2014 22nd International Conference on Pattern Recognition

Texture Classification using 2D LSTM Networks

Wonmin Byeon*, Marcus Liwicki*, and Thomas M. Breuel†
{wonmin.byeon, marcus.liwicki}@dfki.de*, and tmb@cs.uni-kl.de†
University of Kaiserslautern† and DFKI*†, Kaiserslautern, Germany

***Abstract*—In this paper, we investigate the ability of the Long short term memory (LSTM) recurrent neural network architecture to perform texture classification on images. Existing approaches to texture classification rely on manually designed preprocessing steps or selected feature extractors. Since LSTM networks are able to bridge over long time lags, we propose applying them directly on the image, circumventing any hand-crafted prepocessing. We investigate different approaches with several input and output representations. In our experiments on a number of widely used texture benchmarking tasks (KTH-TIPS, OuTex, VisTexL, VisTexP, and NewbarkTex), we show that the performance is comparable to, or better than, existing state-of-the-art methods for texture classification.**

Texture is a rich source of information about the contents of images and identity of objects. However, reliable texture recognition has been challenging because texture is a property of image pixels that is both stochastic and non-local. Most approaches to texture recognition manually design feature extractors to cope with the non-locality, choosing specific ways of integrating information about a region that is robust to changes in phase. Examples of such an approach are Haralick's texture features [1].

Numerous approaches using texture feature extractors had been investigated in 70-80's including Haralick [2], Gabor filters [3], wavelets [4] and grey level co-occurrence matrices (GLCM) [2], [1]. The main drawbacks of these approaches are the need to select the proper size of filter bank or neighborhoods and that they are computationally expensive. These methods were also applicable only on grayscale images.

More recently, Drimbarean and Whelan [5], Mäenpää and Pietikäinen [6] and Iakovidis et al. [7] have incorporated color data into texture descriptors. Their works have been focused on the combination of color and texture either jointly or separately. Different texture descriptors under various color space were compared and different ways of combination were comparatively evaluated. For instance, Discrete Cosine Transform, Gabor filters, and co-occurrence matrices under separate or combined color channel. Their experiments have shown that joint color texture descriptors improve the performance. However, it is unclear what is the best way and method to describe a wide range of textures. It has so far been lacking a general and comprehensive framework to classify textures. It needs either static condition of texture or to be manually designed to obtain an optimal solution.

Only a few methods, which unify the system between the feature extraction and classification step (i.e. machine learning based methods), were proposed to overcome this problem mentioned above. First, in [8], multichannel filtering scheme is combined with the neural network. More recently, Convolutional neural network [9] and Random neural network [10] were used for texture classification. Among all neural networks based approaches, Convolutional neural network has been successfully applied for image processing or recognition tasks [11], [12], [13]. However, it also requires appropriate size of the kernel to recognize contextual patterns. Moreover, the performance is often dependent upon the quality and constraint conditioned data. Appropriate training data is also required accordingly.

In this paper, we investigate 2D Long short term memory (LSTM) recurrent neural network architecture to the problem of texture classification [14], [15]. 2D LSTM involves the recurrent connections which allow to access past and future context along all dimensions. More specifically, each forward (from left to right and from top to bottom) and backward (from right to left and from bottom to top) pass provide the surrounding context of its pixel in all directions. This property makes 2D LSTM suitable to apply for the image analysis tasks. 2D LSTM has so far applied to the problem of image segmentation [14] and offline handwriting recognition [16]. In the image segmentation task [14], each pixel was classified using 2D LSTM under limited conditioned images. Another application (offline handwriting recognition [16]) is the one of successful application using multi-dimensional LSTM network architecture. The task is combination of computer vision with sequence labeling task. To deal with the problem, 2D LSTM networks and connectionist temporal classification are combined. A main characteristic of this LSTM network is hierarchical structure by repeatedly composing 2D LSTM layers. The great strength of LSTM networks is its ability to learn salient features automatically from raw pixel data without any specific preprocessing.

The most important factor for robust and efficient training procedure of the neural network is the use of a large and generalized training data [11]. Especially for visual tasks, several ways of expanding training data were proposed in literature, e.g., a subset of patch training [17] or transformation invariance data generation [11]. For texture classification on general domain datasets, we introduce a number of different ways of applying LSTM networks. First, the input image is redesigned by multi-patches with the variation of textures. Compared to using a 2D image as a whole, multi-patch based input has flexibility to represent the pixels to the wide range of scaled and rotated textures which give remarkable discriminative power. It is also great merit of generating a large amount of training data and avoiding different parameter setup for different tasks since the size of input patch is fixed. The main purpose of this paper is to compare LSTM based texture classification works with other texture based image classification approaches. We successfully apply standard LSTM to raw RGB value of pixels directly, without feature extraction or preprocessing to the problem of texture classification.

The rest of the paper is organized as follows. Section I

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.206

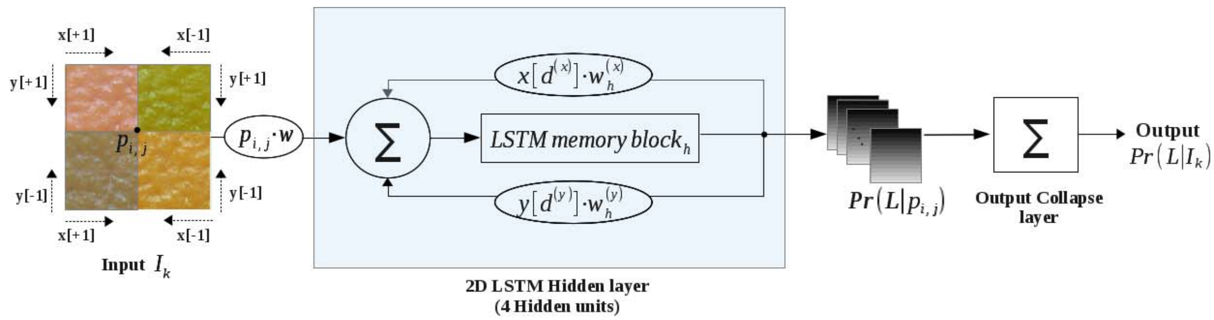

Fig. 1: 2D LSTM network architecture. First, raw RGB values at the pixel $p_{i,j}$ is sent to the network. 2D LSTM hidden layer includes four LSTM hidden units ($2^{(dim)}$ = 2^2 = 4 hidden units) with two recurrent connections (R={$(x[d^{(x)}]$, $y[d^{(y)}])$, $d^{(x)}, d^{(y)} \in [+1, -1]$}). The recurrent connections access to each dimension, and each hidden unit accumulates the information of each direction. Thus, it keep the all surrounding context and process it with the current pixel. Four regions on each side in the input I_k indicate scanned areas of the pixel $p_{i,j}$. The output of LSTM hidden layer for each pixel ($Pr(L|p_{i,j})$ for the pixel at $(i,\ j)$ with the corresponding label L) is fed to the output collapse layer, then outputs the class probabilities for each image ($Pr(L|I_k)$ for the kth image).

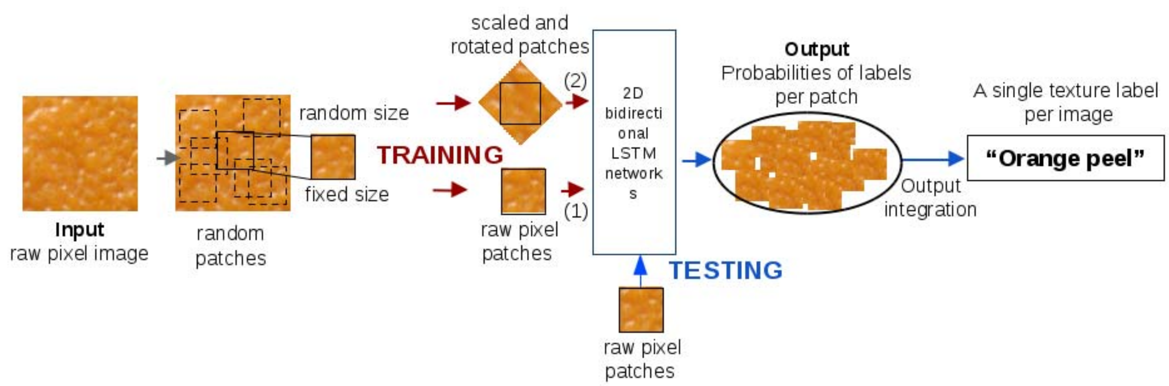

Fig. 2: Our approach: Performing texture classification with 2D LSTM networks. Input image is raw RGB value of pixels. At the training phase (red), the input includes two types of representation: (1) original pixels and (2) scale and orientation invariant representation. The whole procedure of the system is as follow: We first extract the randomly positioned patches with the (1) original pixels (2) randomly scaled and rotated pixels. It is then sent to the network. After the training step, the original patches are tested on the network (blue). The final output of the networks is a single texture label of each patch. The performance has been evaluated by both per-patch and per-image accuracy. The details of the procedure and evaluation criteria are explained in Section I and II. Better viewed in color.

and II describe our approach and evaluation criteria in detail and Section III presents the texture classification results. Finally, concluding remarks are given in Section IV.

I. System Description

A. LSTM recurrent neural network architecture for 2D data

LSTM neural networks are a combination of recurrent neural nets (RNN) [18], [19], [20] and LSTM architectures [21]. LSTM memory blocks in the hidden layer include self-connected memory cell(s) and three different gates, i.e. input, forget and output gates. The self-connected memory cell is functioned as a recurrent connection and controlled by forget gate. This architecture helps to store the information until it is not needed anymore. In the 2D case, each hidden unit includes two recurrent connections with two forget gates. As can be seen in Figure 1, these two recurrent connections access to each axis, $[d^{(x)}]$, $y[d^{(y)}]$ where $d^{(x)}$ and $d^{(y)}$ are the direction of the axis (-1 or $+1$). Thus, four hidden units ($2^{(dim)}$ = 2^2 = 4 hidden units) take care of the information in all directions (see the divided regions of input image for the pixel $p_{i,\,j}$ in Figure 1). Note that each hidden unit includes h LSTM memory block (h = hidden size). Our goal is to classify the complete image with a single output. The network first classifies each pixel independently, then the collapse layer sums up the output of each pixel and softmax function is applied for the final classification. The details of 2D LSTM network architecture are illustrated in Figure 1.

B. 2D LSTM networks for texture classification

To apply 2D LSTM networks for texture classification, the procedure is divided into four parts: input representation, input layer, 2D LSTM hidden layer, and output layer. The complete flow diagram of our approach is shown in Figure 2.

Input representation: The network receives inputs from raw RGB value of pixels. The input can be a 2D image as a whole or in multi-patches. The advantages of patch-wise input are 1) the consistent size of the input is retained, 2) it is transformable, i.e., variation of texture is easily generated, 3) The input is generated as much as is desired. Each dataset in our experiment has different challenging problems for different experimental designs. Multi-patch based input increases its generality. In addition, the datasets contain a huge range of resolution images. In order to obtain the identical model parameters, constant input dimension is required. For the reason, the size of patch pixels used as an input in all of our experiments is fixed to 64×64. Unlike popular texture analysis methods, our approach does not need any prior knowledge to extract features. However, the problem in general is too complex and the model prediction is tough, in order to generalize it, without specific preprocessing under the limited number of training samples and unconstrained conditioned data. To deal with it, scale and orientation invariant representation is introduced. The procedure is as follows (The illustration, see Figure 2). We first extract random sized patches at random position of the image. Each patch is then rotated at random angle and resized it to 64×64. It provides the scaled-up or down and rotated patches. The range of scale covers most of scales missed from original training samples. By training with the scale and rotation invariant input, the network model covers the wide range of scales (from the close-up texture materials to the natural scene textures) and all possible rotations. The original patched input is then tested with the trained model. The effectiveness of scale and rotation based input representation will be discussed in Section III.

For texture classification task, 2D array is sent directly into the network and produces an output that indicates a single texture class. For efficient computation of 2D LSTM, two key features, i.e., input subsampling and output collapse, are used at input and output layer of LSTM network.

Input layer: Input subsampling operation is not a regular process in standard LSTM. It is a part of the hierarchical structure of LSTM introduced in [16]. Received 2D input is divided into the windows and send it to the LSTM layer. Each window is presented as a single input, and each activation includes the local features. It is collected in the network and used as global features. All activations of whole input are then contributed to the final classification. This idea is similar to using windows to localize and obtain stable features from common local feature extractors. It is very effective, especially for 2D LSTM networks since 1) it localizes the contextual information of the image, and 2) it has computational speedup; It does not reduce an amount of data, but downscale the activation array. Evidence of its power will be provided in Section III.

LSTM hidden and output layer: The windows pass through 2D LSTM hidden layer including four hidden units which scan all surrounding neighborhoods of the current position. Output layer eventually receives final activations from LSTM hidden layer and all activations are then collapsed to the output array with the size of the class.

II. Output integration and evaluation

Output integration: LSTM networks perform probabilistic classification which provides the conditional probabilities of the labels given the input: $p\,(label \,|\, input)$. When multi-patch based approach is applied, output is the probabilities of classes for each patch. To determine the final class of an image, the further integration process is required. Since we have used random parts of an image, some patches have higher distinctive patterns and some may contain noise or clutters. For the reason, some smoothing effect is incorporated to find the most probable label of the image. The probabilities of the labels are first averaged, then maximized over these to find the highest score:

$$\arg\max_{label} \frac{1}{no.\,patch} \sum_{j=1}^{no.\,patch} p_j\,(label \,|\, input),$$

Evaluation: The performance had been evaluated with per-patch and per-image accuracy in order to compare the effectiveness of patch-based input representation. The classification accuracy for per-patch is computed as follow:

$$\text{accuracy}_{\text{per-patch}} = \frac{1}{T} \sum_{i=1}^{T} \begin{cases} 1 & \arg\max_l \, p_i\,(l \,|\, x) \\ 0 & \text{otherwise} \end{cases}$$

where x is input image patch, l is predicted texture label and T is total number of patches. Furthermore, per-image accuracy is measured using integrated score of all patches:

$$\text{accuracy}_{\text{per-image}} = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} 1 & \arg\max_l \, \frac{1}{n} \sum_{j=1}^{n} p_j\,(l \,|\, x) \\ 0 & \text{otherwise} \end{cases}$$

where n is the number of patches per image and N is the number of image.

III. Experiments

We evaluated our approach on five challenging texture datasets. Each dataset includes various types of textures, different size, and different number of training and testing samples. The neural network approaches in general require different type of experimental setup depending on the data condition. The advantage of patch-wise scale and rotation invariant input representation is that the parameter tuning is not necessary for each dataset when the optimal network model is found.

The detail of the datasets is explained in Table I.

A. Datasets

The dataset *KTH-TIPS* includes various conditions, that is, nine scales spanning two octaves, three different illumination directions, and three different poses. Some materials have very similar textures like cotton and linen or sponge and brown bread which makes the database challenging. For the comparison, we followed the evaluation setup proposed by Zhang et al. [27].

The dataset *OuTex* contains 68 classes of various color textures with 128×128 pixels. Half of the images were used in the training (680 images out of 1360 images) and remains for testing. Several categories of images have similar color and texture, so the discrimination only by its pixels is not easy.

The next datasets *VisTexL* and *visTexP* are both designed for natural color textures under non static conditions. The same scheme was used to generate the dataset. For VisTexL, 864 disjoint sub-images were generated from 54 texture images. VisTexP includes 55 texture classes with 880 sub-images. For both datasets, each image (size 512×512) is split up into 16 sub-images (size 128×128). These sub-images are considered

TABLE I: The summary of texture datasets used in our experiments. Each database has different challenging problems for different experimental designs. Textures in each dataset are under different or non controlled conditions with higher or lower resolution. KTH-TIPS includes specific texture of materials under varying illumination, pose, and scale. Other datasets are from the natural texture of the scene or the object. OuTex, VisTexL, and VisTexP contain a larger number of textures with a small number of training images. In contrast, NewbarkTex is composed of six tree bark classes with larger number of training images. However, the variation between the classes is not distinctive. All of the dataset can be found at the source link.

	Image size	# Texture	# Training images per class	# Test images	Type of texture	Source
KTH-TIPS	200×200	10	40	410	material	[22]
OuTex (OuTex-TC-00013)	128×128	68	10	680	natural texture	[23]
VisTexL (Contrib-TC-00006)	128×128	54	8	432	natural texture	[24]
VisTexP	128×128	55	8	440	natural texture	[25]
NewbarkTex	64×64	6	136	816	natural texture	[26]

as a same class. As for the Outex set, half of the images were used in the training phase.

Recently, a new benchmark colour texture image test suite, *NewbarkTex* from BarkTex dataset [28], [29], [30], [31] is proposed. Six tree bark classes with 68 images per class (128×128) are divided into 4 sub-images (size 64×64). Total 272 sub-images per classes (total 1,632 images) are built and it is again divided by half for training and testing.

B. Experimental setup

All the experiments have been run by using the RNNLIB library [32]. For the statistical evaluation [33], a preliminary test is repeated five times with different parameters to find the appropriate network architecture. The optimal parameters are then applied to the datasets with randomly divided training and testing samples. It is repeated over 50 times and reported the average accuracy. All five datasets have been examined directly on the raw RGB values of the pixels.

Input representation: As mentioned in Section I, a wide range of scale and rotation are considered as input. To rescale it, patches are randomly sampled between 50×50 and 80×80, then resize it to 64×64. The scaled patches are then rotated at angles of $0°$ - $360°$. Both scale and rotation are with 1 pixel or $1°$ level increment. Besides, the number of patches extracted in an image also affect the performance since randomly rotated and scaled patches increase the diversity. Very small and large number of patches (10 and 200) have been examined for all experiments to evaluate the influence of performance.

Input subsampling and LSTM networks: To find optimal network model with proper size of input and its corresponding window size, a preliminary test with the range of parameters (the hidden size = $\{15, 25, 50, 75, 100\}$, the window size = $\{$no-subsampling (one pixel), 5×5, 10×10, 15×15, 20×20, $25 \times 25\}$ with the input pixels = $\{64 \times 64, 100 \times 100, 200 \times 200\}$) has been carried out. If no input subsampling operation is used, each pixel is processed. At the preliminary test, we found that no proper training performance was actually achieved without or the small size of the window when the size of the input image is big (bigger than 2500 pixels; 50×50). The network is started to be trained when the size of the window is bigger than 5×5. When the network contains the small window size with the large hidden unit, it is also not converged (no subsampling or window size 5×5 when hidden size is bigger than 25). This preliminary experiment has shown the influence of input subsampling operation and relationship of input and hidden size with the window size. At the end, the size of window 5×5 with 15 hidden units was set with the input pixels 64×64 for all of our experiments. The learning rate and momentum have been fixed for all experiment to 1e-4 and 0.9 respectively.

C. Results and discussions

The best texture classification results using LSTM networks compared to other methods are summarized in Table II. We tested five datasets under three different input type: (1) an original 2D image, (2) multi-patches in an original 2D image, (3) multi-patches with scale and rotation invariant representation. With the input type (1), KTH-TIPS and NewbarkTex has already obtained the best accuracy among current feature extraction based approaches (99.48% and 78.2% respectively) and others are comparable (93.09% for OuTex, 89.55% for VisTexL and 90.0% for VisTexP). With multi-patch based input representation (input type (2) and (3)), it is clear that per-image accuracy are much scattered than per-patch. (The difference was about 3%). The number of patches per image have also an important role in classification performance. The large number of patches (200 in our experiment) per image outperform on most of the dataset (around 2% higher for all datasets except OuTex). The best results using LSTM networks compared with different feature extraction based methods are summarized in Table II. Overall, the best accuracy of our approach led to superior performance on most of benchmark datasets. Specifically, 200 patches per-image accuracy of KTH-TIPS dataset achieved 100% (1.5% higher) and NewbarkTex dataset achieved 78.2% (2.3% higher). The Statistical significance is lower for the dataset OuTex, VisTexL and VisTexP because of extremely small number of training samples with a large number of textures (only 10 images per class in OuTex (68 textures) and 8 images per class in VisTexL (54 textures) and VisTexP (55 textures). However, it still gives comparable performance. The results show that the multi-patch based scale and rotation invariant representation is very powerful to discriminate the raw pixel level images with 2D LSTM networks.

IV. Conclusion

Many texture classification methods proposed in the literature rely on manually designed preprocessing steps or feature extraction step. The main contribution of our work is to introduce a new approach to solving the problem of texture classification through LSTM recurrent neural network architecture. The benefit of 2D LSTM networks is an ability to make use of contextual information by itself, which is easily

TABLE II: Correct classification rates (avg. accuracy, %) on five benchmark datasets of texture classification (no. test=50). In order to compare the performance with other methods, all of our experiments have been following the same experimental setup. For each test, same training and test subset divided by provided test suites of OuTex, VisTexL, and NewbarkTex is used. For other datasets, it is randomly divided into the same number of images (Half of them for training and remains for testing). The three different representations of input were tested: 1) an original 2D image, 2) multi-patches in an original 2D image, 3) multi-patches with scale and rotation invariant representation. In addition, the different number of patches was also compared. Overall, scale and rotation invariant representation with 200 patches outperformed among others. The performance is compared to most recent or common methods of texture classification. Finally, the best accuracy of our approach leads to superior performance on most of benchmark datasets. Note that the values in bold denote statistical signicance at 95% confidence among other methods, and underlined numbers indicate comparable results.

Dataset The number of test samples	KTH-TIPS 610	OuTex 680	VisTexL 432	VisTexP 440	NewbarkTex 816
Basic Image Features based on steerable filters (BIF) [34]	98.50	-	-	-	-
Multiscale Local Binary Patterns (LBP) [35]	93.17	-	-	-	-
Principal Curvatures with four scales (PC) [36]	97.52	-	-	-	-
Rotation invariant multi-scale features (MLEP) [35]	96.41	-	-	-	-
Semi-joint Texton descriptor (STD) [37]	-	90.32	99.25	98.89	-
Homogeneous texture (HTD) + color structure (CSD) [38], [37]	-	86.71	<u>99.56</u>	98.53	-
Multispectral co-occurrence (MM) [39]	-	94.1	-	97.9	-
Haralick from reduced size chromatic co-occurrence (RSCCMs) [40]	-	-	-	-	75.9
LSTM networks (the proposed approach)	**100**	<u>94.70</u>	99.09	<u>99.07</u>	**78.2**

and directly applicable without feature extraction or manual preprocessing steps. Furthermore, the architecture is very simple; one hidden layer and a small number of hidden neurons are taken, unlike other complex neural network structures. We also investigated various ways of applying LSTM networks to the texture classification and achieved promising results on a number of widely used texture classification benchmarking datasets. Particularly, multi-patch based scale and orientation invariant input representation is very robust to extreme texture conditions and has an advantage of avoiding parameter tuning for different tasks. Future direction will be to use a variant of our approach to texture segmentation task. The success in texture classification shows the potential of a such application. We also aim at extending our approach to the real world scenes, since natural scenes in general include an amount of textures.

ACKNOWLEDGMENT

We would like to thank Federico Raue for his very valuable comments in revising the paper.

REFERENCES

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

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

[3] M. Petrou, P. G. Sevilla, and J. Wiley, *Image processing: dealing with texture*. Wiley Chichester, 2006, vol. 10.

[4] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on pure and applied mathematics*, vol. 41, no. 7, pp. 909–996, 1988.

[5] A. Drimbarean and P. F. Whelan, "Experiments in colour texture analysis," *Pattern Recognition Letters*, vol. 22, no. 10, pp. 1161–1167, 2001.

[6] T. Mäenpää and M. Pietikäinen, "Classification with color and texture: jointly or separately?" *Pattern Recognition*, vol. 37, no. 8, pp. 1629–1640, 2004.

[7] D. Iakovidis, D. Maroulis, and S. Karkanis, "A comparative study of color-texture image features," in *Proceedings of IEEE International Workshop on Systems, Signal and Image Processing, Halkida, Greece*, 2005, pp. 205–209.

[8] A. K. Jain and K. Karu, "Learning texture discrimination masks," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 2, pp. 195–205, 1996.

[9] F. H. C. Tivive and A. Bouzerdoum, "Texture classification using convolutional neural networks," in *TENCON 2006. 2006 IEEE Region 10 Conference*. IEEE, 2006, pp. 1–4.

[10] A. Teke and V. Atalay, "Texture classification and retrieval using the random neural network model," *Computational Management Science*, vol. 3, no. 3, pp. 193–205, 2006.

[11] P. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis." in *ICDAR*, vol. 3, 2003, pp. 958–962.

[12] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 609–616.

[13] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[14] A. Graves, S. Fernández, and J. Schmidhuber, "Multi-dimensional recurrent neural networks," in *Artificial Neural Networks–ICANN 2007*. Springer, 2007, pp. 549–558.

[15] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Munchen, Techn. Univ., Diss., 2008, 2008.

[16] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2008, pp. 545–552.

[17] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 376–389.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 1, p. 213, 2002.

[19] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[20] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural Networks*, vol. 1, no. 4, pp. 339–356, 1988.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] "KTH-TIPS texture image database," http://www.nada.kth.se/cvap/databases/kth-tips/download.html.

[23] "OuTex image test suite," http://www.outex.oulu.fi/index.php?page=classification.

[24] "VisTexL image test suite," http://www.outex.oulu.fi/index.php?page=contributed.

[25] "VisTexP texture image database," http://vismod.media.mit.edu/vismod/imagery/VisionTexture/distribution.html.

[26] "NewbarkTex image test suite," https://www-lisic.univ-littoral.fr/~porebski/BarkTex_image_test_suite.html.

[27] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007.

[28] C. Münzenmayer, H. Volk, C. Küblbeck, K. Spinnler, and T. Wittenberg, "Multispectral texture analysis using interplane sum-and difference-histograms," in *Pattern Recognition*. Springer, 2002, pp. 42–49.

[29] C. Palm, "Color texture classification by integrative co-occurrence matrices," *Pattern Recognition*, vol. 37, no. 5, pp. 965–976, 2004.

[30] C. Palm and T. M. Lehmann, "Classification of color textures by gabor filtering," *Machine Graphics and Vision*, vol. 11, no. 2/3, pp. 195–220, 2002.

[31] A. Porebski, N. Vandenbroucke, and L. Macaire, "Iterative feature selection for color texture classification," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 3. IEEE, 2007, pp. III–509.

[32] A. Graves, "Rnnlib: A recurrent neural network library for sequence learning problems," http://sourceforge.net/projects/rnnl/.

[33] A. Flexer, "Statistical evaluation of neural network experiments: Minimum requirements and current practice," *Cybernetics and Systems Research*, pp. 1005–1008, 1996.

[34] M. Crosier and L. D. Griffin, "Using basic image features for texture classification," *International Journal of Computer Vision*, vol. 88, no. 3, pp. 447–460, 2010.

[35] J. Zhang, J. Liang, and H. Zhao, "Local energy pattern for texture classification using self-adaptive quantization thresholds," 2013.

[36] J. Zhang, H. Zhao, and J. Liang, "Continuous rotation invariant local descriptors for texton dictionary-based texture classification," *Computer Vision and Image Understanding*, 2012.

[37] S. Alvarez and M. Vanrell, "Texton theory revisited: A bag-of-words approach to combine textons," *Pattern Recognition*, vol. 45, no. 12, pp. 4312–4325, 2012.

[38] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 703–715, 2001.

[39] V. Arvis, C. Debain, M. Berducat, and A. Benassi, "Generalization of the cooccurrence matrix for colour images: application to colour texture classification," *Image Analysis and Stereology*, vol. 23, no. 1, pp. 63–72, 2004.

[40] A. Porebski, N. Vandenbroucke, and L. Macaire, "Comparison of feature selection schemes for color texture classification," in *Image Processing Theory Tools and Applications (IPTA), 2010 2nd International Conference on*. IEEE, 2010, pp. 32–37.