

ML/AI in Intelligent Forest Fire Observer Network

Ljiljana Šerić, Darko Stipaničev, Damir Krstinić
{ljiljana.seric,darko.stipanicev,damir.krstinic}@fesb.hr

University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, Department for modelling and intelligent systems, Ruđera Boškovića 32, 21000 Split, Croatia.

Abstract. This paper discusses advantages of using machine learning and artificial intelligence (ML/AI) techniques in forest fire observer. Observer network was proposed as a framework for artificial perception software system relying on sensor network deployed on monitoring locations and used for detection of phenomenon. The architecture of the system distinguishes three main parts where data is analyzed - *network observer* or proprioceptor who checks upon validity of system parts and data available from sensors, *phenomenon observer* for detection of scenario taking place in the environment that is monitored and *system observer* for analysis of system usefulness and identification of future improvement of the system. In each of these parts ML/AI techniques are used. This framework was implemented as vital part of intelligent forest fire monitoring and surveillance system deployed in coastal part of Croatia and some results are presented.

Keywords: Artificial intelligence, Machine learning, artificial perception, forest fire detection.

1. Introduction

Detecting forest fires from a single or series of digital images has led many scientist to work on applying image processing and computer vision techniques in search for algorithm to detect fire or smoke [1][2]. In these researches scientists aim their focus on precision and accuracy of proposed algorithm. However, before the detection algorithm is put in use in production system, the system must reach stability of the industrial system. Besides the precise and accurate detection algorithm such system must involve mechanisms for observation of vital parts and automatic recover in case of failure.

Artificial intelligence and especially machine learning have found their ways to enter many fields of research and technology. Traditionally artificial intelligence deals with making programs that perform tasks that need intelligence when done by humans and this is done by automating human path in decision making. This approach is commonly referred to as symbolic intelligence. However, since computers have different approach to thinking then humans and thus computers are good in computation rather than exploring, another approach is more suitable for achieving this task.

Machine learning has recently become the more popular field of artificial intelligence, mostly due to the availability of digitalized data (ie. Big Data). Machine learning algorithms are used on set of training data and aim to give output that can be used as a part of intelligent algorithm. Usually they are used to create a model that describes the data and future data are checked upon that model.

Technology of Internet of Things makes more and more data available to the researchers and developers defining services upon those data. Layered architectures for data processing [4] enable having multiple services defined over the same set of data.

Forest fire monitoring and surveillance are important tasks of forest management where forest owners want to prevent and minimize the damages emerged by forest fires. Intelligent forest fire monitoring and surveillance system[10] is a software system supporting cameras mounted on pillars on monitoring locations in the forest. The software system is performs following tasks:

- collecting data from the monitoring locations,
- analyzing validity of data collected from the monitoring location,
- data processing in search for visible signs of smoke and fire,
- data archiving and storing for future use,
- presenting the results of the processing to the users via web interface.

In this paper we will focus on software components responsible for :

- data collection and validation and
- fire detection

Following tasks can be formally described in the framework of formal theory of perception. Formal theory of perception [3] introduces the notion of the observer as a central subject for mathematical formalization of the process of perception. Perception is a process with a task of detecting if the scenario taking place in the environment contains phenomenon of interest.

Observer is 6-tuple:

$$O = (X, Y, E, S, \pi, \eta) \quad (1)$$

Measurable space X is called configuration space and consists of points representing various scenarios that can play in the environment, and whose values are those aspects of the scenarios that observer uses in reasoning about the existence of the phenomenon. Subset of space X , measurable space E , called the event configuration, and consists of the values of the same aspects of those scenarios where we can observe particular phenomenon that specific observer recognizes. The space Y is called observation space usually has smaller dimensions than the space X , and consists of the value of various aspects of scenario available to observer. Subset S consisting of points having assigned values for which the observer concludes that the phenomenon occurred is called observation event. Perspective function π performs mapping from the actual value to the value available to the observer. It is important to emphasize that π is not bijection, i.e., multiple points of X can be mapped to the same point in Y . Combination of values of a point s_1 from S that can be presented the observer can be caused by more than one different points from X , some of which are and some are not from E . The conclusion kernel, η , assigns to each point of the S probability measure supported on a given set of inverse perspective mapping map in X , $\pi^{-1}(s)$. This parts are illustrated in Figure 1.

In the rest of the paper we will describe the observer network three layer architecture for artificial perception used for detection of phenomenon based on sensor data. For each layer advantages of using ML/AI techniques will be stressed out. In the next section we will describe forest fire observer network implemented as a part of intelligent forest fire monitoring system and describe some specific cases where ML algorithms are employed. In final section we will present results after first month of system operation and give conclusions.

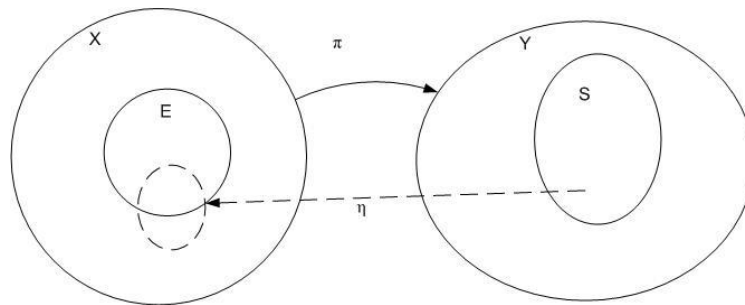


Figure 1. Definition of observer

2. Observer network

The notion of observer network was proposed in [3][6] and described in [7]. Observer network is a framework for artificial perception incorporating three layer architecture. The framework is inspired and formally described using formal theory of perception and mathematical notion of observer.

The framework is defined over sensor network but can also be extended to notion of Internet of Things. The physical part of the framework consists of sensors located in the environment, power and communication infrastructure and computing power. Physical location of computing nodes does not influence on framework definition and functionalities.

Figure 2 depicts the three layer architecture showing: *physical infrastructure*, *network observer* performing data collection and validation, *phenomenon observer* performing detection of phenomenon and *system observer* performing analysis of the system usage and reporting.

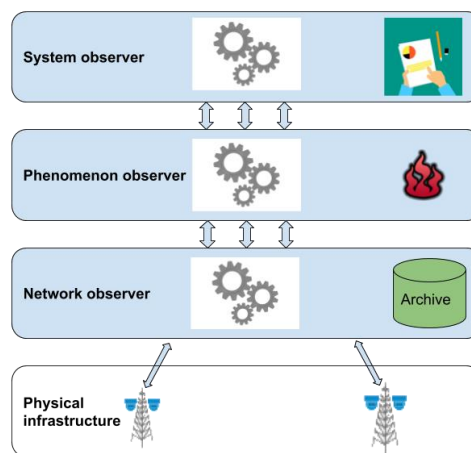


Figure 2. Observer network architecture

Network observer – proprioception

Network observer is part of observer network responsible for sensor data collection and validation. The task of validating its own sensory system validity is referred to as proprioception in psychology. Proprioception validates if all data is correctly gathered and if all data is valid for further processing. This step is important because it reduces the number of false perceptions on the next level where invalid data in the input cause invalid conclusion. Validity of data is checked in several aspects: variety of value span within time series of single sensor, and in the scope of all data collected from the field at certain moment of time.

Network observer besides checking sensor and data validity can also be extended for checking processing parameters and overall computational system state. Sometimes system malfunctioning can be result in hardware, operating system or software environment malfunctioning and this is not easily detected when operating on large number of processing.

Let us denote with N set of all sensors of one monitoring node

$$N_i = \{S_1, S_2, S_3, \dots\} \quad (2)$$

Each sensor maps some aspect of the environment into sensor value. We will refer to sensor value as data and denote with D :

$$S : X \rightarrow D(S, t) \quad (3)$$

Set of all sensors from a monitoring node form an information object we will denote with I

$$I(N_j, t_i) = \{D(S_k, t_i) : S_k \in N_j\} \quad (4)$$

Validity of both $D(S, t)$ and $I(N_j, t_i)$ is checked by calculating probability measure of fitting the value into the model of valid data. Model of valid data is created by applying machine learning techniques on set of past data that is denoted as valid.

If network observer ends its task with conclusion that system is working within desired performance and all data is valid it creates information object containing all aspects of data measured by sensors and retrieved from other sources and passes it to phenomenon observer.

Phenomenon observer - alarm detection

The task of the **phenomenon observer** is recognition or detection of phenomenon of interest. This is only worth trying if all data acting as input data for detection is available and valid as checked by network observer. Set of input data forms an information object denoted with $I(N_j, t_i)$.

Phenomenon observer performs analysis of all associated data and assigns the information object the probability of occurrence of the phenomenon of interest.

On this layer of observer network architecture we can deploy more than one instance of phenomenon observer. This means that it is possible to employ detection of different type of phenomenon on the same data and information objects.

If the phenomenon detected is some sort of unwanted phenome on i.e. natural disaster, appropriate alarm must be raised.

Having multiple phenomenon detection can be deployed in a way that each phenomenon observer acts individually raising alarms to appropriate services. By fusing results of or they can compare their results and we can give another agent for decision making on what phenomenon is assigned largest probability.

One Phenomenon observer thus can be defined as :

$$O = (Sc(N_i), I(N_i), E(N_i), Ph(N_i), S, Pm(Ph(N_i))) \quad (5)$$

Where:

$Sc(Ni)$ is configuration space, the set of all scenarios taking place in the environment of the observer node Ni

$I(Ni)$ is observation space, the set of images of those scenarios available to the observer

$E(Ni)$ is event configuration, the set of images of those scenarios really containing phenomenon occurrence

$Ph(Ni)$ is observation event, the set of images where the observer makes decision that the phenomenon is present

S is perspective map, the function of mapping from scenario to image $S : Sc(Ni) \rightarrow I(Ni)$. It holds the functions of all sensors that are part of observer node. Set $I(Ni)$ contains images of scenario captured by mapping function S ,

$Pm(Ph(Ni))$ – probability measure assigned to each image form $Ph(Ni)$ that the image contains the phenomenon of interest

For each image of the scene, one of the following situations can happen:

$I(Ni,t) \in Ph(Ni)$ or

$I(Ni,t) \notin Ph(Ni)$

Machine learning approach to assigning probability measure to information object can be implemented by training a model on a set of data containign phenomenon and calculating the probability of data fitting to the model. However this is not trivial if information object contains complex data structures such as images, video and audio recordings.

System observer – business analysis of usefulness

Evaluation of the network operation must be carried out regularly, and preferably automatically. In this evaluation we take into account following features of the system:

- operation of sensors
- results of network observer,
- software environment (operating system and installed supporting software tools services and libraries),
- software and hardware performance (memory, CPU and disk usage)
- phenomenon observer results
- type and frequency of user actions.

The results of this actions are presented in the forms of reports used by system managers and team leaders to decide on most vulnerable parts of the systems and possible improvements in hardware, software and functionalities.

3. Forest fire observer network

As a case study we will present forest fire observer network implemented by the authors of this article and being a part of intelligent forest fire monitoring and surveillance system.

The system relies on selected monitoring locations in the coastal part of Croatia equipped with video sensors and communication equipment. Software part of the system runs in data

center. The aim of the system is early forest fire detection so the forest fire observer network is defined.

Intelligent Network observation

Network observer has task of detecting malfunctions in hardware, depending software, and communication infrastructure that can result in fault final decision.

This is performed on two levels: syntactic – a single sensor data level and semantic – on the sensor node level.

Syntactical level consists of checking if all necessary elements are present and in valid format. Machine learning algorithms are used for training model of validity on single value and typically include outlier detection techniques.

Semantical data validation is done by checking the associated values. Analysis is performed on set of available data representing different features of the network: both values from the sensors and processing parameter like memory consumption, available bandwidth, number of active and zombie processes.

Proprioception based on Bayesian network is presented in [8] and extended to Bayesian network shown on figure

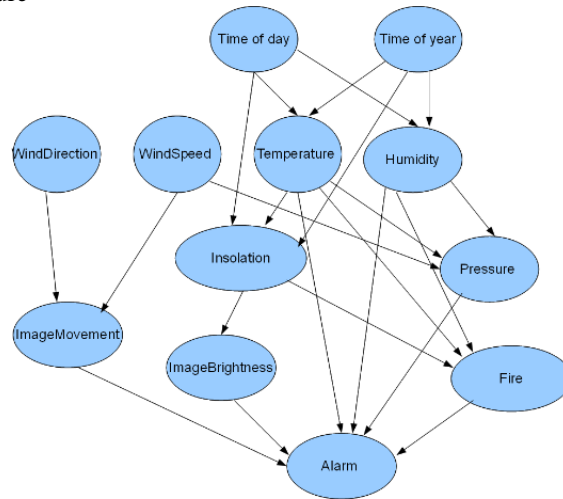


Figure 3. Structure of Bayesian network for sensor data validation in forest fire observer

The decision on data validity is made based on weight of soundness of dataset that is calculated as:

$$WOS = \prod_{Ni}^i P(V(Ni) = Vi | \{Vj = V((Nj); \forall Dependencies(Ni)\}) \quad (6)$$

Another approach to create model of valid data is to perform Principal Component Analysis (PCA) [5] on set of valid data. The process of creating a model and further assignment of weight to the sample is shown in figure 4.

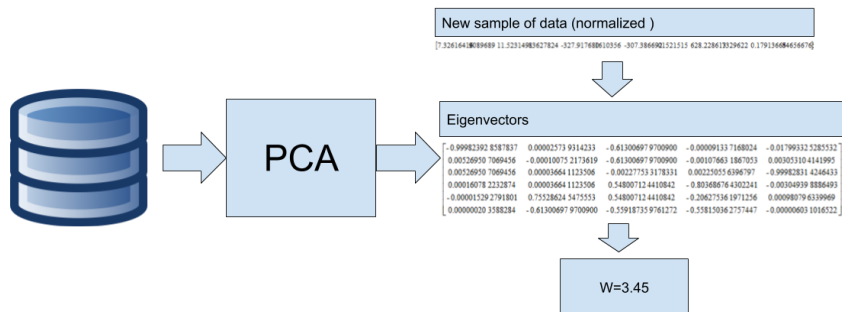


Figure 4. Proprioceptor based on PCA model

Intelligent forest fire observer

Phenomenon observed in forest fire observer network is smoke and fire. Usually detection of forest fire in Mediterranean landscape comes down to problem of detection of rising smoke in video images. Because of hilly landscape it is common that source of fire is not directly visible from the monitoring location, but the effect of forest fire - rising smoke, quickly reaches the altitude so it is visible in image.

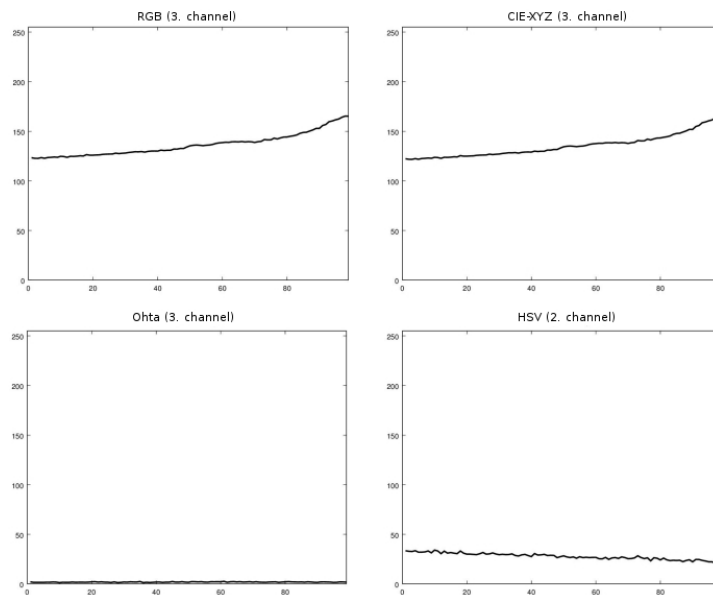


Figure 5. Examples of histograms that show how the average value of a ROI changes through time in different channels. x-axis represents each image in the video sequence, and y-axis represents different average values that the ROI can assume

The problem with detecting smoke is its nature cannot be mathematically described. The measure of probability that an image of environment has smoke can be represented various ways. Smoke has color and features that are easily spotted by humans, but when analyzed by

computer program smoke adds transparent layer to its environment and can not be easily segmented from its surrounding environment.

Relying on other color models [9], besides Red-Green-Blue color model (RGB) can help computer programs more accurately distinguish the segments of images containing smoke. Defining the best color model is not trivial task for analytical approach but evaluating various model on large set of images makes it lot easier and possible. One of the approaches is training a model that describes change of the region in the series of images on large database of collected images. Such change of the specific color channel is shown in figure 5.

Data mining and business analysis

Development of information system is continuous process that never ends, especially if the system is in commercial use and has users accessing the system functionalities. Maintenance of a software system demands performing continuous testing of existing implemented functionalities and correcting malfunctioning of the system that sometimes arises because of changes with upgrade of software environment (ie. operating system, security patches etc.). Implementing new functionalities that are defined in continuous cooperation with users is what makes software successful.

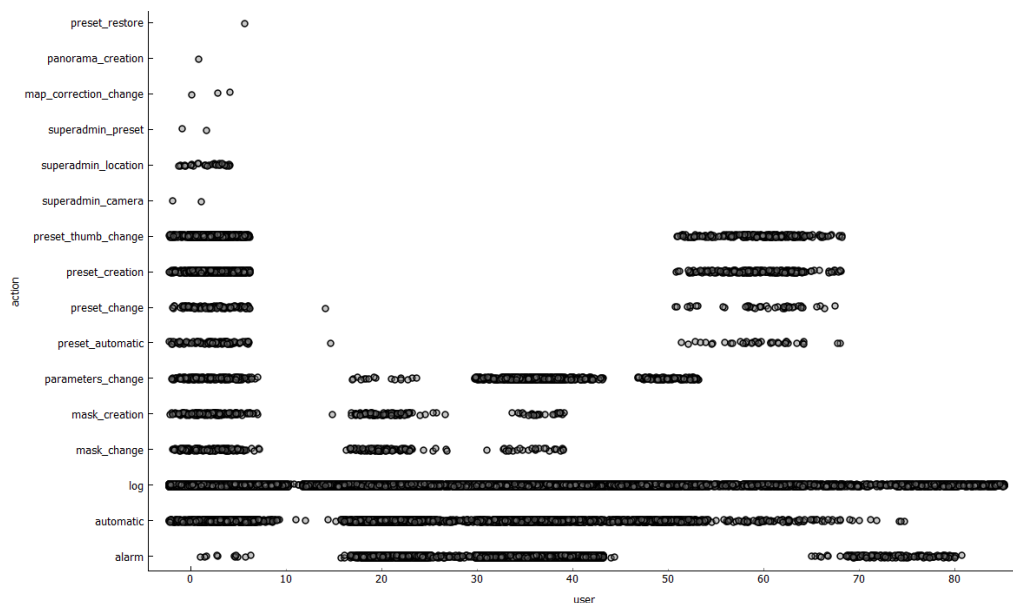


Figure 6. Plot of user activities showing that users could be clustered by preferences

Demand for logging all user actions within the software arises from security and maintenance demand. Analyzing user logs with available machine learning algorithms can make us aware of user preferences. Active user contribute to the system functionalities and can be rewarded with more accurate information for decision making. Users model can be presented to employers so in the future their tasks can be assigned based on their preferences. Figure 6 shows action logs for system users. It is shown that certain users have preferences on certain actions.

Analysis of user actions associated with processing parameters of the underlying infrastructure can be good input in finding software weak points and propose needed improvements.

Performing regular evaluation of observer network detections can enrich database with new cases used for model update. In such a way the system can be continuously improved without adding calculation complexity in processing algorithms.

4. Results

Some numbers that illustrate the results of the forest fire observer network are presented in table 1. The numbers are collected after the first season of intelligent forest fire monitoring and surveillance system. The fire season is period from June 1 till October 1. The system was implemented on 43 locations in forests of coastal part of Croatia and maintained by Croatian Forests Ltd.

Table 1. Results of the analysis of activity logs after the first fire season

Feature	Number
Number of users	74
Number of system accesses	81.101
Number of activities	1.548.338
Number of interventions on sensors	881
Number of automatic interventions on sensors	665
Number of human interventions after alarm	8.899
Confirmed alarms	399

The table shows that system has been used intensively by users in this period and produced over million and a half activity logs. The majority of the sensors problems have been solved automatically, while only 25% of the problems required intervention. Human intervention after alarm describes situations when human operator inspected some event in the environment by manually checking the sensor and the number describes the number of true detections of the phenomenon. During the fire season

Conclusion

Design and development of an intelligent system is nowadays unimaginable without employing some sort of machine learning. Machine learning techniques consist of training a model on collected data and using the model in analysis, prediction and decision making.

While development of mathematical model and evaluating it on data collected in the usage time is traditional approach, nowadays data is needed in both parts of design and evaluation. We have described how machine learning from data is successfully employed on three levels of software architecture of forest fire observer network. The modern approach improves the accuracy of detection and simplifies maintenance of the system. Conclusion that machine

learning and computational artificial intelligence is more appropriate than symbolic artificial intelligence for the tasks of artificial perception is consistent with general opinion that human perception is based on learning from data samples shown in early ages and developing a conclusion model based on training data that can be further updated with new samples for more precision. All models will be updated with data from Intelligent forest fire monitoring and surveillance system installed on 43 location at the beginning of 2018 fire season.

Acknowledgments. This paper summarizes the research carried out throughout the project “iORVO - Intelligent Observers in Real and Virtual Environment, 2013 – 2016” that was supported by Ministry of Science, Education and Sport of the Republic of Croatia and University of Split. A part of research was also conducted within the framework of HOLISTIC project (1°str./0001/0 - Adriatic Holistic Forest Fire Protection) financed by EU through Adriatic IPA Cross Border Cooperation 2007 – 2013). The described commercial system was implemented and deployed with cooperation with Transmitters and Communications Ltd. (OiV) and financed by Croatian Forests Ltd.

References

- [1] V. Burak Celen, M. Fatih Demirci, "Fire detection in different color models", Proceedings of the 2012 International Conference on Image Processing, Computer Vision, & Pattern Recognition, 2012.
- [2] T. Çelik, H. Özkaramanli, H. Demirel, "Fire and smoke detection without sensors: image processing based approach", 15th European Signal Processing Conference (EUSIPCO 2007), pp. 1794-1798, 2007.
- [3] Bennet, B.M. ; Hoffman, D.D.; Prakash, C.: Observer Mechanics - A Formal Theory of Perception, Academic Press Inc., Public Domain, 1989.
- [4] Ning, H.: Unit and Ubiquitous Internet of Things, CRC Press, 2013, ISBN: 9781466561663
- [5] Abdi, H. and Williams, L. J. (2010), Principal component analysis. WIREs Comp Stat, 2: 433-459. doi:10.1002/wics.101
- [6] Šerić, Lj. : Environmental intelligence based on agent observer networks. Doctoral thesis. Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Croatia, 2010.
- [7] Šerić, Lj. ; Stipaničev, D. ; Štula, M. : Observer network and forest fire detection. Information Fusion, 2011, vol.12, pp. 160–175.
- [8] Šerić Lj., Štula M., Stipaničev D., Braović M. Bayesian Proprioceptor for Forest Fire Observer Network. In: Jezic G., Kusek M., Nguyen NT., Howlett R.J., Jain L.C. (eds) Agent and Multi-Agent Systems. Technologies and Applications. KES-AMSTA 2012. Lecture Notes in Computer Science, vol 7327. Springer, Berlin, Heidelberg
- [9] Ohta, Y.-I. ; Kanade, T. Sakai, T. : Color information for region segmentation. Computer Graphics and Image Processing, 1980, vol. 13, pp. 222–241.
- [10] Krstinić, D. ; D. Stipaničev, D.; Jakovčević, T.: Histogram-based smoke segmentation in forest fire detection system. Information Technology and Control, 2009, vol. 38, pp. 237–244.
- [11] Štula, M., Krstinić, D. & Šeric, L.: Intelligent forest fire monitoring system, Inf Syst Front (2012) 14: 725. <https://doi.org/10.1007/s10796-011-9299-8>