A Grey-level Image Corner Detector using a Modular Neural Network

Rajeev Kumar
Department of Computer Science & Information Systems
Birla Institute of Technology & Science
Pilani - 333 031, India
rajeevk@bits-pilani.ac.in

Abstract

In this paper, we describe a reformulation of the corner detection problem as one in statistical pattern recognition enabling us to compute labels that are Bayesian posterior probabilities. We have generated the training data for our classifier using a grey-level model of the corner feature which permits sampling of the pattern space at arbitrary density as well as providing a self-consistent validation set to assess classifier generalization. Since adequate learning the whole mapping by a single neural network is problematic we trained a series of modules on data partitions and labelled corners by a hierarchical classifier. Results on real images are presented together with comparisons with the same labelling task performed by a monolithic network.

1 Introduction

Corners represent important features in images since they are highly localized in two-dimensions and consequently are points of high information content. A great deal of research effort has been expended over the years on feature detection however there is general agreement in the machine vision community that reliable corner labelling is a difficult problem. Conventional techniques for corner detection basically fall into two categories: those which attempt to estimate the second derivative of grey level variation (*e.g.*, [1]), and those which apply some convolution-based approach (*e.g.*, [2]). Central to both 'traditional' approaches is that having computed some (scalar) measure on the image, the presence of a corner is inferred if the computed measure exceeds some user-defined threshold. In general, there is no principled method to set this threshold; indeed what may be considered an 'optimal' threshold generally varies across a single image. These traditional techniques do not possess invariance to the grey-levels at which various corners are presented and robustness to noise is also a significant factor, particularly for techniques which numerically compute a second derivative.

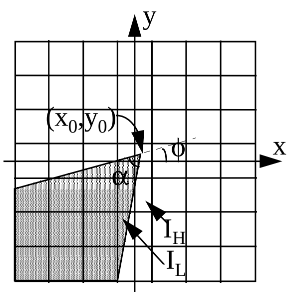
In this paper, we describe a reformulation of the corner detection problem as one in statistical pattern recognition enabling us to compute labels which are Bayesian posterior probabilities. From the consideration that the general labelling problem requires the correlated evidence from a region to be classified, it is clear that a trainable methodology is required and in this work we have adopted standard feedforward MLP neural networks trained by conventional error-backpropagation. An MLP has the advantage that it can be efficiently implemented in real-time hardware. Various attempts have been made in the past to apply neural networks to feature labelling using exemplars obtained from real images but these have been largely disappointing, due principally to use of grossly inadequate training sets. In this work, we used grey-level models of corner features to generate training data which adequately spanned the pattern space and thus give good generalization. The main problem here is that in order to adequately sample the pattern space of all corners with arbitrary resolution the training set needs to be large; additionally, the mapping proved difficult to learn.

We reduced the complexity of the classification problem by applying several pre-processing steps to approximately remove known invariance properties of corners. Rather than attempting to train a single, monolithic network we have stratified the training set and trained a series of modules on parts of the labelling problem. We have used the outputs of the individual modules to generate a second training set of *meta*-patterns which have been used to train a further network which combines the results of the individual modules. We present the results of corner labelling on real images, and compare with the same labelling task performed by a monolithic network. We demonstrate that the results of labelling real images with this modular approach are superior to those from labelling with a single monolithic network. Additionally, the overall training time is greatly reduced compared to that for training a monolithic network.

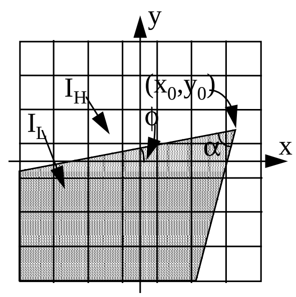
2 Generation of Training Data Set

Grey level corner training patterns were generated from two intercepting straight lines in a 7x7 lattice of pixels. If the intersection point of the lines (the '*corner-tip*') lies within the central pixel of the patch, the pattern is a *corner* (C). Thus an instance of a corner is described by its opening angle (α), inclination to the x-axis (ϕ), the co-ordinates of the tip (x_o, y_o), and high-low intensities (I_H, I_L) - Figure 1(a). Additionally, 'dark' corners on a light ground are distinct from 'light' corners on a dark ground so both 'polarities' of corners need to be represented.

Our ultimate objective is labelling with the Bayes' posterior probability and so the total training set needs to reflect the prior probability of occurrence of corners and non-corners in real images [3]. We typically observe a ratio of corners to non-corners in the range 1:50-1000; we have taken a figure of 1:100 in this work and so for every corner pattern we require 100 non-corner exemplars. Within the class of non-corners we require examples of all possible features. Within our corner definition, when a corner's tip falls outside the central lattice cell it is a non-corner - we refer to this as an non-obvious non-corner (NONC) - Figure 1(b). For a 7 x 7 patch we require 48 NONCs per corner. We also require a fraction of edges (E) and non-obvious non-edges (NONE) which by analogy with NONCs, are edges which do not pass through the central cell. Finally, we require a number of uniform grey-level patches.



(a) Corner sample



(b) Non-Corner sample

Figure 1: Corner training pattern

We have used increments in the corner model of $\Delta\phi = 2.9^o$ and $\Delta x_o = \Delta y_o = 0.2$ pixel; the rational behind this choice is discussed in [4]. We have employed a range of heights of grey-level step of 1:2,4,6,8,10 [5]. With these increments, each corner 'polarity' for a single opening angle yields 900 patterns resulting in 90,000 (900 x 100) non-corners although in practice we sub-sample this number - see Sub-section 3.2.

3 Neural System Architecture

In practice, we would like to detect corners in range of $0^o < \alpha < 180^o$, however, this requires an extremely large training set. Additionally, this training set is highly imbalanced. Chen [4] adopted a *two-pronged* strategy for reducing the data-size: firstly pre-processing the training data to remove the known invariances, and secondly altering the priors by sub-sampling the non-corner data. Despite this data compression Chen found that learning the necessary mapping of *all* corners on a single network was difficult. We attribute this to the possible effects of temporal cross-talk [6] where the presence of conflicting training information retards learning. So we have added a *third* strategy in the form of 'functional-decomposition' where different training subsets are mapped on individual networks.

3.1 Pre-processing to Remove Invariances

As a first step towards easing the network learning task we have pre-processed the grey level data to (approximately) remove the known invariances. To reduce the required number of examples at different step heights we have taken the common logarithm of the pixel intensities; this produces a significant compression in the training set. Then we rotate the patch according to a measure of orientation of the (putative) corner such that the corner is always directed into the positive quadrant. Finally, since we are looking for regions of grey-level discontinuity we calculate image gradient using a 3x3 Sobel operator. Thus the input to neural classifier is a 5x5 vector - Figure 2.

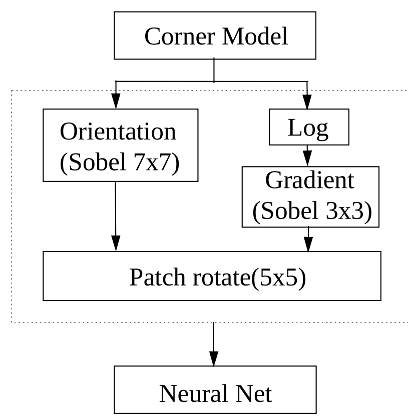


Figure 2: Overall System Architecture

3.2 Altering the Priors and De-biasing

An additional practical problem is that since corners are comparatively rare events, the training set is unbalanced and would produce difficulties in training. One solution lies in a modified back-propagation learning algorithm [7] but such modification alters the posterior probabilities of classification and is not appropriate to Bayesian labelling of the image features. We have surmounted this problem by deliberately training with a corner prior of 1:24 (instead of 1:100) by sub-sampling the non-corner data and then removing the resulting bias in the posterior probability by remapping to the correct prior [8]. Let $N(C|X)$ and $N(NC|X)$ be the numbers of corners and non-corners examples, respectively; then the *original* prior probability $P(C|X)$ of a corner is:

$$P(C \mid X) = \frac{N(C \mid X)}{N(C \mid X) + N(NC \mid X)}$$

On scaling the number of non-corners to $\gamma N(NC|X)$ where $\gamma = 0.24$, the *revised* prior probability of a corner pattern in this recomposed pattern space is:

$$P'(C \mid X) = \frac{N(C \mid X)}{N(C \mid X) + \gamma\, N(NC \mid X)}$$

On rearranging these equations, the original prior-probability is restored to

$$P(C|X) = \frac{\gamma\, P'(C|X)}{1 - (1-\gamma)\, P'(C|X)}$$

This prior remapping has been applied as a final (non-neural) processing step

3.3 Bootstrapping to Modularity

The experience of Chen [4] is that learning the necessary mapping for *all* corners is difficult and so we have employed a modular approach wherein we have stratified the corner data according to opening angle, α. Starting with a single module trained on 90^o corners alone, we have used essentially a bootstrap procedure [9] to partition the training data. From the response of this first trained module to a verification set containing corners with *all* opening angles, we select the *single* opening angle for the next module such that its minimum response would rise to 0.5 at the point where the minimum response of the 90^o module fell to 0.5. Thus we obtain a modular network composite of overlapping responses. In fact this procedure gives two angles, one below 90^o (*i.e.*, 70^o) and one above 90^o (*i.e.*, 110^o). This bootstrap procedure was repeated resulting in five modules each trained on single opening angles of: 50^o, 70^o, 90^o, 110^o and 130^o, respectively.

Adding further modules is unlikely to be of value since there is genuine confusion between the patterns; as the opening angle increases towards 180^o there is increasing confusion with straight lines. Similarly, trying to locate corners with very acute angles is genuinely problematic.

Visualization of the training data using standard ordination techniques showed the two types of corners (light and dark) to be rather disjoint. Indeed it proved difficult to learn both light and dark corner patterns with one network so separate modules were used for each opening angle for both light and dark corners giving a total of ten modules which were used to generate the *meta-pattern* data.

One simple solution for combining the outputs of multiple learned net would be a *winner-takes-all* strategy, however selecting the largest output may not be an ideal choice since potential valuable information may be wasted by discarding the results of less successful modules [10]. This observation motivates us to ignore the winner-takes-all strategy and combine the outputs of several modules through learning of meta-patterns. There are potentially many approaches for integrating multiple learned models for improving class-prediction, see [11] for the state of art of the research. We combined this meta-pattern data, comprising some 224 thousand examples to train the fusing network which in turn outputs the final class probability. This approach has the advantage that we could suppress some of the false positives and lower the confidence level of a few others in the final labelling stage which otherwise would have been present in a winner-takes-all strategy. Some weak labels were also strengthened. The overall classifier system is illustrated in Figure 3.

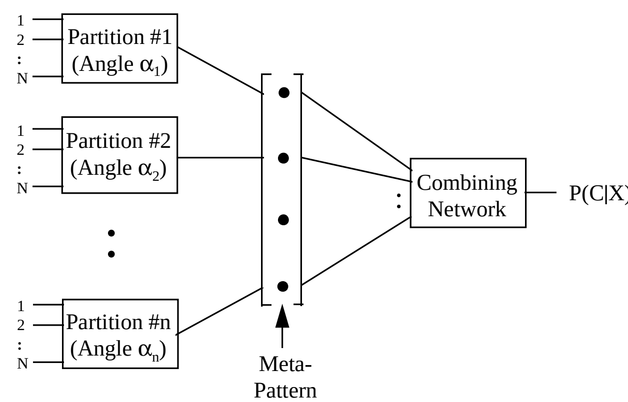


Figure 3: Modular classifier architecture

3.4 Learning and Generalization

Each sub-network was a multi-layer feedforward network with a sigmoid activation function at each node. The modules used to learn the individual corner opening angles comprised 25 input nodes, 4 hidden nodes and a single output node. The fusing network comprised: 10 input nodes, 4 hidden nodes and again a single output node. We used the resilient backpropagation (RPROP) learning algorithm [12], a local adaptive learning scheme for faster learning but we applied a very small weight step in each iteration of weight adaptation so that the weights do not adopt very large values. Additionally, we use a moderate value of weight decay term in the cost function so that the weights are not increased arbitrarily; again this is solely guided by the generalization of the network. Generalization becomes still difficult to control for networks trained with noisy data. Training with noise is equivalent to Tikhonov Regularization [13], the overfitting may be controlled to some extent. However, in this work we restrict training of networks with clean data.

One important aspect of this training is the generalization of the network and the avoidance of overfitting. This phenomenon is important because our training set consists of synthetic data, the validation set is synthetic but the test data is real. We have used the validation data generated with those parameter values which were not used for the training data and with smaller parameter increments for a denser sampling. Though there is no principle for deciding the number of hidden nodes in a network, it is always desirable to minimize the number of free parameters. We have employed the number of hidden node units dictated by the significantly sized eigen-values of the training data [14], which in our case is roughly equal to four. By adopting such a small number of hidden units for a very large training data the training becomes quite slow, but we are more interested in avoiding overfitting than easy learning. There are, however, other heuristics as well for approximating the minimum number of hidden layer neurons, *e.g.*, [15]. We experimented with networks of different number of hidden units but found that the network-responses of four hidden nodes are the best for the corner pattern-data used.

4 Application to Real Images

Figure 4 shows the labelling results of a complex real scene for the module trained with 90^o dark corners on a light ground. We also trained with 90^o light corners on a dark ground. (Throughout, our test images were 256 x 256 pixels and digitized to 256 grey-levels, crosses indicate the points where the label is greater than 0.5). A number of correctly labelled corners are evident in Figure 4, however, it is evident that not all unambiguous corners are labelled consistently by modules. It is also evident that there are a number of false positive labellings, particularly in the results for light corners on a dark ground although many of these originate from the highly cluttered region around the house windows. We discuss the false positive labelling below with reference to the grey levels present in this test image. Interestingly, none of the dark corners labelled in Figure 4 is re-labelled with the module trained with light corners, and vice-versa; this phenomenon indicates a reasonably *disjoint* mapping of dark and light corners on the respective modules.

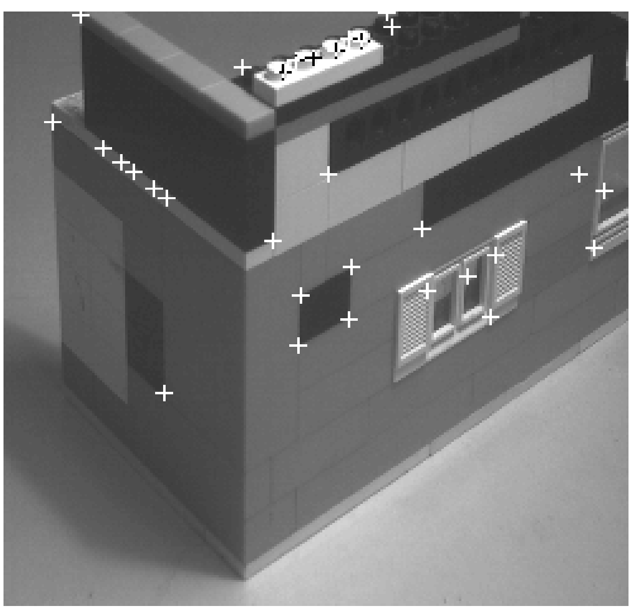


Figure 4: Labelling results for individual modules trained with 90^o dark and light corners.

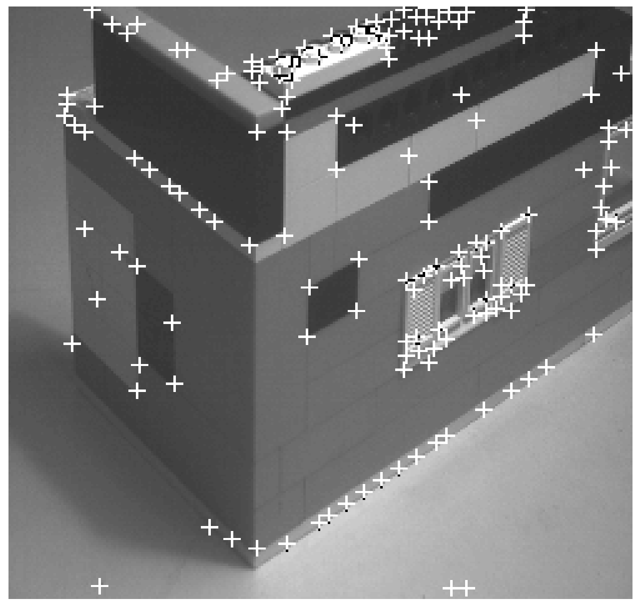


Figure 5: Results of labelling with the modular network - Labels thresholded with 0.5

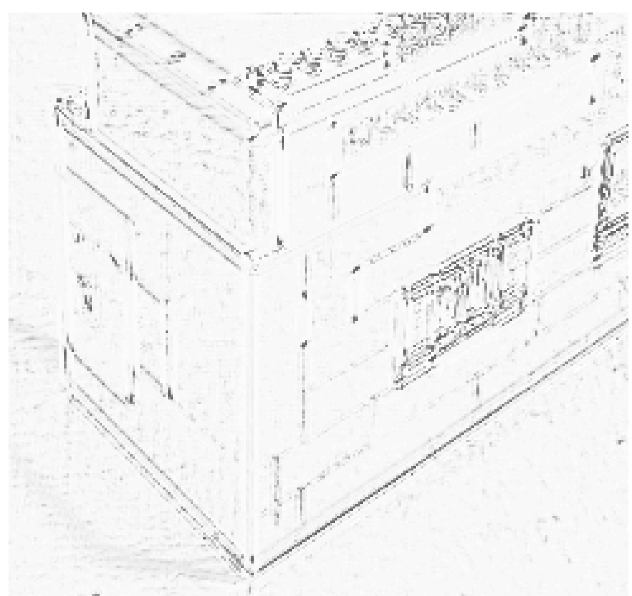


Figure 6: Results of labelling with the modular network - Output probability map.

Figure 5 shows the results from labelling a typical image above using the modular network which combines the individual responses in Figures 4 and nine other modules trained on different opening angles for both th epolarities. Figure 5 shows the original grey level image with the points for which $P_{out} \geq 0.5$ superimposed as crosses, while Figure 6 shows the output probability map. Almost all the perceptual corners in this image are correctly labelled here. From the probability map in Figure 6 the general background label is rather small - as it should be - and only those regions containing significant spatial grey-level changes have significant label values. It is evident that the bottom edges of the house together with the horizontal features on the side of the house are strongly labelled and in a number of places these label values rise above 0.5 producing false positives. Since our primary motivation is to use the outputs of the present corner detector as input to further machine vision tasks, we have been keen to explore the origins of these false positive labels. Close inspection of the original image reveals that these false positives are almost always associated with some extended distortion or region which is not apparent when viewing the image at full scale. The two 'corners' at the very bottom of the image, for example, are some kind of dark blot several pixels across, and the cluster of false corners at the top, front edge of the house are due to the (unresolvable) fixing pegs on top of the Lego bricks - insofar as these are two-dimensional structure they are arguably 'corner-like'. A close-up of the front bottom edge of the house shows significant pixel aliasing, so the consequent two-dimensional structure produces quite strongly labelled false positives.

Figure 7 shows the labelling results from a single monolithic network trained on the union of all the data used to train the individual modules of the modular network. This figure shows the best labelling result we obtained from repeatedly reinitializing and training a number of networks with differing numbers of hidden nodes. The probability map from the monolithic network displays a general background 'fog' in contrast to the background for the modular network in Figure 6. Interestingly, the edge features which are considered somewhat corner-like by the modular network are considered *less* so by the monolithic network than the uniform background. We suggest this may be a consequence of the temporal cross-talk phenomenon discussed by [6]. Monolithic network has a poor mean-square-value (MSE) after a very long training time. Figure 7 shows the points for which the labels were greater than 0.5 and although some corners are correctly labelled, many are not. Overall, the results for the combined modular network are greatly superior to those of the monolithic network; moreover the training time of the monolithic network was a factor of two larger than that of the total time required to train the modular network.

Finally, we make comparison with the results from employing the well-known Harris & Stephen's corner detector in Figure 8. Here we have adjusted the heuristic threshold in this algorithm to obtain the 'best' performance but it is apparent that the labelling is inferior to that obtained with our modular neural network approach. Due to space limitations, we are unable to include more results.

5 Discussion

In presenting the preceding results we have taken a probability of 0.5 as the threshold for deciding whether or

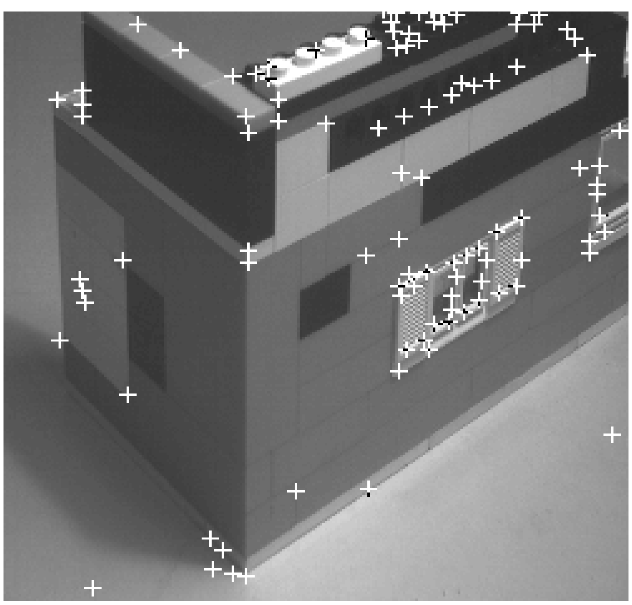


Figure 7: Labelling results from single, monolithic network - labels thresholded with 0.5.

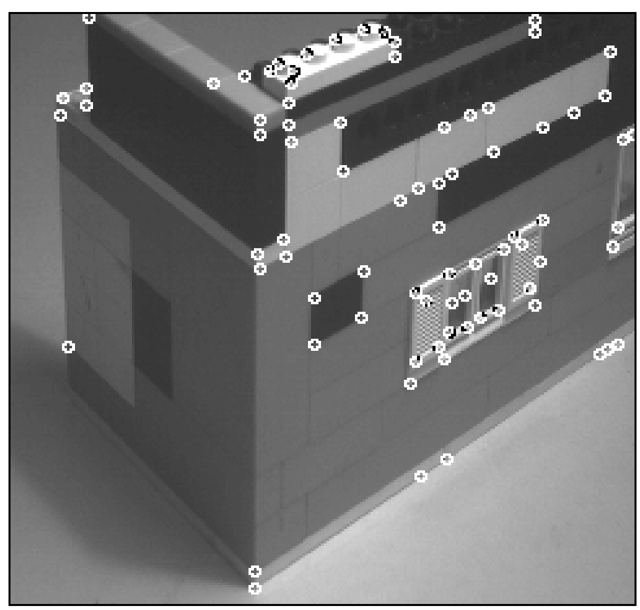


Figure 8: Labelling results with Harris & Stephen's corner detector - thresholding is adjusted to get the 'Best' results.

not a corner is present. In the sense that our modular network approximates the Bayesian posterior probability and that this quantity can be interpreted as a *measure of belief* about the presence of the corner, a decision threshold of 0.5 can be viewed as rather conservative since it represents a level at which we are only marginally more confident that a site is a corner as opposed to a non-corner. Raising our decision threshold to, say, 0.6 means the confidence level of having a corner is 50% greater than our belief that the same site is a non-corner, a significant number of false positives disappear for this raised belief threshold although we also lose a few of the corners from the side of the house. This latter observation is consistent with the observation from the Harris & Stephen's labelling in Figure 8 that the corners on the house's side are somewhat harder to correctly identify.

The problems caused by aliasing that are apparent in Figure 6 are of great practical importance. We speculate that one way of reducing the effect of such aliasing may be to further extend our modular approach to include not only direct pixel-level evidence from the immediate neighborhood of a corner but also *indirect* regional evidence of edge configurations compatible with the existence of a corner.

All the networks discussed in this paper were trained on noise-free corner examples and when applied to real-images (obviously corrupted with noise) give superior performance to those obtained with conventional corner-techniques. We did not experiment with training corner patterns corrupted with noise. But it is beyond doubt that a monolithic network which can not be well trained on noise-free data can not be trained with data corrupted with noise. However, the modular network certainly can have a better mapping. Hence we infer that the modular network if trained with an appropriate level of sensor-noise simulated in the imaging model or with another better modelling of corner patterns will give even better performance than has been achieved in the present work. Thus the current technique is directly applicable to a re-modelling of training-data which better samples a universe. Such flexibility is not possible with conventional methods of corner detection.

6 Conclusions

In this paper, we have presented a modular neural network for labelling corner features with posterior probabilities. The network has been trained on data from a grey-level model of corners which has been partitioned across modules using a bootstrapped procedure. The results from the modular network are shown to be superior to both those from a single, monolithic network and conventional corner detectors.

References

[1] L. Dreschler, and H.-H. Nagel, "Volumetric model and 3D trajectory of a moving car derived from monocular tv-frame sequence of a street scene", *Computer Graphics & Image Processing*, vol. 20, no. 3, pp. 199-228, 1982.

[2] C. Harris and M. Stephens, "A combined corner and edge detector", *Proc. Alvey Vision Conf.*, Univ. Manchester, pp. 147-151, 1988.

[3] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities", *Neural Computation*, vol. 3, no. 4, pp. 461-483, 1991.

[4] W. C. Chen, *Probabilistic labelling of edge and corner image features with neural network*, Ph.D. Thesis, Dept. Electronic & Electrical Engineering, University of Sheffield, 1996

[5] B. P. Amavasai, "Neural network approach for 3-D depth and motion analysis on a saccadic camera system" Technical report, Dept. Electronic & Electrical Engineering, University of Sheffield, 1977.

[6] R. S. Sutton, "Two problems with backpropagation and other steepest-descent learning procedures for networks", *Proc. 8th Annual Conf. Cognitive Science Society*, Lawrence Erlbaum, pp. 823-831, 1986.

[7] R. Anand, K. G., Mehrotra, C. K. Mohan, and S. Ranka, "An improved algorithm for neural network classification of imbalanced training sets", *IEEE Trans. Neural Networks*, vol. 4, no. 6, 962-969, 1993.

[8] A. J. Lacey, N. A. Thacker, and N. L. Seed, "Smart feature detector using an invariance network architecture", in D. Pycock (Ed.), *Proc. 6th British Machine Vision Conf.*, pp. 327-336, 1995.

[9] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, London: Chappman & Hall, 1993.

[10] W. H. Wolpert, "Stacked generalization", *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992.

[11] S. J. Stolfo, *Integrating Multiple Learned Models for Improving & Scaling Machine Learning Algorithms*, Proc. AAAI Workshop, AAAI Press, 1996.

[12] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm", *Proc. IEEE Int. Conf. Neural Networks*, IEEE Press, pp. 586-591, 1993.

[13] C. M. Bishop and C. Legleye, "Training with noise is equivalent to Tikhonov regularization", *Neural Computation*, vol. 7, pp. 1108-1116, 1995.

[14] A. S. Weigend and D. E. Rumelhart, "The effective dimension of the space of hidden units", *Proc. IEEE Int. Jt. Conf. Neural Networks*, pp. 2069-2074, 1991.

[15] K. Agyepong and R. Kothari, "Controlling hidden layer capacity through lateral connections", *Neural Computation*, vol. 9, pp. 1381-1402, 1997.