 DOI: 10.3217/978-3-85125-603-1-16

Image texture classification with morphological amoeba descriptors

Franz Schwanninger[1] and Martin Welk[2]

***Abstract*— We investigate the applicability of quantitative texture descriptors based on morphological amoebas in the context of a machine learning approach to texture classification. Morphological amoebas are a type of contrast-adaptive structuring elements originally designed for adaptive morphological image filters, and they stand in a close relation to local edge-weighted pixel graphs of an image. A recently introduced class of texture descriptors is obtained by applying graph indices from quantitative graph theory to those pixel graphs. Additionally we consider descriptors that refer to the geometric shape of the amoebas. In our approach, these descriptors are histogram encoded and fed into a linear support vector machine (SVM). We demonstrate our approach using a small number of texture samples from the VisTex database as training data. In further experiments, we study how selected parameters of the amoeba construction influence the classification performance.**

I. INTRODUCTION

In this paper we consider texture-based image classification. Textures are intrinsic structures of image regions and can be classified into different categories. Machine learning as well as other approaches that aim at classifying images depending on their textural content often make use of quantitative texture descriptors [10]. In this paper, we focus on descriptors that are constructed from *morphological amoebas.* Originally introduced as structuring elements in adaptive mathematical morphology [15], morphological amoebas are of interest for texture analysis as they encode local image structure [22]. Their construction is inherently related with subgraphs of the edge-weighted pixel graph of an image in which edge weights are computed from image contrast. Using these subgraphs as input for the computation of *graph indices* gives rise to a class of graph-based texture descriptors [22] from which part of the descriptors in this paper are chosen. To these, we add descriptors obtained by evaluating geometric information of the amoebas. The descriptors are then encoded and forwarded to Support Vector Machines (SVMs) [3] where they are aggregated in order to enhance the classification performance. This scalable approach is demonstrated by two graph-index based texture descriptors and one texture descriptor encoding geometric amoeba properties.

*This work was not supported by any organization

[1]The work presented here is part of the master thesis of Franz Schwanninger at the Department of Biomedical Informatics and Mechatronics, Private University for Health Sciences, Medical Informatics and Technology (UMIT), 6060 Hall/Tyrol, Austria, franz.schwanninger@edu.umit.at

[2]Martin Welk is with Department of Biomedical Informatics and Mechatronics, Private University for Health Sciences, Medical Informatics and Technology (UMIT), 6060 Hall/Tyrol, Austria, martin.welk@umit.at

II. RELATED WORK

Image texture analysis has been investigated for a long time. Haralick [9] shows an early overview of structural and statistical approaches; others show frequency-based models [14], filter banks [16], or fractals [20].

Morphological amoebas as spatially adaptive neighborhoods have been introduced by Lerallut et al. [15]. Our short recap on the amoeba construction follows the presentation in [23] which is already adapted to the combination with graph indices in order to construct texture descriptors.

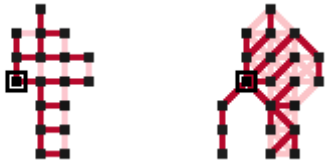
Graph indices have their roots in the analysis of molecular graphs [25], [1], [19] but have meanwhile developed into an important tool for a broad range of network analysis tasks [5]. Although graph models have been widely used in image analysis, see [17], methods originating from quantitative graph theory – like graph indices – have not played a significant role in texture analysis so far. By the construction of texture descriptors from graph indices and amoebas in [22], a first step into this direction has been made. The applicability of these descriptors to image texture segmentation with geodesic active contours has been investigated in [24]. In these works, the texture descriptors are evaluated by simple local statistics and thresholds.

On the other hand, machine learning approaches play an important role in modern image processing [12], [4], [13]. During the past two decades, SVMs [3] have been used for virtually all kinds of classification tasks in image processing.

III. AMOEBA CONSTRUCTION

In this section we describe in more detail the construction of amoebas. We assume that images are represented by a regular grid of pixels. The pixels can be interpreted as the vertices of an edge-weighted graph, the *pixel graph*, the edges of which connect adjacent pixels. Regarding what pixels are considered as adjacent, the most common choices are 4-neighborhoods, where adjacency is restricted to vertical and horizontal neighbors, and 8-neighborhoods that include also diagonal neighbors. Figure 1 shows two pixel graphs for the local environment of the seed pixel v_0 highlighted in the center according to the methods described later on in this section: one resulting from a 4-neighborhood and one from a 8-neighborhood.

Following [22], amoebas are constructed using Dijkstra's shortest path algorithm [7]. Similar to a region growing-based approach, local neighborhoods are established by using the neighborhood methods stated before. Starting from the seed pixel, region growing iteratively includes new neighbors to the graph until their distance from the seeding pixel



(a) 4-neighborhood (b) 8-neighborhood

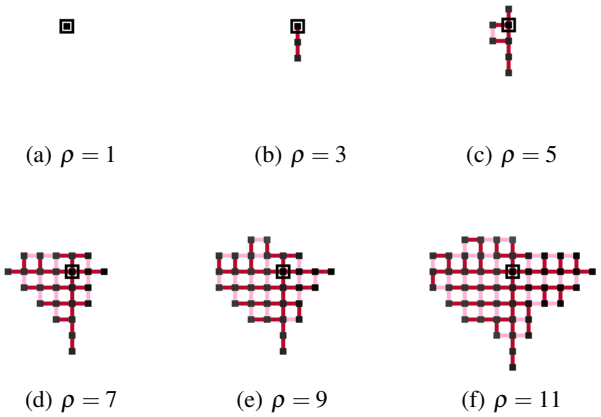
Fig. 1: Pixel graphs based on 4-neighborhoods and 8-neighborhoods for the same originating pixel. The Dijkstra tree is shown in dark red, all remaining edges in light red.

exceeds ρ. The edge weights, which enforce locally adaptive amoeba shapes, are defined according to [22] as

$$w_{p,q} := \left(\|p-q\|^2 + \beta^2 |u_p - u_q|^2 \right)^{1/2}, \tag{1}$$

where $\|p-q\|$ is the Euclidian distance between the vertices p and q; $|u_p - u_q|$ denotes the gray value difference of the corresponding pixels. The *contrast scale* β allows to weight between both values and will be subject to closer investigation in Section VIII.

Figure 2 shows how amoebas evolve for varying ρ together with their Dijkstra trees.



(a) $\rho = 1$ (b) $\rho = 3$ (c) $\rho = 5$

(d) $\rho = 7$ (e) $\rho = 9$ (f) $\rho = 11$

Fig. 2: Amoebas and Dijkstra shortest path trees for one pixel at different ρ. The seeding pixel is in the center.

Amoebas and their respective pixel graphs can be computed for each pixel in an image. However, for the texture classification presented in this paper it might also suffice to compute these for a subset of pixels.

In the following section, we will review some texture descriptors obtained from amoebas and their respective Dijkstra trees.

IV. AMOEBA DESCRIPTORS

Suitable descriptors for texture include graph indices as well as other features encoding amoeba properties. As shown in a large-scale evaluation [6], there exist hundreds of possible candidates which are derived from vertex distances, information-theoretic methods, or edge connectivity. For our current investigation we only use a few selected features, based on previous work [22]. The presented descriptors are invariant to the pose of a texture or a respective object in an image. This is an advantage over some other approaches, like neural networks.

A. GRAPH INDICES

Quantitative graph descriptors can be computed for a graph G. Here, graph indices are obtained from the Dijkstra trees that have been extracted from amoebas before. They can further be divided into distance-based indices, such as the Wiener index and the Harary index, and information-theoretic concepts, such as the Bonchev-Trinajstić information indices or Dehmer entropies. Some of the previously investigated graph indices in [22] are similar to each other, such as the Wiener and Harary indices. We would like to avoid such redundancies because we aim for encoding distinct amoeba properties in order to achieve good classification results.

The **Wiener index** is defined as

$$W(G) := \sum_{1 \le i < j \le n} d(i,j). \tag{2}$$

with $d(i,j)$ representing the distance between vertices v_i and v_j. We complement the distance-based approach of the Wiener index by adding a functional based on a **Dehmer entropy** [5] and derived in [22]. It reads

$$f^V(v_0) = \mathrm{e}^{M \sum_{j=1}^{n} q^{d(0,j)}} \tag{3}$$

where v_0 is the seeding vertex as used in the region growing approach. For our current investigation, we follow [22] in fixing the parameters to $M = 1$ and $q = \mathrm{e}^{-0.1}$. There are candidates for additional descriptors, including the methods shown in [22]. In the context of a machine learning based approach, the texture classification does not benefit from a high number of descriptors if they do not encode additional amoeba properties. Depending on the pixel graph they are based on, some graph indices, like the Bonchev-Trinajstić information indices, generate values that are extremely spread between high or low values. That makes them harder to handle within a machine learning framework, where a proper scaling or normalisation of features is required. Preprocessing by suitable transformations of the value range may be necessary for these.

B. GEOMETRIC AMOEBA DESCRIPTORS

The second class of amoeba descriptors to be considered does not rely on graph-based concepts like the graph indices shown in the section before. Instead, geometric features of the amoeba shapes are used. In order to compute the eccentricity of an amoeba A given by the set V of its pixels, we start by introducing the moments

$$m_{l,k} = \sum_{V} \left(u(x,y) \cdot (x - \bar{x})^j \cdot (y - \bar{y})^i \right) \tag{4}$$

where $(\bar{x},\bar{y})$ is the mass center of V,

$$\bar{x} = \frac{m_{10}}{m_{00}}\,,\ \bar{y} = \frac{m_{01}}{m_{00}}\,, \tag{5}$$

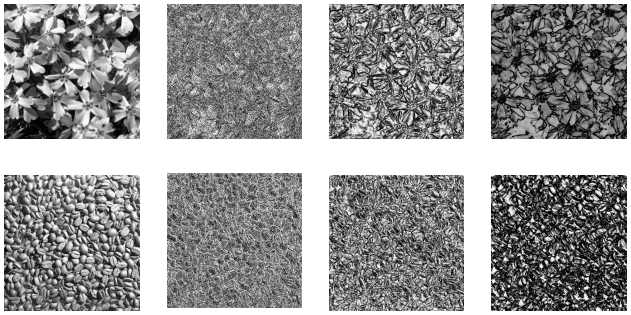
and $u(x,y)$ represents the grayvalue in the respective location. We write down the matrix of second order moments for the respective combinations of l and k as

$$M = \begin{bmatrix} m_{02} & m_{11} \\ m_{11} & m_{20} \end{bmatrix}. \tag{6}$$

From the spectral decomposition of M, we obtain the major moment $m_{\max}$ as the larger eigenvalue, and the minor moment $m_{\min}$ as the smaller one. The **eccentricity** of the amoeba is then given by

$$\varepsilon(A) = \sqrt{1 - \frac{m_{\min}}{m_{\max}}}, \tag{7}$$

and serves as a geometric amoeba descriptor. Further candidates for descriptors of this class include other measures derived from moments, like Hu-moments [11], but are not included in the current investigation. Figure 3 shows the amoeba descriptors that have been presented so far for two exemplary gray-value images.



(a) u_g (b) $f^V(v_i)$ (c) $\varepsilon(A)$ (d) $W(G)$

Fig. 3: Examples for texture classes *Flowers2* and *Food1*: Grayscale image, Dehmer entropy, Eccentricity and Wiener index, $\beta = 0.5, \rho = 11$. The input images are converted to grayscale, downscaled and clipped from the *VisTex* database [18], see Fig. 4.

V. FEATURE ENCODING

The texture descriptors obtained in the last section do not yield good results when directly applied to machine learning on a pixel-based approach due to their spatial variation. Therefore the descriptors are subjected to an encoding step before forwarding them to SVMs.

Prior to feature encoding, we rearrange the computed values of the N_D descriptors in a matrix structure with one column per descriptor, and $M \times N$ rows each corresponding to one pixel of a training image. Here, M is the number of rows, and N the number of columns of the training images. For training, square and randomly sampled subimages with $M = 64$ and $N = 64$ are used. Using the three amoeba-based descriptors presented before, this results in a 4096×3 matrix. For the following training tasks, the columns of this matrix are interpreted as feature vectors

$$\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{M\times N} \in \mathbb{R}^{N_D}. \tag{8}$$

The subsequent encoding is based on **histograms** obtained from cluster information. Clustering is here applied to the full training set that includes several training images for each class. In this way, we obtain a *textural vocabulary* comparable to the *visual vocabulary* as stated in [2]. It describes all possible textural variants present in the training set, and encodes each training sample as a histogram, composed of the number of occurencies of the respective texture in an image.

First of all, feature vectors are obtained for all classes, all graph indices and all images in the training set. The training set is encoded as

$$\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{M\cdot N\cdot N_C\cdot N_I} \in \mathbb{R}^{N_D}. \tag{9}$$

Note that for the given subimage size as well as for the number of classes $N_C = 16$ and for the training images per class N_I, the structure can potentially be very large. However, this is feasible for the given small dataset with a low number of training samples; for larger databases a Monte-Carlo approach can be considered.

Clustering through K-means is applied to the feature vectors from (9). The resulting cluster centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^{N_D}$ represent the textural vocabulary on which the following classification is based. Each of the training image samples $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{M\cdot N} \in \mathbb{R}^{N_G}$ is assigned to its closest cluster, with the assignments given as $q_1, \ldots, q_{M\cdot N} \in \{1, \ldots, K\}$. The histogram $\boldsymbol{f}_{\text{hist}} \in \mathbb{R}^K$ is then given by $\left[\boldsymbol{f}_{\text{hist}}\right]_k = |\{i : q_i = k\}|$.

The training data for the entire data set then results in

$$\boldsymbol{f}_{\text{hist},1}, \ldots, \boldsymbol{f}_{\text{hist},N_C\cdot N_I} \in \mathbb{R}^K, \tag{10}$$

describing a K-dimensional feature space, where K is the number of clusters chosen for the K-means clustering, that contains training data for all N_C classes and N_I training samples per class. The data set for classification testing is obtained in the same way.

Table I shows the textural vocabulary, using the three graph indices shown in the columns.

VI. TRAINING DATA

As our focus is on image data with rich texture information, we choose the VisTex database [18] to test the discriminative power of texture descriptors. This database is well-suited for such a task as it contains many images with pure texture information without additional objects, variation in lighting or occlusions, which might distract from the desired goal. A subset of the original color images from the VisTex database, each with a resolution of 512×512 pixels, is manually grouped by visual textural similarity prior to classification, resulting in 16 classes. Some classes of the original database contain visually highly dissimilar images and are therefore split into subclasses. For instance, the class *Fabric* is divided into six subclasses that can easily

TABLE I: Textural vocabulary, represented by cluster centers, for the selected texture descriptors, $\rho = 7, \beta = 0.29$ and $K = 16$

k	$f^V(v_0)$	$\varepsilon(A)$	$W(G)$
1	116.4	16.2	110.8
2	166.9	22.3	147.2
3	136.4	60.0	124.1
4	120.7	10.6	150.9
5	178.7	72.7	148.9
6	224.4	91.2	177.2
7	210.2	42.0	170.5
8	112.0	53.9	82.6
9	108.0	12.4	65.3
10	106.0	46.9	15.3
11	165.9	127.6	140.3
12	151.8	0.0	6.3
13	229.5	151.6	180.5
14	117.8	116.0	87.0
15	254.7	254.8	251.0
16	110.2	246.1	71.7

be distinguished visually. Our final set of training samples is composed of 56 images. All images are converted to gray scale according to ITU Rec. 601,

$$Y \leftarrow 0.299\,R + 0.587\,G + 0.114\,B, \tag{11}$$

where Y, R, G, B denote the grayvalue and the red, green, and blue intensities, respectively.

Note that the class assignment has been made without attention to the strengths or weaknesses of the methods investigated in this paper. There are classes that may be discriminated easily as well as classes that may be very hard to discriminate from others, or have large intra-class variation, which poses a challenge to texture classification.

Table II displays the assignment of images from the original VisTex database to classes in building the training set. The unique identifier used as an alternative to the subset name is found in the column *idx*, while the column *indices* indicates the original index of the image, for instance, 7 in the first row is the index of Bark1.0007.png. Finally, *imagecount* represents the total number of images in each respective class.

TABLE II: VisTex subclass labels

Subsetname	*idx*	*indices*	*imagecount*
Bark1	a	7,8,9,10,11,12	6
Fabric1	b	0,1,2,3	4
Fabric2	c	8,9,10	3
Fabric3	d	11,12	2
Fabric4	e	13,14	2
Fabric5	f	15,16	2
Fabric6	g	18,19	2
Flowers1	h	0,1	2
Flowers2	i	2,3	2
Flowers3	j	4,5,6,7	4
Food1	k	2,3,4	3
Grass1	l	1,2	2
Metal1	m	1,2,3,4,5	5
Sand1	n	0,1,2,3,4,5,6	7
Stone1	o	4,5	2
Water1	p	0,1,2,3,4,5,6,7	8

We intend that the classification methods that are investigated in the following, achieve a good performance using small training data sizes. However, many approaches to machine learning would require large amounts of training data.

The training samples originate from these classes as 64×64 subimages that have been obtained from the 512×512 images. To compensate for the small size of the data set and the fact that some of the subclasses consist of as few as two images, we generate a larger data set by extracting subimages of size 64×64 as stated in the previous section. The subimages are taken from random locations in the large images.

Furthermore, to avoid boundary effects, we compute amoebas only for seed pixels which have a distance of at least ρ from the image boundary, where ρ is the maximum amoeba radius.

Images containing strong color information like the flower classes could easily be distinguished from others by comparing their colors. As we aim for classification through texture information only, we perform a grayvalue conversion on all images before applying further methods. The average grayvalue or brightness of an image will not directly influence texture descriptors presented above. Figure 4 shows one example image in color for each class.

We employ two distinct data sets for the training and evaluation of the machine learning framework. Thereby, weaknesses like overfitting would be well detected: An algorithm suffering from overfitting would perform well on the training set but would fail to generalize, and thus achieve bad performance on unseen test data. In generating our training data set, the subimage extraction step as described above takes a similar role as common alternative methods for enlarging small data sets such as the data augmentation techniques that are popular in the context of neural networks. As in the shown approach both data sets are based on the same images, they may contain the same image regions and thus are not fully independent from each other. Another strategy to training set organization that could be considered in order to overcome the limitations of small data sets like the VisTex database would be n-fold cross-validation.

VII. TRAINING ARCHITECTURE

Applying a classic machine learning approach, we start by extracting a number of descriptors from an image. The resulting features are encoded using histogram encoding as described above, and then handed over to linear SVMs for training. We refrain from using higher-order kernels, as they suffer from overfitting in this task. To assess how well the methods perform on the testing data set, we measure the accuracy of the classification. Higher accuracy measures indicate better performance. It is well-known that the accuracy score does not assess classification properly if the data set is unbalanced [21]. In our setting, this should not constitute a problem since we use equal data set size for all classes during training. Accuracy is therefore considered an appropriate tool for basic performance evaluation. We refrain thus from

(a) Bark1 (b) Fabric1 (c) Fabric2 (d) Fabric3

(e) Fabric4 (f) Fabric5 (g) Fabric6 (h) Flowers1

(i) Flowers2 (j) Flowers3 (k) Food1 (l) Grass1

(m) Metal1 (n) Sand1 (o) Stone1 (p) Water1

Fig. 4: Example images for each VisTex class obtained from the VisTex database [18]. ©1995 Massachusetts Institute of Technology. Developed by Rosalind Picard, Chris Graczyk, Steve Mann, Josh Wachman, Len Picard, and Lee Campbell at the Media Laboratory, MIT, Cambridge, Massachusetts. Under general permission for scholarly use.

using additional values like recall, precision or mean average precision (mAP), which are popular performance indicators in recent research [8].

SVMs, as introduced by Cortes and Vapnik [3], are a standard method for binary classification, still, they can be applied to a given multiclass classification task. For the discrimination of two classes, one SVM is sufficient. For multiclass classification, as shown here for 16 classes, multiple SVM classifiers are necessary. For each SVM, the current class is interpreted as positive, whereas all other classes are treated as negative. Effectively, one must train one SVM per class, which comes down to 16 SVMs in our setting. Figure 5 summarizes all steps taken to classify texture within this classic machine learning framework.

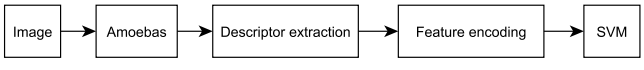


Fig. 5: Classic machine learning model for images applied to texture classification with amoeba descriptors

For the following trainings we use 8 training samples per class and histogram encoding, as well as 32 samples in a testing set for accuracy estimation.

VIII. RESULTS

In this section we evaluate how amoeba-based texture descriptors perform within the previously described architecture with histogram feature encoding. Some of the parameters involved have a large impact on the performance of this approach. We will therefore subject these to a closer investigation.

The size of the textural vocabulary obtained by the training stage depends on the number of clusters defined by the clustering method. Figure 6 shows the classification accuracy for different choices of K between 8 and 64. The values for K are chosen based on the number of classes and different textures used in this data set. As the given data set contains 16 classes, the values of K in our tests are chosen around 16.

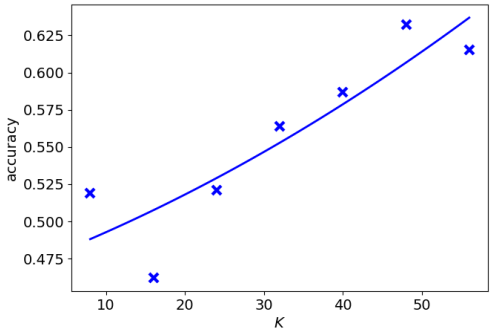


Fig. 6: Accuracy for varying K with $\beta = 0.29$ and $\rho = 7$ and 4-neighborhoods

As can be seen, in general, larger K leads to better performance, with the accuracy reaching a plateau for $K = 48$ and higher. For our further tests, we retain $K = 48$, which warrants faster computation than larger values. Furthermore, we avoid the problems that come with high-dimensional feature spaces and are known as "the curse of dimensionality". As expected, choosing $K < N_C$ results in a performance drop, as different textures have specific cluster centers which tend to be ignored in this case.

The construction of amoebas includes many possible design choices than can be made, while some restrictions are given by the current implementation. Table III gives an overview of the choices.

TABLE III: Amoeba parameters for closer investigation

Parameter	choices
Neighborhood	4-nbhd, 8-nbhd
Norm type	L^1, L^2, L^∞
Patch type	Euclidian patch, amoeba
Edge-weight type	weighted, unweighted
Graph type	fully connected, Dijkstra tree
β	0.10…0.43
ρ	3…9

As described in the introduction, possible choices for the local neighborhood include 4-neighborhoods and 8-

neighborhoods. Alternatives to the L^2 norm in the computation of the distance metric are L^1 or L^∞ norms. Instead of locally adaptive amoebas, fixed Euclidian patch may be considered as mentioned already in [22], [23]. Graph indices can be computed either from fully connected pixel graphs, from Dijkstra trees or even from unweighted Dijkstra trees where the edge weights have been stripped off.

For closer investigation we stick with the L^2 norm in amoeba computation as already mentioned in earlier sections, and restrict ourselves to the Dijkstra tree with edge weights. The amoeba parameters β and ρ have a large potential impact on the classification. Therefore, we vary their values within the boundaries stated in Table III. Classification accuracies for sampled values of ρ with 4-neighborhoods as well as 8-neighborhoods are compiled in Table IV.

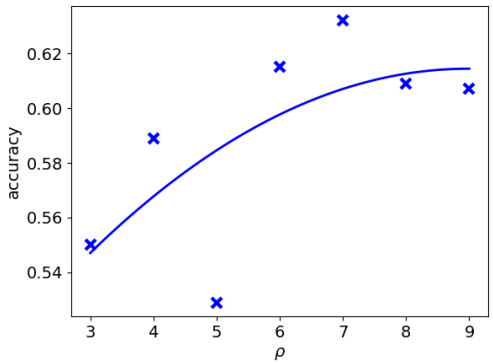
TABLE IV: Classification accuracies for different neighborhoods, and ρ, $\beta = 0.29$

ρ	4-nbhd	8-nbhd
3	0.533	0.548
4	0.589	0.544
5	0.529	0.595

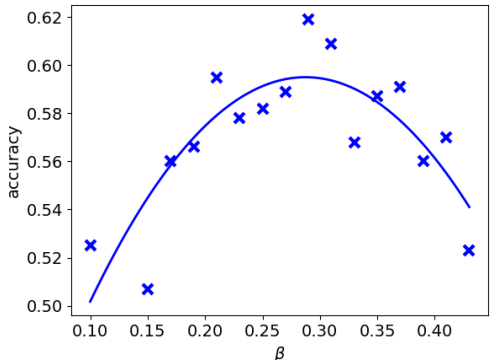
The average values for each parameter from Table IV are shown in Table V.

TABLE V: Average accuracies for parameter values from Table IV

4-nbhd	8-nbhd
0.550	0.562



(a) Accuracy for varying ρ, $\beta = 0.29$ and 4-neighborhoods



(b) Accuracy for varying β, $\rho = 6$ and 4-neighborhoods

Fig. 7: Accuracy for varying ρ and β and their second order polynomial regression

A closer look at the parameters ρ and β is given in Figure 7. The accuracy as a function of β shows a maximum at 0.29, while values in a wider range $\beta \in \{0.20 \ldots 0.40\}$ may still achieve good results. Each data point represents the result of one test set. We expect that using multiple test sets per point would improve the smoothness of the graphs. No class discrimination is possible for $\beta = 0$, as in this case each amoeba has a regular round shape and encodes no texture information whatsoever.

Based on the results, we can summarize our findings regarding the parameters as follows:

- Single values for accuracy vary when using 4-neighborhoods or 8-neighborhoods, there is no clear trend which performes better. Since 8-neighborhoods come with higher computational demand, further investigation will be based on 4-neighborhoods.
- β is best set to lower values. In the test runs shown, best outcomes were observed for $\beta = 0.29$. A more detailed analysis showed that smaller or larger values may still lead to good results. The performance almost continuously decreased for $\beta > 0.3$.
- In most test cases, large values for ρ performed better than smaller ones, with $\rho = 7$ ranging best among the investigated cases. Larger values may perform better, but their use is currently precluded by excessive computational expense.
- Computational expense also increases when β decreases. Note that amoeba shapes get more roundish and approach Euclidian patches for $\beta \to 0$, thus for smaller β amoebas contain more pixels.
- Optimal values of β and ρ show no obvious dependency from each other for $\beta \in \{0.10, \ldots, 0.43\}$ and $\rho \in \{3, \ldots, 9\}$.

Finally, we investigate how classification performs on a smaller number of classes. To this end, we train SVMs on subsets of the 16 classes. Due to the high number of possible combinations only a small selection will be displayed here. Table VI shows the results from choosing random subsets from the training set. Throughout these experiments, 8 training samples per class were used, and 32 samples per class were used for testing.

As expected, the training sets consisting of only two classes are discriminated easily, and the accuracy generally decreases as more classes are involved. For some combinations of two classes, the accuracy may even reach 100%.

However, not all sets containing a certain number of classes result in the same accuracy, as some classes are discriminated more easily from each other, while others are

TABLE VI: Classification accuracies for 30 random class subsets for 2, 4, and 8 different classes. Classes are named according to Figure 4. All tests used $\rho = 7$, $\beta = 0.29$, and $4-neighborhoods$

classes	accuracy	classes	accuracy
b l	1.0	c e j n	0.843
n o	1.0	a d i p	0.820
m p	1.0	a e h j k l o p	0.800
c e	1.0	a g j k	0.796
b c e l	0.992	a g j m	0.796
d p	0.968	b c d e i l m o	0.785
h n	0.937	j m	0.781
d l	0.937	a b d g h k n p	0.777
i m	0.921	a b c e f j l o	0.765
d e f o	0.875	c g h j l n o p	0.765
a o	0.875	a g m p	0.765
b c e g h i k o	0.867	b f g i j l m o	0.757
f j l p	0.859	a b d g i j k n	0.738
a h n o	0.851	e h j p	0.734
a b e f g k l o	0.847	a d g j l m n p	0.683

harder. The strategy used to create this table may also be used to measure how good individual pairs of classes can be distinguished. A thorough investigation would require to train $\binom{16}{2} = 120$ class pairs.

IX. CONCLUSION

In this paper we have demonstrated that texture descriptors derived from morphological amoebas can be used within a classic machine learning approach to texture classification, and achieve a reliable discrimination of textures. A small number of selected features based on graph indices and geometric properties of amoebas was combined in order to encode texture. Feature values were histogram encoded and fed into a SVM.

By additional experiment series, we have investigated the influence of important parameters of the feature computation on the classification performance. Finally the relation between the number of classes and the classification performance was tested. As expected, fewer classes can be distinguished more accurately.

This work represents a first step into the combination of amoeba-based texture descriptors with machine learning techniques. Future work will be needed to better assess the capabilities of this approach in comparison with other texture descriptors and other machine learning techniques. Alternative clustering methods as well as strategies for optimal parameter choice could be studied in more detail. Moreover, it will be interesting to investigate the applicability of the proposed technique for more advanced texture analysis tasks such as texture-based image segmentation.

REFERENCES

[1] D. Bonchev and N. Trinajstić, "Information theory, distance matrix, and molecular branching," *Journal of Chemical Physics*, vol. 67, no. 10, pp. 4517–4533, 1977.

[2] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proceedings of the 22nd British Machine Vision Conference*. BMVA Press, 2011, pp. 76.1–76.12, http://dx.doi.org/10.5244/C.25.76.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[5] M. Dehmer, "Information processing in complex networks: Graph entropy and information functionals," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 82–94, 2008.

[6] M. Dehmer, F. Emmert-Streib, and S. Tripathi, "Large-scale evaluation of molecular descriptors by means of clustering," *PLOS ONE*, vol. 8, no. 12, pp. 1–10, 12 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0083956

[7] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[9] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[10] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[11] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.

[12] C. Huang, L. Davis, and J. Townshend, "An assessment of support vector machines for land cover classification," *International Journal of remote sensing*, vol. 23, no. 4, pp. 725–749, 2002.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[14] G. G. Lendaris and G. L. Stanley, "Diffraction-pattern sampling for automatic pattern recognition," *Proceedings of the IEEE*, vol. 58, no. 2, pp. 198–216, Feb 1970.

[15] R. Lerallut, É. Decencière, and F. Meyer, "Image filtering using morphological amoebas," *Image and Vision Computing*, vol. 25, no. 4, pp. 395–404, 2007.

[16] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[17] O. Lezoray and L. Grady, Eds., *Image Processing and Analysis with Graphs: Theory and Practice*. Boca Raton: CRC Press, 2012.

[18] R. Picard, C. Graczyk, S. Mann, J. Wachman, L. Picard, and L. Campbell, "VisTex database," Online ressource, http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html, 1995, retrieved 2013-11-20.

[19] D. Plavšić, S. Nikolić, and N. Trinajstić, "On the Harary index for the characterization of chemical graphs," *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 235–250, 1993.

[20] P. Soille and J.-F. Rivest, "On the validity of fractal dimension measurements in image analysis," *Journal of Visual Communication and Image Representation*, vol. 7, no. 3, pp. 217–229, 1996.

[21] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[22] M. Welk, "Discrimination of image textures using graph indices," in *Quantitative Graph Theory: Mathematical Foundations and Applications*, M. Dehmer and F. Emmert-Streib, Eds. CRC Press, 2014, ch. 12, pp. 355–386.

[23] ——, "Amoeba techniques for shape and texture analysis," in *Perspectives in Shape Analysis*, M. Breuß, A. Bruckstein, P. Maragos, and S. Wuhrer, Eds. Cham: Springer, 2016, ch. 4, pp. 73–116.

[24] ——, "Graph entropies in texture segmentation of images," in *Mathematical Foundations and Applications of Graph Theory*, M. Dehmer, F. Emmert-Streib, Z. Chen, X. Li, and Y. Shi, Eds. Weinheim: Wiley-VCH, 2016, ch. 7, pp. 203–231.

[25] H. Wiener, "Structural determination of paraffin boiling points," *Journal of the American Chemical Society*, vol. 69, no. 1, pp. 17–20, 1947.