CERIAS Tech Report 2003-45
Change Detection in Overhead Imagery Using Neural Networks
by Christopher Clifton
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Applied Intelligence 18, 215–234, 2003

Change Detection in Overhead Imagery Using Neural Networks

CHRIS CLIFTON
Department of Computer Sciences, Purdue University, 250 N. University St. West Lafeyette, IN 47907-2066
clifton@cs.purdue.edu

Abstract. Identifying interesting changes from a sequence of overhead imagery—as opposed to clutter, lighting/seasonal changes, etc.—has been a problem for some time. Recent advances in data mining have greatly increased the size of datasets that can be attacked with pattern discovery methods. This paper presents a technique for using predictive modeling to identify *unusual* changes in images. Neural networks are trained to predict "before" and "after" pixel values for a sequence of images. These networks are then used to predict *expected* values **for the same images used in training**. Substantial differences between the expected and actual values represent an unusual change. Results are presented on both multispectral and panchromatic imagery.

Keywords: change detection, overhead imagery, neural networks

1. Introduction

Detecting changes based on a sequence of overhead imagery has been studied for some time. As early as 1972, work was done on pixel-by-pixel image differencing [1]. This early approach used intensity scaling, followed by differencing, to identify changes. However, such a simple approach is useful only when no uninteresting changes occur; in wide area overhead imagery, this is not the case. Two pictures of the same location taken at different times of the year will be quite different, but most of the changes are a result of natural effects (e.g., falling leaves). Advances have been made in change detection, most recently it has been shown to have promise in such diverse applications as treaty verification [2] and assessment of environmental impact [3].

The goal of the research presented in this paper is to identify *unusual* changes, without a pre-defined notion of what is usual or unusual. Instead, neural networks are used to determine what is an expected change, and highlight the changes that do not meet these expectations. The goal is not to *understand* the change, but to highlight areas deserving further analysis. Change detection thus serves as a "first pass" in image analysis, weeding out significant visual differences that are not likely to be of interest.

An example of the types of changes detected is given in Fig. 1. The center circle encloses a new road, the left appears to be a new parking lot, and the right encloses a ship. They represent changes actually identified by using the process described in this paper on these images. There is a bias toward detection of small scale changes such as the appearance or disappearance of a feature. Large-scale changes such as a large area of clear-cut logging are deemed "normal", in fact trees left standing are tagged as changes. While not appropriate for all types of image analysis, tagging small areas of change for further analysis is useful in a wide class of problems.

Recently, data mining has given commercial impetus to solving computational issues of pattern discovery in large data sets. This opens up new possibilities for applying advanced pattern discovery techniques to image analysis problems. The basic approach of the work presented in this paper is to use neural networks to model the expected new value of a pixel based on the old value. Each point in a region is treated as a vector of one or more old values and one or more new values. The *or more* can come from different views of the point, different spectral bands of multispectral imagery, or even widely different types of imagery such as visual and radar. The images are thus represented as a set of point vectors; an artificial neural network (ANN)

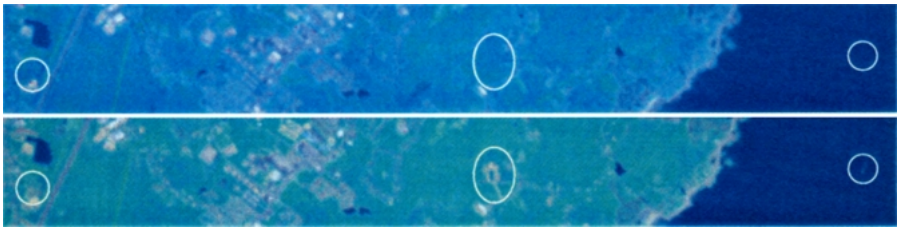

Figure 1. Two images with unusual changes circled.

is then trained to predict the new values from the old values. The ANN is then used to predict the new values *on the same images*; points where the prediction is significantly in error are deemed unusual changes. Note that there is no "training data" as such, the training is done against the actual test data. It is the ability of neural networks to generalize, rather than their ability to learn against a manually-tagged corpus, that is key.

This is a brief overview. Before going into details, a discussion of prior research will show where this technique falls in the spectrum of related work. Section 3 gives an example showing specific data requirements and results. The black box of the process is opened in Section 4. A discussion of the use of this process (and the effect of various parameters) is provided along with more results in Section 5.

2. Background

The imagery and vision communities have a long history of work on change detection. A survey can be found in [4], and recent work in [5, 6]. Please refer to those papers for a full survey, only a quick breakdown of prior work is given here. Change detection work falls into two categories: Change vector analysis, and pixel-level comparison. Change vector analysis requires developing a model of what *should* be in an image (e.g., a vector diagram of buildings and roads). The actual (new) image is then compared with the diagram, and differences are highlighted [7, 8]. The necessity of constructing a vector diagram of what should be found in a region appears to pose a high overhead, and would limit this technique to a few locations important enough to justify constructing a diagram. However, this presumes that diagram definition is manual. An alternative is to develop the diagram automatically [9] (perhaps independently for the before and after images), then compare.

Change vector analysis is dependent on the diagram capturing the types of changes of interest. It requires that pre-defining what is, and is **not**, important. An alternative is to directly compare the images. Earlier research in direct image comparison suffered from an inability to filter *uninteresting* changes. A simple differencing can be foiled by changes in lighting intensity. Better threshold formulation [10] and scaling [11, 12] techniques have been developed, but these still face problems with environmental changes (imagine two pictures of a farming area, before and after a snowfall—the highly reflective snow versus a dark plowed field gives a huge difference in intensity, but not an interesting one).

One approach is to model the expected spectral values for certain known "interesting items" (such as buildings and roads) [13], or explicitly model background noise [14]. Artificial neural networks have been applied to the change detection problem [3, 15], specifically using images *and* land use category "training data" to identify changes in land cover. These techniques still require considerable manual effort to define what is or is not interesting. In addition, this leaves the possibility that a pre-conceived notion of what is interesting may be wrong.

The approach presented here is similar in that a model is built to describe the expected differences between images. However, the key difference is that the model is based solely on what actually appears in the image. The "expected change" is based on what is common; an unusual change is simply one that occurs infrequently. Therefore a building appearing in the middle of a farm may show up as unusual, however if the entire farm is cleared and built over, the change would be classified as normal. Using the method presented here, even a non-change can show up as interesting: If an image of grassland in the spring is compared with the same snow-covered grassland the next winter, but one spot remains green (perhaps due to heat from an underground structure), the *unchanged* spot will be identified. This is because the model expects based on the images that green (grassland)

will become white (snow), so a lack of change is flagged.

Similar techniques have been tried. Morisette and Khorram [16] make use of generalized linear models to determine when intensity values between images are significant enough to constitute a change. They make use of tagged "training data" with significant changes identified, and develop a model based on those changes. However, limitations of linear models, and the potential differences between available training data and the real data of interest, limit applicability of their technique. Bruzzone and Prieto [17] employ Bayesian decision theory to set thresholds for change, helping to overcome limitations of a linear model. The method presented in this paper shares a similarity with theirs that both use change of neighboring pixels to determine if a change is significant, and appears more applicable to detecting small-size, localized changes. Schaum and Stocker [18] utilize a sequence of images to determine expected changes, then identify inserted man-made objects as deviations from the expected changes. Their technique is able to detect sub-pixel changes (as low as 1.6% of a pixel). However, it also requires extremely high-quality image registration. Their technique is more applicable to detecting small items (e.g., a vehicle), rather than wide-area, long-term changes.

Data mining technology has been previously applied to other imagery problems. The SKICAT [19] applied data mining technology to the classification of astronomical objects, discovering new types in the process. JARtool [20] looked for specific features in overhead images; a specific application was identifying volcanoes on Venus. Birch [21] used data mining technology to study foliage. A closer application to the one presented here was Quakefinder [22], which looked for a specific type of change (earth movements) in overhead imagery. One aspect of Quakefinder that is particularly relevant to this paper is the ability to accurately register different images; this is discussed in Section 3.1.

3. Data Requirements

The change detection process presented in this paper finds locations with unusual *spectral intensity* changes. A location is modeled as a real-valued vector of *n before* values and *m after* values. An image is a collection of these vectors. The values of n and m are determined by the imagery available. For example, given two Landsat-TM images of the same location, taken at different times, then $m = n = 7$, corresponding to the 7 spectral bands of Landsat-TM imagery.

The corresponding case for panchromatic imagery (a single before and after image), gives $m = n = 1$. Using a single before and after value to represent a location is insufficient for the method presented here. However, the process presented here is not limited to multi-spectral imagery; additional information on a location can come from multiple images (either at different times, or from different views) for both before and after shots. It is also possible to use imagery of different types, such as a Landsat-TM before image ($m = 7$) and two panchromatic views as an after image ($n = 2$).

The vectors represent *locations*, not *pixels*. The simplest approach is to use a pixel as a location. However, better results are achieved when the resolution used is relative to the size of an "interesting feature". For example, if the goal is to find changes in permanent structures (e.g., roads, buildings), a typical feature is roughly 10 meters in the smallest dimension. If a vehicle in an unusual place, say, the middle of a field, is considered interesting, feature size would be roughly 2 meters. The change detection process presented here works best when the size of a location is roughly one third to one half the smallest dimension of a feature. (This agrees with the findings of Roy [23].) In many cases, it is okay to use a coarser resolution than the pixels of the raw imagery.

3.1. Image Registration

The primary difficulty with the approach presented here is the need for image registration. To build a vector for a location, that location must be identified on all of the images. This will generally require warping the images to a common standard orthorectification. Although techniques for automatic image registration are known [22, 24, 25], it is non-trivial to achieve pixel-level matching (although this is improving, see [26]). However, as discussed in the preceding paragraph and shown in the example in Section 5.2, pixel-level resolution is not usually necessary. Therefore registration is only needed to within a few pixels, based on the size of a "location" or feature. The ability to work with poorer than pixel-level registration is significant, as previous studies have shown accuracy losses of 50% with less than one pixel misregistration [27, 28].

In the experiments presented here, images were manually orthorectified. The Landsat-TM images

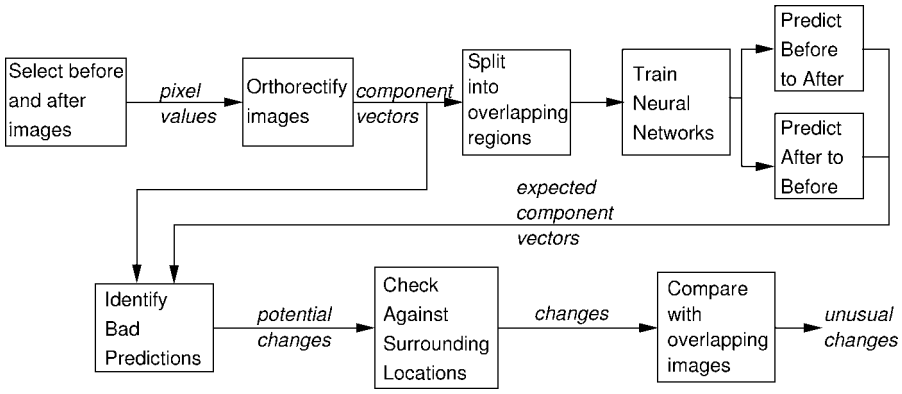

Figure 2. Overview of the change detection using neural networks process.

required only a slight vertical and horizontal translation. The panchromatic images of Section 5.2 were taken from a variety of angles, orthorectification was done using ERDAS IMAGINE [29]. This required selecting several pairs of points in two images that correspond to the same actual locations. IMAGINE then scales the images so that these points correspond to the same pixels, stretching intermediate points appropriately. The automated techniques referenced above use similar scaling, but automatically match "features" (such as lines) instead of relying on manual selection of points.

3.2. Data Preparation Process

Some of the key issues involved in preparing data have been outlined above. The specific steps involved are:

1. Select the imagery to be used for before and after, where the goal is to identify unusual changes happening between the time ranges represented by the set of before and set of after images.
2. Select a resolution based on the type of analysis being conducted, i.e., the minimum size of an interesting feature. The resolution should be one third to one half of the minimum dimension of a feature.
3. Orthorectify the imagery to within the chosen resolution (e.g., if the resolution is 10 pixels, orthorectification to within 5 pixels is adequate).
4. Build the vectors representing each location. For each image, and each spectral band within the image, construct a value by averaging the pixel values nearest that location (i.e., for a 10 pixel goal resolution, average the surrounding 10×10 region to get the value) in that band/image. Note that the difference between the target resolution and the actual resolution can be different for different components of the vector.

The result is a set of vectors, with each vector corresponding to a location in the underlying imagery. An overview is included in Fig. 2.

Note that the set of vectors is not "training data" in the usual sense—no definition of what is an interesting feature or unusual change has been provided. The next section shows how the data serves as both training and test data.

4. Process

The meaning of change is subjective, making the methods for change detection difficult to define and evaluate. For purposes of this work, change is defined as follows: Given a set of component details $D = \{d_1, \ldots, d_n\}$, $d_i \subseteq \Re$ and a set of location identifiers $L \subseteq \Re$, an image is a set of vectors $(l, c_1, \ldots, c_n)$ where $l \in L$ and $c_i \in d_i$. Given two images A and B, a vector pair is the vector $(l, (c_A 1, \ldots, c_A n), (c_B 1, \ldots, c_B n))$ where $(l, c_A 1, \ldots, c_A n) \in A$ and $(l, c_B 1, \ldots, c_B n) \in B$. An unusually changed location l_u is a location where the vector $(c_A 1, \ldots, c_A n)$ is similar to the corresponding c_A vector in other vector pairs, but the $(c_B 1, \ldots, c_B n)$ vector is significantly different (or vice-versa). The notion of "similar", "significantly different", or even "location" is open to considerable interpretation—it is variations in these that lead to differences in what is considered an interesting change.

The change detection process consists of several steps. A high level outline of the process is given first, before proceeding to the details of each step. Figure 2 gives a block diagram of the entire process, including the data preparation of the previous section.

1. Build several models to predict the "after" vectors based on the "before" vectors. The models are Artificial Neural Networks, trained on the location vectors constructed in Section 3.2. A more complete description is given in Section 4.1.
2. Feed the "before" vectors into the models built in step 1 to obtain *predicted* after vectors.
3. Use the predictions from step 2 for each location to construct a range of expected values for each component of the "after" image at that location, as will be described in Section 4.2.

The above steps are then repeated, switching the "before" and "after" vectors (i.e., get an expected range for the before values based on the after values).

4. Compare the actual values (both before and after) with the range of expected values for each location. If a significant number of the values (discussed in Section 4.2) are outside the range of expected values, the location is marked as a *potential* change.
5. For each potential change, look for potential changes in the surrounding locations. If a significant number of potential changes are found, the location is marked as a change.

To support scaling, the image is divided into several overlapping regions, such that each location appears in four regions. The above steps can be performed on each region in parallel. A location is flagged as an unusual change only if it marked a change in two of the four regions, as will be discussed in Section 4.3.

The remainder of this section discusses the above steps in more detail.

4.1. Expected Change Prediction

The changed detection process presented in this paper makes use of a quickprop-trained neural network as a predictor. Studying the use of other predictive data mining techniques is an area for further research; decision tree approaches have not proven successful, but there are other possibilities.

First, a neural network is trained on one half of the data, using the other half as an evaluation holdout set. The minimum error achieved on the holdout set is used as a target error. A model is then trained on the entire set until the target error is reached. Specifically:

Determine Target Training Error

1. Randomly divide the data into equal-sized training and holdout sets.
2. Construct a three-layer sigmoidal neural network, with the number of input nodes equal to the size of the "before" vector for a location, and output nodes corresponding to the "after" vector. The appropriate number of hidden nodes grows as the size of the data set increases, 11 works well for a 10,000 location set. Seed the initial weights in the network with random values. A sample network for Landsat-TM data (as used in Fig. 1) is given in Fig. 3.
3. Train the network (using QuickProp [30]) until a minimum on the holdout set is found:
   (a) Run a training pass across the training set.
   (b) Input the "before" vectors of the holdout set to the network (feed-forward), and compute the root-mean-squared error between the output and the "after" vectors.

   The above process continues for $2n$ training epochs, where n is the epoch where the minimum error on the holdout set was found. This allows the algorithm to train past local minima.
4. Save the minimum error found.

The above process is repeated three times. The saved minimum errors are averaged to obtain a target training error.

Train Final Network. Construct and train a network as above, but train on the entire data set. Training continues until the error (on the entire data set) equals the target training error determined above.

The training technique described above is a standard technique to avoid over-specificity and obtain a final network that can be expected to generalize. Overtraining would in theory result in a network capable of predicting every value in the training set, thus giving no unusual changes; training to get a network that could be expected to generalize to imagery with similar characteristics alleviates that problem.

ANNs give a single set of output values (predictions) for each set of input values. However, some features are

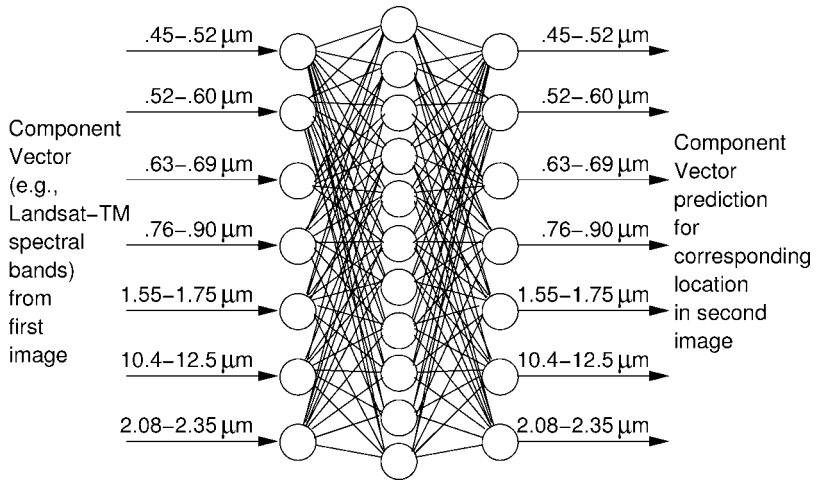

Figure 3. Example neural network for landsat-TM data.

more varied in their changes than others—a deciduous forest may go from green in the summer to red and yellow in the fall, while a farm field goes from uniform green to uniform brown. To capture the variation, a range of expected output values (predictions) is obtained by training multiple neural networks. The above process is repeated, using different random values to seed the network each time, to give several predictions for each location. For spectral values corresponding to terrain where the changes are consistent, such as water or pavement, the predictions from the networks will be close to each other. However, for spectral values corresponding to difficult to predict terrain (such as plowed fields, that may have different types of crops in later pictures), the predictions from the separate networks are likely to be farther apart. Since each network starts with a different set of random weights, and stops at a local minima, for difficult to predict spectral values the networks may not reach the same local minima. The range of predictions is used to automatically vary the threshold for declaring an unusual change based on terrain type, without any preconceived notion of terrain type. Training five networks was sufficient to get a good range of predictions for the tests shown in Section 5. The use of these five predictions to determine unusual changes is described in the next section.

The experiments discussed here used the NevProp4 package [31] (based on the QuickProp algorithm [30]). However, the process described could be optimized using a custom-designed package (computing the target error once, and reusing it to construct each of the five final networks, for example).

4.2. Deciding When to Deem Changes Unusual

The neural networks give a range of values for each vector component of each location. Next, a "prediction error" is calculated for each component of each location as the difference between the actual value and the average prediction divided by the difference between the high and low predictions. For each component, the average and standard deviation of the prediction error is taken over all locations.

A **bad** prediction is defined as a location component where the prediction error is greater than the *average error* $+ k *$ *standard deviation of error* for that component. Note that the value k can be adjusted; $k = 4$ works well and is used for all the results presented in this paper. The results are not too sensitive to this parameter, however adjustment can be made to give best results for the type of imagery and analysis being used. Increasing k results in only more extreme changes being flagged as unusual.

The next step is to look for corroborating evidence to get potential changes. The first form of corroboration is multiple vector components for each location; the requirement that one third of the components for a location give a bad prediction is used in the examples presented in Section 5. Again, the value can be adjusted to fit the needs of the particular analysis. Note that the multiple vector components includes both predictions for each component of the after values based on the before values, and before values based on the after values. The meaning of this parameter is most dependent on type of imagery. For example, in multispectral

imagery a low value would be more effective at detecting *camouflaged* changes; the camouflage may prevent detection in most but not all spectral values.

Finally, corroborating evidence in the form of nearby potential changes is tested. Here the requirement is that at least two thirds of the locations in a 3×3 region to be potential changes for the center point to be considered a change. Again, this is an adjustable parameter. Adjustments here affect primarily the size of unusually changed features that will be discovered—a 3×3 region finds features of size at least six pixels. For the man-made features that have been the focus of this work, using small regions and decreasing resolution (increasing pixel size) gives better results and faster computation than using a large region at full image resolution. A larger region might be appropriate for changes that are spread across a region, e.g., scattered damaged or diseased plants in a field where the entire field is the "feature".

The three parameters used to determine *unusual change* based on the neural network predictions are summarized in Table 1. The default parameters were chosen based on empirical study using imagery other than that presented in Section 5. The effectiveness on the widely different types of imagery presented in Section 5 (panchromatic and multispectral imagery of vastly different resolutions) suggests that the defaults are good choices. Figures 6–8 and 16–18 in Section 5 show the effect of varying these parameters. A key point is that these parameters are used *after* the computationally intensive part of the process, and thus could be adjusted interactively by a user to obtain results appropriate to that user's particular task.

Table 1. Parameters for determining unusual change.

Value	Calculation	Default parameter
Bad prediction	Distance between prediction and value for a particular component. Distance must be greater than *average error* $+ k *$ *standard deviation of error*, where average and standard deviation are based on errors for that component in the image pair.	$k = 4$
Component agreement	Number of components that must have bad predictions to consider a location bad	1/3 of components
Surrounding changes	Number of surrounding locations that must be bad to consider a location an unusual change	5 the 8 immediate neighbors

4.3. *Overlapping Regions*

Scaling the method presented in this paper poses some difficulties. As the image grows, so does the number of "training instances" for the neural network (the dominating computational factor). Perhaps more important, larger regions increase the *diversity* of training instances, substantially increasing the difficulty of developing a good model relating the "before" and "after" images. In addition to the increased training time, diversity can pose a logical problem. Two features may have the same "before" values, but different "after" values. An example would be an ocean, and a seasonal lake. By itself, the ocean is highly predictable: It will remain water. Likewise, a seasonal lake will go from water (in the wet season) to relatively uniform dirt (in the dry season). Depending on the relative sizes of the regions, such differences in normal change can pose two problems:

1. The "smaller" feature may be dominated by the larger, and deemed an unusual change; or
2. The two features may be deemed "unpredictable" (some of the neural networks predict one, some the other, giving a wide range of possible values). The result is to miss truly unusual features (e.g., a building appearing in the middle of a lake).

Note that both of these are instances of this process working as expected, it is just the scale that is off. The human definition of interesting can vary as well. A source of water disappearing, e.g., a swamp being drained in preparation for construction, would probably be of interest. However, a naturally occurring seasonal lake would not be.

The solution to this problem is to look for changes relative to a small region. An image is partitioned into small regions, based on an assumption that a region is relatively homogeneous: even if something is unusual on a global scale, if it is usual in a region it should be ignored. Other techniques have made the expectation that changes are most significant relative to the neighborhood; [32] used multifractal techniques to inject locality considerations. The choice of region size is important, however on the test data presented

Figure 4. Changes detected in Landsat-TM images: Greater Portsmouth, New Hampshire on July 27, 1989.

in Section 5 little difference within the range of 1000–50000 hectares.

The primary problem is when a feature is unusual in one region, but common in a neighboring region. Such locality problems are handled by looking at overlapping regions (each location is part of four regions). Based on experiments a change is deemed to be unusual if it is identified as a change in at least two of the four regions.

Over several tests, the partitioning approach obtained the same results as training a single network on an entire image. However, the total computational time was decreased (even though each point is being used four times). In addition, each region could be processed independently, allowing the method to be parallelized across loosely-coupled machines (e.g., a network of workstations).

5. Sample Results

A pictorial overview of results on two different tests are presented. The first uses two Landsat-TM (multispectral) images taken four years apart. The second uses multiple panchromatic aerial photography images taken two days apart. These examples demonstrate applicability to two very different types of imagery.

The formulas used to determine settings for parameters were derived through experimentation on different images; the parameters presented here were computed prior to testing on these images.

Rather than marking the locations where unusual changes were detected (which would obscure the original image), the images have been overlaid with "bounding boxes" for the detected unusual changes, shown in white. These are intended only to support visual evaluation of results, and the method for constructing them was based on ease/convenience rather than technical correctness. A production system would use a more sophisticated method, such as constructing convex hulls around high-density regions of changed locations.

However, for completeness the method used is described here:

1. Compute all such bounding box regions where at least 50% of the contained locations are deemed

Figure 5. Changes detected in Landsat-TM images: Greater Portsmouth, New Hampshire on July 14, 1993.

changed. This creates a bounding box for every location, as well as larger boxes containing at least a 50% density of changed locations.
2. Where bounding boxes, replace them with the smallest rectangle completely containing the overlapping boxes.
3. Draw a white line on the image two pixels "outside" each box generated by the preceding step to avoid covering the interesting parts. (The resulting lines may overlap as a result.)

Step 2 has the potential to reduce the density of locations in a box (especially if the unusual changes occur on a diagonal), but this did not occur in the imagery presented—the boxes shown reflect areas with a reasonably high density of detected changes.

In addition to a pictorial overview using the "default" parameter values, statistics on the number of changes found and on the effect of the various post-neural-network filtering parameters are presented.

5.1. Multispectral Imagery

Landsat-TM images consist of six spectral bands capturing light from 0.45 to 2.35 micrometers wavelength at a 30 meter pixel resolution, and an infrared band at 120 meter resolution. A detailed description can be found at [33]. The images used are of Portsmouth, New Hampshire and the surrounding region. The first image was taken on July 27, 1989; the second on July 14, 1993. Figures 4 and 5 show the results on these images. (Marked changes are duplicated on both figures for ease of reference.) Many of the changes near the upper left are obvious to the naked eye. The lower right change is the new circular road of Fig. 1. (The other changes detected on the small region of Fig. 1 were viewed as "common" with respect to the larger image, and were not shown as unusual.) Of particular note is the number of visually significant changes that were *not* marked. In the lower lower left corner, there are numerous fields that are green in the 1989 image, but brown (i.e., plowed) in the 1993 image. The "green to brown" transition was

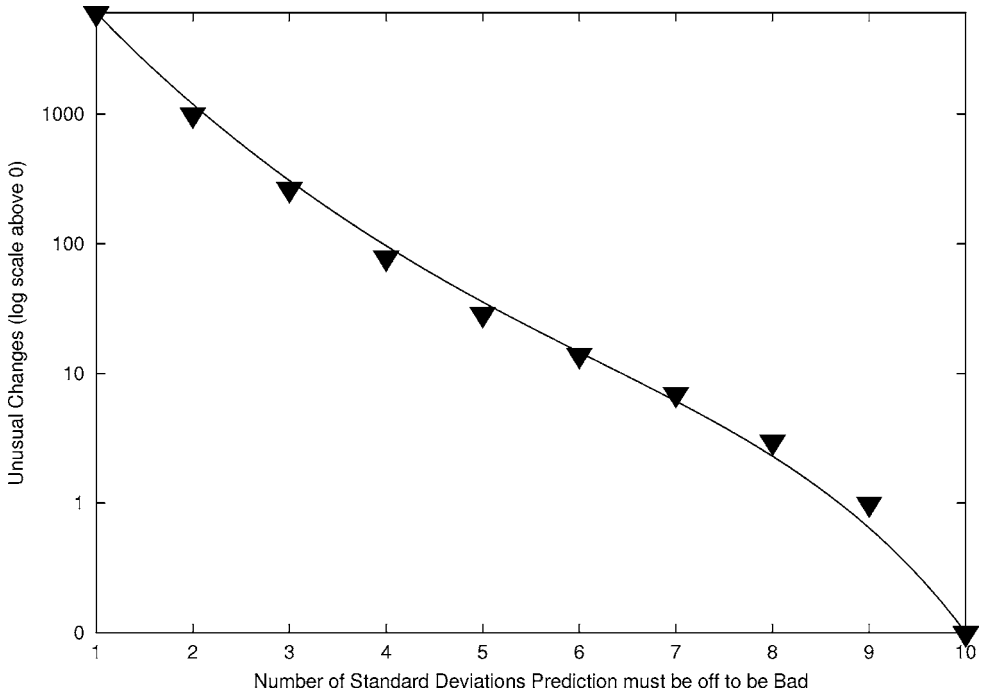

Figure 6. Number of unusual changes versus number of standard deviations beyond average error required to declare a bad prediction: multispectral imagery.

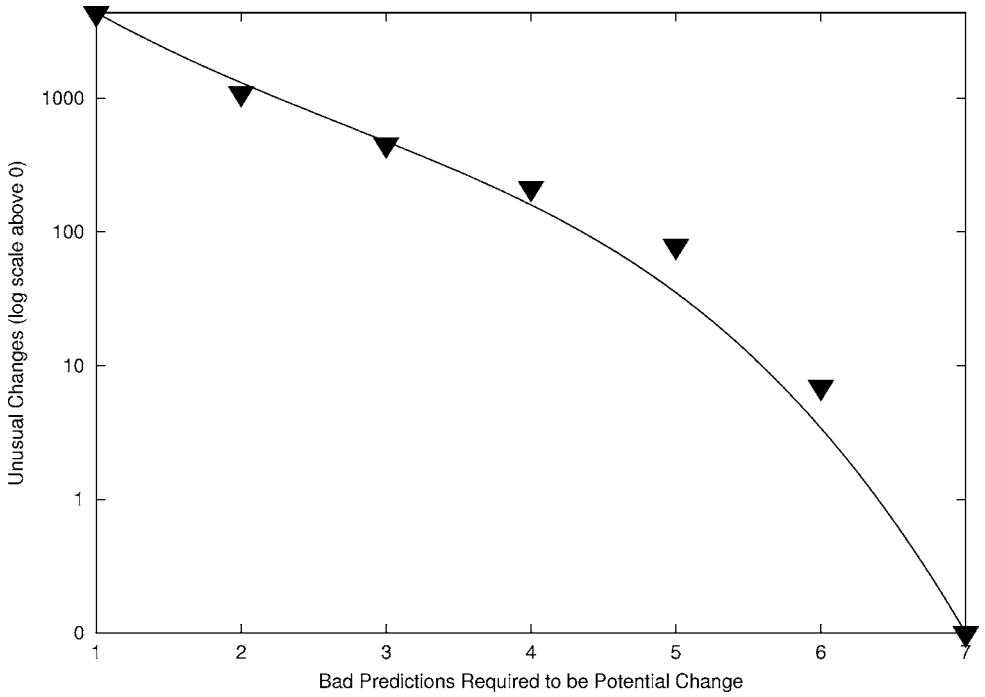

Figure 7. Number of unusual changes versus number of bad predictions required declare a potential change: multispectral imagery.

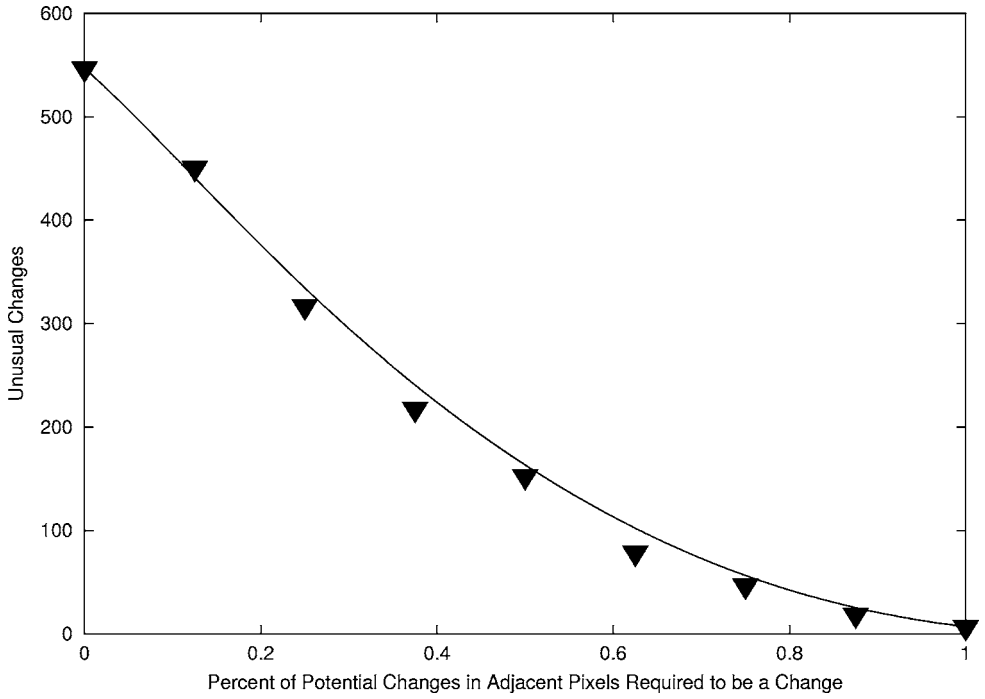

Figure 8. Number of unusual changes versus percent of potential changes in adjacent locations required to declare a change: multispectral imagery.

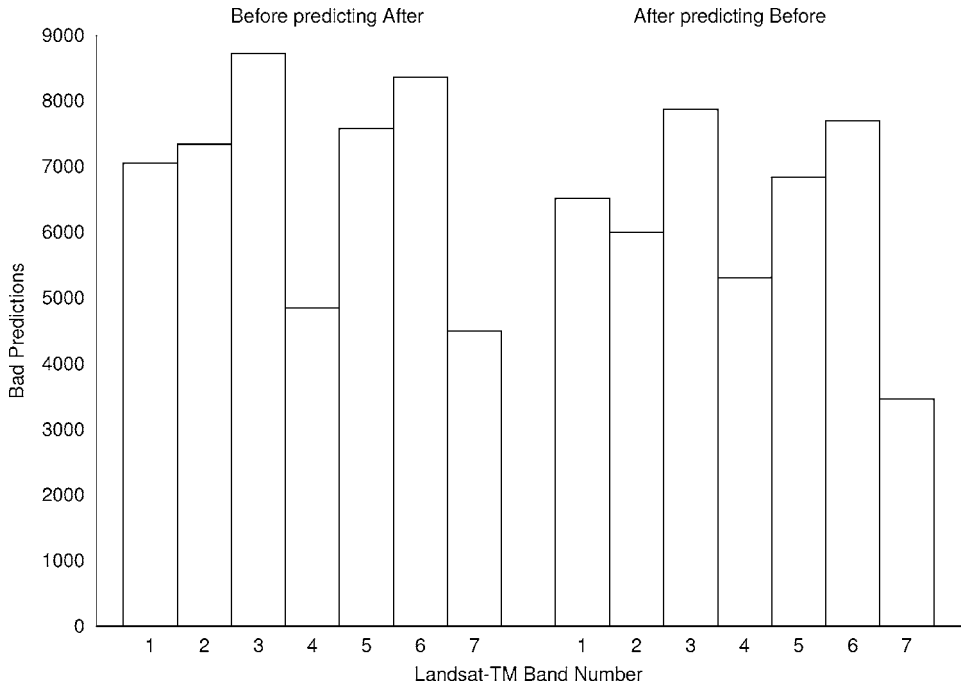

Figure 9. Number of bad predictions for each Landsat-TM spectral band. Left values are using the before image to predict the after image; right values are using the after image to predict the before image.

Figure 10. Reduced resolution display of upper quarter of Fig. 11.

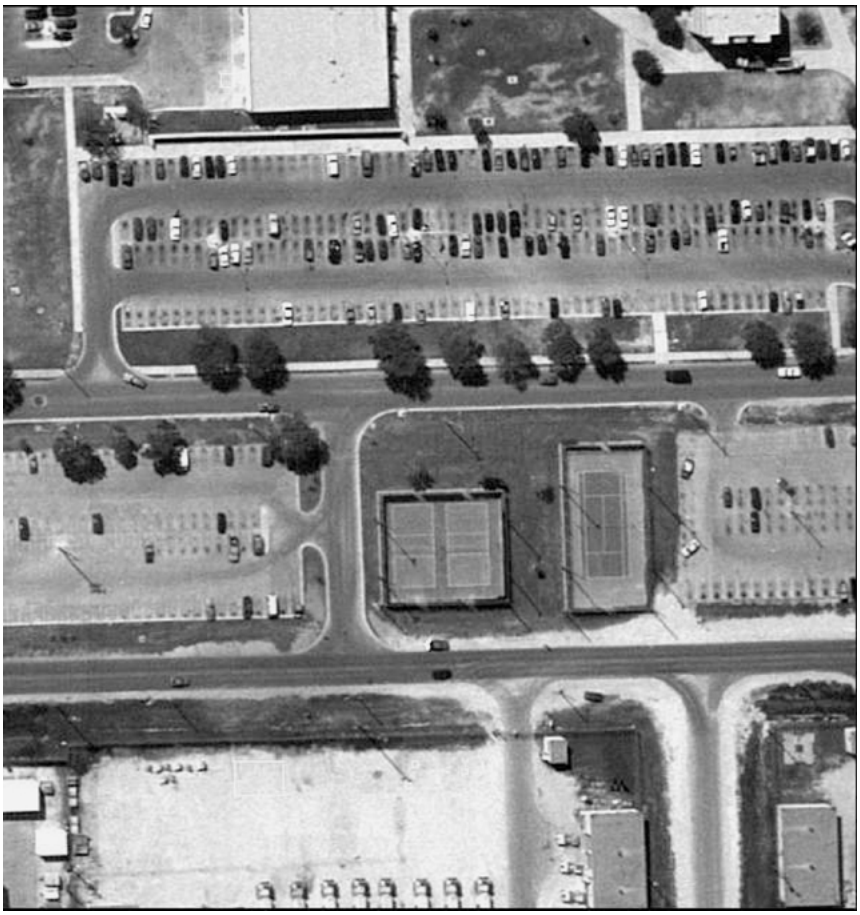

Figure 11. Changes detected in Panchromatic images: Overhead view taken 16:43:05 on 10/25/1993.

automatically factored out as an expected/predictable change.

There were a total of 79 unusually changed locations discovered in this test, divided among the 13 regions shown in Figs. 4 and 5. Figures 6–8 show the effect (in terms of number of changes discovered) of varying the parameters for declaring an unusual change. Each graph varies one parameter, the others are fixed at the default values of:

Standard deviations beyond average error	4	
Bad predictions	5	(1/3 of 14 predictions, 2 predictions per component)
Percent of adjacent locations with potential changes	60%	

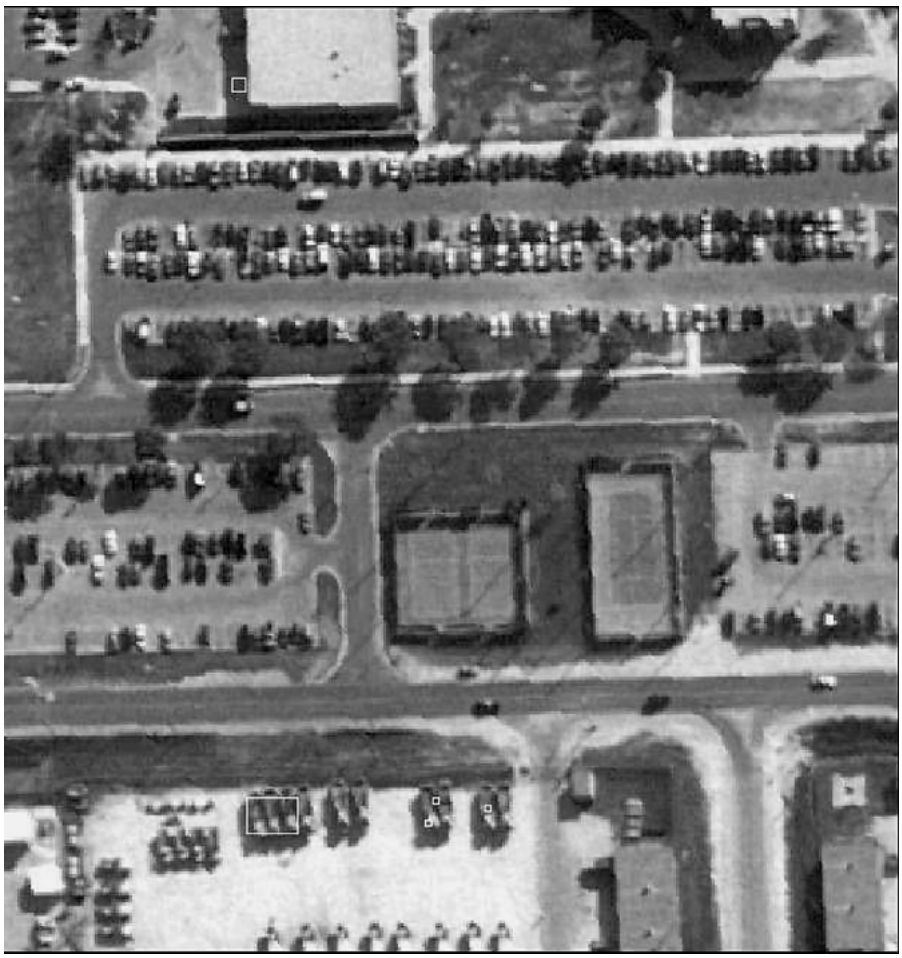

Figure 12. Changes detected in Panchromatic images: Oblique view taken 21:15:32 on 10/27/1993.

There was some variation in the value of specific spectral bands for predicting unusual changes. The number of "bad" predictions for each band, raw single-location predictions out of range before filtering based on the requirement for nearby potential changes, is given in Fig. 9. More differences between predicted and actual show up in the lower wavelengths. Also interesting is that the wavelengths that show the most "bad" predictions are similar even if the after (chronologically) image is used as the input, producing a network to predict the corresponding pixels in the before image.

Carrying the "before vs. after" comparison through the filtering process gives 548 unusual changes found using the old image to predict values in the new image, and 49 using the new to predict the old (versus 79 using both in combination). Note that in the "one-way" cases, there are only seven predictions, so three bad predictions were needed to reach the one third cutoff.

5.2. Panchromatic Imagery

The second example uses panchromatic (grayscale) imagery from aerial photographs. The images are much higher resolution than Landsat (the highest resolution images are approximately 0.3 meter pixels), but gives substantially less information for each location. To handle the "too little information" problem, three views taken on October 25 and two taken on October 27 are used. This gives multiple datapoints for both the before and after images.

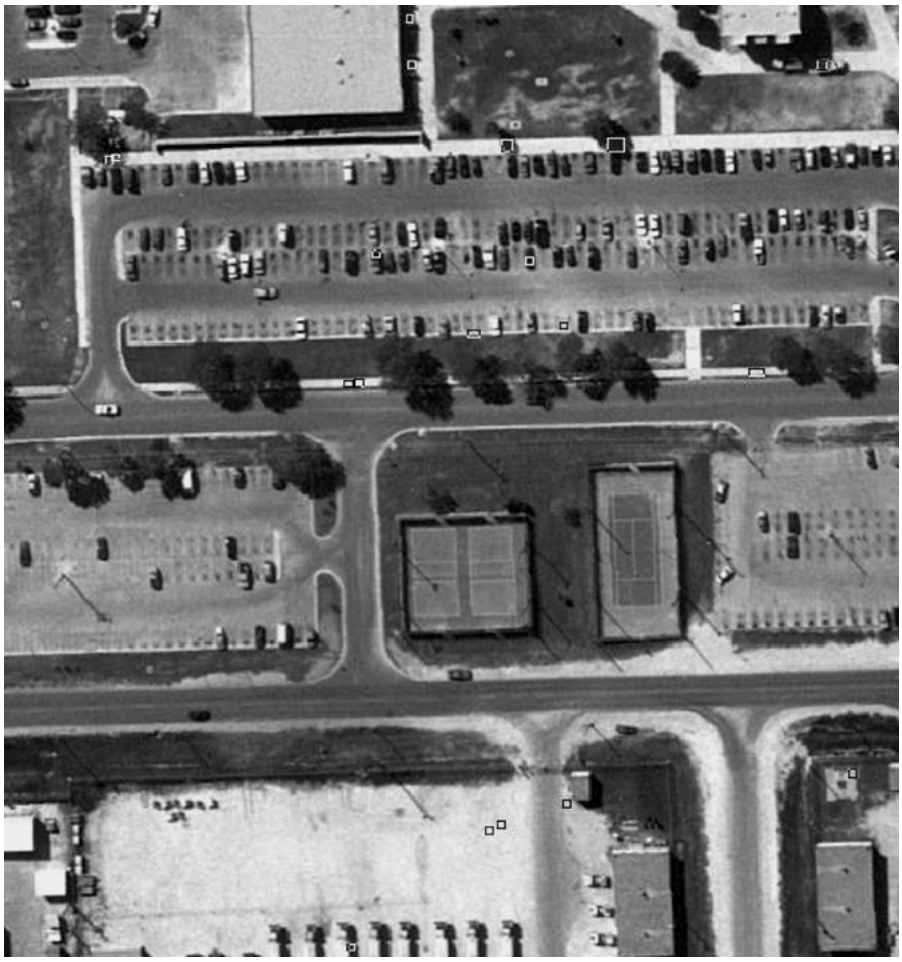

Figure 13. Changes detected in Panchromatic images, full resolution: Overhead view taken 16:43:15 on 10/25/1993.

A full description of the image collection can be found at [34]. The images were manually orthorectified as described in Section 3.1. The images are not used at full resolution; each "location" for the change detection process is composed of the average of a 3×3 pixel square. Figure 10 gives an example of what the image looks like at the resolution used; compare with the full resolution image in Fig. 11. Reducing the resolution discounted "changes" less than approximately 2 meters across.

All five images used are shown, however changes from different variations on the process are shown on the latter three. Except as otherwise noted, the process used is the full process with default parameters as described in Section 4, run on the lowered resolution images.

Figures 11 and 12 give the clearest view of the before and after images, and highlight the discovered changes. Note there were only two changed regions. The one in the upper left doesn't reflect anything "real", just an unusual change in shadowing. The change identified at the lower left, the arrival of several trucks, is more interesting.

Note that the cars/parking spaces were not identified as changes. This is not just a result of the decreased resolution, even running at full resolution the majority of the car/parking space changes were ignored (the exception seemed to be cars parked on some unusually dark oil spots.) As a comparison, Fig. 13 shows the changes identified running at full resolution.

The use of "predicting" the before state from the after state, as well as after from before, cuts false

Figure 14. Changes detected in Panchromatic images, using only "before" to predict "after": Oblique view taken 17:19:56 on 10/25/1993.

alarms significantly. Figure 14 demonstrates the results only using the new images to predict values from the old. Note the additional "false alarms" in the upper right, and along the walkway near the top (although limiting the features in this way *does* get both the large and small trucks).

The added information of multiple views isn't always necessary. Figure 15 shows the results using just the images from Figs. 11 and 15, instead of three before and two after.

There were a total of 108 unusually changed locations found in these images. Figures 16–18 show the effect (in terms of number of unusual changes discovered) of varying the parameters for declaring an unusual change. Each graph varies one parameter, the others are fixed at the default values of:

Standard deviations beyond average error	4	
Bad predictions	2	(1/3 of the components)
Percent of adjacent locations with potential changes	60%	

6. Conclusions and Future Work

This paper has presented a method for identifying unusual changes using overhead imagery. The

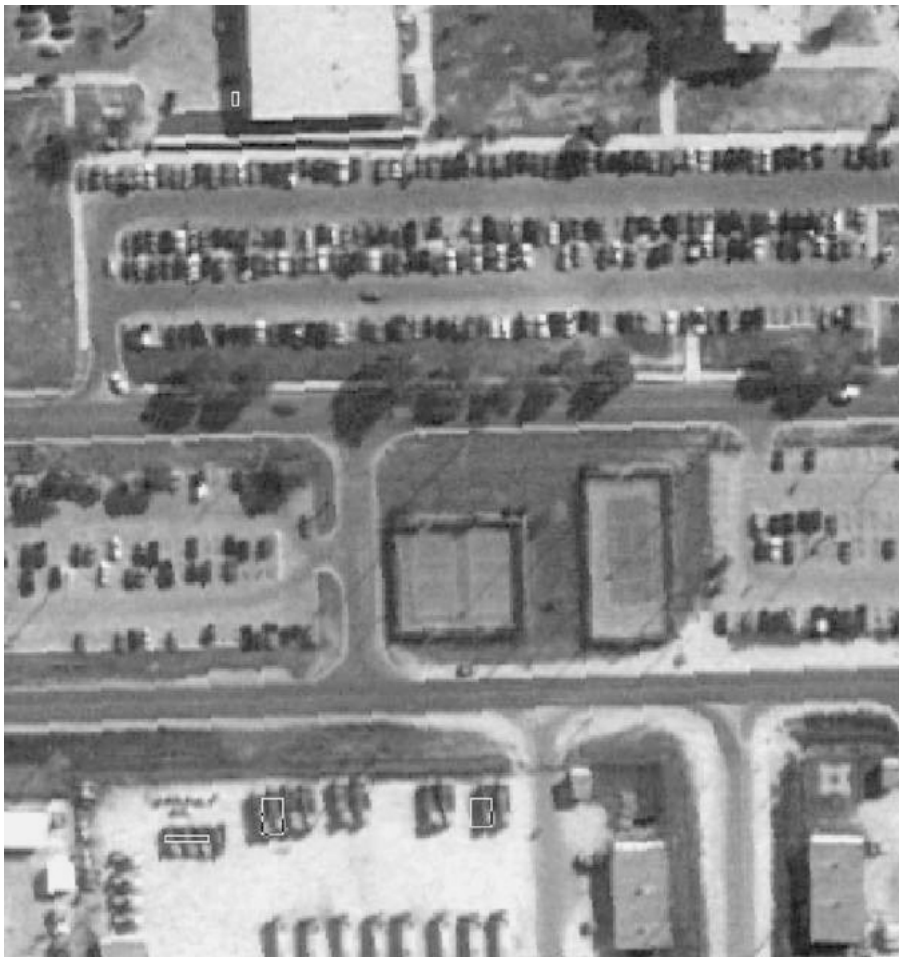

Figure 15. Changes detected in Panchromatic images, only two images used: Oblique view taken 21:19:49 on 10/27/1993.

method presented uses neural networks to model expected/unexpected change, without prior knowledge of what is expected/unusual. This provides significant benefits relative to model-based methods, as human effort is only needed to evaluate the identified changes.

There are disadvantages to the approach presented here. The types of changes identified are harder to control. Unusual changes are not always interesting, and sometimes the items of interest are not unusual. An example would be a forest that has been cleared—the method presented here would show trees left standing as the interesting change, not the area cleared. The method is probably best used in conjunction with manual analysis or other types of automated analysis, e.g., for:

– Prioritizing workload. If the data exceeds the human resources to process it, looking first at the unusual changes should provide the best cost/benefit.
– Identifying areas for further processing. Computationally intensive technologies, such as Automated Target Recognition (ATR) [35, 36], are best applied to small images likely to contain such a target. Passing changes to an ATR system can give these systems a more focused (and likely of interest) image "chip" to work on, and speed the search for *new* targets.

Although further testing is necessary to validate the method presented here, the success on two widely different types of imagery (low-resolution multispectral, high-resolution panchromatic) using the same parameters gives confidence that it will work in a wide variety

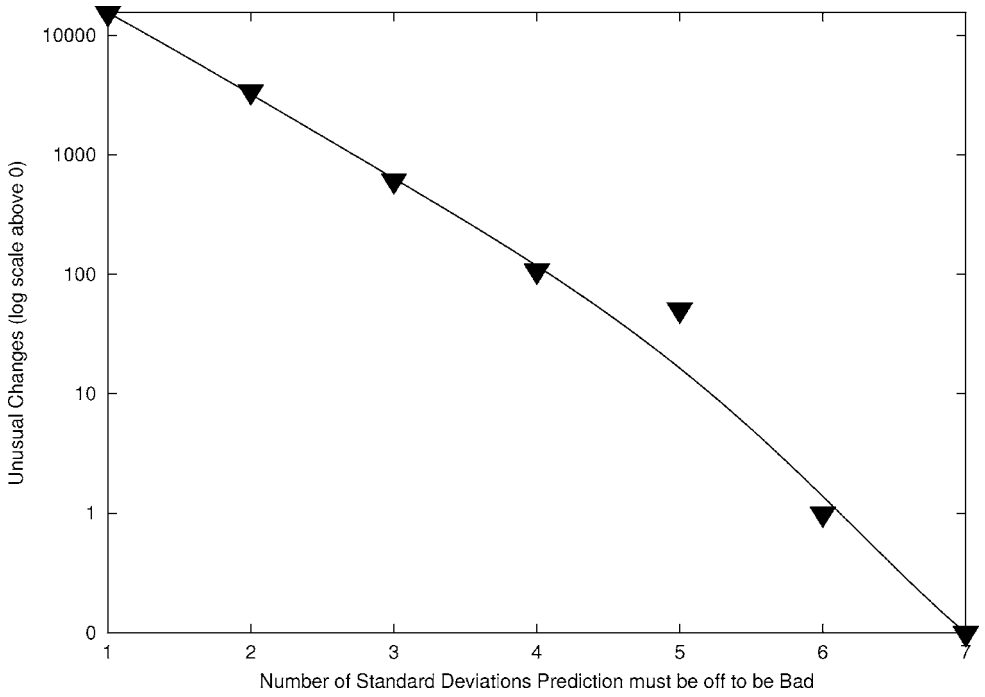

Figure 16. Number of unusual changes versus number of standard deviations beyond average error required to declare a bad prediction: Panchromatic imagery.

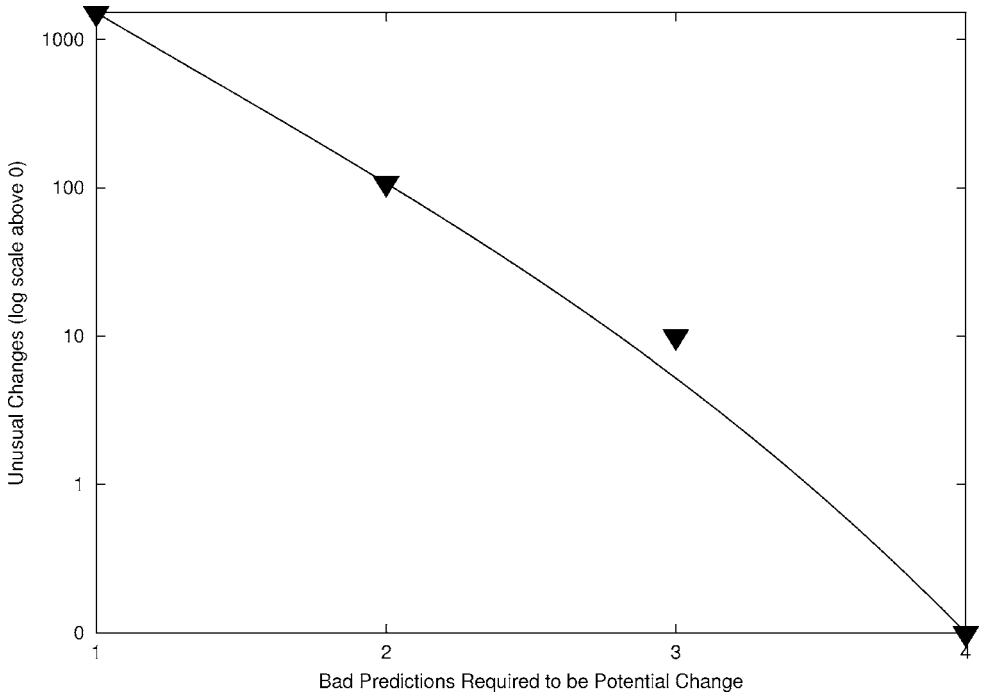

Figure 17. Number of unusual changes versus number of bad predictions required to declare a potential change: Panchromatic imagery.

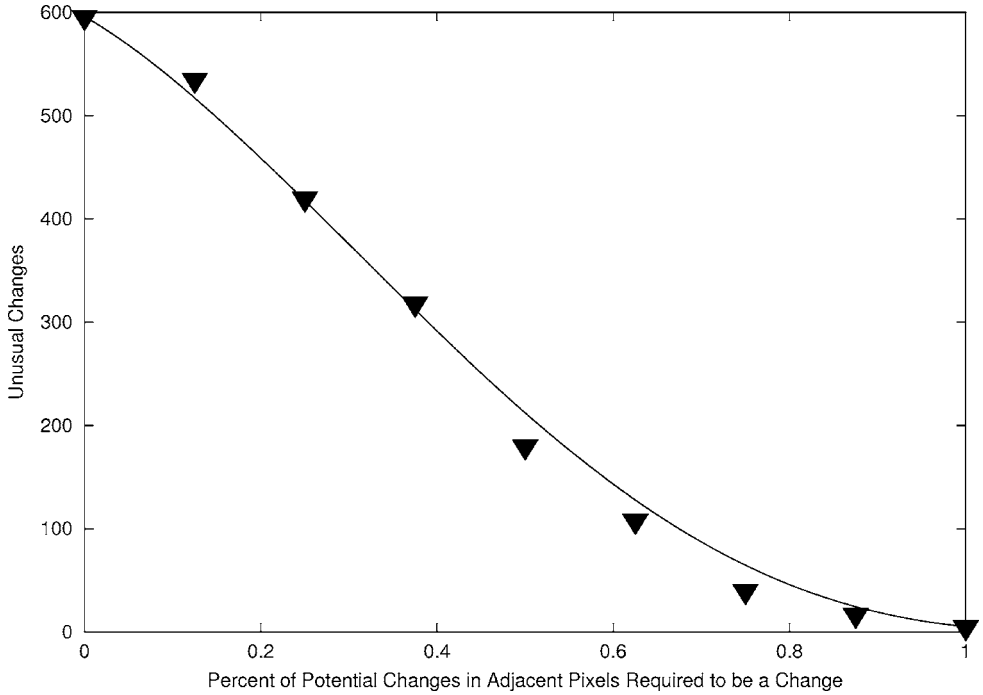

Figure 18. Number of unusual changes versus percent of potentially changed adjacent locations required to declare a change: Panchromatic imagery.

of conditions. One type of imagery that would be interesting for further tests would be Synthetic Aperture Radar (SAR) imagery modeled as a length two vector (real and imaginary parts of the complex value). There has been research specifically looking at SAR, and in many ways parallels visual imagery work (including model-based methods [37] and differencing [38, 39].) In addition to direct application to SAR, the approach presented here could be used to combine SAR and other types of imagery.

6.1. Further Work

The performance of the method presented here is crucial to its applicability. The primary issue is the neural network training time; all other steps are $O(n)$ (where n is the number of pixels). Training time on a 62712 pixel Landsat-TM image varied from 23150 to 50557 CPU seconds on an SGI 10K, averaging 35542 seconds. Training time per epoch/cycle is roughly linear in the number of pixels, but as the number of pixels increase, the number of cycles needed to converge increases. Tests of Thinking Machine Corporation's (now Oracle Corporation's) Darwin® [40] on a smaller image, resulted in substantial improvement (roughly 550 seconds, versus 33000 on comparable ULTRASparc hardware).

Another possible use would be on *sequences* of images. The goal would not be to identify individual changes between times, but areas undergoing constant change. Precisely what areas of constant change would mean, and how to identify them, remains to be determined.

One possible means of testing is to use synthetically-generated imagery, enabling a more thorough analysis of the limitations of this change detection method.

Acknowledgments

This work was sponsored by the MITRE-Sponsored Research Program, and was performed while the author was at The MITRE Corporation. The support of Mike Viszmeg was invaluable in selecting and preparing data for the experiments presented here.

References

1. R.L. Lillestrand, "Techniques for change detection," *IEEE Transactions on Computers*, vol. 21, no. 7, pp. 654–659, 1972.

2. I. Niemeyer, M. Canty, and D. Klaus, "Unsupervised change detection techniques using multispectral satellite images," in *Proceedings of the 1999 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'99)*, 1999, pp. 327–329.
3. S. Gopal and C. Woodcock, "Remote sensing of forest change using artificial neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 2, pp. 398–404, 1996.
4. P.J. Deer, "Digital change detection techniques: Civilian and military applications," in *ISSSR-95*, 1995.
5. O. Firschein and T.M. Strat (eds.), *RADIUS: Image Understanding for Imagery Intelligence*, San Francisco: Morgan Kaufmann, 1997.
6. X. Dai and S. Khorram, "Requirements and techniques for an automated change detection system," in *Proceedings of the 1998 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'98)*, vol. 5, 1998, pp. 2752–2754.
7. A. Huertas and R. Nevatia, "Detecting changes in aerial views of man-made structures," in *Proceedings of the Sixth International Conference on Computer Vision (ICCV98)*, 1998, pp. 73–79.
8. R. Chellappa, Q. Zheng, L.S. Davis, D. DeMenthon, and A. Rosenfeld, "Site-model-based change detection and image registration," in *DARPA93*, 1993, pp. 205–216.
9. S. Kuttikkad, R. Chellappa, and L. Novak, "Building 2-D wide area site models from single- and multi-pass single polarization SAR data," in *SPIE Aerosense '96*, Orlando, FL, 1996.
10. L. Bruzzone and D.F. Prieto, "A Bayesian approach to automatic change detection," in *Proceedings of the 1999 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'99)*, 1999, pp. 1816–1818.
11. H. Hanaizumi and S. Fujimura, "Change detection from remotely sensed multi-temporal images using multiple regression," in *Proceedings of the 1992 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'92)*, 1992, pp. 564–566.
12. T. Yamamoto, H. Hanaizumi, and S. Chino, "A change detection method for remotely sensed multi-spectral and multi-temporal images using 3-D segmentation," in *Proceedings of the 1999 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'99)*, 1999, pp. 77–79.
13. M.J. Carlotto, "Detection and analysis of change in remotely-sensed imagery with application to wide area surveillance," *IEEE Transactions on Image Processing*, vol. 6, no. 1, 1997.
14. V.K. Shettigara, "A generalized procedure for change detection and semi-automatic extraction of man-made objects from multispectral images," in *ISSSR-95*, 1995.
15. X. Dai and S. Khorram, "Development of a new automated land cover change detection system from remotely sensed imagery based on artificial neural networks," in *Proceedings of the 1997 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'97)*, vol. 2, Singapore, 1997, pp. 1029–1031.
16. J.T. Morisette and S. Khorram, "An introduction to using generalized linear models to enhance satellite-based change detection," in *Proceedings of the 1997 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'97)*, vol. 4, Singapore, 1997, pp. 1769–1771.
17. L. Bruzzone and D.F. Prieto, "Automatic analysis of the difference image for unsupervised change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1171–1182, 2000.
18. A. Schaum and A. Stocker, "Spectral/temporal detection of military targets in clutter," in *44th National IRIS*, 1997.
19. U.M. Fayyad, S.G. Djorgovski, and N. Weir, "Automating the analysis and cataloging of sky surveys," in *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, 1996, pp. 471–493.
20. M. Burl, U. Fayyad, P. Perona, P. Smyth, and M. Burl, "Automating the hunt for volcanoes on venus," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 1994.
21. T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997.
22. P.E. Stolorz and C. Dean, "Quakefinder: A scalable data mining system for detecting earthquakes from space," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland, OR, 1996, pp. 208–213.
23. D.P. Roy, "The impact of misregistration upon composited wide field of view satellite data and implications for change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, (no. 4 part 2), pp. 2017–2032, 2000.
24. L. Fonseca and B.S. Manjunath, "Registration techniques for multisensor remotely sensed imagery," *Photogrammetric Engineering and Remote Sensing*, vol. 62, no. 9, 1996.
25. X. Dai, S. Khorram, and H. Cheshire, "Automated image registration for change detection from landsat thematic mapper imagery," in *Proceedings of the 1996 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'96)*, vol. 3, 1996, pp. 1609–1611.
26. A. Can, C. Stewart, and B. Roysam, "Robust hierarchical algorithm for constructing a mosaic from images of the curved human retina," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 286–292.
27. X. Dai and S. Khorram, "Effects of image misregistration on the accuracy of remotely sensed change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 5, pp. 1566–1577, 1998.
28. J.R.G. Townshend, C.O. Justice, C. Gurney, and J. McManus, "Impact of misregistration on change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 5, pp. 1054–1060, 1992.
29. ERDAS, "ERDAS IMAGINE," http://www.erdas.com, 2001.
30. S. Fahlman, "Fast-learning variations on back-propagation: An empirical study," in *Proceedings of the 1988 Connectionist Models Summer School*, 1988, pp. 38–51.
31. P. Goodman, D. Rosen, L. McKnight, A. Purankar, and F. Sha, "NevProp4: Artificial neural network software for statistical prediction," http://www.scs.unr.edu/nevprop/, 1994.
32. C. Canus and J.L. Vehel, "Change detection in sequences of images by multifractal analysis," in *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-96)*, vol. 4, 1996, pp. 2172–2175.
33. landsat, "Multi-resolution land characteristics (MRLC)," Earth Resources Observation Systems (EPOS), Data Center, U.S. Geological Survey (USGS). Available at http://edcwww.cr.usgs.gov/Webglis/glisbin/guide.pl/glis/hyper/guide/mrlc, 2001.
34. fthood, "Ft. Hood Datasets," Model Based Vision Laboratory (MBVL), Air Force Research Laboratory (AFRL), United States Air Force. Available at https://www.mbvlab.wpafb.af.mil/ public/sdms/datasets/fthood/index.htm, 2001.

35. ATR, "Special issue—automatic target recognition," *Neural Networks*, vol. 8, nos. 7/8, pp. 1005–1358, 1995.
36. Nassar M. Nasrabadi and Aggelos K. Katsaggelos (eds.), "Session 1: Use of Neural Networks for Automatic Target Recognition, Classification, and Characterization," in *Applications of Artificial Neural Networks in Image Processing IV*, vol. 3647, SPIE, 1999, pp. 2–57.
37. R.G. White, and C.J. Oliver, "Change detection in SAR imagery," in *Proceedings of the 1990 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'90)*, 1990, pp. 217–222.
38. D.J. Wehdahl, "Change detection in SAR images," in *Proceedings of the 1991 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'91)*, 1991, pp. 1421–1424.
39. M. Bao, "Backscattering change detection in SAR images using wavelet techniques," in *Proceedings of the 1999 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'99)*, 1999, pp. 1561–1563.
40. Darwin, "Oracle Darwin Data Mining Software," http://www.oracle.com/ip/analyze/warehouse/datamining/, 2001.

Chris Clifton has recently joined Purdue University as an Associate Professor of Computer Science. While this work was being performed he was a Principal Scientist at The MITRE Corporation. He has also held a position as an Assistant Professor of Computer Science at Northwestern University. He has a Ph.D. from Princeton University, and Bachelor's and Master's degrees from the Massachusetts Institute of Technology. His research interests include data mining, data security, database support for text, and heterogeneous databases.