Predicting People’s Perceived Street Safety with Convolutional Neural Networks

Lingling Zhang  
*School of Continuing Studies  
 York University*Toronto, Canada

Durai Nachiappan  
*School of Continuing Studies  
 York University*Toronto, Canada

***Abstract*—In this project, we created Convolutional Neural Networks that can classify people’s perceived safety towards street views into 2 classes: More Safe and Less Safe. Ensemble learning techniques were attempted, and applied to the predictions of people’s perceived safety on Toronto street view images. We further adapted a model interpretation tool - LIME to better understanding of our models’ predictions.**

***Keywords—Perceived Safety, Street View, Computer Vision, Deep Learning, Convolutional Neural Networks, Transfer Learning, Ensemble Learning, Model Interpretation***

# Introduction

Understanding people’s perceived safety of street view could help us to infer the city scene, provide residents with additional information about where they live, and assist city planners to build a more friendly and equal living environment for the residents. The appearance of neighborhoods might impact the health and behavior of the individuals. To understand a city’s built environment that is relevant for the experiences of citizens. It would also be useful for scholars to examine the social and economic consequences of urban perception.

To achieve this, some prior research has been made. In 2013, the Place Pulse project [1] measured people’s perception of safety, class, and uniqueness in four cities (Boston and New York in the United States, and Linz and Salzburg in Austria). The research found that the range of perception elicited by the images of the 2 US cities is larger than that of in the 2 Austria cities. The authors interpreted this as the evidence that the cityscapes of the 2 US cities are more unequal than the 2 Austria cities. This research showed that online images can be used to create reproducible quantitative measures of urban perception and characterize the inequality of different cities. The authors argued that the cities are heterogeneous and often unequal with respect to not only the income of the residents but also with respect to the cleanliness of the neighborhoods, the beauty of the architecture and the liveliness of the streets, among many other evaluative dimensions. The users were shown two images, selected randomly from the dataset and asked to lick on one in response to one of the 3 questions: Which place looks safer? Which place looks more upper-class? or Which place looks more unique? It used 4136 geo-tagged images, among them 2942 images in the 2 US cities were obtained from Google Street View and images from the 2 Austria cities were collected manually on site. It is a crowdsourced study. 7,872 Participants were shown images in an online game using 208,738 pairwise comparisons.

The Broken Windows Theory [2] suggests that evidence of environmental disorder, such as broken windows, litter, and graffiti, can induce other kinds of disorder, like crime and hence, policies that focus on the amelioration of minor offenses can help fight more severe forms of criminal activity. Other research some were in favor of this theory and some were against this theory.

The paper argued that besides crime and disorder, many other dimensions, such as whether the place look lively, modern, inspiring, clasy, abandoned, congested, colorful or beautiful, can be used to explore connections between aspects of urban perceptions and other social dimensions, such as entrepreneurship, civic engagement and high-school completion, among other things.

[3] created a computational model for perceived safety based only on features. It used support vector regression on commonly used features extracted from the images, such as geometric texton, color histograms, and GIST. The project used the predicted data to create high-resolution maps of perceived safety for 21 cities in the United States, scoring around 1 million images from Google Street View. The research show that the predictor can be used on images and cities for which evaluative data was available. This research converted the Q-score in [1] to a ranked score for each image using the Microsoft Trueskill algorithm [4]. This research argued that an algorithm trained with a few images from a given area can be used to map a significant area of its surroundings, which is useful to know for future crowdsourced studies. The project used the predicted Streetscore to create high-resolution maps of urban perception.

With new developments in deep learning methodologies on computer vision, Convolutional neural network (CNN) is the most commonly applied and state-of-the-art technology in image recognition. We believe we can further the research of [3] with CNN. We would also specifically like to apply such a model on Toronto street view images so as to help the City of Toronto better understand people’s perceived safety in the city.

The authors of [3] also pointed out that future works can focus on identifying the objects and features that help explain the evaluative dimensions of a streetscape. Therefore, we also worked on model interpretation using the LIME package. This would further help the scholars, city planners to understand what types of elements in the living environment would be considered as safer by the residents so as for them to better improve our living environment.

To achieve our research objectives, we made use of the predicted data open-sourced by [3].

We created 2 groups of deep learning classification models - one was generated from transfer learning using pre-trained weights of ResNeXt50 developed by [5]; another one was generated from our self-designed CNN. We used averaging and weighted ensembling methods to produce the final prediction.

Models were trained to predict the More Safe / Less Safe labels. Our models were finally applied to predict the perceived safety for Toronto street view images. We generated the interpretations and explanations for our models using the LIME package.

Residual Learning Framework (ResNet) is a pre-trained deep convolutional neural network. It has been proved to have excellent results in not only image classification, but also image detection, image localization and segmentation in ILSVRC 2015 and COCO 2015 competitions.

ResNet is proved to be able to overcome the optimization difficulty of degradation when the convolutional neural network becomes deeper. Residual learning can effectively reduce the error rate on extremely deep systems. It can also ease the optimization by providing faster convergence at the early stage. Being deep and thin by design, deep residual nets adapt parameter-free identity shortcuts so as to control the complexity of the architecture.

Locally Interpretable Model-Agnostic Explanations (LIME) was used to interpret and explain the predictions of our final model. LIME is a technique that explains the predictions of any machine learning classifier or regressor in an interpretable and faithful way, by learning an interpretable model locally around the prediction. It can also explain models by presenting representative individual predictions and their explanations in a non-redundant way, framing the task as a submodular optimization problem.

LIME can be used to: decide if one should trust a prediction; choose between models; improve untrustworthy classifiers; and identify why a classifier should not be trusted. When explaining a prediction, LIME presents textual or visual artifacts that provide qualitative understanding of the relationship between the instance's components and the model's prediction.

# Methodology

## Data Collection

**Proportion of Toronto Zoning Data** (Table I) was the percentage for each Zone class in Toronto. It was provided by the City of Toronto. As we aim to build models that also predict well on Toronto images, when we choose our Training and Test set, we considered this proportion.

## **Boston and New York Data with Predicted Q-score.** There were 2 datasets - streetscore\_boston.csv and streetscore\_newyorkcity.csv open-sourced by the Streetscore project available at [6]. The two datasets both have 3 columns: latitude, longitude and q-score.

**Boston Zoning data.** We obtained a file *Zoning\_Subdistricts.shp* from City of Boston's open data hub [7]. It contained a ‘geometry’ column which is geometric polygon data as well as a ‘Zone\_Desc’ column, which was relevant to the zoning class in Table 1.

**Toronto data.** We obtained file *ADDRESS\_POINT\_WGS84.shp* from the City of Toronto’s Open Data Catalogue [8]. It contained 525,545 geolocation points in Toronto, but no zoning class. We randomly sampled 2,100 Toronto data to fetch Toronto images. We would use our final ensemble strategy to predict on those Toronto images.

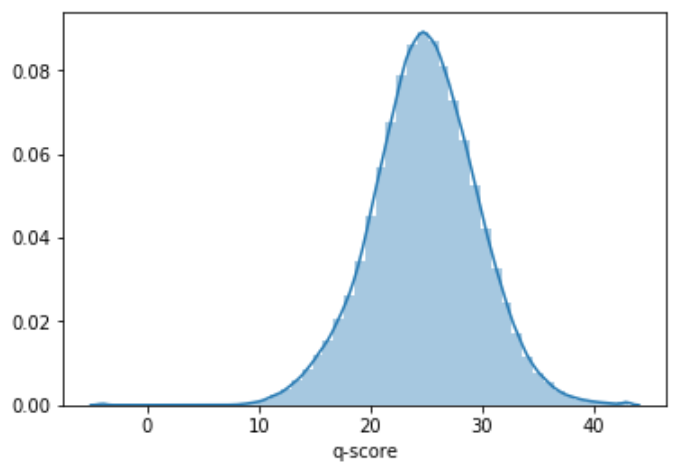
**Street View Images.** We fetched the images through Google Street View Static API [9] based on the geolocation information (latitude and longitude) provided as in the data mentioned above, and the google-streetview package [10]. Note that the API provides several parameters with which users can define while fetching images. However, it is not clear what parameter settings were adopted by the MIT Streetscore project. Therefore, we decided that it is better to set some parameters (fov, heading, pitch, radius) as default.

1. Proportion of Toronto Zoning Class

|  |  |
| --- | --- |
| **Zone Class** | **Percent (%)** |
|
| residential | 49.89 |
| open space | 17.86 |
| employment industrial | 13.9 |
| unassigned | 5.83 |
| commercial | 5.34 |
| utilities | 5.08 |
| institutional | 2.1 |

* 1. *Data Exploration and Preparation*
     + 1. *Creating Target Column “safety”:*

In the MIT Streetscore data of New York City and Boston, perceived safety scores (q-score) is continuous, ranging from -4.0 to 43.0 (with the mean of 24.86 and the median of 24.88). Note that this contradicts with [3] stated, which said the q-score ranged from 0 to 10. The distribution of the q-score is shown in Fig 1.



1. Distribution of the q-score

To create a target binary column “safety”, we first combined all samples in the 2 cities and then separated the samples into 2 bins (relevant to cutting from the median) based on the distribution of the q-scores. For samples in bin 1, we defined their safety as 0, which represents Less Safe; for samples in bin 2, we defined their safety as 0, which represents More Safe. Finally, we saved the two cities’ data into separate data tables: boston\_safety.csv and ny\_safety.csv.

#### Creating a subdistrict variable for the Boston data:

The objective of this procedure is to make sure we can subsample Boston data which are representative to Toronto Zoning class in the next procedure (Stratified Downsampling). Among the 2 cities, only for Boston, we have available zone data comparable to the zone classes provided in Table I. Therefore, we decided to use only Boston data in order to meet the stratified sampling requirement.

For *streetscore\_boston.csv,* we used the Shapely package [11] created a column called ‘Coordinates’, which was the geometric point generated based on the original ‘latitude’ and ‘longitude’ columns. Then we used the GeoPandas package [12] to spatial join with the Zoning\_Subdistricts.shp which has a ‘geometry’ column with geometric polygons.

Note that after the spatial join, only 23.26% (53,401 out of the 229,564) of the original Boston Data samples fell into the zones provided by Zoning\_Subdistricts.shp and so they were left for stratified subsampling.

#### Stratified Downsampling of Boston Data:

After the ‘safety’ and ‘subdistrict’ columns were generated, we performed stratified downsampling based on both of them.

**Stratified sampling based on Zone classes of Toronto**

Note that the subdistrict classes of Boston Data (Table II’s ‘subdistrict’ column) and the Zone classes of Toronto provided in Table I did not exactly match. Therefore, We considered “Miscellaneous” + “Mixed Use” of Boston subdistrict as one class, and when subsampling, we would make it stratified of the proportion of “unassigned” + “utilities” in Toronto.

1. Count and Proportion of 53,401 Sample’s Boston Subdistrict Class

|  |  |  |
| --- | --- | --- |
| **Subdistrict** | **Count** | **Percent (%)** |
|
| Residential | 33749 | 63.2 |
| Open space | 5273 | 9.87 |
| Business | 4650 | 8.71 |
| Industrial | 3452 | 6.46 |
| Mixed Use | 2973 | 5.57 |
| Miscellaneous | 1890 | 3.54 |
| Comm/Instit | 1414 | 2.65 |

As we were to select 20,000 Boston samples, we calculated for each subdistrict, how many samples we should select based on Table I, and listed it in Table III.

1. The 20,000 Boston Samples We Aimed to Select by Subdistrict

|  |  |
| --- | --- |
| **Subdistrict** | **Samples should select** |
|
| Residential | 9978 |
| Open space | 3571 |
| Industrial | 2780 |
| Mixed Use and Miscellaneous | 2182 |
| Business | 1068 |
| Comm/Instit | 420 |

**Stratified sampling based on target column ‘safety’**

While we subsample Boston data based on Table III, we also made sure that for each subdistrict in Boston, the proportion of ‘safety = 0’ in the target column is equal to that of ‘safety = 1’ as much as possible.

Note that not all subdistrict classes meet the desired quantity requirement. In subdistrict “Industrial”, there were only 505 samples which have “safety = 1”, less than the ideal 1390 “safety = 1” samples that we would like to sample. So for ‘Industrial’, more samples with safety = 0 were selected. Table IV shows the subsampled 20,000 Boston data, count by both ‘subdistrict’ and ‘safety’.

1. Subsampled 20,000 Boston Data - Count by Subdistrict and Safety

|  |  |  |
| --- | --- | --- |
| **Subdistrict** | **Safety** | **Number of Samples** |
|
| residential | 0 | 4989 |
| residential | 1 | 4989 |
| open space | 0 | 1786 |
| open space | 1 | 1786 |
| business | 0 | 534 |
| business | 1 | 534 |
| Industrial | 0 | 2275 |
| Industrial | 1 | 505 |
| Mixed Use | 0 | 663 |
| Mixed Use | 1 | 652 |
| Miscellaneous | 0 | 428 |
| Miscellaneous | 1 | 439 |
| Comm/Instit | 0 | 210 |
| Comm/Instit | 1 | 210 |

#### Train / Test split:

For the downsampled 20,000 Boston samples, we split them into 80 : 20 as Boston Train and Boston Test sets.

#### Image Fetching and Pre-Processing:

**Image Fetching.** Note Not all geolocation in our datasets had an image can be fetched. For the 16,000 Boston Train Data, we were able to fetch 15,928 Boston Train Images. For the 4,000 Boston Test Data, we fetched 3,976 Boston Test Images. For the 2,100 Toronto Data, fetched 2,034 Toronto Images. Fig 2 shows some sample Boston street view images labeled as More Safe and Less Safe.

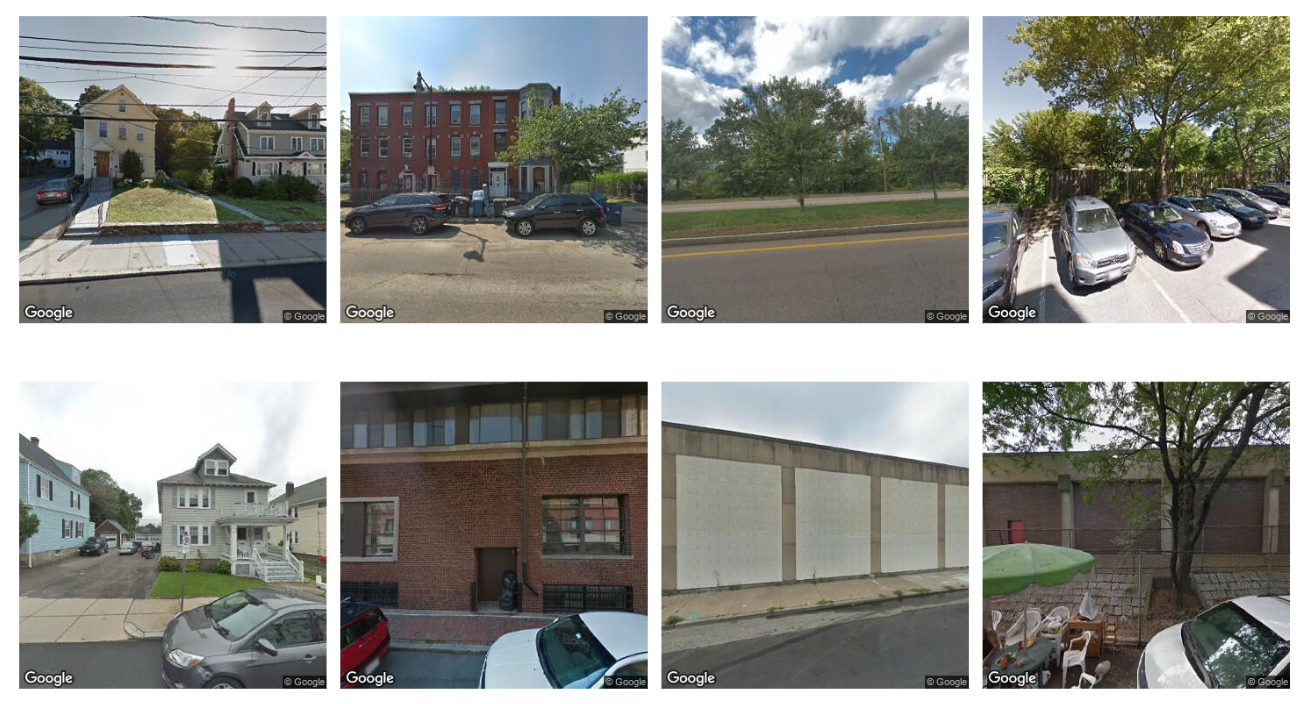
It is notable that Google periodically updates street view images, which means the street view images we fetched and then used to build and test models were mostly updated from the images used by the Streetscore project to arrive at the q-score. Table V lists the updated year of images we fetched for the Boston Train and Test Images, according to the metadata of the fetched images provided by the Google API.

**Image Pre-Processing.** Each downloaded street view image has Google’s logo. We cropped out a small edge from the bottom and from the right so as to remove the Google logo in each image.

As we fit images into CNN, images were resized to 224\*224 pixels (in transfer learning) and 100\*100 (in our own CNN) pixels.

1. Boston Train and Test Images’ Update Year

|  |  |  |
| --- | --- | --- |
| **Year Updated** | **Boston Train Image** | **Boston Test Image** |
|
| NaN | 11 | 2 |
| 2007 | 130 | 38 |
| 2008 | 6 | 2 |
| 2009 | 79 | 18 |
| 2010 | 26 | 6 |
| 2011 | 706 | 201 |
| 2012 | 23 | 6 |
| 2013 | 1853 | 440 |
| 2014 | 428 | 83 |
| Sum NaN, 2007 to 2014  (Percent out of Total) | 3262  (20.5%) | 796  (20.0%) |
| 2015 | 243 | 82 |
| 2016 | 660 | 174 |
| 2017 | 1540 | 387 |
| 2018 | 10200 | 2534 |
| 2019 | 23 | 3 |
| Sum 2015 to 2019  (Percent out of Total) | 12666  (79.5%) | 3180  (80.0%) |
| Total | 15928 | 3976 |

****

1. Sample Boston street view images. The above 4 images were with label ‘more safe’ and the bottom 4 images were with label ‘less safe’.
   1. *Model Building*

### Our own CNN model

We created 2 CNN models. The first model was trained and validated on Boston Train Images and tested on Boston Test Images. This model was used to experiment with the best ensembling strategy.

The second model was trained on both the Boston Train and Test Images. This model was used to investigate whether increasing training data could boost model performance. When we ensemble the predictions on Toronto Images, we used the prediction of this model.

**Pre-processing:** As part of pre-processing each training image was converted to a matrix of 100\*100\*3. We applied one-hot encoding to the target variable to improve efficiency.

**CNN layers:** CNN models were built with 2 convolutional layers, each followed by a pooling layer. The output from these layers were passed to a flatten layer and were fed to a couple of dense layers. After the initial set of runs, we realized that the model was over-fitting. Therefore, we added 2 dropout layers.

Dropout is a technique used to prevent overfitting and to approximately combine exponentially many different neural network architectures effectively. Dropout refers to temporarily removing a unit from the neural network along with its incoming and outgoing connections. As a result of Dropouts, hidden units trained with Dropout learn to work with a randomly chosen sample of other units. This makes the hidden units more robust and they do not rely on other units to correct its mistake.

After multiple runs, we found that Adam optimizer gave us the best results with default values for learning rate, decay and momentum. For the first model, we ran 20 epochs and for the second model, we ran 25 epochs.

### Transfer Learning

We chose ResNeXt50 as our pre-trained models in transfer learning. We chose this pre-trained model as it won the second place in image classification in the 2016 ImageNet Large Scale Visual Recognition Challenge (LSVRC) competition, at the same time, it has a relatively small amount of tunable parameters which allows us to go with smaller GPU memory.

We used Adam as optimizer, andtrained the models with 50 epochs and a starting learning rate of 0.0001.

Image augmentationon the training set was performed randomly on the fly during the model training process. For one particular training image, 1 augmented image was generated.

Five-fold cross-validation was implemented in the training process. For each cross-validation fold, the model with the best performance among all 50 epochs was saved. We could use the 5 submodels in ensembling. After cross-validation, the pre-trained networks were used to train models on the full training data set.

Therefore, a total of 6 models were generated (5 submodels from cross-validation + 1 model trained on full data set) in the transfer learning process.

* 1. *Ensembling*

We implemented the following ensembling methods:

**Averaging Predictions on Augmented Test Images.** For the 6 models produced in transfer learning, when generating predictions on the test images, for each of the test images, 5 random augmented images were generated. Those 5 predictions were given the same weight and averaged out to serve as the prediction of this particular image.

To find the best ensemble strategy, we further used 7 models (the first model produced by our own CNN structure and the 6 models produced in transfer learning) to predict on the Boston Test Images.

**Averaging Ensemble.** We tried 4 averaging ensemble strategies on the 7 models: averaging prediction of all single models; averaging predictions of models with best accuracy; averaging predictions of models best at predicting safety = 0 and safety = 1 respectively.

**Averaging + Conditional Ensemble.** We tried 2 conditional ensemble strategies: the first one is when predicting safety = 0, we use the prediction by predictions best at predicting safety = 0 produced by averaging ensemble, else we use the prediction by predictions best at predicting safety = 1 produced by averaging ensemble; the second one is when predicting safety = 1, we use the prediction by predictions best at predicting safety = 1 produced by averaging ensemble, else we use the prediction by predictions best at predicting safety = 0 produced by averaging ensemble;

**Averaging + Weighted Ensemble.** We tried 3 weighted ensemble strategies with different weight pairs to averaged predictions best at predicting safety = 0 and safety = 1 respectively.

**Final Ensemble Prediction Strategy.** We would compare the performance of ensembled predictions and pick one strategy that has both good overall accuracy and is good at predicting of both targets.

* 1. *Model Evaluation*

For each of single model prediction and ensembled predictions on the Boston Test Images, we calculated the confusion matrix to navigate us to the next ensembling strategy and assist us to judge which should be our final ensemble prediction strategy.

* 1. *Predictions on Toronto Images*

We applied the final ensemble prediction strategy for the prediction of the 2,034 Toronto images.

* 1. *Model Interpretation*

At last, we applied LIME on a number of randomly chosen Toronto street view images predicted by our models and conducted qualitative analysis on the results.

# Results

## Performance of 7 Single Models on Boston Images

The loss and accuracy of single models produced by our own CNN model and transfer learning are shown in Table VI. Most of our models experienced overfitting, except for the one that produced by our own CNN model on both Boston Train and Test Images. Our single models’ validation accuracy was all around 70%.

For the 2 models produced by our own CNN structure, the model trained on both Boston Train and Test Images had better performance and less overfitting comparing the model trained by the same CNN structure on Boston Train Images only.

* 1. *Confusion Matrix and Performance of Single and Ensembled Models on Boston Test Image.*

The confusion matrix and performance of 7 single models and all ensembled models on Boston Test Images are shown in Table IV.

**Single Models.** Among the 7 single models, 2 models - cv3 and full were better at predicting safety = 0; the other models - cv0, cv1, cv2 cv4, own were better at predicting safety = 1. For the 6 models produced by transfer learning, their accuracy on the Boston Test Images was better than the validation accuracy on Boston Train Images.

**Averaging Ensemble.** Averaged predictions had better accuracy than single models.

Overall, ensembling of ‘all models’ yielded the best accuracy (73.9%). However, this ensembled model is neither best when predicting safety = 0 along nor best at predicting safety = 1 alone.

Model ‘cv3full’ was best at predicting safety = 0 (TN\_Rate 0.756619) and model ‘cv0cv1cv2cv4own’ was best at predicting safety = 1 (TP\_Rate 0.779363), but they were weak at predicting the other target respectively. We would use these 2 models in conditional and weighted ensembling.

**Averaging + Conditional Ensemble.** We tried 2 conditional ensembling:

‘Conditional ensemble1’ - When predicting safety = 0, we use the prediction by ‘cv3full’; else we use the prediction by ‘cv0cv1cv2cv4own’.

Conditional ensemble2: - When predicting safety = 1, we use the prediction by ‘cv0cv1cv2cv4own’; else we use the prediction by ‘cv3full’.

Confusion Matrix showed that show that ‘conditional ensemble1’ model was good at prediction target 0, at the expense of prediction 1. On the contrary, ‘conditional ensemble2’ model was good at prediction target 1, at the expense of prediction 0. Therefore we further tried weighted ensemble.

**Averaging + Weighted Ensemble.** We tried 3 weighted ensembling with different weight pairs to model ‘cv3full’ and model ‘cv0cv1cv2cv4own’, as detailed in Table VIII.

Comparing results throughout Table VIII, we found that the ‘weighted ensemble2’ had an accuracy of 74.1%, and was good at predicting both safety = 0 and safety = 1. Therefore, we would consider the method used in ‘weighted ensemble2’ as our final ensembling strategy.

* 1. *Predictions on Toronto Street Views*

We applied the ensemble strategy the same way as ‘weighted ensemble2’ on the 2,034 Toronto images. As a result, 1,031 of the images were predicted as 0 (less safe), and 1,003 were predicted as 1 (More Safe). Fig 4 is a map visualization of the result.

* 1. *LIME*

In order to trust a model, it is important to understand the underlying mechanics behind the decisions made by the model. We decided to use LIME to generate Model interpretations/explanations as it was well suited for image data and can generate a local interpretation for a particular image, which will in turn help us to understand all the factors that contribute towards or against a prediction.

LIME is model-agnostic, meaning that it can be applied to any machine learning model. The technique attempts to understand the model by perturbing the input of data samples and understanding how the predictions change. LIME for images works differently than LIME for tabular or text data. It creates variations of the image by segmenting image into superpixels and turning superpixels off or on. Superpixels are interconnected pixels with similar colors.

Our qualitative analysis on LIME’s interpretation of our models’ predictions on some randomly chosen Toronto street view images yielded useful insights. We found that plants, trees, lawn, newer looking and brightly colored houses seemed to heavily contribute towards a More Safe perception. On the contrary, electrical wires hanging over houses, dull colored and older houses, shadows, and crack on the ground contributed towards Less Safe perception.

Fig.3 shows 4 images and their LIME interpretations. Among them, image (a) and image (b) were predicted as More Safe by our models; image (c) and image (d) were predicted as Less Safe by our models. LIME highlighted the parts of images that contributed towards a prediction in green color (pros) and the parts of the image that contributed against a prediction in red color (cons).

In all of the 4 images, the trees contributed towards a More Safe perception, whereas dull colored and older houses, as well as shadows, contributed towards a Less Safe perception. In image (a), crack on the ground at the right corner of the image contributed to the Less Safe prediction. In image (c), wires running overhead the house made the model to predict the image as Less Safe.



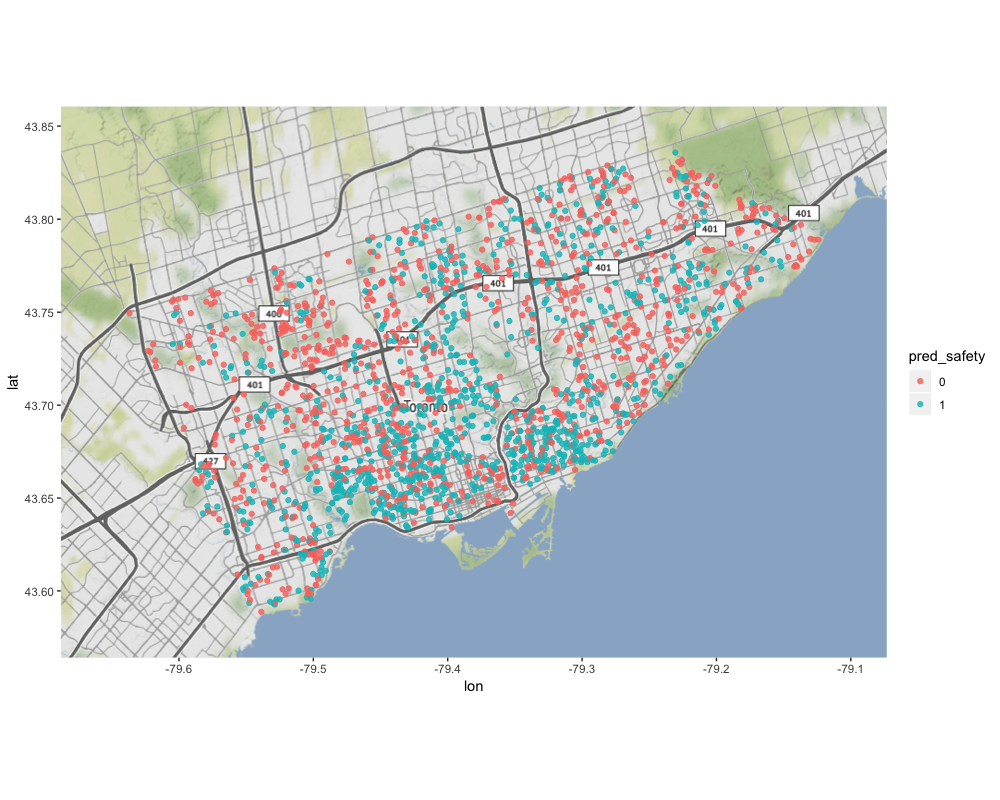
1. Examples of model interpretation by LIME.

1. Performance of Single Models trained from Transfer Learning and Our Own CNN on Boston Images

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Transfer Learning from ResNeXt50** | | | | | | | | **Own CNN** | |
| Training data | CV0  of  Training | CV1  of  Training | CV2  of  Training | CV3  of  Training | CV4  of  Training | mean  of  5 CVs | std  of  5 CVs | Full  Training | Full Training | Full Training + Test |
|
| train\_loss | 0.167847 | 0.070979 | 0.085303 | 0.051624 | 0.052649 | 0.085680 | 0.042939 | **0.058055** | **0.3158** | 0.4645 |
| train\_acc | 0.947100 | 0.972844 | 0.970884 | 0.980929 | 0.979752 | 0.970302 | 0.012226 | **0.978464** | **0.8696** | 0.7783 |
| val-loss | 1.802394 | 1.540489 | 1.852702 | 1.808886 | 1.562203 | **1.713335** | 0.133567 | - | **0.6401** | 0.5703 |
| val-acc | 0.669805 | 0.672003 | 0.691994 | 0.672214 | 0.692622 | **0.679727** | 0.010308 | - | **0.6800** | 0.7270 |
| Name in Ensemble | CV0 | CV1 | CV2 | CV3 | CV4 | - | - | Full | Own | - |

1. Confusion Matrix and Performance of Models on Boston Test Images

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ensemble Type** | **Ensemble Strategy** | **Model Name** | **FN** | **FP** | **TN** | **TP** | **TN\_Rate** | **TP\_Rate (Recall)** | **Accuracy** | **F1 score** | **Precision** |
| Single Model | NA | cv0 | 485 | 704 | 1449 | 1337 | 0.673014 | 0.733809 | 0.700881 | 0.692208 | 0.655071 |
| cv1 | 442 | 775 | 1378 | 1380 | 0.640037 | 0.757409 | 0.693836 | 0.693990 | 0.640371 |
| cv2 | 537 | 649 | 1504 | 1285 | 0.698560 | 0.705269 | 0.701635 | 0.684239 | 0.664426 |
| cv3 | 653 | 558 | 1595 | 1169 | 0.740827 | 0.641603 | 0.695346 | 0.658777 | 0.676896 |
| cv4 | 505 | 684 | 1469 | 1317 | 0.682304 | 0.722832 | 0.700881 | 0.688988 | 0.658171 |
| full | 579 | 536 | 1617 | 1243 | 0.751045 | 0.682217 | 0.719497 | 0.690364 | 0.698707 |
| own | 552 | 742 | 1412 | 1270 | 0.655829 | 0.697036 | 0.674717 | 0.662666 | 0.631527 |
| Averaging Ensemble | All Models | all models | 435 | 602 | 1551 | 1387 | 0.720390 | 0.761251 | 0.739119 | 0.727893 | 0.697335 |
| Models with Best Accuracy | cv0cv2cv4full | 462 | 620 | 1533 | 1360 | 0.712030 | 0.746432 | 0.727799 | 0.715413 | 0.686869 |
| **Models Best at Predicting Safety = 0** | **cv3full** | 576 | 524 | 1629 | 1246 | **0.756619** | 0.683864 | 0.723270 | 0.693764 | 0.703955 |
| **Models Best at Predicting Safety = 1** | **cv0cv1cv2cv4own** | 402 | 664 | 1489 | 1420 | 0.691593 | **0.779363** | 0.731824 | 0.727087 | 0.681382 |
| Averaging +  Conditional Ensemble | For safety = 0, use cv3full; else use cv0cv1cv2cv4own | conditional ensemble1 | 650 | 430 | 1723 | 1172 | 0.800279 | 0.643249 | 0.728302 | 0.684579 | 0.731586 |
| For safety = 1, use cv0cv1cv2cv4own; else use cv3full | conditional ensemble2 | 328 | 758 | 1395 | 1494 | 0.647933 | 0.819978 | 0.726792 | 0.733432 | 0.663410 |
| Averaging  +  Weighted Ensemble | cv3full \* 0.4; cv0cv1cv2cv4own \* 0.6 | weighted ensemble1 | 455 | 574 | 1579 | 1367 | 0.733395 | 0.750274 | 0.741132 | 0.726548 | 0.704276 |
| **cv3full \* 0.45; cv0cv1cv2cv4own \* 0.55** | **weighted ensemble2** | 460 | 570 | 1583 | 1362 | **0.735253** | **0.747530** | **0.740881** | 0.725626 | 0.704969 |
| cv3full \* 0.55; cv0cv1cv2cv4own \* 0.45 | **weighted ensemble3** | 490 | 563 | 1590 | 1332 | 0.738504 | 0.731065 | 0.735094 | 0.716707 | 0.702902 |



1. Map of our models' final predictions of perceived safety on street view of 2,034 Toronto geolocations.

# Discussions

* 1. *Transfer Learning and Our Own CNN*

ResNeXt50 performed marginally better than own CNN, at the expense of much higher computational resources. In [13], the author of ResNet and ResNeXt challenged the conventional wisdom of Imagenet pre-training for dependent task. Their research found that in in object detection and instance segmentation tasks on *COCO* dataset, standard models trained from random initialization performed no worse on ImageNet pre-training. Our research further demonstrated that in image classification, standard models also performed no worse on ImageNet pre-training.

* 1. *Ensemble Learning*

**Averaging Predictions on Augmented Test Images.** For the 6 models produced by transfer learning, their accuracy on the Boston Test Images (Table VIII) was better than the validation accuracy on Boston Train Images (Table VIII). This proved that generating more augmented images on the test set and then average out the prediction could boost model accuracy.

**Ensembling.** Our research showed the effectiveness of ensembling: the accuracy of all ensembling models was better than that of single models. Our final ensemble strategy boosted the prediction accuracy to 74.1% from ~70% accuracy by single models.Confusion matrix of each ensembled models proved to be an effective tool in guiding us to find the best ensembling strategy.

* 1. *Model Performance*

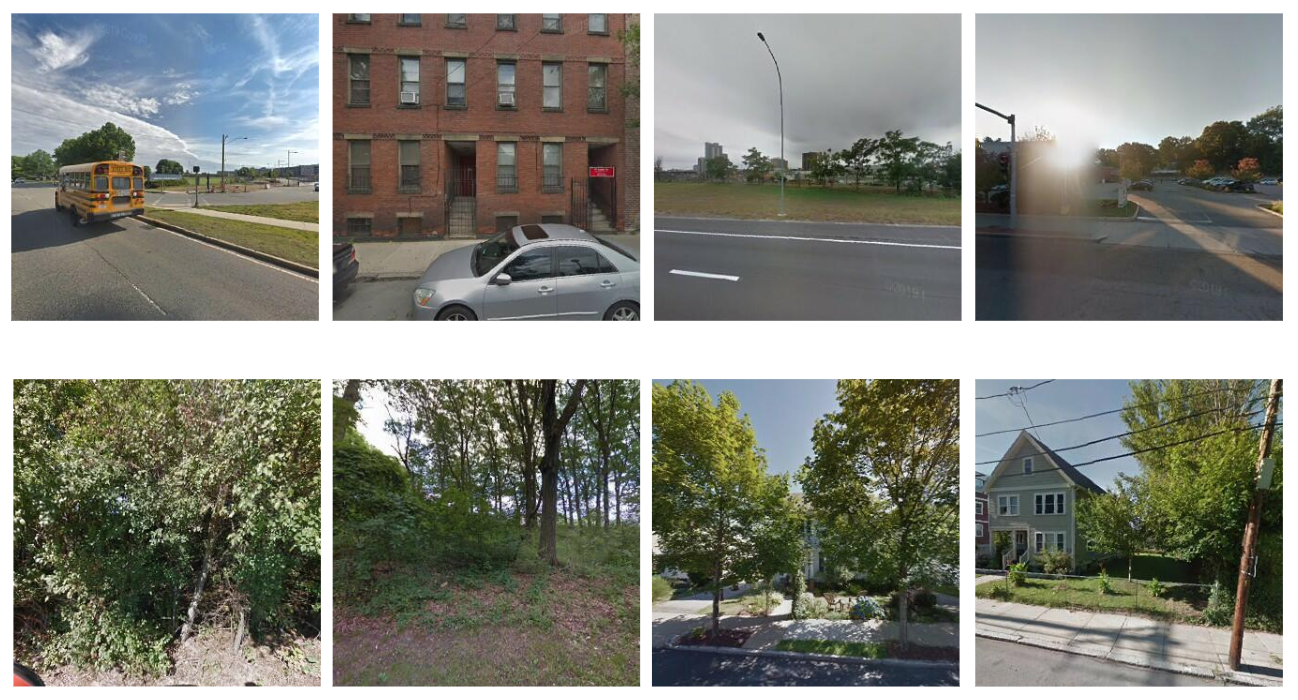
Despite the effectiveness of ensembling, our single models’ performance was not satisfactory. We assume there may be several reasons:

First, most images have changed. Streetscore project [3] was published in 2014, and according to Table V, only ~20% of the Boston images we fetched were updated by Google before year 2014. This indicates ~80% of the images we used for training and testing are not the same images used by the Streetscore project, which produced the q-score. As city scenery changes rapidly, we argue that the images for the same locations might have changed a lot during the past 4 years. Also it is not clear what parameters the Streetscore project has used when fetching images using the Google API. For a same location, the different parameter settings would get different images.

Second, the number of training images was not large enough. We used only 15,928 Boston Train Images, for the submodels saved during cross-validation, the training images are even fewer (~12,700). The fact that when applying our own CNN structure on both the Boston Train images and Boston Test images, the model accuracy (0.7270) was much higher than that of the model trained on Boston Train Images alone (0.6800) proved our this argument.

Fig 5 shows some of our extremely wrongly predicted Boston Test Images by our final prediction. The upper 4 images were predicted over 0.993 probability of being safety = 0, but the real label should be safety = 1. The below images were predicted over 0.991 probability of being safety = 1, but the real label should be safety = 0. It seemed that our models tended to predict images with empty spaces and gloomy color have a more probability of being Less Safe, and images with trees have a more probability of being More Safe.

The failure cases echoed what we had observed in qualitative analysis, and indicated that our models gave too many weights to trees in the predictions of More Safe; and gave too many weights to dull colored and older houses in the prediction of Less Safe. We believe that this issue could be improved if more training images are used in the future.



1. Failure cases. The upper images are top 4 wrongly predicted as safety = 0; The below images are top 4 wrongly predicted as safety = 1.

Third, information shifting when generating the label. The Streetscore project used the PlacePulse1.0 dataset provided by [1] for prediction. The original PlacePulse1.0 dataset had continuous 3 labels: QS Safer, QS Unique and QS Upperclass, ranging from 1-10. In [1], the author noted that the mapping between images and locations is not one-to-one. Therefore, for a large number of locations, they captured more than one image, by pointing the camera in two or more directions.

The Streetscore project transformed the PlacePulse1.0 dataset to create one continuous label - q-score, ranging from 1-10. It is notable that our project used data *predicted* by the Streetscore project. In Streetscore project’ open-sourced datasets, the label was a continuous q-score, ranging from -4.0 to 43.0. And we further generated a binary label ‘safety’ based on that. This series of information shifting may have contributed to the model inaccuracy as well.

Fourthly, we could try more aggressive image augmentation setups.

* 1. *Model Interpretation*

Our research showed that model interpretation tools are effective in understanding the models’ predictions. Qualitative analysis of model interpretation results could provide profound insights for researchers, city planners to understand what elements are mostly perceived as safe by residents. This makes it possible for the city planners to improve our living environment, and provide quantitative data for psychologists in the study of people’s perceptions towards the living environment.

1. Conclusions

In this project, we conducted exploratory research in using predicted datasets open-sourced by the Streetscore project and applying CNN, transfer learning, ensemble learning, and model interpretation techniques to predict people’s perceived safety on Toronto street views.

Our research laid a foundation for the City of Toronto to develop an automated predictor that can predict city-scene at scale for the entire city, so as to provide the residents and urban planners with this information for all parts of the city.

Our research also suggested that model interpretation results on the models’ prediction on perceived street safety could help researchers and city planners to understand what elements are mostly perceived as safe by residents and thus improve people’s living environment.

This research has several limitations, mainly about the training data. First, the purpose of this project is to predict the perceived safety for Toronto, however, we do not have labeled training images for Toronto. The performance of our model would not be as robust as it would be when using Toronto based training data. It is also not possible to effectively evaluate our prediction on Toronto images.

Second, the target label we generated was based on predicted values.

Third, there is also a time gap between the time we fetched the images and the images were fetched by the Streetscore project (2014) to predict q-score.

We suggest future research to collect manual-coded safety score data towards Toronto street view images. We argue that the availability of such data would largely enhance model performance.

Future works could also analyze the correlation between predictions on Toronto image with social factors such as crime and income like [1] and [3] did.

It would be also interesting to try whether different image augmentation setup and different pre-trained models could boost model performance.

We have just conducted qualitative research on some randomly selected LIME results. It would be very interesting to see systematic qualitative research using the LIME results.

To conclude, our project proved it is possible to create a high-resolution map of urban perception in Toronto, similar to what the Streetscore project did for other cities.

##### Acknowledgment

We would like to thank the kind guidance by Faezeh Khabbaz, our course instructor of the ML1030: Machine Learning at Scale course; Matthew Tenney, Data Science Supervisor of Data Analytics and Visualization team in Geospatial Competency Centre of the City of Toronto; and Ryan Garnett, Manager Geospatial Data Integration & Access of City of Toronto.

##### References

1. Salesses, P., Schechtner, K., & Hidalgo, C. A. (2013). The collaborative image of the city: mapping the inequality of urban perception. PloS one, 8(7), e68400.
2. Kelling GL, Wilson JQ (1982) Broken windows. Atlantic monthly. 249(3):29-38
3. Naik, N., Philipoom, J., Raskar, R., & Hidalgo, C. (2014). Streetscore-predicting the perceived safety of one million streetscapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 779-785).
4. Herbrich, R., Minka, T., & Graepel, T. (2007). TrueSkill™: a Bayesian skill rating system. In Advances in neural information processing systems (pp. 569-576).
5. Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500).
6. http://streetscore.media.mit.edu/data.html
7. https://bostonopendata-boston.opendata.arcgis.com/datasets/b601516d0af44d1c9c7695571a7dca80\_0
8. https://www.toronto.ca/city-government/data-research-maps/open-data/open-data-catalogue/#f71a13c4-fb51-6116-57b7-1f51a8190585
9. https://developers.google.com/maps/documentation/streetview/intro
10. https://pypi.org/project/google-streetview/
11. https://pypi.org/project/Shapely/
12. http://geopandas.org/
13. He, K., Girshick, R., & Dollár, P. (2018). Rethinking imagenet pre-training. arXiv preprint arXiv:1811.08883.