3050 and 0.2112, respectively. The most decisive building features for these incorrect predictions are the windows on the second floor and the sloping roof. In contrast, the model has effectively distinguished this building from the other four building styles, because the corresponding prediction scores are all lower than 0.1. Interestingly, the LIME analysis reveals that only some irrelevant information in the image—such as the sky, the fence of the garden, and the road—may potentially "explain" these four incorrect styles. In other words, the model does not establish associations between any physical features of this house and the four incorrect styles, which further supports that the model has confidently rejected these four styles based on the relevant information.

Apart from improving the model's trustworthiness by interpreting the associations with relevant information, it is also important to assure that the model does not capture the irrelevant information to reach the correct classifications by coincidence. Figure 4 shows two examples of the analysis results. The *Edwardian*-style building in Figure 4a is correctly classified by our model. The analysis shows that this prediction is mostly based on the brick patterns and sash windows, which are the key features of *Edwardian* buildings. However, in a much smaller number of cases, we find the model's prediction is also influenced by irrelevant information, even though the final prediction result turns out to be correct coincidentally. Figure 4b shows a *Postwar*-style building that is also correctly classified by the model. However, the model classifies this image mainly based on the top of a black car, which just accidentally passes by the building.

Figure 3: Interpretation for the Model's Prediction Scores on Each Architecture Style

Notes: This figure shows the local model interpretation results for an interwar-styled building, which is correctly classified by the model but with low confidence (HHI = 0.3117). The model's prediction score for each architectural style is reported, which denotes the probability that the building belongs to the corresponding style. The images show the most decisive super-pixels that explain each architectural style.

Figure 4: Interpretation for the Classifier's Ability of Capturing Relevant Information

(a) Relevant Information

(b) Irrelevant Information

Notes: Figure (a) is an Edwardian-styled building that is correctly classified by the model. The local model interpretation analysis shows that the model captures the relevant information of the bricks and sash windows for this prediction. Figure (b) shows a postwar building that is also correctly classified by the model. However, the model classifies this image mainly based on the top of a black car, which accidentally passes by.

The previous visual inspection is only the interpretation for each input instance, and it does not easily scale up in the context of big data. Thus, we formalize the inspection by calculating the proposed verification test score of each image. As described in our hypothesis (Section 4.2), we use the test score for houses—the most essential objects for classifying building styles—as the baseline result, and we use the test scores for windows and doors as the robustness checks to address the concern that some pictures only capture parts of the buildings. We also calculate the test score for trees and cars as the indicators for capturing irrelevant information.

Panel A of Table 1 reports the average verification test scores for the sub-samples of each building style. For the classification of *Georgian*, *Victorian*, *Edwardian*, or *Revival* buildings, the verification test scores of houses range from 0.857 to 0.890, which indicates that the model well captures the housing features for the classification of these building types. Similar patterns are observed if we use the verification test scores of windows or doors as alternative measurements. For the *Edwardian*-style buildings, the verification test score for windows is as high as 0.302, which affirms the finding in Figure 4a that the model captures the unique design of windows for this architectural style. However, for the classification of *Postwar*-style buildings, only 69% of the most explainable areas are from the houses. This is also a support for the unique case we observe in Figure 4b that the irrelevant information may threaten the trustfulness in classifying *Postwar*-style buildings. To further eliminate the potential bias from images showing the only parts of the buildings (i.e., the verification test score of house in these images will always be one), we exclude these samples in an additional robustness check and report the updated verification test scores in Appendix Table B3. Similar trends are observed, which indicates that our main conclusions are robust.

One major concern of the proposed verification test score is that the areas of objects in each image are not equal. To address this issue, we first compare the test scores with the benchmark score—the percentages of the objects' areas in the images, which are reported in Appendix Table B4. The model verification scores of houses are all larger than the benchmark score. This assures that the model is trustworthy because it is intentionally capturing information from the houses for

Table 1: Verification Test Results of Vintage Classifier

Panel A. Verification Test Score								
	Architect's Classification							
	(1) All	(2) Georgian	(3) Victorian	(4) Edwardian	(5) Interwar	(6) Postwar	(7) Contemp.	(8) Revival
House	0.801	0.857	0.890	0.874	0.779	0.690	0.728	0.888
	(0.005)	(0.029)	(0.008)	(0.009)	(0.010)	(0.012)	(0.013)	(0.012)
Window	0.188	0.241	0.176	0.302	0.187	0.136	0.137	0.178
	(0.004)	(0.029)	(0.008)	(0.010)	(0.008)	(0.007)	(0.007)	(0.010)
Door	0.036	0.071	0.081	0.045	0.011	0.010	0.031	0.029
	(0.002)	(0.019)	(0.006)	(0.005)	(0.002)	(0.003)	(0.004)	(0.005)
Tree	0.093	0.082	0.030	0.083	0.143	0.130	0.085	0.078
	(0.004)	(0.027)	(0.005)	(0.008)	(0.010)	(0.010)	(0.009)	(0.011)
Car	0.052	0.005	0.074	0.053	0.052	0.049	0.042	0.045
	(0.002)	(0.004)	(0.007)	(0.006)	(0.006)	(0.006)	(0.005)	(0.008)
Panel B. Verification Test Ratio (Verification Test Score/Benchmark)								
	Architect's Classification							
	(1) All	(2) Georgian	(3) Victorian	(4) Edwardian	(5) Interwar	(6) Postwar	(7) Contemp.	(8) Revival
House	1.103	1.114	1.068	1.127	1.140	1.080	1.061	1.176
Window	1.336	1.493	1.024	1.581	1.524	1.414	1.081	1.441
Door	1.399	1.765	1.515	1.406	1.062	1.579	1.211	1.304
Tree	0.787	0.774	0.769	0.701	0.782	0.764	0.985	0.760
Car	1.167	2.065	1.359	1.181	1.126	1.248	0.968	0.992

Notes: In Panel A, the *verification test score* presents the proportion of the interpretable area (super-pixels) that overlap with objects detected in the image (e.g., house or window), by vintage. Standard errors are reported in parentheses. In Panel B, the *verification test ratio* normalizes the verification test scores by dividing by the share each object takes up of the entire image. A ratio larger than 1 means that the ML model uses relatively much information from the object type to classify building styles, a score below 1 indicates a lack of emphasis. The ratios for the façade, windows, and doors are larger than 1 overall, lower than 1 for trees, and mixed for cars.

the predictions. Taking a closer look at each category, it is questionable whether the low test score is driven by the low proportion of buildings in the images. For the *Postwar* category, the area of buildings on average only constitutes 63.9% of the image area in the testing data, which implies more noises and thus increases the difficulty of classification.

We further calculated the ratio of our verification test score to the benchmark, which is denoted as the verification test ratio. The results are reported in Panel B of Table 1. This ratio represents the effort that the model selects information from that type of object, in comparison with randomly selecting information from the image. After controlling for this heterogeneity in the testing data, the test ratio of the *postwar* building is still relatively low at 1.08 and the test ratio for *Edwardian* buildings is still the highest at 1.127. For all the seven building styles, our model intentionally selects the relevant information from houses, windows, and doors, because these test ratios are larger than one. In contrast, the model also disregards irrelevant information from the trees as we hope, because the ratio is lower than one.

One interesting finding is that the model also captures information from cars for the classification of several building categories, as those verification test ratios are larger than 1. When we initially design the verification test, we consider that cars are irrelevant to the building vintages. However, it is plausible that cars are not really "irrelevant" information due to endogeneity: The architectural styles may correlate with some other specific designs (i.e. large spaces in the front door), which allow cars to be parked in front of the buildings. Unobserved factors like the demographics and wealth status of homeowners may also both correlate with the type of their cars and the architectural style of their homes (Bricker *et al.*, 2020).

To answer this question, we further test the impact of capturing information from each category on the model's prediction accuracy. The hypothesis is that, if the model captures more real relevant information from the training samples, it will have a higher prediction accuracy in the testing samples. In other words, if capturing information from cars decreases the model's classification accuracy, we can reject the hypothesis that unobserved correlation exists between cars and building styles, and we can be affirmed that the cars are irrelevant information.

We first examine this hypothesis using a univariate t-test, of which the results are reported in Table 2. Specifically, we compare the means of verification test scores between the correctly and the incorrectly classified images. It reveals that, in the correct classifications, the model relies more on relevant information like houses, windows, and doors, and it captures less irrelevant information about trees and cars. All the differences in mean are statistically different at the level of 1%. The results remain robust if we compare the difference in mean using sub-samples of each category (Appendix Table B5).

Table 2: Verification and Accuracy of Vintage Classifier - Univariate Test

	(1)	(2)	(3)
	Y: Verification Test Score		
	Correct Classifications	Incorrect Classifications	Difference (1)-(2)
House	0.8186	0.7624	0.0562***
	(0.0052)	(0.0093)	(0.0107)
Window	0.1984	0.1640	0.0344***
	(0.0042)	(0.0064)	(0.0077)
Door	0.0392	0.0271	0.0121***
	(0.0024)	(0.0029)	(0.0038)
Tree	0.0853	0.1095	-0.0242***
	(0.0042)	(0.0073)	(0.0084)
Car	0.0485	0.0609	-0.0123**
	(0.0028)	(0.0050)	(0.0057)

Notes: This table compares the verification test scores for the subsample of correct classifications and incorrect classifications. A higher verification test score for an object type means that the ML model uses more information from the object for classification. Column (1) reports the model verification score for the correctly classified sample. Column (2) reports the model verification score for the incorrectly classified sample. A positive difference in Column (3) means that the ML model uses more information of the object (e.g., house) for the correct predictions than for the incorrect predictions, and vice versa. Standard errors are reported in parentheses. *** p<0.01, ** p<0.05, * p<0.1

In addition, we conduct a logit regression to further examine the impact of verification test scores on the prediction accuracy. The estimation results are reported in Table 3. In Column (1), we regress an indicatory variable denoting the correctness of classification on the test scores of the five object types. In Column (2), we further include the fixed effects of individual building styles to capture the style-dependent difficulties in the predictions. We find that, if the verification test score

of house increase by 0.1, the odds ratio of a correct classification will increase by approximately 0.058 (Column (1)) to 0.074 (Column (2)), and these estimates are statistically significant at the level of 1%. Similarly, the odds ratio improves with higher verification test scores of windows and doors, and it decreases with higher verification test scores of cars and trees. Since capturing information on cars is not associated with higher classification accuracy, it implies that cars are irrelevant information as we originally hypothesized, and we should adjust the model from capturing information of cars.

Table 3: Verification and Accuracy of Vintage Classifier - Logit Regression

	(1)	(2)
	Y: Correct Classification (Yes = 1)	
House	0.5832***	0.7445***
	(0.1883)	(0.1994)
Window	0.6105**	0.6910**
	(0.2744)	(0.3042)
Door	0.8968*	0.8705*
	(0.4878)	(0.5048)
Tree	-0.4670**	-0.4726**
	(0.2064)	(0.2109)
Car	-0.2696	-0.2358
	(0.3132)	(0.3186)
Building Style Fixed Effect	N	Y
Observations	2,791	2,791
Pseudo R-Squared	0.014	0.027

Notes: This table presents logit regression estimates for the impact of verification test scores on prediction accuracy. The dependent variable is a binary variable denoting whether the building is correctly classified. The explanatory variables are the model verification scores for each object of interest. A higher verification test score for an object type means that the ML model uses more information from the object for classification. A positive estimate of the coefficient means that a higher verification test score of the object (i.e., using more information from the object for classification) correlates with a better prediction accuracy, and vice versa. Robust standard errors are reported in parentheses. *** p<0.01, ** p<0.05, * p<0.1

In summary, the model verification analysis on the exemplar model demonstrates that, compared to the F_1 score for evaluating the prediction accuracy, our proposed verification test provides an additional dimension to examine the performance of an ML model. Moreover, we also find a

positive correlation between the model's ability in capturing relevant information and the model's prediction accuracy. Therefore, apart from improving the trustworthiness of the "black box", the verification test result also serves as an effective indicator of the prediction accuracy.

5.2 Example 2: AVM Residual Value Analysis

We report the analysis results corresponding to our six AVM-related tests. First, we find that based on building images only, the ML models classify the price residuals of Model 1 more accurately than the residuals of Model 2. Panel A of Table 4 reports the confusion matrix of the Model 1, and Panel B reports the confusion matrix of the Model 2. We find that the average F_1 scores for predicting residuals of Model 1 and 2 are 36.10% and 33.31%, respectively. This means that, after including the architectural styles in the hedonic model, the F_1 score of price residual prediction by around 3 percentage points. Therefore, this supports our argument that the residuals of Model 1 contain more information on how building appearances impact the price, so the ML classifier can extract more of this information from the building images.

Second, we confirm that including predicted residuals from the ML image classifiers in the hedonic price models will increase their price prediction accuracy. For Model 1, the RMSE and MAE of the prediction results are 0.2229 and 0.1613, respectively. After including the predicted residual quintiles as additional control variables in Model 1, the RMSE and MAE of Model 1 decrease to 0.2078 and 0.1492, respectively. It translates to a decrease of 6.8% in RMSE and a 7.5% decrease in MAE. Similarly, we find that after including predicted residual quintiles in Model 2, the RMSE decreases from 0.2169 to 0.2018, and the MAE decreases from 0.1562 to 0.1445. Therefore, these results show that adding information from images to the hedonic price models improves the price prediction accuracy. It is also noteworthy that the prediction accuracy is of Model 2 is slightly higher than the accuracy of Model 2, which verifies the existing but relatively small impacts of architectural styles on housing prices (Lindenthal & Johnson, 2021).

Third, we find larger overlaps between the *InterpretAreas* of building styles and price residuals in Model 1. In Table 6, we report the average size of the overlapped *InterpretArea*, which is scaled

Table 4: Confusion Matrix of AVM Residual Classifier

Panel A. Hedonic Price Model 1 (Without Architectural Style)						
	(1) All	(2) 1st Quintile	(3) 2nd Quintile	(4) 3rd Quintile	(5) 4th Quintile	(6) 5th Quintile
Precision	36.93%	46.84%	17.70%	10.92%	20.23%	88.97%
Recall	35.31%	36.62%	39.47%	32.48%	34.16%	33.81%
F1 Score	36.10%	41.10%	24.44%	16.34%	25.41%	49.00%
Panel B. Hedonic Price Model 2 (With Architectural Style)						
	(1) All	(2) 1st Quintile	(3) 2nd Quintile	(4) 3rd Quintile	(5) 4th Quintile	(6) 5th Quintile
Precision	33.03%	71.57%	27.25%	9.28%	17.96%	39.09%
Recall	33.60%	27.79%	32.97%	23.88%	28.84%	54.54%
F1 Score	33.31%	40.03%	29.84%	13.37%	22.14%	45.54%

Notes: Panel A reports the confusion matrix of the residual prediction results for the hedonic price model without controls for architectural styles (Model 1). Panel B reports the confusion matrix of the residual prediction results for the hedonic price model with controls for architectural styles (Model 2). The *Recall* denotes the fraction of relevant instances among the retrieved instances. The *Precision* denotes the fraction of retrieved relevant instances among all relevant instances. The *Recall* and *Precision* for all samples are the average values across quintiles. F_1 score is the harmonious mean of *Precision* and *Recall*.

over the *InterpretArea* for price residuals. In Model 1, 19.02% (Column (1)) of the *InterpretArea* for price residuals are from the *InterpretArea* of building styles. In Model 2, only 17.79% (Column (2)) of the *InterpretArea* for price residuals overlap with the *InterpretArea* of building styles. The difference is statistically significant at the 5% level. Therefore, it verifies the Model 2 uses less information that is determinant for architectural styles to predict price residuals. This is because we have explicitly controlled for architectural styles in Model 2, so the price residuals of Model 2 are expected to contain less information that correlates with architectural styles than Model 1. More interestingly, we find that *Georgian* and *Edwardian* vintages have strong impacts on predicting price residuals in Model 1. These are consistent with the hedonic coefficient estimates in Model 2 that *Georgian* and *Edwardian* styles have the largest price premium (Appendix Table B2). Therefore, this provides further evidence that the ML residual classifier captures unobserved housing features that matter for property prices from the images.

Table 5: Prediction Accuracy of AVM Residual Classifier

Models	(1) RMSE	(2) MAE
Model 1 (without Architectural Style)	0.2229	0.1613
Model 1 (without Architectural Style) + Predicted Residuals	0.2078	0.1492
Model 2 (with Architectural Style)	0.2169	0.1562
Model 2 (with Architectural Style) + Predicted Residuals	0.2018	0.1445

Notes: Model 1 refers to the hedonic price model without controls for architectural styles. Model 2 refers to the hedonic price model with controls for architectural styles. A smaller RMSE/MAE means that the model has a better prediction accuracy of the property price.

Fourth, we find that the ML residual classifier of Model 1 takes more information from windows and doors than Model 1 because windows and doors are critical determinants of building styles. Specifically, Table 7 compares the $TestRatio$ for the ML residual classifiers of Model 1 and 2. The $TestRatio$ for window and door are higher in Model 1 than in Model 2, and the differences are statistically significant at the 1% and 5% level, respectively. This pattern is also largely robust if we investigate the $TestRatio$ by each category of residual quintiles (Appendix Table B6). Therefore, our results further reveal the residuals in Model 1 contain more information on buildings styles than residuals in Model 2, while the ML image classifiers for price residuals effectively capture this information.

Last, we also find that trees and cars matter more for price residuals than for architectural styles. For the price residual classifiers, the $TestRatio$ of trees and cars range between 0.806 and 0.807, and between 1.627 and 1.633, respectively (Columns (1) and (3), Table 7). They are larger than the $TestRatio$ of trees (0.787) and cars (1.167) in the vintage classifier, as reported in Column (1) of Panel B, Table 1. These support our hypothesis that greenery and cars matter more for property price than for vintage.

In summary, our analysis results demonstrate that combining ML image classifiers in the classic hedonic housing price models will improve price prediction accuracy. More importantly, our system testing interprets how the "black box" of the ML classifier improves the price prediction

Table 6: Overlaps between the *InterpretAreas* of AVM Residual and Vintage

	(1)	(2)	(3)	(4)	(5)	(6)
	Y: Overlapped *InterpretArea*					
	Model 1: Without Style		Model 2: With Style		t-test	
	Mean	Std. Dev.	Mean	Std. Dev.	Diff (1)-(3)	Std. Err.
All Buildings	0.1902	0.2074	0.1779	0.2005	0.0123**	0.0055
Georgian	0.2459	0.2601	0.1624	0.2161	0.0827***	0.0281
Victorian	0.1873	0.2101	0.1896	0.2003	-0.0024	0.0147
Edwardian	0.2262	0.2200	0.1841	0.2219	0.0421***	0.0167
Interwar	0.1930	0.2157	0.1984	0.2063	-0.0053	0.0184
Postwar	0.1527	0.1919	0.1425	0.1846	0.0102	0.0207
Contemporary	0.1138	0.1229	0.1160	0.0097	-0.0022	0.0139
Revival	0.2216	0.2044	0.2111	0.2222	0.0105	0.0249

Notes: Model 1 refers to the hedonic price model without controls for architectural styles. Model 2 refers to the hedonic price model with controls for architectural styles. Columns (1) and (3) report the overlapped areas between the *InterpretAreas* of AVM residuals and architectural styles. The overlapped areas are scaled over the *InterpretAreas* of AVM residuals. A larger overlapped area means that the model uses more information from the architecture styles to predict price residuals. A positive difference in Column (5) means that Model 1 use more information on architecture styles to predict the price residuals than Model 2 does, and vice versa. *** p<0.01, ** p<0.05, * p<0.1

accuracy. By comparing the *InterpretArea* in Model 1 (i.e., with building-style information in residuals) with the benchmark Model 2 (i.e., without building-style information in residuals), we directly show that the ML classifier for Model 1 effectively capture the unobserved correlation between building styles and housing prices, and thus improves the price prediction accuracy.

Table 7: Verification Test Ratio of AVM Residual Classifier

	(1)	(2)	(3)	(4)	(5)	(6)
	Y: Verification Test Ratio					
	Model 1: Without Style		Model 2: With Style		t-test	
	Mean	Std. Dev.	Mean	Std. Dev.	Diff (1)-(3)	Std. Err.
House	1.0664	0.2857	1.0746	0.2784	-0.0082	0.0103
Window	1.3615	1.3884	1.0153	1.1808	0.3465***	0.0502
Door	1.4063	2.0807	1.1242	1.9057	0.2821**	0.1168
Tree	0.8057	1.0469	0.8066	1.5200	-0.0009	0.0622
Car	1.6267	1.8177	1.6325	1.6652	-0.0058	0.1083

Notes: Model 1 refers to the hedonic price model without controls for architectural styles. Model 2 refers to the hedonic price model with controls for architectural styles. In Columns (1) and (3), the *verification test ratio* equals the verification test score over the benchmark score, and the benchmark score is the ratio of the object size to the image size. If the verification test ratio is larger than one, it means that the ML model intentionally uses information from the object (e.g., window or door) to classify price residuals, and vice versa. A positive difference in Column (5) means that Model 1 uses more information of the object to predict the price residuals than Model 2 does, and vice versa. *** p<0.01, ** p<0.05, * p<0.1

6 Discussion

The proposed model verification test tends to address the complexity of understanding ML models through simplified explanations at the local level. The trade-off is that the design of a model verification test is very specific to different research questions and contexts. For instance, to quantitatively interpret our illustrative classifier of architectural styles in this study, the categories of relevant and irrelevant information (i.e., houses or cars) need to be identified when designing the test. In this section, we further discuss some other factors that may impact the evaluation of verification test scores and need to be considered in designing the test. We first explore how the sample size of training data and the number of selected super-pixels will impact the calculated verification test score. Then we illustrate a simple methodology to mitigate the impact of irrelevant information by tuning the zooming factors in the image collection process.

6.1 Sample Size of Training Data

It is widely acknowledged in the literature that larger sample size in the training data will generally lead to a better prediction accuracy of ML models (Karimi *et al.*, 2019; Nghiep & Al, 2001; Park *et al.*, 2011). It is therefore of our interest to testify if the verification test score of our model is also dependent on the sample size in the training data. Specifically, we decrease the sample size in the training data by randomly choosing half of the samples from the same training data of our baseline model. As a result, we have ended up with 1,950 training images: There are 300 training images for each building style, except for the *Georgian* style which has 150 images. The rest of the model settings and training processes remain unchanged. We then evaluate the model's prediction accuracy and the verification test score using the same out-of-sample testing set, and results are reported in Table 8.

We first present in Panel A the differences in the F_1 score between the new model with fewer training samples and our baseline model. As expected, the F_1 scores of the new model are lower for the classification of all categories, which indicates that the prediction accuracy of the model improves after doubling the training samples. Panel B reports the differences in the average verification test scores between the new model and the baseline model. In contrast to the F_1 score, we find most of the differences in verification test scores are not statistically significant. This indicates that the verification test scores remain relatively stable after we increase the number of training images from 1,950 to 3,900.

The important implication of this finding is that, with a larger size of training data, the verification test score of a model tends to saturate earlier than the model's F_1 score. Therefore, as long as the training sample size is sufficient to reach a satisfactory level of prediction accuracy, the evaluated verification test score can be regarded as a reliable measure of the model's ability in extracting the relevant information.

Table 8: F_1 Score and Model Verification Test Score with Smaller Training Sample Size

Panel A. F_1 Score(small samples) - F_1 Score(large samples)							
	Architect's Classification						
	(1) Georgian	(2) Victorian	(3) Edwardian	(4) Interwar	(5) Postwar	(6) Contemp.	(7) Revival
F_1 Score	-0.026	-0.039	-0.052	-0.035	-0.031	-0.032	-0.051
Precision	-0.053	-0.005	-0.067	-0.164	0.034	-0.069	0.035
Recall	0.038	-0.070	-0.040	0.090	-0.110	-0.002	-0.159
Panel B. Verification Test Score(small samples) - Verification Test Score(large samples)							
	Architect's Classification						
	(1) Georgian	(2) Victorian	(3) Edwardian	(4) Interwar	(5) Postwar	(6) Contemp.	(7) Revival
House	0.009	0.003	-0.021	0.023	0.039**	0.012	-0.023
	(0.040)	(0.011)	(0.013)	(0.014)	(0.016)	(0.018)	(0.018)
Window	0.068	0.020*	0.031**	0.001	0.014	-0.012	-0.014
	(0.042)	(0.011)	(0.014)	(0.011)	(0.011)	(0.010)	(0.014)
Door	-0.009	0.006	-0.004	0.000	-0.003	0.000	0.002
	(0.026)	(0.009)	(0.007)	(0.003)	(0.003)	(0.006)	(0.008)
Tree	0.001	-0.002	0.005	-0.009	-0.003	0.008	0.005
	(0.039)	(0.007)	(0.011)	(0.014)	(0.014)	(0.012)	(0.016)
Car	-0.003	0.002	-0.001	-0.008	0.009	0.003	0.014
	(0.004)	(0.010)	(0.008)	(0.008)	(0.009)	(0.007)	(0.013)

Notes: Small samples refer to the model trained with 300 images in each category, except for 150 in Georgian style. *Large samples* refer to the model trained with 600 images in each category, except for 300 in Georgian style. Same prediction data is used for the comparison of the two models. In Panel A, the *Recall* of a category denotes what percentage of the buildings in that architectural style are correctly predicted by the machine. The *Precision* of a category represents the percentage of correctly classified buildings among the buildings predicted as that architectural style. F_1 score is the harmonious mean of *Precision* and *Recall*. In Panel B, standard errors of paired t-test are reported in parentheses.

*** p<0.01, ** p<0.05, * p<0.1

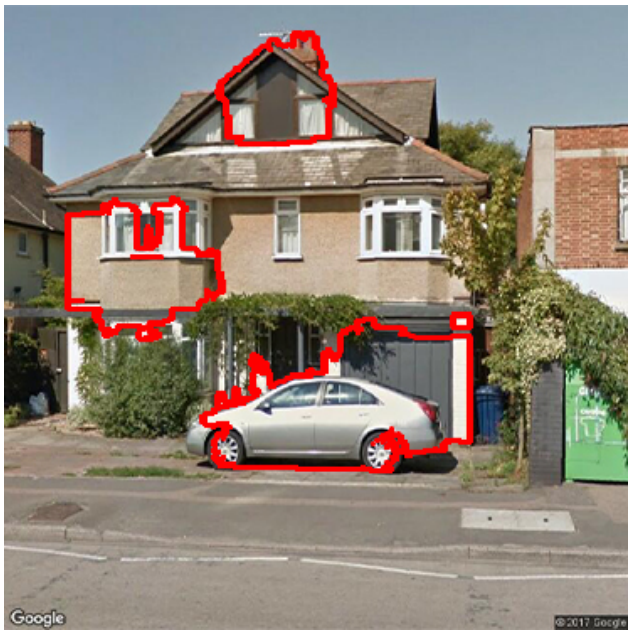
6.2 Size of the Interpretable Super-Pixels

When designing the model verification test, another critical parameter is the number of super-pixels that the diagnostic algorithm returns from each image (denoted as the "feature size" in Ribeiro *et al.* (2016)). Figure 5 shows the interpretable areas of an *Interwar*-style building with different returned feature sizes. With our baseline model, this image is correctly classified. When the feature size is set to five as default (Figure 5a), we identify the top five most important super-pixels that determining this is an *Interwar*-style building. It reflects that the model captures several important elements from the house like the roof, window, and door, but the model also considers the car in front of the building as a decisive element. However, after we change the feature size to three (Figure 5b), the algorithm returns the top three most important elements, and we notice that only the roof and the car remain in the identified interpretable area. This reveals that the model considers the car more important than the widow and the door for the classification.

Setting an appropriate feature size is essential for obtaining a reliable verification test score. If the feature size is too small, the interpretation algorithm will only return the most decisive super-pixels for the classification. If we assume that each super-pixel is solely from one object, the resulted test score will either be too optimistic or too pessimistic. For instance, if the feature size is set to be one for the explanation of Figure 5, only the roof of the building will be returned, and we will not discover that the model significantly captures the irrelevant information from the car. However, if the feature size is too large, the returned interpretable area will also be enlarged. The corresponding verification test score will converge to the benchmark and the test thus loses its sensitivity.

Finding the appropriate feature size is a process of trial and error. A simple rule of thumb for setting the initial feature size is to have the same number of features as the number of categories for which we calculate the test scores. The intuition is that, in a benchmark situation that the super-pixels are just selected randomly, all objects from the image have the equal probability to be returned, and we will obtain a fair distribution of all the investigated categories in the returned

Figure 5: Interpretation Result with Difference Sizes of Interpretable Super-pixels



(a) Feature Size = 5

(b) Feature Size = 3

Notes: This figure demonstrates that the identification of super-pixels depends on the feature size parameter in the LIME algorithm. We use the default value of 5 throughout the paper.

interpretable area.[11] In our example of illustration, we focus on the model verification test of five categories, namely houses, windows, doors, trees, and cars. Then one preferable setting for the verification test, as in our baseline model, is to return exactly five super-pixels from the image.

6.3 Zoom Factor of the Collected Images

The quality of the training samples is the most direct factor impacting the verification test score of the model. Ideally, if the training samples only contain the information of interest, the model will be not be distracted by any irrelevant information and the verification test will reveal which of the included relevant information is mostly decisive. However, removing all the noise information in the training samples is generally challenging in real practice: It either requires too much computational power, or it is even infeasible to differentiate the noise from relevant information. For example, in the context of housing and urban studies, the street images are often captured as the training samples for the classification models, while these street images naturally contain irrelevant information like pedestrians, cycles, or cars.[12] Although existing technologies of image segmentation already enable extracting relevant objects (i.e. houses) and removing backgrounds in the images, this will significantly increase the computational burden in the context of big data.

A quick, although not perfect, methodology to mitigate the impact of noise information in street images is to adjust the zoom factor during image collection. We demonstrate this with our classification model for building styles. Figure 6 shows the images of a *Georgian*-style building in Cambridge captured from the Google Street View with different zoom factors. When the view is zoomed out to contain the full house (Figure 6a), the model captures not only the brickwork of the building but also the wall from the adjacent building, as well as the purely irrelevant information from the cloudy sky. The model incorrectly classifies this building as a *Victorian* one. If we zoom in the view as shown in Figure 6b, some irrelevant information is cropped out of the view and the

[11] This rule of thumb simply ignores the uneven number of objects for each category, or the variation in the size of super-pixels, in the images.

[12] What further complicates the problem is that the relevance of this information depends on the research questions as well. For instance, the pedestrians and cars are most likely irrelevant for the classification of building styles, but they can be decisive for a model learning the vitality of urban environments.

Figure 6: Interpretation Result with Difference Zoom Factor in Image Collection

(a) Zoom Out

(b) Zoom In

Notes: This figure demonstrates the importance of image cropping: the relevant superpixels shift significantly when zooming in on the house, reducing the area covered by other buildings, sky, and roads.

model now captures the information from the sash windows with small panes—one of the most the significant feature of the Georgian buildings, and the image is correctly classified. However, it is also worth to mention that further zooming in the views may also leave out significant information in the image, so the selection of the zoom factor is specific to the data set and involves trial and error. Both the prediction accuracy scores and the model verification scores can be applied and cross-checked to find the most appropriate zoom factor in the image collection process.

7 Conclusion

This paper makes two contributions to the applied ML literature, not only in real estate and urban studies (such as Naik *et al.*, 2016; Ibrahim *et al.*, 2020). First, we suggest a straight-forward methodology transfer from software architecture to ML applications. Adding a system testing stage to the ML system development process forces developers to verbalize how the system should arrive at its results and to formalize system verification tests. Simply throwing in as many data as possible and solely optimizing the predictive power of an ML system is a dangerous strategy in many real-life situations. Opening up the 'black box' of an ML model, if only a bit, enhances its trustworthiness.

To give examples of concrete implementations, we define model verification tests for ML image classifiers, which have become popular in real estate and urban studies (i.a. Johnson *et al.*, 2020; Lindenthal & Johnson, 2021; Liu *et al.*, 2017). We apply a local model interpretation technique to image classifiers trained for different tasks: architectural style detection and residual value classification in an AVM.

Our tests investigate which areas in an image are most important for the models' predictions and quantify the models' abilities to focus on relevant information and to exclude irrelevant noise or maybe even non-permittable aspects (race-, gender-, age-related, for instance). Further following system testing hierarchies from software engineering, we combine the proposed model verification tests with the typical accuracy tests of ML models. The proposed model-verification test and

model-design framework are applicable and recommended for ML applications in real estate and urban studies – but also in other domains where the complexity of interpreting ML models limits real-life applications (Ribeiro *et al.*, 2016).

Our tests show that the investigated models successfully select relevant information, such as windows and doors, and disregard irrelevant information like trees for the prediction of architectural styles. Also, we discover that noise in one classification task could be a signal in another. Cars included in building images reduce the classification accuracy of the architectural style detection model but are informative for the residual value assessment. We discuss how to discern and mitigate these issues using our model verification test results. Finally, we directly show that ML image classifiers improve the performance of classic hedonic housing price models because it captures some unobserved housing features, such as building styles. This adds to the emerging literature on ML techniques, image data, and AVMs (e.g. Law *et al.*, 2019).

Second, we offer a number of practical 'peeks under the hood', investigating how image quality and parameter choices affect e.g. the LIME local model interpretation estimates. Most local model interpretation techniques such as LIME or SHAP do not depend on the classification algorithms of the ML model to be interpreted (Adadi & Berrada, 2018; Ribeiro *et al.*, 2016). These tests are not difficult to implement and should be part of any ML classification analysis – just like regression diagnostics should always accompany even the most simple regression models.

While this paper mainly discusses the application of our system testing approach for image classification (i.e. with neural networks), we would like to emphasize the importance of further research into the application of system testings on interpreting other commonly applied ML models, such as text mining (Ahmed & Moustafa, 2016).

References

Adadi, Amina, & Berrada, Mohammed. 2018. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, **6**, 52138–52160.

Ahlfeldt, Gabriel, & Mastro, Alexandra. 2012. Valuing iconic design: Frank lloyd wright architecture in Oak Park, Illinois. *Housing Studies*, **27**(8), 1079–1099.

Ahlfeldt, Gabriel M, Moeller, Kristoffer, Waights, Sevrin, & Wendland, Nicolai. 2017. Game of zones: The political economy of conservation areas. *The Economic Journal*, **127**(605), F421–F445.

Ahmed, Eman H, & Moustafa, Mohamed. 2016. House Price Estimation from Visual and Textual Features. *Pages 62–68 of: International Conference on Neural Computation Theory and Applications*, vol. 4. SCITEPRESS.

Ambrose, Brent, Han, Yiqiang, Korgaonkar, Sanket, & Shen, Lily. 2020. Contractual completeness in the CMBS Market: Insights from machine learning. *Working paper*.

Ammann, Paul, & Offutt, Jeff. 2016. *Introduction to Software Testing*. Cambridge University Press.

Aubry, Mathieu, Kräussl, Roman, Manso, Gustavo, & Spaenjers, Christophe. 2019. Machine learning, human experts, and the valuation of real assets. *CFS Working Paper Series*.

Bricker, Jesse, Ramcharan, Rodney, & Krimmel, Jacob. 2020. Signaling status: The impact of relative income on household consumption and financial decisions. *Management Science*.

Buitelaar, Edwin, & Schilder, Frans. 2017. The economics of style: Measuring the price effect of neo-traditional architecture in housing. *Real Estate Economics*, **45**(1), 7–27.

Coulson, N Edward, & McMillen, Daniel P. 2008. Estimating time, age and vintage effects in housing prices. *Journal of Housing Economics*, **17**(2), 138–151.

Dastin, Jeffrey. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*, Oct.

De Nadai, Marco, Vieriu, Radu Laurentiu, Zen, Gloria, Dragicevic, Stefan, Naik, Nikhil, Caraviello, Michele, Hidalgo, Cesar Augusto, Sebe, Nicu, & Lepri, Bruno. 2016. Are safer looking neighborhoods more lively? A multimodal investigation into urban life. *Pages 1127–1135 of: Proceedings of the 24th ACM International Conference on Multimedia*.

Fan, Yi, Teo, Ho Pin, & Wan, Wayne Xinwei. forthcoming. Public transport, noise complaints, and housing: Evidence from Sentiment Analysis in Singapore. *Journal of Regional Science*.

Francke, Marc K, & van de Minne, Alex M. 2017. Land, structure and depreciation. *Real Estate Economics*, **45**(2), 415–451.

Freybote, Julia, Simon, Lauren, & Beitelspacher, Lauren. 2016. Understanding the contribution of curb appeal to retail real estate values. *Journal of Property Research*, **33**(2), 147–161.

Gebru, Timnit, Krause, Jonathan, Wang, Yilun, Chen, Duyun, Deng, Jia, Aiden, Erez Lieberman, & Fei-Fei, Li. 2017. Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States. *Proceedings of the National Academy of Sciences*, **114**(50), 13108–13113.

Glaeser, Edward L, Kincaid, Michael Scott, & Naik, Nikhil. 2018. Computer vision and real estate: Do looks matter and do incentives determine looks. *Working paper*.

Ibrahim, Mohamed R, Haworth, James, & Cheng, Tao. 2020. Understanding cities with machine eyes: A review of deep computer vision in urban analytics. *Cities*, **96**, 102481.

James, Gareth, Witten, Daniela, Hastie, Trevor, & Tibshirani, Robert. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.

Johnson, Erik B, Tidwell, Alan, Villupuram, Sriram V, *et al.* 2020. Valuing curb appeal. *The Journal of Real Estate Finance and Economics*, **60**(1), 111–133.

Karimi, Firoozeh, Sultana, Selima, Babakan, Ali Shirzadi, & Suthaharan, Shan. 2019. An enhanced support vector machine model for urban expansion prediction. *Computers, Environment and Urban Systems*, **75**, 61–75.

Koch, David, Despotovic, Miroslav, Leiber, Sascha, Sakeena, Muntaha, Döller, Mario, & Zeppelzauer, Matthias. 2019. Real Estate Image Analysis: A Literature Review. *Journal of Real Estate Literature*, **27**(2), 269–300.

Krause, Andy. 2019. A machine learning approach to house price indexes. *Working paper*.

Law, Stephen, Paige, Brooks, & Russell, Chris. 2019. Take a look around: using street view and satellite images to estimate house prices. *ACM Transactions on Intelligent Systems and Technology*, **10**(5), 1–19.

LeCun, Yann, Bengio, Yoshua, & Hinton, Geoffrey. 2015. Deep learning. *Nature*, **521**(7553), 436–444.

Lei, Jing, G'Sell, Max, Rinaldo, Alessandro, Tibshirani, Ryan J, & Wasserman, Larry. 2018. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, **113**(523), 1094–1111.

Lindenthal, Thies. 2017. Beauty in the eye of the home-owner: Aesthetic zoning and residential property values. *Real Estate Economics*.

Lindenthal, Thies. 2018. Estimating Supply Elasticities for Residential Real Estate in the United Kingdom.

Lindenthal, Thies, & Johnson, Eric B. 2021. Machine learning, architectural styles and property values. *Journal of Real Estate Finance and Economics*.

Liu, Lun, Silva, Elisabete A, Wu, Chunyang, & Wang, Hui. 2017. A machine learning-based method for the large-scale evaluation of the qualities of the urban environment. *Computers, Environment and Urban Systems*, **65**, 113–125.

Lundberg, Scott M, & Lee, Su-In. 2017. A unified approach to interpreting model predictions. *Pages 4765–4774 of: Advances in Neural Information Processing Systems*.

Molnar, Christoph. 2019. *Interpretable machine learning*. Lulu. com.

Mullainathan, Sendhil, & Obermeyer, Ziad. 2017. Does machine learning automate moral hazard and error? *American Economic Review*, **107**(5), 476–80.

Naik, Nikhil, Raskar, Ramesh, & Hidalgo, César A. 2016. Cities are physical too: Using computer vision to measure the quality and impact of urban appearance. *American Economic Review*, **106**(5), 128–32.

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

Nghiep, Nguyen, & Al, Cripps. 2001. Predicting housing value: A comparison of multiple regression analysis and artificial neural networks. *Journal of Real Estate Research*, **22**(3), 313–336.

Olah, Chris, Cammarata, Nick, Schubert, Ludwig, Goh, Gabriel, Petrov, Michael, & Carter, Shan. 2020. Zoom In: An Introduction to Circuits. *Distill*, **5**(3).

Oquab, Maxime, Bottou, Leon, Laptev, Ivan, & Sivic, Josef. 2014. Learning and transferring mid-level image representations using convolutional neural networks. *Pages 1717–1724 of: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Park, Soyoung, Jeon, Seongwoo, Kim, Shinyup, & Choi, Chuluong. 2011. Prediction and comparison of urban growth by land suitability index mapping using GIS and RS in South Korea. *Landscape and Urban Planning*, **99**(2), 104–114.

Ribeiro, Marco Tulio, Singh, Sameer, & Guestrin, Carlos. 2016. Why should I trust you? Explaining the predictions of any classifier. *Pages 1135–1144 of: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Rossetti, Tomás, Lobel, Hans, Rocco, Víctor, & Hurtubia, Ricardo. 2019. Explaining subjective perceptions of public spaces as a function of the built environment: A massive data approach. *Landscape and Urban Planning*, **181**, 169–178.

Schmidt, Carolin, & Lindenthal, Thies. 2020. The odd one out: Asset uniqueness and price precision. *Working paper*.

Selvaraju, Ramprasaath R, Cogswell, Michael, Das, Abhishek, Vedantam, Ramakrishna, Parikh, Devi, & Batra, Dhruv. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Pages 618–626 of: Proceedings of the IEEE International Conference on Computer Vision*.

Shen, Lily, & Ross, Stephen. 2020. Information value of property description: A machine learning approach. *Journal of Urban Economics*, 103299.

Simester, Duncan, Timoshenko, Artem, & Zoumpoulis, Spyros I. 2020. Targeting prospective customers: Robustness of machine-learning methods to typical data challenges. *Management Science*, **66**(6), 2495–2522.

Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon, & Wojna, Zbigniew. 2016. Rethinking the inception architecture for computer vision. *Pages 2818–2826 of: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Verma, Deepank, Jana, Arnab, & Ramamritham, Krithi. 2019a. Machine-based understanding of manually collected visual and auditory datasets for urban perception studies. *Landscape and Urban Planning*, **190**, 103604.

Verma, Deepank, Jana, Arnab, & Ramamritham, Krithi. 2019b. Transfer learning approach to map urban slums using high and medium resolution satellite imagery. *Habitat International*, **88**, 101981.

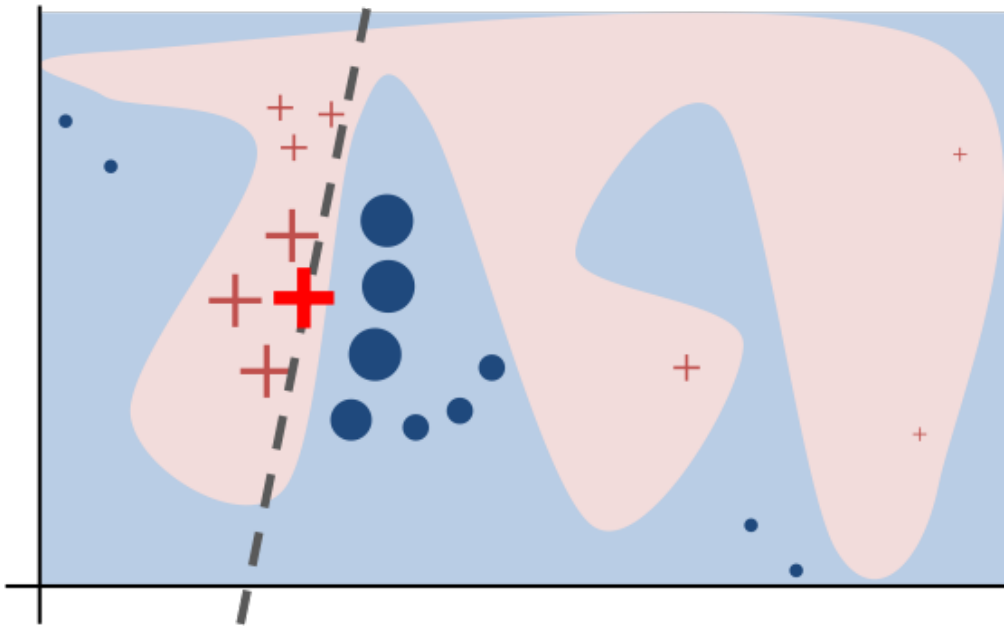
Zeiler, Matthew D, & Fergus, Rob. 2014. Visualizing and understanding convolutional networks. *Pages 818–833 of: European Conference on Computer Vision*. Springer.

Zhang, Fan, Zhang, Ding, Liu, Yu, & Lin, Hui. 2018. Representing place locales using scene elements. *Computers, Environment and Urban Systems*, **71**, 153–164.

Zhao, Junhan, Liu, Xiang, Kuang, Yanqun, Chen, Yingjie Victor, & Yang, Baijian. 2018. Deep CNN-based methods to evaluate neighborhood-scale urban valuation through street scenes perception. *Pages 20–27 of: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE.

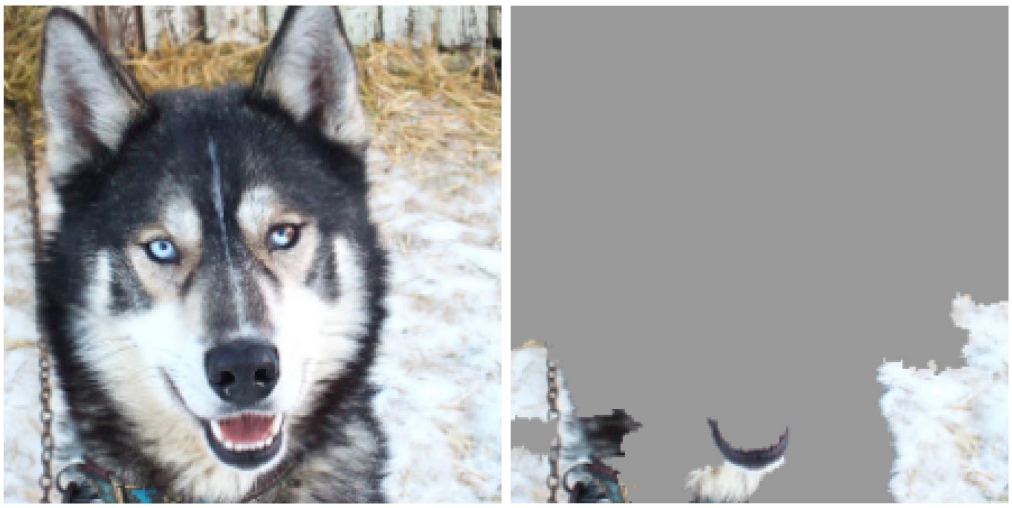
Appendix A: Supplementary Figures

Figure A1: The Intuition of Local ML Model Interpretation



Notes: Figure, as shown in Ribeiro *et al.* (2016). The ML classifier's complex global decision function is represented by the boundary of the blue/pink background, which cannot be well approximated by an interpretable linear function. However, for a specific instance represented by the bold red cross, we can use the dashed line as a proxy for the classification function that is locally (but not globally) faithful. A more detailed discussion is in Ribeiro *et al.* (2016).

Figure A2: Interpretation Result of a Bad Classifier's Prediction in the "Husky and Wolf" Task



(a) Husky classified as wolf **(b)** Interpretable Area

Notes: Figures as shown in Ribeiro *et al.* (2016). This figure demonstrates a bad model for classifying wolves and huskies, which is trained with pictures of wolves that had snow in the background and pictures of huskies without snow. The left image shows a husky that is incorrectly classified as a wolf. The right image presents the image components that are most interpretable for the model's classification result, which is analyzed with the LIME algorithm. It reflects that the model predicts incorrectly because it captures the irrelevant background information of snow.

Appendix B: Supplementary Tables

Table B1: Summary Statistics for the Housing Transaction Data

	(1) N	(2) mean	(3) sd	(4) min	(5) max	(6) p25	(7) p50	(8) p75
price	26,841	259,604	176,975	10,000	1,000,000	129,500	220,000	335,000
log(price)	26,841	12.250	0.680	9.210	13.820	11.770	12.300	12.720
volume	26,841	294.300	193.000	0	2796.000	220.600	292.100	375.800
log(volume)	26,841	4.872	2.124	0	7.936	5.401	5.681	5.932
area	26,841	63.970	30.050	4.938	1059.000	45.830	56.140	73.030
log(area)	26,841	4.096	0.377	1.781	6.966	3.846	4.045	4.304
distance	26,841	2,426	973	113	4,940	1,626	2,322	3,085
log(distance)	26,841	7.708	0.430	4.725	8.505	7.394	7.750	8.034
new (yes = 1)	26,841	0.044	0.205	0	1	0	0	0
district code	26,841	35.66	19.47	1	69	19	37	53
property type	26,841	2.394	0.712	1	3	2	3	3
year	26,841	2,005	6.621	1,995	2,018	1,999	2,004	2,010
month	26,841	6.754	3.293	1	12	4	7	9

Notes: This table presents the summary statistics of the housing transaction data used in the AVM residual classifier. *Distance* refers to the distance between the property and the city center. *District codes* are encoded according to the Lower Super Output Areas (LSOA) in the UK. *Property types* are encoded as: 1=Detached houses; 2=Semi-detached houses; and 3=Terrace houses.

Table B2: Hedonic Regression Estimates

	(1) Model 1 Y: log(price)	(2) Model 2 Y: log(price)
Constant	14.1730***	13.9090***
	(0.1601)	(0.1595)
log(distance to city center)	-0.5101***	-0.4767***
	(0.0205)	(0.0204)
log(area)	0.4707***	0.4709***
	(0.0056)	(0.0056)
log(volume)	0.0111***	0.0099***
	(0.0014)	(0.0014)
new	0.2094***	0.2155***
	(0.0109)	(0.0111)
Base: Contemporary		
Georgian		0.0701***
		(0.0175)
Victorian		-0.0069
		(0.0107)
Edwardian		0.1048***
		(0.0102)
Interwar		-0.0024
		(0.0097)
Postwar		-0.0450***
		(0.0099)
Revival		0.0434***
		(0.0132)
Year Fixed Effects	Y	Y
District Fixed Effects	Y	Y
Observations	21,186	21,186
R-squared	0.864	0.868

Notes: This table presents the regression results of the two hedonic pricing models in Example 2. We exclude transactions in the out-of-sample prediction data set and that missing predicted vintage. Model 1 refers to the hedonic price model without controls for architectural styles. Model 2 refers to the hedonic price model with controls for architectural styles. *** p<0.01, ** p<0.05, * p<0.1

Table B3: Verification Test Results of Vintage Classifier - Robustness Check

	Architect's Classification							
	(1) All	(2) Georgian	(3) Victorian	(4) Edwardian	(5) Interwar	(6) Postwar	(7) Contemp.	(8) Revival
House	0.754	0.814	0.852	0.818	0.744	0.661	0.692	0.836
	(0.005)	(0.035)	(0.010)	(0.012)	(0.011)	(0.012)	(0.014)	(0.016)
Window	0.160	0.207	0.157	0.250	0.169	0.119	0.123	0.157
	(0.004)	(0.028)	(0.008)	(0.011)	(0.008)	(0.007)	(0.007)	(0.011)
Door	0.029	0.063	0.071	0.038	0.010	0.008	0.029	0.019
	(0.002)	(0.021)	(0.007)	(0.005)	(0.002)	(0.003)	(0.005)	(0.005)
Tree	0.100	0.106	0.036	0.088	0.143	0.136	0.088	0.084
	(0.004)	(0.034)	(0.007)	(0.010)	(0.011)	(0.011)	(0.009)	(0.013)
Car	0.062	0.006	0.091	0.076	0.056	0.053	0.047	0.061
	(0.003)	(0.006)	(0.009)	(0.009)	(0.006)	(0.006)	(0.006)	(0.012)

Notes: This table presents the robustness check results for the model verification test by excluding images with the verification test score for house equal to one. The *verification test score* denotes what proportion of the interpretable area for a building style originates from an object type (e.g., house or window). A higher verification test score for an object type means that the ML model uses more information from the object for classification. Standard errors are reported in parentheses.

Table B4: Verification Test Results of Vintage Classifier - Benchmark

	Architect's Classification							
	(1) All	(2) Georgian	(3) Victorian	(4) Edwardian	(5) Interwar	(6) Postwar	(7) Contemp.	(8) Revival
House	0.727	0.769	0.834	0.775	0.683	0.639	0.686	0.755
Window	0.141	0.162	0.172	0.191	0.123	0.096	0.127	0.123
Door	0.025	0.040	0.053	0.032	0.010	0.006	0.026	0.022
Tree	0.118	0.105	0.039	0.118	0.183	0.171	0.086	0.103
Car	0.045	0.002	0.054	0.045	0.047	0.039	0.043	0.046

Notes: The verification test benchmark score equals the size of detected objects in the images over the size of the image. The benchmark score denotes the probability that the ML model captures information from the detected object if the model just randomly selects any image segments to make predictions.

Table B5: Verification and Accuracy of Vintage Classifier - Heterogeneity Test

Panel A. Georgian Style			
	Correct Classification	Incorrect Classification	Difference
House	0.9135	0.7510	0.1625**
	(0.0155)	(0.0733)	(0.0749)
Window	0.3245	0.0835	0.2410***
	(0.0339)	(0.0267)	(0.0431)
Door	0.0501	0.1102	-0.0600
	(0.0164)	(0.0442)	(0.0471)
Tree	0.0881	0.0693	0.01876
	(0.0336)	(0.0470)	(0.0578)
Car	0.0000	0.0130	-0.0130
	(0.0000)	(0.0125)	(0.0125)
Panel B. Victorian Style			
	Correct Classification	Incorrect Classification	Difference
House	0.9474	0.7892	0.1582***
	(0.0074)	(0.0264)	(0.0274)
Window	0.1896	0.1574	0.0322
	(0.0096)	(0.0199)	(0.0221)
Door	0.0350	0.0196	0.0154
	(0.0073)	(0.0062)	(0.0096)
Tree	0.0660	0.0984	-0.0323
	(0.0128)	(0.0211)	(0.0247)
Car	0.0218	0.0844	-0.0626***
	(0.0061)	(0.0189)	(0.0199)
Panel C. Edwardian Style			
	Correct Classification	Incorrect Classification	Difference
House	0.9115	0.7783	0.1331***
	(0.0073)	(0.0238)	(0.0249)
Window	0.3470	0.1877	0.1593***
	(0.0117)	(0.0168)	(0.0204)
Door	0.0521	0.0261	0.0260***
	(0.0062)	(0.0066)	(0.0090)
Tree	0.0737	0.1066	-0.0329*
	(0.0082)	(0.0180)	(0.0198)
Car	0.0412	0.0837	-0.0425**
	(0.0059)	(0.0153)	(0.0164)

Panel D. Interwar Style			
	Correct Classification	Incorrect Classification	Difference
House	0.8250	0.7004	0.1246***
	(0.0110)	(0.0198)	(0.0227)
Window	0.2198	0.1321	0.0877***
	(0.0094)	(0.0114)	(0.0148)
Door	0.0105	0.0108	-0.0003
	(0.0025)	(0.0036)	(0.0043)
Tree	0.1418	0.1458	-0.0040
	(0.0128)	(0.0171)	(0.0214)
Car	0.0485	0.0591	-0.0106
	(0.0070)	(0.0097)	(0.0120)
Panel E. Postwar Style			
	Correct Classification	Incorrect Classification	Difference
House	0.6675	0.7661	-0.0986***
	(0.0129)	(0.0245)	(0.0277)
Window	0.1289	0.1608	-0.0318
	(0.0079)	(0.0181)	(0.0198)
Door	0.0089	0.0140	-0.0051
	(0.0032)	(0.0059)	(0.0068)
Tree	0.1295	0.1329	-0.0034
	(0.0114)	(0.0208)	(0.0237)
Car	0.0452	0.0618	-0.0166
	(0.0057)	(0.0150)	(0.0160)
Panel F. Contemporary Style			
	Correct Classification	Incorrect Classification	Difference
House	0.7146	0.7525	-0.0378
	(0.0167)	(0.0214)	(0.0272)
Window	0.1312	0.1495	-0.0183
	(0.0086)	(0.0140)	(0.0165)
Door	0.0374	0.0185	0.0189**
	(0.0060)	(0.0054)	(0.0080)
Tree	0.0761	0.1010	-0.0249
	(0.0097)	(0.0169)	(0.0195)
Car	0.0444	0.0365	0.0078
	(0.0068)	(0.0079)	(0.0104)

Panel G. Revival Style			
	Correct Classification	Incorrect Classification	Difference
House	0.9474	0.7892	0.1582***
	(0.0074)	(0.0264)	(0.0274)
Window	0.1896	0.1574	0.0322
	(0.0096)	(0.0199)	(0.0221)
Door	0.0350	0.0196	0.0154
	(0.0073)	(0.0062)	(0.0096)
Tree	0.0660	0.0984	-0.0323
	(0.0128)	(0.0211)	(0.0247)
Car	0.0218	0.0844	-0.0626***
	(0.0061)	(0.0189)	(0.0199)

Notes: This table compares the verification test scores for the subsample of correct classifications and incorrect classifications by architectural styles. A higher verification test score for an object type means that the ML model uses more information from the object for classification. Column (1) reports the model verification score for the correctly classified sample. Column (2) reports the model verification score for the incorrectly classified sample. A positive difference in Column (3) means that the ML model uses more information of the object (e.g., house) for the correct predictions than for the incorrect predictions, and vice versa. Standard errors are reported in parentheses. *** p<0.01, ** p<0.05, * p<0.1

Table B6: Verification Test Ratio of AVM Residual Classifier by Categories

Panel A. Hedonic Price Model 1 (Without Architectural Style)					
	(1) 1st Quintile	(2) 2nd Quintile	(3) 3rd Quintile	(4) 4th Quintile	(5) 5th Quintile
House	1.0333	1.0731	1.0735	1.1062	1.0667
Window	1.1468	1.4674	1.2707	1.5707	1.5211
Door	1.4195	1.2297	1.8682	1.1775	1.2658
Tree	0.6621	0.8571	0.8631	0.7897	0.8746
Car	1.6086	2.0047	1.6721	1.3541	1.1629
Panel B. Hedonic Price Model 2 (With Architectural Style)					
	(1) 1st Quintile	(2) 2nd Quintile	(3) 3rd Quintile	(4) 4th Quintile	(5) 5th Quintile
House	1.0569	1.0770	1.0876	1.0621	1.1048
Window	0.7454	0.9156	0.9750	1.1785	1.6366
Door	0.9205	1.1554	0.9540	1.5401	1.4275
Tree	0.8284	0.8979	0.5981	0.9005	0.8224
Car	1.2247	1.9504	1.9637	1.7949	1.1773

Notes: This table reports the verification test ratio of the two hedonic models by each quintile. Model 1 refers to the hedonic price model without controls for architectural styles. Model 2 refers to the hedonic price model with controls for architectural styles. The *verification test ratio* equals the verification test score over the benchmark score, and the benchmark score is the ratio of the object size to the image size. If the verification test ratio is larger than one, it means that the ML model intentionally uses information from the object (e.g., window or door) to classifier the price residuals, and vice versa.

Appendix C: Description of Architectural Styles in the UK

This appendix section summarizes the key features of each architectural style that is classified with our model, cited from Lindenthal & Johnson (2021). We especially thank our colleagues from the Department of Architecture at the University of Cambridge to provide a general description of the British building styles.

- *Georgian*: This building style was popular between the 1710s to 1830s, which is featured by sash windows, fanlights above doors, the use of stucco on façades, often wrought work grilles, railings, etc.

- *Early Victorian*: This style emerged with growing popularity for individualized embellishment, and the development of affordable sheet glass, from the 1830s to 1870s. The key features include carved barge boards and finials, as well as wider sash windows. It is denoted as the Victorian style for short in this paper.

- *Late Victorian/Edwardian*: In the late Victorian era from the 1870s to the early 1900s, rich ornamental details were employed in the formal repertoires, and stained glass was widely used. Another substantial feature is the further widened bay windows in comparison to the early Victorian designs. The following Edwardian style, which emerged in the first decade of the 19th century, was similar but relatively less ornate. It is denoted as the Edwardian style for short in this paper.

- *Interwar*: New housing types were favored during the period of the two World Wars. Specifically, the lowered construction cost enabled better housing quality for the working class.

- *Postwar*: Since the 1950s, an embrace of high-rise buildings with low-rise housing were observed in the postwar architectures. Façades also vary greatly between brick, tiling, pebbledash, and render.

- *Contemporary*: With the continuous development of construction materials and techniques,

contemporary architecture from the 1980s involves more creativity in structures and emphasis more on the expressiveness of form.

- *Revival*: The revival-style building is a unique type of modern buildings that emulates the historic, mostly Victorian, architectural style.

Appendix D: Details of Training the Vintage Classification Model

D1. Image Collection

We collect the addresses of all residential buildings in Cambridge from the UK Land Registry and then input them through the Google Street View API, which returns the coordinates of the nearest panoramic camera snapshot to the given location. The panoramic view captures the major characteristics at the neighborhood and precinct level, which is widely used for the ML applications in urban and housing studies (Glaeser *et al.*, 2018; Liu *et al.*, 2017). However, it fails to accurately align with the orientation of the building and zoom to the appropriate level to capture the front view of the building. Therefore, the street views are not precise enough to satisfy the needs of classification at the individual building level.

To overcome this challenge, Lindenthal & Johnson (2021) propose a detailed method to adjust the orientation of the panoramic views. Specifically, the coordinates of the building of interest are obtained from the official maps of the Ordnance Survey in the UK. By matching the location of the building with the location of the nearest panoramic view, the exterior walls of the building that are visible from the panorama view are identified. The optimal orientation and zoom factor to capture the exterior walls are calculated accordingly. Appendix Figure D1 illustrates the orientation adjustment methodology. We end up with over 48,000 high-resolution (640×640) images of the individual building frontage, and we use them as the main dataset in this study.[13]

D2. Model Training

The transfer learning technique is applied to train our image classification model (Glaeser *et al.*, 2018; Lindenthal & Johnson, 2021). This helps to reduce the complexity in training sophisticated deep learning models from scratch, which normally involves millions of parameters and requires huge amounts of computational power. The transfer learning technique reuses some well-established pre-trained classifiers from other datasets, and it allows us to add additional layers in

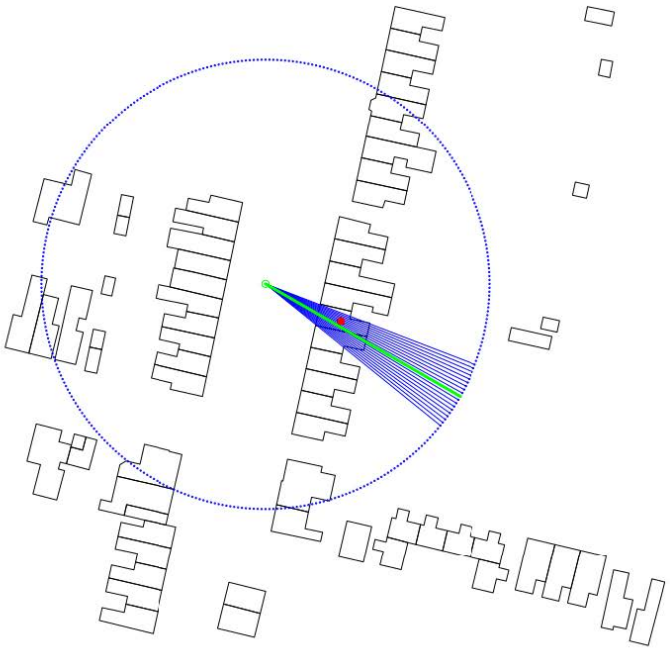
[13]Buildings with ground footprints smaller than 40 sq. m. or larger than 250 sq. m. in the maps of the Ordnance Survey are excluded in our main sample.

the pre-trained classifiers to achieve our specific classification objectives (Oquab *et al.*, 2014). In this study, we transform each image into a 2,048-dimensional feature vector, using the state-of-the-art pre-trained *Incerption-v3* CNN model (Szegedy *et al.*, 2016). The transfer learning models built from *Incerption-v3* have demonstrated the abilities in understanding human perceptions towards urban environments and housing quality in various past studies (Verma *et al.*, 2019a,b; Zhao *et al.*, 2018).

More specifically, we develop our image classifier with the following layers. Firstly, the input layer contains the 2,048-dimensional feature vectors obtained from *Incerption-v3*. Secondly, the intermediate dense layer reduces half of the dimensions with the rectified linear unit (Relu) activation function. Thirdly, we add a dropout layer to avoid overfitting, with the dropout rate set to 0.5. Lastly, the final dense layer maps the feature vectors to the 7 categories with the Softmax activation function.[14] The model is implemented through the Keras/Tensorflow Hub APIs. We train the model through iterations of 100 epochs, after which our model reaches the saturated level of training accuracy but is not yet overfitted.

[14]For the purpose of illustration in this study, our model follows the simplest standard setting as listed in the transfer learning demonstration on TensorFlow—the most commonly used open-source platform for machine learning. Reference: `https://www.tensorflow.org/tutorials/images/transfer_learning`

Figure D1: Image Collection on Google Street View: Camera Orientation and Zoom



Notes: The nearest Google Street View panorama point (green dot) based on the centroid (red dot) coordinates of a given building is obtained from Ordnance Survey maps. A viewshed analysis identifies which exterior walls are visible from the panorama point, ignoring any wall segments where the direct line of sight from the panorama point is obstructed by other buildings. The camera bearing (green line) and zoom factor are based on the angle of the most outer lines of sight (blue lines). Detailed discussion is found in Lindenthal & Johnson (2021).

Appendix E: Cross-validation Results of the Vintage Classification Model

We run the trained model on our out-of-sample testing set and evaluate the model's prediction results both qualitatively and quantitatively. Like in Lindenthal & Johnson (2021), the model we trained demonstrates a strong capability in classifying the architectural styles, as evidenced by the agreement between the prediction results and the classifications by the architects.

In the standard evaluation of the ML models, two quantitative measures are commonly used. The first evaluation uses the $F_1 score$ and the confusion matrix to examine the accuracy of classification. Specifically, the confusion matrix is the two-dimensional tabulation of the test samples by the architect's classification (the "correct" one) and the model's classification (the "predicted" one). The $F_1 score$ consists of two components, namely the *precision* (p) and the *recall* (r). *Precision* measures the probability that the model's classification for a category is correct, which is calculated as the number of correctly classified objects in a category divided by the total number of objects that the model classifies as that category. *Recall*, in contrast, represents the model's ability to find all the instances in each category from the dataset. It is calculated as the number of correctly classified objects in a category divided by the total number of objects in that category. The F_1 score is the harmonic mean of precision and recall.

Table E1 reports the confusion matrix of our model's prediction results and the corresponding F_1 score. The diagonals are the numbers of "true positive" in each category, which constitutes approximately 62.3% to 77.0% of the subsamples in the corresponding category. This indicates that, although not perfect, the model performs much better than random classification, especially considering that the model is transferred from other models that are not pre-trained for classifying architectural styles. The F_1 score ranges from 0.53 to 0.75, but its lower bound improves to 0.69 if we do not consider sub-samples of *Georgian* and *Revival*-style buildings for which prediction accuracy may suffer from the lower presence in our data. In general, the prediction accuracy of our model is comparable with the result in Lindenthal & Johnson (2021).

The second evaluation of the model's performance is the Herfindahl index (HHI), which measures the model's confidence in the classification results. At the back end, the model obtains a classification score for every category. The scores denote the probabilities that the image belongs to each category, and the sum of them equals 1. The model concludes the final classification decision by a majority vote among the scores. Therefore, the HHI—calculated as the sum of the squared term of these scores—represents how concentrated these scores are. A higher HHI means that the scores are less concentrated and the model is more confident in the classification results.

Table E2 reports the measured HHI in our model's prediction results. Panel A summarizes the average HHI for the sub-samples in each category. All the scores are over 0.8, which indicates a strong consensus that the model reaches in its classifications. There exist slight variations across categories, and the model affirms most on predictions of postwar buildings and least on contemporary ones. Panel B reports the average HHI in a two-dimensional tabulation by the correct and the predicted labels. For all the categories, the on-diagonal score is the largest one in the column. This indicates that the model agrees more with the correct predictions for all the categories (Lindenthal & Johnson, 2021).

Table E1: Prediction Accuracy: Confusion Matrix and F_1 Score

Panel A. Confusion Matrix (Number of Instances)							
Machine	*Architects' Classification*						
	(1) Georgian	(2) Victorian	(3) Edwardian	(4) Interwar	(5) Postwar	(6) Contemp.	(7) Revival
Georgian	34	15	8	3	2	5	5
Victorian	11	371	58	12	8	23	17
Edwardian	2	46	359	20	5	8	18
Interwar	0	8	30	314	37	15	5
Postwar	0	16	14	107	385	64	20
Contemp.	5	13	13	8	34	329	25
Revival	0	31	18	36	29	56	149
Panel B. Confusion Matrix (Frequency)							
Machine	*Architects' Classification*						
	(1) Georgian	(2) Victorian	(3) Edwardian	(4) Interwar	(5) Postwar	(6) Contemp.	(7) Revival
Georgian	65.4%	3.0%	1.6%	0.6%	0.4%	1.0%	2.1%
Victorian	21.2%	74.2%	11.6%	2.4%	1.6%	4.6%	7.1%
Edwardian	3.8%	9.2%	71.8%	4.0%	1.0%	1.6%	7.5%
Interwar	0.0%	1.6%	6.0%	62.8%	7.4%	3.0%	2.1%
Postwar	0.0%	3.2%	2.8%	21.4%	77.0%	12.8%	8.4%
Contemp.	9.6%	2.6%	2.6%	1.6%	6.8%	65.8%	10.5%
Revival	0.0%	6.2%	3.6%	7.2%	5.8%	11.2%	62.3%
Panel C. F_1 Score							
	Architects' Classification						
	(1) Georgian	(2) Victorian	(3) Edwardian	(4) Interwar	(5) Postwar	(6) Contemp.	(7) Revival
F_1 Score	0.548	0.742	0.749	0.691	0.696	0.710	0.534
Precision	0.472	0.742	0.784	0.768	0.635	0.770	0.467
Recall	0.654	0.742	0.718	0.628	0.770	0.658	0.623

Notes: In Panel A and B, the predictions of the out-of-sample testing images are cross-tabulated by the machine's classification versus the architects' classification. We consider the architects' classification as the correct category. In Panel C, the *Recall* denotes the fraction of relevant instances among the retrieved instances. The *Precision* denotes the fraction of retrieved relevant instances among all relevant instances. F_1 score is the harmonious mean of *Precision* and *Recall*.

Table E2: Prediction Certainty: Herfindahl Index (HHI)

Panel A. Average HHI by Architect's Classification							
	Architects' Classification						
	(1) Georgian	(2) Victorian	(3) Edwardian	(4) Interwar	(5) Postwar	(6) Contemp.	(7) Revival
Average	0.833	0.847	0.839	0.816	0.857	0.806	0.824
Panel B. Cross-tabulation of HHI by the Machine's and Architect's Classification							
Machine	*Architects' Classification*						
	(1) Georgian	(2) Victorian	(3) Edwardian	(4) Interwar	(5) Postwar	(6) Contemp.	(7) Revival
Georgian	0.878	0.786	0.555	0.461	0.607	0.489	0.731
Victorian	0.815	0.890	0.755	0.691	0.645	0.768	0.724
Edwardian	0.589	0.674	0.900	0.714	0.746	0.669	0.604
Interwar	-	0.627	0.739	0.852	0.752	0.608	0.527
Postwar	-	0.714	0.659	0.781	0.901	0.787	0.742
Contemporary	0.672	0.664	0.701	0.721	0.662	0.874	0.716
Revival	-	0.687	0.636	0.756	0.722	0.702	0.876

Notes: A higher Herfindahl index denotes a higher level of certainty in the model's prediction. No Georgian buildings in the testing samples are classified as the interwar, postwar, or revival style.

Download ZEW Discussion Papers:

https://www.zew.de/en/publications/zew-discussion-papers

or see:

https://www.ssrn.com/link/ZEW-Ctr-Euro-Econ-Research.html
https://ideas.repec.org/s/zbw/zewdip.html

IMPRINT

ZEW – Leibniz-Zentrum für Europäische Wirtschaftsforschung GmbH Mannheim
ZEW – Leibniz Centre for European Economic Research

L 7,1 · 68161 Mannheim · Germany
Phone +49 621 1235-01
info@zew.de · zew.de