Knowledge Based Image Enhancement Using Neural Networks

Claudia Nieuwenhuis
Technical University of Ilmenau
Department of Neuroinformatics
98693 Ilmenau, Germany
claudia@nieuwenhuis.de

Michelle Yan
Siemens Corporate Research
755 College Road East
Princeton, NJ, 08540, USA
michelle.yan@siemens.com

Abstract

In this paper we combine the concept of adaptive filters with neural networks in order to be able to include high level knowledge about the contents of the image in the filtering process. Adaptive image enhancement algorithms often utilize low level knowledge like gradient information to guide filtering parameters. The advantage is that these filters do not need any specific knowledge and can thus be applied to a broad spectrum of images. However, for many problems this low level information is not sufficient to achieve good results. For example in medical imaging it is often very important that some features are preserved while others are suppressed. Usually these features cannot be distinguished by low level information. Therefore we propose a method to incorporate high level knowledge in the filtering process in order to adjust the parameters of any given filter thus creating a guided filter. We present a scheme for acquiring this high level knowledge which allows us to apply our method to all kinds of images using pattern recognition and special preprocessing techniques. The design of the guided filter itself is easy as for the high level knowledge only some sample pixels including their neighborhood and the desired parameters for these pixels are necessary.

1. Introduction

Objectives in image enhancement are noise reduction, feature enhancement, the removal of inhomogeneous background *etc*. Most proposed adaptive filters have relied only on low level information like gradient information or edge direction. These features are usually not sufficient to distinguish between structures that need to be preserved and others that should be filtered. In general, an adaptive filter uses information derived from the region around each pixel to adapt its parameters to the contents of the image. But the contents is only measured in terms of low level concepts such as the gradient. Thus the results of the filter could be improved to a great extent by deriving high level information from the surroundings of each pixel in order to adapt the filter parameters. In our approach this is done by generating a continuous parameter map that assigns specifically adjusted filter parameters to each pixel. The high level knowledge required for each pixel can be derived from some sample pixels associated with their respective desired parameters. To obtain the prior knowledge we use a neural network, for example a multilayer perceptron (MLP) with the backpropagation algorithm. The crucial part of designing a MLP is the preprocessing of the data. We do this in a standardized way that can be applied to many different kinds of images. An advantage of our preprocessing technique is that it reduces the dimensionality of the network to a large extent, when compared with simply using the neighborhood of each pixel as input. Its performance is improved at the same time. To demonstrate the concepts, digital color retinal images are used as an example to better explain the application of our method.

Concerning image enhancement a lot of related work exists. Adaptive filters in general make use of the local image structure, see for example [3], [6], [4]. We combine this concept with neural networks to achieve better performance in image enhancement. The basic concept has been realized in [5] where so-called neurofilters are used to achieve the results of standard filters without having to use mathematical models. However, there are two main differences to our approach: 1) the neurofilters mainly aim at realizing results that can also be achieved with standard filters, e.g. detecting edges or removing noise. In contrast to that we create new filters by adapting the filter to the semantic contents of the image. 2) We also propose a preprocessing framework to improve the results of the neurofilter. The idea of using higher level knowledge to interpret images has been mentioned in [2] where a preliminary segmentation is calculated to improve object detection results. Here we do not segment the image but compute a continuous filter parameter map.

The rest of the paper is organized as follows: Section 2 describes the selection and preprocessing of the training data in order to obtain a continuous parameter map which is used for the definition of guided filters. In section 3 we show results for different kinds of filters applied to different images and our conclusion in section 4.

2. Image enhancement with guided filters

In this section we will first describe how to obtain a filter parameter map by training a neural network. This map includes structure information about the contents of the image that is not contained in low level features like the gradient and thus cannot be used in common filters. Then we will show how this map can be used to define guided filters. When explaining the general approach we will make our concept clear by using retinal images as an application example. In retinal images like in most medical images, important information is usually included in the foreground that must not be changed by the image enhancement algorithm in order to preserve any knowledge needed for a diagnosis. The ideal result of such an image enhancement algorithm would thus leave the foreground unchanged while removing or adjusting the background. In this case the "backgroundness" of the pixel is essential to determine the filtering parameters.

2.1. Training data selection

The first step when training a neural network is the selection of training data. The neighborhoods of some representative pixels are stored in a vector and saved as one training sample which is preprocessed and associated with the desired filter parameter set for this particular pixel as teacher for the network.

Choosing sample data positions: In general it is necessary to choose sample data from every feature that can appear in the image so the network learns to map specifically adapted parameters to these features. Usually an image consists of many different features we call feature classes. In retinal images we have for example small and large vessels, vessel connections, microaneurysms and the background. The problem of selecting the training data is that we want to select the data randomly from several sample images while ensuring at the same time that all feature classes are well represented in the training data set. If we simply chose the data randomly the distribution of the training data would represent the distribution of the feature classes in the image instead of being adapted to the special needs of the learning process for each class. Thus the learning ability of the network would be poor due to the fact that certain rare features, in our case small vessels, would be under represented. To obtain an adaptive distribution of the training data we choose three training images containing together all relevant features and create binary masks for each feature class. This method enables us to select the positions of the sample data randomly but according to a predefined distribution which improves the flexibility of the training process as well as the network's ability to learn.

Creating training data: In order to adapt the parameters of the filter to the contents of the image in the surroundings of each pixel, we define a rectangular region of interest (ROI) around the pixel. The size of the rectangle should be as small as possible to make the network sensitive to small structures (for example small vessels on noisy background), but large enough to contain significant information about the contents of the region.

2.2. Training data preprocessing

By applying two preprocessing techniques to the training data we improve the performance of the network and minimize its size, but we do not use any prior knowledge about the image itself. Therefore these techniques can be applied to many kinds of images.

Rotation: Features that are to be classified by a neural network obey a certain pattern, like for example the line shaped vessels in a retinal image. Often these patterns can appear in different orientations like the vessels which point in arbitrary directions. The problem is that the network has to learn all these orientations even though these patterns could be recognized much more easily if they were all aligned in the same way. Furthermore it does not matter if the ROI is always rotated correctly to align all the different features in the same way - the only thing that matters for classification is that similar patterns are rotated in similar ways so their resemblance is preserved. For this reason the rotation of the ROI is surely helpful for any task where at least some of the different patterns show orientations that are not necessary for classification. The problem is now that we do not know if any feature exists within the ROI at all and which angle has to be used for rotation. However we still can rotate the ROI in the following way. First we define a threshold ensuring that at least one dominant characteristic line of the feature stands out after thresholding. For retinal images the threshold can be set to the mean value of all pixels in the ROI as vessels are among the darkest objects in the image. We then determine the main direction of the remaining pixels after thresholding. Now the ROI is rotated around the smaller angle between this direction and the horizontal axis. If the ROI contains only part of a feature or no feature at all the result is rather arbitrary but still similar for similar features thus ensuring similar classification. However, if there is a feature in the ROI it will afterwards be horizontal. An example for vessels can be seen in figure 1.

Multi-class Principal Component Analysis: Principal component analysis (PCA) is a good preprocessing method to reduce the size of the network and the noise of the sample

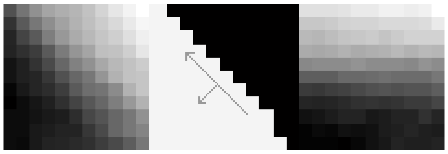

Figure 1. Rotation of network input to simplify learning process; left: original vessel feature; center: binarized feature before rotation around major axis; right: rotated feature

Figure 2. left: original green channel of a retinal image; right: parameter map derived from network and specifically adapted to features such as vessel and microaneurysms.

data provided two requirements are fulfilled: the dependencies between the data samples are linear and the features used to distinguish between the classes do not lie on axes with low variance. In practice we can assume linear dependencies but since there might be classes providing low variance compared to other classes the information necessary to discriminate between them could be lost due to the dimensionality reduction by PCA. For example this could be the case for the small vessel and the background feature classes. But due to the fact that PCA provides the above mentioned advantages we would still like to use it on the training data. This can be done in the following way: Instead of applying PCA to all samples together we apply PCA to each of the feature classes separately thus preserving the most important information of each class in its k eigenvectors. The number of eigenvectors chosen for each class is calculated by sorting the eigenvectors of the class by their eigenvalues e_i in descending order and determining the number of vectors as the maximum n satisfying the condition $\sum_{i=1}^{n} |e_i| \leq \epsilon \sum_{i=1}^{k} |e_i|$ where $0 < \epsilon \leq 1$ denotes the amount of information preserved of each class. The resulting principal components of each class are then combined creating one large set which represents the most important properties of all classes. However, there may be an axis exhibiting low variance for all feature classes but high variance between samples belonging to different classes. In order to preserve this information for classification we add the most important principal components derived from PCA carried out on all data samples together to our set of principal components. Now the dimension of the dataset can be reduced by projecting each sample vector containing the rotated ROI onto each of the principal components, that means into all resulting eigenspaces. The vector of the coordinates in the eigenspaces is finally used as input to the network.

2.3. Filters guided by parameter maps

After selecting and preprocessing the training data we can now use a neural network to learn the parameter map. An example of a parameter map describing the background of each pixel in a retinal image can be seen in figure 2. Given the parameter map obtained from the application of the network, there are numerous possibilities to adapt the parameters of a given filter, for example the sigma of a simple Gaussian filter, the parameters of an anisotropic filter, the degree of correction of inhomogeneous lighting (called shading correction) or of contrast enhancement.

3. Results

We use the green channels of 3 color retinal images with a size of 1024x1280 to test the guided filter performance. Comparing the network answer for each pixel with the manually labeled ground-truth the performance of the network is measured in terms of sensitivity and specificity (where sensitivity means the ratio of as foreground classified pixels to all foreground pixels and specificity means the ratio of as background classified pixels to all background pixels). To demonstrate the improvement of different filters by incorporating high level knowledge, we choose 4 different applications: Gaussian filtering, anisotropic filtering, contrast enhancement and shading correction.

Network and preprocessing results: The network we applied to a series of retinal images is a MLP with one hidden layer trained with the backpropagation algorithm using a momentum term. It consists of 47 neurons in the input layer due to the number of principal components chosen in the preprocessing step. 5 neurons have been found a good choice for the hidden layer. The output layer depends on the number of filter parameters. The learning rate and the momentum term decreased over time and the block size was set to 11.

Application of the network without preprocessing - that means without rotation and multi-class PCA - resulted in 89.6% (with 20 hidden neurons) and 86.3% (with 5 hidden neurons) correctly classified pixels. By adding our preprocessing methods we achieved 91.8%. Rotation by itself improved the result by 0.9% whereas combined multiclass PCA improved the results of normal PCA by 1.3%. This

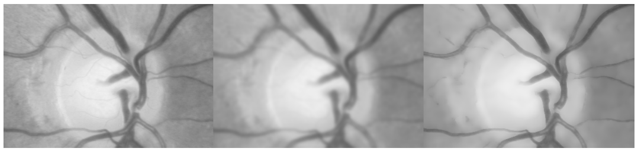

Figure 3. left: original cropped retinal image; center: Gaussian filter result; right: result of the same filter guided by a parameter map

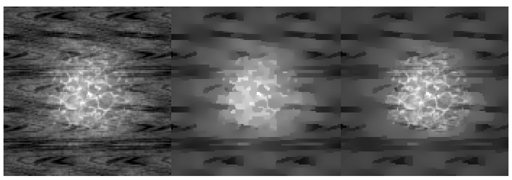

Figure 4. left: original artificial image containing two different patterns; center: anisotropic filter result; right: result of the same filter guided by a parameter map

demonstrates that our preprocessing techniques improve the performance and reduce the dimensionality of the network from (121,20,2) to (47,5,2). When comparing the binarized background map with the hand labeled ground-truth images we achieved 90.3% sensitivity and 97.1% specificity.

Guided filtering on different images: A simple way to design an adaptive Gaussian filter is to adjust the sigma of the filter to the parameter map returned by the network. Figure 3 shows that in this way features are preserved while the background is smoothed. Figure 4 shows the results of a guided anisotropic filter, where the image containing two different structures (wood and light refractions) is first filtered using an anisotropic filter which effects both structures. To preserve the light refractions structure we train a network which assigns each pixel a value for the resemblance to the light refraction structure. This information can be used to adapt the filter parameters. Our method can also be used to improve the contrast in an image without incre-

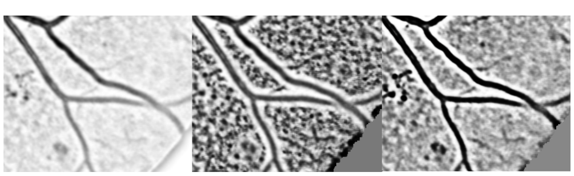

Figure 5. left: original cropped retinal image; center: adaptive contrast enhancement result; right: result guided by a parameter map

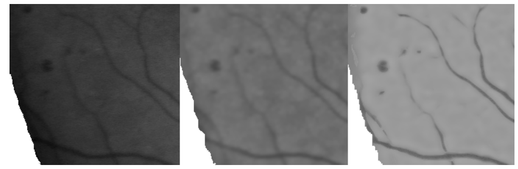

Figure 6. left: original cropped retinal image; center: shading correction result, right: result of the filter guided by a parameter map

menting the noise level as is a common problem with local adaptive filters. Here the parameter map is used to adjust the sigma in the contrast enhancement filter described in [1]. Guiding an adaptive contrast enhancement filter yields the results in figure 5. Another way to make use of the prior knowledge obtained from the neural network result is to design a filter for shading correction. The parameter map contains the number of control points for a spline which approximates the background and is then subtracted from the image. Figure 6 demonstrates that in this way the correction can be made stronger without losing details.

4. Conclusion

We proposed a new adaptive filtering concept providing a framework to create prior high level knowledge of images and use it to locally adjust the filtering parameters. Furthermore we showed that the binarized rotation of the sample data and the application of multi-class PCA improves the performance of the network. The results demonstrate a clear advantage of our method compared to the use of non-adaptive and adaptive filters that cannot make use of high level knowledge about the image contents. This framework can easily be adapted for many kinds of images due to the very general approach.

References

[1] E. Grisan. *Automatic Analysis of Retinal Images*. PhD thesis, City University London, 2005.

[2] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. *IEEE International Conference on Computer Vision*, 2005.

[3] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion, 1990.

[4] P. Saint-Marc, J. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1991.

[5] J. Varona and J. Villanueva. Neural networks as spatial filters for image processing: Neurofilters. *SPIE - The International Society for Optical Engineering*, 1997.

[6] J. Weickert. *Anisotropic Diffusion in Image Processing*. PhD thesis, University of Kaiserslautern, 1996.