A Neuromorphic Architecture for Anomaly Detection in Autonomous Large-Area Traffic Monitoring*

Special Session

Qiuwen Chen[1], Qinru Qiu[1], Hai Li[2] and Qing Wu[3]

[1]Dept. of Electrical Engineering &
Computer Science
Syracuse University
Syracuse, NY 13244, USA

[2]Dept. Electrical and Computer
Engineering
University of Pittsburgh
Pittsburgh, PA 15261, USA

[3]Air Force Research Laboratory
Information Directorate
Rome, NY 13441, USA

Abstract — **The advanced sensing and imaging capability of today's sensor networks enables real time monitoring in a large area. In order to provide continuous monitoring and prompt situational awareness, an abstract-level autonomous information processing framework is developed that is able to detect various categories of abnormal traffic events with unsupervised learning. The framework is based on cogent confabulation model, which performs statistical inference in a manner inspired by human neocortex system. It enables detection and recognition of abnormal target vehicles within the context of surrounding traffic activities and previous events using likelihood-ratio test. A neuromorphic architecture is proposed which accelerates the computation for real-time detection by leveraging memristor crossbar arrays.**

Keywords – cogent confabulation, anomaly detection, neuromorphic architecture

I. Introduction

Anomaly detection, which refers to the techniques of identifying patterns that do not conform to the regular observations in a given data set, is of the utmost importance. The problems of anomaly recognition and detection frequently arise from different domains, such as medical diagnosis and network intrusion detection. This paper introduces an abstract-level autonomous information framework that detects abnormal traffic behavior over a very large monitoring area using unsupervised machine learning. Taking advantage of the innovative sensing and imaging capability of today's sensor networks, our framework may enable anomalous traffic situation detection for large-area traffic monitoring which is not achievable solely by the human.

Many techniques have been studied for anomaly detection [1]~[5], including SVM classifier, neural networks, nearest neighbor, Bayesian networks and trajectory clustering. But none of these systematically solve the problem of monitoring very large area, nor were they able to pin point the type of anomaly that was detected.

In this paper, we present an autonomous anomaly recognition and detection (AnRAD) framework. The fundamental of the proposed framework is based on *cogent confabulation* [9], which is a computation model that mimics human information processing. It extracts conditional probability among symbolic representations of features in an unsupervised environment. In this work, the large area is firstly partitioned into smaller zones that can be independently processed with balanced effort. Then, a *knowledge base* (*KB*) is built for each zone by extracting vehicle behavioral features and their inter-relations from traffic records. When new traffic information is received, anomaly scores will be calculated by means of *likelihood-ratio test*. The uniqueness of AnRAD can be summarized as the follows:

1. The confabulation based model has very low complexity for both training and recall. Therefore, the system can be trained even in operating time, and this enables continuous improvements to the KB quality.
2. By proper modeling, the system is capable of capturing the contextual information between vehicles and their neighbors. Thus, abnormal events caused by interaction between vehicles, such as tailgating, can be detected as well.
3. The model is able to handle large volume of vehicles over a big area. The overall system has hierarchical architecture and the work load in each level of the hierarchy is inherently parallel.
4. The likelihood calculation is analogous to the working mechanism of the synapse and neuron. It is transformed into matrix-vector operations and can be accelerated using analog domain operation with the help of memristor crossbar arrays.

The rest of the paper is structured as the follows. Section II provides the background concepts of cogent confabulation. The designs of the system framework and algorithm model are elaborated in Section III. Preliminary results are presented in Section IV. Section V discusses potential optimization problems that can be solved using computer aided design (CAD) techniques and Section VI gives the conclusions.

II. Background

Cogent confabulation [6] is a cognitive computing model that mimics the learning, the information storage and the recall process of human brain. It uses a set of features to construct the basic dimensions that describe the world of applications, e.g. vehicle speed or coordinates. Different observed attributes of a feature is referred as *symbols*. The set of symbols used to describe the same feature are candidates of this feature and they are mutually exclusive. *Knowledge links* (*KL*) are established among lexicons. They are directed edges from the source lexicons to target lexicons. Each knowledge link is associated with a matrix. The *ij*th entry of the matrix gives the log-conditional probability $\log[p(s_i|t_j)]$ between the symbol s_i in the source lexicon and t_j in the target lexicon. The knowledge matrix is constructed during training by observing and extracting features from the inputs.

The cogent confabulation model has close resemblance to the neural system. The symbols are analogous to neurons and knowledge links between symbols are analogous to synapses connecting neurons. Whenever an attribute is observed, the corresponding symbol (i.e. neuron) is activated, and an excitation is transmitted to other symbols (i.e. neurons) through knowledge links (i.e. synapses). The conditional probability $\log[p(s_i|t_j)]$ is analogous to the strength of the synapse between s_i and t_j, which (according to the Hebbian learning rule) increases when the two neurons are activated simultaneously.

* This work was partially supported by the National Science Foundation under Grants CCF-1337198 and CCF-1337300, and the Air Force Research Laboratory, under contract FA8750-12-1-0251.

The excitation level of a symbol t in lexicon l is calculated by summing up all incoming knowledge links as Equation (1):

$$el(t) = \sum_{k \in F_l} \left(\sum_{s \in S_k} I(s) \ln\left(\frac{p(s|t)}{p_0}\right) + B \right), \qquad (1)$$

where F_l denotes the set of symbols that have connections to l, and S_k is a set that consists the collections of symbols in lexicon k; $I(s)$ is the firing strength of source symbol s, and it is set to 1 if s is observed without ambiguity; p_0 is the minimum probability that is considered informative. Parameter B is a constant called *band gap*, it is 0 if none of the active source symbol in S_k has knowledge link goes into t. The band gap ensures symbols with more active knowledge links (KLs with active source symbol) receive higher excitation over those with fewer active KLs.

As we can see the excitation level of a symbol is actually its log-likelihood given the other observed attributes. In [7], the excitation levels are used to remove the ambiguity in observation via maximum likelihood inference. In this paper, the excitation level enables us to detect anomaly using the likelihood ratio test method.

Comparing to other schemes, the training and recall process in confabulation model are simple and massively parallelizable. Also, since this model is highly configurable, the system can be easily modified to fit diversified applications, or be optimized for better performance. Finally, because the training and recall processes share the same knowledge data structures, the model offers unsupervised learning and online updating.

Our previous work [10], which performs confabulation-based text recognition on high performance computing clusters (HPC), demonstrates the framework's ability to handle incomplete data and to capture the causal relationship between observations.

III. SYSTEM DESIGN AND MODEL CONSTRUCTION

The input to the AnRAD for both training and recall are radar data formatted as series of vehicle records ordered by time (0.8 sec per time slot). Each record consists of a timestamp, vehicle type, the location and speed of the vehicle represented in the Earth-Centered, Earth-Fixed (ECEF) format.

A. Model construction

To detect an abnormal event, we consider the behavior of a vehicle within the context of its location and neighbors during current and previous observations. If we define all observations made at the same time slot as a *frame*, the current detection algorithm involves 3 consecutive frames. Four classes of objects are defined, *target*, *neighbor*, *auxiliary center*, and *supporter*. Each vehicle appearing in the detection zones in a frame t will be considered as a target. The nearest 10 vehicles of the target in the same frame are called neighbors. Based on current location and speed of target, we can roughly locate it in previous frames. The vehicle records of the target in frames t-1 and t-2 are referred as auxiliary centers. The nearest 10 neighbors of the auxiliary center in the corresponding frame are called supporters. Figure 1 shows an example of the 4 types of vehicle records. An input vector is generated based on the observations of each target within the context of neighbors, auxiliary centers and supporters.

Overall 97 features are used to describe the status and context of a target vehicle.

- Three features are used to describe the status of a target vehicle: target location (L), target speed (V), and target size (S).
- Two features are associated with each auxiliary center: center displacement (ΔL^{-t}) and center acceleration (ΔV^{-t}), t = 1, 2.
- Two features are associated with each neighbor or supporters: relative location (denoted as ΔL_i for the ith neighbor and ΔL_i^{-t} for the ith supporter in frame t) and speed (denoted as V_i, for the ith neighbor and V_i^{-t} for the ith supporter in frame t)
- Three lexicons are associated with each target and neighbor pair: pairwise location ($L_i^{\sim}$), pairwise speed ($V_i^{\sim}$), and pairwise speed changes ($\Delta V_i^{\sim}$).

The set of observed attributes of these features form the input vector, which is the basic operating unit for confabulation training and recall. We treat each vehicle in the detection zone as a target, and generate an input vector for each target.

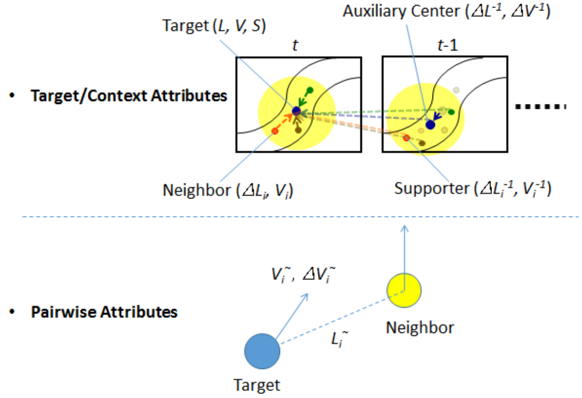

Target (L, V, S)
Auxiliary Center (ΔL^{-1}, ΔV^{-1})
t
t-1
• Target/Context Attributes
Neighbor (ΔL_i, V_i)
Supporter (ΔL_i^{-1}, V_i^{-1})
$V_i^{\sim}$, $\Delta V_i^{\sim}$
• Pairwise Attributes
$L_i^{\sim}$
Neighbor
Target

Figure 1. Classification of vehicle records and features

The features are called *lexicons* in the confabulation model. Figure 2 shows the overall confabulation model with lexicons. Each arrowed connection represents a KL. Lexicons S, L, V and $L_i^{\sim}$, $1 \le i \le 10$, are represented using dashed circles. Each of them corresponds to a general category of an abnormal behavior of the target vehicle, such as abnormal location, speeding, inconsistency between vehicle size and its status, and abnormal interactions with neighbors. We refer these lexicons as *key lexicons* and others as *regular lexicons*. Only the excitation levels of the key lexicons need to be calculated. All other lexicons simply provide contexts for them. A key lexicon has incoming knowledge links from any other lexicons while a regular lexicon only has outgoing knowledge links.

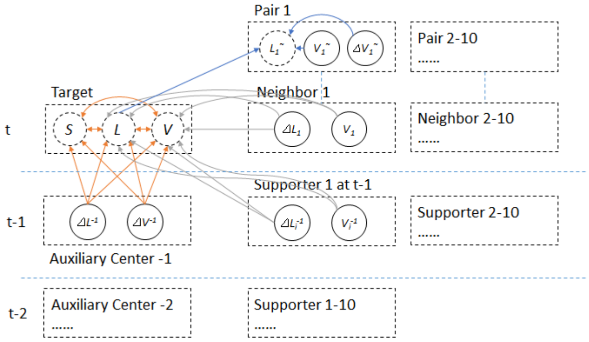


Figure 2. Knowledge links between lexicons

B. Model training and anomaly detection

The input data are collected from a large monitoring area with hundreds or thousands of vehicles appearing at the same time. The complexity of finding the nearest neighbors is a quadratic function of the number of the vehicles in the detection zone. The traffic situation varies significantly at different locations within this large area. Thus increases the complexity of the model and reduces its accuracy. Therefore, before training and recall, we first divide the large area into multiple smaller detection zones. Each detection zone can be processed independently to allow parallel processing. The criterion of partitioning is to create detection zones with near equal average vehicle density and uniform traffic environment. An example result of zone partition is shown in Figure 3. The grids with yellow borders are detection zones to be processed individually.

The confabulation module primarily conducts two procedures: training and anomaly detection. Both procedures are based on the same data structure. For both procedures, the aforementioned features are collected from each vehicle in the given detection zone and assembled into an input vector. Each observed attribute is mapped into a globally unique reference number called a *symbol*. For a given lexicon, all attributes observed during the training process form its *candidate* set.

For each knowledge link, the training process counts the co-occurrence of the source and target symbols and the log-conditional-probability is calculated at the end. All conditional probability will be arranged in matrix format and stored in the knowledge base.

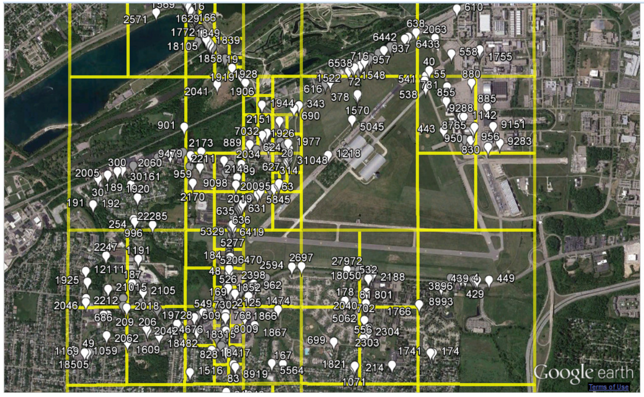


Figure 3 Partition of surveillance area

The detection procedure calculates the excitation levels $el(t)$ for all candidates of the key lexicons using Equation (1), which is the likelihood of the corresponding observations given the context of target and neighbor status. The likelihood-ratio test is then performed to calculate the anomaly score of each key lexicon using Equation (2).

$$as(l,t) = \frac{el(t_{best}) - el(t)}{el(t_{best})} \qquad (2)$$

where t is the observed attribute and t_{best} is the candidate with the highest excitations. A very high anomaly score for symbol t indicates that the likelihood of observing t is much lower than the likelihood of other typical observations in current traffic context. Therefore, t will be marked as an anomaly.

Since abnormal events usually last for multiple frames, a vehicle is reported as abnormal only when the anomaly score of one of its lexicons exceed a threshold in 3 continuous frames. This constraint is specified by Equation (3)

$$\min_{j=0,1,2}\{as^{-j}(l,t)\} > \theta_l \qquad (3)$$

C. Proposed neuromorphic architecture

As the demands on high performance computation continuously increase, traditional Von Neumann computer architecture becomes less efficient. In recent years, neuromorphic hardware systems that potentially provide the capabilities of biological perception and information processing within a compact and energy-efficient platform have gained a great deal of attention [8][9].

Our latest research shows that memristive devices have great potential in the matrix computations with high parallelism [10][11]: Firstly, as a two-terminal device, a memristor is very small and can be easily programmed to different resistance states by biasing the voltages at its two ends; Secondly, the crossbar array built on memristors can efficiently perform matrix-vector multiplication approximation by transforming one group of electrical excitations to another one. Let's use an $N \times N$ crossbar array as the example to demonstrate its matrix computation functionality. We apply a set of input voltages $\mathbf{V_I}$ on the *word-lines* (*WL*) and collect the current through each *bit-line* (*BL*) by measuring the voltage across resistor R_s with conductance of g_s. Assume the memristor sitting on the connection between WL_i and BL_j has a conductance of $g_{i,j}$. Then the output voltages can be represented by $\mathbf{V_O} = \mathbf{C} \times \mathbf{V_I}$, indicating that a trained crossbar array can be used to construct the connection matrix $\mathbf{C}$, and transfer the input vector $\mathbf{V_I}$ to the output vector $\mathbf{V_O}$. Here, $\mathbf{C}$ is determined by the conductance of memristors such as:

$$\mathbf{C} = \mathbf{D} \times \mathbf{G} = diag(d_1, \cdots, d_N) \times \begin{bmatrix} g_{1,1} & \cdots & g_{1,N} \\ g_{2,1} & & g_{2,N} \\ \vdots & \ddots & \vdots \\ g_{N,1} & \cdots & g_{N,N} \end{bmatrix}$$

where, $d_i = 1/(g_s + \sum_{k=1}^{N} g_{i,k})$.

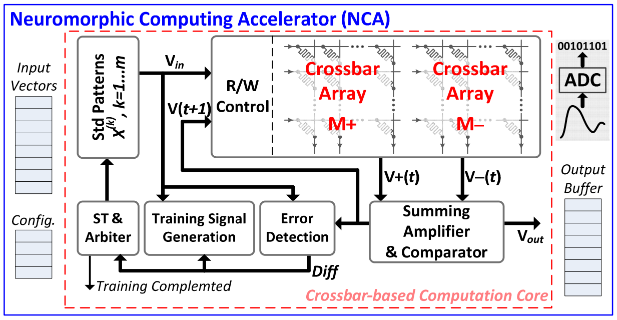


Figure 4 An overview of NCA architecture

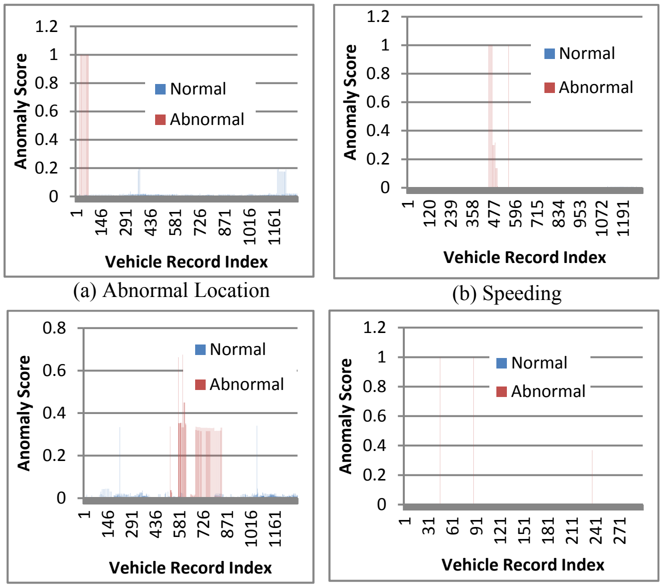
As we can see from Equation (1), the knowledge links of the confabulation model is matrix of conditional probabilities and the calculation of excitation level is dominated by matrix and vector operations. We propose a crossbar-based neuromorphic computing accelerator (NCA) for matrix computations, which can be regarded as a processing element (PE) in Network-on-Chip (NoC) systems.

Figure 4 shows an overview of the NCA architecture. Crossbar arrays conduct matrix-vector multiplications in normal (or recall) operation. Because the device resistance is always larger than zero, two crossbar arrays **M+** and **M–** are required to program the positive and negative elements of a matrix, respectively; Summing amplifiers conduct vector computations at the outputs of the crossbars, such as scaling and summation. The voltage signal generated by the summing amplifiers $\mathbf{V_{(t+1)}}$ is either sent out of the computing module, or fed back to the inputs of crossbar arrays if more iterations are needed.

I/O interface of the crossbar-based NCA includes input/output buffers and configuration queue, which carries the information required for crossbar array programming etc. Since the default operating data type for the NCA is analog, the input/output buffers are able to retain analog data, e.g., by using the variable resistive states of the memristor devices. The data communication among the different NCAs will be managed by a novel analog network-on-chip (NoC) while the analog-digital converters only exist at the interface between the NCA array and the conventional pipeline.

IV. EXPERIMENTAL RESULTS

Experiments are conducted to evaluate the performance of the AnRAD. The monitoring data are collected over a 10-by-10 (mile^2) area. We focus on one partitioned zone of 500*500 (m^2) with moderate traffic density. The training set contains 240 minutes of normal traffic data. The testing set is based on 10 minutes of traffic data that is not included in the training set with one manually inserted abnormal behavior from each category. These abnormal events include: deviating from the road, speeding, 18-wheeler in abnormal speed, tailgating, abnormal starting or stopping activities). Figure 5 shows the anomaly score calculated for each vehicle at different time in these tests. In these figures, the red bars represent the anomaly scores of those vehicles with abnormal behavior, while the blue bars are the anomaly scores of normal vehicles. As we can see that the red bars are significantly higher than the blue bars. This can be easily exploited by a decision threshold. Furthermore, the anomaly scores reveal obvious temporal continuity for most categories of abnormal events, except that of abnormal start/stop of vehicles, which give spikes only in the moment of moving status changes. By these means, the abnormal events can be differentiated from the normal ones.



(a) Abnormal Location

(b) Speeding

(c) Tailgating

(d) Abnormal stop and go

Figure 5 The detection of different anomaly events

V. DESIGN OPTIMIZATION AND PERFORMANCE ENHANCEMENT

The project involves many design optimization and performance enhancement problems that can be solved using traditional CAD algorithms. The detection zone partition problem is a typical balanced graph partition problem. The entire road network can be divided into trellis based on the minimum resolution of the partition. Each segment of the road inside a single grid will be considered as a vertex whose weight is defined by the vehicle density in that segment. The adjacent vertices on the same road are connected by edges. The goal of partition is to generate detection zones with relatively independent zones with approximately equal vehicle density and minimum crossing zone traffic. This is equivalent to finding sub-graphs that have balanced weight and minimum interconnections.

How to map and schedule the training and detection procedures on a high performance computer (HPC) is essentially a real-time scheduling problem. The objective is to use minimum computing resources to cover maximum surveillance area while satisfying the throughput requirement of the input data stream. As we mentioned before, when the size of the detection zone increases, its processing complexity also increases. However, at the same time, the number of zone reduces. From performance perspective, there is a tradeoff between the size of the zone and the number of zones. Proper performance model should be established to determine the best size for the zone partition.

With the help of NCA, the matrix vector operation can be accelerated via analog domain operations. How to map the computation to the NCA is also a CAD problem. The knowledge links of the detection problem are sparse matrices. Mapping each KL into one NCA does not have high utilization. The more efficient way is to partition the sparse matrices into sub-matrices with higher density. However, as the size of each NCA reduces, their peripheral overhead increases. There is a fundamental tradeoff between the NCA utilization and its overhead. Furthermore, the connections among multiple NCAs need to be carefully routed through the NoCs in order to fully realize the speed up provided by the hardware.

VI. CONCLUSIONS

In this paper, we presented the modeling and implementation of an anomaly detection system for traffic monitoring. The detection is realized using likelihood-ratio test and the statistical knowledge of the traffic collected from unsupervised learning. Given the analogous between the proposed model and human neocortex system, a neuromorphic architecture is proposed that accelerates the computation through analog domain operations.

ACKNOWLEDGMENT OF SUPPORT AND DISCLAIMER

This work was partially supported by the National Science Foundation under Grants CCF-1337198 and CCF-1337300, and the Air Force Research Laboratory, under contract FA8750-12-1-0251.

Received and approved for public release by AFRL on 08/26/2013, case number 88ABW-2013-3818. Any Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or its contractors.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey,” ACM Computing Surveys (CSUR), Volume 41 Issue 3, July 2009.

[2] V. Roth, “Outlier Detection with One-class Kernel Fisher Discriminants,” Proc. of the Conference on Advances in Neural Information Processing Systems, 2004.

[3] A. A. Sebyala, T. Olukemi, and L. Sacks, “Active platform security through intrusion detection using naive Bayesian network for anomaly detection,” Proc. of the London Communications Symposium, 2002.

[4] Z. Fu, W. Hu, T. Tan, “Similarity based vehicle trajectory clustering and anomaly detection,” Proc. of International Conference on Image Processing, 2005.

[5] H. Sheng, C. Li, Q. Wei and Z. Xiong, “Real-time Detection of Abnormal Vehicle Events with Multi-feature over Highway Surveillance Video,” Proc. of International Conference on Intelligent Transportation Systems, 2008.

[6] R. Hecht-Nielsen, Confabulation Theory: The Mechanism of Thought, Springer, August 2007.

[7] Qinru Qiu, Q. Wu, M. Bishop, R. Pino, and R. W. Linderman, “A Parallel Neuromorphic Text Recognition System and Its Implementation on a Heterogeneous High Performance Computing Cluster,” IEEE Transactions on Computers, 2013.

[8] P. Camilleri, M. Giulioni, V. Dante, D. Badoni, G. Indiveri, B. Michaelis, J. Braun, and P. del Giu-dice, “A neuromorphic avlsi network chip with configurable plastic synapses,” in *International Conference on Hybrid Intelligent Systems*, pp. 296–301, 2007.

[9] J. Partzsch and R. Schuffny, “Analyzing the scaling of connectivity in neuromorphic hardware and in models of neural networks,” *Neural Networks, IEEE Transactions on*, vol. 22, no. 6, pp. 919–935, 2011.

[10] M. Hu, H. Li, Q. Wu, and G. Rose, “Hardware Realization of Neuromorphic BSB model with memristor crossbar network,” *IEEE Design Automation Conference* (*DAC*), pp. 554–559, 2012.

[11] M. Hu, H. Li, Q. Wu, G. Rose, and Y. Chen, “Memristor Crossbar Based Hardware Realization of BSB Recall Function,” *International Joint Conference on Neural Networks* (*IJCNN*), pp. 1-7, 2012.