Google Maps Imagery for Income Prediction

Antonio Ferris
Stanford University
antoniof@stanford.edu

Johannes Hui
Stanford University
jnhui@stanford.edu

Abstract

In this paper, we attempt to classify household income from images available via public APIs. We first categorize the household incomes provided by US census data into buckets via K-means, and then design a CNN to classify images into these buckets. Our training data was generated through the Google Street View and Google Maps APIs. We generated data from three different cities: Providence, RI, Atlanta, GA, and Washington DC.

This problem is incredibly difficult as there is often little to no obvious information in given images about the wealth of the area. Despite the difficulty of the problem, we were able to obtain impressive accuracies on our test dataset around 80% for all cities demonstrating that our CNN was able to pick up on minute details that differentiate high- and low-income areas. However, our classifier's good performance was localized to the city that it was trained on. We show that a CNN can be made to accurately distinguish income inequality in a city, but future work is needed to design classifiers that can work on unseen cities.

1. Introduction

Each year, the U.S. Census Bureau expends vast resources, contacting over 3.5 million households to collect, among other information, income data through the American Community Survey. Despite the time-consuming nature of this task, it likely remains the most accurate method of a variety of statistics, including those on demographic, economic, social, and housing characteristics across the U.S. population. While we understand the necessity of this endeavor, we also recognize the benefits of alternative, more efficient information-gathering techniques. These methods, by leveraging existing publicly accessible data, may also allow to be used with greater frequency and flexibility when surveys are not possible.

Our goal is to predict median household income level from Google Maps images. While we initially began by using Street View images, we ultimately developed models using satellite imagery, which we yielded more promising results. Our method is designed for urban areas, and we trained models on three cities: Atlanta, GA; Washington, D.C.; Providence RI. These three cities were identified in 2018 as the three U.S. cities with the greatest income inequality. [1] Additionally, preliminary results from our Street View models developed before switching to satellite imagery were based on datasets from Atlanta, GA and Phoenix, AZ, the home city of one of the authors.

Each of our final models was trained independently on a single city. The input to each model was a satellite image taken from Google Maps, which was then passed through a convolutional neural net (CNN) that utilized transfer learning. Rather than predicting the precise income of each neighborhood, our algorithm predicts 1 of 3 income classes, namely high, medium, and low income. We developed two sets of models. The first used income brackets found by performing k-means clustering on incomes across each city's census tracts with k = 3. The second set used standardized cutoffs that remained the same regardless of the city.

To further evaluate our model, we used confusion matrices to understand its performance, and to interpret it, we created saliency maps of input images to understand what image features our algorithm was looking for. We believe this type of examination is crucial if we are to develop a robust model that might contribute to alternative methods of gathering income data.

2. Related Work

2.1 Techniques with Street View Imagery

In develop our preliminary models, our method sought to rely on Google Street View images. Previous approaches to predicting income or other types of Census information had been approached by a variety of authors using such data. Glaeser [2] demonstrated that this method of sampling in a uniform grid produced a reasonable method of using street view images to predict 5-year median income in New York and Boston without the use of CNN's, and we hoped that a deep learning approach would provide even better results without the need for explicit feature extraction.

In more recent examples, Archaya, Fang, and Raghvvendra [3] and Gebru et al. [4] utilized a method of collecting street view imagery to extract 6 images, each with a 60-degree rotation. In this way, they can then associate a 360-degree view of 6 images with each location, a well-chosen method we adopted when collecting Street View data. Gebru et al. [4] was unique in adding post-processing steps, namely the detection of the type of vehicle in the image, before preforming the final prediction task. While effective, we sought to move directly from an image to prediction.

2.2 Techniques with Satellite Imagery

Many efforts to capitalize on the availability of satellite imagery have emerged in recent years as well. Heitmann and Buri [5], Pandey, Agarwal, and Krishnan [6], and Jean et al. [7] have all made efforts to predict poverty with satellite imagery data and CNNs. Much of this has focused on areas with less regular survey information, particularly in rural areas or the developing world. The prevalence of this research on regions without dense buildings seems to indicate that CNN-based models would be able to learn not only from more populated areas in a city, but also sparser areas such as a local park with lots of tree coverage.

At the same time, other authors have sought to develop models targeting urban areas, including those outside the United States. Vallebueno [8] applied CNNs to estimate income levels in Mexican cities. Najjar, Kaneko, and Miyanaga [9] took satellite images from Chicago to build models for predicting crime levels in the city. This last paper provided useful insights into effective approaches to gathering satellite imagery from U.S. cities and informed the way that we scaled our images. Looking at this body of work as a whole, it seems that the use of deep learning and CNNs both common and the state-of-the-art approach. Transfer learning also appeared to be quite prevalent.

2.3 Architecture

While transfer-learning using VGG16 has become a common practice, we found Xu et al. [10] to be a helpful starting point in terms of planning out the data pipeline and writing the code for it. This was a paper that also used VGG with a shallow top and most of the params frozen for a slightly different purpose. It served as a proof of concept for us as we developed our own additional layers to stack on top of VGG16.

3. Methods

3.1 Income Brackets: K-Means Clustering

In selecting our income brackets, we experimented with two different methods. For our first method, we used k-means clustering, with k=3, to determine our income brackets. We did this separately for each city and performed our clustering across the list of incomes by census tract. In other words, the income brackets for each city was unique to that city, and clustering was done using the distribution of incomes over the census tracts in the city's county.

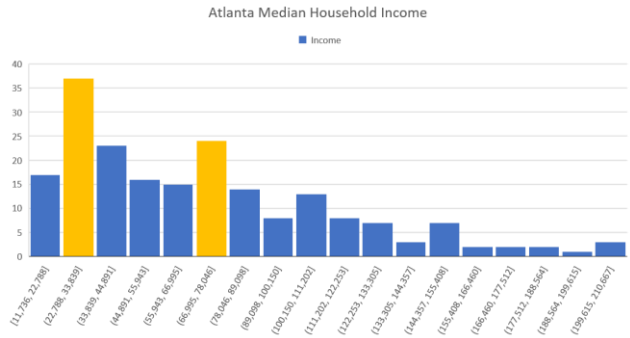

Figure 1: Histogram showing distribution of incomes over census tracts in Fulton County, GA, where Atlanta is located. Income bracket cutoffs are marked in yellow and are at $26,604 and $75,375.

In some cases, we realized that using this method resulted in brackets that seemed to be too narrow. This would be a problem for at least two reasons. First, it would be a useful categorization if one bracket captured very few households and another a vast majority of households. Second, it would make it even more difficult to train and evaluate our model as there would be a scarcity of examples for some income brackets. Given the difficulty of determining whether an income bracket captured real-world differences in wealth or income, we decided to address the second problem. Further visualizations of the distribution of incomes with our cutoffs revealed that what appeared to bad cutoffs in the distribution over census tracts sometimes were more appropriate cutoffs in the distribution over images in our dataset. Since we were training with the distribution over images, this addressed a major concern.

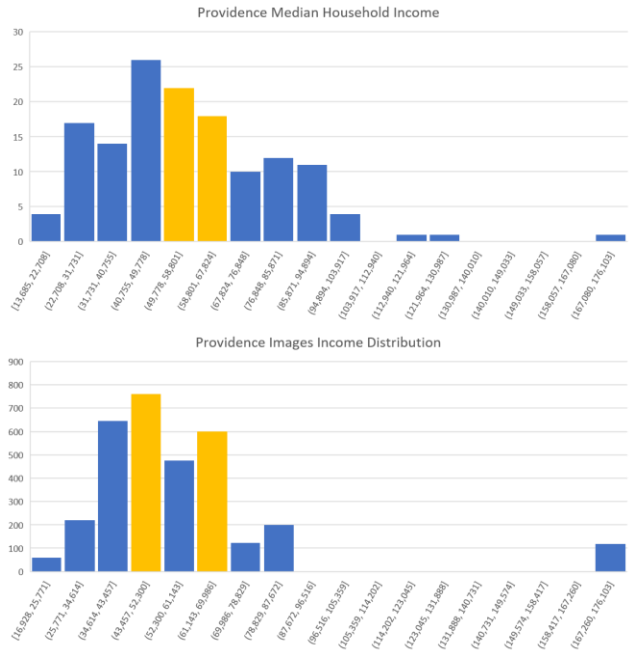

Figure 2: Histograms showing distribution of incomes in Providence County, RI, where Providence is located. The top graph shows the distribution over census tracts, and the bottom graph shows the distribution over images in our dataset. Income bracket cutoffs are marked in yellow and are at $50,117 and $65,129.

As can be seen in Figure 2, the cutoffs determined using k-means clustering appeared to be unhelpful in the distribution over Providence County census tracts but divided the distribution of incomes over images in the dataset quite nicely.

3.2 Income Brackets: Fixed

While k-means clustering provided a convenient way to create income brackets for a given city, it also presents certain problems. We have already seen how the brackets found using clustering may not adequately capture real-world differences in income. Additionally, because our clustering was performed on regional data, it would be difficult to generalize these brackets across cities. To counter these problems, we also trained models using a different set of income brackets, namely fixed brackets. We settled on two sets of fixed brackets. The first was taken directly from Archaya, Fang, and Raghvvendra [3] where they predicted income levels in Oakland using Street View Images. These cutoffs were at $75,000 and $150,000 and seemed to work better for some cities, such as Washington, D.C., and not as well for others, such as Atlanta.

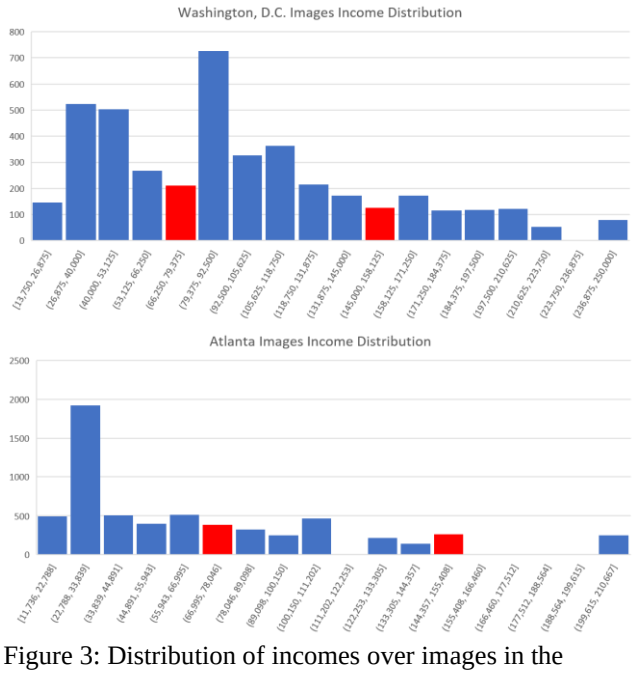

Figure 3: Distribution of incomes over images in the datasets for Washington, D.C. and Atlanta, GA. Cutoffs for both are in red and at $75,000 and $150,000.

The second set of cutoffs we selected were based on a qualitative evaluation of the distributions for the three cities we were building models for. These cutoffs were at $60,000 and $100,000 and seemed to correspond with real-world descriptive income brackets for most states in the United States [11].

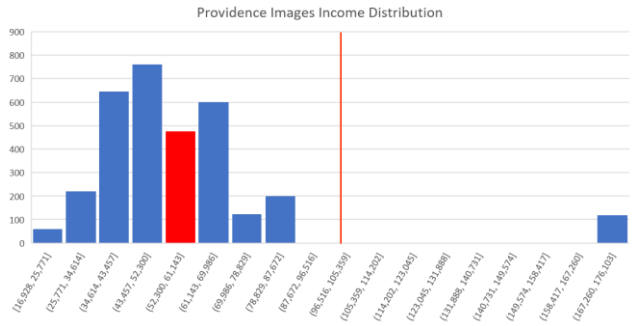

Figure 4: Distribution of incomes over images in the Providence, RI dataset. Cutoffs are marked in red and at $60,000 and $100,000.

As seen in Figure 4, even distributions that seem difficult to split up meaningfully, such as that for Providence, seem to do better with this second set of cutoffs at $60,000 and $100,000. They appear to provide real-world meaning and divide the data not too unreasonably. More even divisions might make training and evaluating our model easier, but comes at the cost of making the income brackets less interpretable.

3.3 Model Architecture

After we have finished processing the income data, our output is now one of the labels in the set {0, 1, 2}, corresponding to low, medium, and high income respectively. We now construct a model capable of taking an image of a residential area, extracting information relevant to the income of the neighborhood the picture was taken from, and then predicting an income bracket.

Since our problem involves extracting information from image features, it was natural to look to previously trained networks for this task. In this vein, our models were always prepended by a frozen copy of the bottom of the network VGG16 with pretrained weights on the Imagenet dataset. This network allowed us to do some initial processing of the image from its initial pixel values to the compressed VGG16 feature output. The VGG16 feature output is of shape 7 x 7 x 512.

Once we have the VGG16 feature output, the first layer we use in our final model is a global average pooling layer. This is a pooling layer (without trainable parameters) that takes the VGG16 output and flattens a H x W x C size tensor into a C size vector by averaging each H x W slice of the tensor. In this case, this layer further compresses the 7 x 7 x 512 output into a 512 vector.

During the course of training, we attempted to reshape this layer to preserve the information captured by the VGG features, but it resulted in a steep drop in initial performance that, even after much training, remained worse than with global average pooling. We suspect that this is because income is a very abstract measure to draw from images, and there is a lot of useless information present in each image. The more complex output from VGG without global average pooling might exacerbate this problem and allow our model to overfit.

Once we have the features as a vector, we can directly apply computation to extract the information pertaining to neighborhood income from the given features. We do this with a series of fully connected layers. Our final model had, following the pretrained VGG16 model with a global average pooling layer:

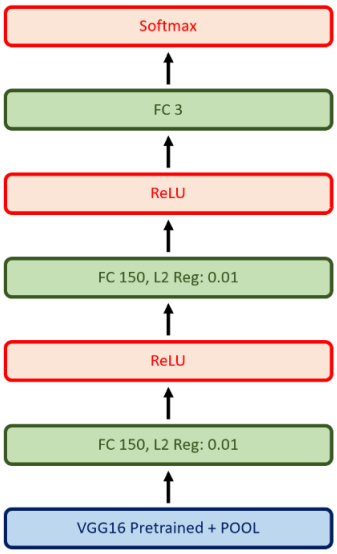

Figure 5: General architecture of our final model

This architecture was the result of a series of iterations and fine tuning based on results from our validation data. Initially we had larger layers, but these this gave no increase in performance and even sometimes decreased our performance on the validation data. We concluded that this was, again, a result of the varying amounts of useful information in each image. Smaller networks are as capable of dealing with small amounts of complexity as larger networks, but are incapable of overfitting to the extent we saw with larger networks. We found two layers of size 150 to be a sweet spot in network that was complex enough to fit our data well but not too complex to radically overfit.

We initially had good results with small networks with no regularization, but as we continued to incrementally increase the complexity of our network, it became clear that we needed to regularize. We first tried dropout, which dropped both our train and test performance significantly. After some iteration, we settled on L2 regularization on the dense layers with 0.01 regularization scale.

4. Dataset and Features

4.1 Income Data

For the income data, we are using the American Community Survey 5-year averages of median household income by census tract [12]. Each census tract is identified by a unique Census Geographic Identifier (GEOID). This is the most detailed level of public income data provided by the Census Bureau, with each tract generally including

between 2,500 and 8,000 people [13]. The Census Geocode API allowed us to map each latitude and longitude coordinate to its appropriate GEOID [14].

4.2 Street View Imagery

Our initial models used images collected through the Google Maps Street View Static API. We adopted a technique used by past researchers by collecting over a relatively uniform grid defined by longitude and latitude bounds. We stepped through the grid at various longitude and latitude step sizes between 0.005 and 0.01, noting that the grid is not perfectly uniform due to distortions resulting from the earth's curvature and the way in which longitude and latitude are measured. At each location, we took 6 street view images of size 224 x 224 at 60-degree rotations. Together, these formed a 360-degree view of the location. Our datasets here ranged from around 6000 images to around 18,000 images. We used data from Atlanta, GA, and Phoenix, AZ, the U.S. city with the most income inequality in 2018 and one of the author's home cities, respectively.

Figure 6: Six separate images from one location in our Atlanta, GA Street View dataset. Each rotated 60-degrees from the previous image to form a panoramic view.

4.3 Satellite Imagery

For our satellite imagery, we collected images using the Google Maps Static API. We used the same methodology as when collecting Street View images, moving across a relatively uniform grid to collect 224 x 224 sized images. Here, we used fixed latitude and longitude step sizes of 0.002 and a zoom level of 17, as defined by the Maps Static API. We chose these numbers to provide a good coverage of the target area, resulting in images that had often had a very slight horizontal overlap. Our data here was drawn from Atlanta, GA (6132 images), Washington, D.C. (4252 images), and Providence, RI (3213 images), the three cities with the highest income inequality in the U.S. as of 2018.

Figure 7: Three consecutive images from the satellite image dataset for Washington, D.C. Note the slight horizontal overlap, ensuring complete coverage for each latitude.

4.4 Train-Validation-Test Splits

To train our model, we split our data into train-validation-test sets with 70%, 20%, and 10% of our data respectively.

5. Results

5.1 Street View Image Models

We began with models trained on street view images that we ultimately abandoned. Using transfer learning with a VGG16 network and pretrained weights on the Imagenet dataset, we start with 50% accuracy within the first epoch of training. However, further training with the VGG16 layers frozen does not improve accuracy much, if at all. None of our models reached a validation accuracy above 60%, and they did not seem to be improving given more training time. Below is a graph of our initial model's (VGG16 - GlobalAverage2DPooling – Dense + ReLU – Dense + Softmax) accuracy on the training set (train) and the validation set (test).

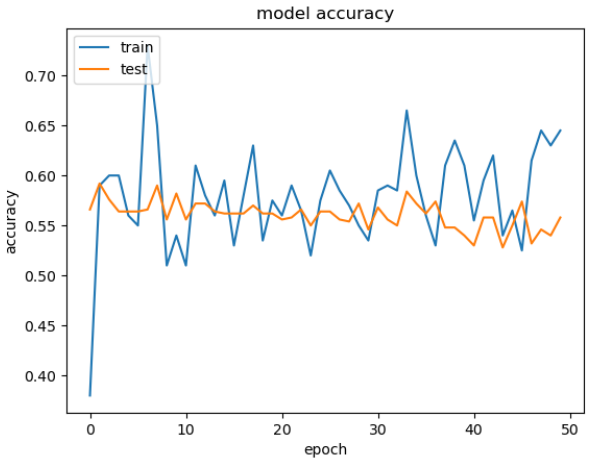

Figure 8: Model accuracy for one of the street view datasets for Atlanta, GA using 50 epochs

Some reasons for our model's inability to train well may be due to the invariability of the images. In all our datasets, we saw that a large number of images were simply of the road, which is unsurprising since we are using street view

imagery. At each location, as we rotated the camera angle, we would have at least a couple images that were simply of the road with little useful information. Additionally, in the Arizona dataset, for example, we saw many similar images of houses that were not particularly different or were simply desert landscapes.

Figure 9: middle-income ($63K) area image on the left vs high-income ($143K) area image on the right

The similarities between so many images in our dataset made them difficult for us to differentiate just by looking at them, and our model likely had similar difficulty telling apart images of higher and lower income areas.

5.2 Satellite Image Models: Unregularized

Our initial models performed poorly until we stumbled upon a high-performer on the Atlanta dataset with an incredibly simple model. This model had a single fully connected layer with size 150 and achieved 60% accuracy.

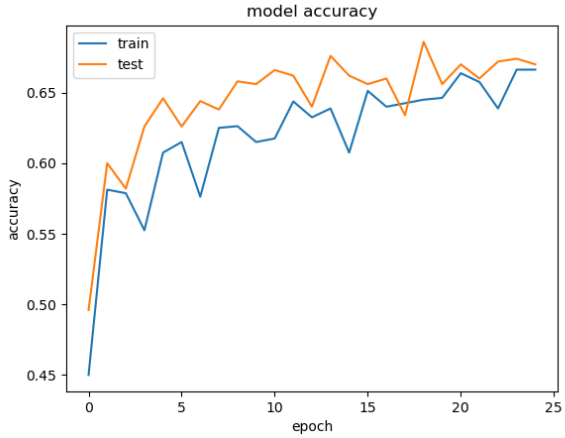

Figure 10: Model accuracy for one of the satellite image datasets for Atlanta, GA using 25 epochs

We then developed two unregularized models, one using the k-means clustered income brackets, and another using income brackets with cutoffs at $75,000 and $150,000.

For the k-means clustered models, we found them to perform extremely well on the test set. We achieved the following accuracies on the test sets:

- Atlanta, GA model: 81%
- Washington, D.C. model: 86%
- Providence, RI model: 77%

Unfortunately, despite our strong performance across all three models, we realized from the graphs of our model loss that we were severely overfitting our training data. Even our Washington, D.C. model, which performed the best on its test set, was still overfitting, as can be seen below in Figure 11.

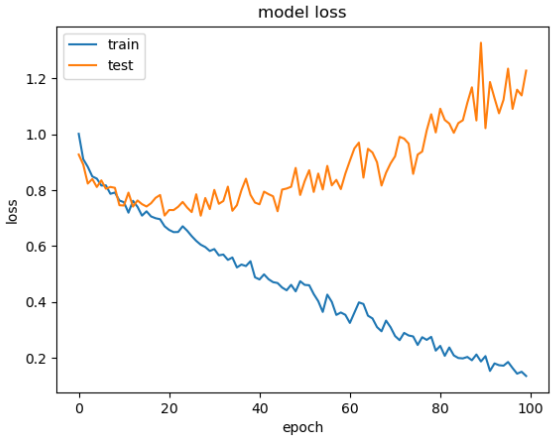

Figure 11: Model loss for the unregularized Washington, D.C. model using k-means clustered income brackets. Here, the “train” refers to training loss and the “test” line refers to validation loss.

We then created models using the income brackets with cutoffs at $75,000 and $150,000 and achieved the following test accuracies.

- Atlanta, GA model: 75%
- Washington, D.C. model: 85%
- Providence, RI model: 83%

We see here slightly decreased performance compared to our models based on k-means clustered income brackets. However, looking at the model loss curves revealed similar problems with overfitting.

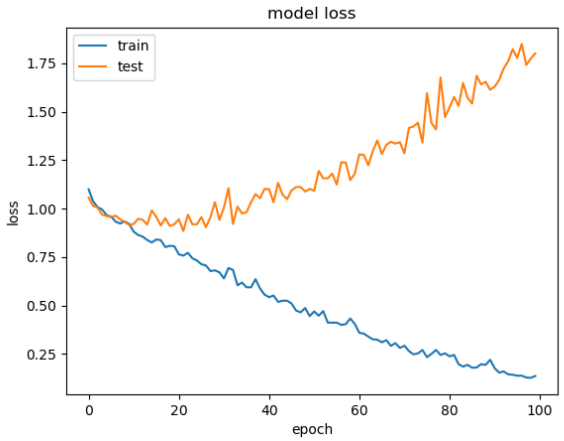

Figure 12: Model loss for the unregularized Providence, RI model using income bracket cutoffs of $75,000 and $100,000. Here, the "train" refers to training loss and the "test" line refers to validation loss.

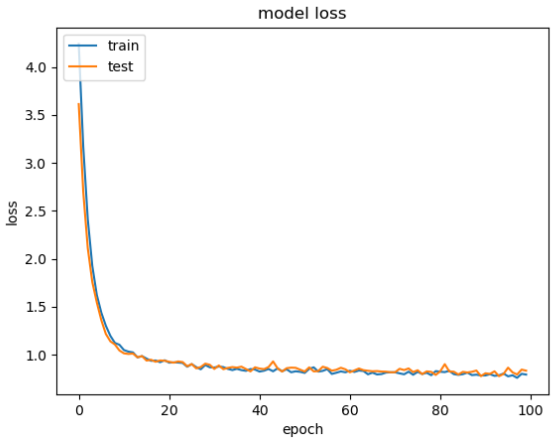

Figure 13: Model loss for the regularized Atlanta, GA model using k-means clustered income brackets. Here, the "train" refers to training loss and the "test" line refers to validation loss.

5.3 Satellite Image Models: Regularized

Given the overfitting we saw from previous models, we attempted to regularize our weights, first using a dropout layer, which yielded dismal results. We replaced with L2 weight regularization and trained two sets of models, one with k-means clustered income brackets, and another with fixed income brackets. Our fixed income brackets here used cutoffs at $60,000 and $100,000. This is different from the cutoffs used in the fixed income bracket unregularized models, which underperformed the k-means clustered models.

For the regularized k-means clustered income bracket models, we achieved the following test accuracies.

- Atlanta, GA model: 74%
- Washington, D.C. model: 64%
- Providence, RI model: 59%

Strangely, our performance here decreased on the test set even though our models were no longer overfitting.

We saw similar performance for the regularized models using fixed income brackets with cutoffs at $60,000 and $100,000. They achieved the following test accuracies:

- Atlanta, GA model: 81%
- Washington, D.C. model: 63%
- Providence, RI model: 60%

As with the models using k-means clustered income brackets, these models did not overfit the training data, but also did not perform particularly well with the exception of the Atlanta, GA model, which outperformed the other two in all the regularized models. Looking at these results, one possible reason for this could be insufficient training data. Among the regularized models, Atlanta, GA always performed the best, followed by Washington, D.C. and then Providence, RI. These were also in descending order with respect to the number of images in their datasets, with around 6100, 4200, and 3200 images respectively. Another reason, perhaps, was simply insufficient training time. With further training and more data, it could be true that our regularized models were, in fact, better than our unregularized models.

5.4 Best Satellite Image Models: Confusion Matrices

In our best regularized and unregularized models, it seems that our models had the most trouble correctly identifying low income neighborhoods. We are uncertain if this is due to the distribution of images, or perhaps the features that our models are activating on. This problem can be seen below in our confusion matrices in Figure 14. Additionally, regularization seems to drive our models to predict medium income more often.

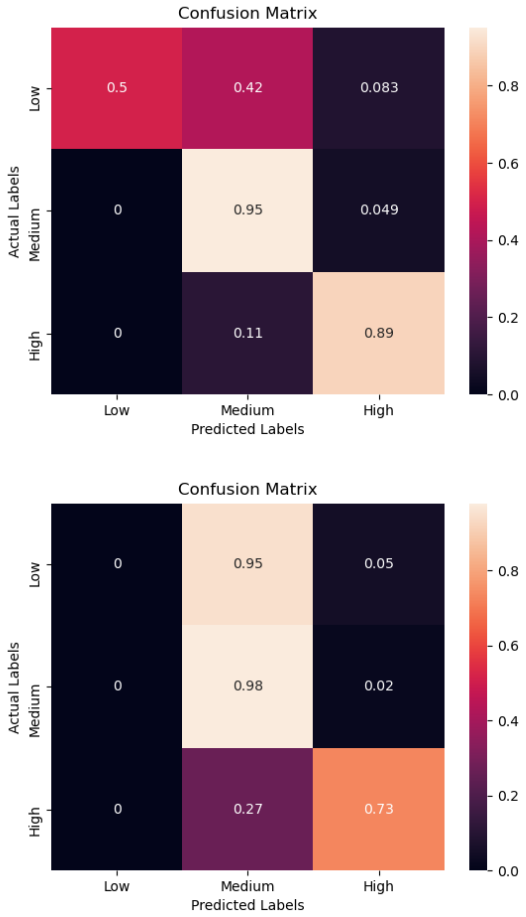

Figure 14: Confusion matrices based on validation data for the unregularized Washington, D.C. model using k-means clustered income brackets (top) and regularized Atlanta, GA model using fixed income brackets (bottom).

5.5 Best Satellite Image Models: Saliency Maps

In hopes of better understanding and interpreting our model, we created saliency maps, which visualize the parts of an image that contribute most to its final classification. Looking through a number of images, we could not identify clearly what our model was activating on. One thing that seemed to be true, as seen in Figures 15 and 16, are that our model activates on regions with large flat surfaces, be it a football field or a large building roof.

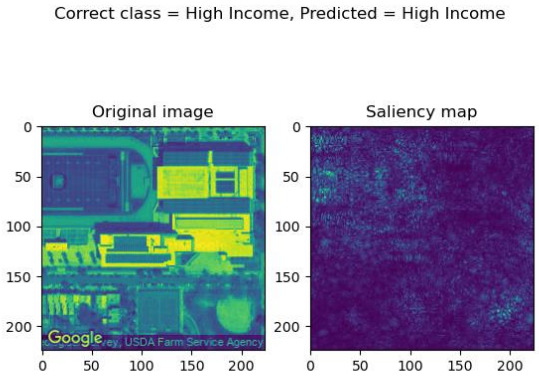

Figure 15: Saliency map example from training data for unregularized Washington, D.C. model using k-means clustered income brackets

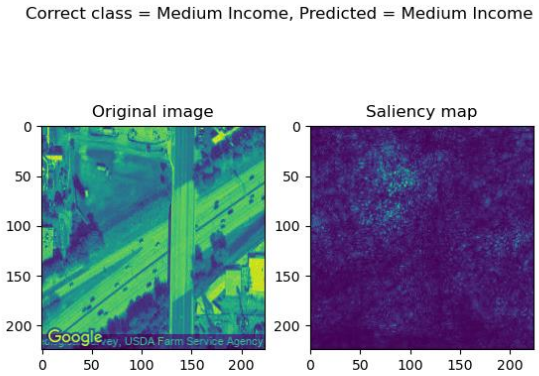

Figure 16: Saliency map example from training data for regularized Atlanta, GA model using k-means clustered income brackets

Additionally, our saliency maps helped us confirm that our model was, at the very least, looking at the significant areas of an image. In Figure 17, we see that one of our models activates much more on the region of land neighboring a body of water, while mostly ignoring the region in the water. Similarly, in Figure 18, our model identifies regions near a building in the image and seems to ignore the dense trees around it. Both of these, taken from regularized and unregularized models, indicates that our model is roughly choosing the right place in the image to look.

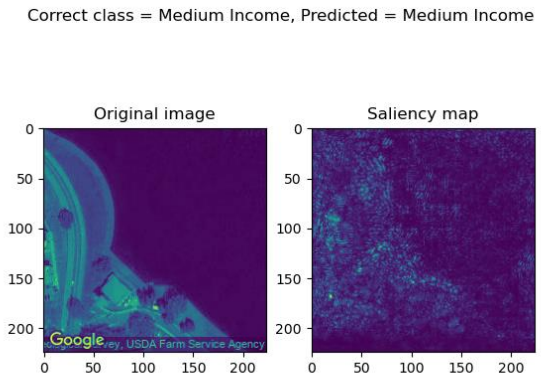

Figure 17: Saliency map example from training data for regularized Washington, D.C. model using fixed income brackets

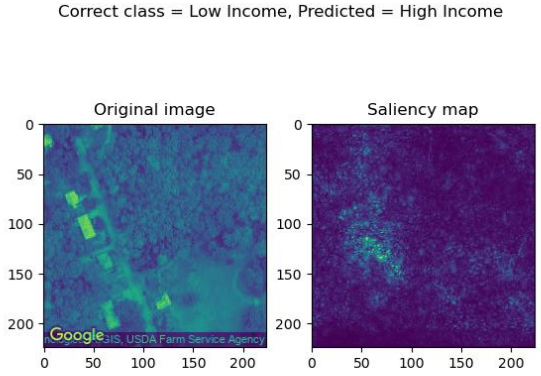

Figure 18: Saliency map example from training data for unregularized Providence, RI model using fixed income brackets

6. Conclusion and Future Work

In conclusion, our highest performing algorithm was a simple unregularized CNN that used our K-means generated buckets. Our final model was incredibly simple, with only two fully connected layers with 150 neurons.
A promising direction of future work is to gather data across a series of diverse cities and attempt to train a model on the combined, larger dataset of cities. This model might then be able to generalize to new, unseen cities.
Another direction of future work is more regularized models to match our model's performance without overfitting. We were surprised by the performance of our unregularized model and think that with a better regularized model could be designed. We would just need to cross validate across the shape of the network and the amount of regularization. With different models, a much longer training period for fine tuning could help reach the performance by the unregularized model.

7. Acknowledgments

Johannes Hui did the data collection and preprocessing using google APIs and K-means. Antonio Ferris assembled the model and wrote the code to visualize accuracies, loss, and confusion matrices. Johannes wrote the code to visualize the saliency maps, using the code at https://www.machinecurve.com/index.php/2019/11/25/visualizing-keras-cnn-attention-saliency-maps/ as a starting point. Johannes wrote most of the paper, with Antonio providing assistance for the Abstract, Methods Experiments, and Future Work sections.

References

[1] Picchi, Aimee. "9 American Cities with the Worst Income Inequality." CBS News. CBS News, February 8, 2018. https://www.cbsnews.com/media/9-american-cities-with-the-worst-income-inequality/.https://pypi.org/project/censusgeocode/

[2] E. Glaeser, S. Kominers, M. Luca, N. Naik. Big Data and Big Cities: The Promises and Limitations of Improved Measures of Urban Life. The National Bureau of Economic Research, 10.3386/w21778, 2015.

[3] A. Archarya, H. Fang, S. Raghvendra. Neighborhood Watch: Using CNNs to Predict Income Brackets from Google Street View Images. http://cs231n.stanford.edu/reports/2017/pdfs/556.pdf

[4] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, E. L. Aiden, and L. Fei-Fei. Using deep learning and google street view to estimate the demographic makeup of the us. arXiv preprint arXiv:1702.06683, 2017.

[5] S. Heitmann, S. Buri. Poverty Estimation with Satellite Imagery at Neighborhood Levels: Results and Lessons for Financial Inclusion from Ghana and Uganda. IFC 2019. https://www.ifc.org/wps/wcm/connect/2cae89ee-dea3-4a7e-ba79-77c9011cbd0f/IFC_2019_Poverty+Estimation+with+Satellite+Imagery+at+Neighborhood+Levels.pdf?MOD=AJPERES&CVID=mHZhcxB

[6] S. Pandey, T. Agarwal, N. Krishnan. Multi-Task Deep Learning for Predicting Poverty from Satellite Images. AAAI Conference on Artificial Intelligence, North America, Apr. 2018. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16441

[7] N. Jean, M. Burke, M. Xie, W. Davis, D. Lobell, S. Ermon. Combining satellite imagery and machine learning to predict poverty. Science, 10.1126/science.aaf7894, 2016.

[8] A. Vallebueno. Using Satellite Imagery and Computer Vision to Estimate the Income Levels of Mexican Households. RPubs, 2019. https://rpubs.com/andyvallebueno/Mexican_income

[9] A. Najjar, S. Kaneko, Y. Miyanaga. Crime Mapping from Satellite Imagery via Deep Learning. IEEE WACV 2017. arXiv:1812.06764, 2018.
[10] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. arXiv, arXiv:1502.03044v3, 2016.
[11] Frankenfield, Jake. "Which Income Class Are You?" Investopedia. Investopedia, February 5, 2020. https://www.investopedia.com/financial-edge/0912/which-income-class-are-you.aspx.
[12] U.S. Census Bureau; American Community Survey, 2018 American Community Survey 5-Year Estimates Subject Tables, Table S1903; < https://data.census.gov/cedsci/table?t=Income%20and%20Earnings&tid=ACSST5Y2018.S1903&vintage=2018&hidePreview=false&g=0500000US44007.140000>
[13] "LibGuides: Finding Census Tract Data: About Census Tracts." MSU Libraries. Michigan State University. Accessed June 8, 2020. https://libguides.lib.msu.edu/tracts.