Manuscript submitted to *Neural Computing & Applications*, Version 1, October 22, 1996.

An Empirical Comparison of Neural Techniques for Edge Linking of Images

Stuart J. Gilson* and Robert I. Damper*†

*Image, Speech and Intelligent Systems (ISIS) Research Group,

Department of Electronics and Computer Science,

University of Southampton,

Southampton SO17 1BJ,

UK.

†Department of Computer Science and Engineering,

Oregon Graduate Institute of Science and Technology,

PO Box 91000,

Portland,

OR 97291-1000,

USA.

(emails: `sjg95r|rid@ecs.soton.ac.uk`)

Address for correspondence:

Until 1 February 1997:

R.I. Damper, PhD,
Visiting Research Professor,
Department of Computer Science and Engineering,
Oregon Graduate Institute of Science and Technology,
PO Box 91000,
Portland,
OR 97291-1000,
USA.

Tel: +1 503 690-1517
FAX: +1 503 690-1306
Email: `damper@cse.ogi.edu` or
`rid@ecs.soton.ac.uk` (will auto-forward)

After 1 February 1997:

Dr. R.I. Damper,
Image, Speech and Intelligent Systems (ISIS) Research Group,
Department of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ,
UK.

Tel: +44 1703 594577
FAX: +44 1703 594498
Email: `rid@ecs.soton.ac.uk`

List of Figures

List of Tables

Appendices

Abstract

Edge linking is a fundamental computer-vision task, yet presents difficulties arising from the lack of information in the image. Viewed as a constrained optimisation problem, it is NP hard – being isomorphic to the classical travelling salesman problem. Self-learning neural techniques boast the ability to solve hard, ill-defined problems and, hence, offer promise for such an application. This paper examines the suitability of four well-known unsupervised techniques for the task of edge linking, by applying them to a test bed of edge point images and then evaluating their performance both quantitatively and qualitatively.

Techniques studied are the elastic net, active contours, Kohonen map and Burr's modified elastic net. Of these, only the elastic net and the Kohonen map are realistic contenders for general edge-linking tasks. However, the other two exhibit behaviour which may make them particularly suited to some specific image-processing and computer-vision applications.

Keywords: computer vision, edge linking, elastic net, active contours (snakes), Kohonen net, constrained optimisation.

1 Introduction

Extracting features from images is a common computer-vision problem. In particular, edge-point images are problematic since there is inherently very little information available. Such images are taken here to consist of binary points on a two-dimensional plane which represent the edges and corners of an object. A feature-extraction algorithm then has to infer from the points a description of the object or objects present in the image. For most non-trivial images, however, there could be many interpretations of the object(s). Consider, for example, a character written by (human) hand using a coarsely-sampled electronic pen. The resulting image will consist of disjoint points which represent the curves describing the character. There will be many ways in which these points could be interpreted, but only one will reproduce the strokes of the original character.

Semantic knowledge of the form of the object may assist the extraction process, but generally, this is not available. So the task is simply to extract the most sensible object purely from the data in the image. Thus, the extraction mechanism must be able to cope with various fundamental object primitives, and tolerate affine transformations of composite objects.

One approach for the extraction process is to assume the existence of a single object only, and then to link the edge points in a meaningful way – to form a contour which should represent the object in the image. Such edge linking reduces the problem to that of enumerating the image points, and then ordering them into a sequence which describes a path corresponding to a computer model of the contour. One advantage of this approach is that the problem is now isomorphic to the classical travelling salesman problem (TSP) – see [1, section 4.2]. Hence, established techniques for solving the TSP can be used or adapted for edge linking.

Neural solutions to the TSP were first proposed by Hopfield and Tank [2], who imposed soft constraints on a recurrent network (fully-connected except for the absence of self-loops). These constraints were derived from a cost function formally equivalent to the network's Liapunov ('energy') function. The activation of the ijth neuron is either 1 or 0, according to whether city j is visited at step i of the tour, or not. As the energy is guaranteed to reduce with each neural update until equilibrium is achieved, so the cost function for the tour is minimised. Wilson and Pawley [3] later showed that this approach became overwhelmed by large problems ($>$ 30 cities). The elastic net of Durbin and Willshaw [4] overcame this limitation by using the same cost function as Hopfield and Tank, but enforcing some of the constraints more strongly based on the notion of a deformable model with an explicit structure. Indeed, Simic [5] has shown that the techniques share the same statistical-mechanics framework, and also proposes a 'correct' elastic net algorithm to overcome a small oversight in the original. He also shows that the elastic net's simulated-annealing mechanism, although differing from the more classical Metropolis method (see [6]), is an intrinsic part of the algorithm. That is, removing it – as Abrantes and Marques [7] have done – will result in very poor performance. The latter paper, attempted to show that the elastic net, the Kohonen map [8] and the active-contours algorithm of Kass, Witkin and Terzopoulos [9] are actually special cases of a common class of recursive algorithms which lend themselves to the edge-linking task.

This paper provides an empirical comparison of selected unsupervised 'neural' techniques which can be used for the edge-linking task. The three algorithms just mentioned are examined, in order to assess their applicability to the task. A fourth technique is also examined: a modification by Burr [10] of the elastic net algorithm, which is claimed to reduce learning times significantly without sacrificing the quality of the results.

The remainder of this paper is structured as follows. After concisely formulating the edge-linking problem in the next section, we then describe (Section 3) in some detail the four neural techniques to be compared. Each of the four is tested on a common, comprehensive test bed of images, described in Section 4, which allows the characteristic strengths and weaknesses of each technique to be seen. An important issue is how the free parameters of each neural model are set: this is detailed in Section 5. The methods are then evaluated with respect to their performance, using the methods described in Section 6. The results of this evaluation are detailed in Section 7 before concluding (Section 8).

2 Problem Formulation

Given a vector of units, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]^T$, the problem is to map its elements onto the vector of edge points, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N]^T$, such that the elements of $\mathbf{U}$, when traversed in ascending order, represent a path through the edge points of minimal cost. That is, both $\mathbf{p}_i$ and $\mathbf{u}_j$ are two-dimensional vectors specifying position in the plane. For a neural solution, the units will be formal 'neurons', inputs to the network are the $(x,\ y)$ coordinates of the current image point, and weights between inputs and a neuron are interpreted as the 'position' of that neuron. The number of units, M, can be equal to the number of edge points, N, but is usually greater (since more that one unit can be attracted to the same point), so the process will be a many-to-one mapping. The cost function and, hence, the path of minimal cost varies between applications, and in the general case the Euclidean metric is used, so the problem is to find the shortest path (i.e. TSP).

Two complications can arise when using TSP solutions for edge linking. First, the shortest path through the edge points is not necessarily the correct path. As edge-point density increases, however, the shortest path will tend towards the object contour. Secondly, the TSP assumes noise-free data, which is not usually realistic in computer-vision applications. Part of the motivation for this paper was to explore the effect of these complications.

Finally, on a notational point, we emphasise that emboldened upper-case symbols in this paper represent matrices while emboldened lower-case symbols represent vectors.

3 Neural Techniques for Edge Linking

Of the four techniques compared here, only three are actually unique in their approach. The elastic net, active contours, and the Kohonen map each attack the problem from a different perspective, while the fourth technique, Burr's modified elastic net algorithm is essentially a hybrid of the elastic net and the Kohonen map.

All the techniques are unsupervised. We view this as essentially beneficial since they do not make any assumptions about what they expect to find in an image. Additionally, all four are partially model-based – they try to map a generic model onto the image, by deforming the model until it matches the image as well as possible. Thus, knowledge of the problem (e.g. whether the object has a closed or open contour) could be encoded into the technique via the model, and the techniques then have a measure of the 'correctness' of their solution. Caution must be taken with such models, since spurious edges and false curves can easily arise if the model does not map naturally onto the data.

3.1 Elastic Net

Following the earlier work of Hopfield and Tank [2], Durbin and Willshaw [4] introduced a technique for solving the travelling salesman problem based on 'brain-style' computation, rather than the more common and obvious combinatorial methods.

The vector $\mathbf{U}$ (see Section 2) is represented by the units of the network, with the goal being to move each of the M units onto each of the N cities, while still retaining the essentially circular configuration of the model. This means that the algorithm has three objectives or constraints:

1. The path must pass through every edge point. The TSP specifies that every 'city' must be visited, irrespective of its position on the 'map'.

2. Adjacent units of the path should be moved to neighbouring edge points – this should ensure that the distance between units of the network is kept at a minimum.

3. The path should be kept as circular and as short as possible. This means that the path should not cross over itself, and that units which have not been assigned to an edge point are positioned such that they form a straight line between the units which do have assignments.

To achieve these goals, the units are subjected to two forces. The first simply pulls each unit toward the nearest edge point (satisfying objectives 1 and 2), while the second pulls each towards its two immediate neighbouring units (hence satisfying objective 3). These two forces act in competition, with each controlled by a governing factor allowing the algorithm to favour one over the other.

Thus, at each iteration of updating, the change made to the weights of the jth unit, $\mathbf{u}_j$, is given by:

$$\Delta \mathbf{u}_j = \alpha \sum_{i=0}^{N} G_{ij} (\mathbf{p}_i - \mathbf{u}_j) + K\beta (\mathbf{u}_{j+1} - 2\mathbf{u}_j + \mathbf{u}_{j-1}) \tag{1}$$

where α and β are factors governing the influence of the two components of the update rule, and K is a simulated-annealing term which is gradually reduced during learning.

As stated above, the first term is intended to move the jth unit towards whichever edge point has the greatest attraction, given by the normalised Gaussian term:

$$G_{ij} = \frac{\phi(|\mathbf{p}_i - \mathbf{u}_j|, K)}{\sum_{k=0}^{M} \phi(|\mathbf{p}_i - \mathbf{u}_k|, K)} \tag{2}$$

where $\phi(d, \sigma) = \exp\left(\frac{-d^2}{2\sigma^2}\right)$ is the Gaussian distribution function, with standard deviation σ.

The second (regularisation or 'smoothing') term considers the two immediate neighbours of the unit, and moves the unit towards their mid-point, hence minimising the distance between them and acting as an elastic or *tension* term.

Thus, the net behaves in use much like an elastic band (hence its name). Initially, it moves through image features, forming a contour which is coarsely centered and oriented with respect to the distribution of edge points. It then contracts and deforms back towards the edge points as the net cools. During the early stages of learning, when the 'temperature' is high, the net's structure is enforced quite strongly, allowing it to form a rough or vague solution yet retain its important, essentially circular configuration. As learning progresses, the temperature drops and the net becomes less rigid and more sensitive to image features, allowing it to deform more and specialise on a solution.

Note that the algorithm moves each unit according to the positions of every edge point. This has a two-fold advantage. First, every unit always moves – there are no 'dead' units – and although some may never actually be assigned to an edge point, they will still form part of a coherent contour. Secondly, every edge point will be linked (given enough iterations). Consequently, the elastic net is the only technique of those examined here that can almost guarantee to link every edge point in the image *and* do so in a sensible manner. This is a powerful argument when deciding which of these techniques is 'best'.

As already mentioned, the simulated annealing is a very significant contributor to the performance of the technique. Abrantes and Marques [7] omitted the annealing schedule in their mathematical framework, and the performance of the simplified algorithm was very poor, often failing to link even simple images. Hence, controlling the annealing schedule is an important consideration, and must be done correctly to ensure fast convergence and good solutions. Durbin and Willshaw recommend that the initial temperature, $K(0)$, is set to 0.2 and reduced by 1% every n iterations. For the purposes of this paper, $n = 1$ was found to be adequate, but very complex images may require higher values to prevent the net cooling too quickly and creating false edges (such as crossing over itself).

Annealing should cease when $K(t)$ reaches around 0.02 or 0.01 at iteration t – the lower value producing more accurate linking. The annealing mechanism is so powerful that this termination criterion is sufficient to define convergence, and so iteration can stop. This means that, for $n = 1$, the technique will only require just under 300 iterations to find a solution for *any* image. While not as low as some of the other techniques studied here, this number can also be considered as a guaranteed maximum number of iterations. None of the other techniques has such a guaranteed maximum.

However, of the three basic approaches, the elastic net is the most computationally expensive. For every unit, the positions of each of the cities must be considered, each normalised by the positions of all the other units. Thus, each iteration of updating requires M^2N operations, and time complexity is $O(N^3)$.

3.2 Active Contours ('Snakes')

The active contours technique proposed by Kass, Witkin and Terzopoulos [9] is another deformable model, but with much greater regularisation ('smoothing') forces. This means that the technique's behaviour is much more stable and predictable than the elastic net: because of the way the model behaves, active contours are also known as *snakes*.

This technique tries to model the structural aspects of the target object (the shape which is to be edge linked), such as its edge-coherence, the smoothness of its outline, whether the object is open or closed, etc. It does this via the deformable model – which has external energy (in much the same way as the elastic net), but which also has internal energy governing the model's behaviour.

3.2.1 Internal Energy

The snake's internal energy is composed of two terms. The first governs how it deforms elastically, effectively making it behave like a membrane, while the second component makes the snake behave inelastically (like a thin plate). Here lies part of the difficulty of setting up the snake, namely finding a suitable compromise between these two properties. Clearly, too much of the former will make the snake deform too easily, forming false edges, while too much of the latter will make the snake unduly rigid, preventing it from following the desired contour.

However, the internal energy does allow certain image features to be 'encoded' into the snake. For example, the thin-plate component could be made discontinuous (set to zero) if it is known that the image object includes distinct corners. These discontinuities will allow the snake to deform easily at those points and, hence, link the image with greater ease.

3.2.2 External Energy

To attract the snake towards image features, suitable forces must be applied in the form of external energy. These may take several forms, such as lines, line terminals, and even texture functionals. The one of interest in the context of this paper, however, is an edge-point functional: that used here is a Gaussian-scaled Euclidean distance measure (the same as equation (2), but without any normalisation):

$$\mathbf{F}_{ext}(\mathbf{u}_j) = \frac{1}{\sigma^2}\sum_{i=0}^{N}(\mathbf{p}_i - \mathbf{u}_j)\phi(|\mathbf{p}_i - \mathbf{u}_j|^2, \sigma)$$

The matrix form of the external energy makes the algorithm easier to work with. The update rule (as given in somewhat simplified form in [7]) is then:

$$\mathbf{U}^{t+1} = \mathbf{U}^t - \gamma \mathbf{A} \mathbf{U}^t + \gamma \mathbf{F}^t_{ext}$$

where $\mathbf{A}$ is an $M \times M$ pentadiagonal regularisation matrix, governing the snake's internal energy:

$$\mathbf{A} = \begin{bmatrix} c_1 & d_1 & e_1 & 0 & \dots & 0 & a_1 & b_1 \\ b_2 & c_2 & d_2 & e_2 & 0 & \dots & 0 & a_2 \\ a_3 & b_3 & c_3 & d_3 & e_3 & 0 & \dots & 0 \\ & & & \vdots & \vdots & & & \\ e_{M-1} & 0 & \dots & 0 & a_{M-1} & b_{M-1} & c_{M-1} & d_{M-1} \\ d_M & e_M & 0 & \dots & 0 & a_M & b_M & c_M \end{bmatrix}$$

where:

$$\begin{aligned} a_i &= \frac{\beta_{i-1}}{h^4} \\ b_i &= -\frac{2\beta_i}{h^4} - \frac{2\beta_{i-1}}{h^4} - \frac{\alpha_i}{h^2} \\ c_i &= \frac{\beta_{i+1}}{h^4} + \frac{4\beta_i}{h^4} + \frac{\beta_{i-1}}{h^4} + \frac{\alpha_{i+1}}{h^2} + \frac{\alpha_i}{h^2} \\ d_i &= -\frac{2\beta_{i+1}}{h^4} - \frac{2\beta_i}{h^4} - \frac{\alpha_{i+1}}{h^2} \\ e_i &= \frac{\beta_{i+1}}{h^4} \end{aligned}$$

with α_i and β_i, $i = 1 \dots M$, controlling the snake's elasticity and resistance to bending respectively.

A significant point about the snake technique is that it is effectively 'local', in that it only considers the information available in the immediate area: edge points more than three standard deviations away from a unit become insignificant. There are applications where this can be advantageous, but in edge linking it can be a problem as detailed in Section 7.2.2.

Note that, since the snake is constantly trying to minimise its internal energy, it will always contract in the absence of any external forces. This can be undesirable, such as when the density of features in the image is low. Alternative internal energies can be specified to avoid this problem (see [11]) but have not been considered in this paper.

3.3 Kohonen Map

The Kohonen map was primarily designed for classification purposes, and has been used for many applications in computer speech and vision (see [8, 12] for some classical examples). It also seems quite well suited to edge-point linking since the Kohonen map will always try to form solutions with some geometric coherence, i.e. similar feature vectors will be nearby in the output map. Thus, in the context of edge linking, nearby edge points will be mapped by nearby units of the network, hence forming a coherent path through the points.

As a result, the Kohonen map is subject to the same problems as the elastic net in this application, since the 'correct' path does not necessarily run through adjacent edge points.

The vector **U** is again modelled by the one-dimensional array of network units. Upon application of the input in the form of the edge points, vector **P**, we find:

$$k_w = \arg\min_k |\mathbf{p}_i - \mathbf{u}_k| \qquad\qquad 0 \le k \le M$$

where k_w denotes the index of the *winner* unit. The network is then updated by modifying the units in the neighbourhood of the winner unit according to:

$$\mathbf{u}_j(t+1) = \mathbf{u}_j(t) - \alpha(t)[\gamma(\mathbf{p}_i - \mathbf{u}_j(t))] \qquad j = k_w - \beta, \ldots, k_w + \beta$$

where α is a decay term for γ (the learning step) and β is the radius of the neighbourhood centered on the winner unit, and j is modulo-M. Note the similarity of this to the first term in the update rule, equation (1), of the elastic net.

Both the neighbourhood and the learning rate should be gradually reduced during the learning phase, allowing the map to specialise on a solution. However, an important side effect of the rate of neighbourhood reduction is the risk of having 'dead' units which, after learning, are left unassigned to an input vector. In certain applications, this is not too important, but in object recognition (and in vector quantisation) dead units can seriously affect performance. While suitable parameter choices can generally avoid dead units, the user should be aware of the problem and its solutions (e.g. the conscience-ness method of Ahalt *et al* [13]).

As can be seen from the algorithm (and the absence of anything like the second term of equation (1)), the Kohonen map has no explicit structure, and so would appear to be unsuitable for the task of edge linking. However, its use of neighbourhood regions means that adjacent units are often moved towards the same – or nearby – edge points, thus regions of the map are moved towards regions of the image, creating a topological map of the input points. The implicit structure allows the map to form solutions very quickly, since it is not constrained by any imposed geometric structure. But it also means that the solution found may not be appropriate, since it may lack any geometrical coherence. Often the map will link a shape, but will create false edges between points which shouldn't be linked, while another technique with explicit structure will avoid false edges since their creation would violate its cost function too much.

A strong point in favour of the Kohonen map is that it seems to be able to handle very high learning rates without becoming too unstable. Consequently, the method requires very few learning iterations. Further, the computational cost per iteration is also low at $2\beta N$ operations. Hence, this is a very fast technique. It opens up the possibility of using several

Kohonen maps for a particular problem and picking the best solution, without taking any more time than the other approaches considered in this paper.

3.4 Burr's Modified Elastic Net

The final technique studied here is a modification by Burr [10] of the elastic net algorithm. Essentially, the modification entails the addition of a new attraction term, and a new annealing mechanism, in an effort to reduce learning times. For this reason, Burr's technique (while not embodying any new principles) has been included in our comparison, to address the question: can the elastic net's learning times be reduced without affecting the quality of its results?

A new force term is added to modify the elastic net's nearest-neighbour (tension) term in an effort to speed up convergence of the units towards edge points. Whereas the existing term dictates where the unit will move by considering all the edge points, the new term only considers the attraction from the nearest edge point. So, although Burr doesn't draw this parallel, the new term effectively implements the Kohonen map style of learning [1, p. 246].

The update rule is the same as for the elastic net except that the new term $v_{ij}(\mathbf{p}_i - \mathbf{u}^*(\mathbf{p}_i))$ is added to the summation in equation (1). Here, $\mathbf{u}^*(\mathbf{p}_i)$ is the nearest unit to the point $\mathbf{p}_i$, and:

$$v_{ij} = \frac{\phi(|\mathbf{u}^*(\mathbf{p}_i) - \mathbf{u}_j|, K)}{\sum_{k=1}^{M} \phi(|\mathbf{u}^*(\mathbf{p}_i) - \mathbf{u}_k|, K)}$$

The elastic net algorithm simply allowed the temperature to drop independently of the net's state. Burr, however, uses an interactive (state-dependent) control which reduces K according to how close the net is to a solution:

$$K = \gamma \sqrt{\frac{\sum_{i=1}^{N} |\mathbf{p}_i - \mathbf{u}^*(\mathbf{p}_i)|^2}{2N}}$$

where the factor 2 in the denominator is included to account for some minor differences between Burr's formulation and Durbin and Willshaw's. Hence, K is effectively the root-mean-square of the distance of the network from the edge points. Thus, as a solution is approached, the temperature is reduced allowing the network to become more specific to that solution, and learn more detail. This is a very powerful annealing mechanism, since it is state-dependent. Thus, it avoids some of the 'useless' temperatures (those in which the network doesn't really change much, and only really serve to extend learning times) that other mechanisms suffer from. As such, it is probably the most significant contributor to Burr's claims of rapid ability to find solutions.

Consideration must be given, however, to the cost of each iteration of the algorithm. While the technique may be able to find a solution in fewer iterations than the standard elastic net, the computational cost of the modifications means that each iteration requires $(2M^2N + N)$ operations. This is significantly higher than any of the other techniques presented here, although the time complexity remains $O(N^3)$.

Another point worthy of consideration is that the metric used for the annealing mechanism is edge-point-based so that, as the number of edge points linked increases, the temperature drops, allowing the network to specialise further. The use of this metric, although it appears to work, could cause problems. As more edge points are linked, the temperature drops with the result that the net moves more slowly. If parts of the network still lie some distance away from image features, then these parts will move very slowly towards the solution. In this case, the network will learn very slowly (as illustrated in Section 7.2.4) and could well learn incorrect 'solutions'.

4 Image Test Bed

In order to test the techniques thoroughly, a set of artificial images was devised with the aim of demonstrating the sort of features which may occur in real images. There are 20 images, each 100 pixels square and consisting of black pixels on a white background. The test bed images are detailed in Appendix A. They include simple geometric shapes, progressing through to more complex contours with concave corners, to the more complicated 'signature'-like images. Each was drawn by hand onto the unit square, and was drawn deliberately sloppily to simulate the noisy edge contours expected in real applications. Note that some of the test images are duplicated, but with more edge points used to describe the contour – so examining the techniques' sensitivity to edge-point density. On the other hand, *Test01* and *Test02* are very low density images, included to assess how the techniques cope in the absence of strong image forces.

Also worthy of note are images *Test06*, *Test07* and *Test11*, which contain more than one feature. Further, *Test06* is ambiguous: is it two concentric circles, or a star shape? (See Figure 1.)

*** FIGURE 1 ABOUT HERE ***

Test12 places the two vertical lines of *Test11* closer together to encourage the nets to form a 'zig-zag' contour – jumping from one line to the other. *Test15* through *Test17* are examples of hand-written characters, including some contours which overlap and 'double back'. Finally, *Test18* is a signature minus its initial letter (*S*) which was written disjointly and whose inclusion would probably only confuse the techniques.

Although a somewhat arbitrary choice in certain cases (e.g. *Test11* and *Test12*), each image was labelled as either *closed* or *open* (see Appendix A) according to what seemed to be its most natural interpretation. This was, of course, necessary because the edge-linking techniques take different forms in the two cases.

5 Parameter Settings

For each technique, parameter values had to be determined that would give the best results for the test images. This involved an iterative process of initially guessing suitable parameter values, evaluating the results, and then refining the values further. Since this was a largely *ad hoc* procedure, two points should perhaps be made:

1. The values used are not the result of an exhaustive search of the parameter space. Such a search would be impractical, requiring an inordinately long time. Consequently, the parameters used may not be optimal for the task, although our experience suggests that they are at least near-optimal.

2. The parameter values were adjusted in an attempt to give best average performance across all the test images; they were not individually set for each image. This 'batch' approach reflects the likely application of these techniques, i.e. within an automated process where collections of images are presented to the network without human intervention. Hence, better values almost certainly exist for individual images (and, indeed, for other applications).

The above is especially true for the Kohonen map, which is rather under-specified in that there are many parameters to be set. Not only must values for its learning rate and initial neighbourhood size be derived, but also the rate of decay of both the learning parameter and the neighbourhood radius have to be decided. Linear decay seems a good starting point, but

Nasrabadi and Feng [14] used an exponential decay, and we have had some success with a state-dependent neighbourhood decay term.

There is some compromise when using almost any algorithm. Here speed, accuracy, stability and robustness co-vary, and can be traded. In the context of this paper, *stability* refers to how well a technique handles small changes in its parameters: changing a single parameter should not lead to a wildly different result. On the other hand, *robustness* refers to a technique's ability to handle a wide variety of images without requiring fine tuning of its parameters. Our aim is to find a balance between fast learning and accuracy, while maintaining both the stability and robustness of the technique.

Parameter values actually used in the remainder of this work are specified in Appendix B.

6 Evaluation

Evaluation of each technique's ability to edge-link the test images is a two-stage process. First, the accuracy of the edge linking needs to be quantified, i.e. how well the units have mapped onto the edge points and if there are any stray units or unlinked edge points. This first stage gives no consideration to the order in which the edge points are linked. This is the role of the second stage of evaluation, i.e. to decide if what has been linked is 'correct' in some sense.

6.1 Quantitative Measures

The goodness of the edge-linking process can be evaluated computationally. For a given output image, two scores are produced as below. Starting with both set to zero:

1. for each edge point in the image, if there is a unit within a radius δ of the edge point then increment the first score by one (even if more than one unit is found) and, finally,

normalise by the number of edge points;

2. similarly, for each unit of the network, if there is an edge point within a radius δ of the unit then increment the second score by one (even if there is more than one edge point found within the radius) and, finally, normalise by the number of units.

These two scores give a measure of *support* and *explanation* respectively [15]. The first measure refers to the way the data support the model, i.e. does the model encompass all the image features? The second measure considers how well the model explains the data, i.e. does every part of the model refer to an image feature? The image can only be considered fully linked if these two measures, lying in the range 0 to 100%, have maximal values (see Figure 2).

*** FIGURE 2 ABOUT HERE ***

Clearly, the specific value(s) of δ chosen will affect the particular support and explanation obtained. We used a range of δ values – from 1 to 5. At the lowest radius, $\delta = 1$, the scoring is maximally stringent, so support and explanation will both be low. We expect, of course, a general trend for support and explanation to increase with δ. At the highest radius, $\delta = 5$, the scoring is lax, so support and explanation will both be relatively high.

6.2 Qualitative Evaluation

This second form of evaluation is harder. The interpretation of exactly what is the 'correct' order in which to link an image is subjective. Thus, it can only be a qualitative measure and must be treated as such. In spite of this, there are some results which are clearly either 'wrong' or 'bad'. In designing the test images, we tried to ensure that they mostly had an obviously correct interpretation, so it is this type of result which we focus on in this paper.

Thus, qualitative evaluation considers how well the resulting contour describes the object assumed to be present in the image, rather than how well units map to image points, and vice versa – without regard to any notion of 'object'.

6.3 Network Initialisation

Having chosen a set of parameters, each algorithm was applied to edge-link all the test images. For closed contours, the network was initialised as points on a closed, centered circle of radius 0.4 (in terms of the unit square on which the images were drawn), and which encompassed most, if not all, of the image features. For an open contour, the network was initialised as a straight, horizontal line, centered in the image. In both cases, the M units were evenly distributed over the length of the initial configuration. These initial configurations were considered the most 'neutral' positions, but will inevitably handicap some of the techniques on some of the images. As with parameter derivation, there are likely to be better individual starting configurations which will improve performance for certain types of image.

In all cases, $M = \left\lfloor \frac{5N}{2} \right\rfloor$, where the floor function $\lfloor \cdot \rfloor$ denotes the integer part of its argument, since $5N/2$ is the value recommended by Durbin and Willshaw [4] for the elastic net. By and large, the appropriate number of units for the other techniques is not specified in the literature.

7 Results

7.1 Average Performance of the Techniques

Figure 3 shows the average values of support and explanation for the four techniques as a function of the radius parameter δ as defined in Section 6.1.

*** FIGURE 3 ABOUT HERE ***

The predicted trend for increasing support and explanation as δ increases is very obviously seen. More interestingly, the elastic net and Kohonen map clearly out-perform the other two techniques. In view of the efforts put into finding suitable parameters, we do not believe these differences can be explained merely on the basis of inappropriate parameter settings for the snake and the modified elastic net. There is a clear tendency for support to exceed explanation (except for the snake where the reverse is true – see below). For the elastic net and its (Burr) modification, this can be understood in terms of the competition between the attractive forces and regularisation terms – which act like a tensile force on the network. As a result, some units will remain unassigned to an edge point which also explains why a many-to-one mapping of units to edge points is used.

Since these results are for gross averages taken across the 20 images – having very different character – we do not infer that the elastic net and Kohonen map are superior to the two other techniques in all cases. To explore the possibility that even the poorly-performing techniques have some special strengths, making them particular suitable for particular applications, we have carried out a more detailed evaluation, as now described.

7.2 Performance on Image Groups

In order to assess how the techniques' edge-linking abilities vary with type of image, the test bed images were grouped into eight broad categories, according to our intuitive assessment of their nature. These groupings are detailed in Table 1. Note that there is some overlap in the groups in that some images are illustrative of more than one category.

*** TABLE 1 ABOUT HERE ***

For clarity, results are presented for a single, representative value of δ (= 2) in all cases. This value would be a reasonable choice for a practical application.

7.2.1 Elastic Net

Results for the Durbin-Willshaw elastic net assessed on groups of similar images are depicted in Figure 4. Generally, performance is good, particularly in terms of support which is high for all image groups. Explanation is lower because, as noted above, the elastic net's tension prevents all the units from clustering around their nearest edge point. Thus, some units will be too far from an edge point to contribute to the explanation. Explanation is especially low for the two 'trivial', low-density images of Group 1. These are 'trivial' in the sense of having a very simple solution, rather than implying that it is necessarily trivial to find that solution.

*** FIGURE 4 ABOUT HERE ***

Of the four techniques presented here, the elastic net is perhaps the most reliable. Given the parameter values used, its linking is both accurate and intuitive (in the sense that it is usually clear why it chose to link the edge points in the manner it did). It is also very stable and robust, being capable of handling a wide variety of images using the same set of parameter values. The simulated-annealing mechanism allows the net to position itself on the ‘centre of gravity’ of the image features before deforming to link them. This means that it is reasonably insensitive to its initial configuration. As previously mentioned, however, it is quite a slow technique although it will always link the image within 300 iterations. By comparison, the other techniques require an indeterminate learning period.

Attempts to reduce learning times by lowering $K(0)$ resulted in non-convergence after the predicted number of iterations. The lower initial temperature caused the network to move more slowly at first and, hence, it was unable to position itself near the mean distribution of the edge points. The resulting dead units would eventually be utilised, but only after many more iterations. Thus, the values suggested by Durbin and Willshaw do seem to be close to optimal for most problems.

Consider next the qualitative evaluation. The elastic nets, as well as being accurately mapped to the edge points, also produce quite meaningful contours. Most of the test images have been linked correctly, but in situations where edge points representing two different lines are close together (such as *Test12*), it zig-zags between the lines. This does question the elastic net’s ability to find the shortest path through a set of ‘cities’. In spite of this, good solutions were found to some of the more complex test images – the solution for *Test18* was not far short of perfect.

7.2.2 Active Contours

Results for the snake on the grouped images are depicted in Figure 5. Overall, these are disappointing, with the technique actually scoring zero on the simple 3- or 4-point problems of Group 1. In no case are both support and explanation reasonably high on the same type of image.

*** FIGURE 5 ABOUT HERE ***

Alone among the four techniques, the snake yields a better explanation than support. This is because it is a local, highly regularised technique, which prevents it from deforming sufficiently to link all the edge points. Consider the snake's solution to image *Test06* (Figure 6) where only the outer circle has been linked. Since there are two features in this image (and hence one model couldn't hope to link them both), the snake has arrived at the next best solution. That is, link one (giving good explanation) and ignore the other (leading to poor support). Although these properties have proved undesirable here, they may be of great significance in other applications where the ability to ignore some features could be advantageous.

*** FIGURE 6 ABOUT HERE ***

The snake is unable to link correctly images *Test04*, *Test06* and *Test07*, since it does not get the opportunity to 'see' all of the image, only the parts which are local. Incidentally, the higher-density version of image *Test04* allowed it to make a better job of linking the concave corners, since the extra edge points cause a stronger external force to be applied to the snake, allowing it to deform more.

To its credit, the snake is quite robust. Once suitable parameters have been set, it can make a good attempt at linking an image. However, finding those parameters is difficult since a small change in the learning rate can make the snake behave erratically. As a consequence, the learning rate must be kept small, requiring a large number of iterations to link the image. Even with the comparatively low computational cost of this technique, it can still require a significant amount of time to link an image (especially the higher density images which the snake seems to favour).

While the performance of the snake technique has been relatively poor in this comparison, it does allow easy inclusion into the algorithm of available semantic knowledge of the object. This can improve performance markedly. Unfortunately, because of the deliberate generality of the test bed, such knowledge could not be utilised here.

7.2.3 Kohonen Map

Results for the Kohonen net on the grouped images are depicted in Figure 7. Performance is comparable to the elastic net, with support about the same and explanation higher if anything. Performance is pleasingly even across the groups.

*** FIGURE 7 ABOUT HERE ***

Thus, despite its inherent simplicity, the Kohonen map is a very powerful technique. Its ability to link any image, irrespective of its complexity, combined with its speed means that it is a serious contender for feature-extraction tasks. It doesn't appear particularly sensitive to its starting position, nor to problem size, and is reasonably stable in use. The method actually performs better on this task with a near-maximal learning rate.

In spite of the lack of a tension term, the Kohonen map parallels the elastic net in having better explanation than support. This is because the technique makes no effort to avoid dead units, which is a significant problem in edge linking. It also links non-intuitively: under some circumstances it will create false edges for no obvious reason. The map is also under-specified in its usage. How the neighbourhood is reduced and how fast the learning rate decays can make significant differences to the performance, yet there is no theory to guide parameter choices.

As far as the qualitative evaluation is concerned, once a good set of parameters has been found, the contour produced by the Kohonen map are very encouraging. The correct contour was found on nearly all images: even *Test12* and *Test17* which caused the elastic net difficulty. Only the very complex images, such as *Test14* and *Test15* produced incorrect results.

7.2.4 Modified Elastic Net

Results on the grouped images for the modified elastic net are depicted in Figure 8.

*** FIGURE 8 ABOUT HERE ***

Despite the benefits claimed in Burr's paper, this net proves to be disappointing here. This is because the modifications result in a (fixed) tension term which is too high. Consequently, the network has difficulty expanding to link large images, and fails to link the image properly.

The net has become sensitive to initial configuration and has also become slower. While Burr claims improvements on the TSP with up to 50 cities, the net still performs badly here on smaller problems (as well as on the larger ones). The crippling factors appear to be the high tension term and the temperature control, which is based on the net's current state. The former prevents units becoming sufficiently close to edge points to reduce the latter significantly. That is, at a high temperature the attraction force between units and edge points never grows strong enough to overcome the high tension. The problem is compounded by some parts of network moving slowly once other parts have converged (see Figure 9).

*** FIGURE 9 ABOUT HERE ***

Qualitatively, Burr's technique produced results that would be expected of an elastic net at high temperature. The linking was poor as a result of a too-high tension term. This, however, did allow the method to produce correct solutions to some of the images that the elastic net failed on, notably *Test12*.

8 Conclusions

This paper compares four unsupervised neural techniques on the task of edge linking of images. The emphasis has been towards real, fully-automatic applications where the techniques would be expected to link a set of images without any user intervention, as would be the case in handwritten text recognition. Each technique has been implemented and then applied to a comprehensive test bed of 20 images which aim to expose the various characteristics of the techniques. The ‘best’ set of parameter values has been determined, and the results of these ‘best’ tests have been evaluated both quantitatively and qualitatively.

To summarise, the elastic net has proven to be a very powerful and stable edge linker. Although relatively slow, it is reliable and seems capable of dealing with large problems. The active contours technique is quite the opposite – unable reliably to link images, and very sensitive to its initial position with respect to image features. It should not be entirely discounted, however, since it does have several characteristics which may make it suitable for particular, well-controlled applications [9]. Although the Kohonen map (unlike the other three techniques) has no regularisation term, it also manages to link the test images smoothly. It is also very fast, which is an important consideration for on-line feature extraction (as in automatic mail sorting based on postcode recognition, for example). Finally, the modified elastic net provides slightly faster linking, but poor results for all problem sizes, large and small.

Several avenues for future work suggest themselves. First of all, the same value of M in terms of N – that recommended by Durbin and Willshaw [4] – was used for all the networks, as these are the only authors to specify a suitable value. Unfortunately, the recommendation appears incorrect: Simic [5] has shown that the elastic net only approximates the TSP tour length for large M, with the error tending to zero as $M \to \infty$. Hence, future work to explore suitable values for network size is planned.

While Kass *et al* [9] do not suggest an appropriate number of snake units to use for a given problem, it is clear that only enough units are required to give an accurate contour. Thus, there is not necessarily a relationship between M and N, but rather a relationship between M and some measure of the complexity of the target contour.

The Kohonen map, being designed for classification purposes, only requires in principle as many units as there are classes. Using more than the minimum number, however, can allow the map to produce better classifications. In this application, this can result in a more coherent contour.

An interesting area for future research is the investigation of dynamic node creation and/or destruction for these techniques. Amin [16] has done this with an elastic net/Kohonen map derivative with encouraging results.

The purpose of this work was to assess neural techniques for edge linking of images. Initially, we had hoped to identify a 'best' approach, which could then be the subject of further development: such as the incorporation of some notion of a 'correct' contour in the constrained optimisation – rather than just the shortest path. This might involve the development of a new cost function and/or imposing new constraints in addition to the existing ones. For example, Burr's algorithm (for reasons demonstrated above) may benefit from using a different error function for its temperature control. Similarly, the Kohonen map needs some method for dead-unit avoidance.

The work shows, however, that to identify a 'best' algorithm which should be used to the exclusion of all others is unrealistic. The specific characteristics of each and every application mean that no single technique is ideal for all of them. However, both the elastic net and the Kohonen map are quite well suited to most applications, yielding high support and explanation as well as being both stable and robust. As such, they provide a good starting point when searching for a solution to a specific problem. Interestingly, a combination of the two (which is effectively what Burr's technique is) did not necessarily produce a superior technique.

Acknowledgement

Author SJG is supported by a Research Studentship from the Engineering and Physical Sciences Research Council.

References

[1] J. Hertz, A. Krogh, R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison Wesley, Reading, MA, 1991.

[2] J. J. Hopfield, D. W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics* 1985; 52:141–152.

[3] G. V. Wilson, G. S. Pawley. On the stability of the travelling salesman problem algorithm of Hopfield and Tank. *Biological Cybernetics* 1988; 58:63–70.

[4] R. Durbin, D. Willshaw. An analogue approach to the travelling salesman problem using an elastic net method. *Nature* 1987; 326:689–691.

[5] P.D. Simic. Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimisations. *Network* 1990; 1:89–103.

[6] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi. Optimization by simulated annealing. *Science* 1983; 220:671–680.

[7] A.J. Abrantes, J.S. Marques. Unified approach to snakes, elastic nets, and Kohonen maps. *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, ICASSP '95*, vol. 5. Detroit, MI, 1995, pp 427–3430.

[8] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 1982; 43:59–69.

[9] M. Kass, A. Witkin, D. Terzopoulos. Snakes: Active contour models. *Int. J. Computer Vision* 1987; 1:321–331.

[10] D.J. Burr. An improved elastic net method for the travelling salesman problem. *Proc. IEEE Int. Conf. on Neural Networks*, vol. 1. San Diego, CA, 1988, pp 69–76.

[11] G. Xu, E. Segawa, S. Tsuji. Robust active contours with insensitive parameters. *Pattern Recognition* 1994; 27:879–88.

[12] T. Kohonen. The 'neural' phonetic typewriter. *IEEE Computer Magazine* 1988; March:11–22.

[13] S.C. Ahalt, A.K. Krishnamurthy, P. Chen, D.E. Melton. Competitive learning algorithms for vector quantisation. *Neural Networks* 1990; 3:277–290.

[14] N.M. Nasrabadi, Y. Feng. Quantisation of images based upon the Kohonen self-organizing feature maps. *Proc. IEEE Int. Conf. on Neural Networks*, vol. 1. San Diego, CA, 1988, pp 101–108.

[15] C.K.I. Williams. *Combining Deformable Models and Neural Networks for Handprinted Digit Recognition*. PhD thesis, University of Toronto, Canada, November 1994.

[16] S. Amin. A self-organised travelling salesman. *Neural Computing & Applications* 1994; 2:129–133.

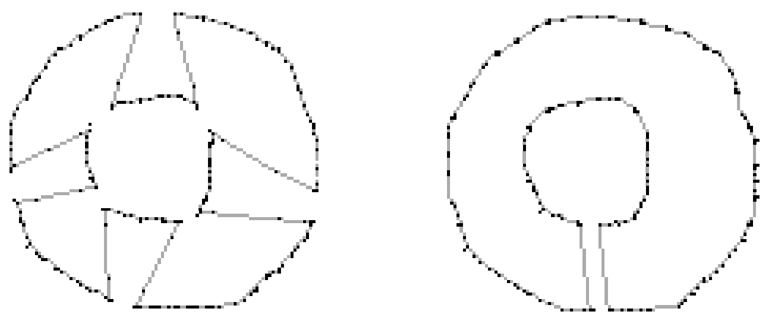

Figure 1: Two solutions to linking of image *Test06*. (These were actually produced by the Kohonen map with different parameters.).

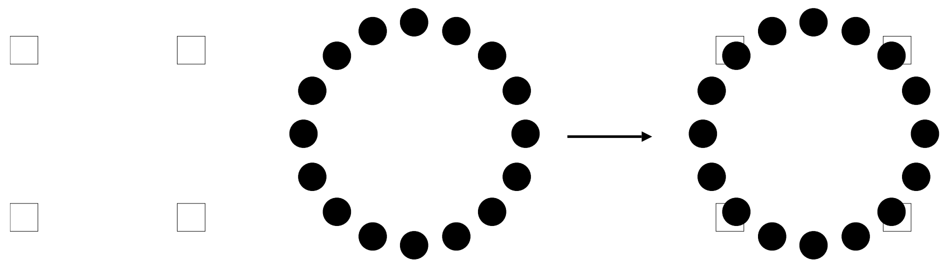

Figure 2: Demonstrating the difference between *support* and *explanation*. The left-most figure is the image, in the middle is a non-deformable model, and on the right is the result of mapping the model to the data. Clearly the data support the model well (100% in fact), but the model doesn't explain the data at all well.

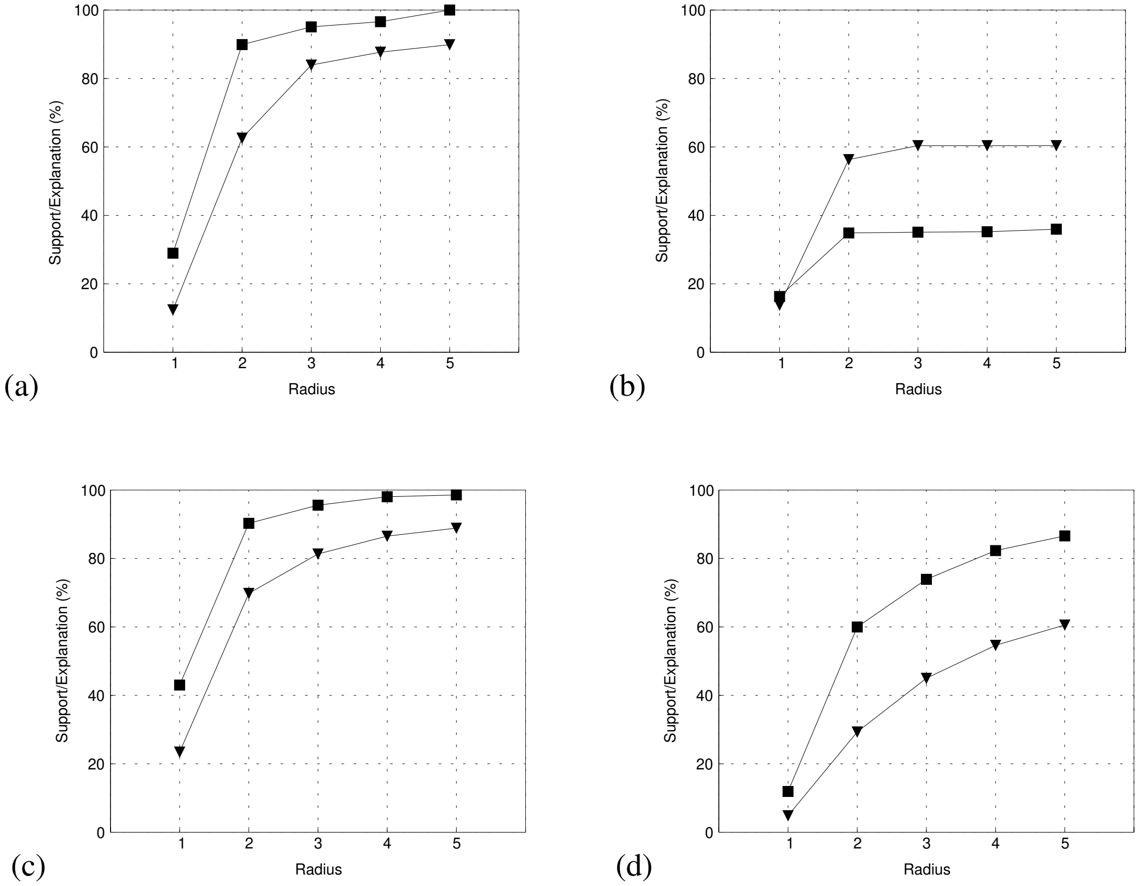

Figure 3: Support (filled box) and explanation (inverted filled triangle) as a function of the radius parameter δ for the four techniques studied: (a) elastic net; (b) active contours ('snake'); (c) Kohonen map and (d) Burr's modified elastic net. Values are averages across all 20 test bed images.

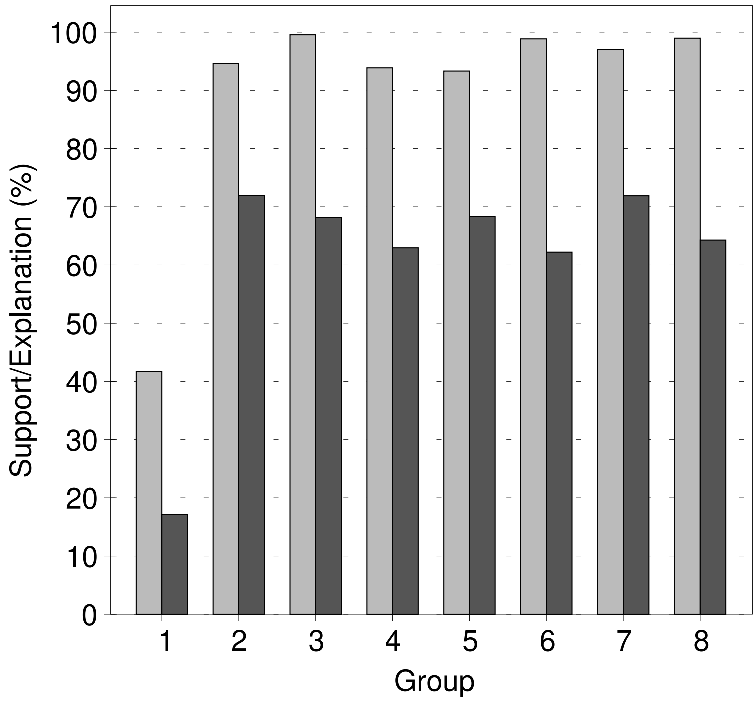

Figure 4: Histogram of support (grey bars) and explanation (black bars) values for the Durbin-Willshaw elastic net for the groups of images with radius parameter $\delta = 2$.

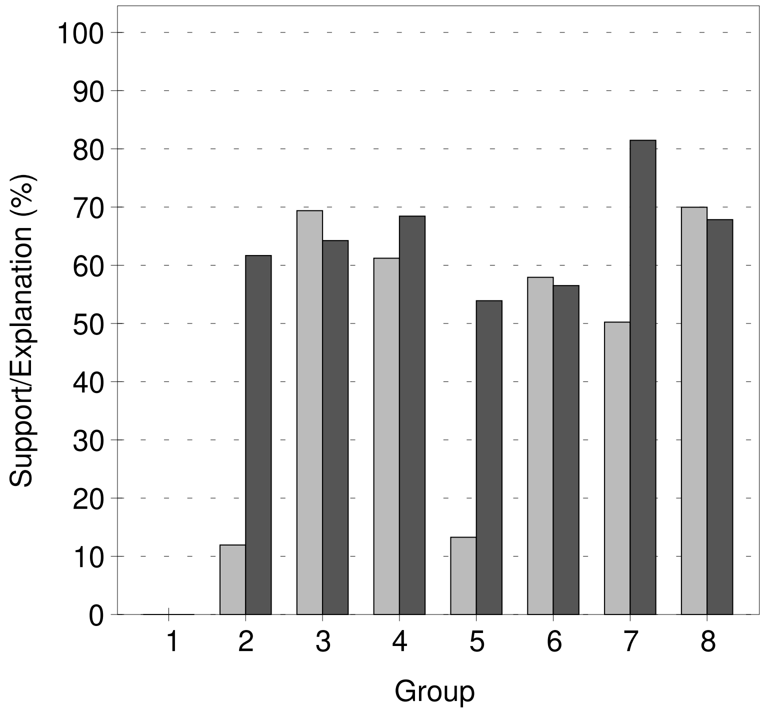

Figure 5: Histogram of support (grey) and explanation (black) values for the active contour model for the groups of images: $\delta = 2$.

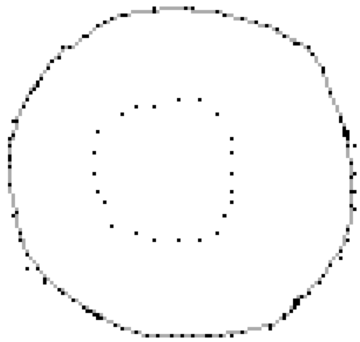

Figure 6: The solution to image *Test06* found by the snake. It has formed the convex hull of the outer circle, ignoring the inner one.

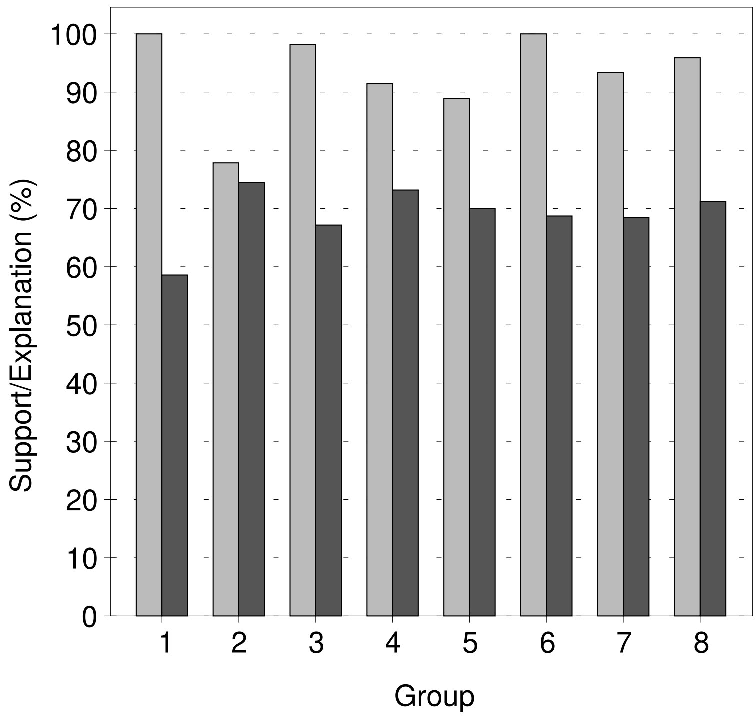

Figure 7: Histogram of support (grey) and explanations (black) values for the Kohonen net for the groups of images: $\delta = 2$.