Training Convolutional Neural Networks on Satellite Imagery to Predict U.S. population Health Status

Raja Noureddine [*]
Jackson School of Global Affiars
Yale University
New Haven, CT 06520
raja.noureddine@yale.edu

Inyoung Shin
Department of Computer Science
Yale University
New Haven, CT 06520
inyoung.shin@yale.edu

Lihao Zheng
Department of Electrical Engineering
Yale University
New Haven, CT 06520
lihao.zheng@yale.edu

Abstract

This project explores the use of Transfer Learning for Convolutional Neural Networks (CNNs) to create a model that can predict variations in health status across the United States (US) based on satellite imagery. We utilize transfer learning from the model developed by a prior work, which was originally designed and trained to predict economic outcomes. We extract convolutional features from the pre-trained model and re-train custom-designed linear layers to predict new health labels. However, our experimental model did not perform satisfactorily in the prediction task. Nonetheless, our project highlights the strengths and limitations of Transfer Learning in forecasting public health-related measures and emphasizes the significance of "balanced data." We conclude our project by discussing future directions to enhance the model's performance, with a focus on addressing imbalanced data and varying data distributions that restrict our ability to achieve higher accuracy.

1 Introduction

Socioeconomic status is not evenly distributed across geographic regions, as evidenced by various studies [6, 36, 23]. People with higher or lower socioeconomic status tend to live in clusters within specific areas. These social and economic disparities between regions, such as unequal access to job opportunities, education, and healthcare, contribute to and exacerbate patterns of socioeconomic inequality [7, 27, 4]. Such disparities also have significant implications for health inequalities that are geographically based [22].

In recent years, the intensity of these patterns has increased, and addressing the issue has become a top priority not just in the United States [2], but across the globe [29]. Collecting data and identifying trends related to social and health geographical inequality are accordingly essential steps toward addressing this issue. While survey methods such as the census have long been considered the most reliable approach for this process, they are expensive and not always feasible in the short term. In particular, developing countries face challenges in conducting representative surveys [28].

[*] Link for presentation video: `https://youtu.be/kAcdVWGD5Uc` ; The link for code repository : `https://drive.google.com/drive/folders/1chxCm1V3dKPOxO7Oxa1rBntxYjeKgdAS?usp=share_link`

Preprint. Under review.

There is a growing interest in utilizing machine learning techniques to analyze geographical images, particularly satellite imagery [12, 30, 14, 25, 1]. Satellites have been consistently producing geographic images such as night light luminosity imagery since 1980 [37]. With advancement of satellite technologies, several governmental and private organizations, such as Google, now offer high-resolution daytime satellite imagery on a 24/7 basis. Research has shown the potential of using neural network models trained on satellite imagery to predict geographic patterns in socioeconomic status, which may overcome the limitations of traditional survey and statistical methods [12]. While some researchers have attempted to use convolutional neural networks (CNNs) to predict economic aspects such as poverty and income levels over the past five years [30, 14], identifying health-related statuses in geographic areas is still a work in progress and has yet to be fully established.

Our project is inspired by previous research on neural networks with satellite imagery. We specifically aim to develop a CNN model to predict a geographic health status indicator, such as the percentage of adults without health insurance, based on satellite data. Training CNNs poses a number of challenges. Most relevantly for our topic, training a CNN to predict socioeconomic labels from satellite imagery requires significant computational power. For example, [14] find that training a CNN to predict census block-group level income, using over 200,000 separate images, takes approximately a month, even when training on "dozens of GPUs." Thus, rather than training a model from scratch, we propose to experiment with transfer Learning, which aims to "transfer learning from *source* to *target* domains . . . by adapting classifiers trained for other categories (p. 1718)" [24]. In the domain of CNN, this entails re-using the mid-level feature-extractors of a trained model, but rebuilding other layers of the model on a new dataset. Thus, the objective of our research is twofold: firstly, to expand the focus of previous work and generalize it to public health contexts, and secondly, to test the applicability and practicality of Transfer Learning for CNNs in the domain of public health. We hope to provide valuable insights into the potential applications of neural networks in predicting public health-related outcomes using satellite imagery.

2 Related Works

2.1 Application of CNNs to satellite imagery data

During the early stages of utilizing machine learning techniques to estimate socioeconomic levels in geographical areas, researchers used night light luminosity satellite imagery as input data [10, 5, 26]. Models trained on this imagery provided relatively accurate estimations of poverty at the country level [10, 26]. However, this imagery has low resolution, which ultimately limits the model's capability to predict poverty levels in urban or developed areas where the levels of luminosity are relatively consistent [12, 1]. Furthermore, the night light luminosity satellite imagery is not an ideal proxy for other specific social activities beyond economic wealth [12].

Recent advancements in earth observations and computer science have provided opportunities for researchers to develop neural network models that use higher-resolution daylight satellite imagery. For instance, Jean et al. [12] proposed the CNN model to extract image features from daylight satellite images that explain the variation in local-level economic outcomes in five African countries: Nigeria, Tanzania, Uganda, Malawi, and Rwanda. Specifically, they trained the model to capture the features corresponding nighttime light intensities in daytime satellite images. Head et al. [9] expanded on the work conducted by Jean et al. and demonstrated that the deep learning approach proposed by Jean et al. was applicable to other measures of development, such as educational attainment and access to drinking water, across other countries and continents.

Rolf et al. [30] focused on developing a CNN model that had been trained solely on daylight satellite images. They first encoded the high-resolution daylight satellite imagery and then discussed how to use these encoded embeddings for various prediction tasks, including forest cover, house prices, and road length. Their work contributed to extracting a single set of features in the daylight satellite imagery that could reduce computational costs. Unlike Rolf et al's work focused on the generalized model, Abitbol and Karsai [1] attempted to specify neural network models to predict socioeconomic status across cities in France. Similarly, Khachiyan et al.[14] complemented Rolf et al.'s approach by focusing on implementing CNN models specialized in predicting income levels and changes in the U.S. Although the methods proposed by Khachiyan et al. are computationally demanding, their work showed high accuracy in estimating income levels in the smallest U.S. census areas.

2.2 Transfer learning

Despite their power, traditional approaches to Machine Learning are limited by their need for often large labelled data-sets for training [35]. In addition, models trained on large (e.g., image) data sets may require significant computation time and resources to train. Re-using pretrained models to perform prediction tasks in different domains to the one in which they were trained would significantly alleviate this problem - for example, by allowing a model trained to predict economic outcomes, to predict health outcomes. However, as noted by [35], traditional ML "is characterized by training data and testing data having the same input feature space and the same data distribution." Differences in these distributions can lead to diminished performance.

Transfer learning [35], aims to solve this issue, allowing knowledge developed in one domain to be transferred to tasks in another. Often this takes place by selectively reusing parts of pre-trained models, while adding new architectural features aimed at bridging between the old and new domain. Regarding transfer learning on CNNs whose tasks is prediction labels based on image data, Oquab et al.[24] proposed transferring weights from the convolution or inner layers of a trained model, to a new one. According to Oquab et al.[24], internal layers of a trained CNN can serve as “generic extractors of mid-level image representation”, which help identify generic features of images that can apply to multiple specific learning tasks. Consequently, by transferring these pretrained features to a new model, while building and training new sets of linear layers that create complex non-linear combinations of feature maps extracted using the trained filters, it is possible to apply a pre-trained CNN to a new domain.

In this paper, we aim to extend the model developed by Khachiyan et al. [14] to predict economic outcomes such as income level, to the domain of health-status predicting by employing transfer learning. Khachiyan et al. made their data, code, and output needed for transfer learning available. Compared to other countries, the U.S. Census Bureau provides community-based survey results on health conditions every year. By combining this data with Khachiyan et al’s approach, we seek to generalize CNN models to predict public health-related conditions in U.S. geographic areas.

3 Data and Methodology

3.1 Data

3.1.1 Satellite image data

The model developed in Khachiyan et al.[14] is trained on (80 x 80 x 3) (RGB) satellite imagery from the NASA LandSat satellite. In for trained filters of this model to transfer to our model, we therefore use imagery of the same dimension. We create a new dataset of images taken by the Eurosat Sentinel Satellite and sourced from Google EarthEngine. Images were restricted to the continental United States, and to the highest population-density areas of the US that together represent over 80% of the US population. We followed the data processing procedures proposed by Khachiyan et al.[14]. In total, our dataset included over 1.2 million images, including images of over 215,000 individual localities, each over 4 years from 2016 to 2019. The Appendix contains further information on the (involved) data extraction and cleaning process.

To accurately estimate the out-of-sample performance and determine the optimal capability of the model, we partitioned the dataset into three subsets: training, validation, and testing. With a dataset of approximately 1.2 million images, we allocated around 60% for training, while the remaining 40% was split evenly between validation and testing, with each subset comprising roughly 20% of the dataset. The training set is used to train the models, and the validation set is used to estimate the sample error for a given set of hyperparameters. We determined the final model based on the estimations from the validation set. The test set was used to test the generalizability of the final model.

3.1.2 Label data

We obtained census tract-level data from the Population Level Analysis and Community Estimates (PLACES) program (www.cdc.gov/places/), conducted by the Centers for Disease Control and Prevention (CDC) in collaboration with the Robert Wood Johnson Foundation and CDC Foundation. PLACES has extracted data on 24 health measures from the Behavioral Risk Factor Surveillance

System Survey and the American Community Survey. It has provided yearly model-based estimates of these measures at census tract levels for the 500 largest US cities since 2015, expanding to cover all areas of the country since 2018 (see Matthews et al. [20] for more information). Our focus was specifically on extracting data related to the percentage of adults without health insurance in census tract areas between 2017 and 2020 – that are hihgly correlated to income levels [21, 18], that our source model in transfer learning [14] predicted.

3.2 Methods

We build a CNN based on the CNN model developed by Khachiyan et al. [14]. We kept the structure of Khachiyan et al's model (e.g., CNN layers) and transferred weights from it to our base model. To enhance our prediction task, we re-architected a series of linear layers and trained them separately. In this way, our model used the same features as the model of Kachiyan et al. [14] , but trained on identifying different combinations and relationships between these features.

Our model is comprised of two parts: a convolutional part (for which we load weights trained by Khachiyan et al. [14], followed by a linear part (for which we train weights). The convolution section of the model includes nine convolutional layers, each using (3x3) kernels with a stride size of one pixel and ReLu activations. A Max Pooling layer (using a 2 x 2 pixel window) was included after each three convolutional layers. Figure 1 shows the overall structure of our model.

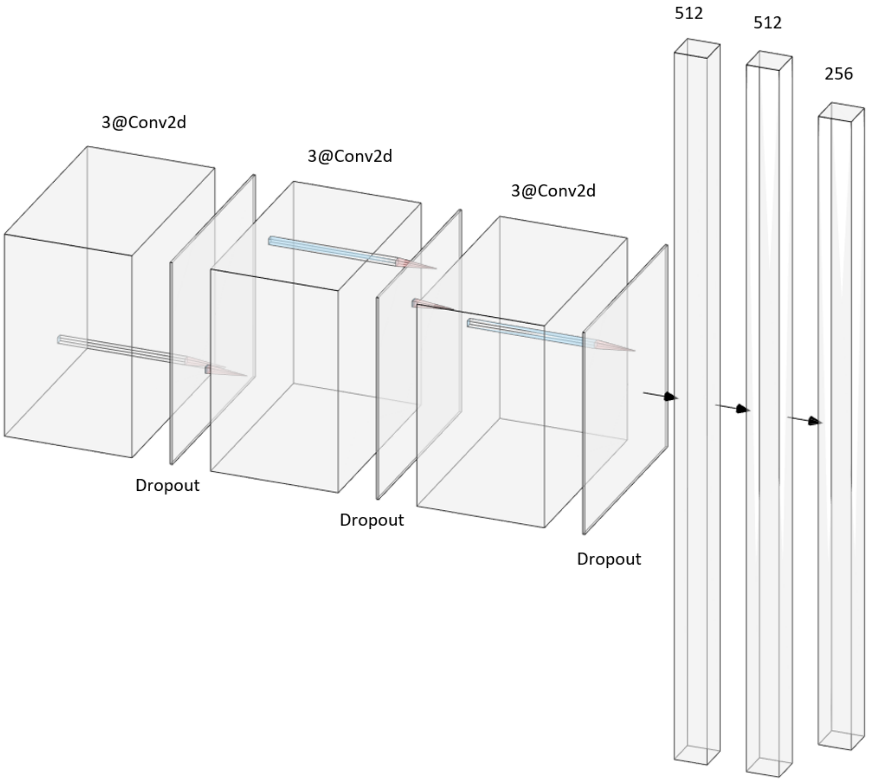


Figure 1: Default Model Architecture

4 Training Procedures and Results

4.1 Model predicting continuous label

In the first model, we attempted to predict the percentages of adults without access to insurance at geographic areas. The outcome labels were continuous levels, ranging from 0 to 100, but normalized 0 to 1 by subtracting the mean from each data point and a division by the standard deviation.

Once we obtained the outputs from the convolution blocks, we flatten them into a vector and then passed to fully connected linear layers that performed regression predicting the normalized continuous outcome.

The Model was trained to minimize the MSE of the prediction using the Adam optimizer. The dropout probability in hidden layers is fixed at 0.5. The fully connected layers also used ReLu activations and were regularized by an L2 norm penalty.

Figure 1 presents the results of the model predicting the percentages of adults without access to insurance at geographic areas (5.4 km^2). As our outcome variable was continuous, we reported R^2 values for model accuracy.R^2 is computed through the formula of $1 - \frac{SSR}{TSS}$. [2]

As seen in Figure 1, as the number of training epochs increased, MSE error decreased, indicating lowering loss within the training data set, but negative R^2 values were consistently observed in the training data set.

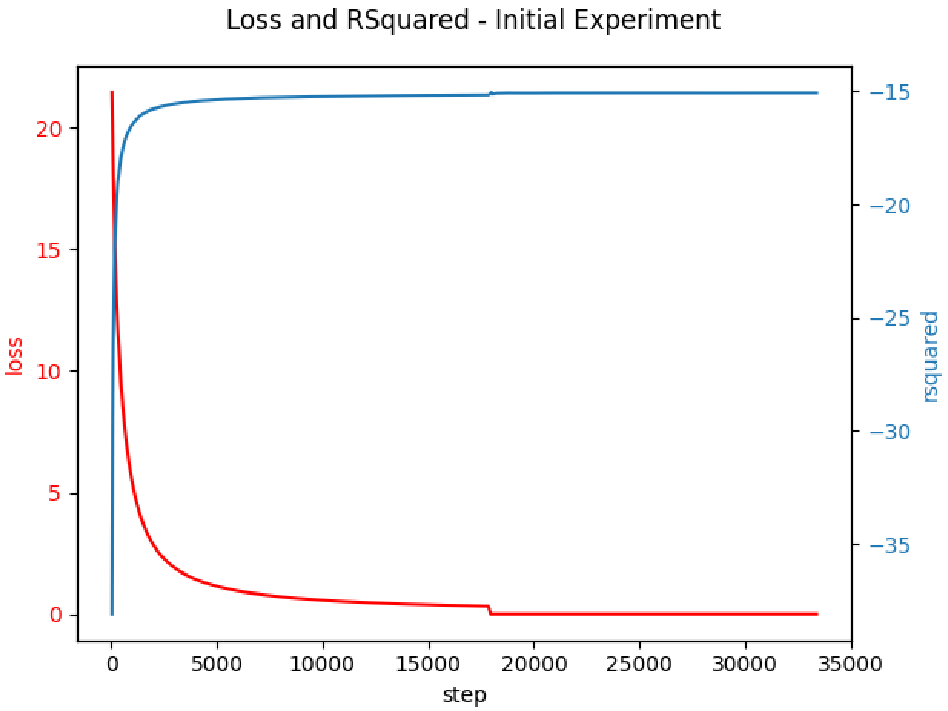


Figure 2: Results for Initial Experiment

Blue Line: MSE Loss; Red Line R^2

Although we do not have a clear explanation for these results, we suspect that our data may be imbalanced [8]. This means that the data are skewed in such a way that the majority of label points belong to zero, or that the events of interest occur rarely.

Learning algorithms based on imbalanced data sets can end up end up learning patterns that correspond to the majority of data points [8, 11, 34]. This issue is particularly common in tasks related to detecting health problems where majority of people are healthy or do not have issues[13]. In the US, since the Affordable Care Act went into effect in 2014, the number of adults without access to health insurance has dramatically declined [3]. This might partly explain the imbalanced nature of our data set.

As shown in Figure 2, We found that our data set were skewed toward Zero. Because we did not address this issue in our first CNN model, it is possible that our model was biased toward learning

[2] SSR is the sum of the squared difference between the actual value and the predicted value where TSS is the sum of squared differences between the actual value and the mean of actual value. $1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$

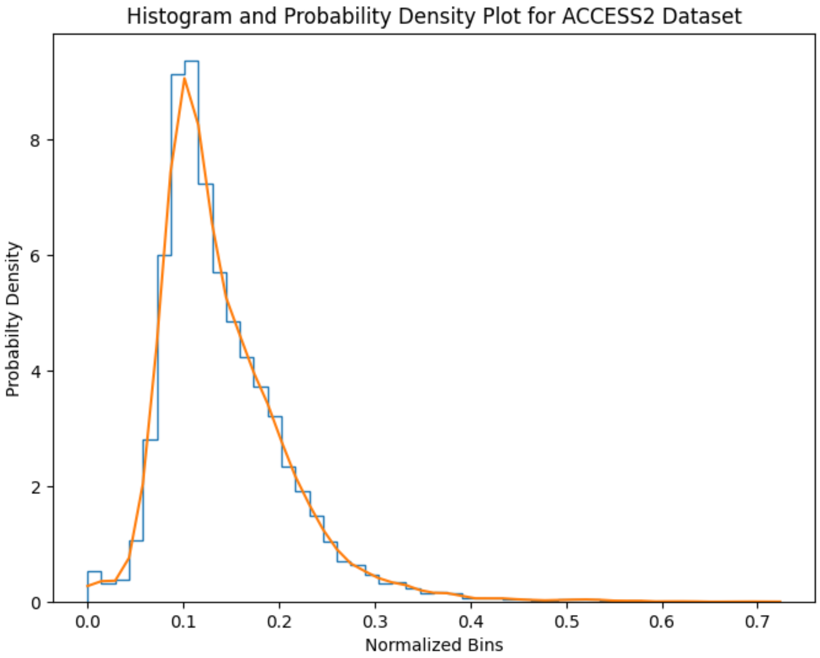


Figure 3: The Distribution of Continuous Label Data

imagery corresponding "zero" or "low score" outcomes. This could cause the decrease in loss but negative but r^2 value

4.2 Models classifying categories

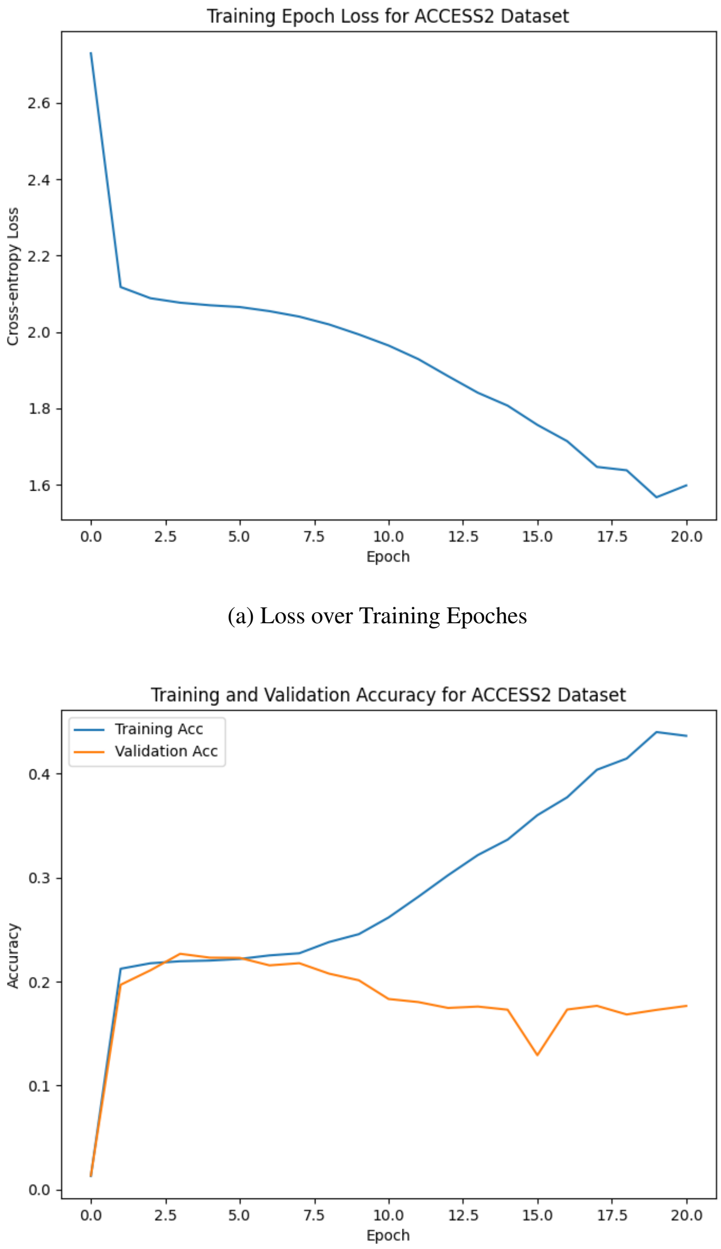
We re-conducted the model training by converting the continuous label data into discrete categories. Although we could lose some information by converting the continuous variable to the categorical one, we anticipated that this data transformation would make the differences between images corresponding to different labels more salient. The model would converge faster because it would classify a discrete number of categories rather than predicting continuous outcomes.

To our knowledge, there are no standard cut-offs used in the public health field regarding the lack of insurance access. Therefore, we used an ad-hoc approach to bin the continuous variables into 10 categories. Data points below 0.05 in the normalized score were assigned to the first category, while data points above 0.25 in the normalized score were assigned to the highest category. The range between 0.05 and 0.25 was divided into eight intervals of consistent width in the range of the normalized score.

As our task was converted to classification, we employed a different loss function, a categorical cross-entropy function to train the model. We also utilized the sigmoid for the final outcome layer, as our task was converted to classifying the outcomes from the final layers.

4.2.1 Model with weights transferred from Khachiyan et al.

As previously discussed, our initial model was trained using transfer learning, where the weights of the three convolutional blocks were transferred from Khachiyan et al. In our subsequent attempt, we still kept transferred the weights from Khachiyan et al. for the first three convolutional blocks. As we did before, the weights for the fully connected layers for the regression task were trained from the values at 0. Figure 3 illustrates the loss over the training epochs, while Figure 4 displays the training loss and accuracy rates for both the training and validation datasets. As shown in Figure 3, the loss



(a) Loss over Training Epoches

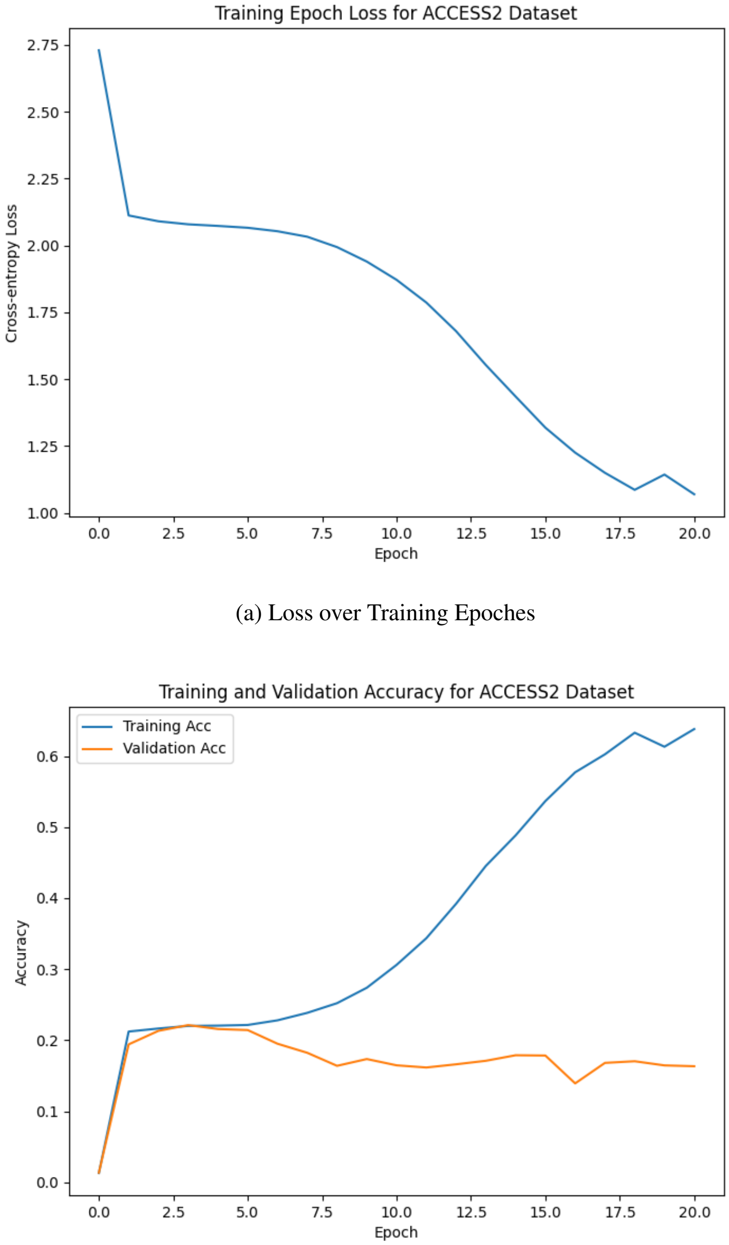
(b) Accuracy Rates

Figure 4: Results for the Second Experiment

decreased in the training dataset as the epochs progressed. However, the accuracy rate only increased to 0.42 in the training dataset, whereas the accuracy for the validation dataset remained at 0.2.

4.2.2 Model with one unfrozen convolutional block

To improve the model, we decided to unfreeze the weights of the convolutional blocks. Upon examining the weights of the third convolutional block transferred from Khachiyan et al, we noticed that the majority of the weights had the values of zero. Consequently, we re-trained the model with the weights of the third convolutional block unfrozen. However, for computational efficiency, we



(a) Loss over Training Epoches

(b) Accuracy Rates

Figure 5: Results for the Model with One Unfrozen Convolutional Block

kept the weights of the first two convolutional blocks frozen, thus sharing the same weights with those of Khachiyan et al merely for the two convolutional blocks. As illustrated in Figure 3, the loss values in the training dataset decreased for most of the epochs. Figure 4 depicts that the accuracy rate for the training dataset improved significantly, increasing to around 0.60. However, the accuracy rate for the validation dataset remained low at approximately 0.20, indicating that our trained model was overfitting and had low generalizability.

We considered another attempt where the model is trained after unfreezing all convolutional blocks to improve the capability of our model. However, we realized that training the weights for the convolutional blocks may not necessarily solve the problem, and may even exacerbate it. Due to time constraints for this project, we decided to halt the experiments and instead reviewed the literature and engaged in discussions on potential future directions to improve our approach. The details of these discussions will be presented in the following section.

5 Conclusion and Future Directions

Although we aimed to demonstrate the effectiveness of CNNs in predicting health measures in geographic areas closely related to socioeconomic status, our CNN model did not perform well in predicting lack of access to health insurance in these areas. While the exact reasons for the model's poor performance are unclear, we have identified potential contributing factors based on a comprehensive literature review and suggested the future directions.

5.0.1 Need to improve imbalanced data

Theoretically, machine learning models can achieve good performance regardless of data disproportion if the training data are well represented [15]. However, real-world problems often have higher complexity, making learning algorithms more sensitive to imbalanced data [13]. Our task also faced challenges with imbalanced data. To address this issue, we converted our continuous label variable into categorical ones. However, this may not be the most effective method. Our cut-off was based on our ad-hoc examination. Our cutoff point might be generalize to out of sample. Furthermore, this transformation did not change the variations across the data fundamentally. We merely converted the label continuous data to the categorical one. The differences of a majority images may remained trivial even after the transformation. The limitation of our current data may have caused the overfitting issue.

There are many research approaches to address the issue of imbalanced data (see Johnson and Khoshgofaar [13] for details), including random under-sampling (RUS) [16], (i.e., dropping some cases randomly from the majority group) and random over-sampling [32] (i.e., randomly adding samples to the minority group). We could try these method to resolve the imbalance issue in our data. Another option is to use a K-nearest neighbors classifier to remove majority samples based on their distance from minority samples [19].

Modifying learning algorithms using new loss functions can be another solution to address data imbalance. Indeed, Wang et al. [33] argued that MSE loss function poorly captures features from the minority group in data and propose two new loss functions that are more sensitive to the minority class: mean false error and mean squared false error. On the other hand, Lin et al. [17] presented the focal loss, which reshapes the cross-entropy loss to reduce the impact of majority groups on the loss.

As discussed, we suspect that the percentages of adults without health insurance in different areas may not be appropriate label data for supervised learning on satellite imagery. Using other health-related measures such as coronary heart disease, diabetes, and depression [31] could also be better approach to train the model with satellite imagery.

5.0.2 Feeding features related to controls

In CNN model, controlling certain features could be necessary. In fact, Khachiyan et al.'s work [14] served as the primary reference for our project. Their approach included incorporating local economic characteristics of geographic areas, which have been known to influence significantly to income levels, (e.g., gender and ethnicity ratios, as well as the employment status ratio of the population in those areas) as input features to control for their influence on health outcomes such as income levels. However, including controls into the model can lead to overfitting as well. Thus, We did not include such characteristics in our model. As discussed above, however, the accuracy in the trainding data was not overly high. Future research in this area could consider incorporating input features that control for such characteristics to improve the accuracy and effectiveness of the model.

5.0.3 Using other source models for transfer Learning

We utilized transfer learning based on the models proposed by Khachiyan et al.[14]. Their model was designed to predict income levels from satellite imagery, although it was not specifically developed for generalizing to other tasks. We presumed that our task of predicting the levels of populations without health insurance was similar enough to Khachiyan et al.'s task to leverage their model. The use of the transfer learning technique makes training a large complex model on a large (200GB+) dataset feasible in a short time frame. However, the use of transfer learning cannot ensure the model capability. We ended up partly using our trained weights for the convectional blocks. Although the access to health insurance is significantly correlated with income levels in geographic areas, it is possible that predicting it based on satellite imagery involve a different set of features and requirements. Transferring learning based on the model of Khachiyan et al. would not be suitable for our project. If that is case, transfer learning might be an ineffective approach for our project. If we have sufficient time and computational resources, we may consider training and optimizing a CNN model from scratch. Alternatively, we could explore using other pre-trained models that are more applicable to diverse prediction tasks, such as "Multi-task Observation using Satellite Imagery and Kitchen Sinks" developed by Rolf et al.[30].

5.0.4 Solving discrepancy between input data and label data in geographic levels

Khachiyan et al.[14] constructed their label data (i.e. income levels) at the census block level, which represents a geographic entity containing data from 600 to 3,000 people. In contrast, our outcome label variable, percentages of adults without access to health insurance, was estimated based on the larger census tract levels containing data from 1,200 to 8,000 people. We used the same size of satellite images as Khachiyan et al. (i.e. $2.4km$ * $2.4km$). The difference in label data estimates could potentially impact the capability of our model. To be specific, our input images may contain less underlying information for outcome variables compared to the dataset used by Khachiyan et al., as our outcome measure covered more people's characteristics. In fact, Khachiyan et al. reported that models trained with images covering smaller geographic areas (e.g. 1.2 km * 1.2 km) had lower generalize accuracy less underlying information about outcome variables. Moreover, in our data set, satellite images covering remote areas are more likely to share the same label data with neighboring images since they are covered by the same census tract district. This may cause the model to fail to learn relevant features related to access to health insurance across neighboring images. In the future, we should consider using satellite imagery that covers larger geographic areas if the outcome data is based on areas larger than census blocks.

5.1 Conclusion

Although our project did not result in a trained model with optimal capability, we still believe in the potential of CNNs to predict not only economic activities but also health-related measures in geographic areas. Collecting geographic data on health can be expensive and rare, especially in developing countries. Developing CNN to predict population health can assist researchers, practitioners, and policymakers in identifying health inequities in different geographic areas, and prioritize areas that require public health interventions at low cost. Therefore, efforts to explore the use of CNN for this purpose should continue. We hope that our attempts and suggestions can provide useful insights for future research, and that the advancement of CNN-based methods can contribute to promoting health equity and improving public health outcomes.

References

[1] Jacob Levy Abitbol and Márton Karsai. Socioeconomic correlations of urban patterns inferred from aerial images: interpreting activation maps of convolutional neural networks. *arXiv preprint arXiv:2004.04907*, 2020.

[2] María P Aranda, Ian N Kremer, Ladson Hinton, Julie Zissimopoulos, Rachel A Whitmer, Cynthia Huling Hummel, Laura Trejo, and Chanee Fabius. Impact of dementia: Health disparities, population trends, care interventions, and economic costs. *Journal of the American Geriatrics Society*, 69(7):1774–1783, 2021.

[3] Thomas C Buchmueller and Helen G Levy. The aca's impact on racial and ethnic disparities in health insurance coverage and access to care: an examination of how the insurance coverage expansions of the affordable care act have affected disparities related to race and ethnicity. *Health Affairs*, 39(3):395–402, 2020.

[4] Tim Butler and Chris Hamnett. The geography of education: Introduction, 2007.

[5] Xi Chen and William D Nordhaus. Using luminosity data as a proxy for economic statistics. *Proceedings of the National Academy of Sciences*, 108(21):8589–8594, 2011.

[6] C Cindy Fan and Emilio Casetti. The spatial and temporal dynamics of us regional income inequality, 1950–1989. *The Annals of Regional Science*, 28:177–196, 1994.

[7] Michael Greenstone, Richard Hornbeck, and Enrico Moretti. Identifying agglomeration spillovers: Evidence from winners and losers of large plant openings. *Journal of political economy*, 118(3):536–598, 2010.

[8] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[9] Andrew Head, Mélanie Manguin, Nhat Tran, and Joshua E Blumenstock. Can human development be measured with satellite imagery? *Ictd*, 17:16–19, 2017.

[10] J Vernon Henderson, Adam Storeygard, and David N Weil. Measuring economic growth from outer space. *American economic review*, 102(2):994–1028, 2012.

[11] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[12] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[13] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[14] Arman Khachiyan, Anthony Thomas, Huye Zhou, Gordon Hanson, Alex Cloninger, Tajana Rosing, and Amit K Khandelwal. Using neural networks to predict microspatial economic growth. *American Economic Review: Insights*, 4(4):491–506, 2022.

[15] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[16] Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing (ICIP)*, pages 3713–3717. IEEE, 2016.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[18] Vasanti S Malik, Walter C Willett, and Frank B Hu. Global obesity: trends, risk factors and policy implications. *Nature Reviews Endocrinology*, 9(1):13–27, 2013.

[19] Inderjeet Mani and I Zhang. knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126, pages 1–7. ICML, 2003.

[20] KA Matthews, JM LeClercq, B Lee, et al. Places: local data for better health, 2022.

[21] Michael Millman et al. Access to health care in america. 1993.

[22] Adler NE. Overview of health disparities. In M Williams GE Thompson, F Mitchell, editor, *Examining the Health Disparities Research Plan of the National Institutes of Health: Unfinished Business*, page 129–88. Natl. Acad. Press, Washington, 2006.

[23] Jan Nijman and Yehua Dennis Wei. Urban inequalities in the 21st century economy. *Applied geography*, 117:102188, 2020.

[24] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[25] Christian Otchia and Simplice Asongu. Industrial growth in sub-saharan africa: Evidence from machine learning with insights from nightlight satellite images. *Journal of Economic Studies*, 48(8):1421–1441, 2021.

[26] Maxim Pinkovskiy and Xavier Sala-i Martin. Lights, camera… income! illuminating the national accounts-household surveys debate. *The Quarterly Journal of Economics*, 131(2):579–631, 2016.

[27] Stephen J Redding and Matthew A Turner. Transportation costs and the spatial organization of economic activity. In *Handbook of regional and urban economics*, volume 5, pages 1339–1398. Elsevier, 2015.

[28] UN Data Revolution. A world that counts: Mobilising the data revolution for sustainable development. united nations: Independent expert advisory group on a data revolution for sustainable development, 2014.

[29] Thomas Rice, Pauline Rosenau, Lynn Y Unruh, Andrew J Barnes, Richard B Saltman, Ewout Van Ginneken, World Health Organization, et al. United states of america: health system review. 2013.

[30] Esther Rolf, Jonathan Proctor, Tamma Carleton, Ian Bolliger, Vaishaal Shankar, Miyabi Ishihara, Benjamin Recht, and Solomon Hsiang. A generalizable and accessible approach to machine learning with global satellite imagery. *Nature communications*, 12(1):4392, 2021.

[31] Andrew Steptoe and Paola Zaninotto. Lower socioeconomic status and the acceleration of aging: An outcome-wide analysis. *Proceedings of the National Academy of Sciences*, 117(26):14911–14917, 2020.

[32] Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pages 935–942, 2007.

[33] Haishuai Wang, Zhicheng Cui, Yixin Chen, Michael Avidan, Arbi Ben Abdallah, and Alexander Kronzer. Predicting hospital readmission via cost-sensitive deep learning. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(6):1968–1978, 2018.

[34] Gary M Weiss. Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1):7–19, 2004.

[35] Karl R Weiss and Taghi M Khoshgoftaar. An investigation of transfer learning and traditional machine learning algorithms. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 283–290. IEEE, 2016.

[36] Jeffrey G Williamson. Regional inequality and the process of national development: a description of the patterns. *Economic development and cultural change*, 13(4, Part 2):1–84, 1965.

[37] Jun Xiong, Prasad S Thenkabail, Murali K Gumma, Pardhasaradhi Teluguntla, Justin Poehnelt, Russell G Congalton, Kamini Yadav, and David Thau. Automated cropland mapping of continental africa using google earth engine cloud computing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 126:225–244, 2017.

Appendices

Link for Video

https://youtu.be/kAcdVWGD5Uc

5.2 Link for Code Repository

https://drive.google.com/drive/folders/1chxCm1V3dKPOxO7Oxa1rBntxYjeKgdAS?
usp=share_link

Data Processing

Satellite data

Satellite imagery was acquired from Google EarthEngine, an product made available by Google that contains satellite imagery from a range of different national and private space services. While Khachiyan et al.[14] employ imagery from the NASA Landsat satellite, we choose to use imagery from the EuroSat Sentinel satellite, which is more recent and offers a higher quality, providing a resolution as fine as 10m x 10m pixels.

We extracted (80 x 80) pixel square images, with three color bands (red, green, blue), to produce images with dimension (80 x 80 x 3). Using a pixel resolution of 30 m, each image covers a (2.4 x 2.4) km square of the Earth's surface. In an earlier iteration of our project, we attempted using 10 m pixel resolution, but this led to an infeasible dataset size of well over 5 terabytes. Khachiyan et al.found that employing higher resolution imagery does not lead to a meaningful improvement in model performance. However, given that they employ artificial resolution enhancement techniques rather than truly higher resolution images, we anticipate that our model would have improved better, had we had the storage and processing power to use 10 m pixels (i.e. 3 x 3 = 9 times the resolution).

Our dataset is restricted to 4786 'blobs' located within the continental United States. These blobs are not units of analysis and are not associated with label data. They are used only to restrict the geographical scope of the dataset, since downloading imagery for the entirety of the continental United States would lead to thousands of images of essentially uninhabited areas. The blobs are formed by Khachiyan et al. by first ranking all Census Blocks in the United States by population density, before choosing the top Blocks that together contain over 83 percent of the United States' population. Contiguous groupings of these Census Blocks were then created using GIS software, to create 4786 'blobs' that are used to filter data extraction from Google EarthEngine. Again, these blobs are used only to restrict the geographical regions where images are extracted. They are not units of analysis.

All in all, we extract approximately 1.2 million (80 x 80 x 3) images, comprising approximately 215,000 unique image locations, with an image for each of four years (2016, 2017, 2018, 2019) in each location. We use approximately 60% of our dataset for test, 20% for test, and 20% for validation. No further processing is performed on the images (other than converting between file types required to feed to the TensorFlow model).

Label data

We obtained census tract-level data on health populations from the Population Level Analysis and Community Estimates (PLACES) program (www.cdc.gov/places/), conducted by the Centers for Disease Control and Prevention (CDC) in collaboration with the Robert Wood Johnson Foundation and CDC Foundation. PLACES provided the esimated health measure data based on the census tract-level data every year since 2015. We downloaded each year CSV files from PLACE website (www.cdc.gov/places/) for the measure of portion of adults without health insurance in census tract areas between 2017 and 2020 in order to match satellite imagery one year before. The data collected from PLACES images were merged into one file.

Matching of Satellite and Label Data

Each image in our dataset is matched to label data drawn from the CDC's PLACES / 500 Cities dataset. This matching is performed using ArcGIS software. To perform matching we use latitude and longitude data extracted along with each image from Google EarthEngine, to identify the centroids of each image. Following this, ArcGIS is used to define a 2.4km x 2.4km square around each image centroid. (This step is performed in the file XX)

We use Shapefiles provided by the Census Bureau (Link) for each US state, and then run an intersection analysis to identify the overlap between image squares and census tracts. This intersection analysis produces a list of approximately 519,000 image-tract intersections (i.e., since each image might intersect multiple tracts, we end up with more intersections than there are images). We use this analysis to weight label data for each image, using the percentage of the image's area in a given tract, as a weight. For example, if we have:

Image	Tract	Percentage	Label
Image 1	Tract 1	50%	10
Image 1	Tract 2	50%	20

Then this is aggregated to:

Image	Tract	Percentage	Label
Image 1	Joined	100%	15

Explanation of Code

Data Download

/download_imagery/export_large_sentinel_imagery.py

Description: This file sets up the download of the Google EarthEngine files, specifying the dataset to be used, area to be downloaded over, channels to download, image size and resolution. Once run, the script pushes commands to the Google EarthEngine server, which prepares the images in GeoTIFF format and places them in a Google Drive folder. While this script only takes approximately 2 hours to run, the extraction process on Google EarthEngine took approximately 4-5 days.

Detail:

- The data set is set as ("Copernicus/S2"). This refers to the Sentinel-2 MSI: MultiSpectral Instrument, Level-1C released by the European Space Agency (ESA).
- The function *cloudMask* is used to remove pixels with high cloud cover
- We select the channels B4, B3, B2, and B8, corresponding to Red, Green, Blue, and Near Infrared (that we later drop)
- We import the blobs object, which restricts the image export to 4786 zones in the United States with the highest population density, comprising approximately 83% of the US population. See above for further detail.

/download_imagery/download_data.py

Description: This file downloads the GeoTIFF files created in the step above, to a local folder (i.e. to the clusters). Downloading data usually takes approximately 48 - 96 hours.

Detail:

- The data is downloaded locally (i.e. to the clusters), with all images placed into a a H5 file. H5 files are used for storing large data sets efficiently in a large table. The *PyTables* package is used to do this. Each row in the table comprises seven images (one for each year from 2016-2022) (i.e., the first seven columns, each contain 80 x 80 x 3 images), a latitude number, a longitude number, an urbanization percentage, and an image ID.
- Our final H5 file was approximately 300GB.
- This script also produces a CSV file that is used in further steps to match images with their labels.

- Note that this script requires the use of the *PyDrive* package to efficiently download from a Google Drive folder locally. A range of helper functions are used to do this.

Label Preparation

/gis_analysis/gis_analysis.ipynb

Description: This file is used to calculate the intersection between images and census tracts, in order to later map labels to images. It requies the use of specialized GIS software.

Detail:

- Must be run on a computer with ArcGISPro installed, within an ArcGIS Environment We used Yale GIS Online, which provides access to remote desktops with ArcGIS installed.
- Running this requires two main inputs:
  - The CSV file containing the latitudes and longitude coordinates of image centroids, produced in the step above.
  - 51 Tiger/Line Census Tract Shapefiles, downloaded from the US Census Bureau. These are the shapefiles that define state and Census Tract boundaries in a GIS Map environment.
- The intersections are calculated by:
  - First, creating a square around each image centroid
  - Secondly, calculating the intersection between this square (representing the image boundaries) and the tracts around it. In many cases, an image lies entirely within a single tract, but in others, an image may cross multiple tracts, in which case its labels are weighted averages of the relevant tracts.

Output: This analysis outputs a CSV file containing the intersections between each image and each census tract. We have 215,000 images and approximately 520,000 intersections (since some images intersect with multiple census tracts). We provide an extract containing the first 10,000 rows of the output file to assist in interpretation.

/create_labels/merging_label_data.ipynb

Description : This file is used to create our initial label dataset. It takes data from the CDC PLACES / 500 Cities dataset and associates them with images using the tract ID (GEOID) to complete the merge

/create_labels/aggregate_labels.ipynb

Description : This file is used to aggregate label data for images spanning multiple tracts, by calculating a weighted average of the labels for different tracts.

Detail

- Aggregation takes place as described in the subsection *Matching of Satellite and Label Data*. Effectively, where an image spans multiple tracts, we label the image with a weighted average of the outcome label, in each of the tracts it spans. The weights correspond to the percentage of the image area in each tract.

Data encoding and final Preparation

prepare_data/prepare_data.py

Description: Combines our 300GB H5 file containing all images and a CSV file containing all labels, into .tensorflowrecord files which can be directly fed in to train the model

Detail

- Initially, label data are stored as percentages between 0 and 100. We use the *scikit-learn.preprocessing* module in order to normalize our label data such that it is always between 0 and 1.

- We loop through the H5 data file, and incrementally pair them one-by-one with the relevant label in the CSV file.
- We choose to use images only for 2017-2020 inclusive (corresponding to labels from 2019-2022).

Output: Outputs three .tfr files, one each for training, testing, and validation. The training data file is approximately 40GB, while the remaining two files are 14GB each. Each file comprises (image, label) samples, where the image is a satellite image for a single year, and the labels are health outcomes (e.g., obesity percentage) corresponding to that image and year. Each example contains *all possible health outcomes* but in a later stage we choose to train on only two outcomes.

Model definition

/models/modelDef_categorical.py

Description : This file is used to create our CNN, by first transferring weights from the pre-trained CNN created by [14]. We freeze the first two convolutiuonal blocks from the existing model, and re-train the final convolutional block, aswell as a series of linear layers.

Detail

- Transfer from pre-trained model created by [14]
  - The function *conv_block* defines a convolutional block comprising of three separate convolutional layers, each using a kernel size of (3x3) and a stride of 1 pixel, followed by a max pooling layer.
  - The function *dense_block* defines a series of dense linear layers separated by dropout layers which aim to reduce overfitting.
  - The function *make_level_model* constructs a CNN using a series of three convolutional blocks, followed by a dense block.
  - The function *base_modelDef* creates a CNN containing only convolutional layers with the same architecture as that used by [14]. The function then transfers the pre-trained convolutional weights one-by-one to the new model. The output of this function is therefore the three convolutional blocks created by [14], loaded with pre-trained weights.
- Definition of new model for training
  - The function *modelDef* creates a CNN comprised of two parts. First, it loads the pre-trained convolutional layers created in textitbase_modelDef, before freezing these layers so that their weights are not updated in training, in line with our transfer learning approach. These pre-trained layers are followed by a dense block whose weights are unlocked for training. Finally, the function creates an output layer comprising of 10 logits using a sigmoid activation function. The output of the model can therefore be interpreted as a 10-vector representing the probability that the image label is in each of 10 classes.

Training and results

/model_training/model_training.ipynb

Description : Trains the models and produces basic results output

Detail

- Data extraction and final preparation
  - The data-set is extracted using the *get_dataset*. This function takes a series of sharded .tfr files as input, and completes the final preparatory steps before training. (Note: we have not shown the sharding process here. This process simply involves taking the large 40Gb training .tfr file and splitting it into smaller pieces to reduce the memory requirements of running the model.)
  - The functions *scalar2vec* and *labelsort* are used to convert continuous labels into one-hot categorical vectors. As we discuss in the main body of our paper, we find that

  - the distribution of our raw label data (which is highly peaked) requires us to convert continuous labels into categorical labels in order to facilitate training.
  - We use 10 bins for our categorical label (percentages of adults without access to insurance at geographic areas). The line *label_sort(sample[1],bin_num=10,low=0.05,high=0.25,batch=True)* places all label values below 0.05 or above 0.25 into the top and bottom categories, and and distributes labels between these boundaries over the eight remaining categories.
- Model training
  - The function *model_train* sets up the final training loop.
  - We use categorical accuracy as our training and validation metrics. We update the training accuracy after every batch, and the validation accuracy after every epoch.

Image Sample

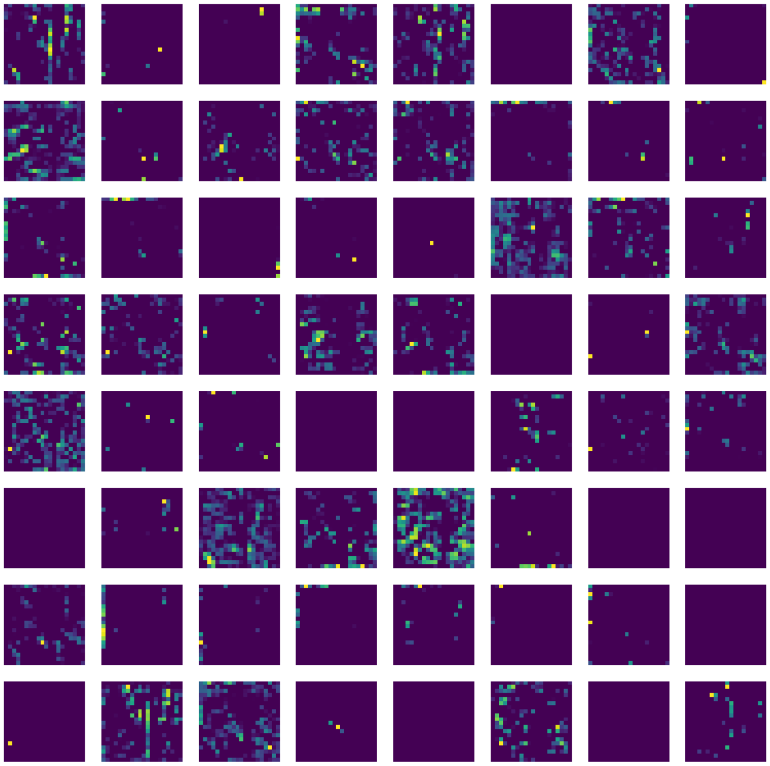


Figure 6: Samples of embeddeings outputted after the second convolution blocks