The Wroclaw University of Technology Participation at ImageCLEF 2010 Photo Annotation Track⋆

Michal Stanek, Oskar Maier, and Halina Kwasnicka

Wrocław University of Technology, Institute of Informatics
michal.stanek@pwr.wroc.pl, oskar.maier@student.pwr.wroc.pl,
halina.kwasnicka@pwr.wroc.pl

Abstract. In this paper we present three methods for image auto-annotation used by the Wroclaw University of Technology group at ImageCLEF 2010 Photo Annotation track. All of our experiments focus on robustness of the global color and texture image features in connection with different similarity measures. To annotate training set we use two version of PATSI algorithm which searches for the most similar images and transferring annotations from them to the target image by applying transfer function. We use both the simple version of the algorithm working only on single similarity matrix, as well as multi-PATSI which uses many similarity measures in order to obtain the final annotations. As third approach to image auto-annotation we use Penalized Discriminant Analysis to train multi class classifier in One-vs-All manner. During training and optimization process of all annotators we use F-measure as evaluation measure trying to achieve its highest value on a training set. Obtained results indicate that our approach achieved a high quality measure only for a small group of terms and it is necessary to take into account also local image characteristics.

1   Introduction

Recently, Makadia et. al. [1] proposed a family of image annotation baseline methods that are build on the hypothesis that visually similar images are likely to share the same annotations. They treat image annotation as a process of transferring labels from nearest neighbours. Makadia's method does not solve the fundamental problem of determining the number of annotations that should be assigned to the target image. Thus they assume a constant number of annotations per image. The transfer is performed in two steps: all annotations from the most similar image are rewritten and the most frequent words are chosen from the whole neighbourhood until a given annotation length has been achieved.

We extend Makadia's approach by constructing PATSI *(Photo Annotation through Similar Images)* annotator which introduce transfer function [2] as well

as optimization algorithm which can be used to find optimal number of neighbours and the best transfer threshold according to the specified quality measure [3]. During our experiments with different similarity metrics we extend this algorithm to multi-PATSI which perform annotation transfer process based onto many similarity matrices calculated using different feature sets and similarity measures and combine results into final annotation based on the quality of each annotator for specific words.

At ImageCLEF 2010 photo annotation track [4] we evaluate PATSI and multi-PATSI approach with global image features. During experiments we use grid segmentation and statistical color informations as well as features extracted using LIRE package [5]. As third type of automatic image annotator we train PDA [6, 7] classifier onto CEDD [8] and Jpeg Coefficient Histogram [9] in One-vs-All manner.

This paper is organized as follows. In the next section we describe used automatic image annotation methods with explanation of used features, distance measures and details of annotation algorithm. The third section describes the experiments and achieved results. The paper is finished with conclusions and remarks on possible further improvements of the method.

2 Annotation process

In this section we describe automatic image annotation methods used during ImageCLEF 2010 Photo Annotation track [4] by our team. First we focus on types of visual features extracted from images and similarity measures used to build similarity matrices then we describe the annotation transfer process.

2.1 Visual Features

The image $\mathcal{I}$ in a training dataset D is represented by a n-dimensional vector of visual features $v^{\mathcal{I}} = (v_1^{\mathcal{I}}, \cdots, v_n^{\mathcal{I}})$. All visual features are a m-dimensional vector of low level attributes $v_i^{\mathcal{I}} = (x_1^{i,\mathcal{I}}, \cdots, x_m^{i,\mathcal{I}})$. The visual features must be extracted from the image and can represent information about color and texture for the entire image, or only selected area of the image $\mathcal{I}$.

For all images in both training and tasting dataset we performed visual feature extraction using self made feature extractor and the image descriptors contained in the LIRE package [5]. We focused mainly on global image characteristics, but we use also more local information obtained after splitting image by rectangular 5-by-5 and 20-by-20 grid. The list of extracted features include:

1. From **MPEG-7** standard [9] we use following image descriptors calculated for the whole image:
   - Fuzzy Color Histogram – 125 dimensions
   - JPEG Coefficient Histogram – 192 dimensions
   - General Color Layout – 18 561 dimensions
   - Color and Edge Directivity Descriptor (CEDD)[8] – 120 dimensions

- Fuzzy color and texture histogram (FCTH)[10] – 192 dimensions
2. **Tamura features** first three from six texture features corresponding to human visual perception [11]:
  - coarseness – size of the texture elements,
  - contrast – contrast stands for picture quality,
  - directionality – texture orientation.

  Tamura features vector has 16 dimensions.
3. **Auto Color Correlogram** features defined in [12, 13] – 256 dimensions
4. **Gabor** texture features [14] – 60 dimensions
5. Statistical color and edges information of image regions (5-by-5 and 20-by-20 grid) in two color spaces RGB and HSV:
  - x and y coordinates of the segment center – 2 dimensions,
  - the mean value of color in each channel of the color space – 3 dimensions,
  - standard deviations of color changes in each channel for a given color space – 3 dimensions,
  - mean eigenvalues of color Hessian in each channel for a given color space – 3 dimensions.
6. **CoOccurance Matrix** [15] calculated for each segment of 5-by-5 and 20-by-20 segmentation – 21 dimensions

2.2 Distance Metrics

To obtain the similarity or rather dissimilarity between two images, we measure the distance between vectors in metric space and divergence between distributions build onto visual vectors. In our experiments we use distance measures described below.

Minkowski distance The Minkowski distance is widely used for measuring similarity between objects (e.g., images). The Minkowski metric between image $\mathcal{A}$ and $\mathcal{B}$ is defined as:

$$d_{\mathrm{MK}}(\mathcal{A}, \mathcal{B}) = \left(\sum_{i=1}^{n} \left| v_i^A - v_i^B \right|^p \right)^{1/p} \quad (1)$$

where p is the Minkowski factor for the norm. Particularly, when p is equal one and two, it is the well known L1 and Euclidean distance respectively.

Cosine distance is a measure of similarity between two vectors of n dimensions by finding the cosine of the angle between them, often used to compare documents in text mining

$$d_{\mathrm{Cos}}(\mathcal{A}, \mathcal{B}) = \frac{1 - (v^{\mathcal{A}})(v^{\mathcal{B}})^T}{\|v^{\mathcal{A}}\|_2 \|v^{\mathcal{B}}\|_2}. \quad (2)$$

Manhattan distance also called cityblock distance or the taxicab metric is the metric of the Euclidean plane defined by:

$$d_{\mathrm{Manh}}(\mathcal{A}, \mathcal{B}) = \sum_i (v_i^{\mathcal{A}} - v_i^{\mathcal{B}}) \tag{3}$$

Correlation distance measures the similarity in shape between vectors defined by

$$d_{\mathrm{Corr}}(u, v) = \frac{1 - (v^{\mathcal{A}} - \bar{v}^{\mathcal{A}})(v^{\mathcal{B}} - \bar{v}^{\mathcal{B}})^T}{\|(v^{\mathcal{A}} - \bar{v}^{\mathcal{A}})\|_2 \|(v^{\mathcal{B}} - \bar{v}^{\mathcal{B}})\|_2^T}, \tag{4}$$

where $\|(u - \bar{u})\|_2$ is L2 distance between vector u and mean vector $\bar{u}$.

Jensen–Shannon Divergence Based on the visual feature vectors $v^{\mathcal{I}}$ one can build a model $M^{\mathcal{I}}$ for the image $\mathcal{I}$. We can assume that $M^{\mathcal{I}}$ is a multi-dimensional random variable described by multi-variate normal distribution and all vectors $v_i^{\mathcal{I}}$ are realizations of this model. The probability density function (*PDF*) for the model $M^{\mathcal{I}}$ is defined as:

$$M^{\mathcal{I}}(x, \mu, \Sigma) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right) \tag{5}$$

where x is the observation vector, μ the mean vector, and Σ the covariance matrix. Both μ and Σ are parameters of the model calculated using Expectation-Maximization -algorithm [16] on all visual features $[v_1^{\mathcal{I}}, \cdots, v_n^{\mathcal{I}}]$ of the image $\mathcal{I}$. In order to avoid problems of inverting covariance matrix (avoid matrix singularity) one may perform regularization of the covariance matrix. Models of images are build for all images in the training set, as well as for the query image.

Distance between the models can be computed as Jensen–Shannon divergence, which is a symmetrized version of Kullback–Leibler divergence:

$$d_{\mathrm{JS}}(\mathcal{A}, \mathcal{B}) = \frac{1}{2}D_{\mathrm{KL}}(M^{\mathcal{A}}\|M^{\mathcal{B}}) + \frac{1}{2}D_{\mathrm{KL}}(M^{\mathcal{B}}\|M^{\mathcal{A}}), \tag{6}$$

where $M^{\mathcal{A}}$, $M^{\mathcal{B}}$ are models (PDF) for the images A and B, and D_{KL} is the Kullback-Leibler distance which for multivariate-normal distribution takes the form:

$$D_{\mathrm{KL}}(\mathcal{M}^A\|\mathcal{M}^B) = \frac{1}{2}\log_e\left(\frac{\det \Sigma_B}{\det \Sigma_A}\right) + \frac{1}{2}\mathrm{tr}\left(\Sigma_B^{-1}\Sigma_A\right)$$
$$+ \frac{1}{2}\left(\mu_B - \mu_A\right)^\top \Sigma_B^{-1}(\mu_B - \mu_A) - \frac{N}{2}, \tag{7}$$

where Σ_A, Σ_B and μ_A, μ_B are covariance matrices and mean vectors from the respective image models A and B.

2.3 Automatic Image Annotation Methods

We use three methods of automatic image annotation, such as PATSI (Photo Annotation through Similar Images) annotator, multi-PATSI annotator and multi-class PDA classificator. Details of all of those methods are described below.

PATSI Annotator In the PATSI *(Photo Annotation through Finding Similar Images)* approach, for a query image $\mathcal{Q}$, a vector of the k most similar images from the training dataset D needs to be found based on the similarity distance measure d. Let $[r_1, \cdots, r_k]$ be the ranking of k the most similar images ordered decreasingly by similarity. Based on the hypothesis that images similar in appearance are likely to share the same annotation, keywords from the nearest neighbours are transferred to the query image. All labels for the image on position r in the ranking are transferred with a value designated by the *transfer function* $\varphi(r_i)$.

To assure that labels from more similar images have a larger impact on resulting annotation we define φ as

$$\varphi(r_i) = \frac{1}{i}, \tag{8}$$

where r_i is an image on position i in the ranking. All words associated with image r_i are then transferred to the resulted annotation with the associated transfer value $1/i$. If the words has been transferred before the transfer values are summed.

The resulting query image annotation consists of all the words whose transfer values were greater than a specified threshold t. The threshold value t has an impact on the resulting annotation length and its optimal value as well as the optimal number of neighbours k which should be taken into account during the annotation process must both be found using an optimization process. The outline of the PATSI annotation method is presented in the figure 1 and is summarized in the Algorithm 1.
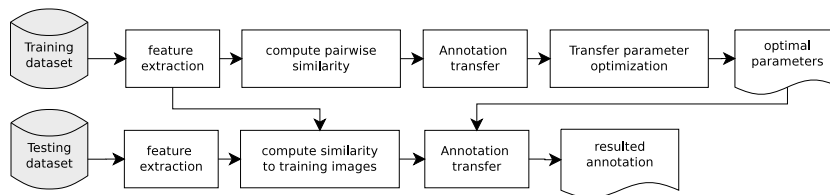

Fig. 1. Schematic diagram of the PATSI algorithm

The optimal parameters k^* and t^* differ greatly not only for different databases, but also between feature sets, methods of distance measure and transfer functions. There exists no optimal choice of them that would be suitable in all cases. We need to adjusting them in each explicit case.

Algorithm 1 PATSI image annotation algorithm

Require: D – training dataset

   d – distance function

   Q – annotation quality function

   φ - transfer function

 1: {Preparation Phase}

   calculate and store visual features all images in training dataset D

 2: calculate similarity matrix using distance function d between all images in training dataset D

 3: {Optimization Phase}

   choose values for k and t maximizing quality function Q on a training dataset

 4: {Query Phase}

   calculate the visual features for query image $\mathcal{Q}$

 5: calculate the distance from query image to all other images in training database D.

 6: take k images with the smallest distances between the models and create a ranking of those images.

 7: transfer all words from the images in the ranking with the value $\varphi(r)$, where r is the position of the image in the ranking.

 8: as a final annotation take the words which transfer values sum is greater or equal to the provided threshold t value.

Finding t^* and k^* proves to be a non-trivial task. The commonly used optimization solvers are inapplicable due to the non-linear character of quality function Q (discrete domain on k and continues on t). To efficiently find t^* and k^* we propose and use the *iterative refinement* algorithm which is described in [3].

Multi-PATSI Annotator During experiments we spot that some of the features as well as distance metrics are more suitable to detect some groups of words, while showing a weak performs for others. By combining them together we can increase overall annotation performance. We propose the multi-PATSI method that take advantage of this observation by joining together the strengths of a number of annotation techniques.
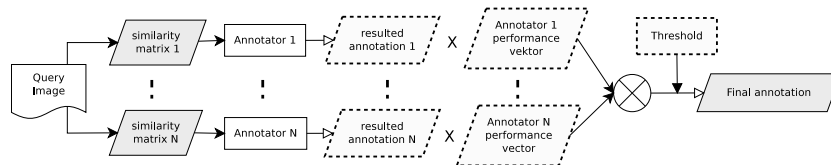
Fig. 2. Schematic diagram of the multi-PATSI algorithm.

The overall schema of multi-PATSI approach is presented in figure 2. In the first step we run PATSI algorithm separately for each features sets and distance functions to obtain annotation vectors. Each element of those vectors represent whether word should be assigned or not to the query image $\mathcal{Q}$ (class $\{-1, 1\}$).

For each of the PATSI annotators at learning stage the performance vector is calculated. The performance vector corresponds to the efficiency of the PATSI-annotator for each of the annotated words on the testing set.

For each PATSI-annotator the resulted annotation vector is multiplied by a performance vector to obtain weighted annotator response. All weighted responses are then summed together creating final annotation. All concepts which obtain value greater than a threshold t_{multi} are treated as a final annotation for a query image $\mathcal{Q}$. Optimal threshold value t_{multi}^* can be calculated using cross-validation method and optimization technique such as iterative refinement [3].

Multi-Class Classification As third annotation method we use Penalized Discriminant Analysis classifier[6, 7] from Python Machine Learning Module – MLPY [17] in One vs. All scenario.

In this approach for each concept we train a separate PDA classifier using the extracted image features. We use all features from images annotated by a specific concept as positive examples and others as negative. In training we use four fold cross-validation.

3 Experimental Results

We submitted five runs for the annotators and features sets described in previous section:

1. **PATSI** with Kullback Leibler divergence - hsv color space and grid 20-by-20,
2. **PATSI** with Kullback Leibler divergence - rgb color space and grid 20-by-20,
3. **Multi-PATSI** with features presented in Table 2,
4. **PDA** classifier with CEDD features,
5. **PDA** classifier with Jpeg Coefficient Histogram features.

The official results of the five runs in terms of Average Precision (AP), Average Equal Error Rate (Avg. EER), Average Area Under Curve (Avg. AUC) are reported in Table 1. A detailed overview of the quality of annotations for each of the submitted methods for the 30 best-annotated words is presented in Table 3.

4 Conclusion

During the training and optimization process the parameters of the classifiers was tuned using the F-measure (harmonic mean of precision and recall) instead of the Average Precision. F-measure resulted that in all submitted annotations

Table 1. Official results of submitted runs on testing dataset

Submited run	AP	EER	AUC
PDA + CEDD	0.188821	0.361605	0.419472
PDA + JpegCoefficientHistogram	0.186649	0.375593	0.398008
PATSI + RGB	0.183472	0.464731	0.125210
PATSI + HSV	0.180601	0.461858	0.128203
Multi-PATSI	0.174149	0.427712	0.240389

Table 2. Feature sets and distance metrics with its optimal transfer parameters used in multi-PATSI annotator

Feature set	L1			L2			Cosine			Manhattan			correlation		
	F	t^*	n^*	F	t^*	n^*	F	t^*	n^*	F	t^*	n^*	F	t^*	n^*
Auto Color Correlogram	0.211	0.1	30		0.1	29		0.1	24		0.10	30		0.10	29
CEDD	0.225	0.10	19	0.240	0.26	24	0.238	0.28	22		0.10	19		0.1	25
FCTH	0.221	0.10	20		0.10	24		0.10	23		0.10	20	0.219	0.1	26
Fuzzy Color Histogram	0.208	0.10	30		0.10	28		0.1	29		0.10	30		0.10	27
General Color Layout		0.02	32	0.199	0.10	31		0.02	35		0.02	32	−	−	−
Jpeg Coefficient Histogram	0.229	0.10	9	0,243	0.34	19	0.236	0.18	13		0.10	9	−	−	−
Gabor		0.02	35		0.02	35	0.196	0.015	36		0.02	35		0.015	36
Tamura		0.06	29		0.05	33		0.10	27		0.06	29	−	−	
Grid 20x20 – RGB		0.1	30		0.1	30		0.10	33	0.213	0.1	27	−	−	−
Grid 20x20 – RGB + dev.		0.1	27	−	0.1	29		0.1	30	−	−	−	−	−	−
Grid 20x20 – RGB + dev. + hes		0.12	29		0.1	30	0.219	0.14	30		0.1	32	−	−	−
Grid 20x20 – HSV		0.1	30	0.213	0.1	31		0.01	34	0.215	0.1	30	−	−	−
Grid 20x20 – HSV + dev.		0.1	27		0.1	30	0.216	0.14	36		0.1	27	−	−	
Grid 20x20 – HSV + dev. + hes		0.1	30		0.1	29	0.219	0.14	36		0.1	30	−	−	
Grid 20x20 – CoOccuranceMatrix	−	−	−		0.06	33		0.06	33		0.02	32	−	−	−
Grid 5x5 – RGB		0.10	30		0.1	30		0.1	30		0.10	30	−	−	−
Grid 5x5 – RGB + dev.		0.10	30		0.1	30		0.1	30	0.10		30	−	−	
Grid 5x5 – RGB + dev. + hes		0.10	30		0.1	30	0.217	0.1	30	0.10		30	−	−	
Grid 5x5 – HSV		0.10	30		0.1	29		0.1	33		0.1	29	−	−	−
Grid 5x5 – HSV + dev.		0.10	29	0.219	0.1	25		0.10	24	0.221	0.1	29	−	−	−
Grid 5x5 – HSV + dev. + hes		0.10	27		0.1	25	0.225	0.16	27		0.1	27	−	−	−
Grid 5x5 – CoOccuranceMatrix	0.18		32	0.1		32	0.10	28		0.219	0.18	32	−	−	−

results we optimize annotation length by providing annotation vectors contained only $\{-1, 1\}$ values. Using vectors prepared in such a way results in low Average Precision quality.

The published results show that the highest measure of quality according to AP measure, reached the multi-class PDA classifier with CEDD features. On the other hand the worst in comparison was multi-PATSI annotator.

Table 3. Average Precision for the 30 best annotated concepts in all submitted results

L.p.	PATSI (RGB) Concept	AP	PATSI (HSV) Concept	AP	multi-PATSI Concept	AP	PDA (CEDD) Concept	A P	PDA (JpegCoefficient) Concept	AP
1	Neutral Illumination	0,947	Neutral Illumination	0,947	Neutral Illumination	0,947	Neutral Illumination	0,973	Neutral Illumination	0,965
2	No Visual Season	0,883	No Visual Season	0,883	No Visual Season	0,883	No Visual Season	0,921	No Visual Season	0,896
3	No Persons	0,726	No Blur	0,719	No Persons	0,717	No Persons	0,776	No Blur	0,833
4	No Blur	0,723	No Persons	0,718	No Blur	0,716	No Blur	0,760	No Persons	0,767
5	natural	0,635	natural	0,635	natural	0,635	Outdoor	0,672	natural	0,654
6	Outdoor	0,608	Outdoor	0,586	Outdoor	0,555	Day	0,669	Outdoor	0,644
7	Day	0,598	Day	0,578	Day	0,535	natural	0,663	Day	0,629
8	Sky	0,542	Sky	0,514	cute	0,511	cute	0,571	cute	0,565
9	cute	0,511	cute	0,511	No Visual Time	0,443	No Visual Time	0,554	No Visual Time	0,563
10	Plants	0,480	No Visual Time	0,431	Sky	0,369	Sky	0,504	Partly Blurred	0,530
11	Landscape Nature	0,438	Landscape Nature	0,409	Visual Arts	0,325	Plants	0,446	Sky	0,462
12	No Visual Time	0,431	Plants	0,389	male	0,307	male	0,387	male	0,432
13	Clouds	0,408	Clouds	0,369	Clouds	0,294	No Visual Place	0,365	Plants	0,375
14	Visual Arts	0,325	Visual Arts	0,325	No Visual Place	0,291	Partly Blurred	0,363	No Visual Place	0,351
15	Sunset Sunrise	0,315	Sunset Sunrise	0,312	Partly Blurred	0,286	Clouds	0,354	Clouds	0,346
16	Indoor	0,315	male	0,302	Plants	0,279	Visual Arts	0,330	Indoor	0,338
17	male	0,304	Indoor	0,299	Night	0,274	Indoor	0,316	Citylife	0,325
18	Citylife	0,296	Citylife	0,297	Sunset Sunrise	0,269	Landscape Nature	0,302	Visual Arts	0,320
19	Partly Blurred	0,284	Partly Blurred	0,283	Citylife	0,266	Adult	0,295	female	0,294
20	Night	0,281	No Visual Place	0,275	Indoor	0,255	Citylife	0,294	Landscape Nature	0,289
21	No Visual Place	0,275	Night	0,260	Adult	0,228	Night	0,272	Single Person	0,287
22	Sunny	0,271	Trees	0,252	Macro	0,222	female	0,261	Adult	0,284
23	Trees	0,268	Sunny	0,250	Building Sights	0,219	Single Person	0,249	Family Friends	0,276
24	female	0,255	Macro	0,235	Portrait	0,215	Sunny	0,244	Building Sights	0,270
25	Water	0,247	Aesthetic Impression	0,234	Water	0,211	Building Sights	0,243	Water	0,258
26	Park Garden	0,238	Park Garden	0,234	female	0,211	Aesthetic Impression	0,239	Portrait	0,241
27	Family Friends	0,235	Water	0,229	Single Person	0,210	Portrait	0,236	Macro	0,233
28	Aesthetic Impression	0,229	Vehicle	0,221	Family Friends	0,203	Family Friends	0,235	Aesthetic Impression	0,230
29	Macro	0,219	Adult	0,215	Landscape Nature	0,201	Park Garden	0,233	Vehicle	0,205
30	Adult	0,216	Single Person	0,197	Aesthetic Impression	0,199	Water	0,224	Sunny	0,196

The results show that the method of transferring annotations seems to be very interesting concept. However, it will be necessary to use outside the global characteristics of the image also the local features as well as adaptive metric functions.

References

[1] Makadia, A., Pavlovic, V., Kumar, S.: A new baseline for image annotation. In: ECCV '08, Berlin, Heidelberg, Springer-Verlag (2008) 316–329

[2] Stanek, M., Broda, B., Kwasnicka, H.: Patsi — photo annotation through finding similar images with multivariate gaussian models. Lecture Notes in Computer Science, International Conference on Computer Vision and Graphics (2010)

[3] Stanek, M., Maier, O., Kwasnicka, H.: PATSI - photo annotation through similar images with annotation length optimization. In: Intelligent information systems. Publishing House of University of Podlasie (2010) 219–232

[4] Nowak, S., Huiskes, M.: New strategies for image annotation: Overview of the photo annotation task at imageclef 2010. In: In the Working Notes of CLEF 2010. (2010)

[5] Lux, M., Chatzichristofis, S.A.: Lire: lucene image retrieval: an extensible java cbir library. In: MM '08: Proceeding of the 16th ACM international conference on Multimedia, New York, NY, USA, ACM (2008) 1085–1088

[6] Ghosh, D.: Penalized discriminant methods for the classification of tumors from gene expression data. Biometrics **59**(4) (2003) 992–1000

[7] Hastie, T., Buja, A., Tibshirani, R.: Penalized discriminant analysis. Annals of Statistics **23** (1995) 73–102

[8] Chatzichristofis, S., Boutalis, Y.: Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. Computer Vision Systems (2008) 312–322

[9] Chang, S.F., Sikora, T., Puri, A.: Overview of the MPEG-7 Standard. IEEE Trans. Circuits and Systems for Video Technology **11**(6) (2001) 688–695

[10] Chatzichristofis, S.A., Boutalis, Y.S.: Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval. Image Analysis for Multimedia Interactive Services, International Workshop on **0** (2008) 191–196

[11] Tamura, H., Mori, S., Yamawaki, T.: Texture features corresponding to visual perception. IEEE Transactions on System, Man and Cybernatic **6** (1978)

[12] Goodrum, A.: Image information retrieval: An overview of current research. Informing Science **3** (2000) 2000

[13] Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), Washington, DC, USA, IEEE Computer Society (1997) 762

[14] Zhang, D., Wong, A., Indrawan, M., Lu, G.: Content-based image retrieval using gabor texture features. In: IEEE Transactions PAMI. (2000) 13–15

[15] Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics **3**(6) (November 1973) 610–621

[16] McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions (Wiley Series in Probability and Statistics). 2 edn. Wiley-Interscience (March 2008)

[17] Albanese, D., Merler, S., Jurman, G., Visintainer, R., Furlanello, C.: Mlpy machine learning py (2010) http://mloss.org/software/view/66/.