Predicting Building Energy Efficiency Using Google Aerial & Street View, OpenStreetMap, Land Surface Temperature Data

Athena Deng (akdeng), Julia Gu (jgu41), Eric Zheng (ezheng37)

Original paper

Introduction

This project aims to estimate building energy efficiency using publicly available data, including street view images, aerial view images, building footprint, and satellite-derived Land Surface Temperature (LST). For ground truth labels, we rely on the EU's EPC registry data. If results are reliable, such a model could be used to identify buildings – even general geographical areas – where buildings are energy-inefficient. This would be immensely useful for sustainable building initiatives and decarbonization plans. Such a tool would be immensely useful in quickly and cheaply identifying high-yield retrofits compared with traditional methods like on-site assessments. We chose this paper because of its potential societal benefits and because it would allow us to apply many concepts we've covered in class. The paper combines many different modalities of data as well as multiple types of machine learning models, giving us the chance to work across different techniques.

This study is a binary classification problem. In this task, we are classifying buildings as energy efficient (EPC rating A-D) or energy inefficient (E-G).

The authors of this related paper used a convolutional neural network (CNN) pre-trained on the ImageNet dataset to convert images into a usable vector of features. The image vectors were then combined with features generated from the EPC data to create a collective feature vector, and then fed the collective vector into a 4-layer deep neural network to output a non-binary classification. The data used in this study included EPC data containing information about each building's features, the UKBuildings dataset from EDINA Geomni Digimap Service, which provided building footprints, and Google Street View (GSV) facade images taken from Scotland, UK. Their model achieved an admirable final performance of 86.8% accuracy on the test set, significantly more difficult considering the variety of classes outside of binary classification.

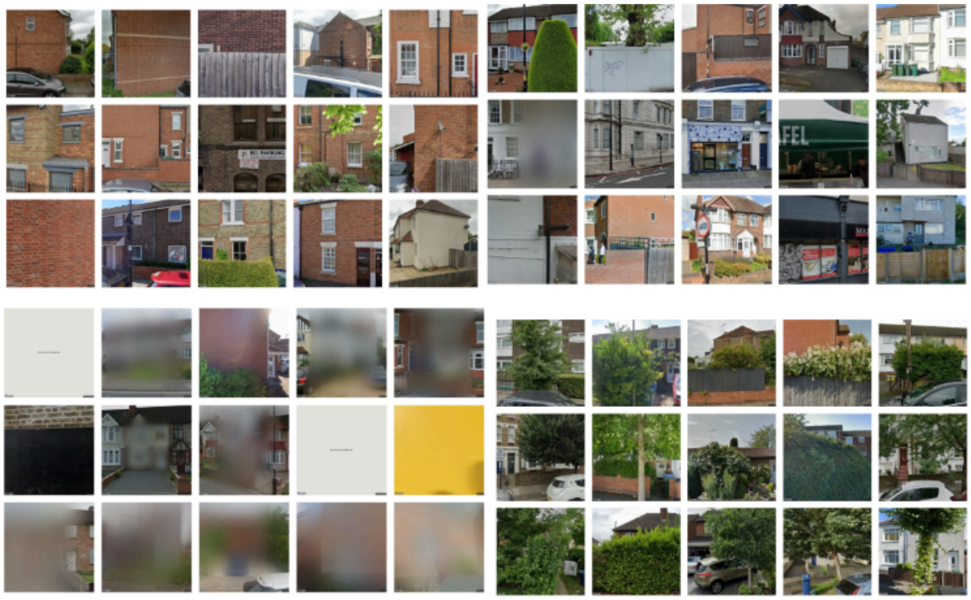
At the time of our study, there were no other public implementations.

Methodology

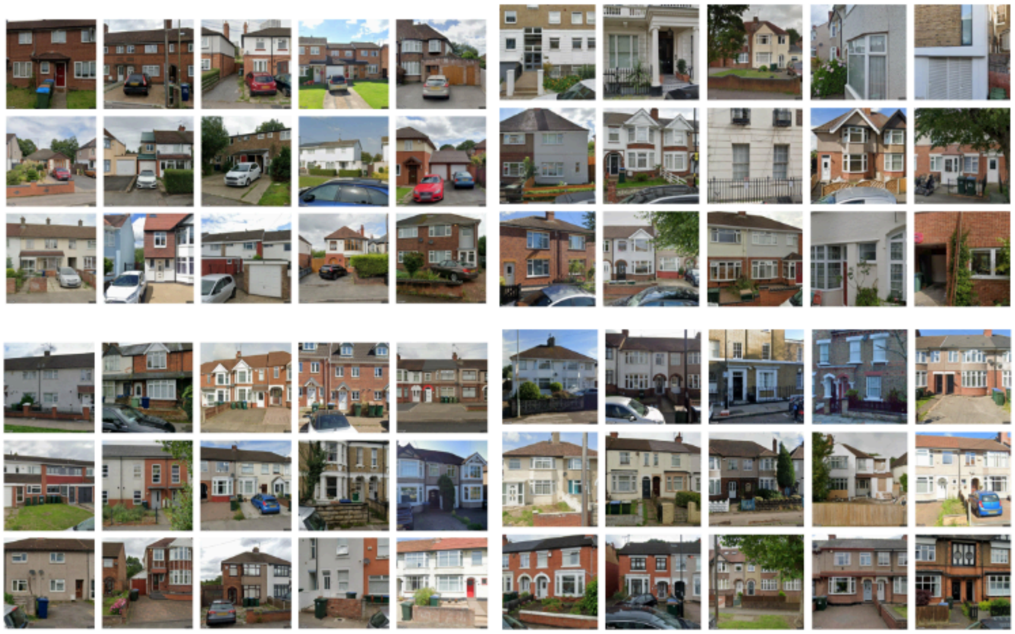
The data used in this study consists of images retrieved from Google Street View (GSV) and Aerial View (GAV), coordinates from OpenStreetMap building footprints, temperatures from Landsat-8 Land Surface Temperature (LST), and EPC ratings from the EU Energy Performance Certificate (EPC) registry.

The coordinate, temperature, and EPC ratings were already compiled in a dataset by the paper's authors, however, we needed to download and clean the approximately 80,000 images representing 40,000 buildings separately.

To preprocess these images, we used an embedder to encode them and k-means clustering to remove unhelpful images from the data (e.g., plain walls, obstructed views, unavailable buildings). After cleaning, we concatenated the embeddings with the rest of the structured data for their corresponding buildings. Below are examples from four of six removed clusters from the dataset:



In comparison, below are examples from four clusters that were kept:



To train the model, we split our data into three sets—train, test, and val—with sizes of approximately 23,000, 3,500, and 3,500, respectively. After we processed the images through the pre-trained InceptionV3 model and added the LST and building area data, we had our final feature vectors. We stored these feature vectors in a numpy array, and the labels in an array with matching indexes. Then, we used the training data on a binary classification MLP, which outputs the final prediction.

Each building is represented by a concatenation of four features: 2048-dim street view image embeddings, 2048-dim aerial image embeddings, 3-dim LST, and 1-dim building area/footprint (Fig. 3). This 4100-dim vector was then passed through a linear prediction head, a single, fully connected linear layer with a sigmoid function to output the probability of a building being energy efficient, or an MLP prediction head, with a hidden layer with 8 neurons and ReLU activation, 50% dropout, and then a final output layer with sigmoid activation. Both are trained with an Adam optimizer, a batch size of 16, a learning rate of 0.0001, and utilize class weights to address the label imbalance. The original authors utilized a grid search to optimize these hyperparameters, thus, we used the same ones.

Evaluation/Results

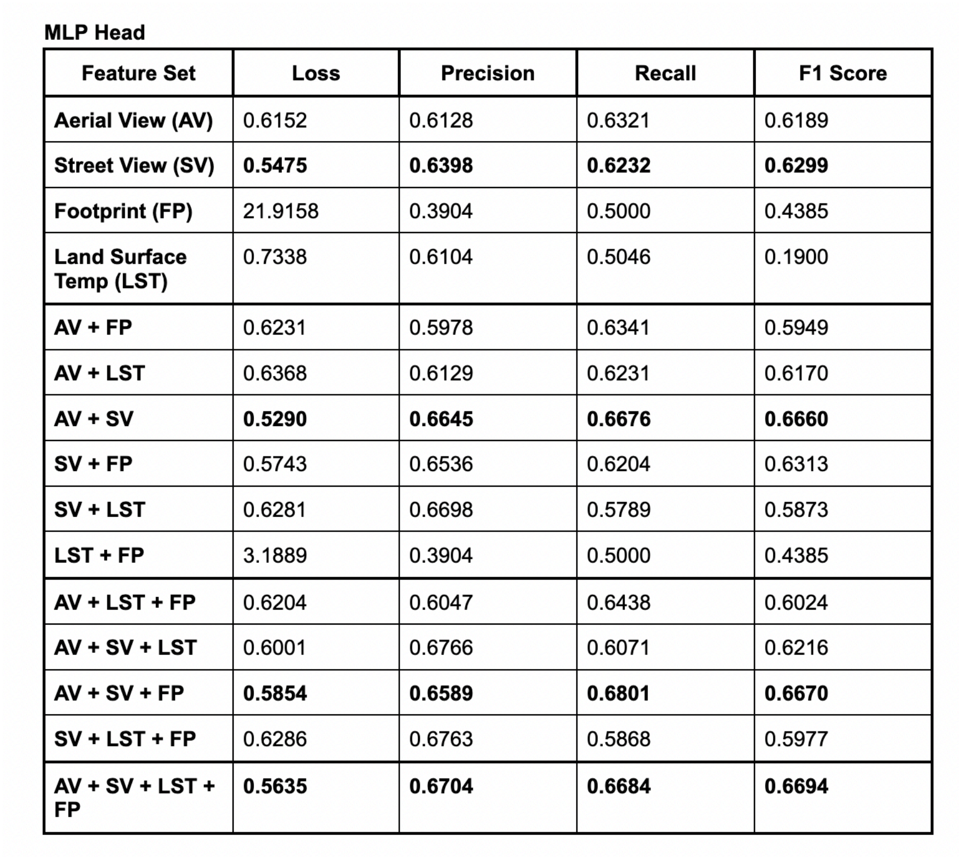
We calculated the precision, recall, and F1 scores by running the model on the test dataset. The MLP head performed with a loss of 0.5770, precision of 0.6655, recall of 0.6858, and F1 score of 0.6736. On the other hand, the linear head performed with a loss of 0.5194, precision of 0.7163, recall of 0.5804, and F1 score of 0.5886.

Like the authors of the original paper, accuracy did not serve as our preferred metric due to the unbalanced data. The authors of the paper were hoping to find that the LST data contributes positively to the overall scores of the model. They did find this by quantifying their results with an ablation study.

We aimed for the following set of goals. Our base goal was to finish training the MLP so that the model trains and runs smoothly without an accuracy threshold. Our target was to get within ~5% of the authors' F1 score. Our stretch goal was to try running some of the comparison tests with the baseline models the authors used, or conducting the ablation tests.

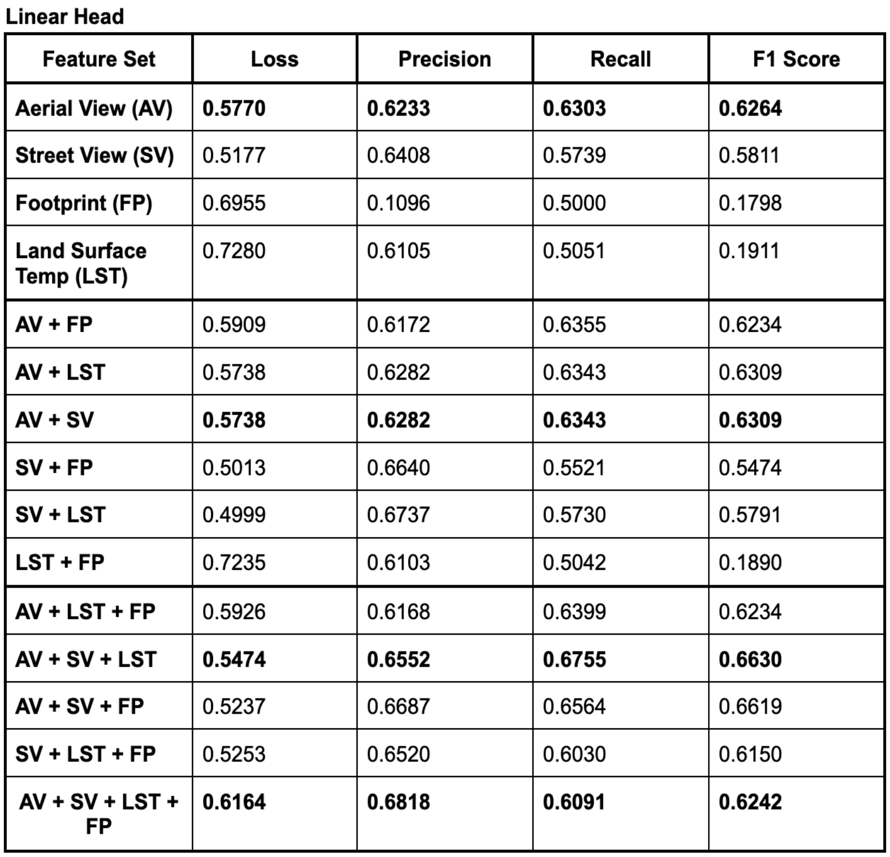
Notably, we performed the same cleaning process as the authors on the already-cleaned data, yet were still able to remove about 5,000 images from the training dataset that were unavailable, blurry, or obstructed by foliage.

We also performed ablation tests, with results shown below:



MLP Head

Feature Set	Loss	Precision	Recall	F1 Score
Aerial View (AV)	0.6152	0.6128	0.6321	0.6189
Street View (SV)	**0.5475**	**0.6398**	**0.6232**	**0.6299**
Footprint (FP)	21.9158	0.3904	0.5000	0.4385
Land Surface Temp (LST)	0.7338	0.6104	0.5046	0.1900
AV + FP	0.6231	0.5978	0.6341	0.5949
AV + LST	0.6368	0.6129	0.6231	0.6170
AV + SV	**0.5290**	**0.6645**	**0.6676**	**0.6660**
SV + FP	0.5743	0.6536	0.6204	0.6313
SV + LST	0.6281	0.6698	0.5789	0.5873
LST + FP	3.1889	0.3904	0.5000	0.4385
AV + LST + FP	0.6204	0.6047	0.6438	0.6024
AV + SV + LST	0.6001	0.6766	0.6071	0.6216
AV + SV + FP	**0.5854**	**0.6589**	**0.6801**	**0.6670**
SV + LST + FP	0.6286	0.6763	0.5868	0.5977
AV + SV + LST + FP	**0.5635**	**0.6704**	**0.6684**	**0.6694**



Linear Head

Feature Set	Loss	Precision	Recall	F1 Score
Aerial View (AV)	**0.5770**	**0.6233**	**0.6303**	**0.6264**
Street View (SV)	0.5177	0.6408	0.5739	0.5811
Footprint (FP)	0.6955	0.1096	0.5000	0.1798
Land Surface Temp (LST)	0.7280	0.6105	0.5051	0.1911
AV + FP	0.5909	0.6172	0.6355	0.6234
AV + LST	0.5738	0.6282	0.6343	0.6309
AV + SV	**0.5738**	**0.6282**	**0.6343**	**0.6309**
SV + FP	0.5013	0.6640	0.5521	0.5474
SV + LST	0.4999	0.6737	0.5730	0.5791
LST + FP	0.7235	0.6103	0.5042	0.1890
AV + LST + FP	0.5926	0.6168	0.6399	0.6234
AV + SV + LST	**0.5474**	**0.6552**	**0.6755**	**0.6630**
AV + SV + FP	0.5237	0.6687	0.6564	0.6619
SV + LST + FP	0.5253	0.6520	0.6030	0.6150
AV + SV + LST + FP	**0.6164**	**0.6818**	**0.6091**	**0.6242**

These ablation tests showed that the greatest deciding factors were the streetview and aerial view photos, which performed decently on their own. It was interesting to see the F1 score actually drop in the linear head after adding the footprint data (AV + SV + LST + FP) compared to the one with just the photos and the LST (AV + SV + LST), but this might just be due to fluctuations in the training process.

It's also interesting to note that in the paper, the authors had access to energy consumption data, which significantly improved the effectiveness of their model. Looking ahead, we would try to gather more data, such as energy consumption and building material data. Or, we could implement grid search to perform our own hyperparameter tuning, since we ended up removing even more images from the dataset using K-means clustering, making the ideal model parameters different from what the authors gave us. Finally, we could utilize a transformer to detect where in the image the model is focusing on, as well as utilize perturbations to test our model.

Challenges

The most difficult part was processing the data—we experienced significant slowdowns we weren't expecting due to the amount of time it took to first gather all the data, clean it, and rewrite and compile it in the form of Numpy arrays (> 30 hours total). After completing data retrieval, though, the model was a straightforward standard multilayer perceptron.

Discussion

This problem space lies mainly at the intersection of the climate and housing crises, mainly on the topics of decarbonization, sustainable development, and retrofitting. In Rhode Island alone, residential energy consumption (e.g., building heating & cooling) makes up more than a third of the state's total energy consumption. However, it would be insensible to just try and replace every building with one we know to be energy efficient, both from an environmental and housing crisis perspective—it would be a massive waste of both existing resources and existing housing. Instead, we should find ways to reduce this energy usage while still retaining old buildings, for example by replacing energy-inefficient systems with efficient ones, or retrofitting buildings to have better insulation and more energy-efficient design. This leads us to our problem—how do we identify these buildings? The process of acquiring the data that will determine which buildings to prioritize is both costly and lengthy. Licensed professionals must go on-site and conduct in-depth analyses to determine each building's energy efficiency, a procedure not many building owners would willingly go through without some sort of incentive. This is the issue this paper attempts to solve, and what we are hoping to replicate.

The major "stakeholders" in this problem would be 1) organizations or 2) homeowners that could potentially use such a model to identify buildings suitable for retrofits or renovations.

The organizations usually are ones promoting sustainable development or energy reduction programs, and while in theory could be both public and private, in practice are usually either nonprofits or branches of government with limited resources that are trying to incentivize homeowners to retrofit their properties. Mistakes made by our algorithm could result in inaccurate predictions, wasting the few (and diminishing) resources they have by telling them to target homes that are energy efficient and missing the ones that aren't.

On the homeowner side, as the beneficiaries of these types of programs, their home usually has to qualify for financial assistance or subsidies. If they initially believe their home does qualify after identifying it as non-energy efficient using this algorithm, but it turns out that after an audit, it is energy efficient, they will have wasted their time and money as well.

Reflection

Ultimately, we were able to replicate the original paper's models and get within our target of being within ~5% of their F1 scores. After we got through the preprocessing part of the data, actually working with the data and training the models was a simple process. We were able to achieve both MLP and Linear models that performed relatively similar to theirs.

Most of the challenges we faced were during the preprocessing portion. Looking back, we should have split our training data into batches, which we did end up doing, but only after hours of downloading images and running into collab time-out errors. We should have also saved our embeddings as numpy arrays, instead of zipped .npz files, which we later had to unzip when we wanted to concatenate them. However, after this, K-means clustering and concatenating the data were pretty smooth.

If we had more time, we would have done our own hyperparameter tuning using grid search. Currently, we are using the hyperparameters the authors said performed the best, but since we removed extra images during our K-means clustering, different hyperparameters could have improved our model performance. We would also like to perform a more in-depth analysis of the model. Due to time constraints, we only did the ablation studies and calculated the metrics, but it would be quite interesting and informative to use transformer attention to see exactly what features of the images the model is focusing on. Since energy efficiency is greatly affected by building facade material and the number/size of windows, we would like to see if the model correctly focuses on those features.

Takeaways

Although we were informed that the most difficult part of this project would likely be the data cleaning and preprocessing, we were definitely not prepared for how long it would actually take. We made many mistakes while preprocessing: for example, forgetting to concatenate the image embeddings to the whole dataset, or saving all embeddings to individual numpy files when converting instead of just concatenating them into one file. This resulted in a significant slowdown of the process. Additionally, implementing the methods used in the paper (e.g., k-means clustering and the ablation studies) gave us a much deeper understanding of how these methods worked, allowing us to ground the concepts in actual work.