A Hybrid Model for Invariant and Perceptual Texture Mapping

Huizhong Long and Wee Kheng Leow
Dept. of Computer Science, National University of Singapore
3 Science Drive 2, Singapore 117543
longhuiz, leowwk@comp.nus.edu.sg

Abstract

Texture is an important visual feature for computer vision tasks. In applications such as image retrieval and computer image understanding, texture similarity should be measured in a manner that is invariant to texture scale and orientation, as well as consistent with human's perception. However, most existing computational features and similarity measures are not perceptually consistent. A solution is to map textures into an invariant and perceptual space such that similarity measured in the space is perceptually consistent. This paper presents a hybrid method, using convolutional neural network and SVM, to perform the invariant and perceptual mapping. Test results show that it's overall performance is better than that of an individual neural network and a SVM.

1. Introduction

Texture is an important visual feature for computer vision tasks. In many applications such as image retrieval and computer image understanding, texture similarity should be measured in a manner that is invariant to texture scale and orientation, as well as consistent with human's perception. Although there exist scale- and orientation-invariant texture features and similarity measures [2, 4, 7, 13, 14], they are found to be not perceptually consistent [9, 10]. A solution to this problem is to perform an invariant and perceptual mapping of a texture into a canonical perceptual texture space, which is constructed using data obtained from psychological test of human's perception [9, 11]. The measurement of texture similarity in the perceptual space would be perceptually consistent and invariant to texture scale and orientation.

In general, the invariant and perceptual mapping problem can be stated as follows: Given the features $\mathbf{t}_i$ of texture i in the *input space* $\mathcal{T}$, map it to a point $\mathbf{I}(\mathbf{t}_i)$ in an *invariant space* $\mathcal{I}$ that is invariant to scale and orientation, followed by mapping it to a point $\mathbf{P}(\mathbf{I}(\mathbf{t}_i))$ in the *perceptual texture space* (PTS) $\mathcal{P}$. Note that it is not practically viable to perform this mapping in the reverse order $\mathbf{I}(\mathbf{P}(\mathbf{t}))$ because this would require the measurement of human's perception under various scales and orientations, which is too tedious to perform and too difficult to obtain reliable results. It is also not advisable to determine $\mathbf{I}$ and $\mathbf{P}$ separately because the optimal coordinates $\mathbf{I}(\mathbf{t})$ of the textures in the invariant space $\mathcal{I}$ are unknown and should be deduced by the optimization process.

Another difficulty with invariant and perceptual mapping is that it requires nonlinear multivariate *regression*. Since it is well known that regression is a much more difficult problem than classification [3], existing invariant texture *classification* methods such as [2, 4, 14] are likely not to perform well on feature mapping without major modifications.

In [8], we presented a neural network for invariant and perceptual mapping. Test results showed that texture image retrieval based on the mapped coordinates are more accurate and consistent with human's perception than that based on raw texture features. This paper further explores the method with a hybrid model that uses convolutional neural network to perform invariant mapping and Support Vector Machines (SVM) to perform perceptual mapping.

It is known that prior knowledge can be included in SVM to capture certain invariances. For example, in [12], a polynomial of vector dot product is used as the kernel function. However, polynomial kernels do not capture exactly the translation invariance needed for invariant mapping.

On the other hand, convolutional neural network is well-known for its success in applications such as character recognition [5, 6]. However, our tests show that SVM can perform the perceptual mapping more accurately than does a multilayer perceptron (Section 4). Therefore, the hybrid model is adopted to solve the invariant and perceptual mapping problem by combining the strengths of neural networks and SVMs.

2. Hybrid Model

The hybrid model consists of a three-layer convolutional network and n SVMs, one for each of the n dimensions of

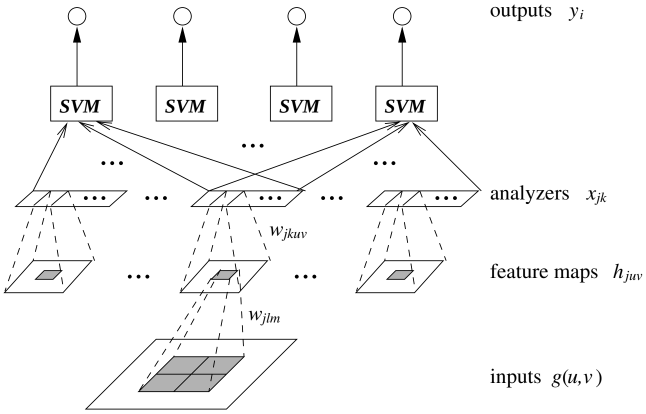

Figure 1. Architecture of the hybrid model. The convolutional neural network performs invariant mapping and the SVMs perform perceptual mapping.

a perceptual space (Fig. 1). The input layer contains Gabor texture features $g(u, v)$ arranged into a 2D map with spatial frequency increasing vertically and orientation changes horizontally. The second layer contains several feature maps. Each neuron h_{juv} in feature map j receives inputs from a local receptive field of the inputs $g(u, v)$:

$$h_{juv} = \sum_l \sum_m w_{jlm}\, g(u + l, v + m)\ . \tag{1}$$

The neurons in the same feature map j share the same weight structure and weight values w_{jlm}. The third layer contains a set of analyzers x_{jk} for each feature map j. Each analyzer computes the weighted sum input s_{jk} from the corresponding feature map j:

$$s_{jk} = \sum_u \sum_v w_{jkuv} h_{juv}\ . \tag{2}$$

Its output x_{jk} is a sigmoid function f of s_{jk}:

$$x_{jk} = f(s_{jk}) = \frac{1}{1 + \exp(-\beta s_{jk})} \tag{3}$$

where β is a constant parameter.

The vector $\mathbf{x}$ of all the analyzers' outputs x_{jk} are fed to the SVMs as their inputs. The SVMs are used to perform nonlinear regression between the inputs x_{jk} and the outputs y_i, which is given by

$$y_i = \sum_r (\alpha_{ir} - \alpha'_{ir}) K(\mathbf{x}_{ir}, \mathbf{x}) + b_i \tag{4}$$

where α_{ir} and α'_{ir} are the Lagrange multipliers, and $\mathbf{x}_{ir}$ and b_i are the support vectors and bias of SVM i. Gaussian functions are selected for the kernel functions $K(\mathbf{x}_{ir}, \mathbf{x})$ because they produce the best results for our data:

$$K(\mathbf{x}_{ir}, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_{ir} - \mathbf{x}\|^2}{\sigma^2}\right)\ . \tag{5}$$

In the current implementation, Gabor features extracted by a Gabor filter bank [7, 8] are used as the inputs. Thus, the input layer consists of a 6×8 map for 6 spatial frequencies and 8 orientations of the Gabor filters. Five feature maps are used to capture various characteristics of textures that manifest in the 2D Gabor input patterns (see [7] for details). Three of them have square (3×3 and 5×5) receptive fields (RFs) for capturing structured information. One map has horizontal (3×5) RF for capturing granularity, and the last map has vertical (5×3) RF for capturing directionality. Each feature map feeds to a set of 10 analyzers, which in turn provide inputs to 4 SVMs, one for each dimensions of the perceptual texture space (PTS) of [9]. The outputs of the 4 SVMs are the 4 coordinates of the PTS.

3. Iterative Optimization

The SVMs are trained using SVM training algorithm for nonlinear regression. The SVM training algorithm minimizes the following error function:

$$E'_i = \sum_p L_\epsilon(y^*_{pi}, y_{pi}), \quad \text{for each } i = 1, \ldots, 4, \tag{6}$$

where y^*_{pi} is the desired value, y_{pi} is the actual output produced by SVM i, and p is the input pattern number. The function L_ϵ is called the ϵ-insensitive loss function:

$$L_\epsilon(y^*, y) = \begin{cases} \|y^* - y\| - \epsilon & \text{if } \|y^* - y\| \geq \epsilon \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The neural network is trained with error back-propagation (BP) through the SVMs. BP minimizes the summed squared error E_T:

$$E_T = \frac{1}{2} \sum_p \sum_i (y^*_{pi} - y_{pi})^2\ . \tag{8}$$

Although E_T is not strictly equal to $\sum_i E'_i$, they are monotonically related to each other. In other words, minimizing E_T does not increase E'_i, and vice versa. Therefore, SVM training is compatible with BP even though they minimize different error functions.

Now, we derive the weight change rules for the neural network. For notational simplicity, let's consider the case of a single input pattern (thus dropping the subscript p). Then, the error E is given by

$$E = \frac{1}{2} \sum_i (y^*_i - y_i)^2 \tag{9}$$

The weight update rule for the weights w_{jkuv} of the connections between the feature maps and the analyzers can be derived as follows:

$$\Delta w_{jkuv} = -\eta \frac{\partial E}{\partial w_{jkuv}} = \eta \delta_{jk} h_{juv} \qquad (10)$$

where η is the learning rate and δ_{jk} is the local gradient:

$$\delta_{jk} = \frac{1}{\sigma^2} \sum_i (y_i^* - y_i) \sum_r (\alpha_{ir} - \alpha'_{ir}) K(\mathbf{x}_{ir}, \mathbf{x})(x_{irjk} - x_{jk}) f'(s_{jk})$$

In a similar manner, the weight update rule for the weights w_{jkl} of the connections between the input layer and the feature maps can be derived as follows:

$$\Delta w_{jlm} = -\eta \frac{\partial E}{\partial w_{jlm}} = \eta \sum_k \delta_{jk} \sum_u \sum_v w_{jkuv} g(u+l, v+m)$$

For batch update,

$$\begin{aligned} \Delta w_{jkuv} &= \eta \sum_p \delta_{pjk} h_{pjuv} \\ \Delta w_{jlm} &= \eta \sum_p \sum_k \delta_{pjk} \sum_u \sum_v w_{jkuv} g_p(u+l, v+m) \end{aligned}$$

The overall training algorithm is summarized as follows:

Initialize neural network weights.
Run training patterns through network and collect SVM inputs for each pattern.
Train each SVM individually.
Repeat
  Update connection weights of neural network.
  Run training patterns through network and collect SVM inputs for each pattern.
  Train each SVM individually.

The training of the SVMs and that of the neural network iterate to optimize the entire hybrid model.

4. Testing Results and Discussion

To fully assess the network's mapping capability, five different tests were performed:

1. I_c: new instance; canonical scale, orientation
   9 non-overlapping samples were cropped from each of 60 distinct Brodatz texture images [1]. For each texture, 7 samples were used for training, 1 for validation, and 1 for testing. 9-fold cross-validation tests were conducted.

2. T_c: new type; canonical scale, orientation
   All samples of 48 (80%) of the textures were used for training. All samples of 6 (10%) of the textures are used for validation, and the rest for testing. 10-fold cross-validation tests were conducted.

3. I_v: new instance; various scales, orientations
   56 variations, with 7 scales and 8 orientations, were produced from each of the 60 textures. 80% of these variations were randomly selected for training, 10% for validation, and 10% for testing. 10-fold cross-validation tests were performed.

4. T_v: new type; various scales, orientations
   Similar to condition 2 except that samples with various scales and orientations were used.

5. R: random
   Of all the scale- and orientation-variations of the 60 textures, 80% were randomly selected for training, 10% for validation, and 10% for testing. 10-fold cross-validation tests were performed.

Note that conditions 2 and 4 are the most stringent conditions because the testing samples are perceptually different from the training samples. Their testing errors are expected to be larger than those of the other conditions. The last condition mimics a practical situation in which random samples of the textures at random scales and orientations are available for training.

As the baseline results, SVM regression and convolutional neural network were individually applied to the five conditions as well. The convolutional neural network was similar to the hybrid model except that the SVMs were replaced by a two-layer feedforward network. Comparison with pure SVM model would verify the effectiveness of the invariant mapping. On the other hand, comparison with the pure invariant network would verify the overall performance of the hybrid system.

For comparison with the mapping accuracy of Gabor features, other texture features such as Tamura texture features, MRSAR, and the invariant features of [7] were also mapped to PTS using SVM regression. All these texture features have been used in various image retrieval systems [9]. For these features, only the test cases with canonical scale and orientation (I_c, T_c) were conducted because it is very difficult to perform invariant mapping of these features.

Table 1 summarizes the testing results. For the cases of canonical scale and orientation (I_c, T_c), SVM can map Gabor features more accurately than other texture features to the perceptual texture space. The hybrid system performs better than pure neural network but marginally poorer than SVM. This results are expected since pure SVM regression takes the original Gabor features as the inputs. On the other hand, the SVMs of the hybrid system take the outputs of the

Table 1. Mean squared errors of various texture mapping tests. First three row tabulates the mapping of three types of texture features by SVM. The last three row shows the mapping of Gabor features by three different methods.

	I_c	T_c	I_v	T_v	R
Tamura	0.144	0.256	—	—	—
MRSAR	0.096	0.244	—	—	—
invariant	0.124	0.268	—	—	—
SVM	0.0052	0.216	0.244	0.245	0.240
NN	0.0099	0.238	0.0074	0.148	0.016
NN+SVM	0.0065	0.226	0.0061	0.143	0.012

convolutional network as the inputs, and inevitably, some information is loss by network processing.

For the cases of variable scale and orientation (I_v, T_v, R), the hybrid system performs much better than pure SVM because the hybrid system performs invariant mapping whereas pure SVM does not. It's performance is also better than that of pure neural network. As a whole, the integration of the convolutional neural network and SVM indeed produces better overall mapping accuracy than individual neural network and individual SVM.

5. Conclusion

This paper presented a hybrid model that can be trained to optimize both invariant and perceptual mappings. The lower portion of the model contains a convolutional neural network that maps Gabor texture features to an invariant space. The upper portion of the system contains four SVMs each mapping the texture from the invariant space to one of the four dimensions of the perceptual texture space. Extensive tests were conducted to assess the performance of the hybrid system. Test results show that the hybrid model can combine the strengths of both convolutional neural network and SVM to perform invariant and perceptual mappings accurately. We expect this model to be applicable to the invariant and perceptual mapping of other visual features.

Acknowledgments

This research was supported by NUS ARF R-252-000-049-112.

References

[1] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, 1966.

[2] F. S. Cohen, Z. Fan, and M. A. Patel. Classification of rotated and scaled textured image using Gaussian Markov random field model. *IEEE Trans. PAMI*, 13(2):192–202, 1991.

[3] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[4] G. M. Haley and B. S. Manjunath. Rotation-invariant texture classification using a complete space-frequency model. *IEEE Trans. IP*, 8(2):255–269, 1999.

[5] J. Keeler, D. Rumelhart, and W. Leow. Integrated segmentation and recognition of hand-printed numeral. In *Proc. NIPS*, pages 557–563, 1991.

[6] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 255–258. MIT Press, 1995.

[7] W. K. Leow and S. Y. Lai. Scale and orientation-invariant texture matching for image retrieval. In M. K. Pietikäinen, editor, *Texture Analysis in Machine Vision*. World Scientific, 2000.

[8] H.-Z. Long and W. K. Leow. Invariant and perceptually consistent texture mapping for content-based image retrieval. In *Proc. ICIP*, 2001.

[9] H.-Z. Long and W. K. Leow. Perceptual texture space improves perceptual consistency of computational features. In *Proc. IJCAI*, 2001.

[10] J. S. Payne, L. Hepplewhite, and T. J. Stonham. Perceptually based metrics for the evaluation of textural image retrieval methods. In *Proc. IEEE ICMCS '99*, pages 793–797, 1999.

[11] A. R. Rao and G. L. Lohse. Towards a texture naming system: Identifying relevant dimensions of texture. In *Proc. IEEE Conf. Visualization*, pages 220–227, 1993.

[12] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik. Prior knowledge in support vector kernels. In *Proc. NIPS 10*, pages 640–646, 1998.

[13] D.-G. Sim, H.-K. Kim, and D.-I. Oh. Translation, scale, and rotation invariant texture descriptor for texture-based image retrieval. In *Proc. ICIP*, 2000.

[14] A. Teuner, O. Pichler, J. E. S. Conde, and B. J. Hosticka. Orientation- and scale-invariant recognition of textures in multi-object scenes. In *Proc. ICIP*, pages 174–177, 1997.