sciendo

JAISCR, 2022, Vol. 12, No. 4, pp. 299 – 306
10.2478/jaiscr-2022-0020

SEMANTIC HASHING FOR FAST SOLAR MAGNETOGRAM RETRIEVAL

Rafał Grycuk[1,*], Rafał Scherer[1], Alina Marchlewska[2], Christian Napoli[3]

[1]*Department of Intelligent Computer Systems, Częstochowa University of Technology, al. Armii Krajowej 36, 42-200 Częstochowa, Poland*

[2]*Institute of Information Technologies, University of Social Sciences ul. Sienkiewicza 9, 90-113 Łódź, Poland*

[3]*Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Roma 00185, Italy*

[*]*E-mail: rafal.grycuk@pcz.pl*

Submitted: 12th May 2022; Accepted: 19th October 2022

Abstract

We propose a method for content-based retrieving solar magnetograms. We use the SDO Helioseismic and Magnetic Imager output collected with SunPy PyTorch libraries. We create a mathematical representation of the magnetic field regions of the Sun in the form of a vector. Thanks to this solution we can compare short vectors instead of comparing full-disk images. In order to decrease the retrieval time, we used a fully-connected autoencoder, which reduced the 256-element descriptor to a 32-element semantic hash. The performed experiments and comparisons proved the efficiency of the proposed approach. Our approach has the highest precision value in comparison with other state-of-the-art methods. The presented method can be used not only for solar image retrieval but also for classification tasks.

Keywords: Content-Based Image Retrieval, Image Descriptor, Solar Analysis

1 Introduction

The Solar Dynamics Observatory (SDO) is a part of NASA's Living With a Star (LWS) Program for researching Sun activity impacts on Earth. SDO provides data concerning the solar atmosphere on small scales of space and time and in many wavelengths. One of the SDO instruments is the Helioseismic and Magnetic Imager (HMI) devised for analysing oscillations and the magnetic field at the solar surface (photosphere). It provides dopplergrams, continuum filtergrams and line-of-sight and vector magnetograms (maps of the photospheric magnetic field). Our work is based on the magnetograms, and we propose a method to fast retrieve similar ones. The SDO spacecraft produces data at an enormous rate, so it is impossible to manually annotate and search this collection. There were developed some image retrieval methods but usually for real-life images. Semantic hashing is a method to reduce the dimensionality by similarity-preserving short codes. Such codes should reflect as much as possible the content of the input data. The term was coined by [1] and was later used for any short codes reflecting data content-similarity.

The rest of the paper is organized as follows. Section 2 describes the proposed method for solar

hash generation. Section 3 presents the outcome of the example experiments with the hash. Section 3 concludes the paper.

2 Semantic Hashing for Solar Magnetogram Images

The SDO's instruments allow not only to take solar images in various wavelengths (AIA), but they also provide a data to build magnetograms of the Sun. The magnetic field in active regions can be 1,000 or more times stronger than the average magnetic field of the Sun. The magnetograms provide data about magnetic fields of the entire solar disk, therefore we can use these images in many areas of solar analysis. From our perspective the usage of magnetograms in solar image description or solar image hashing should increase precision of the hash due to noise reduction. The regular active region images can contain bright pixels that represent flares which extends beyond the solar disk what is usually impractical noise. Therefore, using the magnetograms to analyse Sun's activity (see Figure 1) seams to be more reasonable. This section contains description of the main steps of the hashing process.

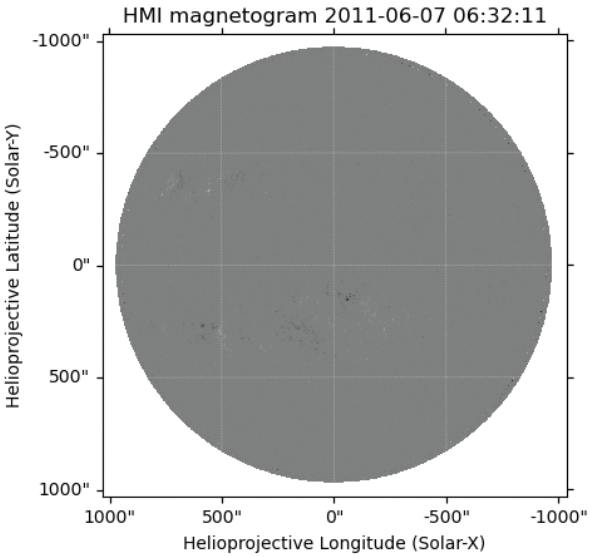

Figure 1. Example magnetogram image. As can be seen the image is difficult to analyze without the pre-processing.

2.1 Magnetic Region Detection

In the first step we adjust magnetogram image by annotating the magnetic regions more distinctively (see Figure 2).

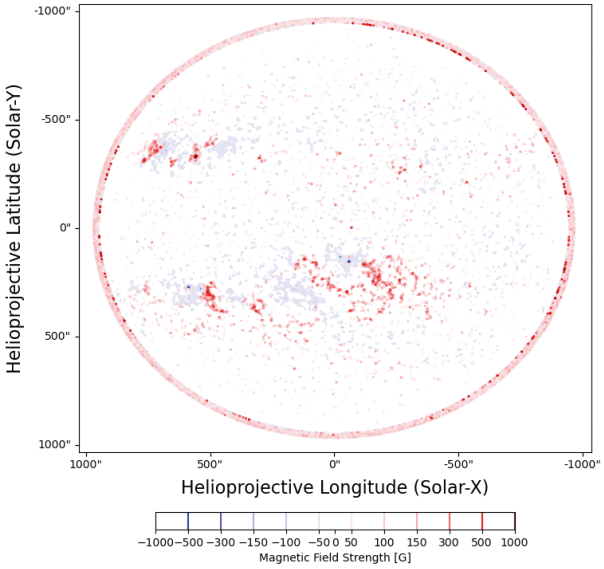

Figure 2. The magnetic region detection and annotation process. The magnetic regions can be clearly visible. We can observe the polarities (red and blue) and their intensities.

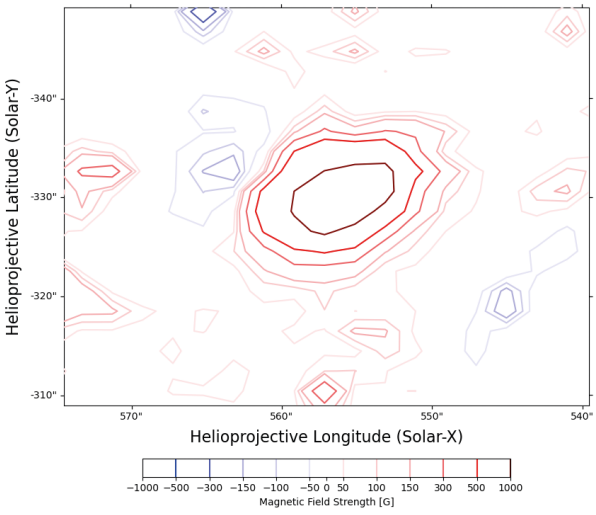

Figure 3. A magnification of magnetic regions.

This process is called magnetic region detection. We obtained the magnetogram images (Figure 1) by using the SunPy library [2, 3]. This step allows determining the strength of the mag-

netic field. As can be seen in Figure 1 and Figure 2 the strength of magnetic field increases around the active regions, and thus we can define the strength of the magnetic field as colour intensities (see Figure 3). The magnetic field can twist, tangle and reorganize itself during the solar cycle.

It should be noted that magnetic regions (MR) are highly correlated with CME's and thus solar flares. Therefore, analysis of them is important from the perspective of life on Earth. As can be seen in Figure 3, magnetic region detection (MRD) allow determining the polarities north (red) or south (blue). The Corona Mass Ejections are most likely to take their origin between these polarities. Moreover, tracking and analysing MRs is useful in predicting solar flares. The magnetic region detection provide data for the next steps of the algorithm.

2.2 Magnetic Field Grid-based Descriptor

This section describes the calculation of the Magnetic Field Grid-based Descriptor. This step is crucial, because processing of full disk images is computationally costly. Therefore, we need to define the mathematical description of the MRII (Magnetic Region Intensity Image) obtained in section 2.1. We applied a grid to MRII, which slices an image into several sub-images (cells). The grid size is based on the parameter N which controls the number of cells to be defined in x and y axis. We set the value of N empirically to 16. In the future work we intend to use optimization methods for determining this value. The descriptor calculation is composed of several stages. At first, we slice MRII into $N \times N$ number of cells. As a result we obtain a list of sub-images (cells). Afterwards, for every cell we calculate the sum of magnetic region areas in the given grid cell. This stage allows us to define a descriptor matrix (DM), which consist of the grid cell sums. For example: if a grid cell with id 11 have sum equal 523, then the $DM_{11} = 523$, etc. The last stage of this step, performs normalization and vectorization and as a result we obtain the descriptor vector (DV). The vectorization step simply concatenate all DM rows into one DV vector. Naturally, the size of the DV is correlated with the value of the N parameter. It should be noted that the N value has significant impact on the results. During the experiments we determined that value 16 for N provides the best results in the solar analysis tasks.

The entire process of this step has been presented in the form of pseudocode (Algorithm 1) and in Figure 4.

Algorithm 1. Algorithm for calculating grid-based descriptor.

INPUT: $MRII$ - magnetic region intensity image
N - grid size in x-axis and y-axis
OUTPUT: DV - magnetic grid based descriptor vector
Local Variables: $ImageCells$ - list for containing grid cells
DM - matrix sums of pixels in the cells
$HSlices := DivideIntoHorizontalSlices(MRII, N)$
foreach $HSlice \in HSlices$ **do**
  $CellsForHSlice :=$
  $DivideSlicesVerticallyIntoCells(HSlice, N)$
  foreach $CellForHSlice \in CellsForHSlice$ **do**
    $ImageCells.Add(CellForHSlice)$
    $CellSum :=$
    $CalcSumOfRegionPixelsInCell(ImageCell)$
    $SumMatrix.SetCellSum(CellSum)$
  end
end
$DV = VectorizeMatrix(DN)$

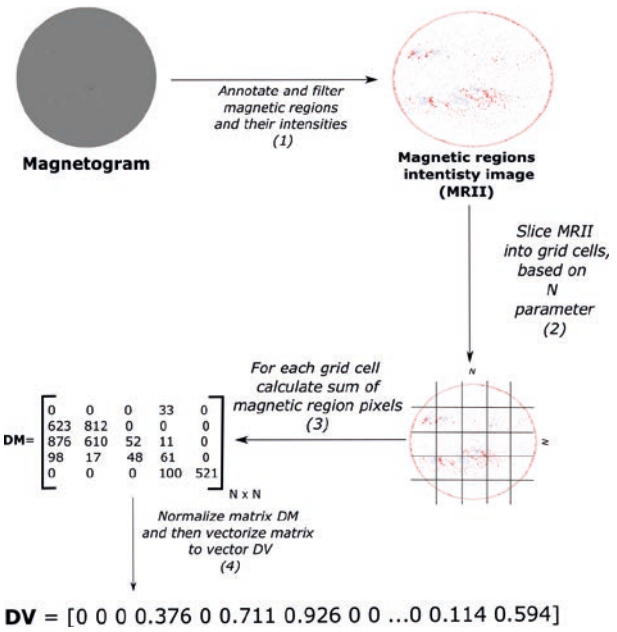

Figure 4. Steps for calculating the magnetic region grid-based descriptor.

The main aim of this stage was to obtain a handcrafted descriptor which will be used in the next steps of the method.

Table 1. Tabular representation of the fully-connected autoencoder model.

Layer (type)	Output	Filters (in, out)	Params
Input(*InputLayer*)	[1, 256]		0
*Linear*_1(*Linear*)	[1, 128]	256, 128	32896
*ReLU*_1	[1, 128]	0	
*Linear*_2(*Linear*)	[1, 64]	128, 64	8256
*ReLU*_2	[1, 64]	0	
*Linear*_3(*Linear*)	[1, 32]	64, 32	2080
*ReLU*_3	[1, 32]	0	
Encoded	[1, 32]		
*Linear*_4(*Linear*)	[1, 64]	32, 64	2112
*ReLU*_4	[1, 64]	0	
*Linear*_5(*Linear*)	[1, 128]	64, 128	8320
*ReLU*_5	[1, 128]	0	
*Linear*_6(*Linear*)	[1, 256]	128, 256	33024
*ReLU*_6	[1, 256]	0	
Decoded(*Tanh*)	[1, 256]		

2.3 Training and Hash Generation

This section contains the description of hash generation process. As input in this step we take a Magnetic Region Grid-based Descriptor (MRGD), which is later used for generating the corresponding hash. The aim of this process is to obtain the representative hash, which describes the corresponding solar image and more precisely the magnetic regions of the Sun in the given timestamp. This step is important because it allows to reduce the data in the retrieval stage (see Section 2.4). In order to perform this operation we used a fully-connected autoencoder (AE) to encode the previously obtained MRGD. The autoencoder are used in various machine learning tasks such as: image compression, dimensionality reduction, feature extraction, image reconstruction [4, 5, 6]. As autoencoders use unsupervised learning, they are perfect for generating semantic hashes. We present the autoencoder model architecture in Table 1. The AE model should be analysed from top to bottom. As can be seen the model is relatively simple but nevertheless allows reducing the hash length without significant loss of the information about magnetic regions of the magnetogram. We would like to emphasize that only the latent space (encoded) part of the trained AE is used for hash generation. The decoding part of AE is used only for training purposes. During a series of experiments we determined that 50 epochs are sufficient to obtain the satisfactory level of generalization without the overfitting phenomena.

2.4 Image Retrieval

In this section we present an application of our semantic solar hash. In this case, we use the hash for solar image retrieval task. We assumed that every magnetogram will have the corresponding hash, generated by the method presented in Section 2. When having such prepared solar image database, we can perform queries on it. We take the query image to generate its hash, and then we compare it with other semantic hashes stored in the database. This approach is one of the textbook methods in image retrieval. In order to perform the comparison operation we used the cosine distance measure. The cosine distance is calculated by the formula

$$\cos(Q_j, I_j) = \sum_{j=0}^{n} \frac{(Q_j \bullet I_j)}{\|Q_j\| \, \|I_j\|},$$

where $\bullet$ is the dot product. After distance calculation we obtain a list of distances D. The distances represent similarity of magnetograms stored in the database and the query image. The lower the distance value is, the greater similarity between the given magnetogram and the query. Based on that, we can sort distances, and easily obtain a list of magnetograms where the head of the list contains the most similar magnetograms. In the final step we take n first images from D list as the most similar images. These images are returned to the user as the retrieved (most similar) images. The entire retrieval process is presented in Figure 2.

Algorithm 2. Image retrieval steps.

INPUT: *Hashes*, *QueryImage*, *n*
OUTPUT: *RetrivedImages*
foreach $hash \in Hashes$ **do**
  $QueryImageHash =$
    $CalculateHash(QueryImage)$
  $D[i] = Cos(QueryImageHash, hash)$
end
$SortedDistances = SortAscending(D)$
$RetrivedHashes = TakeFirst(n)$
$RetrievedImages =$
  $GetCorrespondingImages(RetrievedHashes)$

3 Experimental Results

This section contains the simulation results and experimental methodology. The data that we obtained from HMI via SunPy does not contain any information about image similarity. Therefore, we had to propose a solution how to determine the image similarity. We used the Sun's rotation movement and magnetogram timestamps to define image similarity. We knew that HMI provides the continuum magnetogram images with one minute cadence. The changes between the consecutive images are minimal, if any at all. Based on that, we can assume that these images are similar. The two consecutive solar images should be slightly shifted. This phenomenon can be observed when we run magnetograms as slides. The main challenge was to determine the similarity window, how wide is the similarity window. After analysing the series of experiments, we concluded that 48-hours is the most suitable. Therefore, a magnetogram taken at a certain time has similar images 24 heours before and after. Based on that assumption, we can define similar images (SI), perform the experiments and evaluate the results. The methodology of our experiments is as follows:

1. Execute image query and obtain the retrieved images.
2. For every retrieved image, compare its timestamp with the query image timestamp.
3. If the timestamp is within a 48-hour window, the image is similar to the query.

The necessity of defining the similar images (SI) allows determine the well-known evaluation measures: *precision*, *recall*, $F-measure$ [7, 8]. These measures require defining the following sets:

- SI – set of similar images,
- RI – set of retrieved images for query,
- $PRI(TP)$ – set of positive retrieved images (true positive),
- $FPRI(FP)$ – false positive retrieved images (false positive),
- $PNRI(FN)$ – positive not retrieved images,
- $FNRI(TN)$ – false not retrieved images (TN).

We used the previously defined sets in order to adapt them to the following performance measures.

$$precision = \frac{|PRI|}{|PRI + FPRI|}, \tag{1}$$

$$recall = \frac{|PRI|}{|PRI + PNRI|}. \tag{2}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{3}$$

As can be seen in Table 2, the obtained simulation results are promising. The average value of F_1 proves the method effectiveness, moreover the assumptions concerning the higher accuracy seems to be true. We would like to emphasize the high values of the *precision* measure. The average precision is 0.9241, and Banda et al. [9] – 0.848, Angryk et al. [10] – 0.850. Moreover, we improved the results obtained previously by the authors [11]. The results also prove that most of the solar images with a close distance to the query image were successfully retrieved. Unfortunately, the magnetograms with higher distances can be classified as positive, not retrieved images (*PNRI*) although, this value is significantly reduced when compared to the previous works. The high values of *PNRI* are most likely caused by the Sun's rotation movement, which is responsible for magnetic regions were shifted or missing. Such cases can be observed even during the 48-hour window. Such described phenomena have significant impact on the obtained semantic hash and thus for the query results. The described case was observed during the experiments. We have concluded that lower values of *Recall* are caused by this phenomenon. The simulation environment was developed in Python language, SunPy PyTorch libraries, on the following hardware: Intel Core I9-9900k 3.6 GHz, 32 GB RAM, GeForce RTX 2080 Ti 11 GB, Windows Server 2016. The hash creation time took approximately 4 hour 10 minutes, for $525,600$ images. The learning stage took approximately 21 hours. The average retrieval time is 700 ms.

Table 2. Experiment results for retrieving solar magnetograms with the semantic hashes. Due to lack of space, we present only example queries.

Timestamp	RI	SI	PRI (TP)	FPRI (FP)	PNRI (FN)	Precision	Recall	F_1
2011-01-01 00:00:00	217	241	206	11	35	0.95	0.85	0.90
2011-01-07 18:00:00	436	481	394	42	87	0.90	0.82	0.86
2011-01-10 03:06:00	427	481	383	44	98	0.90	0.80	0.85
2011-01-15 15:12:00	425	481	399	26	82	0.94	0.83	0.88
2011-01-17 19:12:00	414	481	375	39	106	0.91	0.78	0.84
2011-01-23 10:18:00	401	481	385	16	96	0.96	0.80	0.87
2011-01-28 22:18:00	401	481	393	8	88	0.98	0.82	0.89
2011-02-03 21:18:00	400	481	393	7	88	0.98	0.82	0.89
2011-02-05 19:24:00	429	481	383	46	98	0.89	0.80	0.84
2011-02-11 07:30:00	396	481	389	7	92	0.98	0.81	0.89
2011-02-17 06:36:00	401	481	390	11	91	0.97	0.81	0.88
2011-02-24 10:42:00	442	481	393	49	88	0.89	0.82	0.85
2011-03-17 02:48:00	427	481	379	48	102	0.89	0.79	0.84
2011-03-22 13:48:00	420	481	396	24	85	0.94	0.82	0.88
2011-03-27 05:54:00	426	481	401	25	80	0.94	0.83	0.88
2011-03-30 21:00:00	437	481	396	41	85	0.91	0.82	0.86
2011-04-02 22:00:00	434	481	392	42	89	0.90	0.81	0.85
2011-04-08 15:00:00	432	481	395	37	86	0.91	0.82	0.86
2011-04-15 16:06:00	430	481	404	26	77	0.94	0.84	0.89
2011-04-20 19:12:00	426	481	389	37	92	0.91	0.81	0.86
2011-04-27 00:12:00	439	481	400	39	81	0.91	0.83	0.87
2011-04-30 23:18:00	424	481	389	35	92	0.92	0.81	0.86
2011-05-03 06:24:00	444	481	395	49	86	0.89	0.82	0.85
2011-05-11 05:24:00	400	481	392	8	89	0.98	0.81	0.89
2011-05-14 21:30:00	425	481	386	39	95	0.91	0.80	0.85
2011-06-30 06:48:00	398	481	386	12	95	0.97	0.80	0.88
2011-07-04 02:54:00	442	481	398	44	83	0.90	0.83	0.86
2011-07-05 09:00:00	435	481	395	40	86	0.91	0.82	0.86
2011-07-07 22:06:00	441	481	393	48	88	0.89	0.82	0.85
2011-07-09 07:06:00	431	481	386	45	95	0.90	0.80	0.85
2011-07-15 15:12:00	426	481	390	36	91	0.92	0.81	0.86
2011-07-19 22:12:00	429	481	387	42	94	0.90	0.80	0.85
Avg.						**0.9241**	**0.8127**	**0.8641**

Conclusions

We proposed a novel semantic hash for retrieving solar magnetograms similar to the query one. We use the SDO Helioseismic and Magnetic Imager output collected with SunPy PyTorch libraries. We created a mathematical representation of the magnetic regions of the Sun in the form of a vector. Thanks to this solution we can compare vectors of 32-length instead of comparing full-disk images. In order to increase the retrieval time, we used a fully-connected autoencoder, which reduced MRGD (256-element long) to a semantic hash (32-element long). The performed experiments and comparisons presented in Table 2 proved the efficiency of the proposed approach. Our approach has the highest precision value in comparison with other state-of-the-art methods. The presented method can be used not only for solar image retrieval but also for classification tasks. As the proposed algorithm uses magnetograms instead of Atmospheric Imaging Assembly images of the solar atmosphere in one or multiple wavelengths, the obtained semantic hash has a higher resistance to noise.

References

[1] R. Salakhutdinov and G. Hinton, Semantic hashing, International Journal of Approximate Reasoning, vol. 50, no. 7, pp. 969–978, 2009.

[2] The SunPy Community et al., The sunpy project: Open source development and status of the version 1.0 core package, The Astrophysical Journal, vol. 890, pp. 1–12, 2020. [Online]. Available: https://iopscience.iop.org/article/10.3847/1538-4357/ab4f7a

[3] Stuart Mumford, Nabil Freij et al., Sunpy: A python package for solar physics, Journal of Open Source Software, vol. 5, no. 46, p. 1832, 2020. [Online]. Available: https://doi.org/10.21105/joss.01832

[4] C. Brunner, A. Kő, and S. Fodor, An autoencoder-enhanced stacking neural network model for increasing the performance of intrusion detection, Journal of Artificial Intelligence and Soft Computing Research, vol. 12, no. 2, pp. 149–163, 2022.

[5] R. Grycuk, T. Galkowski, R. Scherer, and L. Rutkowski, A novel method for solar image retrieval based on the parzen kernel estimate of the function derivative and convolutional autoencoder, in 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022, pp. 1–7.

[6] P. Najgebauer, R. Scherer, and L. Rutkowski, Fully convolutional network for removing dct artefacts from images, in 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–8.

[7] M. Buckland and F. Gey, The relationship between recall and precision, Journal of the American society for information science, vol. 45, no. 1, p. 12, 1994.

[8] K. M. Ting, Precision and recall, in Encyclopedia of machine learning. Springer, 2011, pp. 781–781.

[9] J. M. Banda and R. A. Angryk, Regional content-based image retrieval for solar images: Traditional versus modern methods, Astronomy and computing, vol. 13, pp. 108–116, 2015.

[10] J. M. Banda and R. A. Angryk, Large-scale region-based multimedia retrieval for solar images, in International Conference on Artificial Intelligence and Soft Computing. Springer, 2014, pp. 649–661.

[11] R. Grycuk and R. Scherer, Grid-based concise hash for solar images, in International Conference on Computational Science. Springer, 2021, pp. 242–254.

Rafal Grycuk received his MSc and PhD degrees in computer science from Czestochowa University of Technology in 2012 and 2017 respectively. Currently he is an assistant professor at the Czestochowa University of Technology. His scientific interests cover computer vision, image description, content-based image retrieval and unsupervised learning.
https://orcid.org/0000-0002-3097-985X

Rafal Scherer is a full professor at the Czestochowa University of Technology, Poland. His research focuses on developing new methods in neural networks, computer vision, computational intelligence and data mining, ensembling methods in machine learning, content-based image indexing. He authored more than 160 research papers and two books: on multiple classification techniques (2012) and Computer Vision Methods for Fast Image Classification and Retrieval (2020) published by Springer. He co-organizes every year the International Conference on Artificial Intelligence and Soft Computing in Zakopane (http://www.icaisc.eu/) which is one of the major events on computational intelligence. Czestochowa University of Technology Al. Armii Krajowej 36, 42-200 Czestochowa, Poland.
https://orcid.org/0000-0001-9592-262X

Alina Marchlewska received the M.Sc. degree in mathematics from the Department of Mathematics and Computer Science, University of Łódź, Poland. She obtained the Ph.D. degree in mathematics from the same University. Currently, she is an assistant professor at the Social Academy of Sciences in Lodz. Her research interests include social computing and applications of various artificial intelligence technologies and computing methods in selected IT problems.
https://orcid.org/0000-0003-4386-7715

Christian Napoli is an associate professor with the Department of Computer, Control, and Management Engineering ``Antonio Ruberti'', Sapienza University of Rome, since 2019, where he also collaborates with the department of Physics and the Faculty of Medicine and Psychology, as well as holding the office of Scientific Director of the International School of Advanced and Applied Computing (ISAAC). He received the B.Sc. degree in Physics from the Department of Physics and Astronomy, University of Catania, in 2010, where he also got the M.Sc. degree in Astrophysics in 2012 and the Ph.D. in Computer Science in 2016 at the Department of Mathematics and Computer Science.

Christian Napoli has been a research associate with the Department of Mathematics and Computer Science, University of Catania, from 2018 to 2019, while, previously, Research Fellow and Adjunct Professor with the same department from 2015 to 2018. He was several times an invited professor at the Silesian University of Technology and visiting academic at the New York University. His current research interests include neural networks, artificial intelligence, human-computer interaction and computational neuropsychology.
https://orcid.org/0000-0002-3336-5853