IBM Research at Image CLEF 2015: Medical Clustering Task

Suman Sedai, Xi Liang, Mani Abedini, Qiang Chen, Rajib Chakravorty, Rahil Garnavi

IBM Research Australia
Level 5, 204 Lygon Street, Carlton
Victoria 3053, Australia
{ssedai,xiliang,mabedini,qiangchen,rachakra,rahilgar}@au1.ibm.com
http://www.research.ibm.com/labs/australia

Abstract. In this paper, we present the learning strategies and feature extraction techniques that were applied by the IBM Research Australia team to the Medical Clustering challenge of ImageCLEF 2015. The challenge is to automatically annotate and categorize X-ray images into head-neck, body, upper-limb, lower-limb and foreign object categories. Our proposed methodology and details of experiments for each submitted run has been discussed in this paper, followed by final results provided by the competition organizers. The key components used in our submissions are based on sparse coding of SIFT, local binary patterns and multi-scale local binary patterns with spatial pyramid, advanced fisher vector, various SVM kernels, and an effective fusion methodology, to ensure high classification accuracy. Comprehensive experiments demonstrate the effectiveness of the proposed system. Six out of the ten submissions of IBM Research were among the top 10 best results, where two of our submissions outperformed all other submissions, therefore the team has achieved the first place in the competition.

Keywords: Medical image classification, Local binary pattern, Sparse coding, Fisher vector, Fisher encoding, Spatial pyramid

1 Introduction

ImageCLEF medical clustering task [0] is a new category in ImageCLEF 2015 [0]. The objective of this task is to categorize digital X-ray images into four clusters: head-neck, upper-limb, body, and lower-limb [0]. X-ray is the most common medical image modality as it accounts for one third of the the radiographs taken in a typical radiology department [0]. Automatic categorization of medical images has a number of applications including efficient retrieval, archiving, and patient similarity matching. For example, for search and retrieval task, the image needs to be pre-classified. However, X-ray image classification is a challenging task due to variation in the patients location, exposure, subject motion and the presence of artifacts and foreign objects. In this work, we present a X-ray annotation and categorization system which accurately performs in presence of such artifacts.

Existing methods on X-ray image classification are based on local patch features such as local binary pattern (LBP) histogram, edge histogram and SIFT [0]. Recently, the choice of the local feature has gone beyond the traditional local patch descriptor, and higher dimensional representation such as pooled coding vectors and multi-resolution feature modeling have shown to improve the performance. In this paper, we investigated several feature extraction techniques based on higher level feature coding of local feature and multi-resolution analysis for X-ray image clustering challenge in ImageClef 2015. The rest of the paper is organized as follows: Section describes the methodologies applied for the medical clustering task. Section discusses the experimental setup, which has been applied for training and our internal evaluation, and the comparison of our methods before submission. Section reviews the submission runs and presents the results. Finally, Section concludes the paper.

2 Feature Extraction and Learning Methodologies

2.1 Multi-scale LBP Histogram with Spatial Pyramid

LBP describes gray-scale local texture of the image by detecting local patterns between adjacent pixels. For example, original LBP operator labels the pixels of an image by thresholding the 3×3- neighborhood of each pixel with the center value and considering the result as binary string resulting in 256 different patterns. In multi-scale LBP (MSLBP) [0], comparison operator between single pixels in LBP is simply replaced with comparison between average gray-values of sub-regions where each sub-region is a square block containing neighboring pixels, and the size of the square block is governed by the scale of LBP. Once the MSLBP values are computed for each pixel, a feature vector for a given image region can be computed as 256 dimensional histogram of the LBP values inside the region. However, such global histogram does not encode spatial information that may be crucial for image recognition task.

In this paper, we compute the MSLBP histogram at multiple scales of spatial resolution by partitioning the image into increasingly smaller overlapping subregions and computing the MSLBP histogram inside each region. The resulting spatial pyramid has shown improvements in the performance of image classification tasks. We computed LBP histogram at two levels of spatial pyramid. Let w and h denote the width and the height of the image. In the first level, the MSLBP histogram is computed in a block where the block covers the entire image. In the second level, MSLBP histogram is computed across the 9 overlapping blocks with size of $w/2 \times h/2$ which are obtained by moving a block along x axis with increment of $w/4$, and along y-axis with increment of $h/4$. Therefore, the total number of blocks is 10. The MSLBP feature computed in all the spatial pyramid blocks are then concatenated to form a single feature vector which we name as MSLBP-SP.

2.2 Sparse Coding with Max-pooling and Spatial Pyramid

Sparse coding is a popular approach for adaptively learning feature representations. Given a set of input signals $\{\mathbf{x}_i\}_{i=1}^{N}$, where $\mathbf{x} \in \mathbb{R}^m$, the goal is to find the sparse approximation over a dictionary D in $\mathbb{R}^{m \times k}$, with k columns referred to as basis vectors, so that a linear combination of basis vectors from $\mathbf{D}$ reconstructs the signal $\mathbf{x}$. Rather than using pre-defined dictionaries, sparse coding algorithms aim to learn a dictionary of basis functions. The objective function of sparse coding is stated as:

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \parallel \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i \parallel_2^2 + \lambda \parallel \boldsymbol{\alpha}_i \parallel_1 \tag{1}$$

where λ is a regularization parameter and the l_1 penalty ensures sparse solution. A general approach to minimize the objective function is to alternate between the two variables, i.e., minimizing over one while keeping the other fixed. In this paper, we use on-line algorithm based on stochastic approximation which minimizes the sequential quadratic approximation of the expected cost [0]. Once the dictionary $\mathbf{D}$ is trained, the sparse representation $\boldsymbol{\alpha}$ of a feature vector $\mathbf{x}$ can be computed by minimizing the following objective function:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \parallel \mathbf{x} - D\boldsymbol{\alpha} \parallel_2^2 + \lambda \parallel \boldsymbol{\alpha} \parallel_1 \tag{2}$$

For any image represented by a set of M features, we can compute a single feature vector using a pooling function. For example, the pooling function defined as *average* function results in a histogram feature. In this paper, we define the pooling function as the *max-pooling* function over the absolute sparse codes:

$$z_j = \max\{\mid \alpha_{1,j} \mid, \cdots, \mid \alpha_{M,j} \mid\} \tag{3}$$

where z_j is the j^{th}element of the final pooled vector $\mathbf{z}$ which is compact representation of the given image region. The *max-pooling* process is well established by biophysical evidence is visual cortex [0] and is empirically justified by many image recognition algorithms.

Given an image, we first divide the image into 10 spatial pyramid blocks in a similar manner described in Section and compute the local features in each block. The sparse representation of the local features is then computed in each block using the method described above. In this paper, we investigate two types of local feature for sparse coding: (a) *dense SIFT*(b) *dense MSLBP. Dense SIFT* is a faster version of SIFT where the SIFT descriptors with fixed scale and orientation are computed in densely sampled locations, inside a given image block. In our implementation SIFT descriptor is computed on a 100×100 patches densely sampled in a given block on a grid with step size of 30×30. Similarly, we compute the MSLBP described in Section in 100×100 patch densely sampled in a given block on a grid with step size of 30×30.

The sparse coded features computed in spatial pyramid taking dense SIFT as local features is named SC-DenseSIFT-SP and the sparse coded features

computed in spatial pyramid by taking dense MSLBP features is named SC-DenseMSLBP-SP.

2.3 Fisher Kernel Feature Coding

Fisher Kernel [0,0] feature encoding is one of bag-of-word model [0,0] and recent evaluation [0] shows this encoding method achieved best results in many cases. Fisher Kernel encodes the distribution information of the feature points which can separate the image specific information from the noisy local features. Fisher Kernel encoded features can be represented using a linear model which is computationally efficient.

Let $X = \{x_1, \cdots, x_N\}$ be the set of N local features extracted from an image I and $u_\lambda(x)$ is a probability density function which models a generative process in the feature space. The image I can be described by the gradient vector of log likelihood with respect to the model parameters λ:

$$G_\lambda^X = \frac{1}{N} \nabla_\lambda \log u_\lambda(X). \tag{4}$$

Let F_λ is the Fisher information matrix of u_λ, a natural kernel on these gradients is

$$K(X, Y) = G_\lambda^{X'} F_\lambda^{-1} G_\lambda^Y, \tag{5}$$

$$F_\lambda = E_{x \sim u_\lambda}[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)']. \tag{6}$$

As F_λ is symmetric and positive definite, it can be decomposed as,

$$F_\lambda = L'_\lambda L_\lambda, \tag{7}$$

and the kernel $K(X, Y)$ is defined as a dot-product between normalized vectors, called Fisher vectors:

$$\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X. \tag{8}$$

Linear classifiers typically consume less time than non-linear ones in training and testing phases. Learning a kernel classifier using the Fisher kernel is equivalent to learning a linear classier on the Fisher vectors $\mathcal{G}_\lambda^X$.

The probability density function u_λ in Fisher Vecter encoding is presented using a Gaussian Mixture Model (GMM), defined as

$$u_\lambda(x) = \sum_{k=1}^{K} \pi_k u_k(x) \tag{9}$$

The GMM is trained on local features of a large image set using Maximum Likelihood (ML) estimation. The parameters of the trained GMM are denoted as,

$$\lambda = \{\pi_k, \mu_k, \Sigma_k, k = 1, \cdots, K\}, \tag{10}$$

where $\{\pi, \mu, \Sigma\}$ are the prior probability, mean vector and diagonal covariance matrix of the Gaussian mixture respectively. This GMM is used to describing low level features $X = \{x_1, \cdots, x_N\}$ extracted from an image I. The soft assignments of the descriptor x_i to the kth Gaussian component γ_{ik} is defined as

$$\gamma_{ik} = \frac{\pi_k u_k(x_i)}{\sum_{k=1}^{K} \pi_k . u_k(x_i)} \tag{11}$$

Fisher vector (FV) for X is denoted as $\phi(X) = \{\mathcal{G}^X_{\mu_1}, \mathcal{G}^X_{\sigma_1}, \cdots, \mathcal{G}^X_{\mu_K}, \mathcal{G}^X_{\sigma_K}\}$. $\mathcal{G}_{\mu_k}$ and $\mathcal{G}_{\sigma_k}$ is defined as:

$$\mathcal{G}^X_{\mu_k} = \sum_{i=1}^{N} \frac{1}{N\sqrt{\pi_k}} \gamma_{ik} \frac{x_i - \mu_k}{\sigma_k}, \tag{12}$$

$$\mathcal{G}^X_{\sigma_k} = \sum_{i=1}^{N} \frac{1}{N\sqrt{2\pi_k}} \gamma_{ik} [\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1], \tag{13}$$

Where σ_k is the square root of the diagonal values of Σ_k. When increasing the number of Gaussian kernels, the Fisher vectors gets sparser, and the distribution of the features in a given dimension gets closer to zero. We apply a combination of power normalization and L_2 normalization to each Fisher vector descriptor. In z-dimension of the Fisher vector ϕ, the power normalization is defined as,

$$f(z) = sign(z)|z|^{\alpha}, \tag{14}$$

where $0 \leq \alpha \leq 1$ is a parameter of the normalization and we choose $\alpha = 0.5$ in all the experiments. Subsequently, the Fisher vectors are L_2 normalized.

2.4 Visual global descriptors

This section explains four of our submissions which were based on extracting global visual features [0], and using SVM and Random Forest, two of the very common yet effective classification techniques. For visual features we extracted Edge Histogram and Local Binary Patterns (LBP) using pyramid spatial granularity. The spatial pyramid refers to extracting the entire image as first level then in the second level in 2x2 grids followed by (3x3) grids in the third level. All grid blocks are eventually concatenated.

Edge Histogram: We consider 8 edge direction bins and 8 edge magnitude bins, based on a Sobel filter (64-dimensional).

Local Binary Patterns: We also used LBP histograms of 8-bits local binary patterns, each of which is generated by comparing the gray-scale value of a pixel with those of its 8 neighbors in circular order, and setting the corresponding bit to 0 or 1, accordingly. A pattern is called uniform if it contains at most two bitwise transitions from 0 to 1. The final histogram for each region in our granularity contains 59 bins; 58 for uniform patterns and 1 for all the non-uniform patterns.

We investigated various classifiers implemented in Weka [0], including: Decision Tree, Support Vector Machine (RBF Kernel, Poly kernel, Normalized Ploy kernel and Puk kernel), Random Forest, Logistic Model Tree (LMT) and Naive Bayesian. The validation results suggested that SMO (with normalized poly kernel) and Random forest were the best choice with respect to classification performance.

2.5 Fusion of Multiple Methods

In an attempt to build a strong classifier by leveraging various learning methods explained previously, we have applied two fusion methods: *early fusion* and *late fusion* .

Early feature fusion: In early fusion, we concatenated three types of features to form a single feature vector before classification. Specifically, we concatenated MSLBP-SP (described in Section), SC-DenseSIFT-SP (described in Section) and SC-DenseMSLBP-SP (described in Section).

Late fusion: In late fusion, we combine the classification scores of the classification models trained on the feature described in Section - . Let a model k provides a confidence score $s^k_{i,j}$ for each image i and for each class j. We apply optimization to get the final confident score as the weighted sum: $S_{i,j} = \sum_k w^k * s^k_{i,j}$, using 10-fold cross validation. At each fold, we select the model with the highest accuracy and tune the weight to get the best combined accuracy. The final weight parameters have been used to calculate the confidence score on the test set.

3 Experiments

In this section we explain the detail of experiments and our performance evaluation methodology.

Dataset: The training set provided for the medical clustering task contains 500 images where each image belongs to one of five categories: *head-neck*, *upper-limb*, *body*, *lower-limb* and *true negative (foreign objects)*, and each category has 100 images. An independent test set containing 250 images without any class information was also provided.

Model Tuning: In order to tune the classification models and identify the best parameter values, we used 10 fold cross validation. At each fold we train on 90% of the data and evaluate the models on the remaining 10%. This process is repeated 10 times, each time using different train/test partition. We use average $F - Score$ among all the validation runs to select the best classification model for each feature representation.

Testing: The best classification model trained on each feature is used to evaluate on the test set. The list of submitted runs is described in Section and the performance is reported in Table .

Table 1. Results of the runs in the test set. Three metrics (Exact Match, Any Match and Hamming distance) have been used to evaluate the accuracy of submissions. Two of our runs which achieved the highest scores across all submitted runs in the competition, have been highlighted by bold font in the table.

	Exact Match	Any Match	Hamming distance
Run1	**0.752**	**0.864**	0.863
Run2	0.695	0.840	0.889
Run3	0.672	0.812	0.874
Run4	0.599	0.724	0.838
Run5	0.692	0.832	**0.896**
Run6	0.692	0.732	0.755
Run7	0.470	0.568	0.835
Run8	0.603	0.708	0.778
Run9	0.510	0.616	0.849
Run10	0.689	0.820	0.890

4 Submitted Runs and results

Run1: Early fusion of three features: (a) MSLBP-SP (described in Section) (b) SC-DenseSIFT-SP (described in Section) (c) SC-DenseMSLBP-SP (described in Section). SVM classifier with homogenous kernel map and Chi-square kernel is used and multi label classification is employed.

Run2: Same as Run1, except that a single label classification is employed.

Run3: MSLBP-SP feature described in Section , and SVM classifier with homogenous kernel map and Chi-square kernel is used.

Run4: SC-DenseMSLBP-SP described in Section , with SVM classifier with homogenous kernel map and Chi-square kernel is applied.

Run 5: Advanced feature encoding explained in : First, we extracted dense SIFT feature, then applied Fisher Kernel encoding, by learning Mixture of Gaussian (GMMs). As a result, each image has represented by a Fisher Vector. In the next step, we trained linear SVM on the training set and applied it on test set.

Run 6: Edge Histogram for visual feature descriptor and SMO as classification: in this run, we used global edge histogram using spatial pyramid technique and then SMO, normalized poly kernel, which has been explained in section .

Run 7: Edge Histogram for visual feature descriptor and Random Forest as classification: in this run, we used the same feature extracted in run 6, and Random Forest as the classification method.

Run 8: LBP Histogram for visual feature descriptor and SMO as classification: in this run, we extracted LBP histogram using spatial pyramid technique. SMO was used as classification method.

Run 9: LBP Histogram for visual feature descriptor and Random Forest as classification: Similar to run 8, we used LBP global features, followed by training a Random Forest.

Run 10: This run was presented from the late fusion model as explained in section .

5 Conclusion

In this paper, we described feature extraction and learning methodologies and the fusion strategy applied by the IBM Research Australia team to the medical clustering challenge of ImageCLEF 2015. We utilized advanced feature extraction methods to extract local and global features, as well as advanced feature encoding and classification techniques. We also applied early fusion of low-level features, and late fusion of the results of all trained classifier. Overall, six out of the ten submissions of IBM Research team were among the top 10 best results. All runs has been evaluated based on three metrics: *exact match*, *any match* and *hamming distance* metrics. Two of our ten submitted runs demonstrated outstanding results, and outperformed all other submissions across all teams participating in the competition. The early fusion of MSLBP-SP, SC-DenseSIFT-SP and SC-DenseMSLBP-SP with homogeneous kernel map and Chi-Square kernel based SVM classification achieved highest *exact match* and *any match*, whereas Fisher vector resulted in highest *hamming distance*.

References

1. M. Ashraful Amin and Mahmood Kazi Mohammed. Overview of the ImageCLEF 2015 medical clustering task. In *CLEF2015 Working Notes*, CEUR Workshop Proceedings, Toulouse, France, September 8-11 2015. CEUR-WS.org.
2. Mauricio Villegas, Henning Müller, Andrew Gilbert, Luca Piras, Josiah Wang, Krystian Mikolajczyk, Alba García Seco de Herrera, Stefano Bromuri, M. Ashraful Amin, Mahmood Kazi Mohammed, Burak Acar, Suzan Uskudarli, Neda B. Marvasti, José F. Aldana, and María del Mar Roldán García. General Overview of ImageCLEF at the CLEF 2015 Labs. Lecture Notes in Computer Science. Springer International Publishing, 2015.
3. T. M. Lehmann , O. Guumlld , D. Keysers , H. Schubert , M. Kohnen and B. B. Wein Determining the view of chest radiographs J. Dig. Imag., vol. 16, no. 3, pp.280 -291 2003
4. Faruque, M. S. S., Banik, S., Mohammed, M. K., Hasan, M., Amin, M. A. Teaching & Learning System for Diagnostic Imaging; Phase I: X-ray Image Analysis & Retrieval In Proceedings of the 6th International Conference on Computer Supported Education (2015)
5. Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, Sao Deroski Hierarchical annotation of medical images Pattern Recognition, Volume 44, Issues 1011, OctoberNovember 2011
6. Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang, and Stan Z. Li. Learning multi-scale block local binary patterns for face recognition. In Proceedings of the 2007 international conference on Advances in Biometrics
7. Mairal, Julien and Bach, Francis and Ponce, Jean and Sapiro, Guillermo Online Dictionary Learning for Sparse Coding Proceedings of the 26th Annual International Conference on Machine Learning, 2009.
8. Perronnin, F and Dance, C. Fisher Kernels on Visual Vocabularies for Image Categorization. In *Computer Vision and Pattern Recognition*, 2007.
9. Florent Perronnin, Jorge Sanchez, and Thomas Mensink. Improving the Fisher Kernel for Large-Scale Image Classification. In *European Conference on Computer Vision*, 2010.

10. J Sivic and A Zisserman. Video Google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, 2003.
11. L Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition*, 2005.
12. K Chatfield, V Lempitsky, and A Vedaldi. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011.
13. Abedini, Mani and Cao, Liangliang and Codella, Noel and Connell, Jonathan H. and Garnavi, Rahil and Geva, Amir and Merler, Michele and Nguyen, Quoc-Bao and Pankanti, Sharathchandra U. and Smith, John R. and Sun, Xingzhi and Tzadok, Asaf IBM Research at ImageCLEF 2013 Medical Tasks
14. Hall, Mark and Frank, Eibe and Holmes, Geoffrey and Pfahringer, Bernhard and Reutemann, Peter and Witten , Ian H. The WEKA Data Mining Software: An Update SIGKDD Explorations (2009), Volume 11, Issue 1