Identifying Burglary Risks with Google Street View Images

Sam Bodmer, Junyi Li, Shenghan Lyu, Tim Yupeng Sun, Xingyu Gong

Abstract

Resident safety is an important aspect of city life. As the city's population surges, so does criminal activity, making it especially difficult to monitor and predict crime. Burglary, one of the most common forms of crime in New York City, poses a serious threat to residents' property and personal safety. To address these issues, this study utilizes Google Street View (GSV) imagery, burglary complaint data history, and demographic data to build a predictive model of burglary risk in New York City through different machine learning methods, especially convolutional neural networks (CNNs) and decision trees. Focusing on Manhattan, we identified visual cues in Street View that may predict higher burglary risk. We have completed broad studies and demonstrated the feasibility of utilizing these models to assist in effective policing. By comparing different machine learning methods, this project aims to help policy makers better predict the burglary risk and implement targeted interventions to enhance urban safety.

Introduction

Ensuring the safety of residents in a fast-growing urban center like New York is a complex challenge, especially as the population grows and crime rates, such as burglaries, climb. Traditional methods of crime prediction typically rely on historical crime data and demographic analysis, and often fail to adapt to the fast-paced changes occurring in urban environments. These traditional methods cannot adequately address the nuances and emerging patterns of modern urban crime (Kounadi et al. 2020). Recognizing the limitations of existing methods, this study adopts a new approach that leverages advanced technology to enhance crime prediction. By combining Google Street View (GSV) imagery with cutting-edge machine learning techniques such as Convolutional Neural Networks (CNNs) and Decision Trees, we aim to develop a more accurate burglary risk prediction model. This innovative approach aims to capture transient visual cues and environmental factors that are often overlooked by traditional analytics but are crucial for understanding and predicting urban crime patterns.

This paper is organized into different sections to provide a comprehensive overview of the research. It begins with a literature review that locates our study in the field of environmental criminology and machine learning. Then, details of our data collection and the use of convolutional neural networks (CNN) and decision trees to analyze Google Street View images are presented. The results section presents the effectiveness of our model, followed by a discussion of the implications for urban safety strategies.

Literature Review

For a considerable while, criminology research has been centered on urban environments, with researchers aiming to comprehend the intricate interactions between environmental elements that impact crime rates in urban areas. It is commonly agreed that because information important to decision makers is vast, scattered, and challenging to handle effectively, prediction markets are particularly helpful in anticipating crime rates and analyzing criminal policies (Wolfers et al., 2008). Conventional methods of predicting crime have predominantly depended on past crime statistics and demographic evaluations; nevertheless, these techniques frequently encounter difficulties in accommodating the complex and ever-changing landscape of urban crime. There is an urgent need for novel approaches that can identify the complex patterns of criminal behavior in urban settings as cities expand and change.

A theoretical framework for comprehending how social and spatial elements influence the incidence of crime is provided by environmental criminology. This viewpoint holds that social and economic factors, as well as the physical design of urban areas, influence the likelihood of criminal activity (Brantingham & Brantingham, 1995). Building on this framework of thought, current research has been focusing more on using machine learning methods and cutting-edge technology to improve attempts at crime prediction and prevention.

Machine learning, particularly convolutional neural networks (CNNs) and decision trees, has emerged as a powerful tool for analyzing complex datasets and identifying patterns in urban crime data. Originally created for image processing applications, CNNs have been applied as an innovative technique for evaluating a scene's perceived level of safety using Google Street View images (Porzi et al., 2015). Through the utilization of Street View photos' geographical information, scholars are able to discern characteristics linked to elevated levels of criminal activity, thus providing novel perspectives on the correlation between the constructed surroundings and criminal behavior.

Another well-known machine learning technique that provides an understandable and transparent method of crime prediction is decision trees. Decision trees help researchers to pinpoint crime traits that may lead to more investigation by partitioning the feature space according to the most useful features (Ahishakiye et al., 2017). This approach has been effectively used to anticipate crimes of many kinds, including burglaries, giving legislators and law enforcement authorities important information.

The efficacy of combining machine learning techniques with conventional crime analysis approaches has been shown in recent studies. Scholars can create more precise models of crime risk and pinpoint hotspots of criminal activity by merging demographic data, past crime records, and geospatial information (Wolfers et al., 2008). By providing a comprehensive understanding

of the dynamics of urban crime, these integrated techniques make it possible to take preemptive measures to improve public safety and security.

In conclusion, there are many fascinating prospects to deepen our comprehension of urban crime at the intersections of environmental criminology and artificial intelligence. In order to forecast, prevent, and plan for crime more effectively, researchers can use cutting-edge technologies and interdisciplinary approaches. This will ultimately lead to the creation of safer and more secure cities.

Data

NYPD Complaint Data Historic

This data is from NYC Open data. NYPD Complaint Data-Burglary primarily covers data on complaints about burglary received by nypd between December 2005 and December 2022 across nyc's 5 boroughs. The data includes the location of the incident, geographic information, and a brief description of the incident. In selecting the appropriate data, we chose between nypd arrest data and nypd complaints data; this data more accurately responds to the location of the incident compared to nypd arrest data, and the fact that this dataset includes a timeframe of up to 17 years allowed us to analyze Burglary trends over time. After data selection, we performed a number of pre-processing steps to ensure the quality and applicability of the data: we excluded complaint types other than burglary to ensure consistency in the dataset. Geographic location information was standardized for subsequent geographic information analysis, and missing data was processed, using methods such as padding and deletion to ensure data accuracy.

Demographic data

To examine the interaction between different demographic variables and those that may be correlated, including housing characteristics, total population, racial/ethnic composition, socioeconomic status, and housing characteristics, I utilized datasets from the 2020 Census section of the official website of the New York City Department of City Planning. They are from separate excel files named demo_2021acs5yr_nta, econ_2021acs5yr_nta, soc_2021acs5yr_nta, hous_2021acs5yr_nta.The investigation of these different factors was conducted within the framework of Neighborhood Tabulation Areas (NTAs). The primary demographic units chosen are NTAs, which are intended to approximate the boundaries of recognizable communities, due to their ability to encompass larger geographic areas in comparison to individual census tracts. It is more likely for NTAs to promote a more intuitive understanding of various urban locales. According to the consistent GEOIDs between different files, we could use spatial join of geographical shapefiles and datasets and analyze and conclude the characteristics of high-crime-rate areas.

Google Street View (GSV)

Street view imagery (SVI) can be obtained from the Google Street View (GSV) Server by utilizing the Google Map Street View Static Application Programming Interfaces (APIs).In this study, we first built sample sites for both burglary and no burglary places in order to collect GSV to represent the aspects of burglary. Based on the coordinates of these sample sites, we further developed a Python script to collect the metadata of GSV in NYC and download them. The image files were labeled with the coordinates of latitude and longitude. A total of 54,914 photographs from the experimental group, where a break-in took place, were saved in distinct folders called after the borough. Additionally, 49,412 images from the control group, where no break-in occurred, were placed in a different folder.

Method

Classification

We first tried multiple classification methods to build a prediction model.The study utilized a dataset comprising features such as location, median household income, and a binary response variable indicating the occurrence of a burglary (1 for occurred, 0 for not occurred). The data was split into training and test sets to validate the effectiveness of the predictive models.
We employed several machine learning algorithms to predict the likelihood of a burglary:

- **Random Forest Classifier:** This ensemble method used 100 trees and was initialized with a random state of 42 for reproducibility.This model is less sensitive to outliers and can handle a mix of features well
- **K-Nearest Neighbors (KNN):** This model is a simple and effective classification algorithm that works based on the principle of proximity: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.Given our data includes geographical coordinates, KNN could be a particularly relevant choice as it can naturally handle spatial relationship.We chose 9 neighbors to classify since this model has the highest accuracy .
- **Logistic Regression:** Logistic Regression is a robust statistical method for binary classification. Implemented with a 'liblinear' solver, this model aimed to provide a probabilistic understanding of burglary occurrences.
- **XGBoost Classifier:** XGBoost is a powerful ensemble learning technique used for regression and classification problems.It is known for its speed and performance, and is an implementation of gradient boosted decision trees designed for speed and performance.

Decision tree

Decision Tree Algorithm Overview

The decision tree algorithm is a powerful machine-learning technique used for classification and regression tasks. It constructs a tree-like structure where each internal node represents a decision based on the feature value, and each leaf node represents the final decision or prediction. One of the key advantages of decision trees is their interpretability. The resulting tree structure is intuitive and easy to understand, making it widely popular for tasks requiring strong interpretability. Additionally, decision trees can handle numerical and categorical data and are robust to outliers and missing values. When building decision trees, the algorithm aims to minimize impurity or maximize information gain at each split, ensuring that each node contains as homogeneous data as possible. This recursive process of partitioning feature space results in a tree that effectively separates different categories or predicts target values. For our problem, we use the decision tree algorithm to identify whether burglary exists in surveillance images from various regions of New York.

Feature Selection and Data Preprocessing

Before applying the Decision Tree algorithm, we first preprocess and select features from surveillance image data. Feature selection involves image processing of surveillance images to extract features related to burglaries, such as crowd density and frequency of suspicious objects. We also cleaned and normalized the dataset to ensure data quality and availability.

Model Training and Tuning

During the model training phase, we used the surveillance image dataset from various regions of New York to train the Decision Tree model. We employed the DecisionTreeClassifier from sklearn.tree as our model for training and tuning, using burglary and non-burglary images from different areas of New York to enhance the model's accuracy and generalization capability.

Convolutional neural networks

Convolutional neural networks (CNN) are often used for image processing and classification, because they are able to capture the two-dimensional relationships present in images. Color images are normally digitally represented as three 2D arrays. Although there are different color conventions such as HSV, RGB, and BGR, they all usually have three channels. Objects or features in an image will not take up just a single pixel, so the relationships between groups of near pixels are very important in classification, hence the CNN is very important. The convolutional kernels are basically a matrix of weights. This kernel is slid across the image arrays, where its weights are multiplied by the corresponding pixel values at each step, and then added together. This sum is then the output value for the corresponding position in the new output array. There are different ways to use convolutional filters by changing the stride and padding of the layer. The stride is how many pixels the kernel is shifted at each step and the

padding represents what method is used at the edges of the image for calculating the output. In this work we used a stride of one, meaning that at each step in calculating the convolutional output, the kernel is only shifted by one pixel. Note that a stride of one is the minimum value that stride can be.Since the kernels need a grouping of pixels that is the same size as the kernel itself, padding can be used to create outputs that are the same size as the input, however we used no padding, meaning that the output is always smaller than the input. Convolutional layers also can be used to try and extract more features. This is done by having many kernels at one layer, so that it outputs more channels, where the output of each kernel becomes one of the output channels. The output from a convolutional layer is then sent through a nonlinear activation function, and in our case we used the rectified linear unit (ReLU), which is one of the more common activation functions. The activation function is an essential part of any neural network, because it allows neural networks to approximate nonlinearities that would be impossible with only linear transforms. The ReLU function is computationally efficient to calculate, as well as its gradient, and it does not suffer the same problems of vanishing gradients that something like the sigmoid does. The vanishing gradient problem occurs with the sigmoid function when the absolute value of the input is very large, because at this point the sigmoid function will be approaching either one or negative one, with a gradient approaching zero. When this happens it can cause a lot of numerical difficulties during optimization.

After the data has been passed through a convolutional layer and an activation function, it gets passed through a max pooling layer. Although there are average pooling layers, I believe in practice max pooling layers are used much more, and they are what we used in this work. The max pooling layer takes in arrays and slides a kernel across them taking the maximum value found within a kernel to be the value of one of the elements in the output array. It basically reduces the size of the individual channels, while maintaining the total amount of channels. So convolutional layers increase the total size of the data in the axis of the channels, and max pooling layers decrease the total size of the data in the other two axes. Another way to think about it is that convolutional layers extract features, and pooling layers reduce dimensionality.

Lastly, in any classification CNN, the output of the final pooling layer is flattened, and sent through at least one fully connected layer and ultimately sent through a sigmoid activation function to give the probability prediction. In our case this was a binary classification problem so the sigmoid activation function is exactly what we needed. If it was a multi-class classification problem we would have had to use the softmax function.

In our implementation, we used the pytorch library, and implemented a CNN classifier in a class that allows a user to customize the number of convolutional and pooling layers, the kernel sizes of every convolution and pooling layer, the output channels of every convolutional layer, and the number and output sizes of fully connected layers. The model that we ended up with based on training losses and accuracy was one consisting of four convolutional and pooling layers, each

with a kernel size of three, and two fully connected layers. The convolutional layers output 16, 32, 64, and 128 channels, in that order, after taking in a 3 channel 300x600 pixel image. The first fully connected layer outputs 256 points per image which is fed through a ReLU function and then to another fully connected layer which outputs one number to be fed through the final sigmoid activation function.

The model was trained using the binary cross entropy loss function and the gradient descent on the model parameters was done using the built-in Adam optimizer with a learning rate of 0.001. Through trial and error, it was found that to increase accuracy, one model for each borough was required.

A custom data-loader was created for training the CNNs. The data-loader was a class that would allow you to select batch-size, and which borough to select data from. Since the images were organized in respective folders according to borough, the data-loader would read in all file-names from the correct folder, shuffle the names to randomize what the model received, and at each timestep read the images using an opencv function, convert them to a pytorch sensor, find the corresponding labels, and pass all of that along to be sent through the CNN. Also at every step the data-loader would ensure that any batch it passed along contained an equal amount of images from burglary and non-burglary locations.

At every step, the binary cross entropy of the predictions was calculated, and the Adam optimizer was used to optimize the model parameters. Batch sizes of 32 images were used and 500 steps per epoch was also chosen, because that was more than enough steps to ensure every image in each borough was trained on at least once. Each model was trained for 20 epochs, which took roughly 6 hours per model. At every step, the binary cross entropy was calculated for model evaluation, and if a new minimum was reached, the model would be validated using the test set, and if it achieved a greater accuracy than the previous accuracy, the model parameters were saved, and finally at the end of training, the model with the highest accuracy was chosen.

Result

Classification

Each model's performance was evaluated using metrics like accuracy, precision, recall, and F1-score to assess their effectiveness in predicting burglary occurrences.

- **Random Forest:** The Random Forest model achieved an accuracy of 73.19%, highlighting its reliability. Notably, it excelled in recall for predicting burglaries, scoring 0.87 for the positive class. This indicates that the model is proficient at identifying actual burglary cases, which is crucial for minimizing false negatives in real-world applications.

	precision	recall	f1-score	support
0	0.48	0.33	0.39	5175
1	0.79	0.87	0.83	14550
accuracy			0.73	19725
macro avg	0.64	0.60	0.61	19725
weighted avg	0.71	0.73	0.71	19725

- **KNN:** The KNN model recorded a higher accuracy of 76.68%. Its standout performance is in recall for burglary cases, with a remarkable score of 0.94 for the positive class. This suggests that KNN is highly effective in detecting burglaries, although its overall accuracy indicates room for improvement in other areas.

	precision	recall	f1-score	support
0	0.68	0.26	0.38	5175
1	0.78	0.96	0.86	14550
accuracy			0.77	19725
macro avg	0.73	0.61	0.62	19725
weighted avg	0.76	0.77	0.73	19725

- **Logistic Regression:** The Logistic Regression model presented a mixed performance with an accuracy of 73.74%. However, it failed to predict non-burglary instances, reflected in a recall of 0.00 for the negative class. This indicates a significant limitation, as it means the model struggles to identify instances where no burglary has occurred, leading to high false-positive rates.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	5175
1	0.74	1.00	0.85	14550
accuracy			0.74	19725
macro avg	0.37	0.50	0.42	19725
weighted avg	0.54	0.74	0.63	19725

- **XGBoost:** XGBoost exhibited the highest accuracy among the tested models at 77.83%. Its recall for the positive class was an impressive 0.98, suggesting a high sensitivity to burglary predictions. This makes XGBoost a strong candidate for practical implementation, given its ability to accurately identify burglary occurrences.

	precision	recall	f1-score	support
0	0.80	0.20	0.33	5175
1	0.78	0.98	0.87	14550
accuracy			0.78	19725
macro avg	0.79	0.59	0.60	19725
weighted avg	0.78	0.78	0.73	19725

Decision tree

Model Performance Evaluation

We used test set data to evaluate the performance of the trained Decision Tree model. We employed metrics such as accuracy, precision, recall, and F1-score to assess the model's classification performance. Experimental results demonstrate that our trained Decision Tree model achieved high accuracy and good performance in identifying burglary in surveillance images from various regions of New York.

We tested our model using Manhattan data as an example, and the overall performance results are as follows:

Manhatten Data using Decision Tree model

	precision	recall	f1-score	support
0	0.93	0.93	0.93	9856
1	0.47	0.47	0.47	1301
accuracy			0.88	11157
macro avg	0.70	0.70	0.70	11157
weighted avg	0.88	0.88	0.88	11157

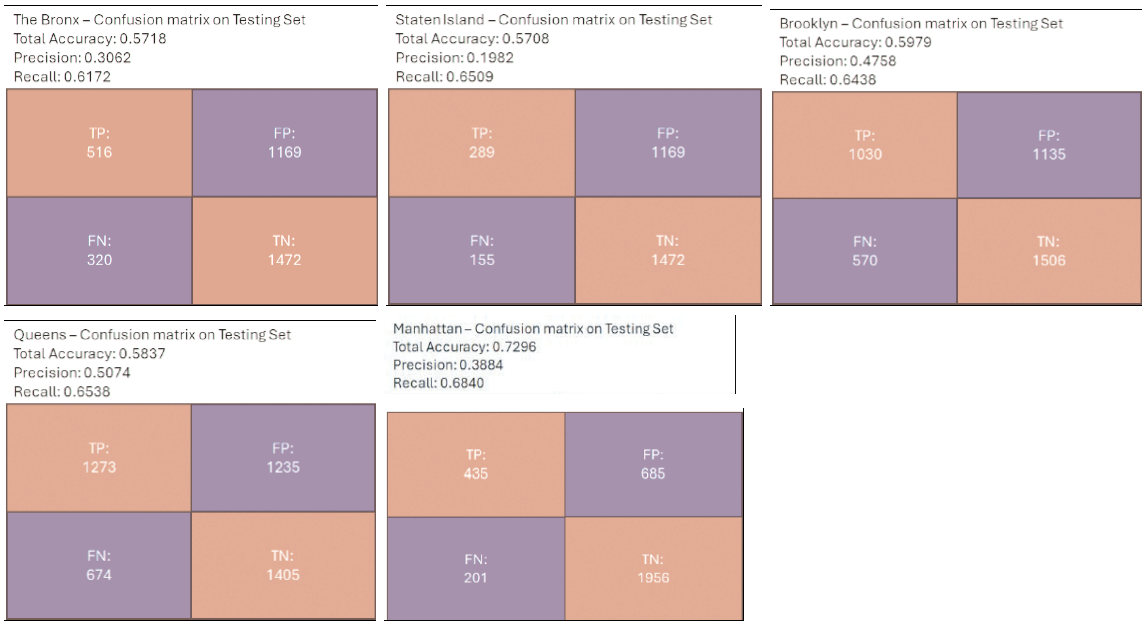
Model Application and Practical Effects

We applied the trained Decision Tree model to real-world scenarios and conducted real-time identification and monitoring of surveillance images from various regions of New York. Through the application of the model, we successfully identified some potential burglary situations and took timely measures to address them, effectively improving the city's safety and security level.

Convolutional neural networks

The CNN’s were trained for each borough. The architecture of the model took some time to get right. Initially, the loss function during training was stuck at a value of 0.6931, which is equal to

ln(2). If you look at the equation of binary cross entropy, you can derive that if it's consistently returning a value of ln(2), then the model is predicting 0.5 for everything, meaning that it is not extracting enough information to make any good type of prediction. This is what led us to using four sets of convolutional and pooling layers, and increasing the final convolutional layer channel output to 128. After making these changes we started to see some good results. The figures below show the confusion matrix, accuracies, precision, and recall of all five models.



The highest accuracy attained was in the model trained for Manhattan and it had a value of 0.7296. The lowest accuracy attained was for Staten Island with a value of 0.5708. We think that Manhattan has one of the largest disparities in wealth and so that it may be more visually apparent than in other boroughs, leading to a better classification accuracy. Also there is a higher proportion of apartment buildings in Manhattan than the other boroughs, which tend to have less burglary incidents, meaning that the Manhattan model may not have to find visual features on or around the building, but can try to abstract the type of building for predicting burglary probability.

Staten Island on the other hand has a much lower standard deviation in median income than the other boroughs, so the working theory is that the neighborhood has a more similar amount of money and will have a relatively more similar appearance when compared to the other boroughs. Since this model only takes into account the images, the appearance of the environment is very important.

One interesting thing is that all models had a recall value ranging from about 0.62 to 0.68. This means that all models were able to predict most of the data-points that were burglary, meaning

that if one can trust the no burglary predictions more than the burglary predictions. This in itself gives some direction on how one could use this model.

Although these accuracies aren't as high as someone might want, the models were able to abstract some useful features and make successful predictions, better than just choosing the mean. Most examples of image classification problems are things that humans could do such as deciding whether there is a cat in an image or not, but this problem is not something that a person could do with any certainty. This is a tough classification problem only using visual cues, so we think that the accuracies that were attained are actually quite good, all things considered.

We did attempt to look at some of the trained convolutional kernels to see what features were being extracted, but they were all pretty abstract and did not give anything conclusive. The next steps for this model would be to integrate some demographic data into it. Either by training it with demographic data that gets added to the model in the fully connected layers, or by using the output of the current models as a new feature to be run through something like a decision tree, a random forest, or some other machine learning model.

Discussion

The evaluation of multiple classifiers for predicting burglary occurrences revealed that XGBoost and KNN emerged as the top performers, with accuracies of 77.83% and 76.68%, respectively. XGBoost achieved the highest recall for burglaries at 0.98, indicating its strong sensitivity and reliability. While the Random Forest model also performed well with a recall of 0.87, Logistic Regression showed significant limitations, failing to predict non-burglary instances, resulting in a recall of 0.00 for the negative class. The CNN models, although less accurate than XGBoost and KNN, demonstrated consistent recall values (0.62 to 0.68) across different boroughs, suggesting their potential when combined with additional data sources. The Decision Tree model, with high accuracy in practical applications, particularly in Manhattan, underscores its utility in real-time surveillance and monitoring scenarios.

In summary, XGBoost and KNN are the most promising models for practical implementation due to their high accuracy and recall. However, the CNN models highlight the complexity of visual-based predictions and the necessity for integrating demographic or other contextual data to enhance performance. The Decision Tree model's effectiveness in urban settings like Manhattan further supports the need for tailored approaches based on specific regional characteristics.

Despite some limitations, the models demonstrated the ability to abstract useful features from surveillance images and make successful predictions, outperforming random guessing. Given the complexity of predicting burglaries based solely on visual cues, the achieved accuracies are commendable.

Future improvements could involve integrating demographic data into the models. This could be achieved by incorporating demographic features in the fully connected layers of the CNN or using the current model outputs as new features in ensemble models like decision trees or random forests. This integration could enhance the models' predictive capabilities by combining visual and demographic insights, leading to more robust burglary prediction systems.

Conclusions

The comparative analysis of various classifiers for predicting burglary occurrences demonstrates that XGBoost and KNN are the most effective models, achieving high accuracy and recall. XGBoost stands out with an accuracy of 77.83% and a recall of 0.98, indicating its strong sensitivity to burglary predictions. While the CNN models showed potential, their performance varied significantly across different boroughs, highlighting the importance of incorporating additional data sources for more robust predictions. The Decision Tree model's success in real-time applications, especially in Manhattan, underscores its practical utility in enhancing urban security. For policy makers who want to use machine learning to help identify the high risk burglary area, we have following advices:

1. Implement XGBoost and KNN Models for Surveillance

Given their high accuracy and recall, XGBoost and KNN models should be prioritized for deployment in surveillance systems across urban areas. These models can effectively identify burglary occurrences, allowing for timely interventions and improved public safety.

2. Tailor Models to Regional Characteristics

The significant variation in CNN model performance across boroughs suggests the need for tailored approaches. Policy makers should consider region-specific models that account for unique environmental and socioeconomic factors, as demonstrated by the higher accuracy achieved in Manhattan.

3. Utilize Decision Tree Models for Real-Time Monitoring

The Decision Tree model's effectiveness in real-time identification and monitoring makes it a valuable tool for immediate deployment in high-risk areas. Policymakers should leverage this model to enhance real-time surveillance capabilities, particularly in densely populated urban regions.

References

[1]Ahishakiye, E., Taremwa, D., Opiyo, E., & Niyonzima, I. (2017, May). Crime prediction using decision tree (J48) classification algorithm. https://www.researchgate.net/publication/316960839_Crime_Prediction_Using_Decision_Tree_J48_Classification_Algorithm

[2]Brantingham, P. J., & Brantingham, P. L. (1995, January). Criminality of place: Crime generators and crime attractors. European Journal on Criminal Policy and Research, 13(3), 5-26. https://www.researchgate.net/publication/321478569_Criminality_of_Place_Crime_Generators_and_CrimeAttractors

[3]Porzi, L., Buló, S. R., & Lepri (2015, October 13). *Predicting and understanding urban perception with convolutional neural networks: Proceedings of the 23rd ACM International Conference on Multimedia*. ACM Conferences. https://dl.acm.org/doi/10.1145/2733373.2806273

Wolfers, J., Zitzewitz, E., & Henderson, T. (2008, April). Predicting crime. https://chicagounbound.uchicago.edu/cgi/viewcontent.cgi?article=1374&context=law_and_economics

[4]Kounadi, Ourania, Alina Ristea, Adelson Araujo Jr, and Michael Leitner. "A Systematic Review on Spatial Crime Forecasting - Crime Science." BioMed Central, May 27, 2020. https://crimesciencejournal.biomedcentral.com/articles/10.1186/s40163-020-00116-7.

[5]Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[6]Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (June 2017), 84–90. https://doi.org/10.1145/3065386

[7]Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR, abs/1409.1556*.

Appendixes

Name	NetID	Contribution	Percentage
Sam Bodmer	SGB329	CNN	35%
Junyi Li	jl14864	GSV data Classification	20%
Xingyu Gong	xg2337	Demographic data	15%
Shenghan lyu	sl10942	Burglary data	15%
Tim Wu	ts5106	Decision tree	15%

All authors contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript and poster presentation.