SCHMEAR: Scalable Construction of Holistic Models for Energy Analysis from Rooftops

Thomas R. Dougherty*
tomdou@stanford.edu
Civil and Environmental Engineering
Stanford University
Stanford, CA, USA

Tianyuan Huang
tianyuah@stanford.edu
Civil and Environmental Engineering
Stanford University
Stanford, CA, USA

Yirong Chen
chenyr@stanford.edu
Civil and Environmental Engineering
Stanford University
Stanford, CA, USA

Rishee K. Jain
rishee.jain@stanford.edu
Civil and Environmental Engineering
Stanford University
Stanford, CA, USA

Ram Rajagopal
ramr@stanford.edu
Civil and Environmental Engineering
Stanford University
Stanford, CA, USA

Abstract

As the world moves to decarbonize, the built environment commands attention for its intensity of energy consumption. Potential pathways for decarbonizing the built environment can be discovered through the aid of building energy modeling, which helps identify potential retrofit strategies and simulate integration with renewable energy sources. Energy modeling is complicated however, due to compound interactions between building materials, structural design, and urban form. Significant domain knowledge, modeling expertise, and extensive time investment are required for accurate modeling to accommodate this complexity. In this work, we explore the potential of accurately modeling building energy consumption at scale through the application of modern computer vision algorithms. We demonstrate that our computer vision system can accurately predict energy consumption through the extraction of meaningful features contained in satellite imagery. To accomplish this, we introduce a data-collection pipeline and a computer vision architecture to process satellite photos and contextual information from the urban texture. We also demonstrate a method of comparing the relative significance of the automatically extracted features in informing building decarbonization decision making and policy. Our results indicate that this approach reveals valuable insights into the dynamics of building energy consumption on the city scale and enables the rapid analysis of urban energy dynamics with readily available data.

BuildSys '21, November 17–18, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9114-6/21/11…$15.00
https://doi.org/10.1145/3486611.3486666

CCS Concepts: • **Computing methodologies** → **Model development and analysis**; **Computer vision**; • **Information systems** → Information retrieval.

Keywords: Convolutional Neural Network, Building Energy Modelling, Feature Extraction, Computer Vision

ACM Reference Format:
Thomas R. Dougherty, Tianyuan Huang, Yirong Chen, Rishee K. Jain, and Ram Rajagopal. 2021. SCHMEAR: Scalable Construction of Holistic Models for Energy Analysis from Rooftops. In *The 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '21), November 17–18, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3486611.3486666

1 Introduction

Buildings are a centerpiece of climate change policy as they account for over 40% of the total primary energy use in the US and the EU [4]. As the world moves to curb global emissions, legislators looking to make informed policy can benefit from a more nuanced understanding of how broad retrofitting strategies might impact their cities.

Research into urban energy dynamics is typically pursued through one of two approaches. The first approach is purely data-driven. While this allows researchers to study macroscopic effects with more easily obtainable datasets, data-driven models have historically neglected the specific morphology of a buiilding. Instead, they have generally opted to statistically model a building using features like age, material, or HVAC (Heating, Ventilation, and Air Conditioning) equipment. Data-driven models can provide a good framework for the prediction of energy consumption but fall short of capturing energy dynamics from unique structural features. The second approach relies instead on simulation, which models a building's unique structural features with a high level of granularity through the aid of design software. This advantage makes simulation a more appropriate choice for retrofit analysis [5]. However, traditional simulation based

approaches require a high level of modeling expertise to craft an accurate model as well as curated data sets describing the structure. An example curated data set will contain extensive information about the structure. These may include features such as floor plans, window heights, wall thicknesses, material properties, and HVAC design. The high degree of technical expertise and extensive data requirements make simulation based energy modeling resource prohibitive for large scale analysis.

This project looks to bridge the gap between these two paradigms by proposing a system with both statistical significance and consideration of unique building features. The application of deep learning and computer vision allows for the automatic extraction of useful features from images. We take advantage of these properties to predict the energy consumption of the building from satellite photos and explore the significance of urban context as a predictor for energy consumption.

Modern research in the urban energy domain suggests that the accuracy of a building's energy model will plateau without consideration of the unique features from adjacent structures [24][9]. The plateau in accuracy is likely due to added uncertainties from light reflected off adjacent structures, HVAC exhaust from a neighbor, or nearby heat island effects from excess asphalt. Attempts have been made to bridge the gap of urban context in simulation based models through the use of machine learning [18], but this preliminary work is highly labor intensive and challenging to scale. We show that an entirely data driven approach based on satellite imagery captures the subtle details of modern urban energy dynamics while improving the statistical significance by folding a much larger sample size into the analysis. In this way, our study also explores the effects of urban context on energy consumption.

There have been a few attempts to map the energy consumption of a city region to its constituent buildings. This research has leaned on nighttime satellite photos, which can roughly approximate economic activity in emerging markets [16][27]. The use of nightlights was extended to approximate the economic activity of US markets[10], but has not received extensive utility as dense urban regions often saturate in nighttime satellite photos. In general, satellite based analysis for the task of energy prediction has primarily relied on linear regression to identify correlations. As such, most of the work has yet to consider nonlinear relationships which may exist in the urban domain.

Outside of energy prediction, other objectives in the urban domain have received more attention from the computer vision and data science community. This includes work to distill large scale spatial features of cities [17], classify regional characteristics [1], and to identify the connectedness of the grid [29], among others. In the design domain, researchers use computer vision techniques to distill stylistic elements of architecture [7][31], which identifies patterns of doorways and colors which uniquely identify a theme. Some other works use satellite imagery to map the energy infrastructure, such as solar photovoltaic panels[30] or oil refineries/petroleum terminals[26]. There is also work in the economic domain attempting to predict the income of a region based on satellite photographs using convolutional networks [23] or the density of housing in developing countries [25]. The utilization of satellite photography, which has global coverage, makes these works particularly promising. Some works also utilize multiple modalities of information and incorporate both images and textual data to analyze neighborhood patterns, such as combining human movement data as well as points of interest (POIs) data with satellite imagery[12].

While previous work has explored satellite imagery driven analysis in other urban domains, there has not been extensive work exploring the use of emerging computer vision methods (e.g., convolutional neural networks) for urban scale building-level energy modeling and prediction. Therefore, the objective of this study is to explore the potential utility of computer vision in urban energy modeling and introduce an approach for rapidly constructing a system capable of producing urban scale building energy models.

2 Methods

This work relies on satellite imagery to capture meaningful features from buildings, leaning on the assumption that aerial photos contain semantically significant information for urban analytics. If this assumption is correct, energy modellers may use the features extracted from satellite photos to predict the energy consumption of the building.

We break down the proposed approach into into four segments which are detailed below: (1) data collection and definitions, (2) train / validation / test split, (3) model architecture, and (4) saliency maps.

2.1 Data collection and definitions

There were two distinct classes of data used in the analysis, *satellite photos* and *contextual information*. We chose to maintain the system's scalability by selecting three data sources which have global coverage.

- **Building Energy.** A real, positive number quantifying the amount of electricity purchased by the building from the grid for the year of 2016.
- **Satellite Imagery.** The satellite photos capture cloudless representations of buildings from 2021 under the assumption that none of the buildings have changed since 2016. The photos then express the visible light emitted from the structures as a unique RGB array used in the analysis.
- **Urban Context.** The context is crafted by capturing the social function of adjacent buildings.

Building Energy

The City of New York provides our labels through the use of publicly available benchmarking data. This data set was made available by the New York City Benchmarking mandate [20], which in 2016 mandated that buildings over 50,000 square feet measure and report their annual consumption of resources. At the time of writing in 2021, this mandate now extends to buildings over 25,000 square feet. The EnergyStar++ project [2] provides a cleaned version of the data set combined with the PLUTO data [19]. The final dataset of 14,971 buildings also contains building heights, floor numbers, and gross floor area for each building in addition to consumption statistics.

There are yearly reports on different metrics such as direct and indirect carbon emissions, energy consumption, water consumption, district steam use, and natural gas use within the consumption statistics. The data set also provides an address for each building used to gather the satellite photo and context vector. Among these metrics, we selected electricity purchased from the grid as the dependent variable, measured in kBtu per year. We converted this term to MWh for all subsequent analysis (mean = 1,639, median = 514).

Among these 14,971 buildings, several buildings had no electricity consumption from the grid, and a long, thin tail of mega consumers. We identified potential outlieres as buildings with an energy consumption greater than 4,000 MWh and less than 200 MWh and excluded them from the analysis. After cleaning, there were 8,305 distinct points to use in training (mean = 832, median = 519). To accommodate the remaining skew of the data (skew = 1.89), the log of the data was used as the final predictor (skew = 0.59).

Satellite Imagery

The centroid coordinates for each building in the cleaned data set was obtained from the publicly available Building Footprints data set for New York [21]. We matched each building in the energy data set to their coordinates by using the building identification number (BIN). As part of cleaning the energy data set, we discarded buildings with no BIN number and only selected the first BIN number for buildings with multiple BIN numbers. We then used the centroid for the building as a the origin point to query two top-down satellite photos from Mapbox's satellite API for each latitude and longitude with a resolution of 500x500 pixels. We captured the first photograph for each building at zoom level 16.5, which is roughly 0.646 meters per pixel at the latitude of New York City. Each one of these macroscopic photos cover an area of around 104,000 m^2 and reveal nearby bodies of water, parks, and highways. We captured the second photo at a zoom level of 18.5 which measures around 0.422 meters per pixel. Covering around 45,000 m^2, the zoom 18.5 photos provide insights into the unique features of each building. These include elements of the building's HVAC system, adjacent trees, parking lots, and shadows cast by the structures. Of note, we also use zoom level 15 photos for some analysis, where each photo covers an effective area of 837,225 m^2 at 1.83 meters per pixel. The level of detail for each photo can be seen in Fig 1.

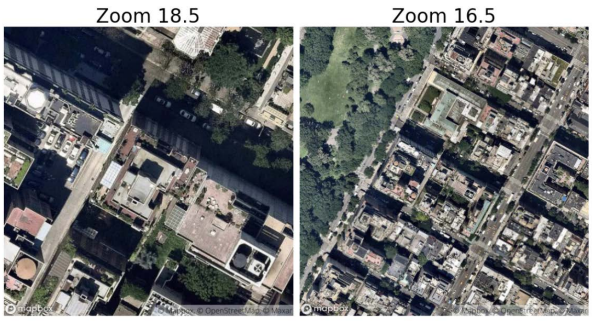

Figure 1. Level of Detail - Satellite Photos

Urban Context

In this section, we introduce a method to aggregate contextual information from the urban context and compress it into a fixed size vector. The influence of occupancy patterns on building energy consumption drives the motivation for the context vector [6]. The fixed length context vector is intended to describe the potential type of occupancy based on the social function of nearby establishments. The establishments are typically businesses like offices, barbershops, schools, etc. From this, the system may recognize that if offices surround the building, it is unlikely to be heavily populated on weekends as it might have a higher possibility of being an office itself.

To construct the context vector, we first query Google's search engine for a list of prominent establishments within radius R of the structure of interest. According to the Google's documentation, the radius input simply biases the search results to elements within this area and typically returns about 20 results for a query. Each establishment descriptor is a unique string e_i which comes from a set of establishment names $E = \{e_1, e_2, ..., e_N\}$. The full set of establishment names is maintained internally by Google, and the full size of this set is not known. Google's search query may register multiple establishments within a single building, so the number of nearby buildings may not equal the number of nearby establishments.

The next step is to then encode this list of establishments for use in subsequent analysis. A naive option would be to take the sum of their one hot embeddings, where the maximum length of this summed vector would be the same as the size of the set of unique establishment names. As the length of the establishment names is not known, we instead opt to hash the terms into a fixed size vector. The hash function takes an arbitrary input, in our case the establishment description string, and uniquely maps this input onto a number which is guaranteed to fall within the length of our vector.

We then compute the context vector as the sum of hashed vectors for each establishment within the radius R. An example of the context barcode can be seen in Fig.2. For visualization, this example uses a vector length of 20 instead of the typical 1,000.

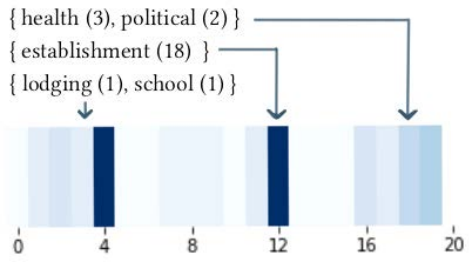

Figure 2. Context Barcode

The downside of hashing is that there's a nonzero chance two different terms might hash into the same bucket, mangling their representation. The probability of collision increases with the number of unique descriptor terms or with a smaller hash vector. The expected number of collisions can be roughly approximated as $\approx n - k + k\left(1 - \frac{1}{k}\right)^n$ [8]. There are k = 1,000 buckets in the hash vector and $n \approx 110$ unique descriptors to be hashed for our system, indicating that our expected number of collisions is close to 6. As we sum embeddings, each element will still have representation regardless of collision.

2.2 Training Setup

We trained the model using PyTorch [22]. The data was split into three classes, as outlined in the table 1. We used a batch size of 64 as a default for most of the computation. We tracked the models using Weights and Biases [3].

Table 1. Train/Validate/Test Split

Class	Percent	Count
Training	70%	5,813
Validation	15%	1,246
Testing	15%	1,246

2.3 Model Architecture

The primary architecture used in this analysis, a convolutional neural network, is evaluated against our null model. The two models are defined as follows.

Null Model. The null model simply predicts the mean energy of the training set, which is defined as $\frac{1}{n}\sum_{n=1}^{n} e_i$.

Null+ Model. The null plus builds a linear regression of the form $\hat{\mathbf{Y}} = \alpha + \beta\mathbf{X}$ using features collected in the PLUTO dataset. The features used in the regression were the construction year, roof height (ft), and ground level (ft).

Convolutional Neural Network. The convolution is mathematical operation which enables us to artificially simulate the structure of neuron firings in the visual cortex [11] when utilized in the image domain. Convolutions, typically referenced as *filters* for their behavior of filtering for certain characteristics, can do this by capturing low level details from the images. The first filters which interact with the images are quite simple, looking for patterns of colors or edges. The output of the images interacting with our first four convolutional filters can be see in Fig.3.

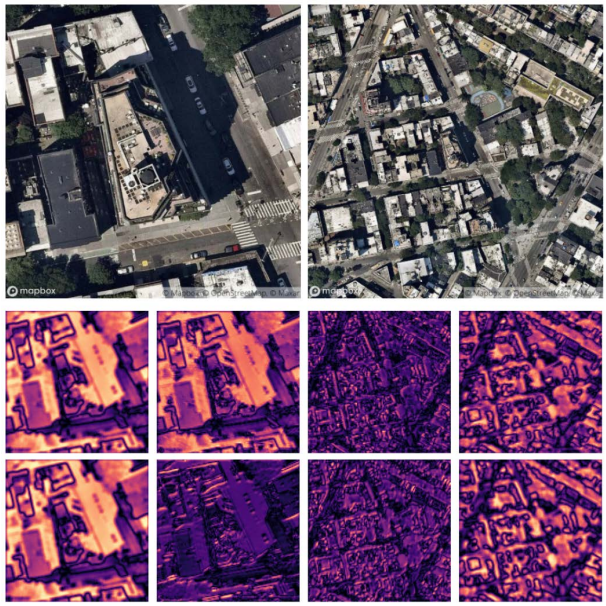

Figure 3. Images interacting with first four convolutional filters

The composition of filters with other filters permits the composition of abstract representations. This repeated stacking of convolutional filters creates a "deep" network that enables the representation of more complex concepts, also called latent structure. A network of stacked convolutional filters, made famous for their implementation in convolutional networks, have been used with success in pattern recognition applications as varied as autonomous driving, analysis of particle accelerator data, and audio spectrogram analysis [15].

Architectural Overview.
Our proposed system manages two streams of image data. Each image stream is of dimension N x 3 x H x W, where N is the cardinality of the data set, H is the height of each image and W is the width of each image. We normalize each image along color dimensions by subtracting the mean of each channel and dividing by the standard deviation before

passing into the ResNet systems. We resized each image to a size of 256 x 256 pixels before analysis. This resolution was selected as the images still capture semantically significant features yet are small enough to maintain tractability in computation.

The system is also required to accommodate the sparsely filled context vector, which has a dimensionality of N x 1,000. We use two identical models of the ResNet34 architecture, and utilize them for our two streams of images. These architectures are complemented by a fully connected (FC) net which is used to preprocess the context vector. The fully connected network is a vanilla neural network, which can be thought of as a series of linear operators and nonlinearities [14]. The outputs of each of these systems then gets concatenated as a 2,050 length vector then fed into a final fully connected network. The final fully connected network has five layers (2,050; 500; 100; 50; 5), each of which sequentially reduces the vector's dimensionality until it is represented as a single floating point number. We use this floating point number to compute the loss of the system by comparing it to the actual value recorded by the building.

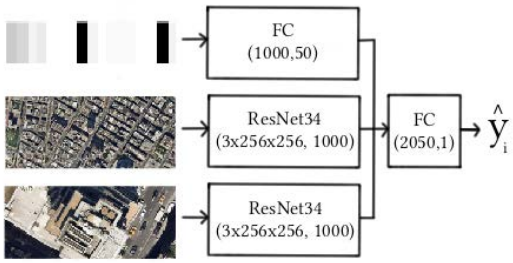

Figure 4. System Architecture

We also conducted feature significance tests to determine the relative importance of different data sources in prediction. There were four potential data sources for each location, with three satellite images and one context vector (radius 300 meters). The satellite imagery options were zoom 15, zoom 16.5, and zoom 18.5.

Loss Function The *Euclidean Norm* (L2) was selected for use as the loss metric, defined in equation 1.

$$\mathcal{L} = \sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

Optimizing against the euclidean norm seeks to globally reduce the estimated error of the system. As such, the error is more likely to be evenly distributed across the data set. Due to this behavior, massive buildings are likely to have smaller relative error.

Optimizer Adam was selected as our optimizer for its stability, variable learning rate, and momentum behavior with consideration of historical gradient [13].

2.4 Saliency Maps.

In addition to the scalability provided by satellite imagery, computer vision systems can provide intuitive feedback on their decision-making through modern visualization techniques. To highlight the potential applications for informing urban and decarbonization policy, this section explores the use of *saliency maps* to visualize the focus of the system as well as gain an intuition as to the relative value of neighborhood level information. The section then explores the applications of computer vision in grouping similar structures based on external and regional features which may be driving energy consumption.

Saliency maps were introduced in 2014 by Karen Simonyan [28] as a way to visualize the key points within an image that are driving decision making for the machine. As a summary, we would like to have an intuition into how much the prediction of our system might change in relation to a single pixel of the source image. If we first stretch out the pixels of an image into a single vector of length HxW, we have a set of pixels $P = \{p_0, p_1, ..., p_{HW}\}$. If we were to build a linear regression $S(P) = w^T P + b$ where our prediction for energy is defined by the linear function S, then the estimated influence of a single pixel on prediction is found by computing the partial derivative $\partial S = w^T \partial P$. Our architecture is highly nonlinear, so we instead attempt to find an approximation $\partial S \approx w^T \partial P$. The primary shift in logic is that instead of looking to discover ∂S through perturbations $P \pm \delta p$, we select to bother S with small perturbations $S \pm \delta s$. We can then backpropogate the perturbed scores into the original image and examine what features would need to change to significantly influence the prediction.

Figure 5. Saliency Map

From this example saliency map in Fig 5, we see that the system's focus on asphalt surfaces matches our intuition that paved roads might be locally influencing the temperature of the region. In this way, saliency maps provide intuition into to the role of discrete features in energy prediction.

From our knowledge, there exists no reference utility which indicates the importance of context in unique regions while modeling building energy consumption. In addition to improving the perceptibility of the system's decision-making,

we can use saliency maps to better understand the role of urban context in energy prediction. To approach this topic, we look capitalize on the utility of saliency maps to measure the relative importance of pixels as we move away from the centroid of the building by exploring the change in *saliency density*. For a set $P = \{p_1, p_2, ..., p_N\}$ of pixels contained by a circle of radius R measured in meters, there is a corresponding set of saliency values $S_p = \{s_0, s_1, ..., s_N\}$ which has a one to one mapping to the indices of the pixels. That is, for each pixel p_i there is a saliency value s_i. We define α to be the area of a single pixel, which is assumed to be uniform for every pixel. The *saliency density* ρ is then defined by equation 2.

$$\rho = \frac{1}{\alpha N} \sum_{i=1}^{N} s_i \tag{2}$$

We computed the saliency density using a system trained against the zoom 18.5 and zoom 15 images. The intuition behind this design decision is that microscopic (zoom 18.5) photos will provide the system with information on specific features of the structure of interest, which allows macroscopic (zoom 15) photos to focus more on contextual features of the region. Although use of the zoom 15 images did not have the most value in prediction, we selected the zoom 15 photos for their extensive coverage which enables analysis to a greater region. The saliency was then extracted for every macroscopic photo and used in the density analysis.

Figure 6. Saliency Density

By examining how the density of saliency changes with increasing contextual radius, as seen in Fig.8, we can gain intuition as to how much context is sufficient to accurately predict energy purchased from the grid. Saliency density computed for a uniformly increasing radius should produce a line with no slope if saliency was distributed uniformly throughout the image. Thus, any deviation from a line with no slope provides intuition into the relative role of urban context.

2.5 Prototype Identification.

Through the process of predicting a single term to describe the energy consumption of the building, we compress the large 2,050 dimensioned vector through a series of smaller layers. The hierarchical compression allows for more nonlinear relationships to be learned, thus increasing the expressivity of the model. We can take advantage of this architecture by artificially imposing a low dimensional vector prior to prediction, forcing each batch of data to flow through this low dimensional vector. This last batch is then passed through a single linear operator W which then determines how much each prototype affects the final prediction.

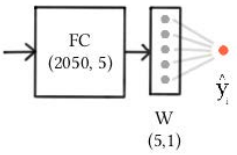

Figure 7. Artificial Bottleneck

The artificial bottleneck provides incentive for the system to compress the incoming data as much as possible such that it can still make accurate predictions. The system might, for example, learn to count the number of trees in a picture and store them in one of its precious five slots to use in prediction time. Although it cannot recreate the image of the tree from this count, it can still achieve high accuracy in prediction by storing the quantity of trees in a slot of the compressed vector. In this way, the system learns valuable prototypes for the task of prediction, where a prototype might map to a discrete concept like the density of vegetation. The final prediction step then develops scores against each prototype where a high score indicates that it shares a high level of similarity to the prototype. We can use these scores to cluster for similar buildings, grouping things nicely for policy making. If we are interested in finding buildings that can most improve their energy efficiency through the use of double paned windows, we can craft policy to target buildings that score highly against the window prototype.

Before we can cluster similar locations based on their prototype signatures, we want to build an intuition as to the content of each prototype. This is a particularly challenging task with convolutional neural networks as the bottleneck vector can expand a single latent concept into multiple nodes or conversely compress multiple latent concepts into a single node. To shed light into the system's inner workings, we manipulate the saliency maps a bit to express themselves only for a unique node in the bottleneck layer. Formally, we can do this by modifying the gradient as it flows through the bottleneck layer and into the rest of the system. Our system can technically by defined as a directed, acyclic computational graph. To train the system, we optimize against this computational graph to find a mapping between our source data and the projected output $\hat{y}$. The optimization process utilizes back propagation, which nicely interacts with each element in our network. Starting with our loss function defined in eq.1 as the "mouth of the river", back propagation

conveniently takes the upstream gradient and multiples it with the local gradient for each layer in the system to pass downstream. In this work, we use back propagation to discover the saliency map with respect to each source image **I** by using the back propagation chain rule defined in equation 3.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{I}} = \frac{\partial \mathcal{L}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{Z_1}} \frac{\partial \mathbf{Z_1}}{\partial \mathbf{Z_2}} \cdots \frac{\partial \mathbf{Z_n}}{\partial \mathbf{I}} \tag{3}$$

Traditional notation for back propagation does not use W to reference the intermediate layers, but we choose to explicitly show how the bottleneck layer W interacts with the chain rule. This explicit notation helps in visualizing our modification to the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{W}}$. We can isolate for the gradient with respect to a single node W_i, and thus find a saliency map with respect to a single prototype, by wiping away the rest of the local gradients in $\partial \mathbf{W}$ through the use of a vector (ξ) populated with one value.

$$\xi^T \frac{\partial \mathcal{L}}{\mathbf{W}} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \begin{bmatrix} \frac{\partial \mathcal{L}}{w_1} \\ \frac{\partial \mathcal{L}}{w_2} \\ \vdots \\ \frac{\partial \mathcal{L}}{w_5} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{L}}{w_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4}$$

This allows us to build saliency maps with respect to a single element in the bottleneck. This untangles the influence of other nodes on the saliency map and permits us to interpret the regional focus of each element. We once again pass every image through the system and back propagate to build saliency maps. The new dimensionality of our output is 5xNxHxW, which reflects that we have 5 unique images for each macro and micro photo of the structure. To identify the significance of the node, we fixate on a single element and identify which images maximally fire the node.

3 Results

Satellite photos and contextual features are found to be valuable predictors in improving the quality of prediction for energy consumption, reducing the estimation errors by nearly one third compared to the null model. To build this result, the model's hyper-parameters like learning rate and batch size were tuned against the training data.

3.1 Hyperparameter Selection

A series of trials were conducted to assess the influence of the learning rate and batch size on system performance. The results of the trials can be seen in table 2. These results led us to select an epoch size of 20 and learning rate of 0.1 as our default parameters for the rest of the analysis.

3.2 Model Summary

Null Model. The null model, which was predicting the mean of the training set, achieved a validation loss of 0.57.

Table 2. Summary of Hyperparameter Tuning

Epochs	Learning Rate	Validation Loss
40	0.08	0.33
40	0.10	0.31
20	0.08	0.33
20	0.10	**0.31**

Null+ Model. The null plus model achieved a validation loss of 0.36. If we also include the gross floor area (ft^2) of the structure, as defined by the NYC Department of Finance, this validation loss drops to 0.29.

Convolutional Neural Networks. We tested different permutations of features to examine their influence on validation loss. Each row of table 3 indicates a permutation of data sources, where a check mark indicates that the data element was used in training. For each permutation of data, a learning rate of 0.1 was used to train the model over 20 epochs with a batch size of 64. The model with the best validation score was recorded for each permutation.

Table 3. Feature Selection

15	16.5	18.5	Context	Validation Loss
				0.57
✓				0.41
	✓			0.38
		✓		0.32
			✓	0.45
✓	✓			0.35
✓	✓		✓	0.35
	✓	✓		**0.31**
	✓	✓	✓	0.33
✓		✓		0.33
✓		✓	✓	0.34

From these results, the final version of the trained ResNet architecture used two photos, zoom 16.5 and zoom 18.5. Using this amalgamation of features, the system achieved a validation loss of 0.31 after running for 14 epochs. It achieved a similar result against the test set, with a loss of 0.32. If we translate this loss into a prediction of MWh, we can be reasonably confident that the model with a validation loss of 0.31 will produce a prediction between 3,030 MWh and 1,598 MWh for a building which consumes 2,200 MWh. This prediction limit is contrasted to the null model which will always predict the training set mean of 609 MWh.

3.3 Saliency and Urban Context Results

As a case study into the functionality of saliency maps in measuring the role of urban context, we isolate two regions

of New York City and explore the role of context in energy prediction. In this small sample, we see that context in the dense urban core of Manhattan plays a more significant role in prediction than in the more sparsely developed boroughs of Queens and Brooklyn. We find that context is typically around 26% more valuable as a predictor in Manhattan than in Queens.

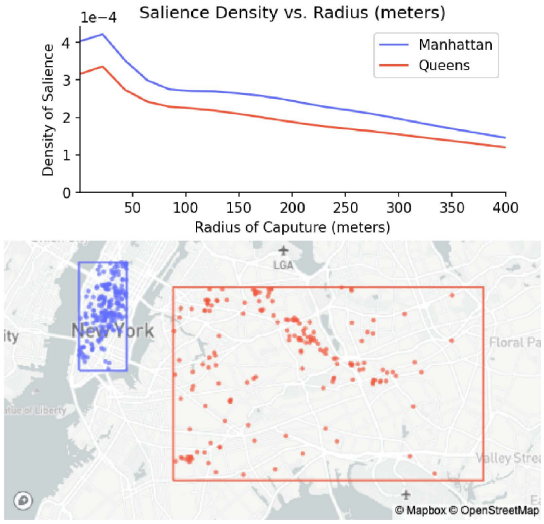

Figure 8. Comparison of Urban Context Importance

3.4 Prototype Results

After computing the saliency with respect to each node in the bottleneck layer, we focus on one location to demonstrate the value. We pull two of the saliency maps for a single location which targeted unique regions in the macroscopic image 9. These demonstrate that the nodes capture multiple concepts within the urban texture and saliency maps can be used to help tease out the representations captured.

As an example of how we might look to collect intuition from this system using the saliency maps, we recall that the prediction for the energy of the system first scores the photo against one of five prototypes. In this example, we will reference these scores as the vector V. The final prediction can thus be found through the dot product of V and W. Thus we can examine both the score of each image against the prototype v_i, as well as the estimated influence of the prototype on the overall prediction of energy w_i.

In our example saliency map, we might intuit that node one attempts to target residential neighborhoods to the south and west of the image. For this example, node one collected a prototype score (v_0) of -1.60 and the trained model has an activation weight (w_0) of -0.95. Armed with this information, we conclude that the concept captured the node is estimated to increase model's prediction by roughly 1.53. If we instead take the dot product of the activation weights W with the prototype scores and average across each node,

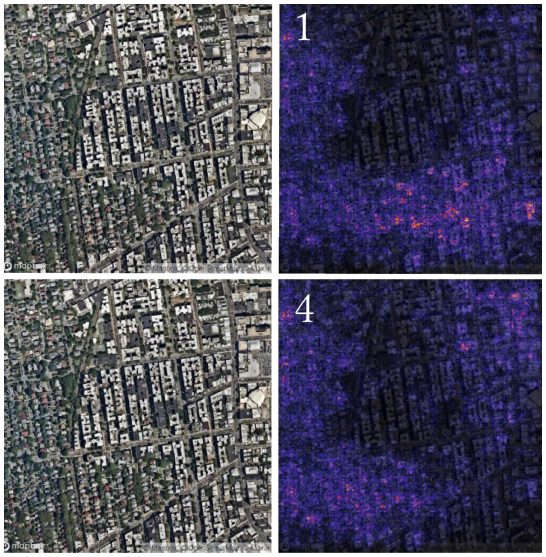

Figure 9. Nodal Saliency. Top: Saliency from Node 1, Bottom: Saliency from Node 4.

we can infer the typical effect of the node on prediction. In our example, the average nodal influences are given as: [1.33, 0.30, 1.47, 0.11, 1.11], indicating that three of the nodes are responsible for a majority of the influence in prediction. The magnitude of the terms and the semantics captured by the nodes gives relative significance to the features which should be prioritized for the city's retrofit strategy.

3.5 Regional Bias

The system demonstrates a tendency to underestimate predictions for tall buildings, namely Lower Manhattan and Midtown Manhattan. We also see that it is overestimating in Queens, likely in an attempt to reduce bias on the mean prediction for the training set. As the distribution of energy purchased by buildings in New York City is heavily right skewed, attempts to reduce bias result in the overestimation of energy in smaller buildings.

In addition to overlaying the error on a map, we can explore trends in the error to better understand why the system is failing. The addition of PLUTO data to the EnergyStar++ data set provides building height, building age, and floor area. We omitted these terms during training in favor of developing a more scalable system, but we can now use them to explore correlations with the error as seen in table.4. We normalized the data terms by subtracting the mean and dividing by the standard deviation prior to determining the correlation.

4 Discussion, Conclusions, Future Work

In this work, we demonstrate a method of extracting meaningful features from readily available satellite data to estimate the electricity a facility will purchase from the grid.

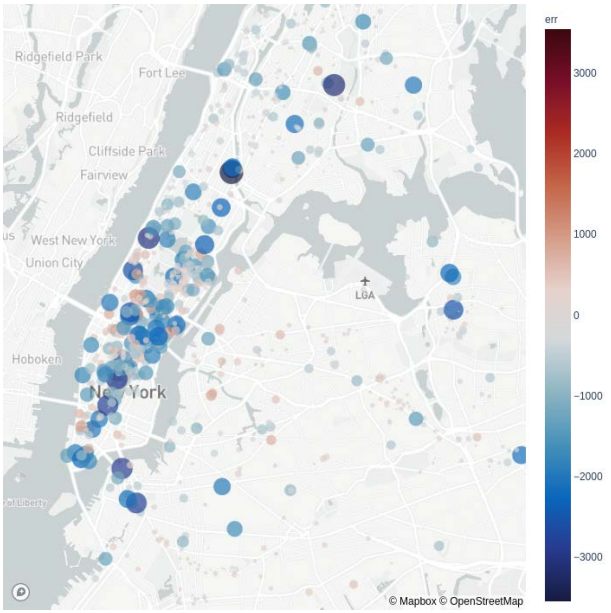

Figure 10. Regional Bias

Table 4. Correlation of Error with Features

	Correlation
Height	-0.322
Area	-0.583
Ground Level	0.071
Year	-0.146

We achieve this by using a framework for the rapid gathering and processing of pertinent data in the urban domain. We demonstrated that each of these features is valuable in improving our vision system's prediction accuracy through the use of our feature significance tests. We found that the context vector was useful to improve prediction, but images were the driving feature in reducing error.

In terms of prediction quality, we show that our proposed model easily outperforms a null model for New York City and can even achieve comparable results to the Null+ model produced via linear regression against PLUTO features. The inclusion of gross floor area in the Null+ model greatly improved the quality of prediction beyond the accuracy we can achieve with the vision system alone. In general, it seems like the system had limited success in accurately estimating building area and height from satellite images. This defect in the model seems to be driving prediction error. As our error is negatively correlated with building area and height, the idea that our system cannot capture structural scope is reinforced.

This also passively indicates that the system was instead able to reduce its error by learning the semantics of neighborhoods throughout New York City. We see evidence of this behavior through interpretation of the saliency maps with respect to individual nodes. We show in Fig.9 that our system seemed to passively segment between the suburbs and midrise buildings before making a prediction. The lack of significant value from the context vector in improving prediction accuracy might be another indication of semantic extraction from images. It seems like the vision system might have already used satellite images to build an internal representation of the information encoded in the context vector, which reduced their value as a feature.

In addition to the extraction of pertinent semantic features, we applied a novel method of visualizing the captured semantics by modifying the construction of saliency maps. We exploited the bottle necking feature of the final fully connected network layer to build an intuition into building *prototypes*, which characterize buildings based on primary drivers of energy consumption. To shed light into the information captured by prototypes, we adapted saliency maps to expose the latent concepts captured by each prototype. In our case study of regions in New York City, we demonstrate that saliency maps can be used as a proxy to identify the role of urban context and provide insight into how density and distance impact energy consumption.

While the system performs reasonably well on New York City for the year of 2016, climate was ignored in this study. As weather plays a significant role in driving the energy performance of buildings, subsequent research will need to incorporate weather for a more robust analysis.

Finally, we note that this architecture is not limited to the prediction of energy purchased from the grid. Although energy prediction was used as a preliminary example to expose the potential of modern computer vision systems in urban analytics, the structure of our system could rapidly be transferred to predicting other target metrics in the urban domain. This may include things like walkability, occupancy trends, or materials classification of buildings. New York City provided a rich source of data, but further study need not be limited to any particular city. The high availability of satellite data makes this approach promising to better understand energy consumption in data sparse metropolitan regions of the Global South and rest of the world.

References

[1] Adrian Albert, Jasleen Kaur, and Marta Gonzalez. 2017. Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale. *arXiv:1704.02965 [cs]* (Sept. 2017). http://arxiv.org/abs/1704.02965 arXiv: 1704.02965.

[2] Pandarasamy Arjunan, Kameshwar Poolla, and Clayton Miller. 2020. EnergyStar++: Towards more accurate and explanatory building energy benchmarking. *Applied Energy* 276 (2020), 115413. https://doi.org/10.1016/j.apenergy.2020.115413

[3] Lukas Biewald. 2020. Experiment Tracking with Weights and Biases. https://www.wandb.com/ Software available from wandb.com.

[4] Xiaodong Cao, Xilei Dai, and Junjie Liu. 2016. Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade. *Energy and buildings* 128 (2016), 198–213.

[5] Yixing Chen, Tianzhen Hong, and Mary Ann Piette. 2017. Automatic generation and simulation of urban building energy models based on city datasets for city-scale building retrofit analysis. *Applied Energy* 205 (Nov. 2017), 323–335. https://doi.org/10.1016/j.apenergy.2017.07.128

[6] Caroline M Clevenger and John Haymaker. 2006. The impact of the building occupant on energy modeling simulations. In *Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada*. 1–10.

[7] Carl Doersch, S. Singh, A. Gupta, Josef Sivic, and Alexei A. Efros. 2012. What makes Paris look like Paris? *ACM Transactions on Graphics (TOG)* 31 (2012), 1 – 9.

[8] Herbert Edelsbrunner and Brittany Fasy. 2009. *CPS 102: Discrete Mathematics for Computer Science - Lecture 18.* Retrieved July 21, 2021 from https://courses.cs.duke.edu/cps102/spring09/Lectures/L-18.pdf

[9] Enrico Fabrizio and Valentina Monetti. 2015. Methodologies and Advancements in the Calibration of Building Energy Models. *Energies* 8, 4 (2015), 2548–2574. https://doi.org/10.3390/en8042548

[10] Derek Fehrer and Moncef Krarti. 2018. Spatial distribution of building energy use in the United States through satellite imagery of the earth at night. *Building and Environment* 142 (Sept. 2018), 252–264. https://doi.org/10.1016/j.buildenv.2018.06.033

[11] D. H. Hubel and T. N. Wiesel. 1959. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology* 148, 3 (Oct. 1959), 574–591. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1363130/

[12] Porter Jenkins, Ahmad Farag, Suhang Wang, and Zhenhui Li. 2019. Unsupervised Representation Learning of Spatial Data via Multimodal Embedding. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1993–2002. https://doi.org/10.1145/3357384.3358001

[13] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[14] Anders Krogh. 2008. What are artificial neural networks? *Nature Biotechnology* 26, 2 (Feb. 2008), 195–197. https://doi.org/10.1038/nbt1386 Bandiera_abtest: a Cg_type: Nature Research Journals Number: 2 Primary_atype: Reviews Publisher: Nature Publishing Group.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (May 2015), 436–444. https://doi.org/10.1038/nature14539 Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7553 Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Computer science;Mathematics and computing Subject_term_id: computer-science;mathematics-and-computing.

[16] Charlotta Mellander, José Lobo, Kevin Stolarick, and Zara Matheson. 2015. Night-Time Light Data: A Good Proxy Measure for Economic Activity? *PLoS ONE* 10, 10 (Oct. 2015). https://doi.org/10.1371/journal.pone.0139779

[17] Vahid Moosavi. 2017. Contextual mapping: Visualization of high-dimensional spatial patterns in a single geo-map. *Computers, Environment and Urban Systems* 61 (Jan. 2017), 1–12. https://doi.org/10.1016/j.compenvurbsys.2016.08.005

[18] Alex Nutkiewicz, Zheng Yang, and Rishee K. Jain. 2018. Data-driven Urban Energy Simulation (DUE-S): A framework for integrating engineering simulation and machine learning methods in a multi-scale urban energy modeling workflow. *Applied Energy* 225 (Sept. 2018), 1176–1189. https://doi.org/10.1016/j.apenergy.2018.05.023

[19] NYC Department of Planning. 2017. *Primary Land Use Tax Lot Output (PLUTO).* Retrieved July 20, 2021 from https://www1.nyc.gov/site/planning/data-maps/open-data.page#pluto

[20] NYC Mayor's Office of Sustainability. 2017. *Local Law 84 (LL84).* Retrieved July 20, 2021 from https://www1.nyc.gov/html/gbee/html/plan/ll84_scores.shtml

[21] NYC OpenData. 2017. *New York City Building Footprints Data Set.* Retrieved July 21, 2021 from https://data.cityofnewyork.us/Housing-Development/Building-Footprints/nqwf-w8eh

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG]

[23] Simone Piaggesi, Laetitia Gauvin, M. Tizzoni, C. Cattuto, N. Adler, S. Verhulst, A. Young, Rhiannan Price, L. Ferres, and A. Panisson. 2019. Predicting City Poverty Using Satellite Imagery. In *CVPR Workshops*.

[24] Anna Laura Pisello, John E. Taylor, Xiaoqi Xu, and Franco Cotana. 2012. Inter-building effect: Simulating the impact of a network of buildings on the accuracy of building energy performance predictions. *Building and Environment* 58 (Dec. 2012), 37–45. https://doi.org/10.1016/j.buildenv.2012.06.017

[25] Rahman Sanya and Ernest Mwebaze. 2020. Identifying patterns in urban housing density in developing countries using convolutional networks and satellite imagery. *Heliyon* 6, 12 (Dec. 2020), e05617. https://doi.org/10.1016/j.heliyon.2020.e05617

[26] Hao Sheng, Jeremy Irvin, Sasankh Munukutla, Shawn Zhang, Christopher Cross, Kyle Story, Rose Rustowicz, Cooper Elsworth, Zutao Yang, Mark Omara, Ritesh Gautam, Robert B. Jackson, and Andrew Y. Ng. 2020. OGNet: Towards a Global Oil and Gas Infrastructure Database using Deep Learning on Remotely Sensed Imagery. arXiv:2011.07227 [cs.CV]

[27] Anja Shortland, Katerina Christopoulou, and Charalampos Makatsoris. 2013. War and famine, peace and light? The economic dynamics of conflict in Somalia 1993-2009. *Journal of Peace Research* 50, 5 (2013), 545–561. https://doi.org/10.1177/0022343313492991

[28] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv:1312.6034 [cs.CV]

[29] Qinghu Tang, Zhecheng Wang, Arun Majumdar, and Ram Rajagopal. [n.d.]. Fine-Grained Distribution Grid Mapping Using Street View Imagery. ([n. d.]), 6.

[30] Jiafan Yu, Zhecheng Wang, Arun Majumdar, and Ram Rajagopal. 2018. DeepSolar: A Machine Learning Framework to Efficiently Construct a Solar Deployment Database in the United States. *Joule* 2, 12 (2018), 2605–2617. https://doi.org/10.1016/j.joule.2018.11.021

[31] B. Zhou, L. Liu, A. Oliva, and A. Torralba. 2014. Recognizing City Identity via Attribute Analysis of Geo-tagged Images. In *ECCV*.