



Toronto Streetscore

Predicting Perceived Street Safety
using CNN





Background





Why does it matter?



Understanding people's perceived safety of street view would:

- Help city planners and designers to build a more friendly environment, improve people's wellbeing.
- Produce quantitative data for researchers to understand what elements contribute to people's perception of street safety.
- Contribute to the research on the connection between crime and urban perception.



Previous Research



Previous Research 1 - the Place Pulse Project

Images: 4136 geo-tagged images

- NY and Boston in US - 2942 images collected from Google Street View
- Linz and Salzburg in Austria - 1194 images collected manually onsite

Game: The participants were shown two images, selected randomly from the dataset, and were asked to click on one in response to one of the 3 questions:

- Which place looks safer?
- Which place looks more upper-class?
- Which place looks more unique?

Images were scored based on the human inputs.

Previous Research 2 - the Streetscore Project

Labels Used for Prediction:

- Converted the human inputs from the Pulse Project to a ranked score for each image using the Microsoft Trueskill algorithm.

Algorithm: support vector regression

Features: Commonly used image features extracted from images: GIST, Geometric Classification Map, Texton Histograms, Geometric Texton Histograms, Color Histograms, Geometric Color Histograms, HOG2x2, Dense SIFT, LBP, Sparse SIFT histograms, and SSIM.

Training data: 2920 images in NY and Boston

Predicted data:

- Generated high resolution maps of perceived safety for 21 cities in United States
- Only Boston and New York's predicted data were open-sourced



Project Objective and Design



Project Objective and Design Guidelines

Objective

Build an explainable machine learning model that could predict people's perceived safety for Toronto street view images, and provide the DAV team at City of Toronto with sufficient background info to implement this solution at scale.

Design

- Use state-of-the-art techniques: CNN, transfer learning, ensemble learning.
- Make use of the data generated by the Streetscore project for model building.
- Be representative of Toronto's zone classes when selecting training and testing data.
- Models would predict a binary target 'safety', with 0 = Less Safe; 1 = More Safe.
- Apply model interpretation tools (LIME) to understand the models' predictions.



Method





Data Collection, Exploration and Preparation



1. Proportion of Toronto Zoning Data

Proportion of Toronto Zoning Class

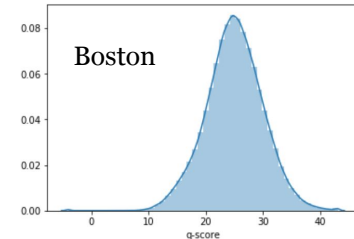
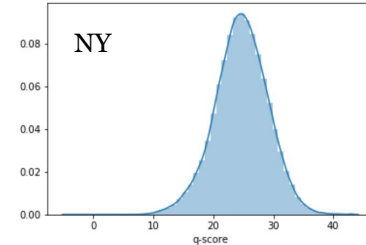
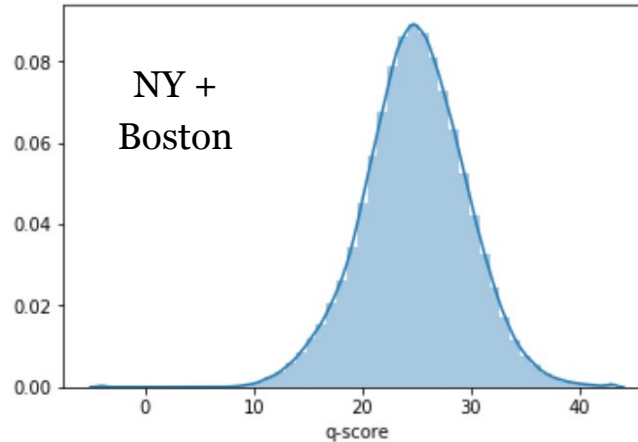
Zone Class	Percent (%)
residential	49.89
open space	17.86
employment industrial	13.9
unassigned	5.83
commercial	5.34
utilities	5.08
institutional	2.1

- Provided by City of Toronto
- Used the same proportion for subsampling Streetscore data, so that our training/validation/test data could be more representative of Toronto

2. Boston and New York Data with a Predicted Target

- The Streetscore project open-sourced two cities' data with ***predicted*** q-score by their algorithm
- Three columns: latitude | longitude | q-score
- The q-score of the 2 cities ranged from -4 to 43. (See analyses on the right)
- Note the q-score used by the Streetscore algorithm to make prediction was originally taken from PlacePulse1.0 dataset and converted using the Microsoft Trueskill algorithm. Both ranged from 0 - 10 *.

city	No. of Instance	Minimum	Maximum	Median	Mean
New York	322,386	-4.0	43.0	25.0	245.0
Boston	229,564	-4.0	43.0	24.7	24.7
NY + Boston	551,950	-4.0	43.0	24.9	24.9



*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).

Creating Our Target Column “Safety” Based on the Streetscore Dataset

- Combine the NY and Boston datasets to one dataset
- Create 2 bins based on the distribution of q-score
- Create binary target variable “safety” based on the 2 bins, 0 = Less Safe; 1 = More Safe
- Split the NY and Boston datasets.

latitude	longitude	q-score	city	binof2	binof2_class	safety
42.273857	-71.050980	36.497364	Boston	(24.882, 43.0]	1	1
42.274235	-71.051178	33.081684	Boston	(24.882, 43.0]	1	1
42.274429	-71.050896	21.050259	Boston	(-4.001, 24.882]	0	0

3. Boston Zoning Data

- The objective is to subsample Boston data which are representative to Toronto Zoning class.
- We obtained a dataset open-sourced by the City of Boston*
- It contained a 'geometry' column which is geometric polygon data, as well as a 'subdistrict' column, which was relevant to the zoning classes of Toronto.
- Didn't find NY Zoning data relevant to zoning classes of Toronto, so we would use Boston data to build our models.

Boston Zoning Data Open-Sourced by City of Boston*

OBJECTID	ZONE_	DISTRICT	MAPNO	ARTICLE	SUBDISTRICT	Unique_Cod	FAR	Shape_STAr	Shape_STLe	Zone_Desc	geometry
54194	CC	Mission Hill Neighborhood	6D	59	Business	Mission Hill Neighborhood CC	3.0	0	0	Community Commercial	POLYGON ((-71.09451646281816 42.33244043424968...
54195	WM	South Boston Neighborhood	4F	68	Industrial	South Boston Neighborhood WM	2.0	0	0	Waterfront Manufacturing	POLYGON ((-71.03554411066089 42.33992389263747...
54196	M-4	South Boston	4	Underlying Zoning	Industrial	South Boston M-4	4.0	0	0	Restricted Manufacturing	POLYGON ((-71.0421385497493 42.3462569178347, ...
54197	D St. NDA	South Boston Neighborhood	4F	68	Mixed Use	South Boston Neighborhood D St. NDA	2.0	0	0	Neighborhood Development Area	POLYGON ((-71.04142638209957 42.34454863344008...
54198	SUMMER ST. LI	South Boston Neighborhood	4F	68	Industrial	South Boston Neighborhood Summer St. LI	3.0	0	0	Local Industrial	POLYGON ((-71.03770255121084 42.33811666220058...

*https://bostonopendata-boston.opendata.arcgis.com/datasets/b601516d0af44d1c9c7695571a7dca80_0

Boston Data - Create a Subdistrict Variable

- First, create a 'coordinate' variable which are geometric points.
- Second, spatial join Boston Data with the Boston Zoning Data, to create a new column - 'subdistrict' for the Boston Data
- After spatial join, only 23.26% (53,401 out of 229,564) of the original Boston Data were left.

Boston Zoning Data

OBJECTID	ZONE	DISTRICT	MAPNO	ARTICLE	SUBDISTRICT	Unique_Cod	FAR	Shape_STAR	Shape_STLe	Zone_Desc	geometry
54194	CC	Mission Hill Neighborhood	6D	59	Business	Mission Hill Neighborhood CC	3.0	0	0	Community Commercial	POLYGON ((-71.09451646281816 42.33244043424968...
54195	WM	South Boston Neighborhood	4F	68	Industrial	South Boston Neighborhood WM	2.0	0	0	Waterfront Manufacturing	POLYGON ((-71.03554411066089 42.33992389263747...
54196	M-4	South Boston	4	Underlying Zoning	Industrial	South Boston M-4	4.0	0	0	Restricted Manufacturing	POLYGON ((-71.0421385497493 42.3462569178347, ...
54197	D St. NDA	South Boston Neighborhood	4F	68	Mixed Use	South Boston Neighborhood D St. NDA	2.0	0	0	Neighborhood Development Area	POLYGON ((-71.04142638209957 42.34454863344008...
54198	SUMMER ST. LI	South Boston Neighborhood	4F	68	Industrial	South Boston Neighborhood Summer St. LI	3.0	0	0	Local Industrial	POLYGON ((-71.03770255121084 42.33811666220058...



Boston Data with 'Safety' and 'Coordinates'

city	latitude	longitude	q-score	safety	Coordinates
Boston	42.273857	-71.050980	36.497364	1	POINT (-71.05098000000002 42.273857)
Boston	42.274235	-71.051178	33.081684	1	POINT (-71.05117800000002 42.274235)
Boston	42.274429	-71.050896	21.050259	0	POINT (-71.05089599999999 42.274429)
Boston	42.274529	-71.050491	12.665337	0	POINT (-71.05049100000002 42.274529)
Boston	42.274685	-71.052315	21.554031	0	POINT (-71.05231500000002 42.274685)



Boston Data with 'Safety' and 'Coordinates'

city	latitude	longitude	q-score	safety	Coordinates	OBJECTID	SUBDISTRICT
Boston	42.277477	-71.053329	21.189661	0	POINT (-71.05332900000001 42.27747700000001)	54219	Open Space
Boston	42.277805	-71.053894	17.655205	0	POINT (-71.053894 42.277805)	54219	Open Space

Boston Data - Stratified Downsampling

- We performed **stratified sampling based on both 'safety' and 'subdistrict' columns**
- Note that Subdistrict classes of Boston Data and the Zone classes of Toronto did not exactly match.
- We considered “Miscellaneous” + “Mixed Use” of Boston subdistrict as one class, and when subsampling, we would made it stratified of the proportion of “unassigned” + “utilities” in Toronto.

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

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

Boston Data - Stratified Downsampling

- Our aim is to **subsample 20,000** Boston data out of the 53,401.
- So we calculated for each subdistrict in Boston, how many samples we should select based on the proportion of Toronto Zoning class.

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



The **20,000 Boston Samples**
Aimed to Select - Count by Subdistrict

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

Boston Data - Stratified Downsampling

- While we subsample Boston data, 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.

The **20,000 Boston Samples**
Aimed to Select - Count by Subdistrict

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



Subsampled 20,000 Boston Data
Selected - 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

Boston Data - Train / Test Split

- For the downsampled 20,000 boston samples, we split them into **Boston Training Data (16000 samples) and Boston Test Data (4000) samples**.
- We would use these two datasets to fetch **Boston Training Image and Boston Test Image**.
- We would then **build models** with Boston Training Image and Boston Test Image.

4. Toronto Data

- We obtained the data from Toronto's Open Data Catalogue*. It contained 525,545 geolocations in Toronto, but no zoning class.
- We randomly sampled **2,100 Toronto data** to fetch Toronto images.
- We would use **final ensemble strategy to predict** on those Toronto images.

*<https://www.toronto.ca/city-government/data-research-maps/open-data/open-data-catalogue/#f71a13c4-fb51-6116-57b7-1f51a8190585>

5. Image Fetching and Pre-Processing

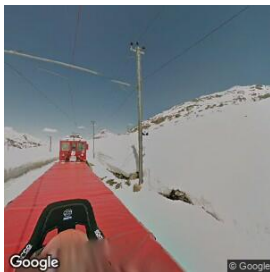
- Fetch images using Google Street View Static API. It has several parameters: location, size, **heading, fov, pitch, radius**
- At the **same geolocation**, images fetched **maybe different**, depending on the parameters.
- It is not clear what parameters were used by the Streetscore project while the image. Therefore, we **used default settings**.
- Not all geolocation in our datasets had an image can be fetched.

16,000 Boston Training Data -> fetched 15,928 Boston Training Images.

4,000 Boston Test Data -> fetch 3,976 Boston Test Images

2,100 Toronto Data -> fetch 2,034 Toronto Images.

5. Image Fetching and Pre-Processing



All default



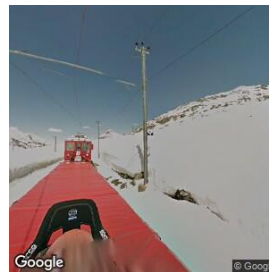
fov = 10



heading = 0



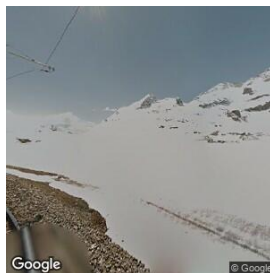
pitch = 90



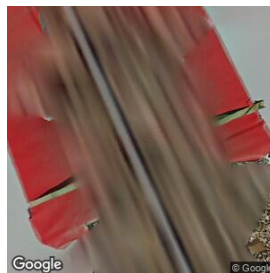
radius = 10



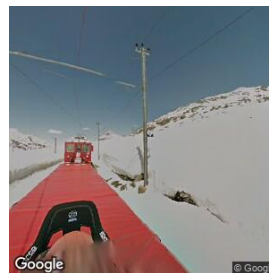
fov = 120



heading = 180



pitch = -90



radius = 100

Default is 90

Default is closest
direct *

Default is 0

Default is 50

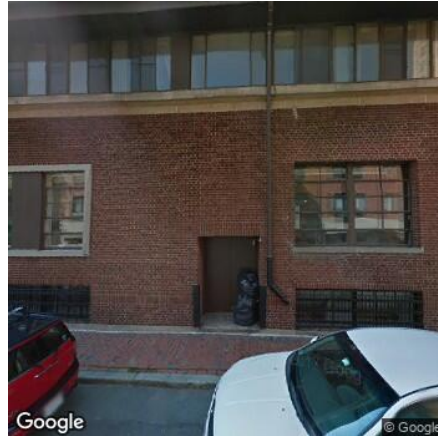
*Default of “heading” means “If no heading is specified, a value will be calculated that directs the camera towards the specified location, from the point at which the closest photograph was taken.”

Sample Boston Street View Images

Safety = 1, More Safe



Safety = 0, Less Safe



5. Image Fetching and Pre-Processing

- Google periodically updates street view images.
- **~80% of images we fetched and then used to build and test models were updated from the images used by the Streetscore project to arrive at the q-score.**
- Table on the right lists the updated year of images we fetched for the Boston Training and Test Images, based to the metadata of the images provided by the Google API.

Boston Train and Test Images' Update Year

Year Updated	Boston Training 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

5. Image Fetching and Pre-Processing

- Cropped out the Google logo in each image.
- When fit images into models, images were resized to 224×224 pixels (in transfer learning) and 100×100 (in self-designed CNN) pixels respectively.



Model Building



CNN

1. Our Self-Designed 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 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.

1. Our Self-Designed CNN Models

- Structure:

- > 2 convolutional layers

- > each followed by a pooling layer

- > passed to a flatten layer

- > fed to a couple of dense layers

- > added 2 dropout layers to prevent overfitting

- Optimizer: Adam

- Default values for learning rate, decay and momentum

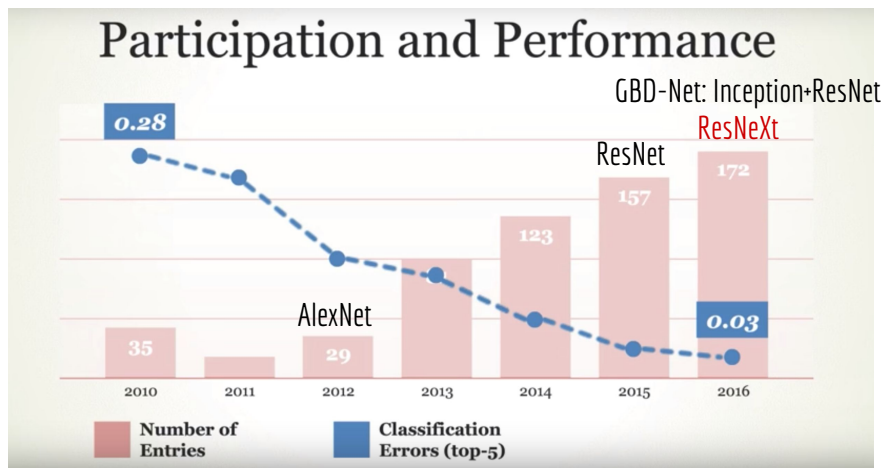
- Epochs

- 20 epochs for the first model, 25 epochs for the second model.

Transfer Learning - ResNeXt50

2. Transfer Learning- About ResNeXt

- 2nd place in image classification in 2016 ImageNet Large Scale Visual Recognition Challenge (LSVRC)
- Aggregated residual transformations. *



- Good balance of performance and number of parameters.

Documentation for individual models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
ResNeXt50	96 MB	0.777	0.938	25,097,128	-
ResNeXt101	170 MB	0.787	0.943	44,315,560	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

* 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).
Figure source: 1. ImageNet: Where Have We Been? Where Are We Going?" with Fei-Fei Li <https://www.youtube.com/watch?v=jYvBmJo7qjc> Association for Computing Machinery (ACM) Date: 9/21/2017
2. <https://keras.io/applications/#usage-examples-for-image-classification-models>

2. Transfer Learning- Pipeline

- Optimizer: Adam
- Learning rate: 0.0001
- 50 Epochs, and save best models among the 50 epochs were saved
- Image augmentation on the training set: for 1 train image, 1 augmented image was produced
- 5-fold cross-validation was implemented to both evaluate model accuracy and and produce 5 submodels for ensembling
- After cross-validation, 1 model was trained on the full Boston Train Images.

Therefore, a total of 6 models were generated in the transfer learning process.

Ensembling

1. 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 image, 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.

We Further Tried to Find the Best Ensemble Strategy:

- Using 7 models (1 from own CNN, 6 from Transfer learning)
- Ensembled the Predictions on Boston Test Images

2. Averaging Ensemble.

We tried 4 averaging ensemble strategies:

- averaging prediction of all single models;
- averaging predictions of models with best accuracy;
- averaging predictions of models best at predicting safety = 0
- averaging predictions of models best at predicting and safety = 1

3. Averaging + Conditional Ensemble.

We tried 2 conditional ensemble strategies:

- 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.
- 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.

4. Averaging + Weighted Ensemble.

We tried 3 weighted ensemble strategies for averaged predictions best at predicting $\text{safety} = 0$ and $\text{safety} = 1$:

- 0.4 vs 0.6
- 0.45 vs 0.55
- 0.55 vs 0.45

Model Evaluation

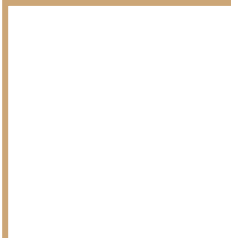
Confusion Matrix

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


Predictions on Toronto Images

Predictions on Toronto Images

- Applied the final ensemble prediction strategy for the prediction of the 2,034 Toronto images.
- No target label, so no way to evaluate effectively
- Visualized on a Toronto map



Results





Performance of Single Models on Boston Images



Performance of Single Models 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	-

- Most models experienced overfitting, single models' validation accuracy was all around 70%.
- the 2nd model produced by our own CNN had better performance and less overfitting comparing to that of our first CNN model. We believe that was thanks to the additional training images.

Results of Ensembling

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

- **2 models - cv3 and full were better at predicting safety = 0.**
- 5 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 were better than the validation accuracy on Boston Train Images. We believe that was thanks to the averaging predictions on augmented test images.

Ensemble Type	Ensemble Strategy	Model Name	FN	FP	TN	TP	TN_Rate	TP_Rate (Recall)	Accuracy	F1 score	Precision
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

Averaging ensemble:

- Averaged predictions had better accuracy than single models.
- Ensembling of 'all models' yielded the best accuracy (73.9%). However, it is neither best when predicting safety = 0 along nor best at predicting safety = 1 alone.
- '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:

- 'conditional ensemble1' model was good at prediction target 0, at the expense of prediction 1.
- 'conditional ensemble2' model was good at prediction target 1, at the expense of prediction 0.

Averaging + Weighted Ensemble:

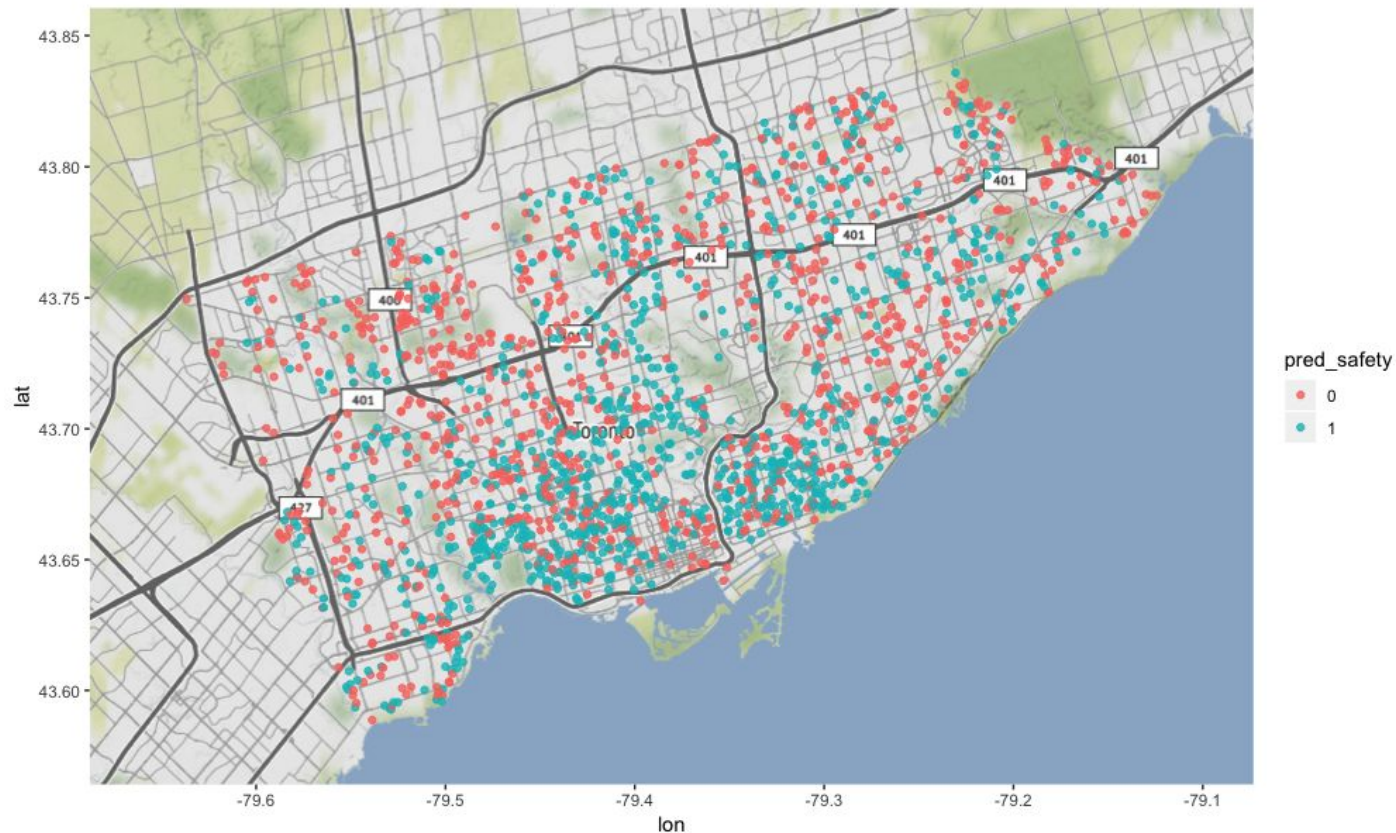
- 'weighted ensemble2' had an accuracy of 74.1%, and was good at predicting both safety = 0 and safety = 1. **Therefore, we would consider the method used in 'weighted ensemble2' as our final ensembling strategy and use it to predict on Toronto images**

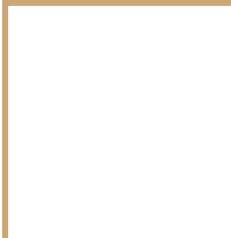
Results on Toronto Images

Ensembled Predictions on Toronto Images


We applied the ensemble strategy same 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).





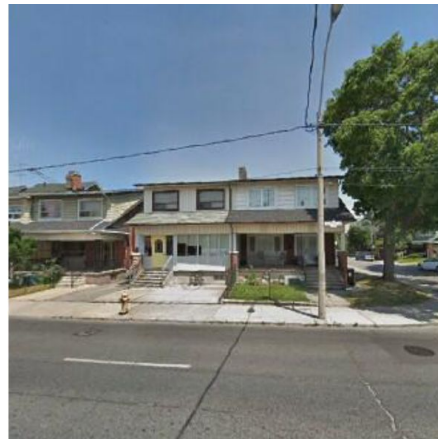
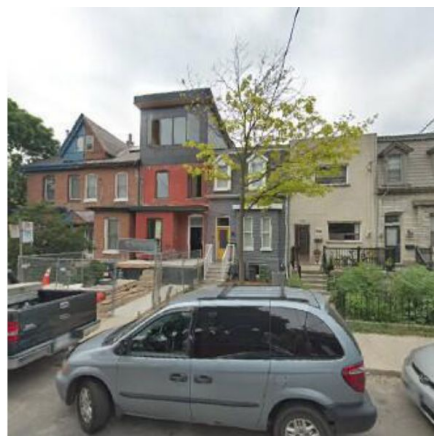
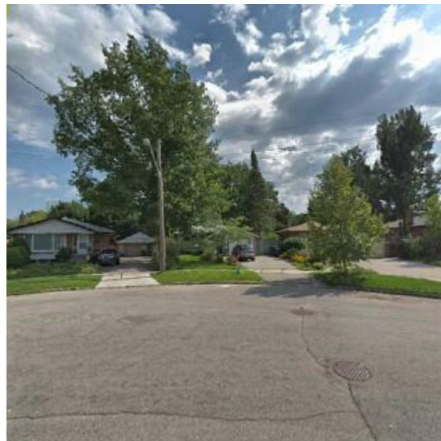
Model Interpretation



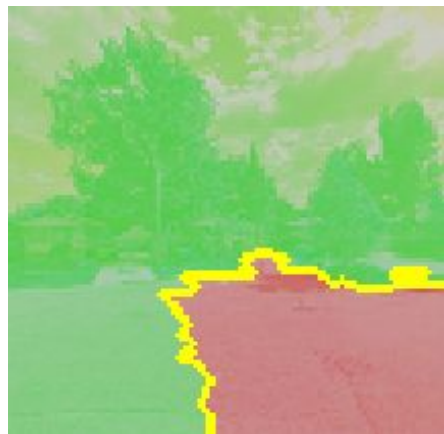
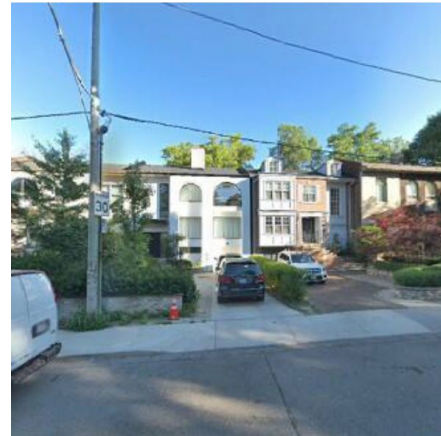
LIME

- Local Interpretable Model-agnostic Explanations.
- LIME uses the technique of varying the inputs and understanding how the outputs change. For image data, LIME creates variations of the image by segmenting image into super-pixels and turning the super-pixels on or off. Super-pixels are interconnected pixels with similar color.
- With Lime, we can interpret the prediction on each image. Also will be able to understand how each segment of the image contributed to (pros)/against (cons) the prediction.

More Safe



Less Safe



a



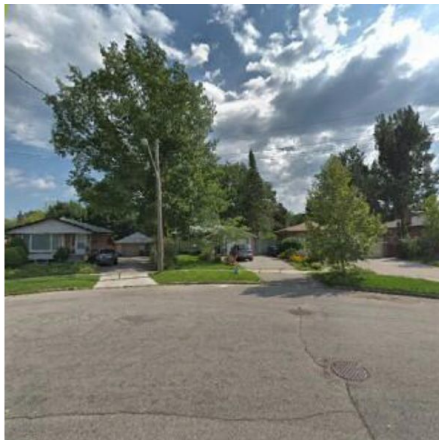
b



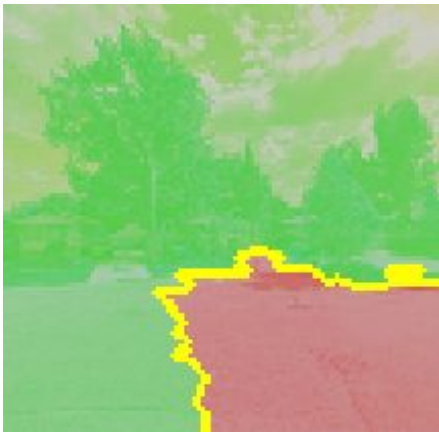
c



d



- LIME has predicted the image to be perceived as **More Safe**.
- The plants and trees in the image contributed to the More Safe prediction.
- Crack on the ground at the right corner of the image contributed to the Less Safe prediction.



a



b

- LIME has predicted the image to be perceived as **More Safe**.
- Brightly colored houses are perceived to be More Safe.
- Dull colored houses are perceived to be Less safe.
- Among the vehicles in the picture, the hatchback has contributed towards More Safe and the pick-up truck has contributed towards Less Safe perception.

- LIME has predicted the image to be perceived as **Less Safe**.
- Electrical wires hanging over the house contributed to the Less Safe prediction.



- LIME has predicted image to be perceived as **Less Safe**.
- Houses on the right side of the image seems to be older and have contributed to Less Safe perception
- Shadow on the ground also contributed to Less Safe perception
- The house on the left seems to be comparatively new. It contributed to the More Safe perception.





Discussion



1. We have spent more efforts on data preparation than running the model and ensembling
2. Transfer Learning vs Our Self-Designed CNN

Transfer Learning and our own CNN performed similarly. But Transfer Learning costed much higher computational resources.

3. The effectiveness of ensembling:

- 1) Averaging predictions on Augmented Test Images was useful.
- 2) The accuracy of all ensembling models was better than that of single models.
- 3) Our final ensemble strategy boosted the prediction accuracy to 74.1% from around 70% accuracy by single models.
- 4) Confusion matrix proved to be an effective tool in guiding us to find the best ensembling strategy.

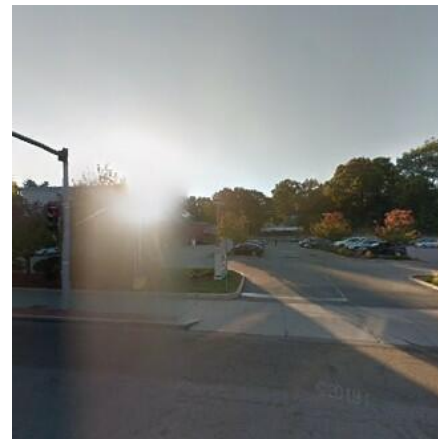
4. Model Performance

We assume several reasons may have affected our model performance:

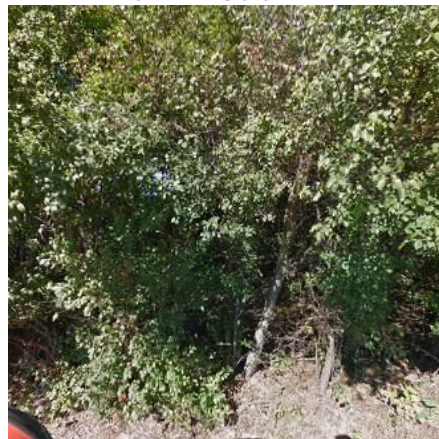
- 1) Most of the training images used for training and testing have been updated from the time Streetscore project was published in 2014. Also it is not clear what parameters did the Streetscore project use when fetching images.
- 2) Second, the number of training images was not large enough. We used only 15,928 Boston Train Images, for the submodels saved during cross validation, the training images are even fewer (~12,700). The different performance of our 2 models produced by our own CNN structure proved that.
- 3) Information shifting when generating the target label.
 - >PlacePulse1.0 dataset had **3 continuous labels** ranging from 1-10.
 - >The Streetscore project transformed it to create **1 continuous label** - q-score ranging from 1-10.
 - >Streetscore project's **predicted** open-sourced datasets had **1 continuous label** ranged from -4.0 to 43.0.
 - > We further generated **1 binary label** 'safety'.
- 4) We could try more aggressive image augmentation.

Failure Cases

Extremely wrongly predicted as safety = 0; Real label was safety = 1



Extremely wrongly predicted as safety = 1; Real label was safety = 0



5. Model Interpretation

- 1) Model interpretation tools were proved to be effective in understanding the models' predictions.
- 2) Qualitative analysis of model interpretation results can be very helpful for the city planners to improve our living environment.
- 3) It can provide quantitative data for research in the study of people's perceptions towards environment.



Conclusion



1. What we have done

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 **layed a foundation** for the 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.

2. Limitations

Training data: predicted data, updated image, no labeled Toronto data.

3. Future works

Investigate the **correlation between the predictions** on Toronto image **with social factors such as crime and income**.

Try different image augmentation setup and different pre-trained models.

Systematic qualitative research using the LIME results.



Thank you!

