Video meteor detection filtering using soft computing methods

Emil Silađi[1], **Denis Vida**[1] **and Emmanuel Karlo Nyarko**[2]

[1] **Astronomical Society "Anonymus", B. Radića 34, 31550 Valpovo, Croatia**

Faculty of Electrical Engineering, J.J. Strossmayer University of Osijek, Kneza Trpimira 2b, 31000 Osijek, Croatia
emil.siladi@gmail.com, denis.vida@gmail.com

[2] **Faculty of Electrical Engineering, J.J. Strossmayer University of Osijek, Kneza Trpimira 2b, 31000 Osijek, Croatia**
nyarko@etfos.hr

In this paper we present the current progress and results from the filtering of Croatian Meteor Network video meteor detections using soft computing methods such as neural networks and support vector machines (SVMs). The goal is to minimize the number of false-positives while preserving the real meteor detections. This is achieved by pre-processing the data to extract meteor movement parameters and then recognizing patterns distinct to meteors. The input data format is fully compliant with the CAMS meteor data standard, and as such the proposed method could be utilized by other meteor networks of the similar kind.

1 Introduction

Please keep in mind this has been the first attempt by the Croatian Meteor Network to use soft computing for the filtering of video meteor detections and as such is limited in its scope and only gives a high level description of the method used. The aim of this paper is to test the validity of such methods, to see if they are even possible in our case and to provide a starting point for future, more robust work. Although it has little directly to do with this work, it is worth mentioning that radio meteor detection using neural networks has also been recently explored (Roman et al., 2014).

Unlike in hard computing where in the end we obtain accurate results through a series of calculations, soft computing allows us a certain degree of imprecision. In addition, the way in which the problem is solved is not pre-defined and the system finds its own way to the solution. Such solutions are not guaranteed to be optimal. How it was reached is also not always known to us and cannot be mathematically described. The payoff is that we have the ability to get a result through knowing its desired parameters but without needing to know the path to that solution. Such methods can save a lot of computation power and are themselves highly parallel algorithms (Jang et al., 1997).

The current software for meteor detection from CAMS standard data was written by Peter Gural (Jenniskens et al., 2011) and is used within the Croatian Meteor Network as of 2014 (Vida et al., 2014). This software has a reputation for producing large volumes of false detections (non-meteoric events) such as bats, birds, airplanes and clouds. The filtering of meteor detections is a classification issue. To provide a solution, we will, in this paper, use neural networks and support vector machines (SVM) that are specifically designed for classification.

2 Preparing the raw data

In order to make use of the soft computing methods, the detection data needs to be properly prepared. Each event is divided into detection points, the number of which depends on the event duration. In our case, however, this format does not work because we need to input a fixed number of variables. The raw data format also includes additional data for each detection that we cannot use. Having tidied up the data we are left with just the video frame number (used for timing), the location of the detection point in the video (represented by X and Y pixel coordinates) and the light intensity of the detection at that time.

In order to have a valid input, we need each event to supply the same number of variables. This is regardless of the number of detection points in the event. For this reason the data for each event is first calculated and then split in 3 sections. The reason for choosing 3 sections is to allow events containing a small number of detections to be processed. The way in which data is processed is shown in *Figure 1*.

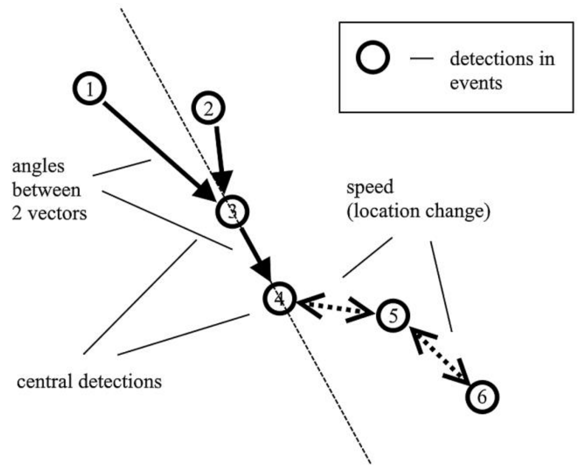

Figure 1 – Sketch of data processing.

Angles between vectors and angular velocities for an event are calculated by using the coordinates of each detection point. The two centermost detection points are found and this gives us the first vector. Starting from the first detection point, we calculate the vector to the nearest centermost detection point and that gives us the second vector. The angle between the two vectors is then calculated for each detection point. Angular velocity is similarly calculated by defining a reference velocity that is equal to the velocity between two centermost detections. Starting from the first pair of detections, the angular velocity between each pair is calculated and is then subtracted from the reference angular velocity to get the pair's velocity relative to the reference. Light intensity is calculated as a percentage, where 100% represents the brightest event of that night, and this is repeated for each detection point.

In order to have the same number of variables, averages are taken for each section for each calculation (angles, velocities, light intensity), giving us 9 variables. When there is not enough data, angles and relative velocities are set to zero. Angles should be close to zero if the event follows a straight path and the velocity is set to be equal to the reference. After the preprocessing, each event has 9 data variables, plus one binary variable that flags it as a meteor or non-meteor. This latter division of events was carried out manually with the aid of CMN_binViewer[1] software. 178 meteors and 222 non-meteors were chosen giving us 400 samples of data.

3 Neural networks

Artificial neural networks are an attempt to imitate real neural networks for the purpose of solving problems. They consist of multiple simple processing units called neurons that are linked together and separated in layers. Each neuron has a certain function that it performs on the input and then sends it to the next neuron. It also has a weight which can be adjusted and is used when training the network. Neural networks can be trained on smaller samples of data and then used to process new data. They are useful when encountering problems difficult to solve with classic computing such as pattern recognition (Krose et al., 1996).

In this paper we used MATLAB’s Neural network pattern recognition tool (nprtool) which classifies inputs into a set of target categories. It uses a two-layer feed-forward network with sigmoid hidden and softmax output neurons. The input is divided in 3 categories: training (70%), validation (15%) and testing (15%). The training set is used during training to adjust the network. The validation set is used to measure networks generalization and to halt training when it stops improving. The testing set provides an independent measure of network performance.[2]

By a method of trial and error, choosing between 5 and 100 neurons it was found that 60 neurons in the hidden layer gave the best results. Results are displayed in *Table 1*. The aim is to minimize the numbers of false positives and of meteor recognition failures.

Table 1 – Classification with neural network.

Meteor detections	Percentage
True positive	89.9%
False positive	10.1%
True negative	91.9%
False negative	8.1%
Positive	91%

4 Support vector machines (SVM)

Although the main aim of the work was to prepare the data for neural networks, it was found that support vector machine can also be used with the same input. SVMs are learning algorithms that can recognize patterns and among other things can be used for binary classification. A model is constructed that categorizes elements of the new data to one of 2 categories defined by the training set. Data can be represented as points in an n-dimensional space where SVM tries to find an n-dimensional hyperplane that best separates the data. Using the kernel trick, data can also be mapped into higher dimensional space which sometimes gives clearer data separation (Cortes et al., 1995).

The software used is called Orange[3], an open source data visualization and analysis tool. SVM was set up by using the graphical user interface shown on *Figure 2*.

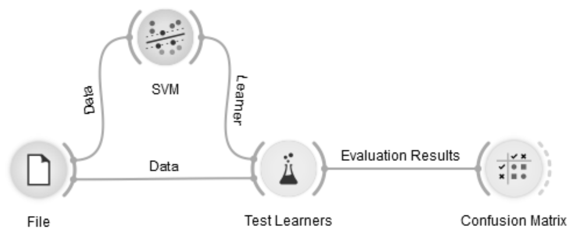

Figure 2 – Orange setup for SVM.

The results are shown in *Table 2* and as we can see there is a lower percentage of false positive results compared to results from the neural network.

Table 2 – Classification with SVM

Meteor detections	Percentage
True positive	92.1%
False positive	7.9%
True negative	88.7%
False negative	11.3%
Positive	90.25%

[1] http://cmn.rgn.hr/downloads/downloads.html#binviewer
[2] http://www.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html

[3] http://orange.biolab.si/

5 Conclusion and future work

As the goal of this paper was to test the validity of soft computing methods, the results look promising. The desired goal was to have the lowest number of false positive detections as possible while preserving the real meteor detections. The approximately 10% false positive detection rate is not good enough for real-life software implementation but can be lowered. The data can be represented in different ways and the implementation of more variables that could better represent the data could in turn give us better results. There are many other classification algorithms and different types of neural networks that were not tried out but which could give us even more options in the future. The results are satisfactory enough to warrant future work and testing with the soft computing methods.

Acknowledgments

We would like to thank Zvonimir Ivešić and Matko Turalija for classifying most of the original data and helping to verify the tests.

References

Cortes C. and Vapnik V. (1995). “Support-vector networks”. *Machine Learning*, **20**, 273–297.

Jenniskens P., Gural P. S., Dynneson L., Grigsby B., Newman K. E., Borden M., Koop M. and Holman D. (2011). “CAMS: Cameras for Allsky Meteor Surveillance to validate minor meteor showers”. *Icarus*, **216**, 40–61.

Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani (1997). “Neuro-Fuzzy and Soft Computing”. Prentice Hall, 1997.

Krose B. and van der Smagt P. (1996). “An introduction to neural networks”. University of Amsterdam, 1996.

Roman V. S. and Buiu C. (2014). “Automatic detection of meteors using artificial neural networks”. In Rault J.-L. and Roggemans P., editors, *Proceedings of the International Meteor Conference*, Giron, France, 18-21 September 2014. IMO, pages 122–125.

Vida D., Šegon D., Gural P. S., Martinović G. and Skokić I. (2014). “CMN_ADAPT and CMN_binViewer software”. In Rault J.-L. and Roggemans P., editors, *Proceedings of the International Meteor Conference*, Giron, France, 18-21 September 2014. IMO, pages 59–63.

The lecture was given by *Denis Vida* (Photo by *Axel Haas*).