Analyzing LiDAR images – a Deep Learning Approach

Prepared by: Zhe Rao

Executive Summary

This report examines in detail if LiDAR images could be used to determine the education deprivation index with deep learning techniques. Specifically, two convolutional neural network architectures were examined: 1) VGG-16 and 2) ResNet50V2. The fundamental difference and performance between the models are explained and compared. In addition, a technique called GradCAM is implemented to deduce how the network is learning the education deprivation information from the LiDAR image. In the last section of the report, some privacy and ethical issues were discussed regarding using LiDAR images to measure multidimensional deprivation.

Motivation and Hypothesis

A light detection and ranging (LiDAR) image gives a graphical representation of the altitudes of objects in a certain area. An airplane sends light pulses to the ground and measures the time for the pulses to return, the longer the time, the lower the altitude at that specific point. Hence, the information from LiDAR images is purely geographical. The question remains, can we determine the education deprivation level from those images? I hypothesize a negative response to the question.

Nieuwenhuis and Hooimeijer's research revealed that education level can be negatively associated with poverty, poor educational climate (quality of school), and proportion of migrant/ethnic groups (Nieuwenhuis & Hooimeijer, 2016). Purely looking at the geographic attributes such as the number of houses in the neighborhood cannot determine the educational climate or the proportion of migrant groups at all. Whether those attributes could determine poverty is debatable. I argue that for example, a crowded neighborhood signal property (people have no choice but to live there) or higher opportunities available (near good school), which makes those variables confounding. Therefore, I argue that LiDAR images cannot determine the educational deprivation level.

VGG-16

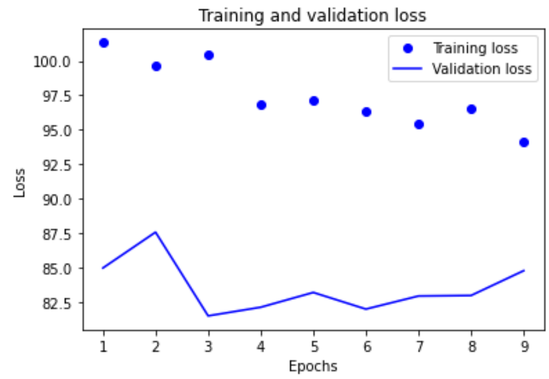
In order to test my hypothesis, neural network models were implemented as attempts to capture the relationship between those LiDAR images and education deprivation level. Specifically, two architectures – VGG-16 & ResNet50V2 – were implemented. This section focuses on the explanation of the first architecture: VGG-16

The general architecture of VGG-16 consists of 5 blocks with 2-3 convolutional layers with a max-pooling layer at the end of each block. For my problem, I decided to make the last two convolutional layers trainable while other layers would have the pre-trained imagenet weights. The rationale is that high-level feature extractions need to be tailored to solve particular problems while low-level feature extractions are similar for all images. For example, extracting low-level features such as lines, edges would be the same for recognizing a dog or a person while extracting high-level features such as the shape of noses would be different for recognizing a dog than for a person. In addition, on top of convolutional layers, I added two dense layers with 128 neurons each and 50% drop-out as further attempts to capture the relationship between the LiDAR image tensors and education deprivation level. The number of layers and neurons should be adjusted if any overfitting/underfitting was to occur. Drop-out was added to compress "learning" of neurons in different layers so that the number of parameters decrease to avoid overfitting. As a final step, a dense layer with 1 output and ReLU activation (since education index > 0) was added to produce an estimation of education deprivation level given LiDAR image as input.

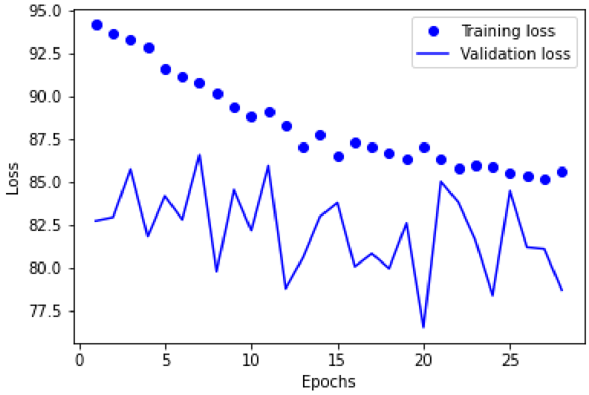
It should be noted that the dataset contains 36,723 LiDAR images which are divided into training set (75.5%) with 20% of images as validation, and test set (22.5%).

I tunneled the images through ImageDataGenerator function with batches of size 128 (the maximum amount that could be handled by my GPU). For the training data generator, I didn't shear the images since they are standard without any angles difference between images. I chose to zoom in or out by 20% which may not be necessary for the same reason as before. On the other hand, I included horizontal and vertical flip because direction of the image may not be consistent. For the testing data generator, I kept the original raw data. It should be noted that this process is identical when deploying ResNet50V2 architecture. I compiled both models with mean squared error as loss function.

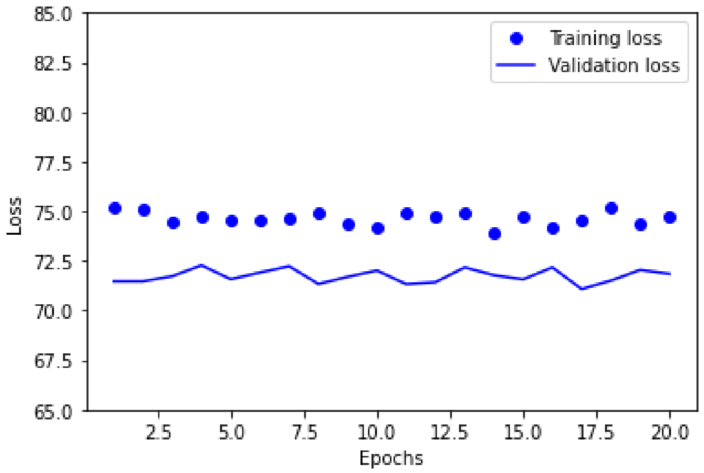
Initially, a learning rate of 10^{-5} and a learning rate scheduler which decrease the learning rate exponentially at the rate of $e^{-0.1}$ were used. The validation loss stabilized around 80 as indicated below.



Thus, I presumed that the loss function may be stuck in a local minimum, I raised the learning rate to 10^{-3} to in response. I immediately observed volatility in validation loss as indicated below



To stabilize the loss, I then decreased the learning rate and resumed training from the best of previous result. After I adjusted the learning rate to 10^{-4}, followed by $5 * 10^{-5}$ and 10^{-6}, I was able to observe convergence with validation loss ≈ 71 after 80 epochs of training. The convergence is shown below



The model chosen produced a prediction loss of 71.768 on the test set. Considering this is squared loss and the index ranged from 0.013 – 58.976, this model exceeded my expectation for reasons indicated in the first section.

ResNet

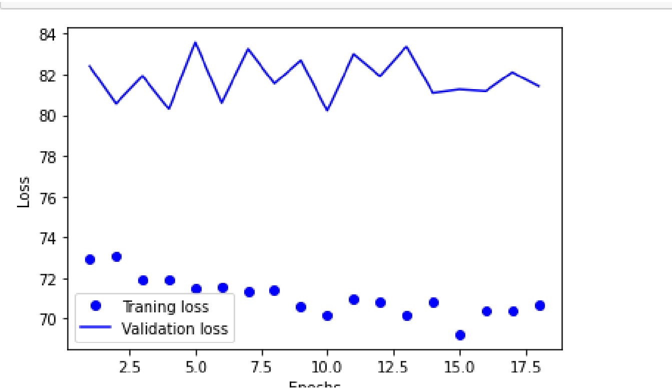
ResNet50V2 is a much more complex architecture than VGG16 where the biggest difference is that ResNet has skip connection. So instead of repeatedly performing convolution followed by another convolution and max pooling, ResNet preserves the original input by adding the input directly to the output of convolutional layers. This could prevent the loss of important information when max pooling is applied.

Since ResNet has skip connections and it's a functional object as opposed to a Sequential object such as VGG, the model needs to be defined differently while the logic remains the same. An initial model of ResNet was first downloaded from Google excluding the top layers since they have to be tailored to the problem at hand, followed by manually defined dense layers and final activation. Due to overfitting, I implemented 4 models of 2 dense layers with

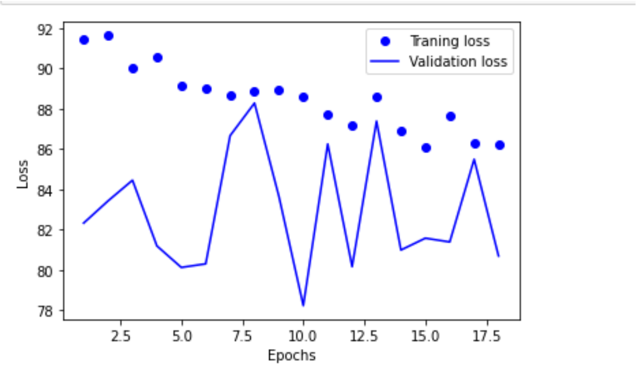
1) 128 neurons and 50% drop-out each
2) 64 neurons and 50% drop-out each
3) 64, 32 neurons with 50% drop-out each
4) 128 neurons with 75% drop-out and 64 neurons with 50% drop-out

A noticeable difference between training VGG model and ResNet model is that I have to reduce the batch size by 4 times because the ResNet model is considerable larger than VGG model (which has 2.3 times more parameters). In addition, for the same reason, the learning rate needs to be smaller so that the loss may converge.

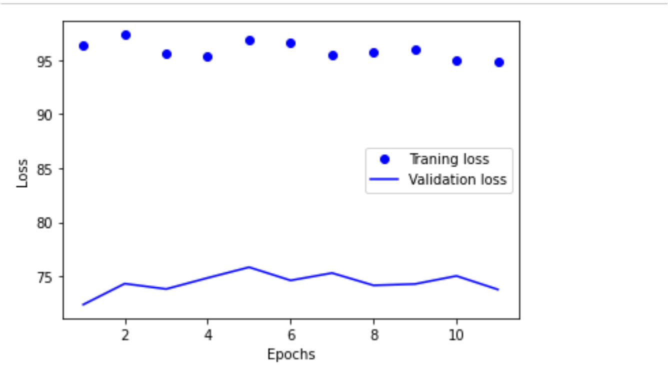
To improve the efficiency in training the whole model, the top layers were trained briefly with learning rate of 10^{-4} while the parameters in the initial model were set to untrainable. After training model 1) for 55 epochs (using learning rates $10^{-5}, 5 * 10^{-6}$ with learning rate scheduler) with the same logic as training VGG model, the validation loss seemed to stabilize around 81 as indicated below. The prediction error was 86.546, this model produced much worse results compared to my VGG model.



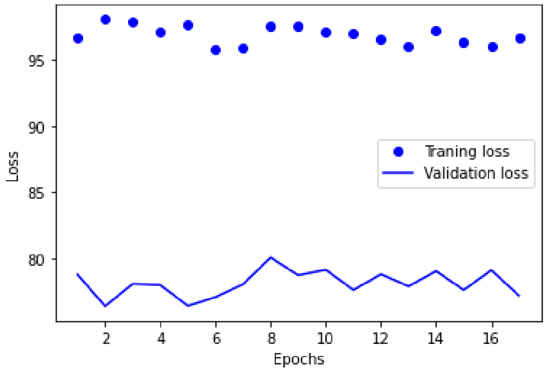
I suspected that the model is overfitting, which leads to my decision to reduce the neurons in each dense layer by half. After training model 2) for another 55 epochs with learning rates $3 * 10^{-6}, 10^{-6}$ with schedular, the model still seems to overfit as indicated below.



I trained the third model with second dense layer's neurons reduced to 32 for 64 epochs, and finally reached a convergence that was not too pessimistic. The final 11 epochs are as follows



I wonder if the loss can be reduced by including more highly compressed information in the dense layers, which led to my fourth model with 128 neurons (75% drop-out) and 64 neurons (50% drop-out). After 71 epochs of training, the result could not outperform my third model. The convergence is as follows



Comparison

The following table summarizes the results that I have obtained.

Model	Validation Loss	Test Loss
VGG	71.003	71.768
ResNet 1)	81.4239	86.546
ResNet 2)	N/A	N/A
ResNet 3)	72.110	74.703
ResNet 4)	77.2346	79.146

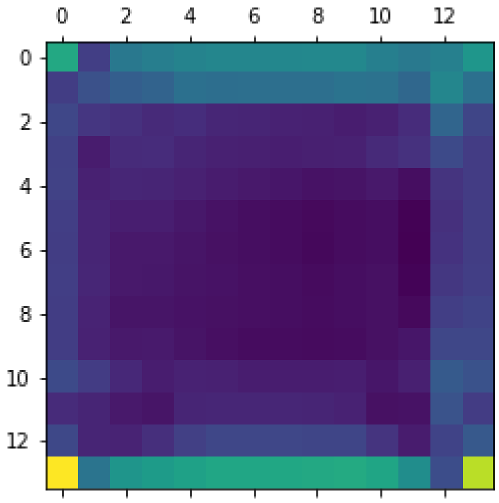
While ResNet should outperform VGG due to its highly sophisticated architecture, my VGG model showed better results. I considered two reasons behind this unexpected observation:

1) The LiDAR images do not possess complex high-level features, in which case a complex model would overfit the data; the logic is comparable to using a cubic model to fit linear data
2) I have not given enough time and effort in training ResNet models due to the time constraints

The following section would proceed with VGG-16 model as this produced the best result.

GradCAM

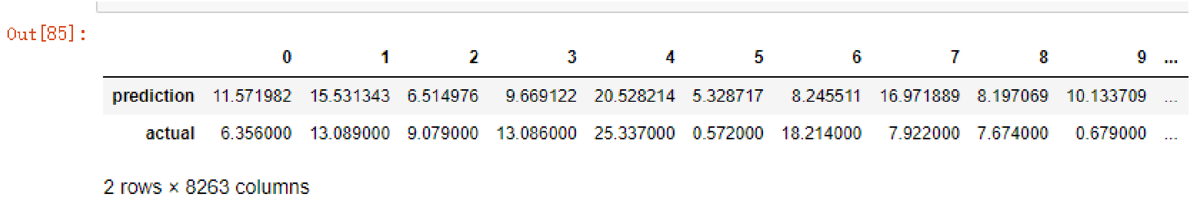
GradCAM is a tool that we could use to see which part of the picture activates the model to derive its prediction. The result of applying GradCAM on VGG-16 model was highly unexpected. I previously commented that the average prediction error of the model exceeded my expectation which led to my belief that there was some predictive power within the model. The result showed otherwise. After implementing GradCAM to 10 pictures with education deprivation indexes ranging from 0.3 to 40.486, it is surprising to observe almost identical heat maps as indicated below. This suggested that the bottom and top of the image trigger the model to make its final prediction regardless of what's in the image. Note that this is possible given 75% of indexes is in the range of 0-19.



Since evidently, those portions of the image cannot help us determine the education deprivation index and the model prediction is always around 7.4 for all cases, it's safe to assume that I have trained a constant model where every variation in the response variable (i.e. education deprivation index) is just random error.

The following conclusion can be drawn

1) The model cannot be used to effectively predict the education deprivation index from LiDAR images
2) There is unlikely to be any relationship between LiDAR images and the education deprivation index
3) Since my predictions for the test generator showed varied outputs, I suspect that there are some errors with the manual prediction function or the tensors. The output is as follows. Unfortunately, I could not figure out the cause of this inconsistency in a timely manner.



Out[85]:

	0	1	2	3	4	5	6	7	8	9	...
prediction	11.571982	15.531343	6.514976	9.669122	20.528214	5.328717	8.245511	16.971889	8.197069	10.133709	...
actual	6.356000	13.089000	9.079000	13.086000	25.337000	0.572000	18.214000	7.922000	7.674000	0.679000	...

2 rows × 8263 columns

Implications

Whether LIDAR images are useful in determining multidimensional deprivation requires further exploration. What can be concluded from this report is that educational deprivation level cannot be determined precisely from LiDAR images based on the two architectures implemented. The average error in estimation is 8.47 (i.e. $\sqrt{71.768}$), considering the range of the index is 58.963, this estimation error is significant.

In addition, I believe that there are considerable privacy and ethical issues associated with using LiDAR images to measure deprivation. Deprivation – by definition – means that material benefits of basic necessities were taken away from someone (not by their choice). This attributes more to someone's identity as opposed to their behavior. An important lesson learnt from credit scoring is that people should be "discriminated" using their behavior rather than identity because people can change their behaviors while not their identities. Consider a scenario where someone with perfectly good health who lives in a neighborhood with high health deprivation index. It is outrageous and unreasonable to hold that deprivation index against the person and deny them of credit or health insurance access. This kind of denial is wrong in principle, which raises ethical concerns regarding the usage of LiDAR images. Consider a second scenario where income deprivation indexes derived from LiDAR images were made publicly available. A child was discriminated against in elementary school because he/she comes from a neighborhood of high income deprivation index. The child was judged and unfairly treated because of his/her private affairs (i.e. family status). This is a textbook-definition of privacy violation (US Legal, n.d.) with significant ethical implications.

Based on the scenarios suggested above, I offer the following proposals as attempts to overcome the potential privacy and ethical concerns:

1) The usage of LiDAR images to measure multidimensional deprivation should be carefully regulated and should never be made publicly available.
2) In each area where the deprivation index might be applied, experts should carefully justify to the regulators why it is necessary to do so subject to the local laws and government regulations.

Recommendation

Following the conclusion of this report, the following recommendations were made to improve the current results

1) Investigate the cause of the discrepancy between manual prediction and test generator prediction
2) Set more layers in the VGG model to be trainable to see if more information about the education deprivation index can be captured by the image
3) Set the learning rate of ResNet 4) model higher to see if the model improves
4) Explore the relationship between LiDAR images and other indexes

Reference

Claudia Mihovk, Ubald, Jake, Trish, GISGeography, Susan M. Grady, Therese Reinsch, Larry Schwartz, Nyoome, Ziad, Samantha, Oskar, Stanley Mugenyi, Mahfuzul Islam, C Hunt, Victor Abbott, Layla, Raxhel M, & Rijo George. (2021, October 29). *A complete guide to LIDAR: Light detection and ranging*. GIS Geography. Retrieved December 16, 2021, from https://gisgeography.com/lidar-light-detection-and-ranging/

Nieuwenhuis, J., & Hooimeijer, P. (2016). *The association between neighbourhoods and educational achievement, a systematic review and meta-analysis*. Journal of housing and the built environment : HBE. Retrieved December 16, 2021, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5748572/

US Legal, I. (n.d.). *What Constitutes a Violation*. Privacy. Retrieved December 17, 2021, from https://privacy.uslegal.com/what-constitutes-a-violation/

Appendix I: Code

Link to my code
https://colab.research.google.com/drive/1obeOECCExaL55-yGCY4GqDlLBtALViJW?usp=sharing