Support Vector Machine Experiments for Road Recognition in High Resolution Images

J. Y. Lai[a], A. Sowmya[a], J. Trinder[b]
[a]School of Computer Science and Engineering
(jlai, sowmya)@cse.unsw.edu.au
[b]School of Surveying and Spatial Information Systems
j.trinder@unsw.edu.au
University of New South Wales, Sydney, NSW 2052, Australia

UNSW-CSE-TR-0413
March 2004

THE UNIVERSITY OF
NEW SOUTH WALES

Abstract

Support Vector Machines have received considerable attention from the pattern recognition community in recent years. They have been applied to various classical recognition problems achieving comparable or even superior results to other classifiers such as neural networks. We investigate the application of Support Vector Machines (SVMs) to the problem of road recognition from remotely sensed images using edge-based features. We present very encouraging results in our experiments, which are comparable to decision tree and neural network classifiers.

1 INTRODUCTION

Road extraction from remotely sensed images is an important process in the acquisition and updating of Geographical Information Systems. Automatic and semi-automatic road recognition is an active area of research [7]. RAIL is a road recognition system that has been under development by our group for a number of years. It serves as a framework to research new directions in applying machine learning to image understanding, and our particular application is road recognition [4], [10].

Support Vector Machines provides a relatively new classification technique that has grown from the field of statistical learning theory. Despite its recent arrival it has proven to be a very powerful classifier.

There are two main motivations to incorporate SVMs into RAIL. First of all, SVMs have been successful in other application domains. However, there have been no results (prior to [12]) published on applying SVMs to the problem of road recognition. Therefore, our experiment will be of interest to pattern recognition communities as well as remote sensing researchers. Secondly, RAIL uses a meta-learning framework to learn the strengths and weaknesses of different machine learning algorithms. Incorporating SVMs into RAIL expands the base algorithm sets to promote meta-learning research.

This paper is organised as follows. Section 2 briefly introduces SVMs and its applications. Section 3 describes implementation improvements on RAIL. Section 4 describes the experiment and the results are presented in Section 5. We summarise our results in Section 6.

2 SUPPORT VECTOR MACHINES

Support Vector Machine is a relatively new method for pattern classification and nonlinear regression. It is based on the principles of statistical learning theory originally proposed by Vapnik in 1979. SVMs construct a hyperplane in the feature space that separates the positive and negative training samples. We wish to find a hyperplane that gives the smallest generalisation error among the infinite number of possible hyperplanes. Such an optimal hyperplane is obtained by maximising the margin of separation. The margin of separation of the hyperplane is the sum of distances from the hyperplane and the closest vectors in each class, known as the support vectors. In Fig. 1, A has a larger margin than B, and is more desirable. The margin can be controlled to avoid overfitting of the test data by penalising misclassification error.

When training samples are not linearly separable in the feature space, kernel functions are used to map the data from the input space to a higher dimensional feature space F via a nonlinear mapping $\Phi : \Re^N \rightarrow F$. The optimal hyperplane is found in F and then mapped back as a nonlinear surface in the original feature space.

2.1 Applications

SVMs have been applied to many classic pattern recognition problems with great success including face recognition, hand-written character recognition, speech recognition and many others [1].

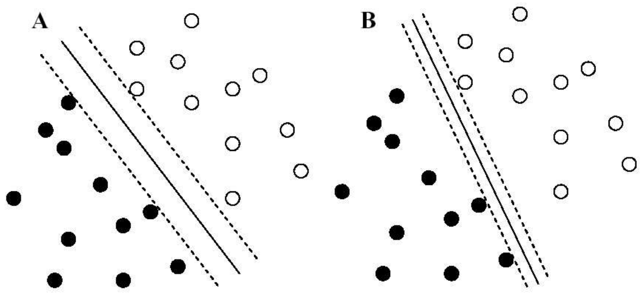

Figure 1: Maximising the Hyperplane Margin

In the domain of remote sensing, SVMs have been applied mostly to land cover classification. In [2], hyperspectral data of 128 bands was used to classify 6 types of crops. SVM yielded better outcome than neural networks. SVMs also performed reasonably well in situations where feature selection was not used. In [8], they reported that SVMs performed better than maximum likelihood, univariate decision tree and backpropagation neural network classifier, even with small training data sets. Both groups used pixel-based features.

3 RAIL

RAIL is an adaptive and trainable multi-level edge-based road extraction system which has been developed within our group for a number of years [10], [5]. Starting with low-level objects (edges), RAIL incrementally builds higher-level objects (road network). The levels of classification are:

1. Road Edge Pairs - pairs of edges that enclose a segment of road.
2. Linked Road Edge Pairs - adjacent road edge pairs that form continuous roads.
3. Intersections - road edge pairs that meet form intersections.
4. Road Network - linked roads and intersections.

SVM has been previously applied to the preprocessing stage (edge extraction) and Level 1 of RAIL with encouraging results presented in [12]. This paper extends the use of SVM to Level 2 while removing SVM use in the preprocessing stage. Several implementation improvements have been made to RAIL that affected the previous SVM experimentation. These include the image processing stage, the reference model, feature extraction and feature selection stages. They are discussed in the following subsections.

3.1 Image Processing

The parameters used in Vista's Canny edge-detector were tuned to produce outputs with less noise. This was accomplished by adding noise to the original image prior to a Gaussian

Table 1: Extracted Features

Level 1	Level 2
Width (mean)	Width (mean)
Enclosed Intensity (mean)	Width (var)
Enclosed Intensity (var)	Width Difference
Pair Length (centreline)	**Enclosed Intensity (mean)**
Length Difference	Enclosed Intensity (var)
Bearing Difference	**Enclosed Intensity Difference**
Intensity Gradient Difference	**Gap Intensity (mean)**
Projection	**Gap Intensity (var)**
	Length Combined
	Length Difference
	Minimum Gap Separation
	Maximum Gap Separation
	Gap Separation (mean)
	Bearing Difference
	Intensity Gradient Difference (left)
	Intensity Gradient Difference (right)

smoothing function with a large standard deviation. Adding artifical noise to our images before blurring removes very small features such as noise that are present in high resolution images. The improvement was a dramatic decrease in the number of extracted edges, up to 90% less in several images, which meant that SVM could be used to learn Level 1 data without an additional SVM preprocessing stage. Removing this preprocessing stage gives results that can be compared to other algorithms in RAIL which also do not use an additional preprocessing stage. Another advantage is the reduction in misclassification during the SVM preprocessing stage (approximately 14%) so that a more complete road network can be recovered at higher levels.

3.2 Reference Model

RAIL has recently adopted a centreline reference model based on [11] which can assess the learned outputs more correctly by checking that the extracted pairs have edges that do in fact lie opposite each other near the reference model. Previously we used an edge based model which produced a slightly more modest value in assessing the correctness of the outputs.

3.3 Feature Extraction

Additional features have been added to Level 1 and Level 2 (see Table 1) and a relevant subset from each level was selected by using feature subset selection (FSS) methods, which is described in section 3.4. The highlighted entries are the feature subsets that were discovered. These are briefly described below.

3.3.1 Selected Level 1 Features:

- *Pair Width (mean)*: Average distance between edge pair.

- *Enclosed Intensity (mean)*: Average intensity between edge pair.
- *Pair Length (centreline)*: The length of an edge pair as measured from an imaginary centreline.
- *Bearing Difference*: Direction difference between edge pair.
- *Projection*: Binary value that defines if the edges in an edge pair are opposite.

Pair width, enclosed intensity (mean), bearing and projection form an intuitive feature subset that describes road segments, i.e. roads have similar width and intensity and their opposite sides are almost parallel. Pair length is a good feature because in our preprocessing stage we have set a maximum length for edges. Generally road sides are long and continuous and get split into smaller segments after preprocessing. When road pairs are formed their lengths do not vary too much. This is because non-road edges are usually of shorter length.

Enclosed intensity variance did not prove to be a good feature since the area enclosed by an edge pair is small and the intensity is fairly similar. Length difference between edges was also discarded by FSS. We expect road pairs to have similar edge length but non-road pairs maybe also have similar edge lengths, thus it does not convery much information. Intensity gradient difference between the two edges do not show consistencies between road pairs and non-road pairs. The assumption that the intensity levels are the same on both the external side of the road is invalid.

3.3.2 Selected Level 2 Features:

- *Enclosed Intensity (mean)*: Average intensity of two road pairs.
- *Enclosed Intensity Difference*: Intensity difference of two road pairs, the average intensity for each edge pair is taken.
- *Gap Intensity (mean)*: Average intensity of the gap bridging the two road pairs.
- *Gap Intensity (var)*: Intensity variance of the gap bridging the two road pairs.
- *Minimum Gap Separation*: Minimum separation between two road pairs.
- *Maximum Gap Separation*: Maximum separation between two road pairs.
- *Gap Separation (mean)*: Average separation between two road pairs.
- *Bearing Difference*: Direction difference between two road pairs, the centreline for each edge pair is taken.

Linked road pairs should have similar enclosed intensity with little difference. Ideally linked pairs should be minimally separated and have no gap, thus gap intensity and gap separation are execellent features to distinguish between linked road pairs and other linked edge pairs. Roads generally have smooth curves except at an intersection, therefore the bearing difference between linked road pairs should not be very large.

Width features are not good attributes for Level 2 because Level 1 outputs all have similar widths. The same arguement applies to length attributes. Enclosed intensity variance and gap intensity variance are not very good features for the same reason discussed earlier, i.e. intensity level do not change much in enclosed edge pair or in a road gap. Again, intensity levels across edges cannot be assumed to be the same on both sides of the linked edge pairs.

3.4 Feature Subset Selection

The goal of FSS is to choose the most relevant features for classification, in other words, removing irrelevant attributes that may distract a machine learning algorithm. We compiled 9 sets of data from our images. 7 were from individual images and 2 were random selections from all the images. The sample size ranges from 130 to 360 examples in each set. We did not use one large test set since we had different road types and having one set of data might cause the result to be biased towards the most frequent road type.

The Weka[1] data mining suite (version 3.4) was used to conduct the FSS experiments. The FSS algorithms used fall into two catagories.

3.4.1 Type I:

- *Correlation-based Feature Selection*: Selects a subset by considering the individual predictive ability of each feature along with the degree of redundancy between them.
- *Classifier*: Selects a subset by using a classifier to estimate the "merit" of a set of attributes. The classifiers chosen were the Decision Table, Decision Tree and Naive Bayes.
- *Wrapper*: Evaluates attribute sets by using a learning scheme. The classifiers chosen were the Decision Table, Decision Tree and Naive Bayes.

3.4.2 Type II:

- *Chi Squared*: Evaluates each attribute by measuring the chi-squared statistic with respect to the class.
- *Relief*: Evaluates each attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class.
- *Information Gain*: Evaluates each attribute by measuring the information gain with respect to the class.
- *Gain Ratio*: Evaluates each attribute by measuring the gain ratio with respect to the class.
- *Symmetrical Uncertainty*: Evaluates each attribute by measuring the symmetrical uncertainty with respect to the class.

[1]Software available at `http://www.cs.waikato.ac.nz/ml/weka`

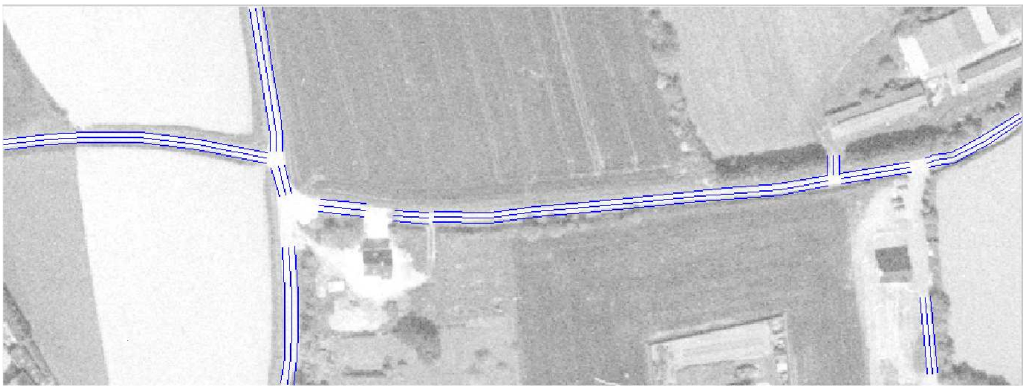

Figure 2: Image A

3.4.3 Selection Process:

Type I algorithms were run using three different search methods, they are best first, rank and genetic search. A total of 21 algorithm-search combinations were used. Type I algorithms select the 'best' subset of features. The frequency of each attribute was recorded and averaged. Type II algorithms rank the individual attributes by assigning them a weighting. These were normalised and averaged.

We wanted to select a subset of features which have a high frequency score in Type I and a high weighting in Type II. We ranked the Type I and Type II results and picked the smallest subset where the features are the same in each type. For example, if the top 4 attributes in Type I and Type II are the same disregarding their relative ranking position, then we would have a subset of 4 features. This has produced good classification results.

3.5 Features Versus Heuristic Preprocessing

Although we are using new image processing parameters to produce less noisy outputs, we are still dealing with fairly large datasets for Level 1 (C_2^n, as each edge can be paired up with every other edge and the ordering is irrelevant). Thus we use heuristic preprocessing to reduce the data size so that it becomes more manageable. We do not use heuristic rules for Level 2 since the data size is comparatively smaller than Level 1.

The heuristic rules throw away cases where an expert would agree that a positive classification is impossible. For example, in Level 1 we used the heuristic that if edges in an edge pair do not project onto each other, then they cannot be classified as an edge pair, since they are not opposite each other. Because this feature has a binary output, by using this attribute as a heuristic filter we have effectively removed projection from the feature space, since the heuristic rule outputs only those edge pairs that do project on to each other. We also have a heuristic rule that leaves out any edge pairs that are wider than twice the maximum road width in the images. We have effectively reduced the feature space that SVM would need to learn from.

Theoretically this should not make any difference to machine learning algorithms be-

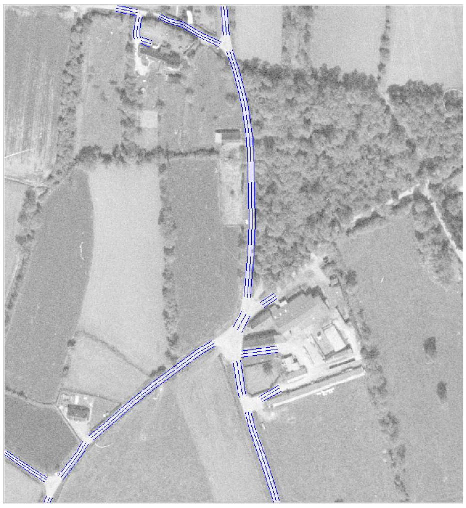

Figure 3: Image B

Figure 4: Image C

Figure 5: Image D

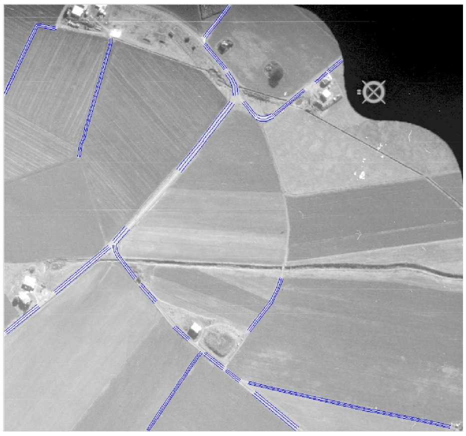

Figure 6: Image E

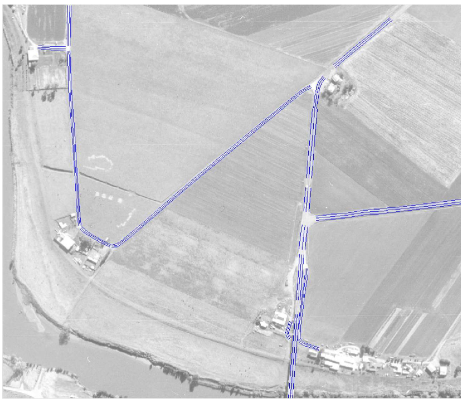

Figure 7: Image F

Figure 8: Image G

Table 2: Image Properties

Image	Dimensions	No. of Edges
A	776*289	1530
B	757*821	3055
C	1500*848	2912
D	1700*1300	3290
E	1400*1300	1858
F	1400*1200	3893
G	1600*1100	3204

cause the data we are leaving out have no influence on how the classes are separated. For SVMs, the data points discarded are distant from the class separation region and the support vectors, thus the construction of the separation hyperplane is independent of them.

3.6 Dataset

Seven high resolution aerial images were used in the experiment. Fig. 2 and Fig. 3 are of a rural area in France. These images have a ground resolution of 0.45m/pixel. The other five images (Fig. 4 to Fig. 8) are of a rural area in Morpeth in Australia. These images have a ground resolution of 0.6m/pixel. The centreline reference is shown on the images. The image properties are give in Table 2.

A total of 333 and 227 positive and negative examples were selected from the images (some images contain more examples) for Level 1 and Level 2 respectively. The test data for Level 1 are the heuristic outputs. For Level 2 we do not use heuristics so we end up with C_2^n twin linked pairs. The size of the test data ranges from 2400 to 11200 instances for Level 1 and between 1500 to 18200 for Level 2.

Since we only had seven images to experiment with, we used 7-fold cross validation technique (leave-one-out) for evaluating the learned output, i.e. we train using six images and test on the unseen image. Note however that at the edge pair and twin linked edge pair level, we have thousands of instances in each image.

4 EXPERIMENTAL DESIGN

SVM experiments have been conducted on Level 1 and Level 2 of RAIL (the level references are different to those in [12]). The SVM implementation used was changed to LIBSVM [2] (version 2.4) which offers more in terms of tools and programming interfaces.

The training data and test data were scaled to [-1,1] to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during SVM calculations [3].

We used five different kernels for training SVMs for Support Vector Classification (C-SVC). They can be separated into two categories: Polynomial and Radial Basis Function (RBF). The polynomial kernels are of the first, second and third degree (with default

[2]Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Table 3: Level 1 Classification

Image	**Classifier**	**Comp.**	**Corr.**	**cxc**
A	SVM Poly. 1^{st}	101	32	34
A	SVM Poly. 2^{nd}	100	32	33
A	**SVM Poly. 3^{rd}**	**101**	**37**	**38**
A	SVM RBF std.	101	35	36
A	SVM RBF opt.	101	35	36
A	Decision Tree	100	31	31
A	Neural Network	101	36	37
B	SVM Poly. 1^{st}	107	34	39
B	SVM Poly. 2^{nd}	96	18	17
B	SVM Poly. 3^{rd}	94	39	34
B	SVM RBF std.	103	36	38
B	**SVM RBF opt.**	**108**	**36**	**42**
B	Decision Tree	102	29	30
B	Neural Network	107	39	44
C	SVM Poly. 1^{st}	92	26	23
C	SVM Poly. 2^{nd}	89	21	17
C	SVM Poly. 3^{rd}	87	33	25
C	SVM RBF std.	94	29	25
C	**SVM RBF opt.**	**94**	**30**	**27**
C	Decision Tree	92	29	25
C	Neural Network	92	31	26
D	SVM Poly. 1^{st}	100	22	21
D	SVM Poly. 2^{nd}	98	23	23
D	SVM Poly. 3^{rd}	97	23	22
D	SVM RBF std.	99	23	22
D	**SVM RBF opt.**	**99**	**24**	**24**
D	Decision Tree	97	27	26
D	Neural Network	98	26	25
E	SVM Poly. 1^{st}	85	47	34
E	SVM Poly. 2^{nd}	80	43	28
E	SVM Poly. 3^{rd}	81	56	37
E	**SVM RBF std.**	**91**	**56**	**46**
E	SVM RBF opt.	87	57	43
E	Decision Tree	37	41	6
E	Neural Network	51	55	15
F	SVM Poly. 1^{st}	83	32	22
F	SVM Poly. 2^{nd}	91	27	22
F	**SVM Poly. 3^{rd}**	**88**	**37**	**29**
F	SVM RBF std.	88	35	27
F	SVM RBF opt.	84	33	24
F	Decision Tree	70	36	18
F	Neural Network	73	43	23
G	SVM Poly. 1^{st}	98	35	34
G	SVM Poly. 2^{nd}	97	30	28
G	**SVM Poly. 3^{rd}**	**96**	**39**	**36**
G	SVM RBF std.	100	38	38
G	SVM RBF opt.	100	38	37
G	Decision Tree	92	31	27
G	Neural Network	95	34	30

C=1). The RBF kernels are standard (C=1, γ=1) and optimised (C, γ picked by a grid search function provided by LIBSVM). C is the penalty parameter that controls the margin and hence the overfitting of data, and γ is an internal variable for RBF.

The SVM kernels are compared to two well known classifiers, decision tree (DT) and neural network (NN). Both algorithms are implemented in Weka and the default settings for each are used. The DT uses a confidence factor of 0.25 and performs pruning. For NN, 3 hidden layers are used for Level 1 and 5 hidden layers are used for Level 2. The setting for learning rate and momentum is 0.3 and 0.2 respectively.

5 EXPERIMENTAL RESULTS

The metrics used to evaluate the results are taken from [11]. They address two questions: 1) How complete is the extracted road network, and 2) How correct is the classification. They are calculated to percentage values, given by:

$$completeness = \frac{length_{TP}}{length_{reference}} \tag{1}$$

$$correctness = \frac{length_{TP}}{length_{classifed}} \tag{2}$$

Completeness measures the percentage of the road reference as detected by SVM. Correctness measures the percentage of the SVM classification that are actual road pairs. A high completeness means that SVM has extracted most of the road network, whereas high correctness implies that SVM has not classified too many incorrect road pairs.

We combine the two measures above into a more general measure of the quality. We call this *cxc* which is expressed as:

$$cxc = completeness^2 * correctness \tag{3}$$

Clearly, this measure is biased towards completeness. RAIL uses the output of Level 1 as the input of Level 2, so it is more important to have high completeness at the lower levels for input to higher levels. For example, Level 2 will only be as complete as its input (Level 1 output). Higher correctness value will result as higher levels discard non-road pairs.

Tables 3 and 4 shows the SVM results (rounded to nearest percent) for Level 1 and Level 2 respectively. The entry with the highest *cxc* for each image in Level 1 is used as input to Level 2. The highest *cxc* obtained by SVM classifer has been highlighted for each image. Fig. 9 to Fig. 29 show the results visually. The images consist of low-level edges as input and the best Level 1 and Level 2 outputs.

Some of the completeness values are a little over 100%, this is because the centreline reference model uses a buffer zone both to the left and to the right of the road reference. Although the buffer width is only set to 3 pixels on either side, on some noisy road sections, two or more edges maybe measured as true positives for that same section. However, this is only true in a few cases. In all images with completeness greater than 100%, detailed analysis show that more than 98% of the reference road network is recognised.

Table 4: Level 2 Classification

Image	**Kernel**	**Comp.**	**Corr.**	**cxc**
A	SVM Poly. 1^{st}	98	53	50
A	SVM Poly. 2^{nd}	98	54	51
A	SVM Poly. 3^{rd}	98	53	51
A	SVM RBF std.	98	54	51
A	**SVM RBF opt.**	**98**	**54**	**52**
A	Decision Tree	98	54	51
A	Neural Network	98	54	51
B	SVM Poly. 1^{st}	105	51	57
B	SVM Poly. 2^{nd}	105	51	58
B	**SVM Poly. 3^{rd}**	**105**	**54**	**60**
B	SVM RBF std.	105	52	57
B	SVM RBF opt.	105	53	59
B	Decision Tree	105	52	57
B	Neural Network	105	51	56
C	SVM Poly. 1^{st}	81	41	27
C	SVM Poly. 2^{nd}	81	41	27
C	SVM Poly. 3^{rd}	81	41	27
C	SVM RBF std.	81	41	27
C	**SVM RBF opt.**	**81**	**42**	**27**
C	Decision Tree	81	41	27
C	Neural Network	81	41	27
D	SVM Poly. 1^{st}	98	30	28
D	SVM Poly. 2^{nd}	98	30	29
D	**SVM Poly. 3^{rd}**	**98**	**30**	**29**
D	SVM RBF std.	98	30	28
D	SVM RBF opt.	98	30	29
D	Decision Tree	98	30	29
D	Neural Network	98	30	28
E	SVM Poly. 1^{st}	84	70	49
E	SVM Poly. 2^{nd}	84	69	48
E	SVM Poly. 3^{rd}	84	66	46
E	**SVM RBF std.**	**84**	**70**	**49**
E	SVM RBF opt.	84	70	49
E	Decision Tree	84	70	49
E	Neural Network	84	70	49
F	SVM Poly. 1^{st}	68	54	25
F	SVM Poly. 2^{nd}	68	55	25
F	**SVM Poly. 3^{rd}**	**68**	**55**	**25**
F	SVM RBF std.	68	55	25
F	SVM RBF opt.	67	55	25
F	Decision Tree	67	55	25
F	Neural Network	68	54	25
G	SVM Poly. 1^{st}	99	42	40
G	SVM Poly. 2^{nd}	98	43	40
G	**SVM Poly. 3^{rd}**	**99**	**43**	**41**
G	SVM RBF std.	99	42	40
G	SVM RBF opt.	99	42	41
G	Decision Tree	96	42	39
G	Neural Network	99	42	41

Level 1 SVM classifiers have an average of 97% completeness and 35% correctness. Level 2 SVM classifiers have an average of 90% completeness and 49% correctness. These results are very encouraging because high completeness values are obtained. However, there does not appear to be a clear pattern as to which kernel function consistently outperforms the others in both levels. The SVM classifiers compare well to DT and NN classifiers. In most cases, the results are very similar. However, on images containing dirt roads in Level 1 (Image E and F), SVM classifiers seemed to outperform both DT and NN, see Table 3. For Level 2, the classifiers achieve very similar levels in terms of their performance.

The low correctness value in Level 1 does not worry us. One of the major causes of the large number of false positives is that SVM classified road pairs have similar road properties, but only a fraction of them actually fall into the category of roads, as represented by the centerline road reference. The others fall into categories such as driveways and crop separations (perhaps for tractors) which are non-roads, but picked up well by the classifiers. Fig. 19 shows a good example of this problem as many classified road pairs are crop separations. The other main reason is that road properties may vary slightly between different images. SVMs learn these variations to a certain degree and thus the classified output may contain a range of road properties, some of which might be non-roads depending on the images.

Some images had lower completeness in Level 2, particularly Images C, E and F. The main causes of this are, 1) because the road is very similar to its surroundings (especially roads with lower intensity), which means edges are not extracted well, and 2) dirt roads have been misclassified in Level 2 since the edge pairs are not closely linked. Fig. 26 is a good example where narrower roads with high intensity have been detected while wider and lower intensity roads have been missed. This problem can be fixed by applying a further preprocessing stage before edge extraction, e.g. multilevel thresholding/segmentation or by using an emsemble of SVMs and combining the results.

We observe that Level 2 completeness can only be as high as its input. We also observe that the correctness has increased as expected of higher levels.

6 CONCLUSIONS

In this paper we have experimented with SVM and road extraction, which is significantly different from other SVM experiments in the remote sensing domain. The results for Level 1 and Level 2 are very encouraging and comparable to clustering algorithms also used in RAIL. We plan to extend SVM to level 3 of RAIL which currently uses a relational learning algorithm to recognise the attributes of junctions [9].

In the future we plan to be able to experiment with other kernel functions and apply machine learning to find the best kernel and the parameters that are associated with them [6]. We also plan to apply meta-learning techniques as we gather more supervised and unsupervised machine learning algorithms to the problem of road extraction.

Figure 9: Image A - Input

Figure 10: Image A - Level 1 output

Figure 11: Image A - Level 2 output

References

[1] Byun, H., Lee, S., 2003. A survey on pattern recognition applications of support vector machines. *IJPRAI*, 17(3), pp. 459-486.

[2] Camps-Valls, G., Gomez-Chova, L., Calpe-Maravilla, J. Soria-Olivas, E., Martin-Guerrero, J. D., Moreno, J., 2003. Support Vector Machines for Crop Classification Using Hyperspectral Data. *Proc. IBPRIA 2003*, pp. 134-141.

[3] Chang, C., Lin, C., 2001. LIBSVM: A library for support vector machines. Available at `http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf` (accessed 28 Feb. 2004).

[4] Chen, A., Donovan, G., Sowmya, A., Trinder, J., 2002a. Inductive Clustering: Automating low level segmentation in high resolution images. *Proc. ISPRS Commission III Symp. Photogrammetric Computer Vision*, pp. 73-78.

[5] Chen, A., Donovan, G., Sowmya, A., 2002b. Recent Progress on RAIL: automating clustering and comparison of multiple classification techniques on high resolution remotely sensed imagery. *Proc ICML-2002 Workshop Machine Learning in Computer Vision*, pp. 1-8.

[6] Gold, C., Sollich, P., 2003. Model selection for support vector machine classification. *Neurocomputing*, pp. 221-249.

[7] Mena, J. B., 2003. State of the art on automatic road extraction for GIS update: A novel classification. *Pattern Recognition Letters*, 24(16), pp. 3037-3058.

[8] Pal, M., Mather, P. M., 2004. Assessment of the effectiveness of support vector machines for hyperspectral data. *Future Generation Computer Systems*, Article in Press.

[9] Nguyen, H. S., Sowmya, A., Trinder, J., 2004. Improved Road Junction Recognition in High Resolution Images using Relational Learning. To appear in *ISPRS World Congress*, Istanbul.

[10] Sowmya, A., Singh, S., 1999. RAIL: Extracting road segments from aerial images using machine learning. *Proc. ICML 99 Workshop on Learning in Vision*, pp. 8-19.

[11] Wiedemann, C., Heipke, C., Mayer, H., Jamet, O., 1998. Empirical Evaluation of Automatically Extracted Road Axes. In *CVPR Workshop on Empirical Evaluation Methods in Computer Vision*, pp. 172-187.

[12] Yager, N., Sowmya, A., 2003. Support Vector Machines for Road Extraction from Remotely Sensed Images. *Proc. CAIP 2003*, pp. 285-292.

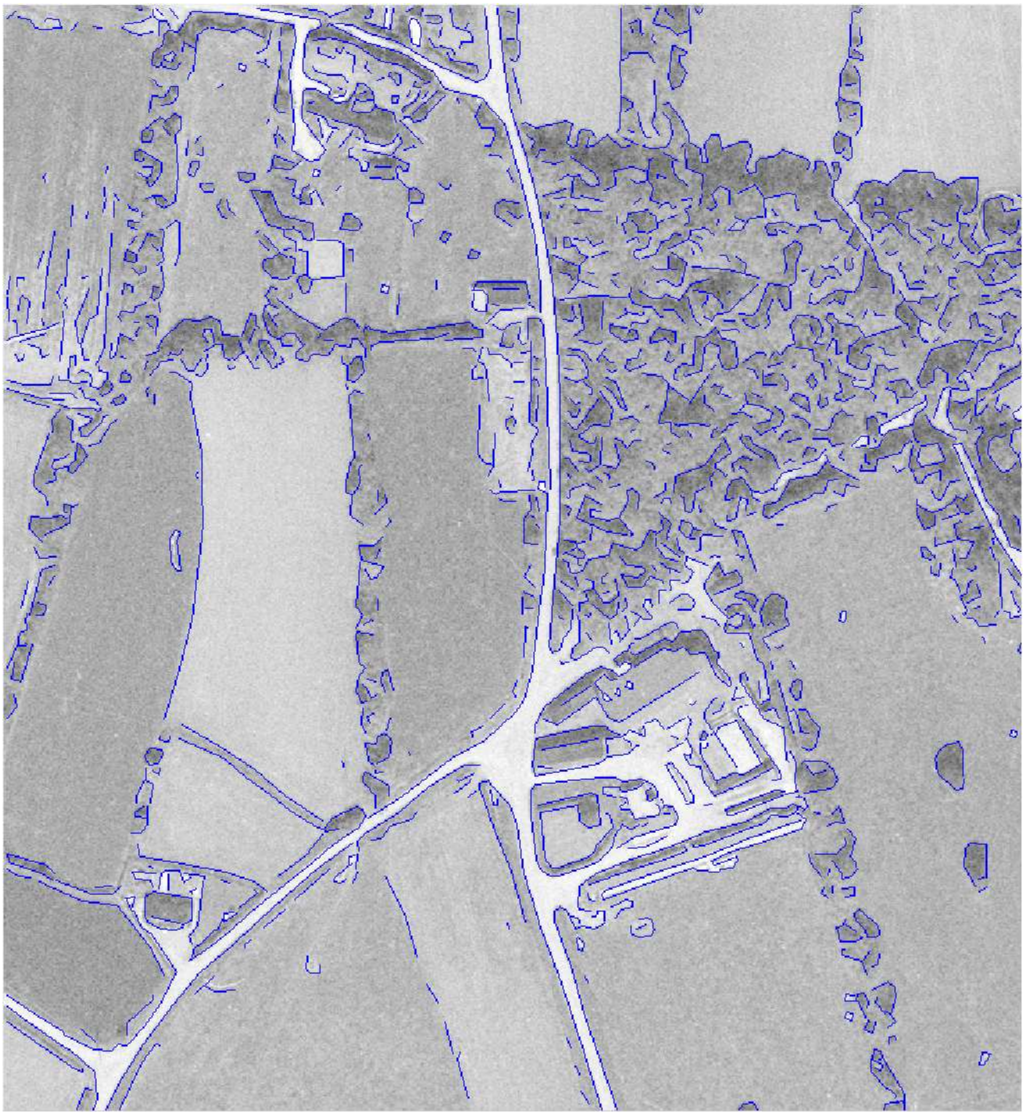

Figure 12: Image B - Input

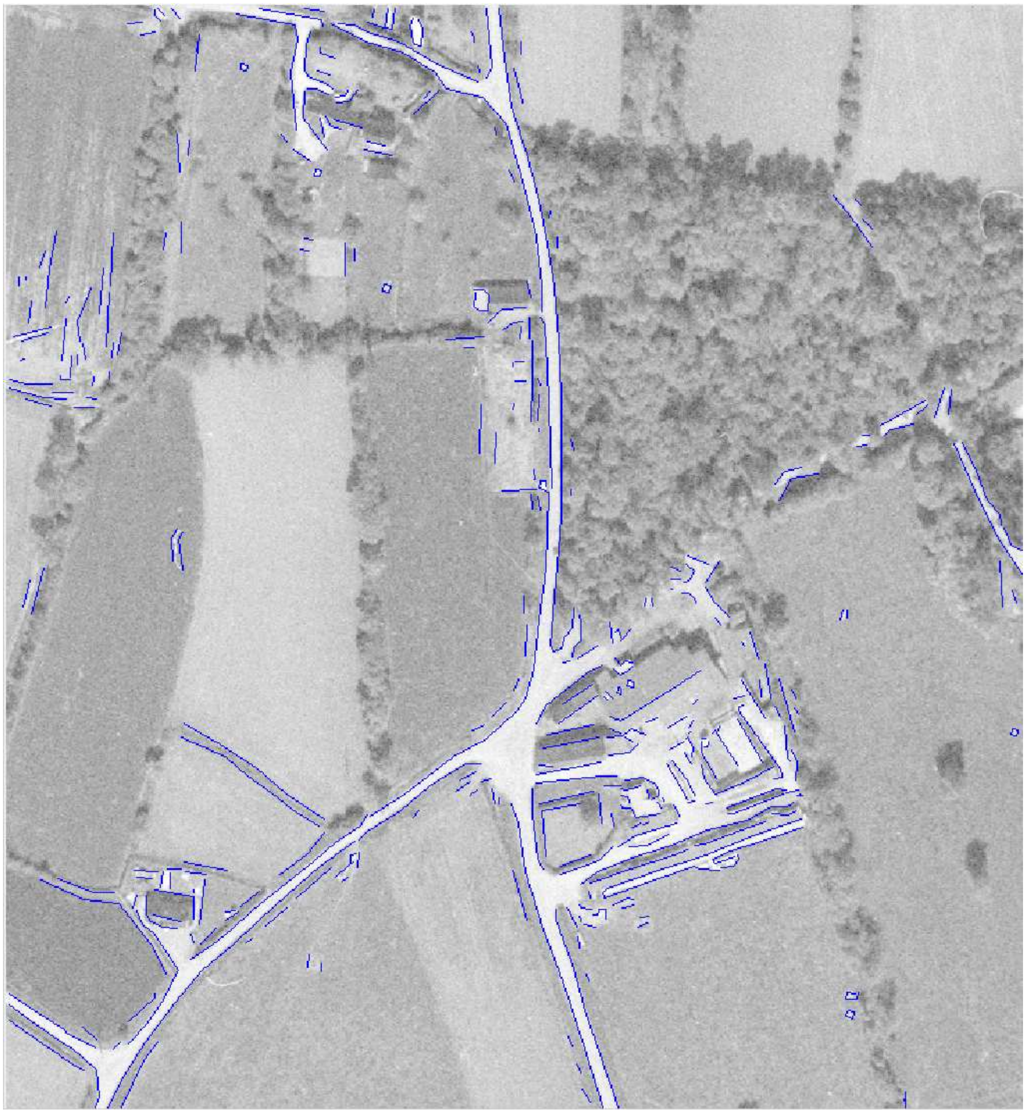

Figure 13: Image B - Level 1 output

Figure 14: Image B - Level 2 output

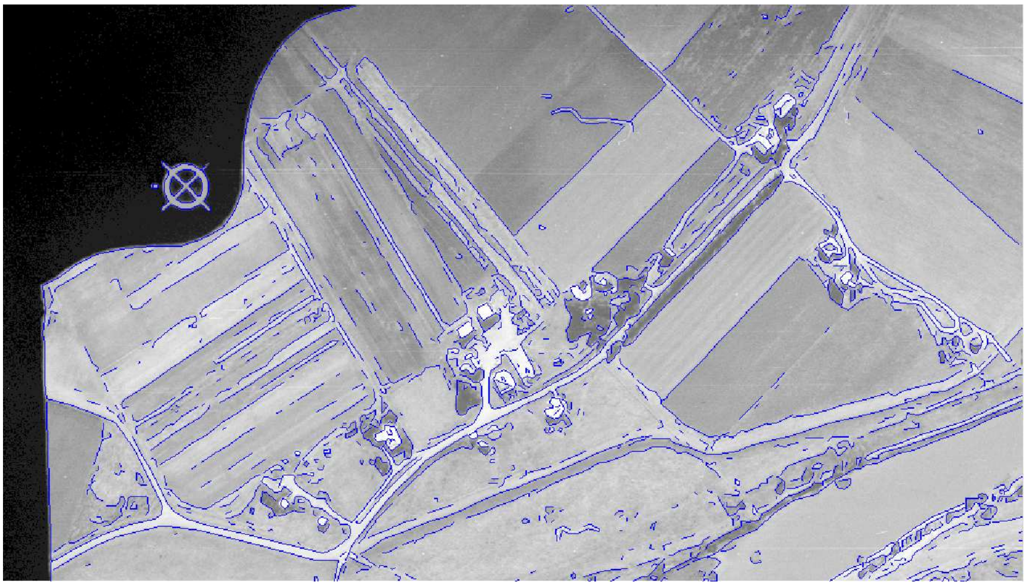

Figure 15: Image C - Input

Figure 16: Image C - Level 1 output

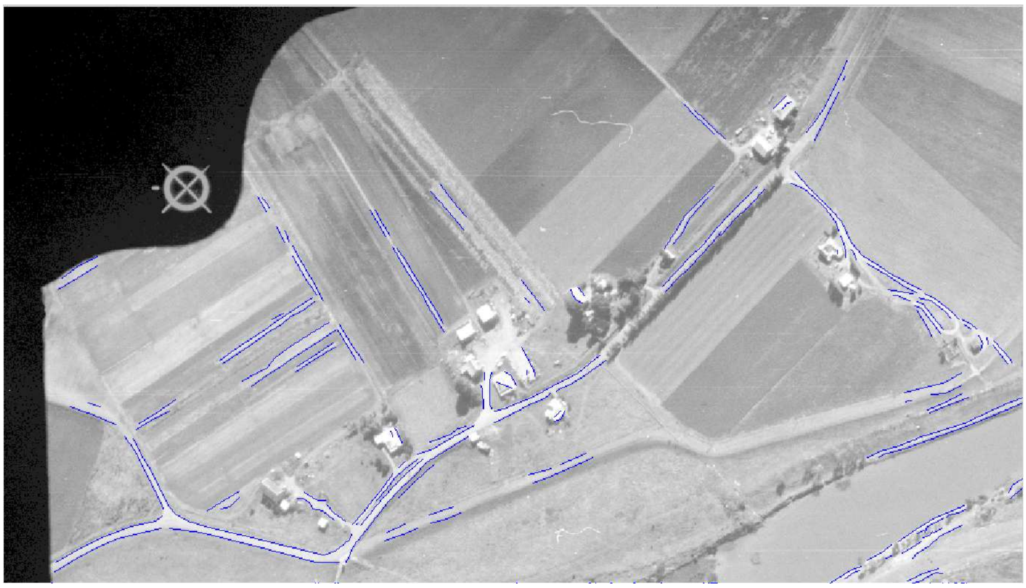

Figure 17: Image C - Level 2 output

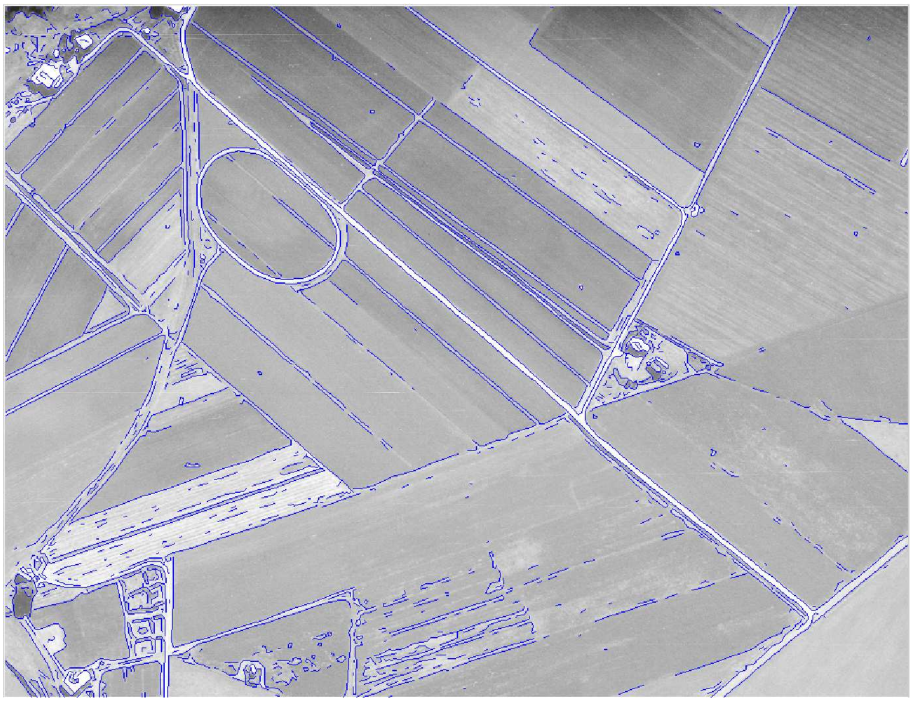

Figure 18: Image D - Input

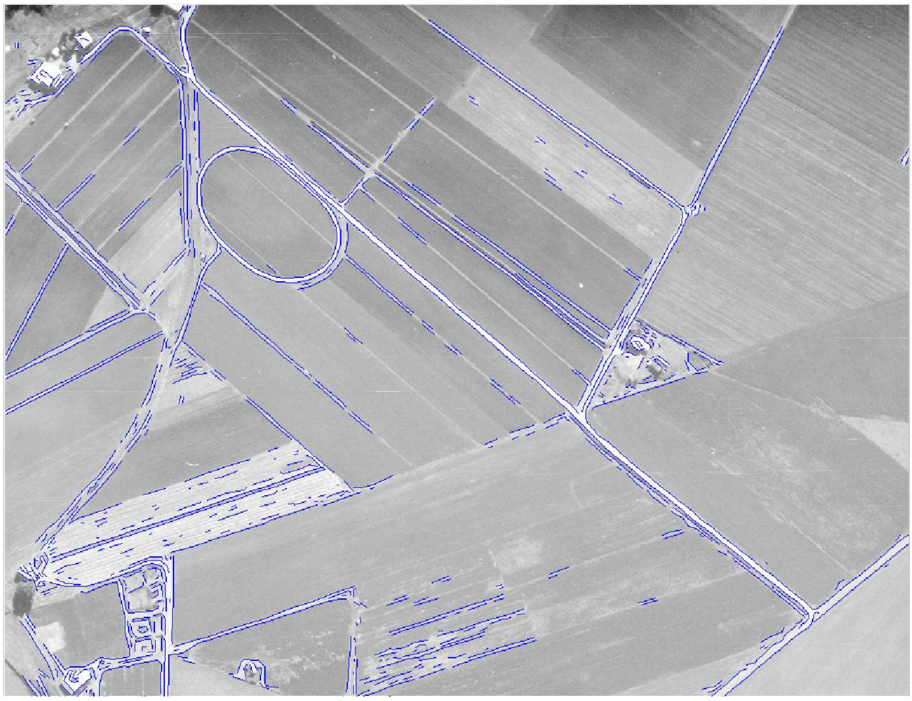

Figure 19: Image D - Level 1 output

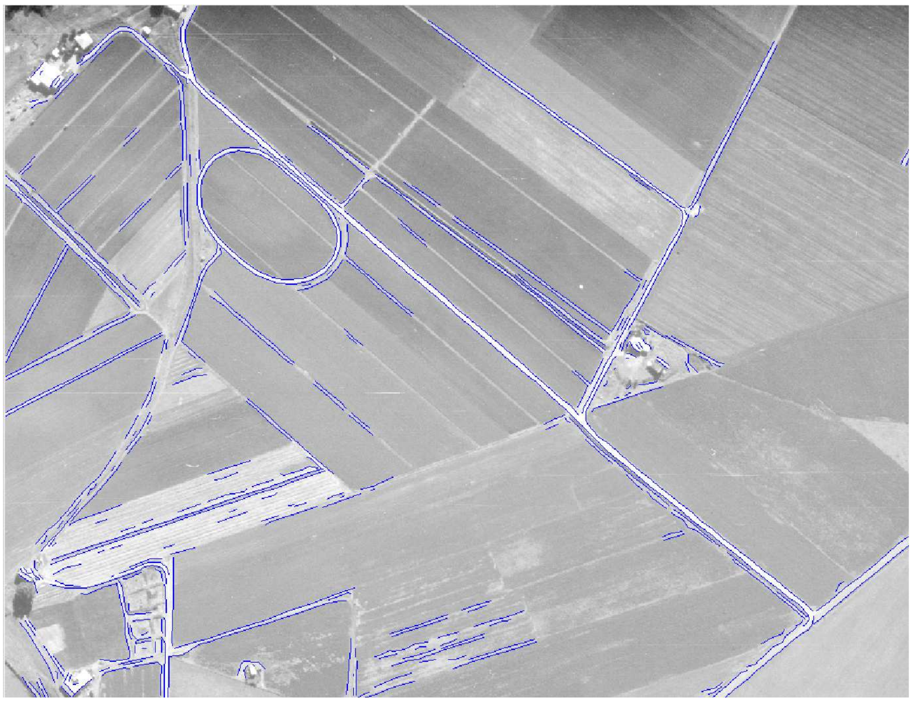

Figure 20: Image D - Level 2 output

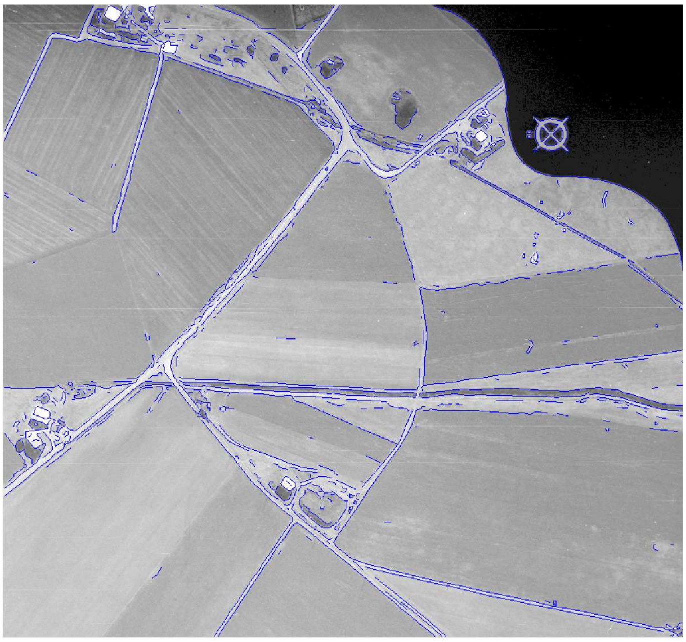

Figure 21: Image E - Input

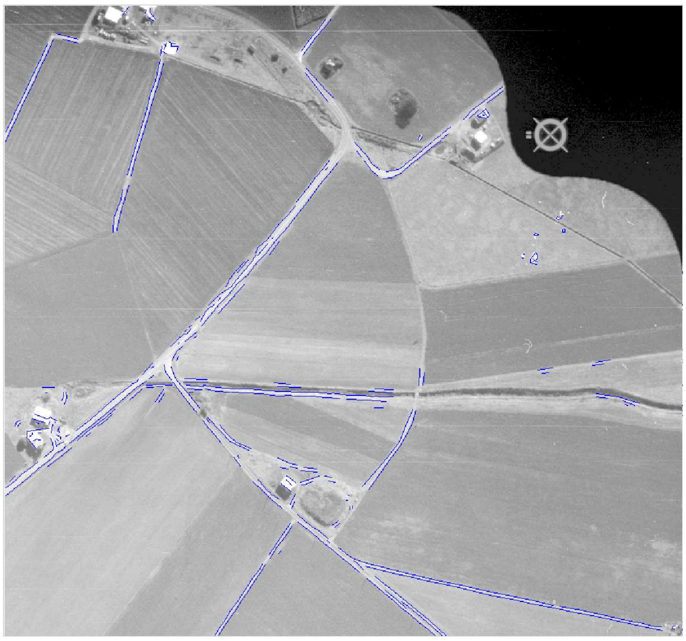

Figure 22: Image E - Level 1 output

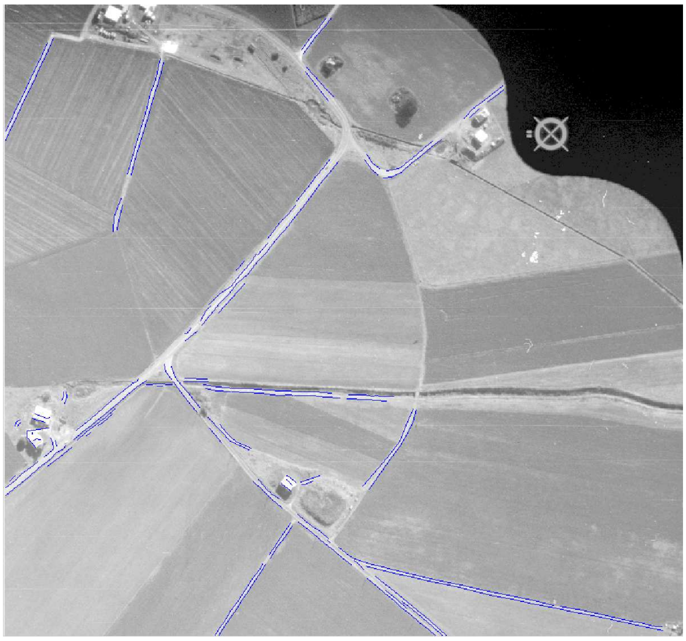

Figure 23: Image E - Level 2 output

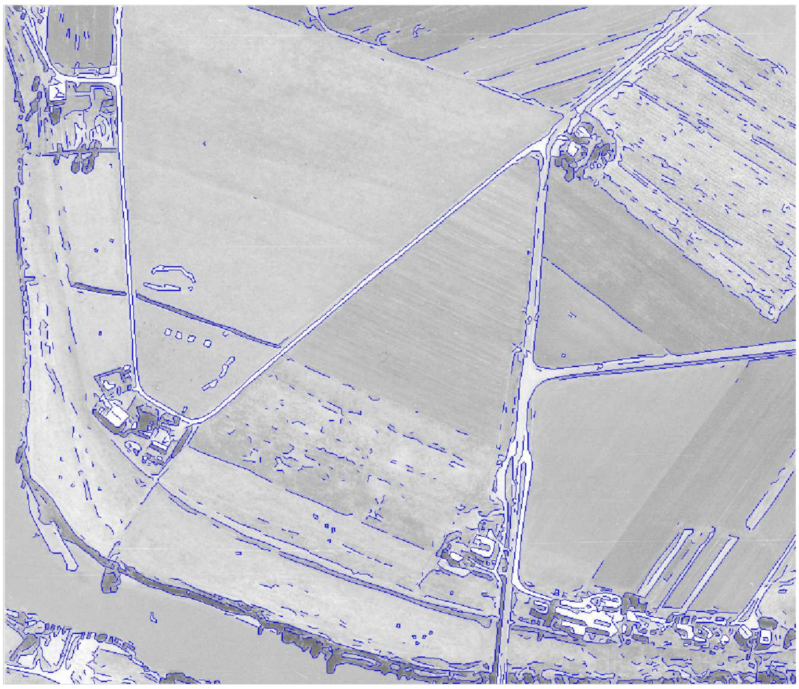

Figure 24: Image F - Input

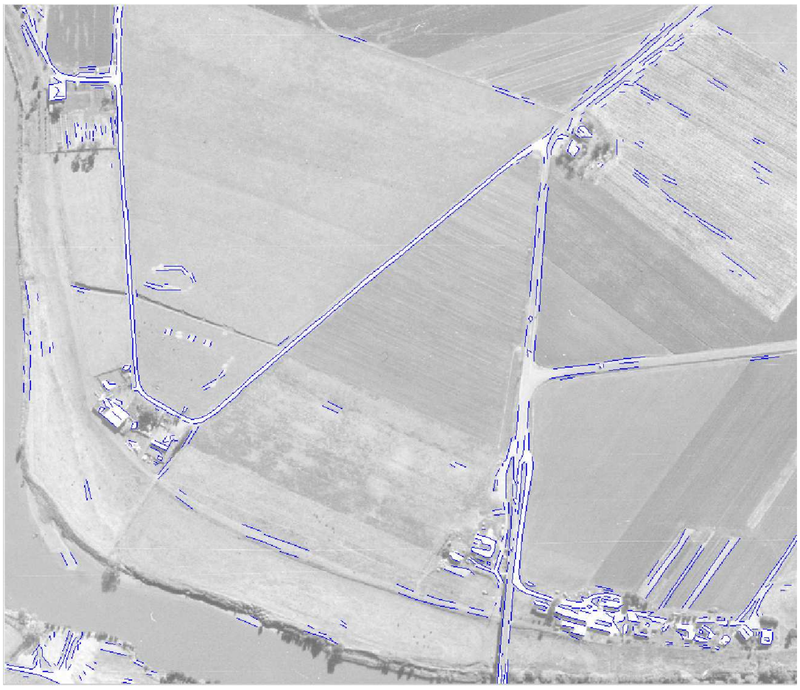

Figure 25: Image F - Level 1 output

Figure 26: Image F - Level 2 output

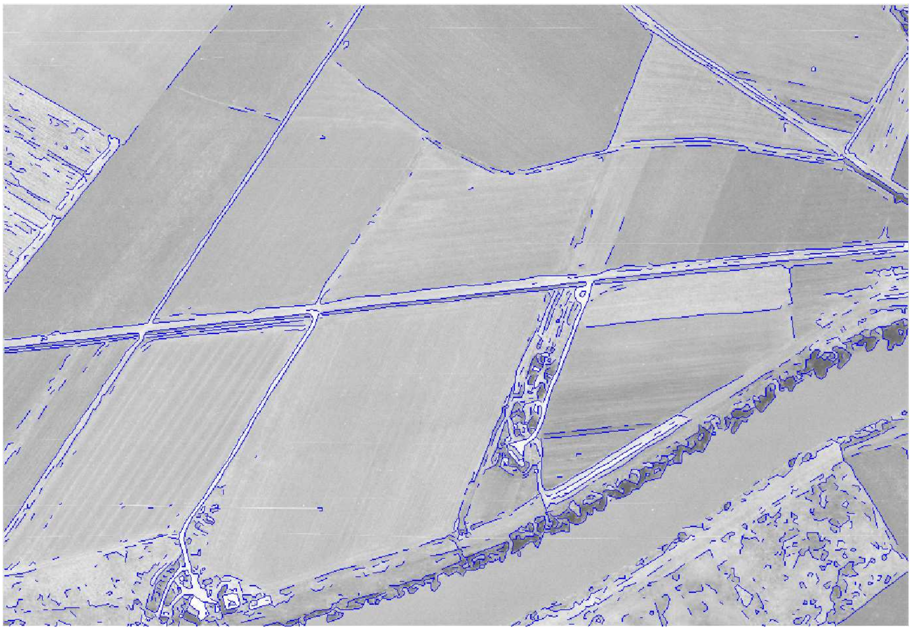

Figure 27: Image G - Input

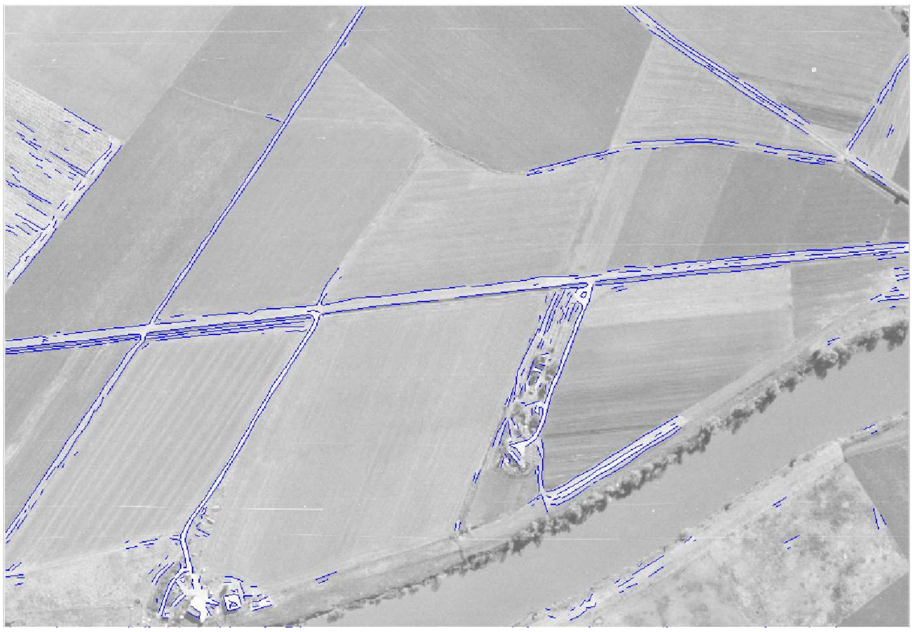

Figure 28: Image G - Level 1 output

Figure 29: Image G - Level 2 output