Virus image classification using different texture descriptors

Loris Nanni,[1] Michelangelo Paci,[2] Sheryl Brahnam,[3] Stefano Ghidoni,[1] and Emanuele Menegatti[1]

[1]DEI, University of Padua, viale Gradenigo 6, Padua, Italy. loris.nanni@unipd.it;
[2]DEI, Dipartimento di Ingegneria dell'Energia Elettrica e dell'InformazioneVia Venezia, 52 47521 - Cesena (FC) - Italy
[3]Computer Information Systems, Missouri State University, 901 S. National, Springfield, MO 65804, USA. sbrahnam@missouristate.edu

Abstract

In this work we proposed an ensemble of texture descriptors for virus image classification. Novel variants of texture descriptors, coupled with support vector machines as the classifier, are proposed. The novel variants of texture descriptors include: 1) a quinary coding of different local binary pattern variants, 2) two new approaches based on quinary coding (a selected multithreshold local quinary pattern and a selected multithreshold local quinary configuration pattern), 3) a new approach based on the co-occurrence matrix, and 4) an ensemble of local phase quantization variants with ternary encoding. Our system is compared with and shown to outperform several state of the art texture descriptors. These results are validated on a dataset of 1500 images with 15 classes. MATLAB code implementing our descriptors is publically available at http://www.dei.unipd.it/wdyn/?IDsezione=3314&IDgruppo_pass=124&preview=

Keywords: protein classification; texture descriptors; primary structure**;** local phase quantization; support vector machines.

1 Introduction

Negative stain transmission electron microscopy (TEM) is a microscopy technique that produces distinctive surface textures. TEM has proven to be very valuable in virus detection, discovery, and taxonomy [1, 2]. Until recently, classification of TEM images was exclusively performed at the microscope by human experts. Since expert inspection is expensive and results depend on the skills and experience of each inspector, automation of TEM image classification has become desirable.

TEM virus images are well suited to machine pattern analysis due to their properties of size, shape, and texture. For instance, virus shapes can range from icosahedral patterns to highly pleomorphic particles. Virus shape and size, alone, however, are insufficient for confirming specific virus types. Texture provides indispensable information, as many viruses show distinct and recurring texture patterns.

In the last two decades, much work in computer vision has focused on image textures. Early texture classification methods explored the statistical analysis of images. Representative of these statistical approaches are methods based on the co-occurrence matrix [3] and filtering [4]. In [5], Ojala et al. proposed a Local Binary Pattern (LBP) histogram for rotation invariant texture classification. LBP, along with its variants, is a simple yet efficient operator for describing the local image pattern and has achieved impressive classification results on benchmark datasets (see, e.g., [6] for virus image classification and [7] for protein subcellular localization) and in real-world applications (see, e.g., [8, 9]). During the last decade, LBP has distinguished itself by its simplicity, effectiveness, and robustness in detection of textural and structural information.

Despite recent advances in image texture analysis and the crucial role texture plays in TEM virus classification, few papers have examined machine analysis of virus textures in TEM images. Ring filters in the Fourier power spectrum were used as features in [10] and higher order spectral features were used in [11] to differentiate four icosahedral viruses. In [12] a radical density profile (RDP) was used to distinguish intensity variations between three maturation stages of human cytomegalovirus capsids in TEM images of cell sections. Finally, in [6] an ensemble combining LBP and RDP is used to discriminate fifteen virus types.

In this paper we compare the performance of some recent local binary pattern (LBP) variants in classifying TEM virus images. For each method, we implement its quinary coding version [7], explained in detail in section 2.1. For two quinary based approaches (the multithreshold local quinary pattern and the multithreshold local quinary configuration pattern), sequential forward floating selection (SFFS) [13] is used to select a set of optimal parameters for building the ensemble. An ensemble is also built by combining the different local phase quantization descriptors (LPQ), i.e., by varying their parameters (not just their filter size), using a ternary coding scheme instead of a binary one [7]. This set of LPQ descriptors thus combined is likewise chosen by SFFS. Moreover, a variant of a very recent method [14, 15] where the features are extracted considering the co-occurrence matrix as a 3D shape (SHAPE) is proposed. The proposed system is tested in a publicly available dataset located http://www.cb.uu.se/~gustaf/ virustexture /index.html. It is composed by 1500 images of 15 different virus types.

2 Texture Descriptors

2.1 Multithreshold Local Quinary Pattern (MLQP)

MLQP operator is a development of the canonical Local Binary Pattern (LBP) operator [5] which assigns a binary label to each pixel of an image based on the local information extracted from a circular neighborhood of P pixels and radius R according to the following equation:

$$LBP(P,R) = \sum_{p=0}^{P-1} s(q_p - q_c) \cdot 2^p$$

where

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

First, we introduced into the binary coding *s(x)* two more thresholds *(τ_1, τ_2)* thus arriving at the following quinary coding:

$$s(x,\tau_1,\tau_2) = \begin{cases} 2, x \geq \tau_2 \\ 1, \tau_1 \leq x < \tau_2 \\ 0, -\tau_1 \leq x < \tau_1 \\ -1, -\tau_2 \leq x < -\tau_1 \\ -2, x < -\tau_2 \end{cases}$$

To reduce the verbosity of the quinary encoded labels assigned to each pixel of the image, the quinary labels are split into 4 sets of binary patterns, according to the binary function $b_c(x), c \in \{-2,-1, 1,2\}$:

$$b_c(x) = \begin{cases} 1, x = c \\ 0, otherwise \end{cases}$$

By using the uniform rotation invariant (*riu2*) LBP mapping and considering the (P=8,R=1) and (P=16,R=2) neighborhoods, the final histogram is made of 112 bins.

The second step in defining MLQP is the threshold selection. As reported in [7], we chose a bunch of 25 threshold couples according to $\tau_1=\{1,3,5,7,9\}$ and $\tau_2=\{\tau_1+2, \ldots, 11\}$, each of them producing a single 112 bin histogram, i.e. a 112 valued feature vector useful for classification tasks. In case of a 2-class dataset, the 25 feature vectors, obtained from the 25 threshold couples, are then used for training 25 SVMs and the 25 different classification results are then fused together according to the sum rule. In case of a multi-class dataset (*m* classes), the "one vs all" classification is used:

- Training *m* SVMs for each of the 25 feature sets;
- Classifying each feature set (out of 25) with its own group of *m* SVMs, getting *m* partial scores (each relative to one class).
- Fusing, according to the sum rule, the 25 partial scores relative to the *m*-th class, thus getting m different final scores;
- Assigning the class out of the *m* classes according to the highest final score

2.2 Multithreshold Local Phase Quantization with Ternary Coding (MLPQ3)

A similar approach applied to the Local Phase Quantization (LPQ) operator, led to the Multithreshold LPQ with Ternary Coding (MLPQ3). LPQ is based on the blur invariance of the Fourier Transform Phase [14] and it labels each pixel of an image with a binary label based on the real and imaginary parts of the 2D Fourier Transform computed at 4 specific 2D-frequencies ($\mathbf{F_x}$ = [Re{ $\mathbf{F}_x^c$ }, Im{ $\mathbf{F}_x^c$ }]^T) for each neighborhood (size 3x3 or 5x5 pixels) centered in the pixel to be labeled. A detailed description of the operator in reported in [16].

We first replaced the original scalar quantizer:

$$q_j = \begin{cases} 1, g_j \geq 0 \\ 0, g_j < 0 \end{cases},$$

where g_j represents the j-th out of the 8 components of $\mathbf{F_x}$, with its ternary version:

$$q_j = \begin{cases} 1, g_j \geq \rho \cdot \tau \\ 0, -\rho \cdot \tau \leq g_j < \rho \cdot \tau \\ -1, g_j < -\rho \cdot \tau \end{cases},$$

where ρ is the standard deviation of the decorrelated F_x components and τ is a threshold.

The quantized coefficients are then represented as integers in the interval 0-255 using the following binary codings:

$$b_+ = \sum_{j=1}^{8} (q == 1) \cdot 2^{j-1}$$

and

$$b_- = \sum_{j=1}^{8} (q == -1) \cdot 2^{j-1}$$

b_+ and b_- values are then summarized in two distinct 256 bins histograms and the two histograms are concatenated thus providing a 512 valued feature vector (for both the neighborhood of size 3 and 5, i.e. the final feature vector is 1024 bins).

Then we applied the multithreshold approach by choosing 5 different thresholds *$\tau \in \{0.2, 0.4, 0.6, 0.8, 1\}$* and thus getting 5 feature sets. As done for MLQP, the "one vs all" classification is used considering the five feature sets (instead of the MLQP 25 ones).

We also built another ensemble, which we call MLQP-FE, by combining sets of LPQ descriptors varying different parameters: the filter size {3, 5}; the scalar frequency {0.8, 1.2, 1.6, 2}; the correlation coefficient between adjacent pixel values {0.75, 1.15, 1.55, 1.95}. A subset of all the possible combination is extracted using SFFS on the training data.

2.3 3D Shape

Texture is analyzed by means of the Gray Level Dependency Matrix (GLDM), which is a technique for measuring pixel transitions. Each element, or bin, of the matrix contains the number of occurrences of a specific transition between two grayscale levels, which are the bin coordinates in the matrix itself. Thus, GLDM is a 256 x 256 matrix irrespective of the dimension of the analyzed image. The pixel couples are chosen based on two parameters, the distance and the angle at which they are taken.

To simplify the texture analysis task, the GLDM is analyzed by means of a number of features that represent a compact way of describing its shape. The GLDM is a three-dimensional function that shows a strong and large peak along the principal diagonal, i.e., along the region representing pixel couples of very similar grey levels that are very likely to be found, as it can be seen in figure 1 (left). The shape of such a peak, also called the main (or principal) component, is analyzed by considering level curves of the GLDM at different heights. Since each curve can be made of more than one contour, the largest one is identified as the main component, which is approximated with an ellipse to simplify the analysis.

Once the main component has been evaluated at multiple heights, ranging from 1 to 20, features are calculated depending on the evolution of the shape of the ellipses over the different height values (obviously, the ellipses at lower heights will have larger areas). This is summarized in figure 1, in which a three-dimensional view of a GLDM is shown (left) together with the result of the main component detection at two different heights (center, right).

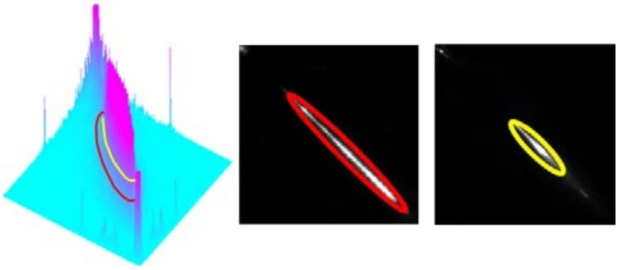

Figure 1: An example of GLDM (left) on which two level curves are considered. In the central and right images, the main component has been identified and approximated with an ellipse at the two heights.

The proposed features are used to measure the following characteristics (details in [7][8]):

1. Evenness: measures how evenly the minor axis of the ellipses decreases.
2. Minor axis spread: the difference between the minimum and maximum values for the ellipse minor axis.
3. Minimum value for the minor axis.
4. Average of the height/width ratio among all ellipses.
5. Total volume under the level curves.
6. Area of the smallest ellipse.
7. The ratio between the smallest and largest ellipses.
8. Volume of the peak, measured as the volume of the GLDM which is above the highest level curve.
9. Number of blank locations.

The above features measure the GLDM in several ways: some of them concentrate on its central part (features 1, 2, 4), while others aim at relating the upper and lower parts (7), and another group focuses on the upper part (3, 6, 8). Feature 5 considers the area under the highest level curve, that is a general characteristic of the GLDM; finally, feature 9 is the only one that does not focus on the principal component, but rather on the surrounding area.

The process described above provides good performances when the GLDM has a large volume, i.e. the sum of the bin heights is large, which leads to a well defined shape of the GLDM itself. Such volume depends on the number of couples of the analyzed image pixels, which in turn depends on the size of the analyzed image, and also on the distance and orientation of the pixel couples. This last dependency is very weak, therefore it is possible to approximate the GLDM volume with the number of pixels of the considered image: this means that the performance of the aforementioned process drops when it is applied to images with small dimensions. From practical experience, this problem appeared when using input images composed of less than 10000 pixels: each level curve was composed of very few points, that made it impossible to get a reasonable result for the elliptic fitting. This problem becomes more evident at higher levels, so it can happen that the analysis can be performed only on a reduced number of levels. In this situation it is still possible to perform the texture analysis by neglecting the levels for which the elliptic fitting cannot be obtained: this is done by forcing the feature values to a conventional value of 0.

For the tested approaches the GLDMs are obtained using d=1 and d=3 with an angle of {0°, 45°, 90°, 135°}. We test several methods based on the co-occurrence matrix:

- *5S*, where five shape based descriptors are combined (each used for training a different SVM, then the set of SVMs is combined by sum rule), the first extracted from the whole co-occurrence matrix; the others from sub-windows of the GLDM (from the coordinates: (0, 0) to (127, 127); (128, 128) to (255, 255); (0, 0) to (191, 191); (64, 64) to (255, 255)).
- *13S*, where we combine thirteen shape based descriptors, the five of *5S*, and other 8 extracted from sub-windows of the GLDM: (0, 0) to (63, 63); (31, 31) to (95, 95); (63, 63) to (127, 127); (95, 95) to (159, 159); (127, 127) to (191, 191); (159, 159) to (223, 223); (191, 191) to (255, 255); (63, 63) to (191, 191).

Here the thirteen descriptors are combined by weighted sum rule[1], the weights of the first 5 descriptors are 1, while the weight of the last 8 descriptors is 0.5.

.

3 Experimental Results

In this work we used the "object scale" virus dataset tested in [6] available at http://www.cb.uu.se/~gustaf/virustexture /index.html. The dataset is composed by 1500 images of 15 different virus types, with the radius of each virus particle is represented by 20 pixels (see [6] for more details).

The following approaches[2] are compared (mean accuracy as performance indicator) in Table 1a:

- HAR, standard method based on 13 features introduced by Haralick[3];
- 5S, the method defined in section 2.3;
- 13S, the method defined in section 2.3;
- NewH, the fusion by weighted sum rule between 13S and HAR[4], where the weight of HAR is 1 while the weight of 13S is 0.51;

The following approaches[5] are compared (mean accuracy as performance indicator) in Table 1b:

- LBP, standard local binary patterns;
- LTP, standard local ternary patterns;
- ELB, the method proposed in [17];
- MLQP, standard multithreshold local quinary pattern [7];
- PLB, the method proposed in [18];
- PLQ, the variant of PLB where the quinary coding is used instead of the standard binary coding;
- NTB, the method proposed in [19];
- NTQ, the variant of NTB where the quinary coding is used instead of the standard binary coding;
- DLB, the method proposed in [20];
- DLQ, the variant of DLB where the quinary coding is used instead of the standard binary coding;
- LCP, the method proposed in [21];
- MLC, the variant of LCP where the quinary coding is used instead of the standard binary coding;
- FE1, a subset of the MLQP descriptors[6] is selected by SFFS for maximizing the performance using only the training data;
- FE2, as FE1 but is a subset of MLC to be selected.

The following approaches are compared in Table 1c:

- Morph, the method proposed in [22] (here we do not consider the features based on the Haralick's approach, since they are already reported in Table 1a);
- FRDP [6], it is the best method in the "object scale dataset" (i.e. the same used in this work) reported in [6]. We use the original code shared by the authors coupled with SVM.
- LPQ, concatenation of the features extracted by local phase quantization with radius 3 and 5;
- MLPQ3, the multi-threshold approach proposed in [7] (exactly the same parameters).
- MLPQ3-FE, selection of a subset of the different descriptors LPQ descriptors varying different parameters (see section 2.2) by SFFS as in FE1.
- FUSION, fusion by sum rule4 among NewH, MLC, Morph and FRDP (the best 4 methods, each belong to a different type of descriptors).

From the results reported in the previous tables we can make the following conclusions:

- Multithreshold quinary approach outperforms the base approaches (i.e. MLQP outperforms LBP, PLQ outperforms PLB, NTLQ outperforms NTLB, DLQ outperforms DLB, MLC outperforms LCP);
- Selection of the set of parameters boost the performance of MLQP but does not improve MLC;
- In this classification problem MLPQ3 works poorly since some feature sets composing the ensemble got low performances (the descriptors that belong to MLPQ3 obtains an accuracy of 62.9%, 58.9%, 52.7%, 49.6% and 48.2%) but MLPQ3-FE outperforms LPQ since the low performance descriptors are not selected by SFFS;
- The proposed NewH approach outperforms standard Haralik's features;
- As widely reported in different classification problems, the fusion of different descriptors (i.e. the method named FUSION) obtains the best performance.

.

4 Conclusion

In this paper we compare different LBP variants and for the first time we report the performance of their multithreshold

[1] Notice that the weights are not optimized in this dataset but we run experiments on ~10 different datasets using the same weighs, these experiments are not yet published

[2] For all the tested approaches the uniform rotation invariant LBP mapping is used and considering the (P=8,R=1) and (P=16,R=2) neighborhoods

[3] Energy; Correlation; Inertia; Entropy; Inverse difference moment; Sum average; Sum variance; Sum entropy; Difference average; Difference variance; Difference entropy; Information measure of correlation 1; Information measure of correlation 2

[4] Before the fusion the scores of both the approaches are normalized to mean 0 and std 1

[5] For all the tested approaches the uniform rotation invariant LBP mapping is used and considering the (P=8,R=1) and (P=16,R=2) neighborhoods

[6] We have used more couple of thresholds with respect to standard MLQP:

```
for threshold=1:2:15
    for threshold2=threshold+2:19
…. Feature extraction with thresholds [threshold, threshold2]
    end
  end
  for threshold=1:2:15
    for threshold2=threshold+4:19
…. Feature extraction with thresholds [threshold, threshold2]
    end
  end
```

quinary variants. The main novelty of the paper is the experimental assessment of the usefulness of the quinary coding coupled with different LBP-based new descriptors. Another interesting result is that we show a method for improving the performance obtained with the information extracted from the co-occurrence matrix.

In the original paper [6], where the dataset used in this paper was proposed, the authors obtain a mean accuracy of 73.8% and a median accuracy of 79.0% (personal communication of the authors) using FRDP and performances lower than 60% with LBP and its variants (on the “object scale" dataset). In the “fixed scale" dataset (not available) the best result in [6] (median accuracy ~80%) is obtained by an LBP variant. Moreover, the authors report that is useful to combine descriptors extracted from the “fixed scale" dataset and descriptors extracted from the “object scale" dataset.

Our tests on FRDP obtain lower performances, but our texture variants obtain good performance (mean accuracy) ~70% (notice that we have used the same 10-fold cross validation used by in [6]) and our fusion outperforms the result obtained in [6] using the “object scale" dataset (their best method obtains a mean accuracy of 73.8% while our fusion obtains a mean accuracy of 80.7%).

a)

HAR	5Sh	13Sh	NewH
69.9	59.3	60.5	71.7

b)

LBP	LTP	ELB	MLQP	PLB	PLQ	NTB	NTQ	DLB	DLQ	LCP	MLC	FE1	FE2
57.6	58.5	70.9	70.0	64.0	70.1	49.9	68.5	55.3	71.8	62.7	73.3	72.4	72.7

c)

Morph [1]	FRDP [3]	LPQ	MLPQ3	MLPQ3-FE	FUSION
71.7	70.0	63.3	57.9	64.5	80.7

Table 1. Comparison among the tested methods, (N.B. we report the mean accuracy among the classes and not the median as in [6]).

Acknowledgements

The authors would like to thank Dr. Gustaf Kylberg for sharing the virus dataset and his Matlab code as well as for the fruitful communication during the draft of this paper.

References

[1] C. S. Goldsmith and S. E. Miller, "Modern uses of electron microscopy for detection of viruses," *Clinical Microbiology Reviews*, vol. 22, pp. 552-563, 2009.

[2] S. S. Biel and D. Madeley, "Diagnostic virology-the need for electron microscopy: A discussion paper," *Journal of Clinical Virology* vol. 22, pp. 1-9, 2001.

[3] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Texture features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 3, pp. 610-621, 1973.

[4] T. Randen and J. H. Husy, "Filtering for texture classification: A comparative study," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 21, pp. 291-310, 1999.

[5] T. Ojala, M. Pietikainen, and T. Maeenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Ieee transactions on pattern analysis and machine intelligence,* vol. 24, pp. 971-987, 2002.

[6] G. Kylberg, M. Uppström, and I.-M. Sintorn, "Virus texture analysis using local binary patterns and radial density profiles," presented at the 18th Iberoamerican Congress on Pattern Recognition (CIARP), 2011.

[7] M. Paci, L. Nanni, A. Lathi, K. Aalto-Setälä, J. Hyttinen, and S. Severi, "Non-binary coding for texture descriptors in sub-cellular and stem cell image classification," *Current Bioinformatics,* 2013.

[8] N. Hervé, A. Servais, E. Thervet, J. C. Olivo-Marin, and V. Meas-Yedid, "Statistical color texture descriptors for histological images analysis.," presented at the Proc. of IEEE International Symposium on Biomedical Imaging (ISBI), 2011.

[9] B. Zhang, "Classification of subcellular phenotype images by decision templates for classifier ensemble," presented at the 2009 International Conference on Computational Models for Life Sciences (CMLS), 2010.

[10] B. J. Matuszewski and L. K. Shark, "Hierarchical iterative bayesian approach to automatic recognition of biological viruses in electron microscope images," presented at the 2001 International Conference on Image Processing (ICIP), 2001.

[11] H. C. L. Ong, "Virus recognition in electron microscope images using higher order spectral features," Ph.D. Thesis, Queensland University of Technology, 2006.

[12] I. M. Sintorn, M. Homman-Loudiyi, C. Söderberg-Nauclér, and G. Borgefors, "A refined circular template matching method for classification of human cytomegalovirus capsids in TEM images," *Computer Methods and Programs in Biomedicine*, vol. 76, 2004.

[13] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 5, pp. 1119-1125, 1994.

[14] S. Ghidoni, G. Cielniak, and E. Menegatti, "Texture-based crowd detection and localisation," presented at the International Conference on Intelligent Autonomous Systems (IAS-12), 2012.

[15] L. Nanni, S. Brahnam, S. Ghidoni, and E. Menegatti, "A comparison of methods for extracting information from the co-occurrence matrix for subcellular classification," submitted.

[16] V. Ojansivu and J. Heikkila, "Blur insensitive texture classification using local phase quantization," presented at the ICISP, 2008.

[17] L. Liu, L. Zhao, Y. Long, G. Kuang, and P. Fieguth, "Extended local binary patterns for texture classification," *Image and Vision Computing*, vol. 30, pp. 86-99, 2012.

[18] X. Qian, X.-S. Hua, P. Chen, and L. Ke, "PLBP: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognition Letters*, vol. 44, pp. 2502-2515, 2011.

[19] A. Fathi and A. R. Naghsh-Nilchi, "Noise tolerant local binary pattern operator for efficient texture analysis," *Pattern Recognition Letters*, vol. 33, pp. 1093-1100, 2012.

[20] Y. Guo, G. Zhao, and M. Pietikainen, "Discriminative features for texture description," *Pattern Recognition Letters*, vol. 45, pp. 3834-3843, 2012.

[21] Y. Guo, G. Zhao, and M. Pietikainen, "Texture classification using a linear configuration model based descriptor," presented at the British Machine Vision Conference, 2011.

[22] P. Strandmark, J. Ulén, and F. Kahl, "HEp-2 Staining Pattern Classification," presented at the International Conference on Pattern Recognition (ICPR2012), 2012.