



Towards a Deep Learning Approach for Urban Crime Forecasting

Freddy Piraján^(✉), Andrey Fajardo^(✉), and Miguel Melgarejo^(✉)

Laboratory for Automation and Computational Intelligence,
Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
1@evocom.com.co, lafajardof@correo.udistrital.edu.co,
mmelgarejo@udistrital.edu.co

Abstract. This paper presents a deep learning approach for urban crime forecasting. A deep neural network architecture is designed so that it can be trained by using geo-referenced data of criminal activity and road intersections to capture relevant spatial patterns. Preliminary results suggest this model would be able to identify zones with criminal activity in square areas of $500 \times 500 \text{ m}^2$ in a weekly scale.

Keywords: Deep learning · Convolutional neural networks · Environmental criminology · Crime forecasting

1 Introduction

Recent advances in environmental criminology rely on pattern recognition, urban factors identification and crime prediction based on temporal patterns [1]. Urban factors may include employment status and home living locations around the cities. In fact, some of them involve relationships between urban areas with high marginalized zones, inequality conditions and poverty with the motivation of people to get involved in criminal activities. For instance, local related neighborhood interactions like house burglary are considered as local crime-related factors [2].

Some traditional methods use time series from historical records to study criminal activity at a local neighborhood area with little or no consideration of the spatial distribution of urban crime data whereas others focus only on the geographic determination of crime clusters [3]. More recently, criminology experts have developed interest to adopt deep learning techniques in their work in order to generate policies to combat criminal activity [4–6].

Environmental criminology has increased its interest in the relationship between crime activity and the urban backcloth associated with it [7]. Experts focus on crime as a complex phenomenon [8] whereas conventional methods study crime activity based on data information like individual economical status, level of education and past crime occurrences [9]. Therefore, information like spatial patterns from spatio-temporal crime signals has received considerable attention [10]. Particularly, in Bogotá city (Colombia, South America), theoretical tools from criminology have been adopted in order to gain a better under-

standing of criminal activity. This approach has been useful to direct police patrolling efforts to zones where criminal activity is highly plausible [11].

There upon environmental criminology perspective, data contributions can be integrated with data visualization techniques [12] and artificial intelligence methods [13,14] to study spatial and temporal features [15] altogether to provide additional statistics related to crime activity. In this work, this particular phenomenological approach is used to compute spatial-temporal signals that might reveal useful information about thefts events in Bogotá city. In addition, strengths and weaknesses of a deep learning architecture for crime prediction are presented using some statistics to assess model performance in conjunction with data visualization while keeping a convenient number of parameters.

This paper is organized as follows: Sect. 2 presents our method: data pre-processing, model architecture, experimental setup, deep learning training strategy and validation scheme. Section 3 presents preliminary results and findings. Finally, we draw conclusions and comment about recommendations for upcoming research in Sect. 4.

2 Materials and Methods

2.1 Data Base

The database¹ contains reports of mobile phone thefts in Bogota City, Colombia. It was collected privately, therefore it is not available online. These data cover a time frame from January 10th, 2012 to May 31st, 2015, which corresponds to 1273 days (176 weeks) characterized by a daily average crime count of 19 thefts.

Formally, the database is composed by two sets: $C = \{c_1, \dots, c_L\}$ is the set of crime² events and $R = \{r_1, \dots, r_M\}$ which is the set of road intersections (or road nodes). Each crime event is reported as a triplet $\{D_q^c, X_q^c, Y_q^c\}$. For an event c_q , with $q = 1 \dots L$, D_q^c is its date, X_q^c is its horizontal coordinate in the cartographic system for Bogotá city and Y_q^c is its vertical coordinate. In the case of R , each road node r_s , with $s = 1 \dots M$, is reported as a duplet $\{X_s^r, Y_s^r\}$, where X_s^r and Y_s^r correspond to the geographic coordinates of the node in the same cartographic system of crime events.

2.2 Data Preprocessing

Spatio-Temporal Resolution: Spatio-temporal resolution for crime analysis and visualization is selected according to [16]. A crime mass is the counting of crime events in a given square areal unit (i.e. box) over a defined time interval. The signal of crime masses in Bogotá city corresponds to a multifractal process where information scaling remains constant over different spatial resolutions as the time scale increases. In fact, the informational self-similarity of this signal is preserved for a spatial resolution $\delta_{xy} \geq 500 \times 500 \text{ m}^2$ over a weekly temporal scale.

¹ The database was provided by *Fundación Ideas para la Paz*.

² The words *theft* and *crime* are used indistinctly throughout the document.

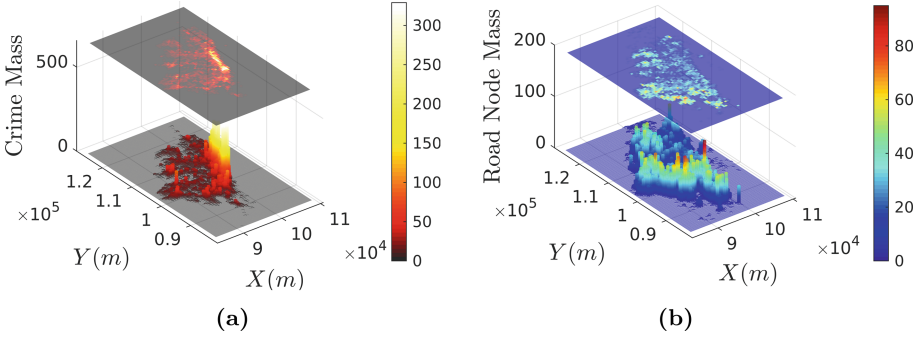


Fig. 1. Data visualization at a spatial resolution of $500 \times 500 \text{ m}^2$: (a) aggregate crime masses for 1273 days, (b) road-node masses.

Data Exploration: Figure 1a presents aggregate daily crime masses for 1237 days computed over $500 \times 500 \text{ m}^2$ boxes. Road-node masses computed with the same spatial resolution are depicted in Fig. 1b. Regarding theft masses it can be noticed that most of the historical activity concentrates in very few regions, which configure strong hotspots. Also note the majority of boxes exhibit little to no theft events. On the other hand, boxes with significant road-node masses are frequent across the study area.

Input Volume Generation: Data selection and data representation are very important criteria to feed models correctly. Thus, the sets C and R are transformed in such a way that can be used to feed in a convolutional neural network. In fact, data are represented as a real-valued tensor \mathbf{T} of order D such that $\mathbf{T} \in \mathbb{R}^{A_1 \times \dots \times A_D}$, where A_d corresponds the d -th direction of the input tensor.

Input data are set up as a three dimensional volume, as shown in Fig. 2a. The input volume is assembled by stacking bi-dimensional maps³. Each map has dimensions $(\Delta y \text{ bins} \times \Delta x \text{ bins})$, where $\Delta y \text{ bins}$ and $\Delta x \text{ bins}$ correspond to the number of boxes in the abscissa and ordinate directions respectively.

Available data are configured as an input volume \mathbf{T} composed of twelve bi-dimensional maps ($depth = 12$), as depicted in Fig. 2a and described in Table 1. More specifically, with one map of crime masses $E(k)$ where k corresponds to the time index, 8 maps of crime masses taken from the first order Moore neighborhood around mass box $e_{i,j}$ at time k ($N_1(k), \dots, N_8(k)$), one map of crime masses at previous time $E(k-1)$, one map with aggregate crime masses (i.e. crime masses history) H and one map with road-node masses RN . Then, the model architecture is fed by three-dimensional volumes and fetches out a bi-dimensional crime masses map $\hat{E}(k+1)$ for every given input $\mathbf{T}(k)$, which represents the model prediction of $E(k+1)$, as shown in Fig. 2b.

The content of each input channel is inspired based upon dynamic and static features in urban areas commonly found in the context of environmental

³ Each bi-dimensional map corresponds to a single channel from the input volume \mathbf{T} .

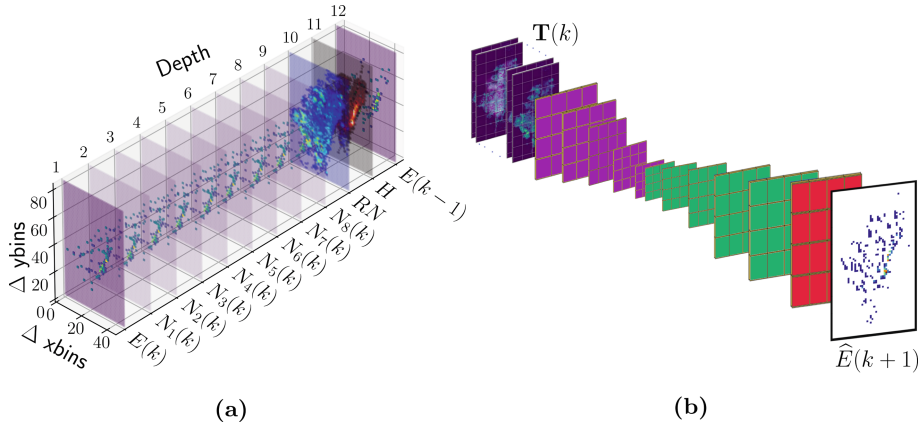


Fig. 2. Organization of training data (a) Input volume composed by 12 channels. (b) Input to output data processing.

Table 1. Channels used to form Input volume \mathbf{T} .

Description	Time dependent		Description	Time invariant	
	Map (channel)	Elements		Map (channel)	Elements
Crime masses at time k	$E(k)$	$e_{i,j}(k)$	Crime masses history	H	$H = \sum_{k=1}^{T_{end}} E(k)$
Moore Neighborhood at time k	$N_1(k)$	$e_{i-1,j-1}(k)$	Road node masses	RN	$rn_{i,j}$
	$N_2(k)$	$e_{i,j-1}(k)$			
	$N_3(k)$	$e_{i+1,j-1}(k)$			
	$N_4(k)$	$e_{i-1,j}(k)$			
	$N_5(k)$	$e_{i+1,j}(k)$			
	$N_6(k)$	$e_{i-1,j+1}(k)$			
	$N_7(k)$	$e_{i,j+1}(k)$			
	$N_8(k)$	$e_{i+1,j+1}(k)$			
Crime masses at time $k - 1$	$E(k - 1)$	$e_{i,j}(k - 1)$			

criminology [12]. Dynamic features are related to crime distribution at the neighborhood level and they correspond to channels $E(k)$, $N_1(k)$, $N_2(k)$, \dots , $N_8(k)$ in the input volume \mathbf{T} . Equally important, the temporal dependence is taken into account by adding the input channel with the map of crime masses for the immediately previous time $E(k - 1)$. On the other hand, static features are those with almost zero time dependency. In this case, the road-node masses map channel RN represents the geographical canvas scenario where crime phenomena take place. In addition, crime masses history H characterizes the past of criminal activity in the city. It can also be interpreted as the aggregate memory of the phenomenon that provides spatial information from a coarse temporal scale.

2.3 Model Architecture

Crime features exploration started with state of the art deep neural network architectures like LeNet, AlexNet, VGGNet moving forward with encoder decoder deep convolutional neural networks [17]. Then, a systematic iterative implementation over different architecture configurations led to a convolutional-deconvolutional architecture with an extra pooling layer on top of it. This architecture is depicted in Fig. 3a. Here, convolutional layers (in purple) match with deconvolutional layers (in green) in number and size. The pooling layer (in red) is plugged in on top of that architecture when predictions are required at different spatial resolutions. In addition, Rectified Linear Units [18] nonlinearities were interspersed with model's layers along the convolutional-deconvolutional architecture.

2.4 Experimental Set-Up

The number of filters f in convolutional layers was chosen to be $f = 2^p$, for $p = 1, 2, \dots, 10$, with an odd number of neurons per filter. Also, *valid* zero-pad and *one strided* convolutions were applied. In the same fashion, the up-sampling layers are based on transposed convolutions [19] with the same number of filters as in convolutional layers. Then, followed by a pooling layer on top of the architecture with a zero-strided 2×2 max-pooling operation.

Experiments were scheduled in a parallel infrastructure with Intel(R) Xeon(R) CPU's E31225 @ 3.10 GHz and Nvidia Quadro P1000 GPU's running experiments in TensorFlow 1.3.0 [20]. The reason behind this approach was to take advantage of the fast data flow offered in online grid computing to distribute tasks that involve the computation of matrix-matrix and matrix-vector operations with data processing in the margin of big data. Hence, this implementation simplified the computation graph related to the analytic gradient computation during the model training iterations.

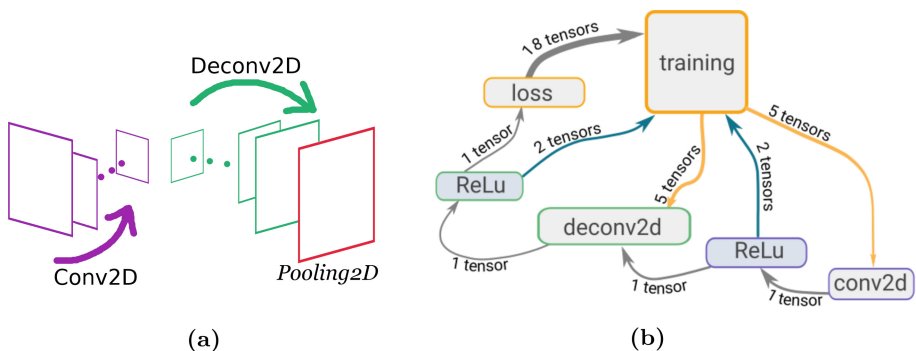


Fig. 3. Architecture and training: (a) Convolutional (purple) - Deconvolutional (green) - Pooling (red) architecture. (b) Data flow during training. (Color figure online)

Operations and data flow during the training process were programmed as a computational graph (See Fig. 3b). Here the model output is fetched by streaming the input data through the nodes that represent model’s architecture during the forward pass. Then, the loss function is computed and the analytical gradient is sent backwards to update model’s trainable neurons. The matrix-matrix and matrix-vector operations are computed in parallel as indicated by the training algorithm.

2.5 Training Strategy

An adaptive moment estimation algorithm called Adam [21] was used during the learning process. It computes individual learning rates for different parameters from estimates of first and second moments of the gradient. Thus, gradient estimation, bias correction of moments, update of moments and parameters update were computed as presented in Table 2. In this case, the training algorithm hyper parameters values were: learning rate $\alpha = 1e^{-3}$, first gradient moment coefficient $\beta_1 = 0.9$, second gradient moment coefficient $\beta_2 = 0.999$ and avoiding zero-division coefficient $\varepsilon = 1e^{-7}$.

In regard of used data, weekly input volumes $\mathbf{T}(k)$ were generated, where $k = 1..12$ for training and $k = 13..16$ for validation. These weeks correspond to the last four months of the database. In addition, H was configured as the aggregation of crime masses of the other 160 weeks.

Training was carried out with a loss function $L(\hat{E}(k+1), E(k+1))$ selected as the Mean Square Error, where $\hat{E}(k+1) = f(\mathbf{T}(k), W)$ and W is the weight matrix of the network. In order to overcome the stochasticity of random initialization, 33 independent runs were scheduled during 500 epochs allowing the model to overfit the training data. The intuition behind this process consists on picking the best model, saving its parameters values at every single epoch and then evaluating its performance at different training steps. Hence, the best model was reserved for further consideration during the model assessment stage.

Table 2. Learning algorithm updates and parameters

Learning algorithm step	Action
Gradient estimation	$dx_t = \nabla f(x_t)$
1 st Moment	$F_{t+1} = \beta_1 F_t + (1 - \beta_1) dx_t$
1 st Moment Bias Correction	$F_{t+1}^{corrected} = \frac{F_{t+1}}{1 - \beta_1^t}$
2 nd Moment	$S_{t+1} = \beta_2 S_t + (1 - \beta_2) [dx_t]^2$
2 nd Moment Bias Correction	$S_{t+1}^{corrected} = \frac{S_{t+1}}{1 - \beta_2^t}$
Parameters update	$x_{t+1} = x_t - \alpha \frac{F_{t+1}^{corrected}}{\sqrt{S_{t+1}^{corrected} + \varepsilon}}$
Hyperparameters	$\alpha, \beta_1, \beta_2, \varepsilon$

2.6 Deep Learning Architecture for Crime Forecasting

In order to assess the model performance, four statistics were chosen. As per [22]: “No one measure is universally best for all accuracy assessment objectives, and different accuracy measures may lead to conflicting conclusions because the measures do not represent accuracy in the same tray”.

Figure 4 shows a comparison between the model output and actual data. For a $500 \times 500 \text{ m}^2$ resolution there are 4080 values in these maps with more than 3000 zero boxes (i.e. no relevant values) and just tens of non zeros boxes (i.e. boxes with crimes or changes in crime counting). Statistics used to assess the model output have been chosen with the class-imbalanced data set challenge in mind (Precision, Recall and F1 score) [23,24].

While accuracy, as measured by quantitative errors, is important, it may be more crucial to accurately forecast the direction of change of crucial variables [25]. In particular, crime masses directional accuracy can be used in a binary evaluation fashion. Thus, either increase or decrease of crime predictions were considered as upward (1 if $e_{i,j}(k+1) > e_{i,j}(k)$) or downward (-1 otherwise) disregarding its quantitative values.

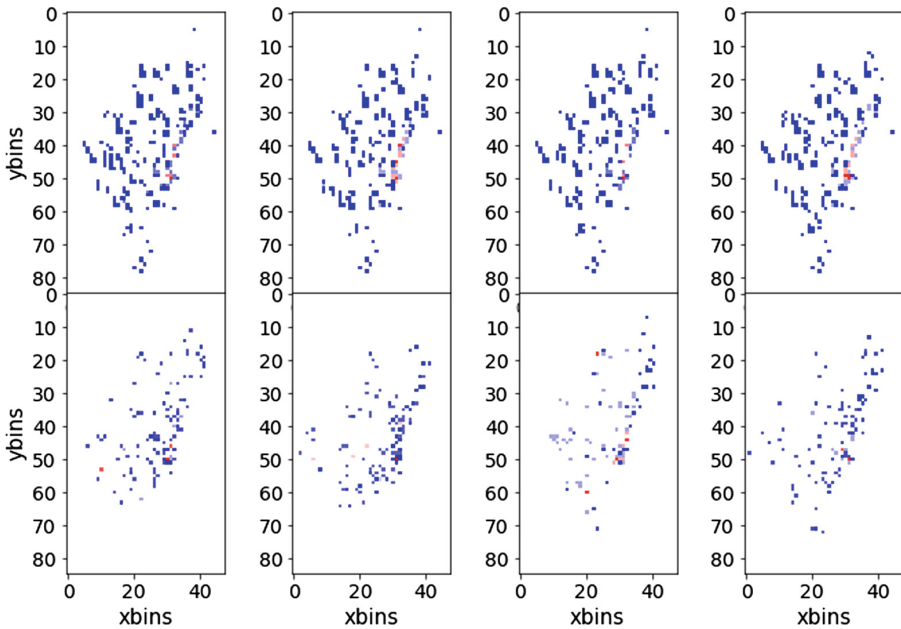


Fig. 4. Comparison between predicted and actual crime masses. Upper row: model output, lower row: actual data. The model outputs crime masses maps where the majority of predictions fall into the region surrounding crime hotspots during the four weeks of validation. Note expected output crime maps are class-imbalanced.

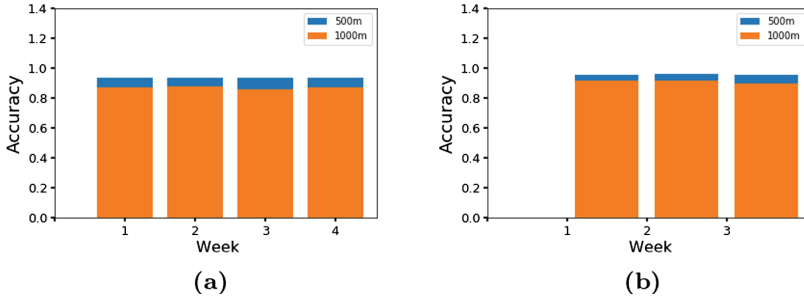


Fig. 5. Results for the four validation weeks: (a) model accuracy, (b) model directional accuracy.

3 Results

Accuracy results are presented in Fig. 5 for the four validation weeks. A high level of accuracy was obtained for two spatial resolutions $\delta_{xy} = 500 \times 500 \text{ m}^2$ and $\delta_{xy} = 1000 \times 1000 \text{ m}^2$. However, this statistic is not reliable given the data imbalance. This can be observed through the $B_R = (B_{nz}/B_z)$ ratio, where B_z corresponds to the number of Zero Crime Boxes and B_{nz} is the number of Non-zero Crime boxes. The average B_R over the four validation weeks is approximately 80/4080 for the former resolution and 80/924 for the latter. Therefore when the model reports a mass of zero crimes there is a high probability that its prediction falls in the zero crime region.

The main interest for crime mass predictions is in regions where crime activity occurs. Thus, precision and recall statistics averaged over the four validation weeks were introduced as presented in Table 3. In the case of precision of crime mass prediction and precision of directional accuracy the model presents better results at $1000 \times 1000 \text{ m}^2$ compared with those at $500 \times 500 \text{ m}^2$. Note that precision values were very poor in both cases. Implying that the model's perception about crime occurrence is not reliable. This problem might be solved including a loss function that focus in more local performance at the neighborhood level of crime occurrences during the learning process. Regarding recall of crimes, the model shows better results for the coarser resolution for all weeks whereas in the case of recall of tendency of crime occurrences it is for most weeks higher at the finer resolution, which means that the model is not good when stating that at certain locations are going to be crimes.

Table 3. Additional results for the four validation weeks in percentage.

Resolution	Crime masses						Directional accuracy					
	Precision (%)		Recall (%)		F1 score (%)		Precision (%)		Recall (%)		F1 score (%)	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
$500 \times 500 \text{ m}^2$	12.87	0.07	31.51	3.40	18.23	0.56	9.55	1.64	14.71	3.32	11.57	2.22
$1000 \times 1000 \text{ m}^2$	24.30	3.62	36.91	2.70	29.22	3.21	17.91	3.27	17.01	2.07	17.43	2.62

In addition, F1 score was introduced to consider the trade-off over results between precision and recall, as well as the database imbalance for model assessment. In this case, the F1 scores reported in Table 3 along with the visual results (see Fig. 6) between the perception of the model $\hat{E}(k+1)$ and the ground truth map $E(k+1)$ favor to understand that the trained model is hitting a very low portion of rare crimes developing at multiple locations in the city. It may be improved by using filters of different shapes. The intuition behind this is that given the multi-scale nature of the crime masses $E(t)$, the usage of filters of multiple sizes in a similar fashion to *Inception Modules* [26] will allow the top level layers to squeeze out information from each region involving dynamics at different spatial resolutions as the filters raster across the maps. In other words, it may be helpful to capture better not only the dynamics at hot spot zones but also to hit rare crimes that distribute across the urban area.

Results show that evaluating the model output at higher resolutions than the one it was trained for increases its performance in most statistics. In fact, the low precision and recall values came out because the model does not predict exactly the same (X, Y) coordinates for most of expected crime masses but in the boxes located around the actual position. Therefore the model prediction

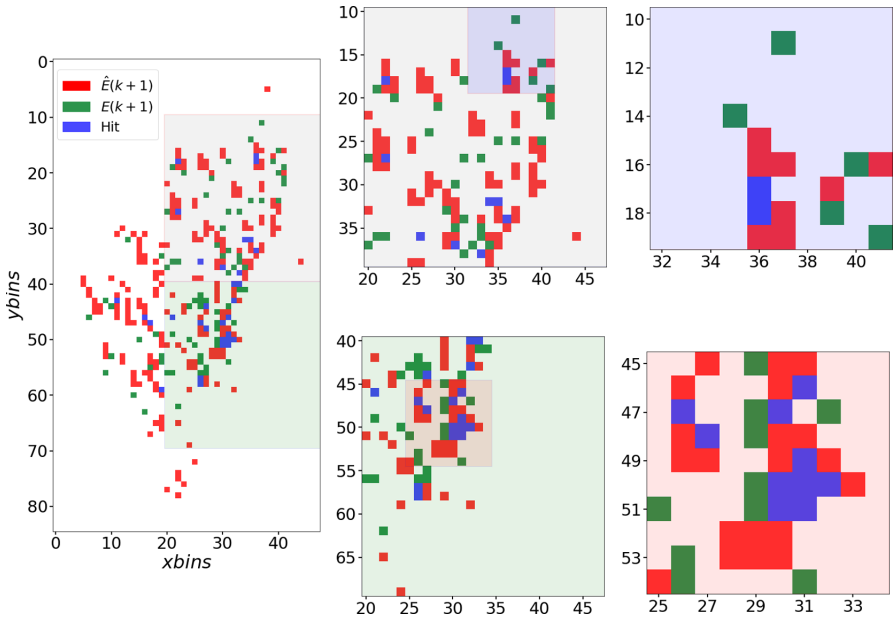


Fig. 6. Left figure depicts an example of the model output (red boxes), expected output (green boxes) and their intersection (blue boxes). It is shown in upper right figures that even if the model predictions do not match the expected values, their positions are close to actual crime boxes in a radius ε_{xy} at a local neighborhood level. In lower right figures, it can be noticed that the model identifies crime hotspots zones in the city. (Color figure online)

capacity will improve as it will be better to predict true positive values at a spatial resolution very close to the one it was trained for. This may be achieved by using the tendency of crimes as expected output during the training stage instead and batch normalization [27] for cases with a very deep architecture setup.

On the other hand this model has also some good advantages. Even if it might not predict the exact (X, Y) coordinates of the expected crime mass, actual criminal activity is likely to happen in boxes located in a small radius $(X \pm \varepsilon_x, Y \pm \varepsilon_y) > (0, 0)$ around predicted $\hat{e}_{i,j}(k+1)$ in the neighborhood region as shown in Fig. 6. Note that the model is able to identify crime hotspots regions. Similarly, this network architecture allows the designer to gain intuition about the kernels that might be used to extract features in correlation operations with the incoming inputs at convolutional and deconvolutional layers. In addition, this model has a very reduced number of parameters when compared with a traditional deep convolutional network where the number of parameters is at the order of millions while the explored architecture has a maximum of $3 \times 3 \times 2^5$ parameters in the biggest configuration.

4 Conclusions

A proposal of an architecture for urban crime forecasting based on convolutional - deconvolutional deep neural networks was presented. The architecture allows to predict crime masses at a resolution of $500 \times 500 \text{ m}^2$ in a weekly scale. Another advantage of this architecture is that it emphasizes its predictions on the hot spot zones, hence it would be convenient for segmenting massive crime regions.

Among the upcoming improvements of the architecture are: increasing the number of model parameters by layer, going very deep in terms of number of layers without including fully connected layers, testing additional inputs in the framework of Risk Terrain Modeling [28]. Moreover, given the multi-scale nature of the input signal [16], testing with filters of different shapes at each layer may contribute to capture the signal texture at different resolutions.

References

1. Piza, E.L., Gilchrist, A.M.: *J. Crim. Justice* **54**, 76 (2018). <https://doi.org/10.1016/j.jcrimjus.2017.12.007>
2. Brelsford, C., Martin, T., Hand, J., Bettencourt, L.M.A.: *Sci. Adv.* **4**(8), eaar4644 (2018). <https://doi.org/10.1126/sciadv.aar4644>
3. Ratcliffe, J.: Crime mapping: spatial and temporal challenges. In: Piquero, A., Weisburd, D. (eds.) *Handbook of Quantitative Criminology*, pp. 5–24. Springer, New York (2010). https://doi.org/10.1007/978-0-387-77650-7_2
4. King, T.C., Aggarwal, N., Taddeo, M., Floridi, L.: *Sci. Eng. Ethics* (2019). <https://doi.org/10.1007/s11948-018-00081-0>
5. Mohler, G., Brantingham, P.J., Carter, J., Short, M.B.: *J. Quant. Criminol.* (2019). <https://doi.org/10.1007/s10940-019-09404-1>

6. Stalidis, P., Semertzidis, T., Daras, P.: Examining deep learning architectures for crime classification and prediction (2018)
7. Brantingham, P.J., Valasik, M., Mohler, G.O.: *Stat. Public Policy* **5**(1), 1 (2018). <https://doi.org/10.1080/2330443X.2018.1438940>
8. Nobles, M.R.: *Am. J. Crim. Justice* (2019). <https://doi.org/10.1007/s12103-019-09483-7>
9. Aaltonen, M., Oksanen, A., Kivivuori, J.: *Criminology* **54**(2), 307 (2016). <https://doi.org/10.1111/1745-9125.12103>
10. Rumi, S.K., Deng, K., Salim, F.D.: *EPJ Data Sci.* **7**(1), 43 (2018). <https://doi.org/10.1140/epjds/s13688-018-0171-7>
11. Blattman, C., Green, D., Ortega, D., Tobón, S.: Hotspot interventions at scale: the effects of policing and city services on crime in Bogotá, Colombia. Technical report, International Initiative for Impact Evaluation (3ie) (2018). <https://doi.org/10.23846/DPW1IE88>
12. Bruinsma, G.J.N., Johnson, S.D.: *The Oxford Handbook of Environmental Criminology*. Oxford University Press, Oxford (2018). Google-Books-ID: qPdJD-wAAQBAJ
13. LeCun, Y., Bengio, Y., Hinton, G.: *Nature* **521**, 436 (2015)
14. Adamson, G., Havens, J.C., Chatila, R.: *Proc. IEEE* **107**(3), 518 (2019). <https://doi.org/10.1109/JPROC.2018.2884923>
15. Quick, M., Li, G., Brunton-Smith, I.: *J. Crim. Justice* **58**, 22 (2018). <https://doi.org/10.1016/j.jcrimjus.2018.06.003>
16. Melgarejo, M., Obregon, N.: *Entropy* **20**, 11 (2018). <https://doi.org/10.3390/e20110874>. <http://www.mdpi.com/1099-4300/20/11/874>
17. Su, J., Vargas, D.V., Sakurai, K.: *IPSN Trans. Comput. Vis. Appl.* **11**(1), 1 (2019). <https://doi.org/10.1186/s41074-019-0053-3>
18. Macêdo, D., Zanchettin, C., Oliveira, A., Ludermir, T.: *Expert Syst. Appl.* **124**, 271 (2019). <https://doi.org/10.1016/j.eswa.2019.01.066>
19. Gao, H., Yuan, H., Wang, Z., Ji, S.: *IEEE Trans. Pattern Anal. Mach. Intell.* **1** (2019). <https://doi.org/10.1109/TPAMI.2019.2893965>
20. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al.: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI 2016*, pp. 265–283. USENIX Association, Berkeley (2016)
21. Kingma, D.P., Ba, J.: [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) [cs] (2014)
22. Stehman, S.V.: *Remote Sens. Environ.* **62**(1), 77 (1997). [https://doi.org/10.1016/S0034-4257\(97\)00083-7](https://doi.org/10.1016/S0034-4257(97)00083-7)
23. Tharwat, A.: *Appl. Comput. Inform.* (2018). <https://doi.org/10.1016/j.aci.2018.08.003>
24. Wang, X., Jiang, X.: *Sig. Process.* **165**, 104 (2019). <https://doi.org/10.1016/j.sigpro.2019.06.018>
25. Pierdzioch, C., Reid, M.B., Gupta, R.: *J. Appl. Stat.* **45**(5), 884 (2018). <https://doi.org/10.1080/02664763.2017.1322556>
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2016). [arXiv:1512.00567](https://arxiv.org/abs/1512.00567)
27. Wu, S., et al.: *IEEE Trans. Neural Netw. Learn. Syst.* 1–9 (2018). <https://doi.org/10.1109/TNNLS.2018.2876179>
28. Caplan, J.M., Kennedy, L.W. (eds.): *Risk Terrain Modeling Compendium for Crime Analysis*, Rutgers Center on Public Security (2011)