Journal of
Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

Object-based classification of earthquake damage from high-resolution optical imagery using machine learning

James Bialas
Thomas Oommen
Umaa Rebbapragada
Eugene Levin

SPIE.

James Bialas, Thomas Oommen, Umaa Rebbapragada, Eugene Levin, “Object-based classification of earthquake damage from high-resolution optical imagery using machine learning,” *J. Appl. Remote Sens.* **10**(3), 036025 (2016), doi: 10.1117/1.JRS.10.036025.

Object-based classification of earthquake damage from high-resolution optical imagery using machine learning

James Bialas,[a,*] Thomas Oommen,[a] Umaa Rebbapragada,[b] and Eugene Levin[c]

[a]Michigan Technological University, Geological and Mining Engineering and Sciences,
1400 Townsend Drive, Houghton, Michigan 49931, United States

[b]Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive,
Pasadena, California 91109, United States

[c]Michigan Technological University, School of Technology, 1400 Townsend Drive,
Houghton, Michigan 49931, United States

Abstract. Object-based approaches in the segmentation and classification of remotely sensed images yield more promising results compared to pixel-based approaches. However, the development of an object-based approach presents challenges in terms of algorithm selection and parameter tuning. Subjective methods are often used, but yield less than optimal results. Objective methods are warranted, especially for rapid deployment in time-sensitive applications, such as earthquake damage assessment. Herein, we used a systematic approach in evaluating object-based image segmentation and machine learning algorithms for the classification of earthquake damage in remotely sensed imagery. We tested a variety of algorithms and parameters on post-event aerial imagery for the 2011 earthquake in Christchurch, New Zealand. Results were compared against manually selected test cases representing different classes. In doing so, we can evaluate the effectiveness of the segmentation and classification of different classes and compare different levels of multistep image segmentations. Our classifier is compared against recent pixel-based and object-based classification studies for postevent imagery of earthquake damage. Our results show an improvement against both pixel-based and object-based methods for classifying earthquake damage in high resolution, post-event imagery. *© 2016 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JRS.10.036025]

Keywords: GEOBIA; machine learning; earthquake; object-based classification.

Paper 16085 received Feb. 10, 2016; accepted for publication Aug. 30, 2016; published online Sep. 21, 2016.

1 Introduction

Earthquakes are a major natural disaster that can cause significant loss of life and property damage. The dangers are not limited to the immediate event. Damage to manmade structures can further endanger the public and emergency responders as a result of structural instability that may be intensified by following aftershocks. Damage to roads and other infrastructure can hamper response by emergency responders as well as evacuation of the public from affected areas.[1]

A clear and accurate picture of both the intensity and the extent of the damage is an important tool in organizing emergency response to an earthquake.[2] Imagery from earth observation platforms has shown much promise in this role.[3] However, there is room for improvement in the classification accuracy of earthquake damage before it can be cataloged and acted upon by emergency responders.

Remote sensing technologies, including imagery from Earth observation systems, have a long history of use in identifying and assessing earthquake damage. With regards to

*Address all correspondence to: James Bialas, E-mail: jpbialas@mtu.edu two-dimensional imagery, classification has been used to quantify, assess, and locate damage within. Traditionally, this classification has been done by human operators or increasingly with pixel-based classifiers applied to multispectral satellite imagery. As satellite sensors increase in resolution and aerial platforms, such as unmanned aerial vehicles provide increasingly high spatial resolution imagery, this presents a benefit in the form of more detailed maps but also a classification challenge. As spatial resolution increases, often with reduced spectral resolution, traditional pixel-based classifiers become less effective.[4]

While there is a vast body of work discussing the theory and application of object-based image analysis, there is limited work in the task of classifying earthquake damage in remotely sensed imagery, particularly when considering only postevent imagery without any sort of digital elevation model or height information. Reference 3 provides a thorough overview of the field of earthquake damage detection with remote sensing by surveying existing work. Studies making use of aerial imagery, satellite imagery, LiDAR data, SAR data, and ancillary data, such as building vector data and other GIS maps are considered. Overall, most studies achieve somewhere in the range of 70% to 90% accuracy in damage identification using either pixel-based or object-based methods. Studies that consider both pre- and postevent data as well as studies that make use of multiple data sources tend to provide better results. However, for studies that look at postevent imagery only, results tend to improve only as spatial resolution increases.[3]

Reference 5 takes a pixel-based approach looking at 0.61-m pan-sharpened multispectral Quickbird imagery of the 2010 Haitian earthquake. Using a pixel-based method with a support vector machine (SVM) classifier, they achieve 81.5% overall accuracy with 63.4% user accuracy and 71.3% producer accuracy for the damage class, where results were generated from an independent validation set.[5] Using similar imagery, Ref. 6 published the best object-based analysis results. This study is based on Quickbird panchromatic data of 0.61-m resolution and pan-sharpened multispectral imagery of the Wenchuan earthquake of May 12, 2008, in China. Using a watershed-based multilevel segmentation, a single class SVM classifier, and features such as spectral values and moment invariants, they show 60% producer accuracy and 91% user accuracy in building damage detection with 79% overall classification accuracy.[6]

Numerous studies comparing pixel and object-based results have also been completed using a variety of data from various applications. Some are applicable to our specific study as they compare object- and pixel-based results using either different methods or applications. Reference 7 compared pixel-based and object-based methods to data obtained from a visual inspection of pre-event and postevent Quickbird imagery. Data was from the 2003 Boumerdes, Algeria earthquake. Unfortunately, results were not very good with damage producer/user accuracies of 32%/23% for pixel-based and 50%/20% for object-based methods. Maximum likelihood was used for the pixel-based classification and nearest neighbor for the object-based approach.[7]

Reference 8 performed a thorough investigation of object versus pixel results within the context of high urban density classification. The study data consisted of Quickbird imagery of both Phoenix and Tempe, Arizona used for the classification of buildings, vegetation, lakes, impervious surfaces, and others. The pixel-based classifier is a traditional maximum likelihood classifier. The object-based method is more complex, using a multilevel segmentation along with nearest neighbor or rule-based classifiers for different classes. Ultimately, for the test image of Tempe, overall accuracies of 87.8% for pixel-based and 95.2% for object-based classifications were achieved when compared against manually delineated validation data sets created through visual interpretation. In the larger Phoenix image, the results were not quite as good, but it is important to consider in our work that the worst case building producer/user classification results were 50%/81.25% for pixel-based and 83.91%/91.25% for object-based.[8]

While the literature seems to indicate that for higher resolution images, object-based methods will produce better results, a robust comparison has yet to be done in the context of earthquake damage. As Ref. 5 demonstrates, pixel-based methods are already a reasonable means of damage classification. We propose that a systematic approach to object-based classification can improve the results in earthquake damage detection over pixel-based methods. Furthermore, a systematic approach to parameter selection in object-based classification can improve the results of the classifier over subjective methods.

2 Study Area and Data

In this paper, we consider the example of the Christchurch, New Zealand, 6.2 magnitude earthquake that occurred February 22nd, 2011. 181 lives were lost and 900 buildings and 10,000 residential structures were destroyed. Damages from the event was estimated at NZ $15-20 billion.[9] The study area and earthquake impacts are depicted in Fig. 1.

Postevent imagery was obtained from Land Information New Zealand.[10] Imagery was obtained on the 24th of February by New Zealand Aerial Mapping Limited at the request of the Christchurch Response Center. The imagery has a 10-cm per pixel spatial resolution and comprises red, green, and blue spectral bands. The resulting orthophotos were generated

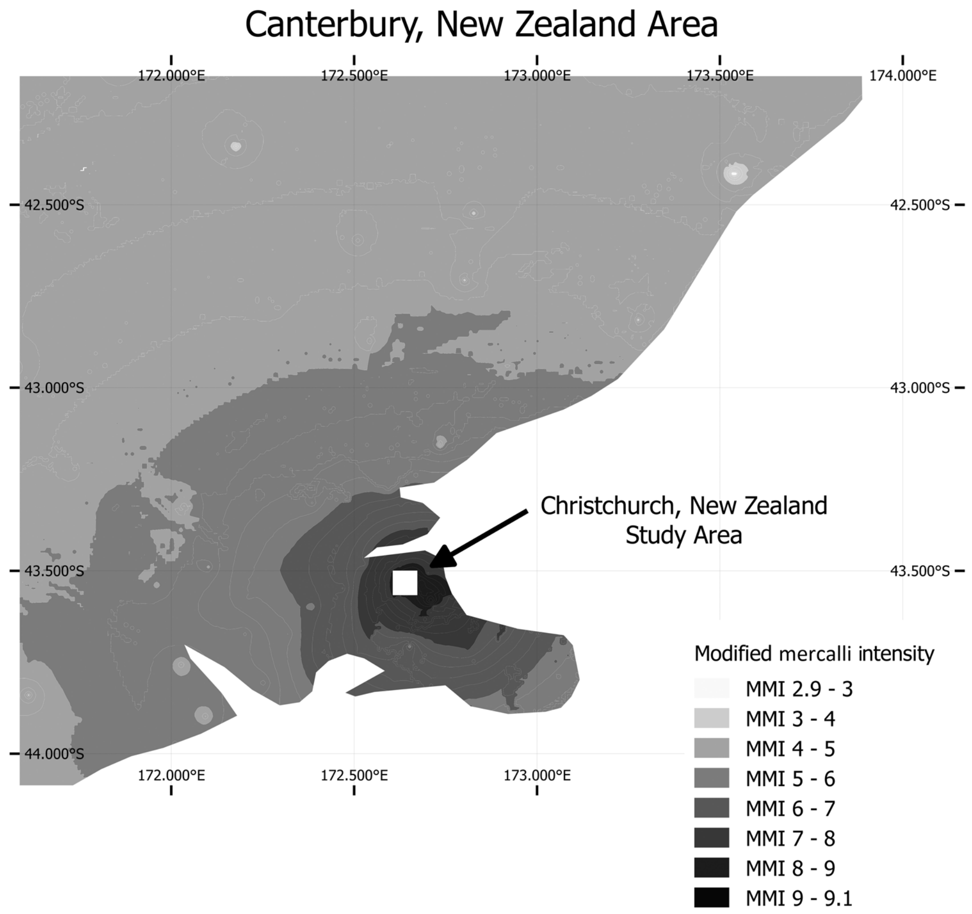

Fig. 1 Earthquake intensity in Canterbury, New Zealand, area and study area in Christchurch, New Zealand.

Table 1 Number of training and validation pixels, as well as objects covered at 75% overlap. Overlap parameters and their impact on results are addressed in Sec. 4.

Class	Training pixels	Validation pixels	Training objects
Building	1,561,546	180,232	983
Pavement	304,553	165,393	114
Vehicle	12,532	9222	19
Vegetation	42,920	69,150	13
Rubble	314,096	235,490	252

from a pre-existing digital elevation model (DEM) and were not checked against ground truth to verify if there was any earthquake damage that may not be accounted for in the DEM.[10] This study is based on tiles 1-0003-0002 and 1-0003-0003. Tiles were projected to WGS_84_UTM_zone_59S and combined into one image in ArcGIS. The resulting image is 6335×8393 pixels in size.

Training and validation data were generated manually in ArcGIS by a single operator. Five classes were able to cover all the land cover seen in the image: building, pavement, vehicle, vegetation, and rubble. The operator selected clearly identifiable examples of different classes. Polygons were used to delineate classes. Generally, polygons were drawn at the edges of visually recognizable objects, such that the polygon enclosed the object as best as possible without including extraneous pixels. This was somewhat challenging with the rubble class as the delineations were not always very clear. It was also impossible with the pavement class as pavement objects are all connected and form a large contiguous recognizable object in the image. As such, sample polygons were drawn only around clear sections of pavement. A set of polygons was created and a randomly chosen subset split off to form the validation data set. Table 1 outlines the number of training samples used both in terms of number of pixels as well as number of objects that result from segmentation.

3 Methods

3.1 *Pixel-Based Classification*

All results are compared against a typical pixel-based classification. Pixel-based classification techniques are highly established. Although many variations of pixel-based methods exist, classification of very high-resolution imagery based purely on spectral values using an SVM classifier is an established method for similar imagery and situations.[5,6] We compare systems using the same training data and classification hierarchies. Final results are compared on a per pixel basis for both object-based and pixel-based methods. Reasoning for this comparison method is further discussed in Sec. 3.3.

We use the Orfeo Toolbox, an open source package, to do our pixel-based classification and accuracy assessment. Figure 2 shows the output of a pixel-based classifier using the Orfeo Toolbox. The workflow for a pixel-based classification is similar to the steps for an object-based classifier outlined in Sec. 3.2, but we ignore the segmentation step.

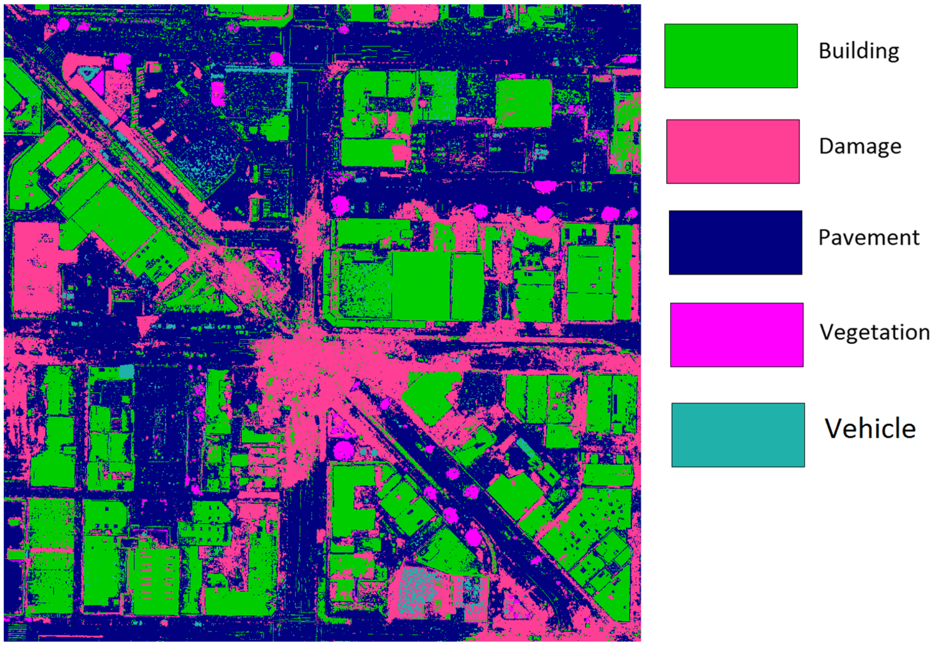

Fig. 2 Output results of a pixel-based classifier delineating the different classes.

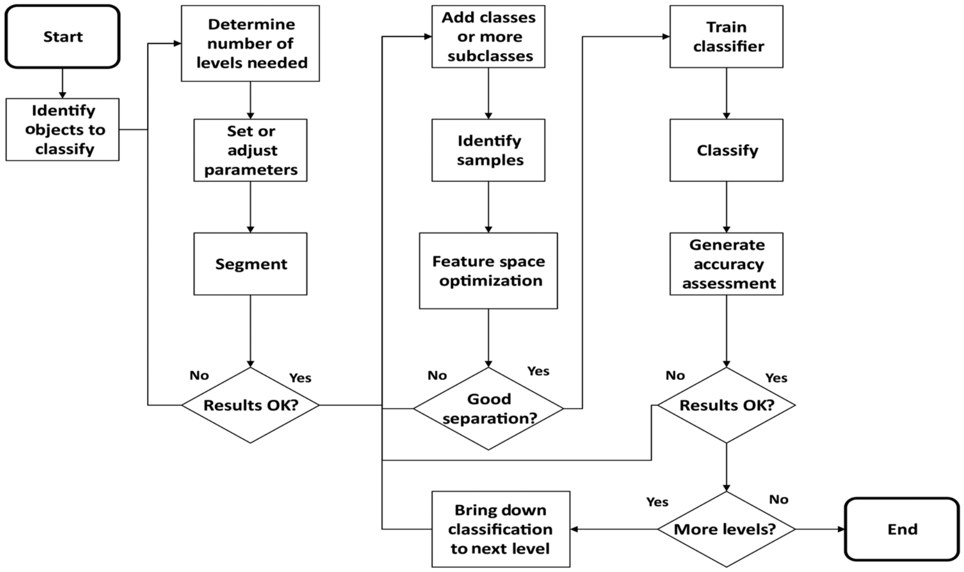
Start
Identify objects to classify
Determine number of levels needed
Set or adjust parameters
Segment
Results OK?
No
Yes
Add classes or more subclasses
Identify samples
Feature space optimization
Good separation?
No
Yes
Train classifier
Classify
Generate accuracy assessment
Results OK?
No
Yes
More levels?
Yes
No
Bring down classification to next level
End

Fig. 3 Flow chart representing a systematic approach to building an object-based classifier.

3.2 *Object-Based Classification*

Given the wide-ranging objectives, a classification system needs to be able to accommodate numerous variables such as imagery type, classifications needed, feature importance, and training data quality. There are four phases in this system: planning, segmenting, sampling, and classifying. The latter three phases require human interpretation of the results. Based on this interpretation, modifications can be made to improve the results before moving on. Figure 3 provides a flowchart outlining these phases.

In the planning phase, we identify the items in the image we wish to classify. It is important to not only consider the object of interest, rubble in our case, but all readily identifiable objects. By classifying as many objects as possible, we can attempt to achieve more readily distinguishable objects to aid in classification later. It is also important to consider how items differ in spectral, spatial, and texture values. A strategy that aims to classify objects that are most different in these categories will be more successful.

All portions of the object-based classifier are performed with the eCognition software package. Segmentation, feature generation, and optimization, as well as classification, are completed using built in software functions. We perform our final results evaluation by exporting a classified image and doing a pixel-based comparison with the validation data using the Orfeo Toolbox.

3.2.1 *Segmentation*

In the segmentation phase, we use eCognition's multiresolution segmentation algorithm to delineate the image into objects for classification. This segmentation algorithm is a merging algorithm. When starting from an unsegmented image, single pixels are considered first. A merging cost is calculated for each possible merge, known as the degree of fitting. If the result is less than the least degree of fitting calculated by the algorithm parameters, a merge is performed. Objects are continually merged until no merges are possible given the initial parameters. For subsequent levels after the base image, the input is the segments from the previous level that are then merged until the given parameters are met.[11]

Segmentation is driven by three main parameters. The most important is the scale parameter, which drives the size of the resulting segments. Scale represents the average size in pixels of the resulting objects. The shape and compactness numbers range from 0 to 1. Shape determines how much influence color versus shape has on the segmentation. A higher value means a lower influence of color. The resulting influence of shape is then further influenced by the compactness parameter. A higher compactness value results in more compact objects while a lower value results in objects with smoother borders. We use shape and compactness parameters of 0.5 and 0.5.

An important feature of eCognition's multiresolution segmentation is its ability to create multilevel segmentations. The operator must decide on the number of levels needed. Ideally, the items we wish to classify will be perfectly delineated by the object boundaries and comprise a single object. If the objects of interest differ in size to a significant degree, several levels of segmentation may be needed using different parameters at each level. Then, independent classifications can be run at each level. Objects at upper levels are large objects comprised of smaller objects at the next lowest level. A logical relationship between superobjects and subobjects at lower levels is maintained, so classifications can be easily shared between levels. Although several tools have been developed to automate some of the parameter selection,[12] human interpretation of the results and subsequent adjustments to the parameters is still a common practice that yields acceptable results. Suitable segmentation is achieved by adjusting the parameters such that we minimize the number of image objects that comprise a physical item in the image while avoiding objects that span to areas outside of said physical object. Figures 3–5 provide examples of segmentation results at different scale parameters.

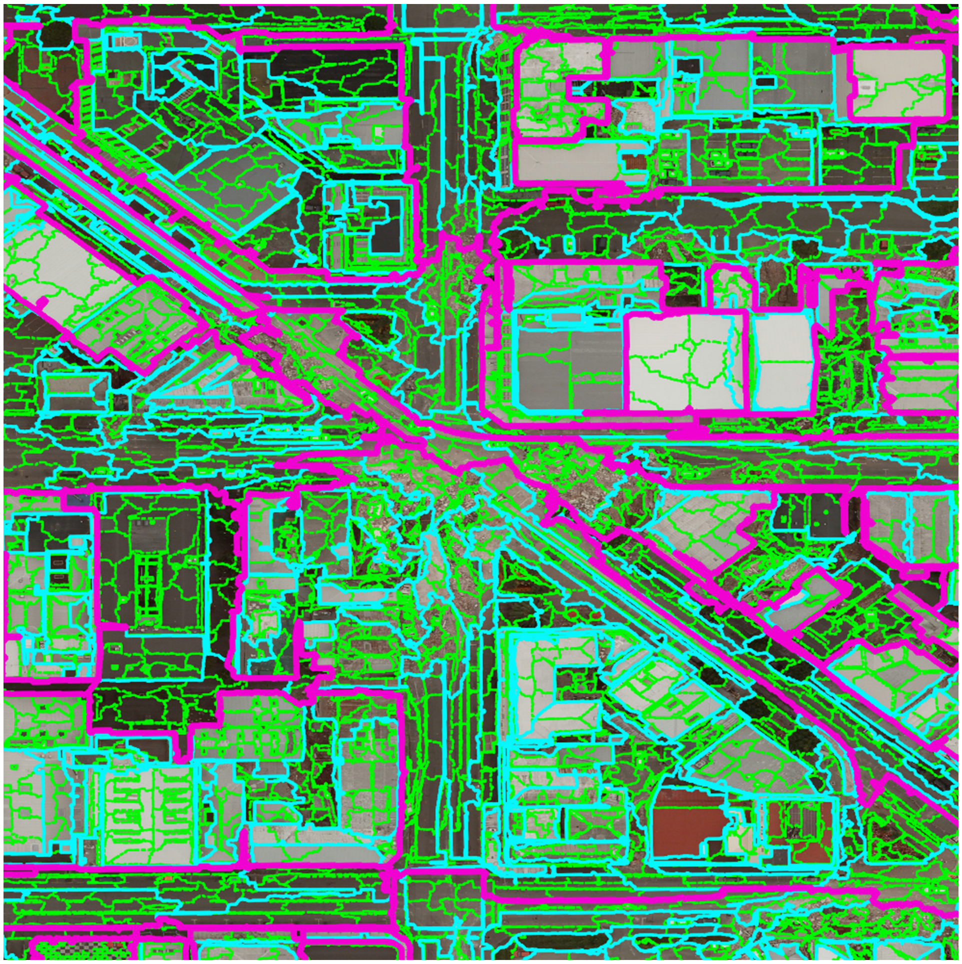

Fig. 4 Example of different scale parameter and segmentation results. A scale parameter of 50 (denoted by green) results in an oversegmented image while a scale parameter of 900 (denoted by pink) results in an undersegmented image. A scale parameter of 200 (denoted by cyan) most closely delineates physical objects.

3.2.2 *Sampling*

In the sampling phase, we identify samples of classes to use in training a supervised classification algorithm. It is important to create training samples for not only the classes we are interested in (rubble and buildings) but also to identify training samples of all recognizable object classes within the whole image. The ideal result is a set of classes that are highly separable by some set of features. We consider a set of 66 features, comprising spectral, geometrical, and textural values, and extract the 10 best features. The 66 features are a subset of preconfigured features available in eCognition. While eCognition offers more features, many of these require configuration in regards to the spectral bands available or parameters specific to the scene being evaluated.

To extract the 10 best features, we use eCognition's feature space optimization (FSO) tool. By providing a set of training samples and feature names, the FSO tool determines which features provide the best class separability. A list of the features considered is listed in Table 2. Based on the experience, separation values of greater than or equal to one provide good classification results. We also consider that it is important to observe a nondecreasing separation curve, which means that the addition of features does not improve the classifier performance. A curve that is not nondecreasing may indicate problems with the sample selection and lead to poor classification results. Typically, adding or removing random samples resolves this issue in

Table 2 All features considered by the FSO tool.

		Geometry			
Layer	Texture	Extent	Shape	Based on polygons	Based on skeletons
Brightness	GLCM homogeneity	Area	Asymmetry	Area (excluding inner polygons)	Average branch length
Mean value	GLCM contrast	Border length	Border index	Area (including inner polygons)	Average area represented by segments
Standard deviation	GLCM dissimilarity	Length	Compactness	Average length of edges (polygon)	Curvature/length (only main line)
Max. diff.	GLCM entropy	Length/ thickness	Density	Compactness (polygon)	Degree of skeleton branching
Hue	GLCM angular second momentum	Length/ width	Elliptic fit	Length of longest edge (polygon)	Length of main line (no cycles)
Saturation	GLCM mean	Number of pixels	Main direction	Number of edges (polygon)	Length of main line (regarding cycles)
Intensity	GLCM Std. Dev.	Thickness	Radius of largest enclosed ellipse	Number of inner objects (polygon)	Length/width (only main line)
	GLCM correlation	Volume	Radius of smallest enclosed ellipse	Perimeter (polygon)	Maximum branch length
	GLDV angular second momentum	Width	Rectangular fit	Polygon self-intersection (polygon)	Number of segments
	GLDV entropy		Roundness	Std. dev. of length of edges	Std. dev. curvature (only main line)
			Shape index		Std. dev. of area represented by segments
					Width (only main line)

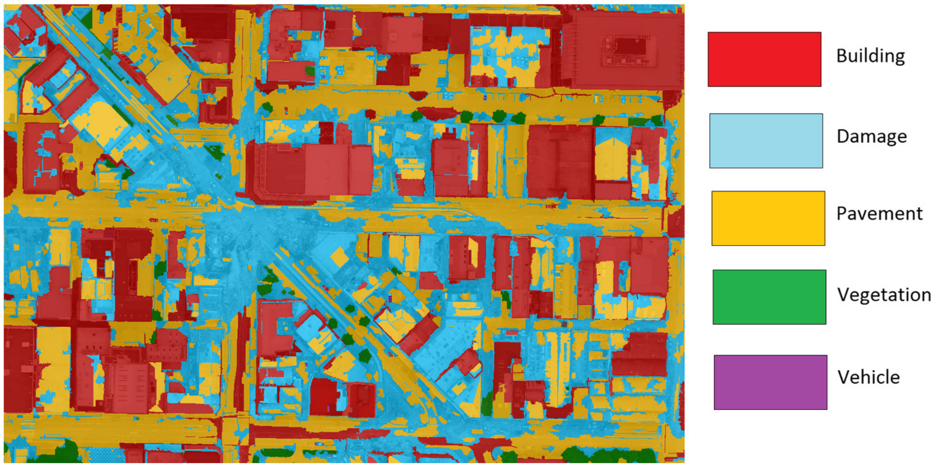

Fig. 5 Output results of an object-based classifier delineating the different classes.

one or two tries. If poor separation results are achieved, we return to the start of the sampling phase and build upon the sample selection by adding more classes or subclasses to improve segment separation until a nondecreasing separation curve is observed. Features selected in this study for segmentation levels 2 and 4 are shown in Table 5.

3.2.3 *Object-based classification*

In the classifying phase, we use the features and samples identified in the previous step to classify objects from the planning phase. Training segments and features are input into a training algorithm. The output classifier is then used to predict the label of new image segments. Different classification algorithms can be assessed and their parameters adjusted to improve performance. From there, an accuracy assessment can be generated in eCognition against either the existing training samples or a separate validation dataset. If the results are not acceptable, we return to the sample phase to further refine the training inputs or the classifier parameters. If the results are accessible, we apply them to any lower levels in the segmentation hierarchy and start the process over again on the next level below the current level if such exists. Below is the classification map for all levels (Fig. 5).

3.3 *Evaluation of Classifier Performance*

There are two important considerations to be made when comparing object-based and pixel-based results. First, as seen in Fig. 7, image objects may not represent training and validation data the same as pixels, eliminating the possibility of a fair comparison. Pixels considered in a pixel-based evaluation as part of one class may be considered as a part of a different class in object-based results depending on where the image object boundaries fall. Second, because the image objects differ in the number of pixels they contain, especially in a multilevel object-based classification, they must be weighted to ensure a fair comparison with pixel-based results. The easiest way to address both concerns is to evaluate both pixel-based and object-based results on a per-pixel basis in the final classification maps they produce.[8]

Image segments were assigned to the appropriate training class on level 2 based on how much the segment overlaped the training polygon. Different mounts of overlap were considered with 75% of pixels yielding the best results. Level 4 training data were assigned based on whether the level 4 object contained any subobjects of either the building class or classes other than building. If at least one subobject was classified on level 2, the entire level 4 object was assigned the class. In the case of conflicting subobjects, the building class would prevail.

Results of a classification process are compared using a confusion matrix (as seen in Tables 3 and 4). A confusion matrix is a table with actual classification in rows, and the predicted classification in columns. As such, the diagonal of this chart represents correct predictions. Items in the columns that do not match are known ground truth items that have been misclassified by the classifier.

For a given class, the accuracy values can be calculated as such

$$\text{Produce accuracy} = \frac{\text{Correctly classified items}}{\text{Sum of all items in validation sample (row)}},$$

$$\text{User accuracy} = \frac{\text{Correctly classified items}}{\text{Sum of all items as classified (column)}}.$$

We calculate the overall accuracy for the entire classification by averaging the producer and user accuracies for all classes.[13]

User accuracy can be considered as a measure of the reliability of classified pixels in the image. User accuracy measures the number of correctly classified pixels compared to the total number of validation pixels for that class. Given an 88% user accuracy for the rubble class in our object-based classifier, we can state that 88% of the pixels classified as rubble in the output map are correctly classified.

Producer accuracy considers the opposite scenario; it is a measure of how accurately the classifier predicted pixels in the validation data as compared to the total of validation pixels. Given a 94% producer accuracy for the rubble class in our object-based classifier, we state that 94% of the ground truth pixels were correctly classified.

Finally, we consider classifier performance on rubble class pixels. Damage inflicted by earthquakes can take many forms; we use the existence of the rubble class as an indicator of areas affected by an earthquake. While it is important to consider overall performance in general image classification problems, in the context of earthquake damage, we are only concerned with damaged versus undamaged areas. While we do consider multiple classes to improve the classification of undamaged areas, ultimately it is a binary problem of damaged and undamaged areas we are considering.

4 Results and Discussion

When considering postevent 10-cm RGB orthophotos, our pixel-based classification system produced a 62% overall accuracy and rubble user and producer accuracies of 88% and 62%. Our object-based approach ultimately improved this to 77% overall accuracy with rubble user and producer accuracies of 88% and 94%. Both cases were evaluated at the pixel level using an independent validation data set. Producer and user accuracies for different classes are shown in Fig. 6 for both object-based and pixel-based classifications. The corresponding confusion matrices can be found in Tables 3 and 4.

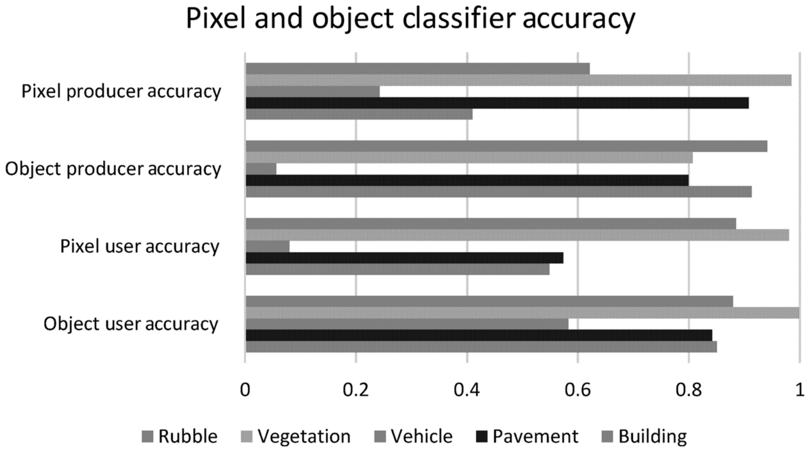

Fig. 6 Object-based and pixel-based results as compared against validation data.

Table 3 Confusion matrix for object-based classifier.

	Building	Pavement	Vehicle	Vegetation	Rubble	Sum
Building	164,640	110,59	2	0	4504	180,205
Pavement	20,945	132,207	0	0	12,213	165,365
Vehicle	2616	1599	524	0	4512	9251
Vegetation	0	4141	0	55,680	9171	68,992
Rubble	5319	7980	373	1	221,726	235,399
Sum	193,520	156,986	899	55,681	252,126	
User accuracy	0.85	0.84	0.58	1.00	0.88	
Producer accuracy	0.91	0.80	0.06	0.81	0.94	
Average	0.77					

Table 4 Confusion matrix for pixel-based classifier.

	Building	Pavement	Vehicle	Vegetation	Rubble	Sum
Building	76,974	79,392	18,558	735	11,885	187,544
Pavement	6239	156,428	2220	8	7355	172,250
Vehicle	3165	3511	2318	91	472	9557
Vegetation	227	753	0	70,968	72	72,020
Rubble	53,718	32,666	5828	567	152,535	245,314
Sum	140,323	272,750	28,924	72,369	172,319	
User accuracy	0.55	0.57	0.08	0.98	0.89	
Producer accuracy	0.41	0.91	0.24	0.99	0.62	
Average	0.62					

Table 5 Features returned by the FSO tool used in object-based classifier.

Level four features	Level two features
(Extent) Length/width	(Layer value) mean GREEN
(Shape) compactness	(Layer value) saturation
(Shape) main direction	(Extent) area
(Based on polygons) average length of edges	(Shape) asymmetry
(Based on skeletons) degree of skeleton branching	(Shape) main direction
(Based on skeletons) length of mainline (regarding cycles)	(Shape) radius of largest enclosed ellipse
(Texture) GLCM contrast	(Based on polygons) number of edges
(Texture) GLCM correlation	(Texture) GLCM entropy
(Texture) GLDV mean	(Texture) GLCM stddev
(Texture) GLDV contrast	

Ultimately, the systematic approach (outlined in Sec. 3.2) resulted in a multilevel segmentation comprised of four levels using scale parameters of 20, 50, 100, and 200. All the levels used shape and compactness factors of 0.5. Classification was ultimately carried out on the fourth level (scale parameter 200) for buildings, and second level (scale parameter 50) for classes of pavement, rubble, vehicle, and vegetation. A Naïve Bayes classifier was used on level four and an SVM on level 2 using the default linear kernel with c parameter of 2. Features used in training the classifier are listed in Table 5.

The feedback mechanism outlined in Sec. 3.2 provides several opportunities to improve upon our results. The first to consider is object selection. In order to provide a more similar comparison between object- and pixel-based results, training samples are taken from polygons in shape files. As shown in Fig. 7, the resulting objects do not always line up exactly with the polygons and some criteria must be used to decide if an object should be classified as a training object or not. We considered percentage of overlap between the object and the polygon. Classification results are compared when looking at overlap of training polygons with image objects of 0%, 25%, 50%, 75%, and 100%. As we can see in Figs. 8 and 9, different classes benefit differently

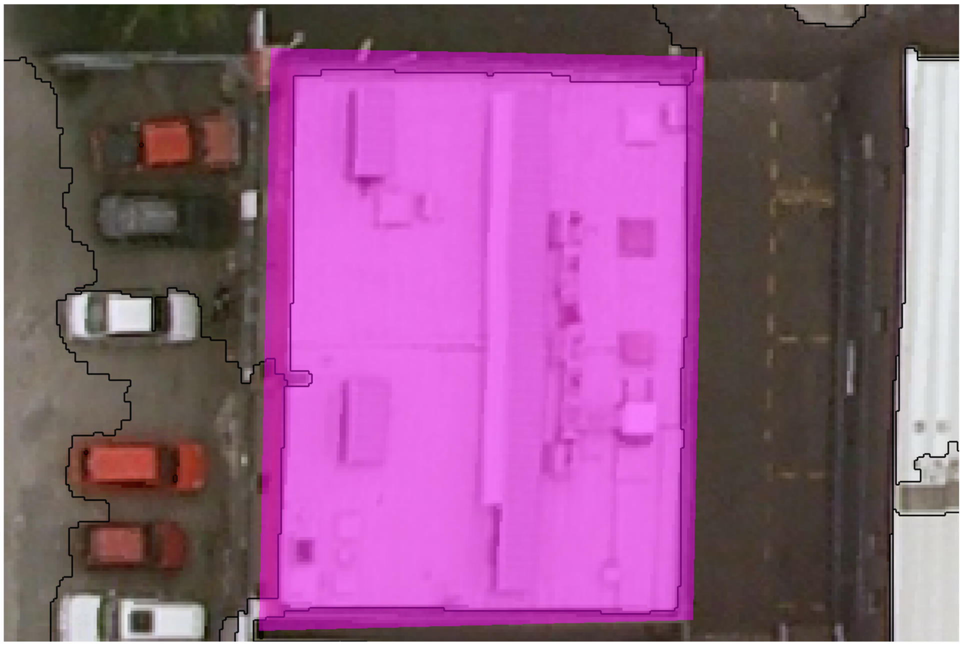

Fig. 7 Example of a training polygon showing overlap with image objects.

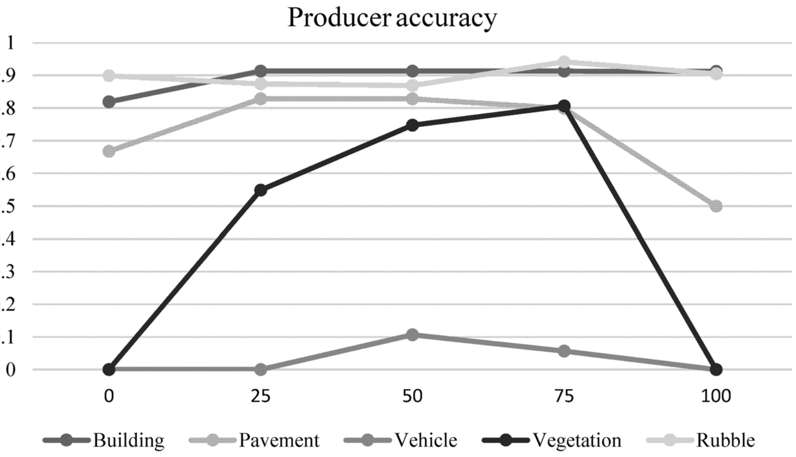

Fig. 8 Producer accuracies measured at various overlap parameter percentages.

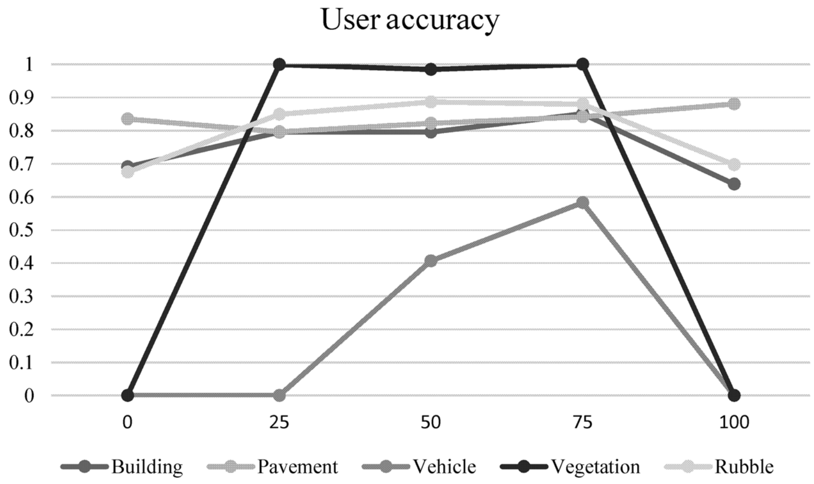

Fig. 9 User accuracies measured at various overlap parameter percentages.

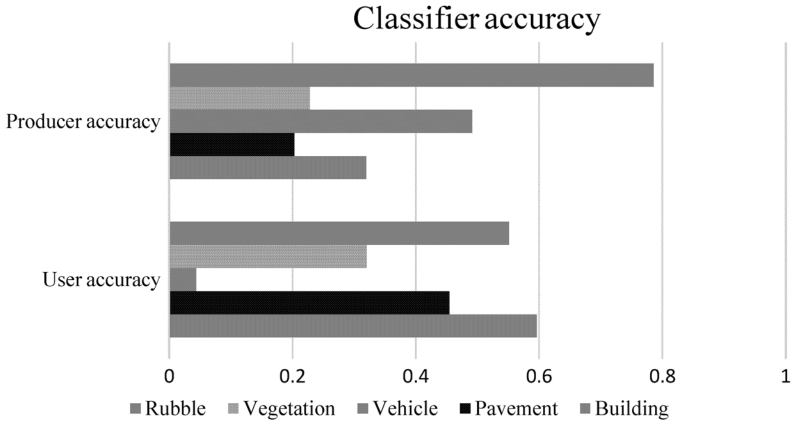

Fig. 10 Performance of an object-based classifier that only consider objects from level 1 to level 2.

from the various settings. An overlap of 0% would still imply at least some degree of overlap, but the extent of which does not matter.

We also considered multilevel versus single level classification. Instead of classifying buildings on one level and everything else on another level of smaller objects, we classified everything on level 2. Notice how building classification goes from 91% producer accuracy to 32% (as shown in Figs. 5 and 10, respectively) when classifying them at an inappropriate level.

In an attempt to improve class separability and overall classifier performance, we break down the rubble class into four individual subclasses: building chunks-rubble that contained visibly identifiable pieces of building, high density-rubble that contained discernible pieces of debris, low density (a class for rubble with no discernable contents), and sticks (a class used to identify debris containing long structural elements, such as steel beams or lumber). Breaking classes up into subclasses offers an encouraging boost in performance when comparing results against the training data. This is likely because a smaller subclass allows greater over fitting. As would be expected in over fitting, comparison against the validation data shows poorer results as shown in Fig. 11.

Choice in classifier algorithm and parameters can have a significant impact on results. Techniques for seeking improvements in results revolve around the reduction of over fitting, this is readily apparent when the classifier does very well on the training data, but returns poor results when looking at the validation data. eCognition offers five different classifier training algorithms: decision trees, random trees, SVMs, k-nearest neighbor, and Naïve Bayes. We test the classifiers against the validation data both at levels 2 and 4 as well as tuned versions of these algorithms intended to reduce over fitting. To try and reduce over fitting we adjust the

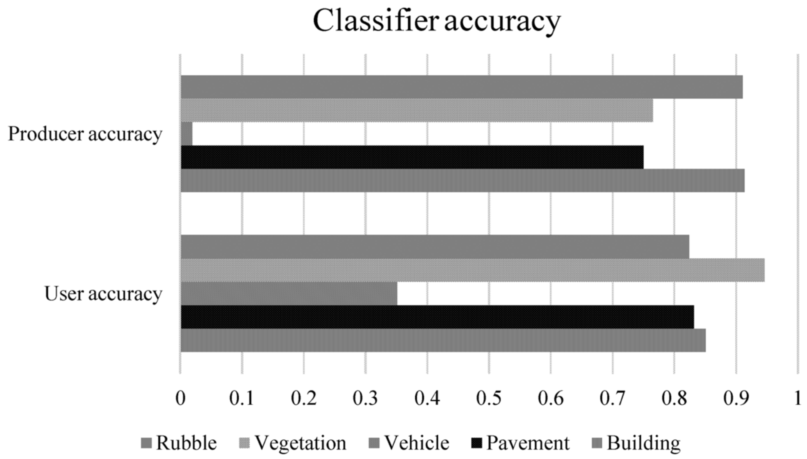

Fig. 11 Effect of using more specific subclasses of rubble.

following parameters. On decision trees, enable the 1SE rule for pruning. On random trees, we increase the minimum sample count to two. For SVMs, we use a radial basis function kernel with a *C* parameter of 3. For *k*-nearest neighbor, the *k* parameter is increased to 3. Naïve Bayes has no parameters that can be adjusted. Figures 12 and 13 show the accuracy of different algorithms.

Often, the choice is not clear as to which classifier is superior. Some may offer excellent producer performance while poor user accuracy performance or vice versa. We attempt to

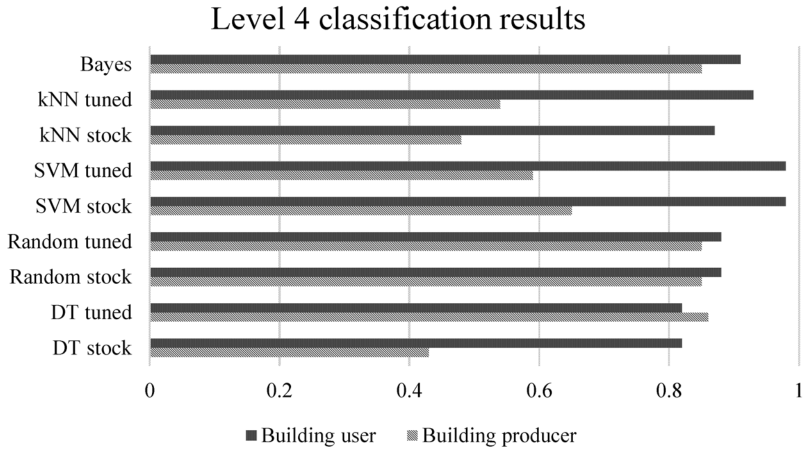

Fig. 12 Performance of various classifier algorithms on building classification.

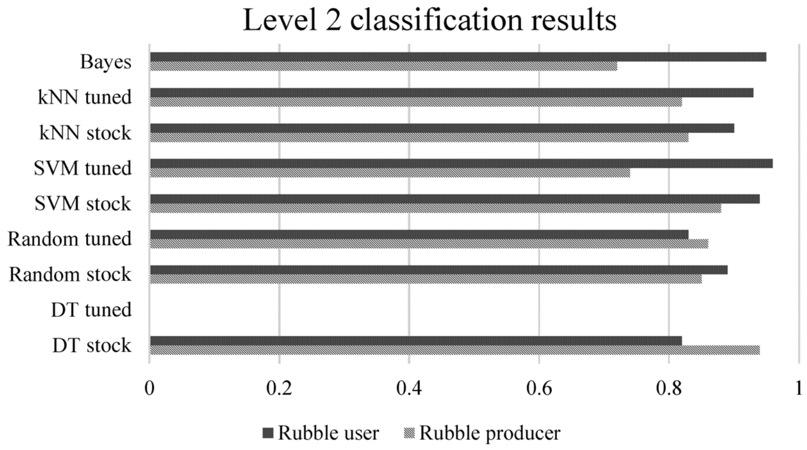

Fig. 13 Performance of various classifier algorithms on rubble classification.

balance these by selecting the classifier, which produces the best average user or producer performance by choosing Naïve Bayes for level four and an SVM for level 2. However, some applications may favor better performance in certain categories. For example, if we wanted to make sure we classified as much earthquake damage as possible without regard to false positives, we should consider the highest possible producer accuracy without regard to user accuracy.

5 Conclusions and Future Work

As is demonstrated above, our pixel-based and object-based classifications of this particular imagery of the 2011 Christchurch, New Zealand, earthquake perform better than established, successful classification methods. Our pixel-based classifier has rubble user and producer accuracies of 89% and 62% compared with that achieved by Ref. 5 of 63.4% and 71.3%. Our object-based classifier has rubble user and producer accuracies of 88% and 94% compared with that achieved by Ref. 6 of 91% and 60%.

Having established robust identification of earthquake damage that is equivalent to current methods, we can make valid comparisons between pixel-based and object-based methods. In both instances, we are comparing results from the same imagery, using the same training and validation data. We conclude that object-based methods can produce better results than pixel-based methods. Furthermore, object-based methods are capable of exceeding 85% overall accuracy in considering image pixels representing rubble versus those of other classes, a metric used by organizations, such as the United States Geological Survey for evaluating classifier performance.[14]

Another important conclusion is that object-based methods alone do not necessarily produce better results than pixel-based methods. As we demonstrate, a systematic approach is necessary to ensure proper classifier parameter choice. Poor choices in classifier design can impact results by 25% or more. Our systematic approach provides a more organized and directed method than subjective trial and error methods.

Future work on this subject is focused on the human decision key points shown in Fig. 3. Elimination of human interpretation of these results and the trial and error necessary in their creation can make for a faster, more robust classification system. For example, rather than trying numerous segmentation parameters and choosing the one considered best, a supervised or unsupervised method of image segmentation might be applied. This has the potential to both save time and eliminate human error.

Recent advances in deep learning methods also hold much promise for improving results. Popular deep learning methods have achieved strong results but unfortunately rely on a large dataset of similar imagery for training.[15] While publicly available datasets containing high resolution remotely sensed imagery of earthquake damage are currently limited, more and more data are being made available making these methods much more promising. While deep learning methods may not be able to outright replace current classification methods, they do hold much promise in improving aspects of our method, such as the feature selection process.[16,17] Furthermore, these methods may be applicable to individual classes with large amounts of available data, such as the building classification problem. As we have seen with the multilevel classification approach show in this paper, identifying nondamaged areas before considering damage alone greatly improves classifier accuracy.

Acknowledgments

Research described in this paper was carried out in part at the Jet Propulsion Laboratory, under contract with the National Aeronautics and Space Administration. Copyright 2015 California Institute of Technology. This material is based upon work supported by the National Science Foundation under Grant No. 1300720.

References

1. A. Coburn and R. Spence, *Earthquake Protection*, Wiley, West Sussex, England (1992).
2. P. Gamba and F. Casciati, “GIS and image understanding for near-real-time earthquake damage assessment,” *Photogramm. Eng. Remote Sens.* **64**, 987–994 (1998).

3. L. Dong and J. Shan, "A comprehensive review of earthquake-induced building damage detection with remote sensing techniques," *ISPRS J. Photogramm. Remote Sens.* **84**, 85–99 (2013).
4. Y. Gao and J. F. Mas, "A comparison of the performance of pixel-based and object-based classifications over images with various spatial resolutions," *Online J. Earth Sci.* **2**(1), 27–35 (2008).
5. G. Taskin Kaya, N. Musaoglu, and O. K. Ersoy, "Damage assessment of 2010 Haiti earthquake with post-earthquake satellite image by support vector selection and adaptation," *Photogramm. Eng. Remote Sens.* **77**(10), 1025–1035 (2011).
6. P. Li et al., "Urban building damage detection from very high resolution imagery using one-class SVM and spatial relations," in *2009 IEEE Int. Geoscience and Remote Sensing Symp. (IGARSS 2009)*, pp. V–112–V–114 (2009).
7. K. Kouchi and F. Yamazaki, "Damage detection based on object-based segmentation and classification from high-resolution satellite images for the 2003 Boumerdes, Algeria earthquake," in *Proc. of the 26th Asian Conf. on Remote Sensing*, Hanoi, Vietnam (2005).
8. S. W. Myint et al., "Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery," *Remote Sens. Environ.* **115**(5), 1145–1161 (2011).
9. A. Kaiser et al., "The Mw 6.2 Christchurch earthquake of February 2011: preliminary report," *N. Z. J. Geol. Geophys.* **55**(1), 67–90 (2012).
10. Land Information New Zealand, Christchurch earthquake imagery, http://www.linz.govt.nz/land/maps/linz-topographic-maps/imagery-orthophotos/christchurch-earthquake-imagery (13 April 2015).
11. M. Baatz and A. Schäpe, *Multiresolution Segmentation: An Optimization Approach for High Quality Multi-Scale Image Segmentation*, pp. 12–23, Herbert Wichmann Verlag, Berlin, Germany (2000).
12. Y. Zhang et al., Development of a supervised software tool for automated determination of optimal segmentation parameters for eCognition, NA (2010).
13. E. Alpaydin, *Introduction to Machine Learning*, MIT Press, Cambridge, Massachusetts (2010).
14. J. R. Anderson, A land use and land cover classification system for use with remote sensor data, US Government Printing Office (1976).
15. D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3642–3649 (2012).
16. W. Zhao et al., "On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery," *Int. J. Remote Sens.* **36**(13), 3368–3379 (2015).
17. Q. Zou et al., "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.* **12**(11), 2321–2325 (2015).

James Bialas received his MSc degree in integrated geospatial technology from Michigan Technological University in 2015. He is currently a PhD student in the computer science and engineering program at Michigan Technological University. His research interests include the application of machine learning to remotely sensed data.

Thomas Oommen received his PhD degree in civil and environmental engineering from Tufts University, Massachusetts, USA, in 2010. He is currently working as an associate professor in the Department of Geological and Mining Engineering and Sciences at Michigan Technological University. His main research activities are in the application of machine learning, image processing, and remote sensing for pre- and postgeohazard characterization.

Umaa Rebbapragada is a data scientist in the Machine Learning and Instrument Autonomy Group at the Jet Propulsion Laboratory, California Institute of Technology. Her work is focused on the infusion of start-of-the-art techniques from the academic machine learning community into large-scale data science systems, with a current focus on astronomical and earth observing imaging systems. She received her PhD in machine learning from Tufts University.

Eugene Levin received his MSc degree in astrogeodesy from Siberian State Geodetic Academy, Novosibirsk, Russia, in 1982 and PhD in photogrammetry from State Land Use Planning University, Moscow, Russia, in 1989. Currently, he is working as a program chair of surveying engineering and director of integrated geospatial technology graduate program at the Michigan Technological University. His research interests include geospatial image processing and human-computer symbiosis.