EasyChair Preprint

№ 5461

A Performance Evaluation of 3D Deep Learning Algorithms for Crime Classification

Tawanda Matereke, Clement Nyirenda and Mehrdad Ghaziasgar

May 4, 2021

A Performance Evaluation of 3D Deep Learning Algorithms for Crime Classification

Tawanda Matereke
Department of Computer Science
University of Western Cape
Cape Town, South Africa
lloydmatereke23@gmail.com

Clement N. Nyirenda
Department of Computer Science
University of Western Cape
Cape Town, South Africa
cnyirenda@uwc.ac.za

Mehrdad Ghaziasgar
Department of Computer Science
University of Western Cape
Cape Town, South Africa
mghaziasgar@uwc.ac.za

***Abstract*—This paper presents a study on crime classification using two 3D deep learning algorithms, i.e. 3D Convolutional Neural Network and the 3D Residual Network. The Chicago crime dataset, which has records from 2001 to 2020, with a record count of 7.29 million records, is used for training the models. The models are evaluated by using F1 score, Area Under Receiver Operator Curve (AUROC), and Area Under Curve - Precision Recall (AUCPR). Furthermore, the effectiveness of spatial grid resolutions on the performance of the models is also evaluated. Results show that the 3D ResNet achieved the best performance with a F1 score of 0.9985, whereas the 3D CNN achieved a F1 score of 0.9979, when training on a spatial resolution of 16 pixels. In terms of future work, we would want to test these algorithms on multi label classifaction and regression crime problems, also we want to improve the performance of the 3D CNN by adding RNN layers and evaluate an implementation of 3D ResNeXt for crime prediction and classification.**

***Index Terms*—Crime classification, 3D Deep learning, 3D CNN, 3D ResNet, Spatio Temporal, Sparsity.**

I. Introduction

According to the United Nations [1] crime is studied in order to prevent it. Crime prevention is thus at the heart of criminology endeavour. Crime prevention is a systematic approach for finding crime patterns and trends. To this end, crime classification and prediction is essential because it speeds up the process of solving crimes and reduces crime rates [2].

Deep learning, a cutting-edge technology for automatic feature identification via a deep neural network (DNN), gives state-of-the-art performance on many predictive scenarios, such as image classification, computer vision, speech recognition [3]. Recent studies show that deep learning techniques have been applied to spatio-temporal data. These techniques include the Recurrent Neural Networks (RNN), which are a superior in mining temporal dependencies, and show high accurate prediction of sequential data [4]. Another technique is the Convolutional Neural Networks (CNN), which are superior at mining spatial features and have shown very high accuracy in various domains including computer vision [5].

The most frequently used framework for spatio-temporal forecasting is a combination of 2D CNNs and RNNs [6]. The 2D CNN is typically used for learning the spatial traits and the RNN for learning the temporal features [7]. These two algorithms can be stacked together in multiple contiguous layers to improve prediction performance. The arrangement of these layers differs in various architectures such as the Spatio Temporal Residual Network (ST-ResNet) [7], Deep Multi View Spatio Temporal Network (DMVST-Net) [10], Spatio Temporal Dynamic Network (STD-Net) and the Recurrent Convolutional Network (RCNN) [9], amongst others.

In this paper, we depart from 2D architectures and focus our attention on 3D deep learning algorithms, i.e. 3D CNN and 3D ResNet, for crime classification. We also propose implementation procedures. We trained the algorithms using the Chicago crime dataset. We focused on classifying the "Theft" crime category. Our goal was to model a binary classification problem thereby predicting if "Theft" will occur or not on a future date. We used a temporal window of 30 days for training and we used spatial features of that dataset to create incident maps. This led us to creating incident maps with a 30 day window period for all theft incidents in the crime dataset. Our contributions are:

- We propose a guide for implementing the 3D CNN and 3D ResNet for crime classification.
- We compare the performance of these algorithms.
- We evaluate the effect of spatial resolution in the crime dataset on crime classification.

The rest of the paper is organized as follows: Section II presents the related work on crime prediction and classification using deep learning techniques. Section III presents the fundamental concepts of 3D deep learning algorithms as well their different architectures. Section IV discusses the implementation procedures of the 3D deep learning algorithms, the training dataset, the data prepossessing, and the metrics used in the experiments. Section V presents and discusses the results in different configurations using various performance metrics. Lastly, we conclude the paper in Section VI.

II. Related Work

This section discusses the related work for crime classification using spatio temporal deep learning algorithms and 3D algorithms for spatio temporal problems such as

video processing.

Zhang, Zheng and Qi [7] applied the ST-ResNet to Citywide crowd flow prediction. They used the Beijing taxi cabs trajectories and meteorological data, and New York City bike trajectory data. In the data prepossessing procedure, min-max normalization was used to scale the data, and one-hot coding to transform metadata (i.e., DayOfWeek, Weekend/Weekday), holidays, and weather conditions into a binary vector. Root Mean Square Error (RMSE) was used for model evaluation. The ST-ResNet achieved better performance than Historical Average (HA), Auto-regressive Moving Average (ARIMA), Seasonal Auto-regressive Moving Average (SARIMA), Vector Auto-regression (VAR), Spatio Temporal Artificial Neural Network (ST-ANN), and Deep Spatio Temporal Neural Network (Deep ST).

Wang et al. [6] carried out real-time crime forecasting on an hourly timescale by applying an ST-ResNet model. They considered all types of crime in Los Angeles (LA) over the last six months of 2015. In total there were 104,957 crimes. Due to the low regularity of the crime data in both space and time, both spatial and temporal regularization of the data was performed. They compared two similar DNN structures except that one had CNN layers (the ST-ResNet). RMSE was used to evaluate the accuracy of the models. The ST-ResNet achieved the best results with an average accuracy of 84.78% in the top 25 predictions. Wang et al. [6] continued experiments on real-time crime forecasting on the LA crime dataset using the ST-ResNet model and proposed some improvements in terms of the model's accuracy and performance on mobile devices. They included weather data in their experiments. RMSE was used for model evaluation. The ST-Resnet was compared with HA, KNN, ARIMA models. The ST-ResNet produced the best results with a low error in crime density of 0.659.

Stalidis et al. [8] demonstrated that deep learning-based methods outperform the traditional methods on crime classification and prediction. They carried out an evaluation of the effectiveness of different parameters in the deep learning architectures. They gave insights for configuring them in order to achieve improved performance in crime classification and finally crime prediction. They used five different datasets. These 5 datasets were incident reports from Seattle, Minneapolis, Philadelphia, San Francisco, and Metropolitan DC police departments. Ten algorithms were compared namely; CCRBoost, ST-Resnet, Decision Trees, Naive Bayes, LogitBoost, Random Forests, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Multi Layer Preceprtron (MLP). F1 score, Area Under Reciever Operater Chracteritic (AU ROC), and PAI (Prediction Accuracy Index) were used to evaluate the model's performance. From the experiment, the ST-ResNet was found dominating other methods.

Stec and Klabjan [9] used a joint RCNN for the purpose of predicting crime. Chicago and Portland crime datasets were used for the experiments. They combined crime data with additional weather, public transportation, and census data. They conducted experiments to determine the best network structure from among the following: the Feed Forward Network, CNN, RNN, and RCNN. Mean Absolute Scaled Error (MASE) was used for model evaluation. The RCNN exhibited the best accuracy results on both Chicago (with an accuracy of 75.6%) and Portland (with an accuracy of 65.3%) datasets.

Yao et al. [10] proposed a DMVST-Net framework to model both spatial and temporal relations. The model consisted of three views: temporal view (modeling correlations between future demand values with near time points via LSTM), spatial view (modeling local spatial correlation via local CNN), and semantic view (modeling correlations among regions sharing similar temporal patterns). They used a large-scale online taxi request dataset collected from Didi Chuxing, which is one of the largest online car-hailing companies in China. Mean Average Percentage Error (MAPE) and Rooted Mean Square Error (RMSE) for model evaluation. The proposed model was compared to HA, ARIMA, Linear regression, MLP, XGBoost, and ST-ResNet. The DMVST-Net achieved the best results with MAPE - 0.1616 and RMSE - 9.642.

Ali et al. [11] proposed a deep hybrid neural network composed of recurrent and convolutional networks to predict citywide traffic crowd flows by leveraging Spatio-temporal patterns. They used the TaxiBj and BikeNYC datasets which were normalized using the Min-Max normalization technique. Root Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE) techniques were used to evaluate the model performances. They compared the performance of their proposed model to that of HA, ARIMA, LinUOTD, XGBoost, MLP, ConvLSTM, STDN, and ST-ResNet.

Zhang et al. [12] proposed a model for attention-based supply demand for autonomous vehicles. The dataset they used was obtained from an online car-hailing company in China. The Mean Average Percentage Error (MAPE), Mean Absolute Error (MAE), and Rooted Mean Square Error (RMSE) techniques were used for evaluating model performance. They compared the performance of their model to that of ARIMA, LSTM, ConvLSTM, Reduced-ConvLSTM, ST-ResNet, DMVST-Net, STDN, and Reduced-STDN, baseline models. Amongst the baseline models, the STDN achieved the best results with MAPE - 21.08%, RMSE - 0.1634, and MAE- 0.1348. The STDN achieved the best results as compared to other baseline models that were tested.

It has recently been reported [13] that other than the popular combination of the 2D CNN and the RNN, the 3D CNN alone is capable of extracting both the temporal and spatial features. Moreover, adapting the 3D CNN to the ResNet

architecture, i.e. 3D ResNet, can improve the performance in predictions [14]. Ji et al. [13] proposed 3D CNN for automated human action recognition in surveillance videos. They used the TRECVID 2008 development dataset consists of 49-hour videos captured at London Gatwick Airport. Precision, Recall and AUC were used to evaluate the model. The 3D CNN outperformed the 2D CNN. On the other hand, Zunair et al. [20], proposed using 3D CNNs for tuberculosis prediction. CT scans were used as the training data. The results were tested against the ImageCLEF Tuberculosis Severity Assessment 2019 benchmark. They reported 73% AUC and binary classification accuracy of 67.5% on the test set outperforming all methods which leveraged only image information. Furthermore, Hara et al. [14] proposed learning spatio-temporal features with 3D residual networks for action recognition. They used the ActivityNet and Kinetics datasets training. They used accuracy as the metric for evaluation. The 3D ResNet outperformed the C3D and ImageNet algorithms.

III. 3D Deep learning Algorithms

In this section discuss about the fundamental concepts of the 3D CNN and the 3D ResNet. We also give a detailed description on the the architectural design we used for the crime classification.

A. 3D Convolutional Neural Network

1) 3D Convolution: The 3D convolution is accomplished by convolving a 3 dimensional kernel to the cube formed by assembling numerous adjacent frames together. With this course of action, the feature maps in the convolution layer are associated with various adjacent frames in the past layer, thereby capturing temporal information. Formally, the feature map resulting 3D convolution can be represented with the equation below:

$$v_{ij}^{xyz} = tanh\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \quad (1)$$

The equation below calculates the value at position (*x,y,z*) on the *j*th feature map in the *i*th layer. R_i represents the size for the 3D kernel along the temporal dimension, and w_{ijm}^{pqr} is the (*p, q, r*)th value of the kernel connected to the *m*th feature map in the previous layer. Fig. 1 shows an illustration of a 3D convolution [13].

2) 3D CNN Architecture: Based on the above described 3D convolution, a variety of CNN architectures can be devised. In the following, we describe a 3D CNN architecture that we have developed for crime classification on the Chicago dataset. Fig. 2 shows an overview of the architecture we developed.

We supplied a 5 dimensional input map with shape (*batch size, sequence length, latitude, longitude, crime types*). We apply batch normalization to make the learning process more steady and reduce the number of epochs required to train the network. We also added dropout layers which randomly set input units to zero with a frequency of rate at each epoch

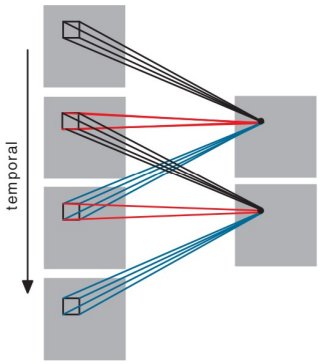

Fig. 1: 3D convolution [13].

during training, this helps prevent over fitting, thus, enabling the network to generalize better. We added 3D max-pooling layers to reduce the number of parameters that the network will learn, and also the computation it will perform. Max-pooling layers can achieve this by reducing the dimensions in feature maps generated by the convolution layer. We used a kernel of size $3 \times 3 \times 3$, and a fixed filter size of 3. Each Convolution layer had an ReLU activation function. We finally flatten the network and add a dense layer with an output dimension of 1, and is configured with sigmoid activation function. This gives us a fully connected network.

B. 3D Residual Network

1) Network Architecture: The 3D ResNet [14] is based on the ResNet [15]. ResNets have skip connections that enable a signal to passed from one layer to the next. Skip connections are the identity shortcut connections that pass through the gradient flows of networks from later layers to early layers, and this lightens the training of very deep networks. Fig. 3 shows the residual block, which is an element of a ResNet. A ResNet can consist of multiple residual blocks. The skip connections pass a signal from the top of the block to the tail.

The 3D ResNet differs from the ResNet [15], in regards to dimensions as it performs 3D convolutions and 3D pooling. The size of the kernels are $3 \times 3 \times 3$, with a stride of 1. Similar to the 3D CNN described earlier, the input has 5 dimensions and has shape; (*batch size, sequence length, latitude, longitude, crime types*). We used the same configuration in the 3D CNN architecture for the convolution layers. Down sampling the inputs was performed with convolution layers with a stride of 2. We adopted identity shortcuts with zero-padding [15] to avoid increasing the number of parameters when the number of feature maps increase. Fig. 4 shows the overview of the network architecture.

IV. Implementation of 3D Algorithms

We implemented both the 3D CNN and 3D ResNet in python, using Keras [16] framework and Tensorflow [17] backend. We used training time and number of trainable parameter to compare the two algorithms complexity as shown in Table I. The experiments were performed on a hyper

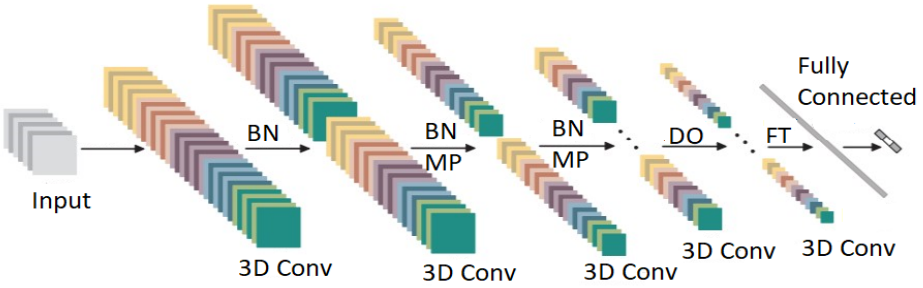

Fig. 2: 3D CNN Architecture. BN - Batch Normalization, MP - 3D Max Pooling, DO- Dropout, FT - Flatten

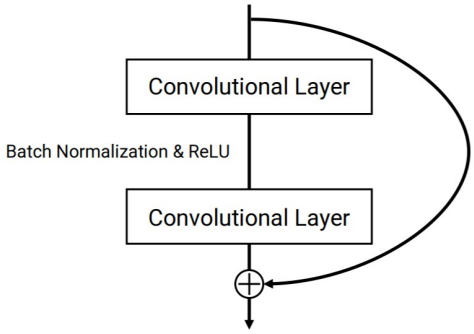

Fig. 3: Residual block. Skip connections pass a signal from the top of the block to the tail. Signals are summed at the tail.

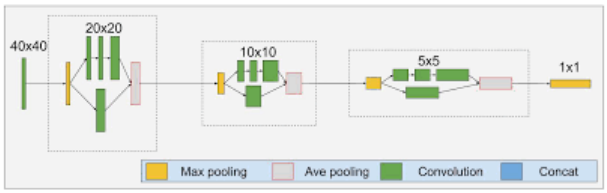

Fig. 4: 3D ResNet Architecture, adopted from ST-ResNet [8]

threaded linux virtual machine on Azure Cloud [18], with 8 virtual CPU's, 128GB RAM, and a processor speed of 2.3 GHz

TABLE I: Complexity of algorithms in total training time and number of trainable parameters

Algorithm	Training time	Number of parameters
3D CNN	01:34:23	215 509
3D ResNet	01:55:17	269 293

A. Procedures

In our experiments for both the 3D CNN and the 3D ResNet, the parameters were set as follows: The size of the convolution filters are fixed to 3×3. The number of epochs used were 10. The learning rate was set at a value of 0.01. Batch normalization was used. We adopted the ADAM optimizer to optimize the loss function. The length of the temporal sliding window was fixed at a value of at 30. We evaluated the effect of different spatial resolutions on the performance of both models for crime classification on the Chicago dataset. The resolutions that were tested vary from a minimum of $p = 16$ (i.e. 16×16 cells per grid) to a maximum of $p = 40$ cells per grid with a step of $p = 8$ cells.

B. Dataset

We obtained our crime dataset from the Chicago Data Portal. The dataset includes crime incident reports dating back to 2001, with 7.29 million records. Each report includes location information (in latitude and longitude), a time and a type of crime. There are 32 distinct crime types in the dataset. We selected records in the 3 year period from 20017 to 2020 for training our algorithms. Fig. 5 shows the different a sample of 10 crime types and their total occurrences.

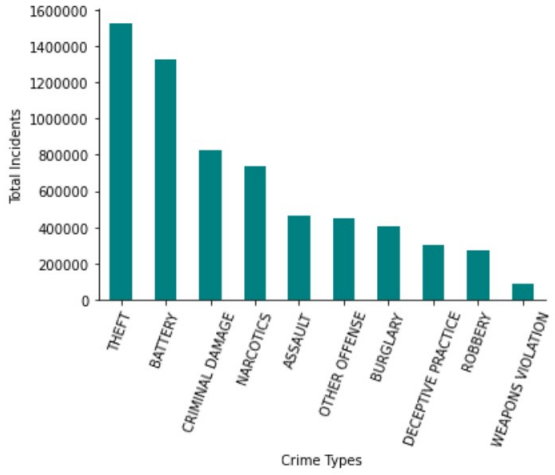

Fig. 5: Total incidents in Chicago crime dataset.

We decided to filter the dataset for a single crime type, i.e. "Theft", in order to suite a binary classification problem. All records with dates when "Theft" was reported were considered as the positive class, and the remaining records, when "Theft" was not recorded were considered as the negative class. We then, binary encoded the crime type, i.e. the positive class was encoded with a 1, and 0 for the negative class. Fig. 6 shows resulting dataset for "Theft" as the chosen positive class. We adopted the spatial grid resolution ranges that are used

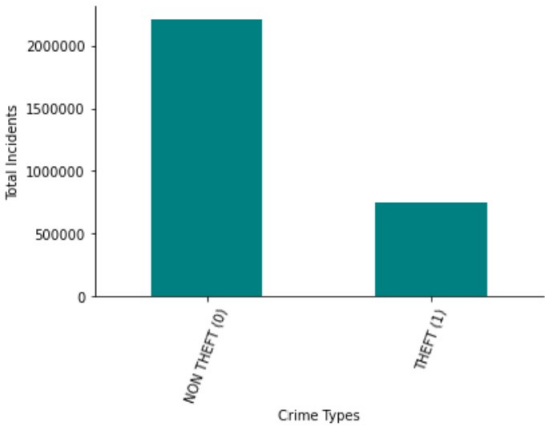

Fig. 6: Total Theft incidents in Chicago crime dataset.

in the related paper [19]. These ranges are 16×16, 24×24, 32×32, and 40×40 cells. We aim to predict whether "Theft" incidents will happen in a given area by evaluating past recorded incidents in the current month. Thus, the past crime incidents were grouped in incident maps *I* of timespan *t* of 1 day, and for a period *T* of 30 days. Therefore, 30 daily incident maps were used as input to forecast "Theft" incidents for the next period. We used a daily timespan for incidents so that enough temporal detail can be extracted while the time series are adequately populated.

C. Metrics

We decide not to use accuracy because we are dealing with imbalanced data. Instead we opted to use *F1 score* because it is the harmonic mean of precision and recall. Precision tells us how many, out of all instances that were predicted to belong to class *X*, actually belonged to class *X*, i.e. the fraction of relevant instances among the retrieved instances. Recall expresses how many instances of class *X* were predicted correctly. The *F1score* is calculated by:

$$F1score = 2 \times \frac{precision \times recall}{precision + recall} \tag{2}$$

The area under the receiver operating characteristic (*AUROC*) is a performance metric that is used to evaluate classification models. It is calculated as the area under the *ROC* curve. A *ROC* curve shows the trade-off between true positive rate (*TPR*) and false positive rate (*FPR*) across different decision thresholds. The *TPR* and *FPR* are defined as:

$$TPR = \frac{TP}{TP + FN} \tag{3}$$

$$FPR = \frac{FP}{FP + TN} \tag{4}$$

AUCPR (Area Under Curve - Precision Recall) equivalently, is the calculated area under a precision-recall curve. *Precision* and *Recall* are defined as:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

In the next section we present the performance of the 3D CNN and 3D ResNet using the metrics described above.

V. Results

In this section we discuss the experimental results based on the *F1score*, *AUCPR*, *AUROC*, and loss values during training. We also evaluate the effect spatial resolution on the 3D models performance.

Table II show results achieved training the algorithms over 5 epochs using 16×16 spatial resolution. The results show that the models perform better after each epoch and also that the 3D ResNet is the winning algorithm on each epoch.

TABLE II: Performance of 3D algorithms over 5 epochs using resolution of 16 pixels

Epoch	3D CNN			3D ResNet		
	F1 score	AUCPR	AUROC	F1 score	AUCPR	AUROC
1	0.4235	0.6226	0.4743	0.5585	0.7011	0.5700
2	0.7140	0.7947	0.7947	0.9387	0.9557	0.9202
3	0.8108	0.8663	0.7467	0.9661	0.9755	0.9554
4	0.8614	0.9017	0.8140	0.9532	0.9660	0.9404
5	0.8851	0.9167	0.8461	0.9700	0.9783	0.9612

We plotted the *F1 score* and loss of both of the models. Fig. 7, and 9 shows that the *F1 score* improves as the number of epochs increase for both models, however the 3D ResNet shows higher *F1 scores* than the 3D CNN network. Fig. 8, and 10 shows that the loss decreases as the training iterations continue, however the 3D ResNet shows a lesser loss value in each iteration of training. Therefore the 3D ResNet is able to interpret the data points better than the 3D CNN during training.

Table III show the best results in 10 epochs for different spatial resolution. Although increasing the number of cells makes the feature maps sparser it is evident from the results that this leads to a very slight deterioration in performance of both of the models.

TABLE III: Performance of 3D algorithms for using different spatial resolutions for classifying theft cases in Chicago dataset

Metric	Resolution	3D CNN	3D ResNet
F1 score	16	0.9979	0.9985
	24	0.9973	0.9970
	32	0.9939	0.9954
	40	0.9930	0.9950
AUCPR	16	0.9989	0.9989
	24	0.9982	0.9985
	32	0.9979	0.9982
	40	0.9941	0.9972
AUROC	16	0.9964	0.9956
	24	0.9955	0.9954
	32	0.9930	0.9933
	40	0.9911	0.9928

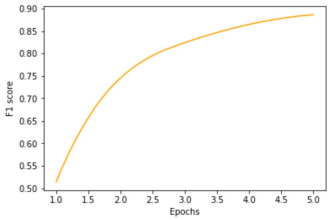

Fig. 7: 3D CNN F1 score

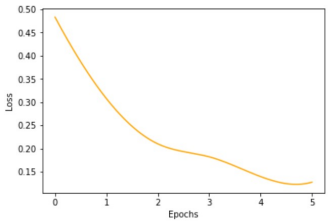

Fig. 8: 3D CNN Loss

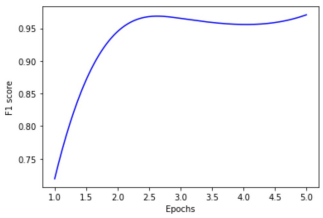

Fig. 9: 3D ResNet F1 score

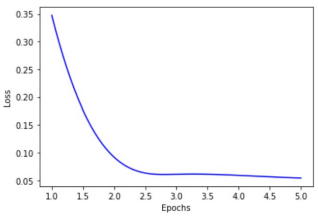

Fig. 10: 3D ResNet Loss

From the experimental results we are convinced that the 3D ResNet is the winning algorithm for the crime classification problem using the Chicago crime dataset.

VI. Conclusion

In this paper we investigated the capability of the 3D CNN and the 3D ResNet in classifying crime using the Chicago dataset. In order to achieve this goal we trained the deep learning methods with data only containing spatial and temporal information. We also evaluated the the effect of spatial resolution on the predictive performance of the models. Our main focus was on binary classification problem. Areas of future study we have noted are:

- Evaluating the 3D CNN and 3D ResNet for multi-label classification of crime.
- Combining the RNN and 3D CNN and to achieve better temporal feature extraction.
- Evaluating a variation of the ResNet, i.e. ResNeXt architecture and implementing it using 3D convolutions.

References

[1] The Commission on Crime Prevention and Criminal Justice. United Nations: Office on Drugs and Crime, //www.unodc.org/unodc/en/commissions/CCPCJ/index.html. Accessed 16 Mar. 2021.

[2] D.N. Varshitha et al., "Paper on Different Approaches for Crime Prediction system," InternationalJournal of Engineering Research and Technology (IJERT), 2017.

[3] Y. LeCun, Y. Bengion, and G. Hinton, "Deep learning," Nature, 521:436–444, 2015.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput, 9:1735–1780, 1997.

[5] H. Lee and H. Kwon, "Going Deeper With Contextual CNN for Hyperspectral Image Classification," in IEEE Transactions on Image Processing, vol. 26, no. 10, pp. 4843-4855, doi: 10.1109/TIP.2017.2725580, Oct. 2017.

[6] B. Wang, P. Yin, A.L. Bertozzi, BP.J. rantingham, S.J. Osher, and J. Xin, "Deep Learning for Real-Time Crime Forecasting and Its Ternarization," Chinese Annals of Mathematics, Series B, 40(6), 949-966.doi:10.1007/s11401-019-0168-y, 2019.

[7] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," AAAI, 2017.

[8] P. Stalidis, T. Semertzidis, and P. Daras, "Examining Deep Learning Architectures for Crime Classification and Prediction," arXiv:1812.00602, 2018.

[9] A. Stec, and D. Klabjan, "Forecasting Crime with Deep Learning," arXiv e-prints, 2018.

[10] H. Yao et al, "Deep Multi-view Spatial-Temporal Network for Taxi Demand Prediction," in Proc. Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, pp. 2588-2595, Feb. 2018

[11] A. Ali, Y. Zhu, Q. Chen, J. Yu and H. Cai, "Leveraging Spatio-Temporal Patterns for Predicting Citywide Traffic Crowd Flows Using Deep Hybrid Neural Networks," 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, pp. 125-132, doi: 10.1109/ICPADS47876.2019.00025, 2019.

[12] Z. Zhang, H. Dong, Y. Li, Y. You, and F. Zhao, "Attention-Based Supply-Demand Prediction for Autonomous Vehicles," In the 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT). doi:10.1109/pdcat46702.2019.00085, 2019.

[13] S. Ji, W. Xu, M. Yang and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 221-231, doi: 10.1109/TPAMI.2012.59, Jan. 2013.

[14] K. Hara, H. Kataoka, and Y. Satoh, "Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition," IEEE International Conference on Computer Vision Workshops (ICCVW). doi:10.1109/iccvw.2017.373, 2017.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.

[16] F. Chollet et al. Keras. https://github.com/kerasteam/keras. 2015.

[17] M. Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org. URL: https : / / www.tensorflow.org/

[18] M. Copeland, J. Soh, A. Puca, M. Manning, and D. Gollob, "Microsoft Azure and Cloud Computing," In Microsoft Azure. Apress, Berkeley, CA, 2015.

[19] C-H. Yu et al. "Hierarchical Spatio-Temporal Pattern Discovery and Predictive Modeling". In: IEEE Transactions on Knowledge and Data Engineering 28.4, pp. 979–993, 2016.

[20] H. Zunair, A. Rahman, N. Mohammed, and J. P. Cohen, "Uniformizing techniques to Process Ct scans with 3d CNNs fortuberculosis prediction," Predictive Intelligence in Medicine, 156-168. doi:10.1007/978-3-030-59354-4

[21] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," CVPR, 2016.