Crime Forecasting using Satellite Imagery

He Zhou

University of Minnesota, 3rd Stats PhD

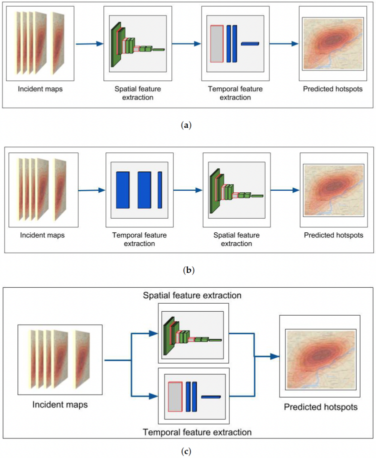
April 27, 2022

Crime Forecasting

- **Goal:** Predict future high-risk crime areas (hot spots) through past spatial and temporal information
- **Tasks:**
  - Classification (hot spot forecasting): crime / no crime
  - Prediction: number of crimes

Motivation: Examine Deep Learning Architectures

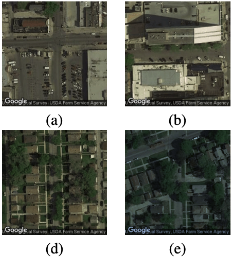
"Examining Deep Learning Architectures for Crime Classification and Prediction" [Stalidis et al., 2021]



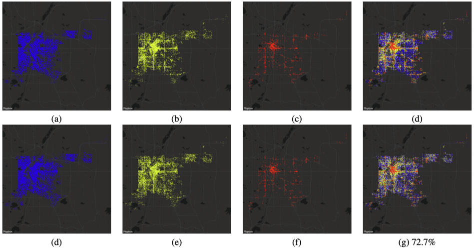
- Use only *location*, *time*, *crime type* from incident reports
- Crime classification and prediction simultaneously

Motivation: Satellite Imagery

"Crime Mapping from Satellite Imagery via Deep Learning" [Najjar et al., 2018]



(a) Satellite images



(b) Crime maps (Accuracy: 79%)

Figure: Visual features contained in satellite imagery can be used as a proxy indicator of crime rates

Question

Question: Forecast crime hotspots with satellite imagery?

Problem Formulation

- **Incident Report Dataset:**
  - Location: longitude & latitude
  - Time: year-month-day & time
  - Crime type: C types
- **Grid [Lin et al., 2018]:**
  - cell edge size ℓ
  - $H \times W$ cells grid
- **Aggregation:**
  - Given timespan t, crime type c, cell (i, j),
  - Count sum of occurences y.
  - Multiple Incident Maps for period T: (T, C, H, W)
- **Satellite Images:**
  - For each cell (i, j), get static map[1] centered around centroid
  - pixels: 256×256; zoom level: 17

[1]Google Static Maps API:
https://developers.google.com/maps/documentation/static-maps

Spatio-temporal Model: Dynamic Feature Extraction

Examine Deep Learning Architectures for Crime Classification and Prediction [Stalidis et al., 2021]

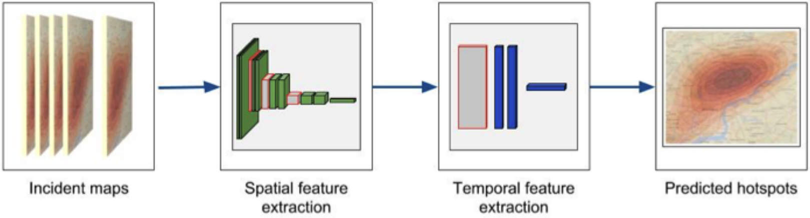


Figure: Feature Extraction: the spatial features first then the temporal (SFTT).

VGGNet : Spatial Feature Extraction

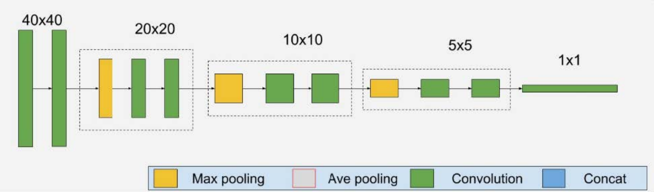


Figure: VGGNet [Simonyan and Zisserman, 2014]

CNN Architecture:

$$\text{incident map} \rightarrow (3\times 3\ \text{conv}, 32)\times 2 \xrightarrow{\text{pool}} (3\times 3\ \text{conv}, 64)\times 2$$
$$\xrightarrow{\text{pool}} (3\times 3\ \text{conv}, 128)\times 2 \xrightarrow{\text{pool}} (3\times 3\ \text{conv}, 256)\times 2$$
$$\rightarrow (1\times 1\ \text{conv}, 256)\times 2 \xrightarrow{\text{flatten}} \text{spatial features}$$

Remark: Batch normalization and dropout layers used to avoid overfitting.

LSTM : Temporal Feature Extraction

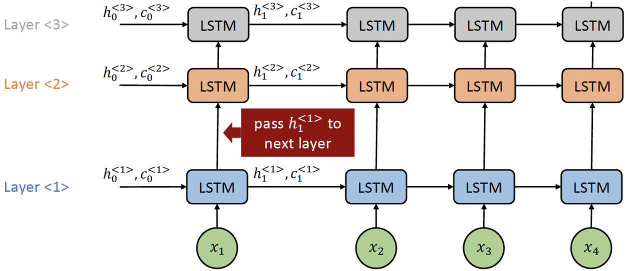


Figure: [2] *Stacked LSTMs [Hochreiter and Schmidhuber, 1997]*

- x_t: spatial features
- $h_t^{<1>} \in \mathbb{R}^{500}$, $h_t^{<2>} \in \mathbb{R}^{500}$, $h_t^{<3>} \in \mathbb{R}^{N}$
- $(h_1^{<3>}, \cdots, h_T^{<3>})$: dynamic spatio-temporal features

[2] Credit: https://yaoyichi.github.io/spatial-ai.html

AlexNet: Static Feature Extraction

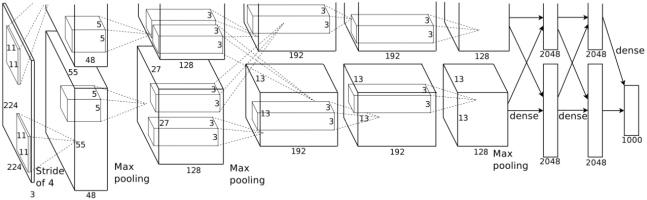


Figure: AlexNet [Krizhevsky et al., 2012] as Static Feature Extractor

- $s_{(i,j)} \in \mathbb{R}^{3\times256\times256}$: satellite image centered at centroid of cell (i,j)
- $f_{(i,j)} \in \mathbb{R}^{1000}$: static features extracted by pre-trained AlexNet
- concatenate static features with the dynamic features
- feed into fully connected output layers

Model Architecture

```
(relu): ReLU(inplace=True)
(dropout): Dropout(p=0.5, inplace=True)
(maxpool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
(conv11): Conv2d(11, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(conv12): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(conv21): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(conv22): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(conv31): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(conv32): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn3): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(conv41): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(conv42): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(conv51): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
(conv52): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
(lstm1): LSTMCell(6400, 500)
(lstm2): LSTMCell(500, 500)
(lstm3): LSTMCell(500, 1600)
(identity): Identity()
(fc_static): Linear(in_features=4096, out_features=1000, bias=True)
(fc1): Linear(in_features=1030, out_features=2, bias=True)
(fc2): Linear(in_features=1030, out_features=1, bias=True)
```

Figure: Model Architecture with 34, 666, 013 *parameters*

Custom Loss

- **Classification Loss:** Binary cross-entropy (BCE) loss

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[\hat{o}_i \log o_i + (1-\hat{o}_i)\log(1-o_i)]$$

- **Prediction Loss:** Mean squared error (MSE) loss

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- **Total Loss:**

$$Loss = BCE + \lambda \cdot MSE$$

where λ is the regularization parameter for MSE.

San Francisco Incident Report

Figure: Data table preview: San Francisco Police Department Incident Reports[3] (2018 to Present)

2021/01/...	2021/01/30	09:43	2021	Saturday	2021/01/...	10006970...	1000697	210038063		VS	Vehicle S...
2021/01/...	2021/01/29	11:20	2021	Friday	2021/01/...	10011800...	1001180	216011027		II	Coplogic I...
2021/01/...	2021/01/17	11:59	2021	Sunday	2021/01/...	10018912...	1001891	216012075		II	Coplogic I...
2021/02/...	2021/02/03	05:31	2021	Wednesday	2021/02/...	10019721...	1001972	210075554	210340313	II	Initial
2021/02/...	2021/02/03	12:25	2021	Wednesday	2021/02/...	10022652...	1002265	216012514		II	Coplogic I...
2021/02/...	2021/02/04	14:57	2021	Thursday	2021/02/...	10024100...	1002410	210041903		VS	Vehicle S...
2021/01/...	2021/01/16	21:00	2021	Saturday	2021/01/...	10025797...	1002579	216012768		II	Coplogic I...
2021/02/...	2021/02/05	13:50	2021	Friday	2021/02/...	10026950...	1002695	210049452		VS	Vehicle S...
2021/02/...	2021/02/06	13:30	2021	Saturday	2021/02/...	10043010...	1004301	216015154		II	Coplogic I...
2021/02/...	2021/02/09	15:00	2021	Tuesday	2021/02/...	10046920...	1004692	210091629	210410059	VS	Vehicle S...
2021/02/...	2021/02/12	13:17	2021	Friday	2021/02/...	10047180...	1004718	200297570		VS	Vehicle S...
2021/02/...	2021/02/13	14:24	2021	Saturday	2021/02/...	10049800...	1004980	210099156		VS	Vehicle S...
2021/02/...	2021/02/13	07:55	2021	Saturday	2021/02/...	10052497...	1005249	210098540	210440509	IS	Initial Sup...

- Start year: 2018 - End year: 2022
- 1572 days, 52 months

[3] https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-2018-to-Present/wg3w-h783

San Francisco Incident Report

Crime Types: {Assault, Theft, Robbery, Burglary, Motor Vehicle, Arson, Homicide, Vice, Narcotics, Other}

Table: Mapping of 47 Crime Categories to the 10 Crime Types [Stalidis et al., 2021].

Crime Type	**Crime Category**
Assault	Assault Disorderly Conduct Offences Against The Family And Children Weapons Carrying Etc Weapons Offence Weapons Offense
Theft	Larceny Theft Malicious Mischief Stolen Property Vandalism
Robbery	Robbery
...	...

San Francisco Incident Report

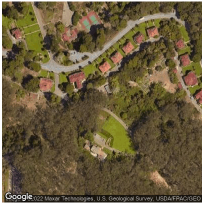
Table: *Column 2: Number of Incidents for each crime type;*
Column 3: Mean number of hotspots per day for every crime type with 40×40 *cell grids.*

Crime Type	Num. of Incidents	Mean Num. of hotspots
Assault	59,773	27.78
Theft	202,506	93.77
Robbery	13,106	7.33
Burglary	32,816	17.99
Motor Vehicle	45,034	25.67
Arson	2,464	1.46
Homicide	64	0.04
Vice	1,791	0.79
Narcotics	13,564	4.48
Other	143,169	62.57
Total	514,287	# CELLS 951

San Francisco Satellite Images

(a) *Cell 985 with 9,194 incidents*

(b) *Cell 1028 with 7,779 incidents*



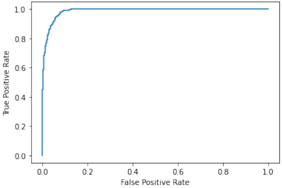
(c) *Cell 1170 with 1 incidents*

(d) *Cell 1135 with 1 incidents*

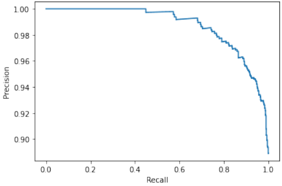
Experimental Settings

- Grid:
  - cell edge size $\ell \approx 450$ m
  - 40×40 cells grid
- Aggregation:
  - timespan t: 1 day
  - period T: 30 days
  - Incident maps: $(30, 10, 40, 40)$
- Training:
  - First 39 periods for training and last 13 for testing
  - Loss regularization parameter λ: tune for different crime types
  - Optimizer: *SGD*
  - Training: $batch\_size = 3$, $epoch = 100$

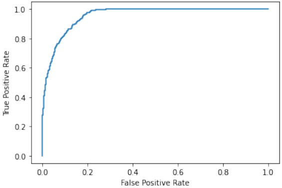
Results



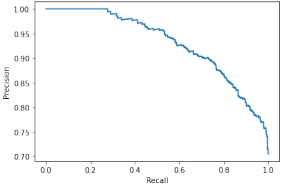
(a) All Crime ROC Curve (auroc = 0.989)



(b) All Crime Precision-Recall Curve (aps = 0.988)

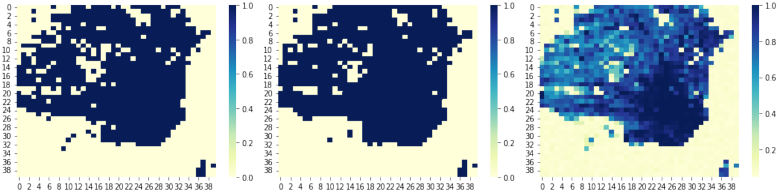


(c) Theft ROC Curve (auroc = 0.957)

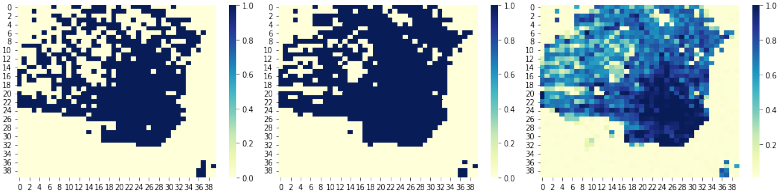


(d) Theft Precision-Recall Curve (aps = 0.934)

Results

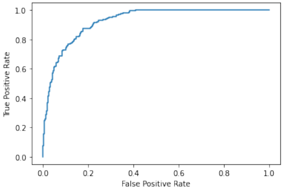


(a) *All crime true hotspots* (b) *pred. hotspots* (c) *pred. prob.*

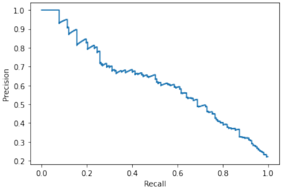


(d) *Theft true hotspots* (e) *pred. hotspots* (f) *pred. prob.*

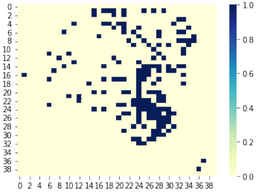
Results



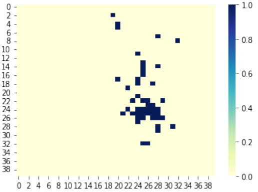
(a) Robbery ROC Curve (auroc = 0.925)



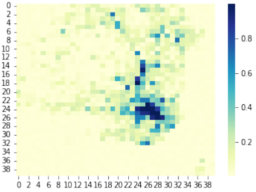
(b) Robbery Precision-Recall Curve (aps = 0.620)



(c) Robbery true hotspots



(d) pred. hotspots



(e) pred. prob.

Conclusions

- Use VGGNet + LSTM to extract dynamic spatio-temporal features from incident maps
- Use AlexNet to extract static spatial features from satellite imagary
- Forecasting for "All Crime" performs better than specific crime type
- Explore more on the effect of satellite imagery
- Ethical issue with crime forecasting [Gstrein et al., 2019]

References I

O. J. Gstrein, A. Bunnik, and A. Zwitter. Ethical, legal and social challenges of predictive policing. *Católica Law Review, Direito Penal*, 3 (3):77–98, 2019.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Y.-L. Lin, M.-F. Yen, and L.-C. Yu. Grid-based crime prediction using geographical features. *ISPRS International Journal of Geo-Information*, 7(8):298, 2018.

A. Najjar, S. Kaneko, and Y. Miyanaga. Crime mapping from satellite imagery via deep learning. *arXiv preprint arXiv:1812.06764*, 2018.

References II

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

P. Stalidis, T. Semertzidis, and P. Daras. Examining deep learning architectures for crime classification and prediction. *Forecasting*, 3(4): 741–762, 2021.

Thank You