Recent Progress on RAIL: Automating Clustering and Comparison of Different Road Classification Techniques on High Resolution Remotely Sensed Imagery

Annie Chen ANNIEC@CSE.UNSW.EDU.AU
Gary Donovan GARYD@CSE.UNSW.EDU.AU
Arcot Sowmya SOWMYA@CSE.UNSW.EDU.AU
School of Computer Science and Engineering, University of New South Wales, Sydney 2052, Australia

Abstract

In this paper we present recent developments in RAIL, a hierarchical road recognition system for high resolution images. We shall introduce a novel classification technique for segmenting remotely sensed images, based on cluster analysis and machine learning. Traditional segmentation techniques which use clustering require human interaction to fine-tune the clustering algorithm parameters and select good clusters. Our technique applies inductive learning techniques using C4.5 to learn the parameters and pick good clusters automatically. We will present the clustering results using this technique along with other classification algorithms implemented for level 1 of RAIL.

1. Introduction

Road detection and recognition from remotely sensed imagery is an important process in the acquisition and update of Geographical Information Systems. A great deal of effort has been put into the development of automated feature extraction methods, mainly in the areas of expert systems and advanced image analysis. In our previous papers we described the RAIL/Recoil system (Singh, 1998; Sowmya, 1999; Trinder, 1999; Teoh, 2000a; Teoh, 2000b), which is a semi-automatic, multi-level, adaptive and trainable edge-based road recognition system intended to demonstrate the use of various Artificial Intelligence approaches in this area.

Our goal is to implement several Artificial Intelligence algorithms to extract roads from remotely sensed images, thus demonstrating their viability for such problems. The system needs to exploit the relative strengths and weaknesses of different algorithms, and we have recently developed a new inductive clustering framework to help with selecting the right algorithm for a given image (Chen 2002a). This uses inductive learning to improve the results obtained from clustering, by learning the optimal clustering parameters for any situation. We have also compared the result of the three different classification algorithms (Chen, 2002b) implemented in Level 1 of RAIL. For this workshop we will summarise this recent progress in RAIL.

In section 2 we give the background of RAIL. Section 3 explains the different classification techniques used in RAIL, whilst section 4 discusses the inductive clustering framework in detail. Finally, section 5 presents an evaluation of our techniques.

2. Background of RAIL

Our research builds upon the existing RAIL (**R**oad **R**ecognition from **A**erial **I**mages using **I**nductive **L**earning) software, which is a road recognition system we are developing. Our previous papers give more details.

Instead of *a priori* rules, RAIL uses supervised multi-level learning to *derive rules* that may be applied during attribute extraction and recognition. The techniques are general, making few assumptions, and are applicable to images of different scales, content, complexity and quality. Starting with edges, complex structures are built from simpler ones in multiple stages, beginning with image pre-processing and edge extraction. The road detection is split into four levels, covering road segment detection, road segment linking, intersection detection and then connecting roads to intersections.

Inductive learning and clustering techniques are used to recognise each of the road structures at the four levels, using information from similar images. The objects from one level become the input to the next level, until finally road-like objects are output from level 4.

All the methods used (that is, inductive learning, kNN and KMeans clustering) have various disadvantages. We have attempted to overcome these problems by combining the methods, in a process we call Inductive Clustering (described in Section 4). Amongst other improvements, this quickly reduces the size of the data set, shortening processing time and allowing more automation.

Our results only describe Level 1 of RAIL, since other levels have not been fully implemented yet. In level 1, we aim to join edges (produced by the Canny operator) into edge pairs, or road segments. A road segment is a pair of edges which are part of a road, and are opposite each other. The attributes, or properties, of such a road segment are:

- **Enclosed Intensity** – The average grey-scale intensity inside road is generally high.
- **Parallel Separation –** The average distance between edges usually falls within a certain range.
- **Difference in spatial direction between edges** – Roads appear as pairs of parallel boundaries.
- **Difference in intensity gradient direction –** Road boundaries often have opposite intensity gradients.
- **Intensity difference** – A road typically appears brighter than its surroundings.

Initial clustering tests with our data showed that the last 3 attributes do not usefully distinguish between different road segments. This can have detrimental affects on the clustering, since clusters tend to distribute themselves along every dimension (or attribute). Hence these attributes were not used in further clustering experiments, but were still kept for the inductive learner.

3. Classification Techniques

We have implemented three classifiers in RAIL, these being an inductive learner, KMeans clustering and kNN clustering. In this section we will briefly discuss these classifiers, and how each of them is applied in RAIL. For more details, please refer to our previous papers.

3.1 Inductive Learning

We use an inductive learner, C4.5 (Quinlan, 1996), to calculate the thresholds used for selecting edge pairs that match road segments. Traditionally these thresholds would be determined by human experts, but inductive learning can provide a more customized result.

3.2 Clustering

Clustering is an automated technique that involves sorting a set of data into groups, based on attributes of that data (Weiss, 1991). In RAIL's level 1 road extraction, clustering is used to create groups of edge pairs that have similar shape, intensity, etc., some of which presumably form part of a road.

The data set in RAIL is made up of points described by level 1 attributes, where each point represents an edge pair. This data set is to be partitioned into *n* clusters. Since the best value of *n* is different for every image, several values have to be tried and tested to obtain the best result.

KMeans groups a new data point to its closest cluster as measured by the cluster centre. kNN looks at the *k* nearest neighbours (i.e. the closest points from existing clusters), and the data point is placed in the cluster containing the most neighbours.

A large number of experiments need to be run with different parameters in order to find the setting that produces the best result for a given problem. This whole process requires a lot of hand tuning to find a suitable algorithm, select the associated parameters, and finally pick out the useful clusters. In the next section we will suggest ways to automate this laborious process by applying inductive learning techniques to each of the stages.

4. Inductive Clustering Framework

Our inductive clustering framework has been designed to learn, from cluster descriptions, what constitutes a good road cluster, and to apply the learned knowledge to perform clustering automatically. The ultimate goal is to allow the system to take a new image, and deduce, from the characteristics of the image, the optimal algorithm and the parameters to use. This process will then automatically identify the road cluster for the user.

This framework uses a multi-level learning strategy to tackle the process systematically in the following three stages (see Figure 1):

Parameter Learning (see section 4.1): Learn the parameters that will give the best result for a given algorithm and image type. Parameters include *n* (the number of clusters) and *k*, in kNN clustering.

Algorithm Learning (see section 4.2): Learn which algorithm is most suitable for a given image type. The previous stage determines the parameters to use for each algorithm.

Cluster Learning (see section 4.3): Learn to identify the road clusters by comparing their characteristics to known road and non-road clusters.

Inductive learning methods are used to derive rules at each level. These rules are combined at the end to allow a one-step clustering process for extracting road clusters.

Our system requires a large number of experiments for training at each level. We used reference models (containing hand-picked edges, as shown in Figures 7 and 8) for each data set, so that the evaluation of the results could be done automatically for each experiment.

However, our reference format unfortunately models what the computer detects, not what is actually there in the real world. In consequence, our stated accuracies are mildly optimistic at best, and should be interpreted appropriately.

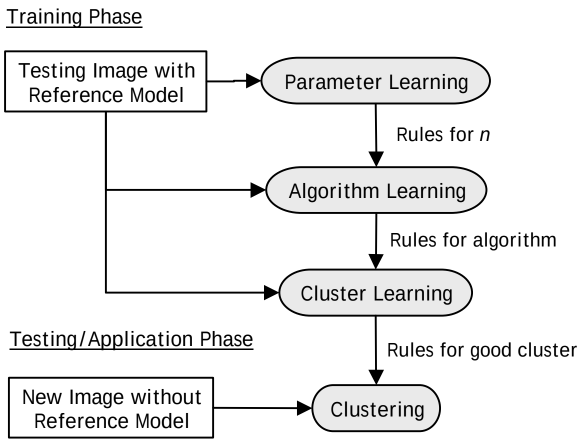


Figure 1. Inductive Clustering Framework Overview.

The measures we use to quantify our results are taken from (Harvey, 1999). They are percentage values, given by:

$$complete = \frac{size_{\text{True Positive Edges}}}{size_{\text{reference model}}} \qquad (1)$$

$$correct = \frac{size_{\text{True Positive Edges}}}{size_{\text{derived}}}$$

where *size* = Number of edges in the relevant edge set.

High completeness means that the cluster has covered the road edges well, whereas high correctness implies that the cluster does not contain many (incorrect) non-road edges. There is usually a trade off between the two measures, since a complete cluster is more likely to contain non-road edges.

It is computationally easier if there is only one criterion to distinguish between clusters. At Level 1 of RAIL completeness is more important than correctness, since we do not want to remove any information too early. A weighted linear combination of the values did not work experimentally, hence our filtering criterion is:

$$cxc = complete^{3} \times correct \qquad (2)$$

Clearly, this measure is biased towards completeness. We also used a threshold (based upon empirical observation) to ensure that the cluster reached a minimum stage of completeness. These two tests (measured out of 1) can be expressed together as:

$$complete \geq 0.8 \text{ and } cxc \geq 0.95 \qquad (3)$$

4.1 Parameter Learning

In the parameter learning stage we want to deduce rules for the value of *n* (the number of clusters) to use on a given algorithm and image.

The attributes that we used to learn clustering parameters are described in Table 1. They include image characteristics, along with the clustering parameters we need to determine. Each set of attributes are classified as either good or bad.

We used inductive learning to classify each setting as containing good or bad road clusters by evaluating the best cluster produced with this setting, using Equation (3).

Table 1. Parameter learning attributes.

NAME	DESCRIPTION	VALUE
Size	Number of edge pairs	Continuous
Algorithm	Clustering algorithm used	KMeans, kNN
n	Number of clusters	[2, 30)
CLASSES	Whether attributes produce a good or bad road cluster	Good, bad

There are two phases in the parameter learning process, as shown in Figure 2. In the training phase, Level 1 RAIL attributes of the given image are calculated and used to cluster with different parameters. The clusters generated are evaluated against the reference model to determine which ones are "good". The inductive learner is then used to generate rules for choosing *n* in other, unseen, images.

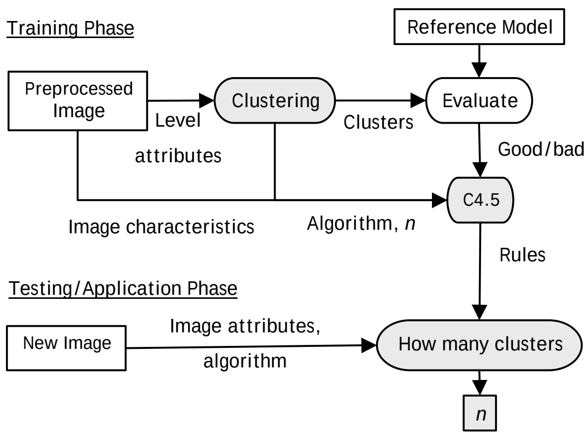


Figure 2. Experiment Design for Parameter Learning

The purpose of the Testing/Application phase is to apply the generated rules on a new image, and obtain the values of *n* to use for a given algorithm on that image.

4.2 Algorithm Learning

The purpose of algorithm learning, as explained earlier, is to learn which algorithm to use for a given image type.

The learning attributes here are image characteristics and the algorithm used (see Table 2). For each algorithm, the optimal *n* deduced from parameter learning was used.

Table 2. Algorithm learning attributes.

NAME	DESCRIPTION	VALUE
Size	Number of edge pairs	Continuous
Algorithm	Clustering algorithm used	KMeans, kNN
CLASSES	Whether the algorithm produces a good or bad road cluster	Good, bad

The two phases of inductive learning for algorithm learning is shown in Figure 3. First, level 1 RAIL attributes of the given image were calculated and used to cluster with different algorithms. The clusters generated were evaluated against the reference model, and the algorithm producing the best road cluster was classified as "good", with the other algorithm being labeled as "bad". The learning attributes (see Table 2) together with the classification of each run were used to generate a decision tree for application on new images.

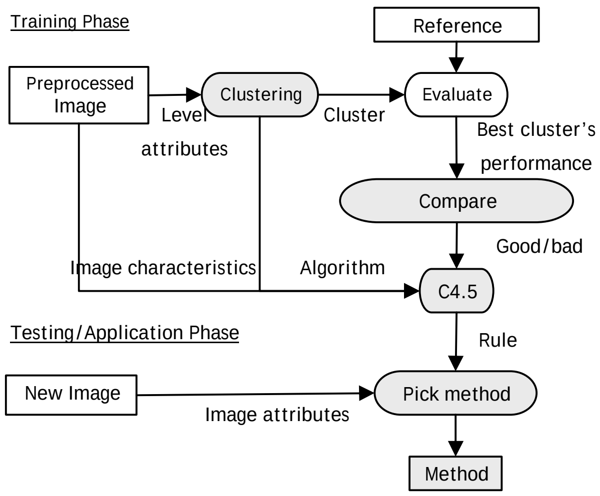


Figure 3. Experiment Design for Algorithm Learning

4.3 Cluster Learning

In cluster learning we want to deduce rules for identifying the road cluster of each clustering experiment.

The learning attributes we have identified in Table 3 are cluster characteristics and image characteristics.

Table 3. Cluster learning attributes.

NAME	DESCRIPTION	VALUE
Size	Number of edge pairs	Continuous
Aspect ratio (of cluster)	$Width_{max} / Height_{max}$	Continuous
Area	$Width_{max} \times Height_{max}$	Continuous
Centroid	Centre of the cluster	Continuous
CLASSES	Whether the cluster contains road edges	Good, bad

Clustering experiments with different algorithms and parameters were run. All the clusters generated were evaluated against the reference model and classified based on the evaluation. The learning attributes from table 3 together with the classification of each run were used in the inductive learner. This process is shown in Figure 4.

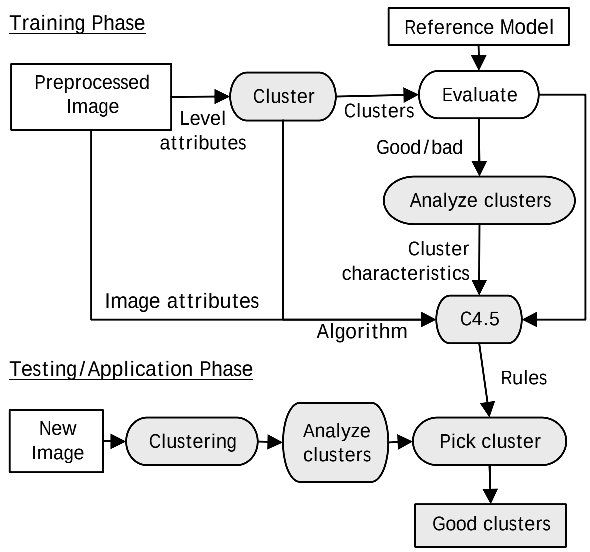


Figure 4. Experiment Design for Cluster Learning

5. Evaluation

The evaluation was performed in two parts. The first part evaluated the inductive clustering framework, whilst the second part compared the performance of KMeans, kNN and inductive learning on level 1 of RAIL. We used inductive clustering to select the best parameters for each clustering algorithm in part one, and then used these parameters in the second part when evaluating clustering algorithms.

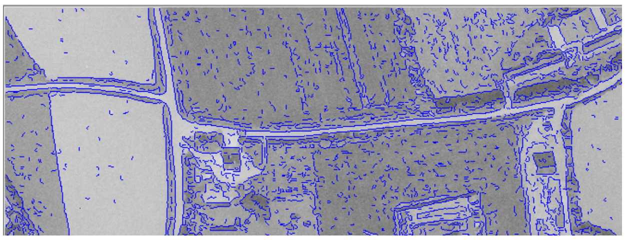


Figure 5. Image A.

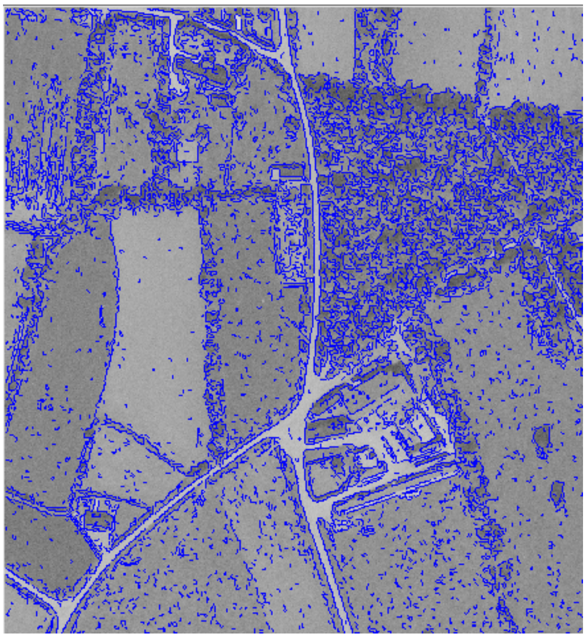


Figure 6. Image B.

Each classification technique was initially tested on two digital aerial images of a rural area in France, which we shall call image A and image B. These images have a ground resolution of 0.45m/pixel. Image A contains 1956 edges (Figure 5), and the other contains 6481 edges (Figure 6). These figures also show the initial edges generated for each image.

Figures 7 and 8 show the reference model created manually for Image A and B. This reference model is used to evaluate different classification techniques, and also to automate the inductive clustering framework. The reference model does not actually include the entire road, due to limitations in our reference model format.

5.1 Inductive Clustering

Two images are not enough to learn from and test on for the inductive clustering framework, so we divided these images into 5 sub-images, giving us 10 sets of edge pairs in total to experiment on. This subdivision was implemented by forming all possible edge pairs in an image, and calculating the RAIL level 1 attributes for those edge pairs. The resultant attributes were randomly split into 5 subsets, which were treated as independent sets for the purpose of testing.

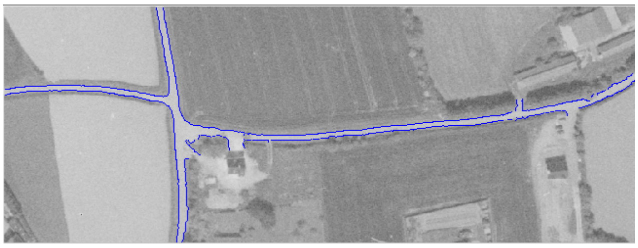


Figure 7. Image A reference model.

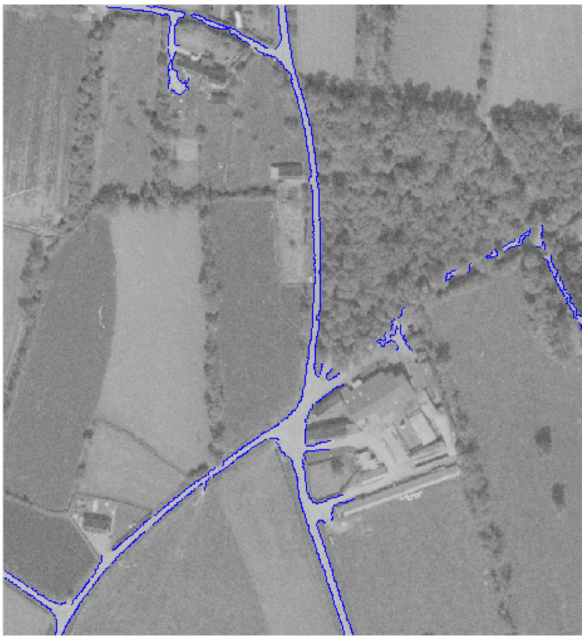


Figure 8. Image B reference models.

Clustering experiments were then run on each subset. Unbiased error rates were calculated for each stage of the framework (see Section 5.1.2).

5.1.1 Inductive Clustering Rules

Here we present the rules generated by inductive clustering on our two images. Note that the inductive clustering framework can be adapted to different applications and all sorts of images. However, the rules that we present below can only be applied to images with similar characteristics (e.g. resolution, complexity, etc.) to the ones used for learning. We include them here as an example output of inductive clustering.

Each rule identifies a partition of data via its learning attributes and gives a classification for that partition (bear in mind that all of the properties have to be true). The percentage after the classification indicates the accuracy of this rule when it is applied to the training data (but note that this accuracy measure is optimistically biased).

Parameter Learning[1]

```
Rule 16:
        size > 7023
        size <= 7057
        algorithm = kmeans
        n > 10
        ->  class good  [70.7%]
Rule 13:
        size > 6887
        size <= 6969
        algorithm = kmeans
        n > 8
        ->  class good  [61.2%]
Rule 23:
        algorithm = kmeans
        size > 7146
        n > 8
        ->  class good  [61.2%]
Default class: bad
```

Summary: For small images use *n* between 5 and 7 and any algorithm. Use *n* greater than 8 for larger images, along with the KMeans algorithm.

Algorithm Learning

```
Rule 1:
        algorithm = knn
        ->  class good  [68.7%]
Rule 2:
        algorithm = kmeans
        ->  class bad  [68.7%]
Default class: good
```

Summary: kNN generally produces better results than KMeans.

Cluster Learning

```
Rule 4:
        size > 8919
        enclosed_intensity > 142.342
        area > 1961.34
        ->  class good  [80.9%]
Default class: bad
```

Summary: Road clusters have an enclosed intensity (measured in the range [0, 255]) greater than 142 and area (see Table 3) greater than 1961.

5.1.2 Inductive Clustering Evaluation

Table 4 shows the evaluation of the rules presented in the last section. We performed 5-fold cross validation (Weiss, 1991) on our data to determine unbiased error rates.

The number of data points for learning in each section are shown in Table 4. In algorithm learning we have one data points for each of the two algorithms available (see Table 3), and 10 sub-images in total. With parameter learning and cluster learning the number of data points is found from a combination of the attributes being tested (see Table 1 and Table 3).

[1] To make the results clearer, some rules in the parameter learning section that did not involve n have been omitted.

Table 4. Evaluation of Inductive Clustering, showing accuracy rate of each fold for parameter, algorithm and cluster learning.

Learning	Param.	Algo.	Cluster
Data Size	521	52	4945
Fold 1	96.8 %	70 %	99.5 %
Fold 2	91.2 %	88.9 %	98.9 %
Fold 3	94.1 %	75 %	99.6 %
Fold 4	95.1 %	90 %	98.9 %
Fold 5	93.6 %	63.3 %	98.9 %
Avg Accuracy Rate	**94 %**	**77%**	**99 %**

The rules induced for parameter learning, algorithm learning and cluster learning show 94%, 77% and 99% accuracy respectively.

5.2 Evaluation of Classification Techniques

Here we will present the results of different classification techniques in RAIL. The parameters of clustering algorithms were deduced from the results of the inductive clustering framework. We used Equation (1) to measure the performance of each classification techniques.

5.2.1 Obtaining Outputs from Classifiers

The output for inductive learning is obtained by filtering all the edge pairs of the image through the thresholds generated by C4.5 to get a set of road edge pairs. We strategically identified 32 positive and 493 negative examples of edge pairs from image A, and calculated the attributes of these edge pairs for C4.5. From this data the following rules were generated (the percentage shows the accuracy of the rules when applied to the training data set):

```
Rule 4:
        IntensityContrast > 29.7
        EnclosedIntensity > 164
        ParallelSeparation > 4.6
        ParallelSeparation <= 20
        ->  class true  [77.0%]
Default class: false
```

These thresholds are used to extract the road edge pairs from each image.

The outputs of each clustering algorithm are defined by the single best cluster generated with the best value parameters of each particular algorithm in a given image. These are taken from the output of the inductive clustering framework.

5.2.2 Classifier Performance

The output of each classifier was evaluated against the reference model, and the results are presented in Table 5.

Table 5 shows that KMeans seems to be the best performer across both images, with kNN being almost as good. Inductive learning performed poorly on Image B, because the rules were learnt on a small sample set from a different image (Image A). This is also why inductive learning performs perfectly on Image A. Correctness is quite low for all classifiers, which is acceptable for level 1 of RAIL as we do not want to risk removing important information too early. Tables 6 and 7 show a visual comparison of the output of these classifiers.

Table 5. Comparison of different classification techniques.

Image	Measure (%)	Inductive Learning	KMeans (n = 9)	kNN (n = 9)
Image A	Complete	100	88	93
	Correct	8	16	15
Image B	Complete	63	89	87
	Correct	23	17	14

The results are confirmed by a Tables 6 and 7, which show that the road edges are mostly covered for all methods, except for inductive learning on Image B. This is an accurate reflection of the completeness measures in Table 5. However, note that because of the pairing technique used, a road may appear to have complete coverage when it does not. The output of KMeans gives a cleaner image than the others, with less non-road edges, as indicated by its relatively high correctness measure in Table 5.

6. Conclusion and future work

In this paper we have compared different classification techniques for low-level road extraction. Clustering with KMeans and kNN gives similar results. We have also introduced a way of automating clustering for road classification using inductive learning techniques. We have implemented and tested this concept on our RAIL system, and preliminary results are encouraging.

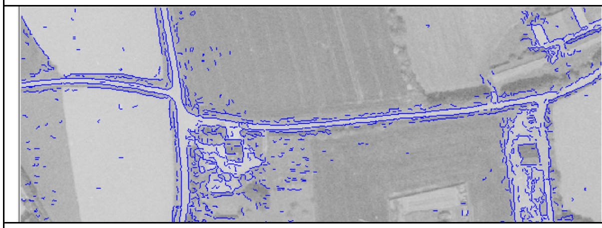
However, the results of inductive learning could and should be improved with more training data. We only used a small number of images because we do not have an extensive collection of images with similar resolution and coverage. Also, the creation of a reference model is a fairly labour-intensive task, as is the selection of positive and negative examples for learning.

In the future we hope to improve the accuracy of our testing. One avenue for doing this is to develop a better reference model, addressing the inherent shortcomings of our current edge-based one. We plan to train our system using a larger set of images in order to generate better rules. This step requires a more complete set of image characteristics to usefully classify images with.

The evaluation measures used (*cxc* and *complete*) are handpicked and their thresholds set empirically. Automation of this process is a future goal. We also plan to extend the clustering framework and these classifiers to the other levels of RAIL.

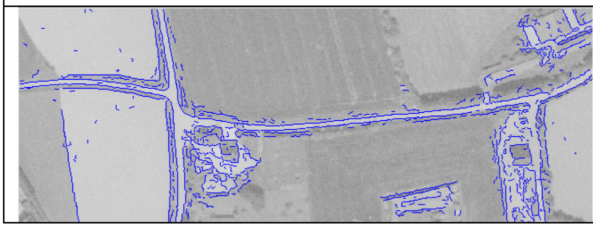
Table 6. Image A Outputs

Inductive Learning



KMeans

kNN

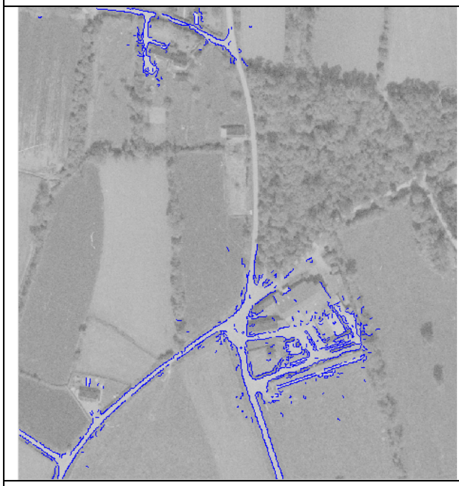
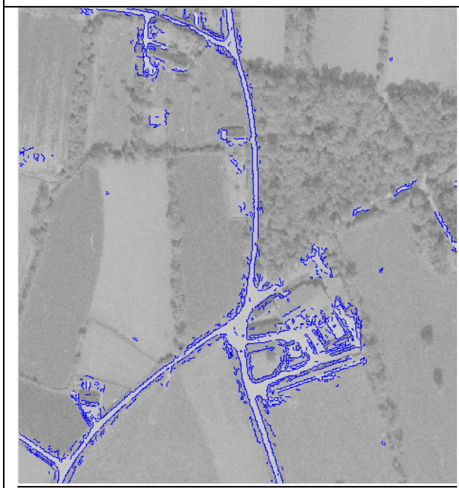
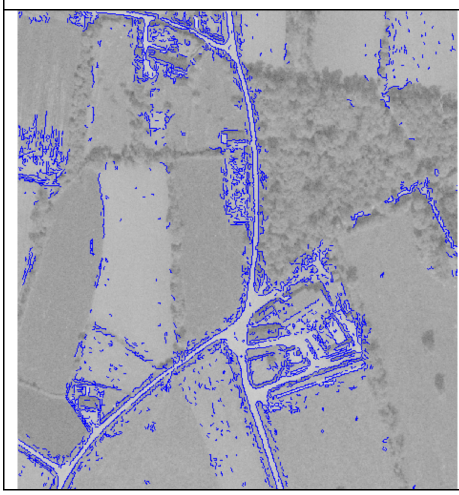


References

Chen, A., Donovan, G., Sowmya, A., Trinder, J. (2002a). Inductive Clustering: Automating Low-level Segmentation In High Resolution Images, Submitted to: ISPRS PCV 2002.

Chen, A., Donovan, G., Sowmya, A. (2002b). A Comparative Evaluation of Road Classification Techniques in High Resolution Imagery, Accepted to: PRRS 2002.

Table 7. Image B Outputs

Inductive Learning

KMeans

kNN


Harvey, W. A. (1999). Performance Evaluation for Road Extraction. In: *The Bulletin de la Société Française de Photogrammétrie et Télédétection,* n. 153(1999-1):79-87.

Quinlan, J. R. (1996). *C4.5: Programs For Machine Learning*, San Mateo, California, Morgan Kaufmann Publishers.

Singh, S., Sowmya, A. (1998). RAIL: Road Recognition from Aerial Images Using Inductive Learning. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXII, Part 3/1, pp. 367-378.

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

Teoh, C.Y., Sowmya, A. (2000a). Junction Extraction from High Resolution Images by Composite Learning. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Amsterdam, Netherlands, Vol. XXXII, Part B3, pp882-888.

Teoh, C.Y., Sowmya, A., Bandyopadhyay S. (2000b). Road Extraction from high resolution images by composite learning, In: *Proceedings of International Conf. Advances in Intelligent Systems: Theory and Applications*, Amsterdam, Netherlands, pp308-313.

Trinder, J. C., Nachimuthu, A., Wang, Y., Sowmya, A., Singh, A. (1999). Artificial Intelligence techniques for Road Extraction from Aerial Images. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* Vol. XXXII, Part 3-2W5, pp. 113-118.

Weiss, S., Kulikowski, C. (1991). *Computer Systems That Learn*, San Francisco, California, Morgan Kaufmann Publishers.