Neural-Fuzzy Feature Detectors

Harvey A. Cohen and Craig McKinnon

School of Computer Science
and Computer Engineering
La Trobe University, Melbourne,
Victoria 3083 Australia
E-mail: H. Cohen @latrobe.edu.au

J. You

School of Computer Science
and Information Technology
Griffith University Brisbane
Queensland 4111
Australia

Abstract

Image features, such as edges, corners, and interesting points, are pixels identified by virtue of relations satisfied by the pixels within an image window. The classical edge operators, and the more elaborate operators for interesting points such as the Plessey and Moravec operators, have clear limitations. We introduce a new approach to the development of neural-fuzzy counterparts of such operators by using a training set comprising a set of pixels within a realistic image, these pixels being crisply scored by use of a classic operator. Our approach is directly compared with training NNs to fuzzy outputs, as proposed by Bezdek. Our method, which leads to relatively fast training, has the notable feature of being extensible over large windows.

Keywords: *neural net, feature points, Sobel, training, crisp, defuzzification, extensible.*

1. Introduction

A very important role is played in image analysis by what are termed feature points, pixels that are identified as having a special property. Feature points include edge pixels as determined by the well-know classic edge detectors of PreWitt, Sobel, Marr, and Canny [1]. Recently there has been much revived interest [2][3] in feature points determined by "corner" operators such as the Plessey, and interesting point operators such as that introduced by Moravec. [4][5] Classical operators identify a pixel as a particular class of feature point by carrying out some series of operations within a window centred on the pixel under scrutiny. The classic operators work well in circumstances where the area of the image under study is of high contrast. In fact, classic operators work very well within regions of an image that can be simply converted into a binary image by simple thresholding. To be definite as to the failings of classic operators: classic edge detector tends to give poor results for labelling edge pixels, when an edge, although definite, represents only a smallish gray-scale jump. Yet often such edges are clearly visible to the human eye. In summary, feature points are characterised by their relationship to pixels values within some local window. In a range of cases classic operators give excellent values. But for many circumstances the classic operators perform poorly. The question arises: would a neural net trained to mimic a classic operator, over a range of situations perform as well as, or maybe outperform, the original classic operator. Our answer, is that if the NN is trained over a paradigm set where the classic operator performs very well, and the training augmented with typical examples of the application of the classic operator, then the NN in fact greatly outperforms the classic operator.

1.1 The Sobel Edge Detector

In this paper, we draw especial attention to NN counterparts of the classic Sobel edge detector.

The classic Sobel Edge detector utilises the two-

$$D_X = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0* & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad D_Y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0* & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

smoothed gradient operators:

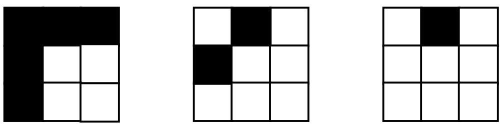
in conjunction with a threshold T, so that an edge pixel is one for which

$$E = (1/6)\ |\ D_X(i,j)\ |\ +\ |\ D_Y(i,j)\ |\ >\ T$$

Here we have assumed normalised pixel values lie in range 0 .. 1.0, to fix an appropriate value for the normalising quantity. Typically T=0.5. For later reference, we call E "edgednesss". Standard texts give examples where the classic Sobel performs well.

In Fig 1, we give an example of abject failure of Sobel, applied to the Kosh image. Visually, the edges of the original Kosh image (at http://www.cs.latrobe.edu/~image/edge.html) are quite apparent.

Applied to a binary image, for pixel values 0 and 1, the Sobel edgedness has values 0, 1/3, ½, 2/3 and 1:



E=1 **E = 0.67** **E = 0.33**

Figure 1: 3x3 Sobel Edgedness for binary images. Note that the state of the centre pixel does not affect the value of E

Figure 2: 316x500x24-bit colour Kosh: Output of Sobel edge detector applied on luminance (threshold T = 0.11). The original image has well defined edges involving small changes in grayscale. (See Fig 3)

1.2 Neural net edge detectors

A number of researchers have addressed the question of training neural nets with reference to classic edge detectors. Weller [6] trained a neural net by reference to a small training set, so that a Sobel operator was simulated. Bezdek [7,8] has recently presented a scheme for producing a neural net edge detector, based on the Sobel. In Bezdek's approach, detailed in Section 2 of this paper, a neural net is trained on all possible exemplars based on binary images, with each windowed possibility being scored by the (normalised) Sobel operator. Bezdek [7,8] describes his algorithm as fuzzy, as the normal output of the Sobel operator, which is here called Sobel edgedness, is interpreted as a fuzzy membership function. In fine Bezdek's approach involves training a neural net on crisp (binary) windows to give the corresponding uncrisp membership function for that window.

2. Neural Net Feature Detectors

Feature Detectors assign a numeric label to any pixel in a colour or gray-scale image, the label specifying a feature characteristic such as edge-like, corner-like, "interesting". We are concerned with developing neural net based detectors that perform a like function to the well-know classic edge detectors, including PreWitt, Sobel, Marr, [1] corner operators such as the Plessey, and interesting point operators such as Moravec. [2][3] Such NN feature detectors has as inputs the pixel values from locations within a window, not necessarily rectangular, about an image pixel.

The simplest NN edge detector was that proposed by Weller [6] who intuitively scored a mere 20 examples of edge-situations in a 3x3 window, these 20 examples serving as the total training set for a neural network. The approach of Weller ignores altogether the capabilities of the classic operators.

Bezdek and co-workers presented a very significant method over several papers. [5][6] The key feature of the Bezdek approach is the use of a training set based on a square window in a binary image. In these examples, only a 3x3 window was considered, and Bezdek used the Sobel operator to "score" all possible binary pixel populations. For a 3x3 window, there are 2^9 different possible window sets, and these binary windows were used by Bezdek in training a neural net. In fact, for the Sobel operator, the window centre value is not used, and from symmetry, the size of population can be further reduced. Scoring was done in a scaling way, so that Sobel outputs ranged from 0 to 1. The approach is fuzzy to the extent that such values are interpreted as fuzzy membership functions, expressing the extent to which the pixel is considered to belong to the edge pixel fuzzy class, or other feature fuzzy class.

2.2 Bezdek's NN counterpart of Sobel

Applied to a binary image, the Sobel operator gives any pixel an edgedness E of value 0, 1/3, 2/3, or 1. Bezdek proposed a neural net based edge detector that is trained on the set of all possible 3x3 windows in a binary image. In Bezdek's scheme, the edgedness, E, is considered as a fuzzy measured of membership of the set of edge points, and the neural net, with one or two hidden layers, is trained to give the appropriate value of E for each window.

In detail, there are just 256 distinct 3x3 windows, as the Sobel does not depend on the value of the centre pixel. Following Bezdek's scheme,

Applied to a grayscale image, with pixel values scaled so as to range from 0 to 1.0, the output of the trained detector ranges from 0.0 to 1.0 at any pixel. A process of defuzzification, equivalent to the choice of a threshold for the classical Sobel, is then applied to the (single) output of the NN.

Figure 3: 316x500x24-bit colour Kosh: Output of NN edge detector trained on Sobel edgedness thresholded at T =0.11. NN input uses luminance value for each colour pixel. (The image is of an alien in an encounter suit)

2.3 Deficiencies of Bezdek's NN Sobel

What is striking is that Bezdek's approach does in fact succeed in producing an excellent edge detector, that agrees with the Sobel on binary images, but has better behaviour on low contrasty grayscale images (See Figs 1 and 2). Bezdek et al [7] also examined a related approach, using the Takagi-Sugeno fuzzy reasoning paradigm.

However Bezdek's approach is limited in a very basic way. It cannot be extended to larger window sizes. The argument is simple:

For a 3x3 there are only some 512 = 2^9 proto-types, and training is readily achieved in a matter of minutes. For a 4x4 NN operator, there are 2^{16} binary prototypes, and for 5x5 =2^25. Assuming training times are linear in the number of inputs we compute as follows: Training times for the 3x3 NN based operators take of the order of minutes, whereas for 4x4 the corresponding time would be of order of 2^7 = 128 minutes. But for a 5x5 operator training times would take of the order of 2^16 = 64K minutes = 45 days. In fact the linearity is not reasonable, and combinatorial explosion would be far worse.

We also question the basic strategy that Bezdek et al have adopted, in training neural nets to match "fuzzy" values produced by the classical operators - values v in the range 0<v<1.0.

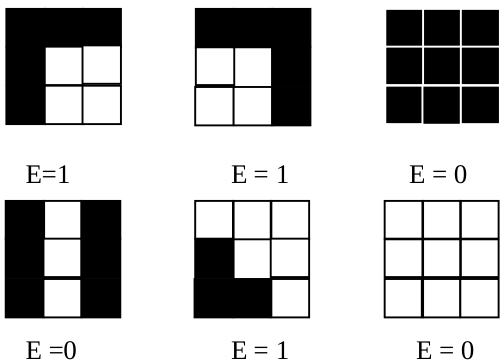


Figure 4: Examples of 3x3 binary windows for which Sobel edgedness E has crisp value. Note that the state of the centre pixel does not affect the value of E

3.0 Our new 3x3 NN Sobel

For the Sobel operator, using binary proto-type 3x3 windows, the crisp values of 0 or 1 are obtained: only for such cases as: indicated in Fig 4. The new strategy we propose involves training to crisp values, so that non-crisp values have to be "defuzzified", assigned to values 0 or 1, using a threshold T = 0.5. The results for Lena image in Fig 5 and 6 are striking: Training on binary prototypes has resulted in a reduction in the number of edge points found. But the new NN operator has detected features, such as the top half of Lena's hat, which are missing in Fig 5.

Figure 5: 256x256x8-bit Lena: Output of NN edge detector trained to duplicate "fuzzy" Sobel edgedness on 256 binary proto-types. Number of edge pixels = 7770.

Figure 6: 256x256x8-bit Lena: Output of NN edge detector trained to duplicate "de-fuzzified" Sobel edgedness on 256 binary proto-types. Number of edge pixels = 5895. Compare with Fig 5, espcially in mirror.