Detecting the presence of meteors in images: new collection and results

Renato Moraes Silva[1], **Ana Carolina Lorena**[2], **Tiago A. Almeida**[1]

[1]Department of Computer Science, Federal University of São Carlos (UFSCar),
Sorocaba, São Paulo, Brazil

[2]Science and Technology Institute, Federal University of São Paulo (UNIFESP),
São José dos Campos, São Paulo, Brazil

renatoms@dt.fee.unicamp.br, aclorena@unifesp.br, talmeida@ufscar.br

Abstract. *In this paper, we present a new public and real dataset of labeled images of meteors and non-meteors that we recently used in a machine learning competition. We also present a comprehensive performance evaluation of several established machine learning methods and compare the results with a stacking approach – one of the winning solutions of the competition. We compared the performance obtained by the methods in the traditional repeated five-fold cross-validation with the ones obtained using the training and test partitions used in the competition. A careful analysis of the results indicates that, in general, the stacking based approach obtained the best performances compared to the baselines. Moreover, we found evidence that the validation strategy used by the platform that hosted the competition can lead to results that do not sustain in a cross-validation setup, which is recommendable in real-world scenarios.*

1. Introduction

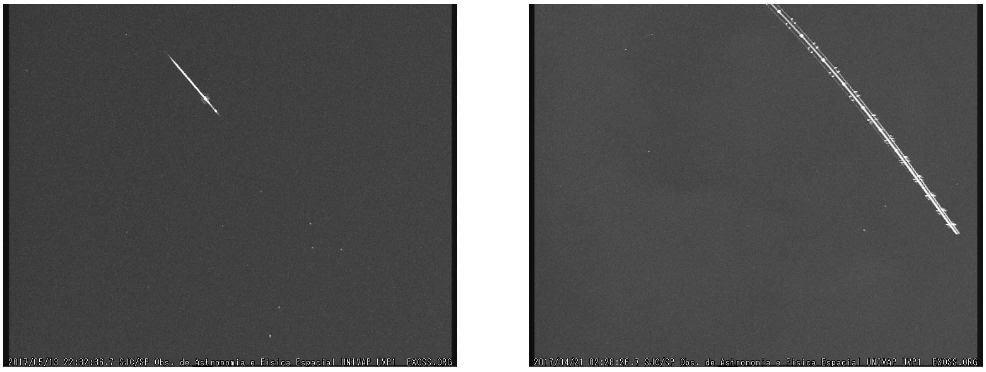
In 2017, the first edition of the KDD-BR (*Brazilian competition on Knowledge Discovery in Databases*) has launched a challenge on identifying the presence of meteors in images. The competition was one of the joint activities of the 2017 editions of the Brazilian Conference on Intelligent Systems (BRACIS), the Brazilian Symposium on Databases (SBBD) and the Symposium on Knowledge Discovery, Mining and Learning (KDMiLe). The objective was to obtain an automatic system for signalizing the presence of a meteor (popularly known as a shooting star) in an image. For this, we collected images captured by a monitoring station installed at the Observatory of Astronomy and Space Physics at University of Vale do Paraíba (UNIVAP), São José dos Campos, Brazil. This is one of the monitoring stations of the EXOSS Citizen Science organization[1]. EXOSS is a Brazilian non-profit organization which monitors meteors that cross the southern skies. It has about 50 active monitoring stations at various locations of the Brazilian territory.

The monitoring station consists of a low-cost video surveillance camera, usually used in security systems, with a dedicated motion capture software named *UFO Capture*[2]. Each time a moving object is detected by the camera during the night, it starts recording. The captured images include meteors, birds, insects, planes, lightnings, and other objects. There is a large variation regarding weather conditions, the presence of stars/moon in the sky, camera noise, among others. Therefore, the challenge is to distinguish between

[1] http://exoss.org

[2] http://sonotaco.com/e_index.html

meteors and non-meteors under varying conditions. Two examples of captured images are presented in Figure 1.


2017/05/13 22:32:36.7 SJC/SP Obs. de Astronomia e Fisica Espacial UNIVAP UVP1 EXOSS.ORG
2017/04/21 02:28:26.7 SJC/SP Obs. de Astronomia e Fisica Espacial UNIVAP UVP1 EXOSS.ORG

(a) Example of meteor.

(b) Example of non-meteor.

Figure 1. Examples of captured images.

The usual procedure adopted by the specialists is to inspect the images periodically in order to discard the non-meteors, whilst meteors are validated with another dedicated software. The validated images of meteors can be used for several purposes, as for instance, to identify their orbit or to figure out the region where the meteor fell, making it easier to find and collect pieces that can be studied later [Gural 2012]. Therefore, an automatic system may support this filtering process by classifying the recorded images into two classes: meteor vs non-meteor. However, labeled instances are mandatory to train these supervised models and, to the best of our knowledge, there are no labeled datasets of meteors publicly available. We are also not aware of any previous attempt to automate this image recognition task, although there are related work on meteor detection in video records [Silađi et al. 2015, Vítek and Nasyrova 2018] and radio spectrograms [Roman and Buiu 2015].

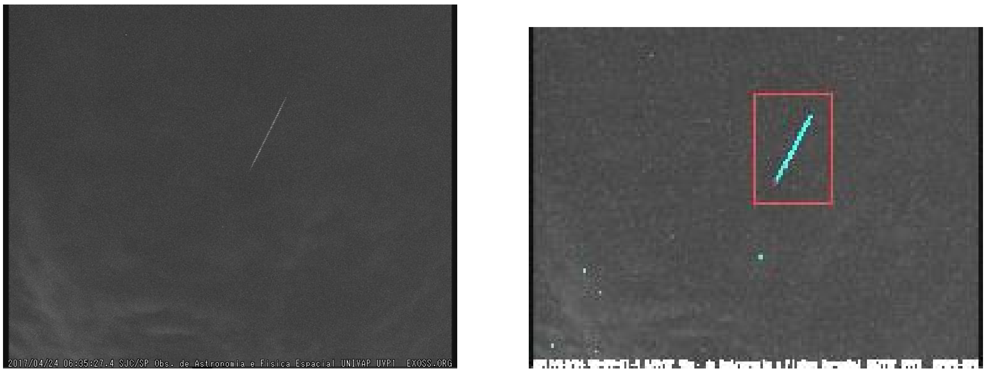
To fill this lack of public data, this paper presents a labeled dataset of meteors and a set of experiments performed which can be used as a baseline in future work. Several standard classifiers are evaluated and, afterward, their performances are compared to one of the best solutions submitted in the competition, which is based on feature transformations and stacking. Interestingly, the results on a repeated five-fold cross-validation experiment vary from those which are obtained using the original training and testing partitions of the competition. This reveals problems in the performance evaluation of such competition platforms, which only perform one holdout round. For a small dataset such as that used in this competition, this characteristic becomes more problematic.

This paper is structured as follows. Section 2 presents the meteor detection dataset used in the competition. Section 3 shows the configuration of the experiments performed in this study. The results are presented in Section 4. Finally, we give our main conclusions in Section 5.

2. Meteor dataset

The meteor dataset is composed by features extracted from images captured by a monitoring station located at the Observatory of Astronomy and Space Physics from the University of Vale do Paraíba (UNIVAP), in São José dos Campos, SP, Brazil. They were recorded during the months of April and March, 2017.

The *UFO Capture* software, which is used in the monitoring stations to capture meteors, stores a set of different files per recording: a movie file in the AVI format; an XML file with profile information; a bitmap file that contains mask and average brightness information; and two JPEG images with a composition of the video frames. The first JPEG file contains a peak hold or snapshot still image of the captured event (Figure 2a). The second file contains a thumbnail image, in which an area of interest where the moving object was detected is also highlighted (Figure 2b). The areas of interest in the corresponding snapshot are then used to figure out whether such image captured a meteor or not.


2017/04/24 06:35:27.4 SJC/SP Obs. de Astronomia e Física Espacial UNIVAP UVP1 EXOSS.ORG

(a) Snap shot of a meteor.

(b) Thumbnail image.

Figure 2. Examples of two JPEG images stored by *UFO Capture* per recording.

A total of 122 images (41 from meteors and 81 from non-meteors) were captured and labeled by specialists. This dataset has an imbalance ratio of 1.97 (ratio of the number of instances in the majority class to the number of examples in the minority class), which can be considered moderate [Fernández et al. 2008]. A large set of features was extracted from regions of interest using 21 image preprocessing techniques. Table 1 presents these techniques and the number of features generated by each one.

This dataset has challenging characteristics which can be used to test different capabilities of classification methods in future researches: (*i*) it is imbalanced, (*ii*) it contains noisy data, (*iii*) each sample is represented by 21 feature sets (totalling 3,451 features) that can be combined in several ways to test new classification approaches, and (*iv*) it contains few instances due to the low period of data collection and due to the difficulty of manually labeling the images.

A competition in the *Kaggle in class* platform[3] was configured using the proposed dataset. The competitors could opt to use all or some subsets of features in their solutions. The dataset was divided into two partitions (67-33%): training, with 26 meteors and 54 non-meteors; and testing, with 15 meteors and 27 non-meteors. The competitors had access only to the labels of the training data. The test partition was also randomly split into two other partitions of equal size. The participants received feedback about their performance on the first test partition (public leaderboard test set), while the performance in the

[3] https://www.kaggle.com/c/can-i-make-a-wish-detecting-shooting-stars

Table 1. Feature sets extracted from the images of the meteor dataset.

Id	Feature set	# features
FS_1	Auto color correlogram [Huang et al. 1997]	768
FS_2	Color and edge directivity descriptor [Chatzichristofis and Boutalis 2008a]	144
FS_3	Color histogram [Novak and Shafer 1992]	64
FS_4	Fuzzy color and texture histogram [Chatzichristofis and Boutalis 2008b]	192
FS_5	Fuzzy color histogram [Han and Ma 2002]	125
FS_6	Fuzzy opponent histogram [Van De Sande et al. 2010]	576
FS_7	Gabor [Fogel and Sagi 1989]	60
FS_8	Haralick [Haralick et al. 1973]	14
FS_9	Histogram [Scott 2010]	256
FS_{10}	Joint composite descriptor [Zagoris et al. 2010]	168
FS_{11}	JPEG coefficient Histogram [Sikora 2001]	192
FS_{12}	Luminance layout descriptor [Sikora 2001]	64
FS_{13}	MPEG7 color Layout [Sikora 2001]	33
FS_{14}	MPEG7 edge histogram [Sikora 2001]	80
FS_{15}	Mean intensity local binary patterns [Ojala et al. 1994]	256
FS_{16}	Mean patch intensity histogram [Taylor and Drummond 2011]	256
FS_{17}	Moments [Abo-Zaid et al. 1988]	4
FS_{18}	Opponent histogram [van de Sande et al. 2004]	64
FS_{19}	Pyramid histograms of oriented gradients [Bosch et al. 2007]	40
FS_{20}	Reference color similarity [Kriegel et al. 2011]	77
FS_{21}	Tamura [Tamura et al. 1978]	18
Total		**3,451**

second one (private leaderboard test set) remained secret until the end of the competition and was used for the final ranking of the solutions submitted based on log-loss.

In the following, we present the results obtained in experiments performed with this dataset using the same data partitions employed in the competition and also employing a repeated five-fold cross-validation on the whole dataset. The solution adopted by one of the winning teams is also compared under the two previous setups.

3. Experimental settings

In all experiments, we applied the Z-score normalization using information from the training examples. Moreover, we replaced missing values by the average of the corresponding feature in the training set.

As the dataset is imbalanced, we first performed experiments with the original class distribution and later using the same number of instances in each class of the training set. To balance the classes, we used the synthetic minority over-sampling technique (SMOTE) [Chawla et al. 2002]. We chose a method of oversampling rather than undersampling because of the small number of instances in the proposed dataset. An undersampling technique could affect the learning process of the classifiers due to the lack of training instances.

3.1. Evaluations

In order to provide a baseline result, we first compared the performance obtained by the following established classification methods: Gaussian naïve Bayes

(G.NB) [Metsis et al. 2006], logistic regression (LR) [Yu et al. 2011], support vector machines (SVM) [Boser et al. 1992, Cortes and Vapnik 1995], k-nearest neighbors (KNN) [Cover and Hart 1967], decision trees (DT) [Breiman et al. 1984], random forest (RF) [Breiman 2001], bootstrap aggregating (bagging) [Breiman 1996], and adaptive boosting (AdaBoost) [Freund and Schapire 1997]. These methods are widely used as baseline in several other studies.

We used the implementations of all methods from `scikit-learn library`[4] [Pedregosa et al. 2011]. As the performance of KNN, SVM, RF, bagging, and AdaBoost can be highly affected by the choice of parameters, we performed a grid search using five-fold cross-validation to find the best values for their main parameters. Table 2 presents the parameters and the range of values tested, where C is the cost for SVM, γ is the parameter of the RBF kernel, k corresponds to the number of neighbors for KNN, and $|E|$ is the number of estimators used in RF, bagging, and AdaBoost. For the other methods, we used their default values.

Table 2. Parameters and range of values used in the grid-search.

Method	Parameter	Range
SVM	Kernel C γ	{linear, RBF} $\{2^{-5}, 2^{-4}, ..., 2^{15}\}$ $\{2^{-15}, 2^{-14} ..., 2^{3}\}$
KNN	K	$\{5, 10, ..., 50\}$
RF	$\|E\|$	$\{10, 20, ..., 100\}$
Bagging	$\|E\|$	$\{10, 20, ..., 100\}$
AdaBoost	$\|E\|$	$\{10, 20, ..., 100\}$

To compare the results, we employed the following well-known performance measures: log-loss and F-measure [Sokolova and Lapalme 2009, Ferri et al. 2009].

4. Results

Table 3 shows the average results obtained by each evaluated method in ten runs of stratified five-fold cross-validation. In these experiments, each training and test example was represented by a feature vector generated by combining the feature sets presented in Table 1. The results are sorted by the log-loss. The scores are presented as a grayscale heat map in which the better the score, the darker the cell color. The bold values indicate the best scores.

The best F-measure was 0.90, and the median of all experiments was 0.86. These values indicate that the sets of features that were used to represent the examples of the proposed dataset are sufficiently informative to distinguish most of the images containing meteor from the ones that do not contain meteor.

SVM obtained the best log-loss in both experiments. However, in the experiment with the original class distribution, the best F-measure was obtained by RF. In the experiment using the SMOTE technique to balance the class distribution, bagging obtained the

[4] The `scikit-learn library` is available at: `http://scikit-learn.org/stable/index.html`

Table 3. Average results obtained by each method in ten runs of stratified five-fold cross-validation.

Method	Log-loss	F-measure
Original class distribution		
SVM	**0.272**	0.890
AdaBoost	0.345	0.853
LR	0.468	0.846
RF	0.598	**0.892**
Bagging	0.740	0.885
KNN	0.893	0.872
DT	4.229	0.815
G.NB	8.815	0.614
Balanced class distribution (SMOTE)		
SVM	**0.256**	0.884
AdaBoost	0.352	0.890
LR	0.475	0.839
RF	0.475	**0.900**
Bagging	0.596	0.874
KNN	1.237	0.797
DT	4.678	0.799
G.NB	8.837	0.612

best F-measure. The worst scores for all performance measures were obtained by G.NB, in both experiments.

The results also indicate the class balancing improved the performance of some methods but decreased the scores of others. For example, SVM, RF, and bagging obtained a better log-loss in the experiments using a balanced class distribution. However, AdaBoost, LR, KNN, DT, and G.NB obtained a better log-loss with the original class distribution. If we analyze the F-measure, we can see that the class balancing improved only the scores of AdaBoost and RF.

4.1. Competition

The proposed dataset was originally used in a competition hosted on the Kaggle platform, as described in Section 2. We evaluated the results of the classification methods using the same training and test partition of the competition to analyze whether the results are in line with the ones obtained using stratified five-fold cross-validation.

In the competition, the solutions were evaluated using the log-loss. Table 4 presents the results obtained on the public leaderboard test set. The results obtained on the private leaderboard test set are shown in Table 5. The results are sorted by the log-loss and bold values indicate the best scores. Moreover, the scores are presented as a grayscale heat map in which the better the score, the darker the cell color.

In the experiment with the public leaderboard test set (Table 4) of the competition, SVM lost three positions in the ranking of log-loss scores in relation to the scores obtained in the experiments based on five-fold cross-validation. On the other hand, KNN gained six positions in the ranking by getting the best log-loss in the experiment with balanced class distribution.

Table 4. Results obtained by each method on the public leaderboard test set.

Method	Log-loss	F-measure
Original class distribution		
LR	**0.293**	**0.889**
KNN	0.347	0.842
RF	0.420	**0.889**
SVM	0.449	**0.889**
AdaBoost	0.530	0.750
Bagging	1.849	**0.889**
DT	4.934	0.824
G.NB	11.513	0.462
Balanced class distribution (SMOTE)		
KNN	**0.262**	0.800
SVM	0.275	**0.889**
LR	0.291	**0.889**
Bagging	0.372	**0.889**
RF	0.457	**0.889**
AdaBoost	0.476	0.667
DT	4.934	0.824
G.NB	11.513	0.462

Table 5. Results obtained by each method on the private leaderboard test set.

Method	Log-loss	F-measure
Original class distribution		
RF	**0.190**	0.750
Bagging	0.198	0.750
AdaBoost	0.227	0.750
KNN	0.299	**0.889**
SVM	0.384	0.750
LR	1.449	0.667
DT	3.289	0.750
G.NB	4.934	0.667
Balanced class distribution (SMOTE)		
RF	**0.171**	0.750
Bagging	0.179	0.750
KNN	0.220	0.800
AdaBoost	0.253	**0.889**
SVM	0.408	0.750
LR	1.491	0.667
DT	3.289	0.750
G.NB	4.934	0.667

In the experiments with the private leaderboard test set (Table 5), the ranking of the methods was also different from the ranking obtained in the experiment based on five-fold cross-validation. Moreover, in general, the F-measure of the methods on the public leaderboard test set was better than the results on the private leaderboard test set. Based on log-loss, the best method of the experiments with the public leaderboard test set (KNN) was the third best in the experiments with the private leaderboard test set.

These differences indicate that the best method of a competition hosted on Kaggle may not be the best method to solve the same problem in a real-world application and the results are largely biased towards the specific data partition left for testing. In the case of a dataset with few samples such as the one used in this competition, this characteristic is highlighted since a single prediction error causes a significant impact on the final score of the method.

4.2. Comparison of baseline results with competition results

In this section, we present a comparison of the results obtained by the baseline methods with one of the winners of the competition. Briefly, one of the top rank solutions submitted is a stacking of LR. It uses a meta-classifier that is trained with the probabilities given by individual models. Each individual model is trained with the training data represented by one of the 21 feature sets presented in Table 1. Figure 3 presents an overview diagram of this approach[5].

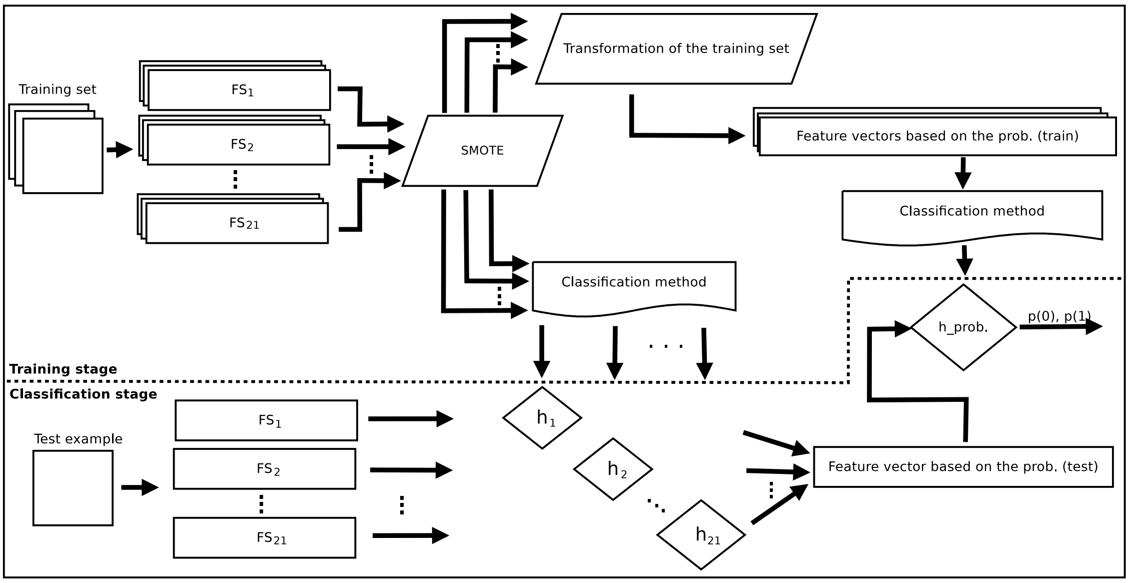


Figure 3. Overview diagram of the stacking approach that won the second place in the competition.

As shown in Figure 3, in the training stage, each training example is represented by 21 feature vectors ($\mathrm{FS}_1, \mathrm{FS}_2, ..., \mathrm{FS}_{21}$) based on the feature sets presented in Table 1. Afterwards, the SMOTE algorithm was applied to balance the class distribution. Then, one classification method (LR) was used to generate 21 predictive models ($h_1, h_2, ..., h_{21}$), one for each type of feature.

All feature vectors of the balanced training set are also presented to the module of *transformation of the training set*. This module performs n rounds of training and classification, where n is the number of examples in the training set. In each round, it creates 21 predictive models using LR, one for each set of feature vectors. In the j-th round, the j-th

[5] The source code is publicly available at `https://github.com/renatoms88/KDDBR`, accessed on Aug. 11, 2018.

training example is classified by the 21 models trained with the other examples. Then, a new feature vector is created with 21 dimensions, where the i-th element of the vector is the probability of the example being meteor given by the i-th predictive model. The new feature vectors generated by the module of *transformation of the training set* are used to train a classification method (LR) that generated a meta-classifier h_prob.

In the test stage, an unseen example is also represented by 21 feature vectors $(\mathrm{FS}_1, \mathrm{FS}_2, ..., \mathrm{FS}_{21})$ based on the feature sets presented in Table 1. The i-th feature vector is presented to the predictive model h_i. Then, a new feature vector is created, where the i-th element is the probability of the example being meteor given by the i-th model. This new feature vector is classified by the meta-classifier h_prob that returns the value of $p(0)$ (probability of the example not being a meteor) and $p(1)$ (probability of the example being a meteor).

Table 6 shows the results obtained by the approach presented in Figure 3. We present the average results obtained in ten runs of stratified five-fold cross-validation. We also present the performance obtained by the approach using the same training and test partition of the competition.

Table 6. Results obtained by the stacking approach.

Validation	Log-loss	F-measure
Ten runs of stratified five-fold	0.313	0.840
Public leaderboard test set	0.351	0.889
Private leaderboard test set	0.072	0.889

In the experiments based on five-fold cross-validation, the stacking approach obtained the second best log-loss. Moreover, the F-measure was smaller than the one obtained by SVM, AdaBoost, RF, bagging, LR, and KNN.

In the experiment with the private leaderboard test set, the log-loss of the stacking approach was significantly better than the log-loss obtained by all other methods. The best log-loss obtained by RF was 0.171 (Table 5), while the log-loss obtained by the stacking approach was 0.072 (Table 6). The F-measure of the stacking approach was also the best value in the experiments. However, in the experiment with the public leaderboard, the log-loss obtained by KNN, SVM, and LR were better than that obtained by the stacking approach.

5. Conclusions

In this paper, we present a new dataset of meteors that was first used in the 1st KDD-BR competition, along with a set of experiments. This dataset has a big potential to become a baseline because it has challenging characteristics which can be used to test different capabilities of classification methods in future researches, such as: imbalanced data, presence of noise, high dimensionality data (21 feature sets, totalling 3,451 features), and few instances. We presented a comprehensive performance evaluation of different established machine learning algorithms on this dataset. The results obtained were compared with one of the winning approaches of the competition.

In general, the scores obtained by the evaluated methods were high, which indicates that the features used to represent the data are informative. We also observed that balancing the class distribution using the SMOTE technique, in general, did not improve the classification results. Moreover, the stacking approach – one of the best solutions in the competition – was the best evaluated method in this paper, since it obtained the best scores in the experiments with the training and testing partitions of the competition, and obtained the second best log-loss in the experiments with repeated five-fold cross-validation.

The results obtained in ten runs of a five-fold cross-validation were different from the ones obtained using the original training and testing partitions of the competition. These differences reveal that the validation strategy used by the Kaggle platform may not represent properly real-world scenarios, mainly when the number of samples is few.

Acknowledgments

We gratefully acknowledge the financial support from the Brazilian agencies FAPESP (grants #2012/22608-8, #2017/09387-6, #2018/02146-6) and CAPES (grant #1797700). We would also like to thank the professionals Daniel dos Santos Kaster, Elaine Ribeiro de Faria, Francisco Carlos Rocha Fernandes, Irapuan Rodrigues, Jennifer Nielsen, Ricardo Cerri, and Vinicius Veloso de Melo who were also involved in the organization of the competition.

References

Abo-Zaid, A., Hinton, O. R., and Horne, E. (1988). About moment normalization and complex moment descriptors. In *Pattern Recognition*, pages 399–409. Springer.

Bosch, A., Zisserman, A., and Munoz, X. (2007). Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory (COLT'92)*, pages 144–152, Pittsburgh, PA, USA. ACM.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth International Group, Belmont, California, USA.

Chatzichristofis, S. A. and Boutalis, Y. S. (2008a). Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *International Conference on Computer Vision Systems*, pages 312–322. Springer.

Chatzichristofis, S. A. and Boutalis, Y. S. (2008b). Fcth: Fuzzy color and texture histogram-a low level feature for accurate image retrieval. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on*, pages 191–196. IEEE.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357.

Cortes, C. and Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Cover, T. M. and Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transaction on Information Theory*, 13(1):21–27.

Fernández, A., García, S., del Jesus, M. J., and Herrera, F. (2008). A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems*, 159(18):2378–2398.

Ferri, C., Hernández-Orallo, J., and Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1):27–38.

Fogel, I. and Sagi, D. (1989). Gabor filters as texture discriminator. *Biological cybernetics*, 61(2):103–113.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Gural, P. S. (2012). A new method of meteor trajectory determination applied to multiple unsynchronized video cameras. *Meteoritics & Planetary Science*, 47(9):1405–1418.

Han, J. and Ma, K.-K. (2002). Fuzzy color histogram and its use in color image retrieval. *IEEE transactions on image processing*, 11(8):944–952.

Haralick, R. M., Shanmugam, K., et al. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621.

Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., and Zabih, R. (1997). Image indexing using color correlograms. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 762–768. IEEE.

Kriegel, H.-P., Schubert, E., and Zimek, A. (2011). Evaluation of multiple clustering solutions. In *MultiClust@ ECML/PKDD*, pages 55–66.

Metsis, V., Androutsopoulos, I., and Paliouras, G. (2006). Spam filtering with naive Bayes – which naive Bayes? In *Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS'06)*, pages 27–28, Mountain View, California.

Novak, C. L. and Shafer, S. A. (1992). Anatomy of a color histogram. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 599–605. IEEE.

Ojala, T., Pietikainen, M., and Harwood, D. (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582–585. IEEE.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Roman, V. Ş. and Buiu, C. (2015). Automatic detection of meteors in spectrograms using artificial neural networks. In *Applied Computational Intelligence and Informatics (SACI), 2015 IEEE 10th Jubilee International Symposium on*, pages 131–134. IEEE.

Scott, D. W. (2010). Averaged shifted histogram. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):160–164.

Sikora, T. (2001). The mpeg-7 visual standard for content description-an overview. *IEEE Transactions on circuits and systems for video technology*, 11(6):696–702.

Silađi, E., Vida, D., and Nyarko, E. K. (2015). Video meteor detection filtering using soft computing methods. In *Proceedings of the International Meteor Conference, Mistelbach, Austria*, pages 27–30.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Tamura, H., Mori, S., and Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, man, and cybernetics*, 8(6):460–473.

Taylor, S. and Drummond, T. (2011). Binary histogrammed intensity patches for efficient and robust matching. *International journal of computer vision*, 94(2):241–265.

Van De Sande, K., Gevers, T., and Snoek, C. (2010). Evaluating color descriptors for object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1582–1596.

van de Sande, K. E., Gevers, T., and Snoek, C. G. (2004). Evaluation of color descriptors for object and scene recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA (June 2008)*.

Vítek, S. and Nasyrova, M. (2018). Real-time detection of sporadic meteors in the intensified tv imaging systems. *Sensors*, 18(1):77.

Yu, H.-F., Huang, F.-L., and Lin, C.-J. (2011). Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1-2):41–75.

Zagoris, K., Chatzichristofis, S. A., Papamarkos, N., and Boutalis, Y. S. (2010). Automatic image annotation and retrieval using the joint composite descriptor. In *Informatics (PCI), 2010 14th Panhellenic Conference on*, pages 143–147. IEEE.