Using Computer Vision to further econometric analysis: A case study with predicting Urban Heat Island

Jake Silberg
jsilberg@stanford.edu

Abstract

While "explainability" is often discussed for the purpose of ensuring we can trust how a model makes a decision, understanding and quantifying how a model makes decisions serves potential additional purpose: aiding econometricians and other data scientists in identifying factors that affect real-world variables of interest. One such area of research is Urban Heat Island, the increase in temperature in urban areas, which many researchers are seeking to understand. In this report, I train a model to predict UHI primarily for the purpose of identifying the factors that affect the model's predictions, under the assumption that these factors likely also drive UHI in the real-world. I develop a relatively successful model (the variations in predictions explains nearly 50% of variance in ground-truth UHI), discuss how traditional notions of model architecture need to be adapted in dealing with a regression rather than classification problem, and qualitatively and quantitatively seek to identify what affect the model's UHI scores. While the qualitative results remain anecdotal, and the quantitative results are inconclusive, I hope the work can inspire others to see computer vision as a tool for analyzing causal relationships out in the world.

1. Introduction

Urban Heat Island (UHI), the increased heat experienced by urban areas vs. nearby rural areas, is believed to have contributed to many of the 10,000 heat-related deaths in the US between 2004 and 2018 [1]. And the effects are unequally distributed – People of Color live in census tracts with more UHI effects than non-Hispanic whites [5]. While it is known that more developed areas and areas with industrial uses have higher UHI [18], computer vision can help us better understand what contributes to UHI by developing a model trained to predict UHI and understanding what the model uses to make its determinations. For example, if the model trained to predict UHI clearly pays attention to the number of trees, or the amount of paved area in an image, then we can infer that trees or pavement may affect UHI in the real-world. This project seeks, then, both to specifically understand what factors of an image contribute to a low/high predicted UHI, as well as propose and document how econometricians can utilize computer vision methods for quantitative analysis beyond prediction tasks.

Specifically, I use a CNN [10] trained on 5,000 satellite images from a Yale UHI dataset [6] to predict the UHI in the area the image depicts, and discuss factors that improve model performance. I then use qualitative methods such as error analysis, saliency maps, and perturbation techniques to try to understand how the model is predicting UHI. Finally, I use quantitative methods to attempt to understand how the model scores UHI and how applying those methods to predicted UHI compares to applying them to actual UHI scores.

2. Related work and Methods

2.1. Urban Heat Index

For comparing my results, I have examined previous attempts to predict and understand UHI. For example, the paper whose data I will be using [6], uses Land Surface Temperature readings to model UHI across the country, resulting in 55,000 observations. However, the model algorithmically/deterministically processed temperature readings, whereas I plan to "predict" UHI without a temperature reading by using visual bands, which I have not seen in the literature. While several papers have attempted to predict human settlement extent and land use and note the connection to UHI research, e.g., [12], I have not seen predicting UHI directly. The Yale paper also established that vegetation is correlated with reducing UHI, as the researchers concluded that a higher vegetation index (NDVI) is associated with lower UHI readings. Another existing paper identifying the factors that contribute to UHI is [2], which finds that pavement is associated with increased UHI. As a result, I should find that images with lots of vegetation (e.g., trees) should result in lower UHI predictions, while images with lots of pavement (e.g., highways) should be associated with higher predicted UHI.

2.2. Computer vision architectures

For architectures for my model to predict UHI, I considered a VGGNet [17] and a ResNet [8]. The ResNet architecture uses skip connections to enable training deeper networks, meaning that deeper layers directly receive the input of much earlier layers in addition to the activations of the immediately preceding layer. The idea behind a ResNet is that, by receiving the inputs to earlier layers as well as the activations of recent layers, the model can easily learn the identity transformation, so a deeper model should never degrade performance. As a result, ResNets are typically considered the highest performing Convolutional Neural Nets for image classification problems.

However, it is important to note here that this is a regression problem (we are predicting a continuous variable, UHI) rather than a classification problem. ResNets, to be efficient, use a global pooling layer after the last convolutional layers to create a 1 dimensional vector, then apply a single linear transformation to produce the output scores. This architecture is easy to adapt to regression: We just only predict a "score" for one class, then replace the softmax loss used in a traditional ResNet:

$\mathcal{L} = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$

with an MSE loss:

$\mathcal{L} = \frac{1}{D} \sum_{i=1}^{D} (\hat{y}_i - y_i)^2$

This will train the model to minimize the squared difference between its prediction and the true UHI.

However, in my previous research, where I have used the 1-dimensional vectors from CNNs for clustering images to identify similar images, I have noticed that VGGNets tend to outperform ResNets in terms of producing homogenous clusters of images. Note that VGGNets flatten the result of their final convolutional layers (converting the 3D tensor to a 1D tensor), then have several layers of non-linear transformations (i.e., linear transformations followed by ReLU layers), before a final softmax layer. As a result, the final 1D vector before the softmax layer, in my view, can be more expressive than (and is simply larger, being 1x4096) the final 1D vector before the softmax layer of a ResNet (1x512) because of these non-linearities. I believe this is what enables better vector-based clustering with VGGNet.

As a result, I will also experiment with using a VGGNet, and a modified ResNet with additional non-linear transformations after the global pooling layer. I have a hypothesis that my regression problem may follow a similar pattern as a clustering problem, where additional fully-connected layers may outperform ResNet's single linear layer, if the UHI prediction is a non-linear function of the final convolutional activations.

Note that, since I have not been able to find this previously done with visible band imagery, it is difficult to know the state-of-the-art. Still, other regression tasks based on remote sensing, such as predicting poverty, have been able to achieve a maximum r^2 of up to around .62. [3]. While this is not a truly apples-to-apples comparison with different data and variables, it provides a ballpark ceiling of what is possible.

2.3. Understanding drivers of model decision-making

Finally, much of my work is trying to understand how the model makes decisions, based on the assumption that factors driving model predictions are also factors that drive real-life variation in UHI. The first method I will use to understand how the model assigns high/low UHI scores is saliency maps, which find the portion of the image with the largest gradients. These are the portions of the image that most affect the model's predictions for a particular image. A more complex method is the Local Interpretable Model-Agnostic Explanation (LIME) developed by [13]. LIME works by perturbing different areas of the input image, then looking at which perturbations most affect the model's predictions. As a result, these are the portions of the image where the model is "paying attention" as changes to these portions most change the output.

In addition to these qualitative measures (looking at LIME explanations and saliency maps and subjectively searching for patterns), I also seek to apply quantitative measures to understand the model's predictions. Theoretically, if the model is using known predictors of UHI to make its conclusions, then the relationship between those predictors should hold steady between the model's predictions and the ground-truth UHI readings. For example, since we already know pavement is associated with higher UHI, more pavement in an image should be correlated with higher ground truth UHI readings. In addition, if the model is using the amount of pavement to predict UHI, then pavement should be positively correlated with predicted UHI as well. I have not been able to find related papers that attempt this with image data, however given it is a relatively vague concept (making it hard to search for), I certainly do not feel comfortable claiming that I am the first to attempt it. In the hope of making it easier to find in future literature, I will call the general concept Preservation of Known Predictive Relations.

I was inspired to consider it based on the discussion of a machine learning model used to predict patient outcomes in hospitals, in [7]. The model found that patients with asthma were predicted to have better outcomes in Intensive Care, which is non-sensical. The contradiction was driven by the fact that, in real-world hospitals, doctors are more likely to pay more attention to patients with more pre-existing conditions (asthma) and that additional care resulted in better outcomes. But the model should have preserved the previously known predictive relationship (that asthma predicts

worse outcomes), and this obvious failure caused doctors to lose trust in the model. Returning to the UHI case, if my model is working as intended, it should preserve known predictive relationships (e.g., more pavement means higher predicted UHI, more trees means lower predicted UHI).

3. Dataset

As described above, the Yale dataset has 55,000 UHI readings at the Census Tract level from [6]. While Census Tracts in rural areas can be quite large, in urban areas they tend to be quite small, meaning that the local built environment in the images should be the primary driver of UHI. I merged the Yale data with Census population density readings [4], and filtered to include the 5,000 most population dense readings. I did this as I am seeking to understand the drivers of UHI, and the full dataset contained thousands of more rural areas and farms that would have negative UHI readings. In addition, focusing on more urbanized Census Tracts means the readings are more likely to be locally contained rather than an average over a huge area. I found the central latitude and longitude of every Census Tract and downloaded an image of the area (each image covers roughly 1.5 km). For testing how much pavement impacted the model, I used Google Maps to download the same area, but in a "maps view". Thus, I could easily calculate how much of the image was taken up by a highway based on how much of the image was yellow (the color of highways in Google Maps). Please note that I had already written this code prior to 231n, but had not used the percentage of highway in an image to try to predict UHI. Unfortunately, there is no analogous technique to find the percentage of an image taken up by trees, as trees are not marked in Google Maps (and "green" in a maps image is usually grass parks rather than more substantial vegetation).

4. Experiments and Discussion

4.1. Optimizing model performance

As a baseline, I first tested a fully-connected model based on extracting the colors in the images using the hue channel. My hypothesis was a concentration of grays (indicative of pavement) and greens (indicative of vegetation) would be fairly strong predictors of UHI. I used the code from CS231N assignment 1 to extract the color histograms. The model was surprisingly predictive.

For ConvNets, I tested VGGNets, ResNets, and modified ResNets on my dataset, with the metric of success being Pearson R^2, the proportion of UHI variance associated with the variance of the model's predictions, using [9]. A higher R^2 means the predicted scores are more closely correlated with the ground-truth. Note that UHI is measured in the difference in degrees between an urban area and a corresponding rural reference point. For example, a UHI reading of 5 means the location is 5 degrees warmer than a location outside the city. For preprocessing, I found that normalizing UHI scores (subtracting the mean and dividing by the standard deviation) improved model performance. Given that I started with a pre-trained ResNet (trained on ImageNet), I also used the typical image preprocessing suggested by PyTorch, of normalizing the image channels based on ImageNet's mean and standard deviation. All models were built and trained in PyTorch [11]. Finally, I divided the data into a training set of 4000 images and a validation set of 1000. The images are 1024 by 1024, but scaled to 256x256 for the pre-trained ResNet based on ImageNet [14].

I performed a hyperparameter grid search on the two most important hyperparameters, learning rate and weight decay (L2 regularization). 1e-4 learning rate performed best, as did 1e-5 for weight decay, even though the training loss dropped surprisingly quickly. I used a ResNet with 50 layers and a VGGNet with 16 layers. I used a batch size of 32, as it was the largest possible in cuda memory, and an Adam optimizer as suggested by PyTorch.

As I predicted, the top performing model was a modified ResNet with additional fully-connected layers.

Model	**R^2**
Color histograms	0.44059
VGGNet	0.46703
ResNet (baseline)	0.45422
Modified ResNet (512x1000 linear, ReLU, dropout, 1000x1 linear) with MSE loss	0.47013
Modified ResNet (512x1000 linear, ReLU, dropout, 1000x1 linear) with L1 loss	**0.48472**
Modified ResNet (512x1000 linear, ReLU, dropout, 1000x256, ReLU, dropout, 256x1linear) with MSE loss	0.45967

Figure 1. Model quantitative results

I believe the modified ResNet outperformed a traditional ResNet for two reasons. First, I included a dropout layer between the ResNet's traditional linear layer and the linear layer I added. This significantly reduced overfitting. The traditional ResNet had a training MSE loss of 0.0451 but a validation loss of 0.5869. Meanwhile, my modified ResNet with the additional dropout layer (with dropout proportion of .8, so 80% of neurons set to 0) had a higher training loss of 0.0964, but a lower validation loss of 0.5583. This shows that the traditional ResNet, where I could only regularize losing weigh decay (L2 regularization) was overfitting more than the modified ResNet. The second reason, I believe, is that the additional ReLU non-linearity between the first and second fully-connected layers in my modified ResNet allowed the model to be more expressive, similar to

how a VGGNet's fully-connected layers lead to more expressive clustering results. While this non-linearity may be unnecessary for classification, it appears to help in regression, which I have not found anywhere in the literature.

However, note that the second type of modified ResNet, with two extra fully-connected layers (with dropout in both), underperformed the ResNet with a single extra fully-connected layer. This may have been because I trained both for 30 epochs (with early stopping) and the deeper network needed longer to train (as deeper networks usually do). Note that the deeper network achieved its best validation loss (0.5811) at a higher training loss (0.1753), consistent with potentially performing better if trained for longer. It is also possible that with multiple dropout and ReLU layers, the deeper network suffers from vanishing gradients that make it more difficult to train, and thus underperform.

Most interestingly, I found the model had an unusual quirk. The predictions, particularly the predictions on images with higher ground-truth UHI, were "shrunken" towards 0, as seen below. Note that this only occurred on the validation set, whereas the training set had a matching distribution.

I have not been able to identify a conclusive reason why this is the case, but first note it holds true across different random splits of training and validation sets. I initially hypothesized this was a result of a lack of expressivity in the ResNet (as described above). However, using the modified ResNet with additional non-linearities did not fully resolve the issue. I then hypothesized that perhaps the issue was caused by the loss function combined with normalization. For example, if most of the data is clustered around 0 (due to normalization), then perhaps the model can reduce squared error by guessing ever closer to 0, as this is likely, regardless of the ground-truth value, to be a smaller magnitude error than guessing away from 0. To attempt to rectify this, I switched the model to training with L1 loss (below) instead of MSE loss:

$\mathcal{L} = \frac{1}{D}\sum_{i=1}^{D} |(\hat{y}_i - y_i)|$

I hypothesized that by penalizing the absolute value of the size of the loss, rather than the square of the size, this might make the model less inclined to predict towards 0. Note that, because the error tends to be less than 1, switching from squared loss to L1 loss increases the relative penalty for small errors, and decreases the relative penalty for large errors. I hoped this consistency might encourage the model to guess further from 0 as often as it does closer to 0. Switching to L1 loss further improved performance (in terms of R^2) and visually seems to have slightly reduced the shrinking, but clearly not fully fixed it. I hope future work can further identify whether some other model architecture can ameliorate the issue, or if it is some fundamental issue with this dataset or problem. One future option would be to "upweight" images with UHIs further from 0, similar to a class imbalance problem.

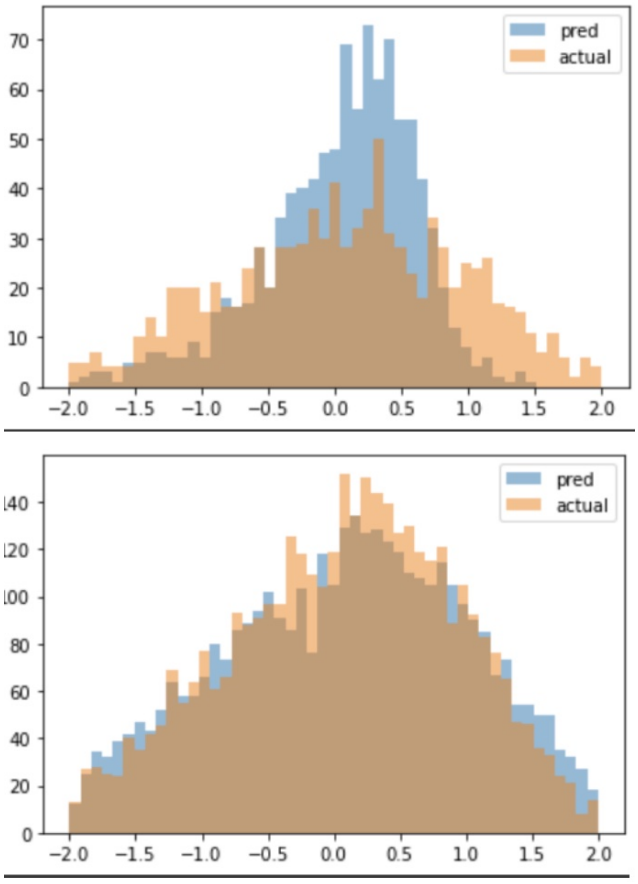

Figure 2. Traditional ResNet50 model prediction distribution on validation (top) and training (bottom) sets

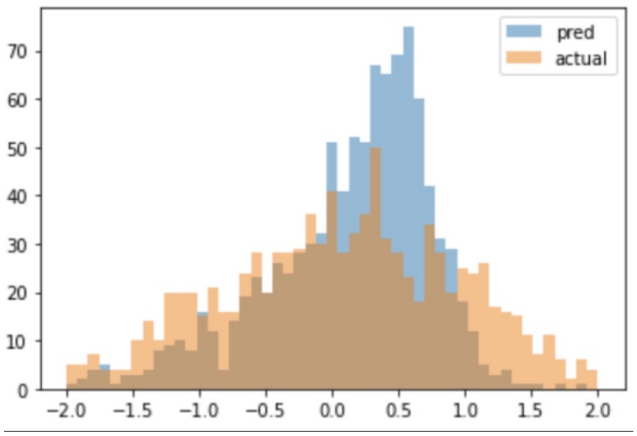

Figure 3. The final model is slightly more willing to make larger positive predictions on the validation set, though the shrinking towards zero is clear still

4.2. Analyzing drivers of model decision-making

After optimizing the model, I wanted to understand what visual factors contribute to the UHI prediction. I believe that methods to make models more "explainable" can double as advancing quantitative econometrics, because if something is affecting the model, that is a strong indication it affects ground-truth variables.

I began by examining the filters the final model used in the first convolutional layer. I am not sure there is much to be gleaned here, though it appears some circular filters may be useful in identifying buildings or structures?

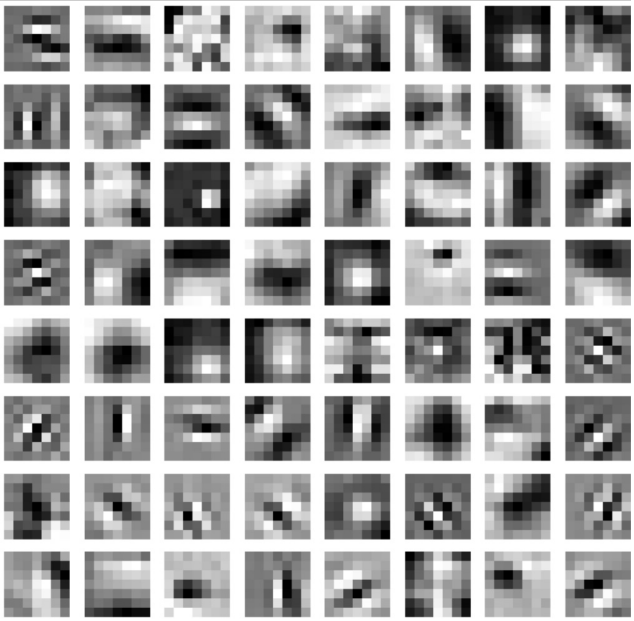

Figure 4. First layer filters

Next, I examined the dataset qualitatively, looking for patterns. Note that all examples shown in this section come from the validation set, so the model did not see them in training.

I looked at the images with the lowest ground-truth UHI readings:

Figure 5. Lowest ground-truth UHI

One can clearly see that bodies of water are, unsurprisingly, associated with the lowest UHI. This is probably driven by both the literal ability of water to absorb heat, and the wind conditions from oceans and large lakes that dissipate heat. There is less of a pattern when visualizing the images with the highest ground-truth UHI:

Figure 6. Highest ground-truth UHI

Clearly all the images are densely populated (with many structures and buildings), but I was surprised they seem to be of residential areas rather than, say, large parking lots.

Then, I visualized the images with the lowest predicted UHI in my model:

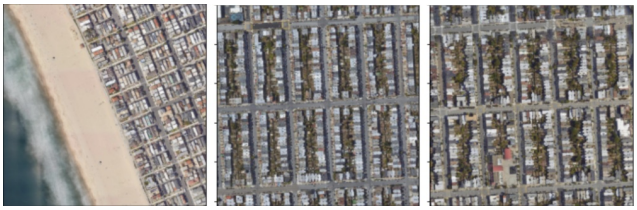

Figure 7. Lowest predicted UHI

It would seem the model is picking up that the image on the left has water, driving the low rating. While I do not know what drives the low rating for the other two (lots of trees in the backyards?), it is somewhat reassuring the two very similar images get similar UHI scores.

Figure 8. Highest predicted UHI

For the highest predicted scores, it seems the model is deciding based on highly populated areas

I also visualized the images with the largest errors between the model and ground-truth UHI. Note that I used absolute error, as relative error spiked with ground-truth readings close to 0 (small denominator). The largest model underpredictions make sense: the water on the left likely resulted in a low prediction, while the extensive greenery may have make the model underpredict.

Figure 9. Model under-predictions

The model's overpredictions may have been driven by the fact that these areas appear densely populated, though it is surprising the model predicted so high given the greenery.

Figure 10. Model over-predictions

To further qualitatively understand how the model makes predictions, I created saliency maps. Note that, unlike in classification on ImageNet or CIFAR, the model clearly pays attention to many different parts of the image (rather than just clearly looking at, say, the elephant in the middle to conclude it is an elephant). These examples were chosen because there were fewer "bright" spots on the map, so it more clearly shows where the model was focused. The model, to my eye, appears to be paying attention to trees in the bottom-right of the second image, and one can make out the sides of large roads lighting up in the saliency maps as well in the first image.

I also lightly adapted the LIME demo for use with regression (by simply using the regression score as the "top predicted class" and removing the softmax layer) to identify where the model was looking. As previously stated, LIME perturbs the image in different areas to identify which perturbations most affect the resulting score. One advantage of using LIME is that it provides the direction by which a patch is altering the prediction. Note that green masks indicate pushing the prediction up, and red masks indicate pushing it down. This is in contrast to saliency maps, where we take the absolute value of the gradient, so the directionality is lost. However, here is where the qualitative work starts to get difficult. The LIME mask above seems to be focusing on roads (the highway in the bottom right and the parking lot push the UHI prediction up), while here the saliency map has a dark spot where the road runs through, indicating it is not paying attention to the road. Both seem to agree however that the dense grouping of houses in the bottom-left of the image push the prediction up.

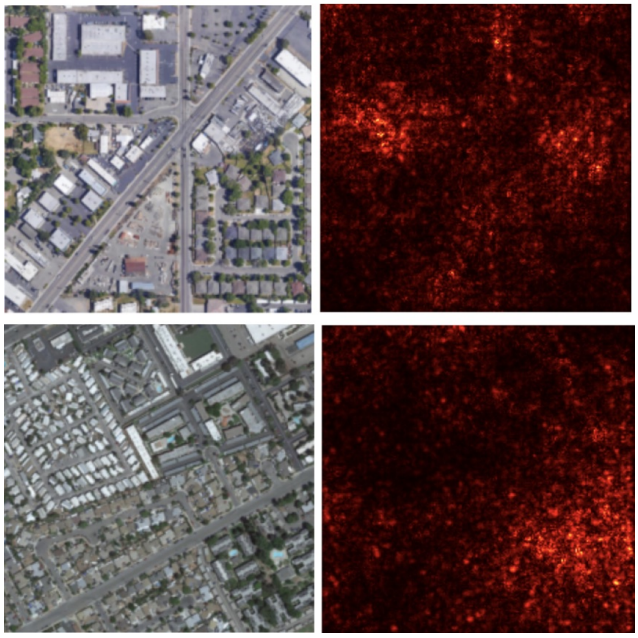

Figure 11. Saliency maps

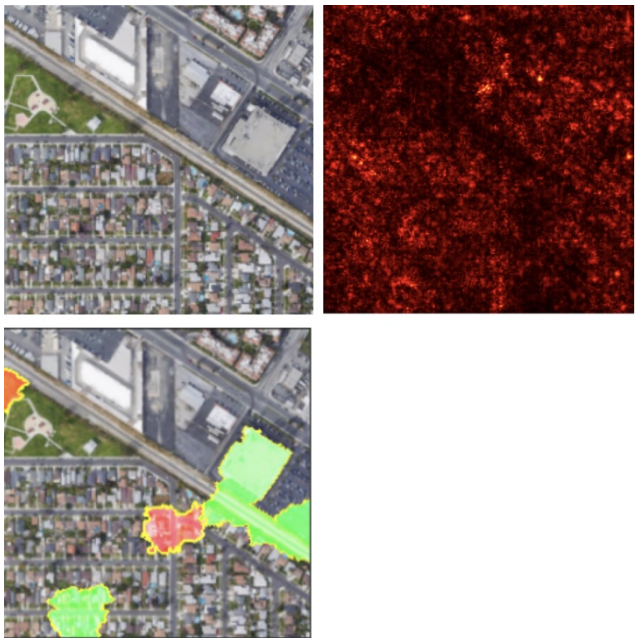

Figure 12. Saliency plus LIME

One creative idea to get more granular is to find "nearest neighbor" images with large discrepancies in predicted UHI scores. This means the model (particularly the final linear layers) must see something different between two images, even though the convolution filters mostly agree. Note here that this is another advantage of using the modified ResNet architecture, as the vectors tend to be more expressive representations than in a traditional ResNet (based on my experience with VGGNets as mentioned earlier). To do this, I took the L1 UHI model and removed the ReLU, dropout, and final linear layers. As a result, each input image generates a 1000 dimensional vector after the convolutional layers and a single linear transformation. For every image, I found its nearest neighbor in that transformed 1000 dimensional space, using Euclidean distance. I then found the pairs of "neighbors" with the largest discrepancy in scores.

Figure 13. First example of nearest neighbors with differing UHI scores

For once, this method makes it very easy to confirm my hypothesis. The two images are very similar (suburban neighborhoods, many red roofs), but the image on the right – which contains a highway – has a higher predicted UHI. LIME then shows that the model is looking at the highway to push up the UHI score! Similarly, in the second example, the image on the left has a higher predicted UHI, and the model seems to be looking at the large patch of pavement where there is a major intersection. These findings, though not all are so clear cut, provide anecdotal evidence that, as expected, the model uses pavement in predicting UHI scores.

Unfortunately, I have mostly negative findings for quantitatively analyzing the correlation between highways and predicted UHI. As I noted, the correlation should be positive, as pavement is considered in the literature one of the largest predictors of UHI. So I regressed predicted UHI on the percentage of image taken up by highway pavement, using [15]. The coefficient was .02, meaning every percent increase in highway was associated with a .02 standard deviation increase in predicted UHI. While directionally correct, the coefficient was not significantly different from 0 (with a p-value of .3). More unusually, the coefficient for highway pavement compared to ground-truth UHI was even smaller, 0.005, which was not significantly different from 0 (p-value of .99!) meaning that the amount of highway pavement did not affect the actual UHI of an area for the validation set.

Figure 14. Second example of nearest neighbors with differing UHI scores

While the two coefficients "agree" in a statistical sense (both are indistinguishable from 0), I was surprised the coefficient of pavement for predicted UHI was much larger. This would indicate the model is very slightly taking into account the amount of highway to increase UHI predictions, however this may even be erroneous, given the true coefficient in the dataset was even closer to 0.

Variable	Coefficient	p-value
Ground-truth UHI	0.0005	0.986
Predicted UHI	0.0210	0.304

Figure 15. Linear regression results of pavement against true and predicted UHI

One possible explanation is that, because I could only compare against highways rather than all pavement, highways just play too small a role, as the vast majority of images did not contain highways, as shown below. Perhaps the correlation for total percentage of pavement would have been more consistent with my hypotheses. Additionally, because trees are not picked up in the "maps" view on Google Maps, I could not quantitatively assess the coefficient on tree cover in the model. Thus, while the idea of looking for preserved predictive relationships remains interesting to me, I have little from this particular analysis to show for it.

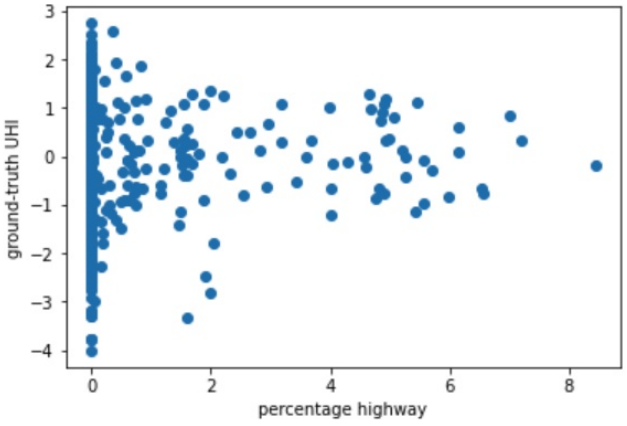

Figure 16. Most images do not contain any highways

5. Conclusion

There are a few key learnings from this project. The first is that regression is a very different problem than classification for computer vision! As a result, a traditional ResNet, which is usually the highest performing classification model, underperforms compared to a model with additional fully connected layers to enable further expressivity in predicting a continuous variable.

The second is that understanding models comprehensively, particularly for regression and with satellite imagery, remains a daunting challenge. While the qualitative analyses resulted in interesting anecdotes – particularly when LIME showed the model was looking at a highway to distinguish otherwise very similar images – most of the qualitative results had counter-examples. Sometimes it seemed the model was looking at pavement and greenery, other times it didn't. Sometimes it seemed residential areas resulted in high predictions, sometimes in low predictions with few ways to understand what distinguished them.

I think these findings point to the importance of more quantitative ways of understanding what a model is doing under the hood. As I have stated, I am interested in this not because I think explaninability in the abstract is all that critical, but because I believe it serves a particular purpose in advancing computer vision as a new avenue by which researchers can identify and quantify how factors of interest affect target variables. Unfortunately, the one comprehensive quantitative analysis I was able to attempt appears to be unimportant to this particular model. However, I hope this idea, of looking for quantitative correlations with known predictors across the whole validation set rather than cherry-picking qualitative examples, can continue to develop in the future.

The clearest next step would be to see what other factors in these images might be strongly correlated with UHI, for example using all streets rather than just highways, or trying to identify and count trees in the images. And, in further research I am conducting separately, I will be using a generative model to remove highways from these images and test the change in UHI predictions. This would most conclusively show how the pavement is affecting the model.

I hope this report gets other researchers thinking about how, at scale, analyzing what causes a model to change scores can teach us not just about the model, but quantitatively teach us about the world as well.

6. Contributions and Acknowledgements

I would like to thank Stanford students Kelly He and Kristy Choi, and Professors Stefano Ermon and Marshall Burke, for the discussions on model design and data analysis that enabled this report. While they did not provide code, they did help me think through how to modify a ResNet for regression and what metrics to use. Additionally, the SustainLab and SAIL provided cluster compute and Google Cloud GPUs that, while not used for the analyses above, were used to download the images used. I also wish to thank Jeremy Irvin and Hao Sheng in SAIL who wrote the code that enabled me to download Google Maps imagery (the repo is non-public but was developed for [16]).

The public repos I used were PyTorch [11], scikit [9], statsmodels [15] and LIME [13]. In particular, I used the LIME demo at https://github.com/marcotcr/lime/blob/master/ doc/notebooks/Tutorial%20-%20images%20 - %20Pytorch.ipynb. For visualizing the filters, I adapted code from https://debuggercafe.com/visualizing- filters-and-feature-maps-in-convolutional-neural-networks-using-pytorch/.

From class, I used the saliency map code and the color histogram extraction code from http://cs231n.stanford.edu/schedule.html.

References

[1] Heat island effect, epa, https://www.epa.gov/heatislands/heat-island-impacts. 1

[2] Hashem Akbari, Surabi Menon, and Arthur Rosenfeld. Global cooling: effect of urban albedo on global tem-

perature. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2007. 1

[3] Kumar Ayush, Burak Uzkent, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Efficient poverty mapping from high resolution remote sensing images. In *Proc. AAAI Conf. Artif. Intell*, volume 35, pages 12–20, 2021. 2

[4] US Census Bureau. Tiger/line shapefiles, Dec 2021. 3

[5] T Chakraborty, Angel Hsu, Diego Manya, and Glenn Sheriff. Disproportionately higher exposure to urban heat in lower-income neighborhoods: a multi-city perspective. *Environmental Research Letters*, 14(10):105003, 2019. 1

[6] T Chakraborty, A Hsu, D Manya, and G Sheriff. A spatially explicit surface urban heat island database for the united states: Characterization, uncertainties, and possible applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 168:74–88, 2020. 1, 3

[7] Brian Christian. *The alignment problem: Machine learning and human values*. WW Norton & Company, 2020. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[9] Oliver Kramer. Scikit-learn. In *Machine learning for evolution strategies*, pages 45–53. Springer, 2016. 3, 8

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 3, 8

[12] Chunping Qiu, Lukas Liebel, Lloyd H. Hughes, Michael Schmitt, Marco Körner, and Xiao Xiang Zhu. Multi-task learning for human settlement extent regression and local climate zone classification. *CoRR*, abs/2011.11452, 2020. 1

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 2, 8

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 3

[15] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, volume 57, page 61. Austin, TX, 2010. 7, 8

[16] Hao Sheng, Jeremy Irvin, Sasankh Munukutla, Shawn Zhang, Christopher Cross, Kyle Story, Rose Rustowicz, Cooper Elsworth, Zutao Yang, Mark Omara, et al. Ognet: Towards a global oil and gas infrastructure database using deep learning on remotely sensed imagery. *arXiv preprint arXiv:2011.07227*, 2020. 8

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[18] Decheng Zhou, Jingfeng Xiao, Stefania Bonafoni, Christian Berger, Kaveh Deilami, Yuyu Zhou, Steve Frolking, Rui Yao, Zhi Qiao, and José A Sobrino. Satellite remote sensing of surface urban heat islands: Progress, challenges, and perspectives. *Remote Sensing*, 11(1):48, 2018. 1