-----------------------

Div150Cred - Dataset readme

-----------------------

----------

\*\*Authors:

Bogdan Ionescu, LAPI, University Politehnica of Bucharest, Romania(bogdanLAPI@gmail.com);

Adrian Popescu, CEA LIST, France (adrian.popescu@cea.fr);

Mihai Lupu, Vienna University of Technology, Austria (lupu@ifs.tuwien.ac.at);

Henning Müller, University of Applied Sciences Western Switzerland (HES-SO) in Sierre, Switzerland (henning.mueller@hevs.ch).

This dataset was supported by the following projects: MUCKE, CUbRIK and PROMISE.

Many thanks to Alexandru Lucian Gînscă, Adrian Iftene, Bogdan Boteanu, Ioan Chera, Ionuț Duță, Andrei Filip, Corina Macovei, Cătălin Mitrea, Ionuț Mironică, Irina Emilia Nicolae, Ivan Eggel, Andrei Purică, Mihai Pușcaș, Oana Pleș, Gabriel Petrescu, Anca Livia Radu, Vlad Ruxandu for their help.

----------

\*\*Citation:

B. Ionescu, A. Popescu, M. Lupu, A.L. Gînscă, H. Müller, “Retrieving Diverse Social Images at MediaEval 2014: Challenge, Dataset and Evaluation”, MediaEval Benchmarking Initiative for Multimedia Evaluation, vol. 1263, CEUR-WS.org, ISSN: 1613-0073, October 16-17, Barcelona, Spain, 2014.

----------

\*\*Description:

The data consists of a development set containing 30 locations (devset), a user annotation credibility set containing information for ca. 300 locations and 685 users (credibilityset) and a test set containing 123 locations (testset).

Data was retrieved from Flickr using the name of the location as query.

For each location in devset and testset, the following information is provided:

- location name: is the name of the location and represents its unique identifier;

- location name query id: each location name has an unique query id code to be used for preparing the official runs at MediaEval benchmark (i.e., numbers from 1 to 153 - the total number of locations; numbers from 1 to 30 are belonging to the devset locations and the rest to the testset locations);

- GPS coordinates: latitude and longitude in degrees;

- link to the Wikipedia webpage of the location;

- up to 5 representative photos retrieved from Wikipedia in jpeg format;

- a set of photos retrieved from Flickr in jpeg format (up to 300 photos per location - each photo is named according to its unique id from Flickr). Photos are stored in individual folders named after the location name;

- an xml file containing metadata from Flickr for all the retrieved photos;

- visual, text and credibility descriptors;

- ground truth for both relevance and diversity.

---Important--: please note that all the photos provided are under Creative Commons licenses (for more information see the previous link and also the information on Flickr website). Each Flickr photo is provided with the license type and the owner’s name. For the Wikipidia photos the owner name is included in the photo file name, e.g., for "acropolis\_athens(Ricardo André Frantz).jpg" the owner name is Ricardo André Frantz.

----------

\*\*XML metadata

Each location in devset and testset is accompanied by an xml file (UTF-8 encoded) that contains all the retrieved metadata for all the photos. Each file is named after the location name, e.g., “acropolis\_athens.xml”. The information is structured as follows:

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<photos monument="acropolis athens">

<photo date\_taken="2013-06-04 02:45:20" description="View of Athens from the entrance of Acropolis" id="9067739127" latitude="37.970805" license="2" longitude="23.721167" nbComments="0" rank="1" tags="athens greece" title="Acropolis - Athens" url\_b="http://farm8.static.flickr.com/7362/9067739127\_edda2711ca\_b.jpg" username="pfischermx" userid="56505984@N06" views="70"/>

...

</photos>

The monument value is the location query used to retrieve data from Flickr. Then, each of the photos are delimited by a <photo /> statement. Among the photo information fields, please note in particular:

- description contains a detailed textual description of the photo as provided by author;

- id is the unique identifier of each photo from Flickr and corresponds to the name of the jpeg file associated to this photo (e.g., “9067739127.jpg”);

- license is the Creative Common license of this picture;

- nbComments is the number of comments posted on Flickr about this photo;

- rank is the position ranking of the photo in the list retrieved from Flickr (a generated number from 1 to the number of photos);

- tags are the tag keywords used for indexing purpose;

- title is a short textual description of the photo provided by the author;

- url\_b is the url link of the photo location from Flickr (please note that by the time you use the dataset some of the photos may not be available anymore at the same location);

- username represent the photo owner’s name;

- userid is the unique user id from Flickr;

- views is the number of times the photo has been displayed on Flickr.

----------

\*\*Visual descriptors

For each location and photo, the dataset contains some general purpose visual descriptors, namely:

- Global Color Naming Histogram (code CN - 11 values): maps colors to 11 universal color names: "black", "blue", "brown", "grey", "green", "orange", "pink", "purple", "red", "white", and "yellow" [v1];

- Global Histogram of Oriented Gradients (code HOG - 81 values): represents the HoG feature computed on 3 by 3 image regions [v2];

- Global Color Moments on HSV Color Space (code CM - 9 values): represent the first three central moments of an image color distribution: mean, standard deviation and skewness [v3];

- Global Locally Binary Patterns on gray scale (code LBP - 16 values) [v4];

- Global Color Structure Descriptor (code CSD - 64 values): represents the MPEG-7 Color Structure Descriptor computed on the HMMD color space [v5];

- Global Statistics on Gray Level Run Length Matrix (code GLRLM – 44 dimensions): represents 11 statistics computed on gray level run-length matrices for 4 directions: Short Run Emphasis (SRE), Long Run Emphasis (LRE), Gray-Level Non-uniformity (GLN), Run Length Non-uniformity (RLN), Run Percentage (RP), Low Gray-Level Run Emphasis (LGRE), High Gray-Level Run Emphasis (HGRE), Short Run Low Gray-Level Emphasis (SRLGE), Short Run High Gray-Level Emphasis (SRHGE), Long Run Low Gray-Level Emphasis (LRLGE), Long Run High Gray-Level Emphasis (LRHGE) [v6];

- Spatial pyramid representation (code 3x3): each of the previous descriptors is computed also locally. The image is divided into 3 by 3 non-overlapping blocks and descriptors are computed on each patch. The global descriptor is obtained by the concatenation of all values.

File format. Visual descriptors are provided on a per location basis. We provide individual csv (comma-separated values) files for each type of visual descriptor and for each location. The naming convention is the following: location name followed by the descriptor code, e.g., “acropolis\_athens CM3x3.csv” refers to the Global Color Moments (CM) computed on the spatial pyramid (3x3) for the location acropolis\_athens. Each file contains the descriptors for each of the photos of the location on one line. The first value of each line is the unique photo id followed by the descriptor values separated by commas. Lines are separated by an end-of-line character (carriage return). An example is presented below:

3338743092,0.51934475780470812,0.40031641870181739,...

3338745530,0.20630411506897756,0.26843536114050304,...

3661394189,0.47248077522064869,0.17833862284689939,...

...

In particular, we provide the visual descriptors also for the Wikipedia photos. The descriptors are provided using the same format as for the location photos. The only difference is that the first value on each line is the photo name:

acropolis\_athens(Christophe Meneboeuf),0.259505,0.218414,...

acropolis\_athens(Gfmichaud),0.348293,0.40948,...

acropolis\_athens(Joanbanjo),0.370256,0.401351,...

...

----------

\*\*Textual descriptors

Text descriptors are provided on per dataset basis. For each set (i.e., devset, testset, or the combination of the two), the text descriptors are computed on: per image basis (file [dataset]\_textTermsPerImage.txt), per location basis (file [dataset]\_textTermsPerPOI.txt) and per user basis, respectively (file [dataset]\_textTermsPerUser.txt).

File format. In each file, each line represents an entity with its associated terms and their weights. For instance, in the devset per image basis descriptor file (devset\_textTermsPerImage.txt) a line will look like:

9067739127 "acropoli" 2 299 0.006688963210702341 "athen" 3 304 0.009868421052631578 "entrance" 1 130 0.007692307692307693 "greece" 1 257 0.0038910505836575876 "view" 1 458 0.002183406113537118

...

The first token is the id of the entity, in this case the unique Flickr id of the image. Following that is a list of 4-tuples ("term" TF DF TF-IDF), where "term" is a term which appeared anywhere in the description, tags or title of the image from the metadata, TF is the term frequency (the number of occurrences of the term in the entity's text fields), the DF is the document frequency (the number of entities which have this term in their text fields) and finally the TF-IDF is simply TF/DF [t1]. The information from the location-based text descriptors is the same as in the image-based case except for the fact that the entity here is the location query. Its textual description is taken to be the set of all texts of all of its images. Additionally, in this case we provide also a set of files which include also the location name apart from the location query (file [dataset]\_textTermsPerPOI.wFolderNames.txt). Here is an example where "acropolis\_athens" is the location name and "acropolis athens" is the location query:

acropolis\_athens acropolis athens "0005" 1 3 0.3333333333333333 "0006" 1 3 0.3333333333333333 "0012" 1 2 0.5

...

The information from the user-based text descriptors is also similar except for the fact that the entity here is the photo user id from Flickr. Its textual description is taken to be the set of all texts of all of her images, regardless of the location.

SOLR Indexes. The term lists provided and described above were generated using Solr 4.7.1. To make it easier to get a baseline system for text retrieval, we also provide all the details to get your own Solr server running, containing all the data necessary for retrieving images, out of the box. First download Solr from http://lucene.apache.org/. We have used version 4.7.1 in generating the files here, but it should also work with 4.7.2 - the latest version available at the time of writing. The download comes with an example folder. Replace the solr folder inside the examples folder with the provided one and start solr as indicated in the tutorial. You will be able to access it at localhost:8983.

Additionally, we provide a data folder which has all the data provided, but in a format ingestible by solr and which can be used with the post2solr.sh script to generate new indexes, with different pre-processing steps or similarity functions. We also provide the mapping between the location queries to the location names (which correspond to the folder names) and scripts that we have been used to generate the [dataset]\_textPerPOI.wFolderNames.txt files, in case the participants wish to recreate their own. All scripts are Bash scripts, so they should run in most \*x systems, but not under Windows.

----------

\*\*User annotation credibility descriptors

We provide user tagging credibility descriptors that give an automatic estimation of the quality of tag-image content relationships. The aim of credibility descriptors is to give an indication about which users are most likely to share relevant images in Flickr (according to the underlying task scenario). These descriptors are extracted by visual or textual content mining:

- visualScore: descriptor obtained through visual mining using over 17,000 of ImageNet visual models obtained by learning a binary SVM per ImageNet concept. Visual models are built on top of overfeat, a powerful convolutional Neural Network feature [c1]. At most 1,000 images are downloaded for each user in order to compute visualScores. For each Flickr tag which is identical to an ImageNet concept, a classification score is predicted and the visualScore of a user is obtained by averaging individual tag scores. The intuition here is that the better the predictions given by the classifiers are, the more relevant a user’s images should be. Scores are normalized between 0 and 1, with higher scores corresponding to more credible users;

- faceProportion: descriptor obtained using the same set of images as for visualScore. The default face detector from OpenCV [c2] is used here to detect faces. faceProportion, the percentage of images with faces out of the total of images tested for each user is computed. The intuition here is that the lower faceProportion is, the better the average relevance of a user’s photos is. faceProportion is normalized between 0 and 1, with 0 standing for no face images;

- tagSpecificity: descriptor obtained by computing the average specificity of a user’s tags. Tag specificity is calculated as the percentage of users having annotated with that tag in a large Flickr corpus (~100 million image metadata from 120,000 users);

- locationSimilarity: descriptor obtained by computing the average similarity between a user's geotagged photos and a probabilistic model of a surrounding cell of approximately 1 km2 geotagged images. These models were created for MediaEval 2013 Placing Task [c3] and reused as such here. The intuition here is that the higher the coherence between a user’s tags and those provided by the community is, the more relevant her images are likely to be. locationSimilarity is not normalized and small values stand for the lowest similarity;

- photoCount: descriptor which accounts for the total number of images a user shared on Flickr. This descriptor has a maximum value of 10,000;

- uniqueTags: proportion of unique tags present in a user's vocabulary divided by the total number of tags of the user. uniqueTags ranges between 0 and 1;

- uploadFrequency: average time between two consecutive uploads in Flickr. This descriptor is not normalized;

- bulkProportion: the proportion of bulk taggings in a user’s stream (i.e., of tag sets which appear identical for at least two distinct photos). The descriptor is normalized between 0 and 1.

File format. Descriptors are provided on a per user basis. We provide information for a significant number of users (the exact numbers are provided with each dataset in particular). We provide separate XML files for each user and in each file we include separate fields for the credibility descriptors enumerated above. XML files have the following format:

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<metadata user="21953562@N07">

<credibilityDescriptors>

<visualScore>0.791442635512724</visualScore>

<faceProportion>0.013</faceProportion>

<tagSpecificity>0.624967978500227</tagSpecificity>

<locationSimilarity>1.52020128875995</locationSimilarity>

<photoCount>6710</photoCount>

<uniqueTags>0.05555555555555555</uniqueTags>

<uploadFrequency>395.91869026284303</uploadFrequency>

<bulkProportion>0.8785394932935916</bulkProportion>

</credibilityDescriptors>

<photos>

<photo date\_taken="2013-08-19 14:11:49" id="9659825826" latitude="42.36115" longitude="-71.03523" tags="boston nhl massachusetts suffolkcounty nationalhistoriclandmark nationalregisterofhistoricplaces nrhp lightshipnantucket unitedstateslightshipnantucketlv112 lightshipno112" title="United States Lightship Nantucket (LV-112)" url\_b="http://farm8.static.flickr.com/7408/9659825826\_55cb51182d\_b.jpg" userid="21953562@N07" views="533" />

...

</photos>

</metadata>

User annotation credibility descriptors are separated by <credibilityDescriptors> </credibilityDescriptors> statements. In addition, to facilitate to participants the possibility of creating their own credibility descriptors, we provide Flickr metadata for a relevant number of images uploaded by these users. These data are separated by <photos> </photos> statements and are structured similarly to the XML photo metadata presented in the XML metadata section above (each photo is separated by a <photo /> statement and provided information include id, tags, title, url\_b, userid, views, etc).

----------

\*\*Topic files

For each dataset we provide a topic file that contains the list of the locations in the current dataset. Each location is delimited by a <topic> </topic> statement and includes the query id code (delimited by a <number> </number> statement), the name of the location (delimited by a <title> </title> statement), the GPS coordinates (latitude and longitude in degrees) and the url to the Wikipedia webpage (delimited by a <wiki> </wiki> statement). An example is presented below:

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<topics>

<topic>

<number>26</number>

<title>Abbey of Saint Gall</title>

<latitude>47.423056</latitude>

<longitude>9.377222</longitude>

<wiki>http://en.wikipedia.org/wiki/Abbey\_of\_Saint\_Gall</wiki>

</topic>

...

</topics>

----------

\*\*Ground truth

The ground truth data consists on relevance ground truth (code rGT) and diversity ground truth (code dGT and dclusterGT). Ground truth was generated by a small group of expert annotators with advanced knowledge of location characteristics (mainly learned from Internet sources). Each type of ground truth consisted on a different protocol and followed the exact definitions adopted for this scenario.

Relevance ground truth was annotated using a dedicated tool that provided the annotators with one photo at a time. A reference photo of the location could be also displayed during the process. Annotators were asked to classify the photos as being relevant (score 1), non-relevant (score 0) or with “don’t know” answer (score -1). The definition of relevance was available to the annotators in the interface during the entire process. The annotation process was not time restricted. Annotators were recommended to consult any additional information source about the characteristics of the location (e.g., from Internet) in case they were unsure about the annotation.

Ground truth was collected from several annotators and final ground truth was determined after a lenient majority voting scheme.

File format. Ground truth is provided on a per location basis. We provide individual txt files for each location. Files are named according to the location name followed by the ground truth code, e.g., “Abbey of Saint Gall rGT.txt” refers to the relevance ground truth (rGT) for the location Abbey of Saint Gall. Each file contains photo ground truth on individual lines. The first value of each line is the unique photo id followed by the ground truth value separated by comma. Lines are separated by an end-of-line character (carriage return). An example is presented below:

3338743092,1

3338745530,0

3661394189,1

...

Diversity ground truth was also annotated with a dedicated tool. The diversity is annotated only for the photos that were judged as relevant in the previous step. For each location, annotators were provided with a thumbnail list of all the relevant photos. The first step required annotators to get familiar with the photos by analyzing them for about 5 minutes. Next, annotators were required to re-group the photos into similar visual appearance clusters. Full size versions of the photos were available by clicking on the photos. The definition of diversity was available to the annotators in the interface during the entire process. For each of the clusters, annotators provided some keyword tags reflecting their judgments in choosing these particular clusters. Similar to the relevance annotation, the diversity annotation process was not time restricted.

In this particular case, ground truth was collected from several annotators that annotated distinct parts of the data set.

File format. Ground truth is provided on a per location basis. We provide two individual txt files for each location: one file for the cluster ground truth and one file for the photo diversity ground truth. Files are named according to the location name followed by the ground truth code, e.g., “Abbey of Saint Gall dclusterGT.txt” and “Abbey of Saint Gall dGT.txt” refer to the cluster ground truth (dclusterGT) and photo diversity ground truth (dGT) for the location Abbey of Saint Gall.

In the dclusterGT file each line corresponds to a cluster where the first value is the cluster id number followed by the cluster user tag separated by comma. Lines are separated by an end-of-line character (carriage return). An example is presented below:

1,outside statue

2,inside views

3,partial frontal view

4,archway

...

In the dGT file the first value on each line is the unique photo id followed by the cluster id number (that corresponds to the values in the dclusterGT file) separated by comma. Each line corresponds to the ground truth of one image and lines are separated by an end-of-line character (carriage return). An example is presented below:

3664415421,1

3665220244,1

...

3338745530,2

3661396665,2

...

3662193652,3

3338743092,3

3665213158,3

...

----------

\*\*MediaEval submission format

The following information will help reproducing the exact evaluation conditions of the MediaEval task. At MediaEval runs were provided in the form of a trec topic file. This file is compatible with the trec\_eval evaluation software (for more information please follow the previous link – you will find two archives trec\_eval.8.1.tar.gz and trec\_eval\_latest.tar.gz - see the README file inside). The trec topic file has the structure illustrated by the following example of a file line (please note that values are separated by whitespaces):

030 Q0 ZF08 0 4238 prise1

qid iter docno rank sim run\_id

where:

-qid is the unique query id (please note that each location name has a certain query id code that is provided with the data set in the topic xml files);

-iter – is ignored;

-docno – is the unique photo id (as provided with the data set);

-rank – is the photo rank in the refined list provided by your method. Rank is expected to be an integer value ranging from 0 (the highest rank) up to 49;

-sim – is the similarity score of your photo to the query and is mandatory for the submission. The similarity values need to be higher for the photos to be ranked first and should correspond to your refined ranking (e.g., the photo with rank 0 should have the highest sim value, followed by photo with rank 1 with the second highest sim value and so on). In case your approach do not provide explicitly similarity scores (e.g., crowd-sourcing) you are required to create dummy similarity scores that decrease when the rank increases (e.g., in this case, you may use the inverse ranking values);

-run\_id - is the name of your run (which you can choose, but should be as informative as possible without being too long – please note that no whitespaces or other special characters are allowed);

Please note that each run needs to contain at least one result for each location. An example of results file should look like this:

1 0 3338743092 0 0.94 run1\_audiovisualRF

1 0 3661411441 1 0.9 run1\_audiovisualRF

...

1 0 7112511985 48 0.2 run1\_audiovisualRF

1 0 711353192 49 0.12 run1\_audiovisualRF

2 0 233474104 0 0.84 run1\_audiovisualRF

2 0 3621431440 1 0.7 run1\_audiovisualRF

...

----------

\*\*Scoring tool

The official MediaEval scoring tool is div\_eval.jar. It computes cluster recall at X (CR@X --- a measure that assesses how many different clusters from the ground truth are represented among the top X results), precision at X (P@X --- measures the number of relevant photos among the top X results) and their harmonic mean, i.e., F1-measure@X (X in {5,10,20,30,40,50}).

The software tool was developed under Java and to run it you need to have Java installed on your machine. To check, you may run the following line in a command window: "java -version". In case you don't have Java installed, please visit this link, download the Java package for your environment and install it.

To run the script, use the following syntax (make sure you have the div\_eval.jar file in your current folder):

java -jar div\_eval.jar -r <runfilepath> -rgt <rGT directory path> -dgt <dGT directory path> -t <topic file path> -o <output file directory> [optional: -f <output file name>]

where:

-r <runfilepath> - specifies the file path to the current run file for which you want to compute the evaluation metrics;

-rgt <rGT directory path> - specifies the path to the relevance ground truth (denoted by rGT) for the current data set;

-dgt <dGT directory path> - specifies the path to the diversity ground truth (denoted by dGT) for the current data set;

-t <topic file path> - specifies the file path to the topic xml file for the current data set;

-o <output file directory> - specifies the path for storing the evaluation results. Evaluation results are saved as .csv files (comma separated values);

-f <output file name> - is optional and specifies the output file name. By default, the output file will be named according to the run file name + "\_metrics.csv".

Run example:

java -jar div\_eval.jar -r c:\divtask\RUNd2.txt -rgt c:\divtask\rGT -dgt c:\divtask\dGT -t c:\divtask\devset\_topics.xml -o c:\divtask\results –f my\_first\_results

Output file example:

--------------------

"Run name","RUNd2.txt"

--------------------

"Average P@20 = ",.784

"Average CR@20 = ",.4278

"Average F1@20 = ",.5432

--------------------

"Query Id ","Location name",P@5,P@10,P@20,P@30,P@40,P@50,CR@5,CR@10,CR@20,CR@30,CR@40,CR@50,F1@5,F1@10,F1@20,F1@30,F1@40,F1@50

1,"Aachen Cathedral",.8,.9,.95,.9667,.95,.94,.1333,.4,.5333,.7333,.8667,.9333,.2286,.5538,.6831,.834,.9064,.9367

2,"Angel of the North",1.0,.9,.95,.9333,.925,.94,.2667,.5333,.8,.8667,.8667,.9333,.4211,.6698,.8686,.8988,.8949,.9367

...

24,"Acropolis of Athens",.6,.8,.85,.8667,.875,.88,.25,.5,.6667,.6667,.8333,.8333,.3529,.6154,.7473,.7536,.8537,.856

25,"Ernest Hemingway House",.8,.7,.5,.5667,.55,.6,.2353,.4118,.5294,.6471,.7647,.8824,.3636,.5185,.5143,.6042,.6398,.7143

--------------------

"--","Avg.",P@5,P@10,P@20,P@30,P@40,P@50,CR@5,CR@10,CR@20,CR@30,CR@40,CR@50,F1@5,F1@10,F1@20,F1@30,F1@40,F1@50

,,.76,.784,.792,.784,.789,.7944,.2577,.4278,.6343,.7443,.8504,.8919,.376,.5432,.696,.757,.813,.834

----------

\*\*References:

[v1] Weijer, Van de, Schmid, C., Verbeek, J., Larlus, D. Learning color names for real-world applications. IEEE Trans. on Image Processing, 18(7), pp. 1512-1523, 2009.

[v2] Ludwig, O., Delgado, D., Goncalves, V., Nunes, U. Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection. Conference On Intelligent Transportation Systems, 2009.

[v3] Stricker, M., Orengo, M. Similarity of color images. SPIE Conference on Storage and Retrieval for Image and Video Databases III, vol. 2420, 1995, 381 ­- 392.

[v4] Ojala, T., Pietikäinen, M., Harwood, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. IAPR International Conference on Pattern Recognition, vol. 1, 1994, 582 - 585.

[v5] Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., Yamada, A. Color and texture descriptors. IEEE Trans. on Circuits and Systems for Video Technology, vol. 11(6), 2001, 703 - 715.

[v6] Tang, X. Texture Information in Run-Length Matrices. IEEE Trans. on Image Processing, vol.7(11), 1998.[t1] J.M. Ponte, W.B. Croft, „A Language Modeling Approach to Information Retrieval”, Research and Development in Information Retrieval. pp. 275–281, 1998.

[t1] Wu, H.C., Luk, R.W.P., Wong, K.F., Kwok, K.L. Interpreting TF–IDF Term Weights As Making Relevance Decisions. ACM Transactions on Information Systems, Vol 26 (3), 2008, 1 - 37.

[c1] Overfeat home page: http://cilvr.nyu.edu/doku.php?id=code:start.

[c2] OpenCV face detector: http://docs.opencv.org/trunk/doc/py\_tutorials/py\_objdetect/py\_face\_detection/py\_face\_detection.html.

[c3] Popescu, A. CEA LIST’s Participation at MediaEval 2013 Placing Task, Working Notes of MediaEval 2013, CEUR-WS, Vol. 1043, ISSN 1613-0073, Barcelona, Spain.