Information Hiding and Covert Communication

Andrew Ker
adk@comlab.ox.ac.uk

Royal Society University Research Fellow
Oxford University Computing Laboratory

Foundations of Security Analysis and Design
Bertinoro, Italy, 25-26 August 2008
Information Hiding and Covert Communication

- Steganography: στεγανος + γραφειν = “covered” + “to write”
- Digital Watermarking: (fingerprinting, tamperproofing)
- Digital Media Forensics
Information Hiding and Covert Communication

Part 1: Introduction to Steganography & Watermarking
Highlight: examples

Part 2: Steganalysis
Highlight: extremely sensitive detectors for covert communication

Part 3: More Efficient Steganography
Highlight: codes which approach the maximum possible efficiency

Part 4: Steganographic Capacity
Highlight: the Square Root Law
Part 1: Introduction to Steganography & Watermarking

- Steganography
  - Examples & countermeasures (steganalysis)
- Digital watermarking
  - Applications
  - Examples & countermeasures
- Typical embedding domains & operations
Communication

Alice

transmit

message

insecure channel

Bob

receive

Eve

message
Communication: cryptography

Alice → encrypt → insecure channel → decrypt → Bob

message → Eve → "cyphertext" → secret key

encrypt

decrypt
Communication

Alice

message

Bob

transmit

noisy channel

damaged

message

receive
Communication: coding

Alice

message

encoding

noisy channel

damaged message

code

Bob
Communication

1. Cryptography
   *Keep the message confidential*

2. Coding
   *Keep the message intact*

Other challenges:

*How can Alice and Bob share the secret key/code?*

*How does Alice know that she is communicating with Bob?*
Steganography

Alice

message

“cover object”

embedding

insecure channel

“stego object”

Bob

Steganography

Alice

message

embedding

“cover object”

Eve

or

?

Bob

extraction

“stego object”

insecure channel

secret key
Steganography

Why digital media?

- *We will see that it has plenty of capacity for hidden data.*
- *Over 70% of all internet traffic is audio, picture, or video.*
- *Over 20% of all internet traffic is YouTube video!*
- *Over 90% of peer-to-peer traffic is audio, picture, or video.*

[Ipoque / Ellacoya Networks, 2007]
Example
Cover object, 512×512 pixels, grayscale
Cover object, 512×512 pixels, grayscale
LSB replacement

The simplest steganographic method for uncompressed ("TIF", "BMP") images.

Embedding

- Form payload as a sequence of bits,
- Take cover as a sequence of bytes,
- Replace Least Significant Bits (LSB) of cover bytes with payload.

Extraction

- Take cover as a sequence of bytes,
- Read LSBS.

Can be performed by an 80-character Perl program on a Unix commandline:

```
perl -n0777e '$_=unpack"b*",$_;split/(\s+)/,<STDIN>,5; 
    @_[8]=~s{.}{$&&v254|chop()&v1}ge;print@_' <input.pgm >output.pgm stegotext
```
LSB replacement

The simplest steganographic method for uncompressed ("TIF", "BMP") images.

**Embedding**
- Form compressed & encrypted payload as a sequence of bits,
- Take cover as a sequence of bytes in pseudorandom order per secret key,
- Replace Least Significant Bits (LSB) of cover bytes with payload.

**Extraction**
- Take cover as a sequence of bytes in pseudorandom order per secret key,
- Read LSBs, decrypt and decompress.

Using pseudorandom order also has the advantage of spreading smaller-than-maximal payloads throughout the cover.
Stego object, 1 secret bit per cover pixel (32KB)
Cover object, 512×512 pixels, grayscale
Stego object, 1 secret bit per cover pixel (32KB)
Cover object, 512×512 pixels, grayscale
Cover object, 512x512 pixels, 24 bits per pixel
Stego object, 9 secret bits per cover pixel (288KB)
Steganography

Alice

message

embedding

“cover object”

insecure channel

“stego object”

Bob

Eve

or

?

secret key

Alice

Eve

Bob
Steganography is not...

- Hiding data in pseudorandom streams e.g. Truecrypt.  
  *(too easy)*

- Hiding data in unused parts of image/video/audio/packet/program headers.  
  *(not secure)*
Steganalysis

- the counter-discipline to steganography, detecting hidden data.

Even when steganography is not perceptible (visually, audibly, ...) it might still be detected by statistical analysis.
The “pairs of values” effect can be used to make a detector, known as the “Chi-Square” detector for LSB replacement.

Digital watermarking

Alice

message

embedding

“cover object”

noisy channel

Bob

damaged stego object

secret key

extraction

noisy channel
Applications

- Broadcast monitoring
  - The watermark is a machine-readable tag identifying the cover content.

- Copyright enforcement
  - The watermark proves ownership of the cover.
  - The watermark indicates a license to play the cover medium on specific device.

- Traitor tracing
  - The watermark identifies the original recipient of the cover.
Traitor tracing

multimedia distributor

A

Andrew

B

Brian

C

Charlie

file sharing network
Example “DSSS Watermarking”

Form cover as a vector of bytes:

\[ d = (d_1, \ldots, d_n) \]

Generate random “watermark” sequence (actually a secret key) of +1 and -1:

\[ w = (w_1, \ldots, w_n) \quad w_i = \pm 1 \quad \text{equiprobably, and independently} \]

Watermarked image is simply \( d^* = d + \alpha w \) (\( \alpha \) is the watermark strength).

To detect the presence of the watermark, form the “similarity” or “normalized correlation” score

\[ nc(d^*, w) = \frac{d^* \cdot w}{\sqrt{d^* \cdot d^*}}. \]

which is high when this particular watermark is present and low otherwise.

---

Example “DSSS Watermarking”

Form cover as a vector of bytes:

$$d = (d_1, \ldots, d_n)$$

Generate random “watermark” sequence (actually a secret key) of +1 and -1:

$$w = (w_1, \ldots, w_n) \quad w_i = \pm 1 \quad \text{equiprobably, and independently}$$

Watermarked image is simply $$d^* = d + \alpha w$$ ($$\alpha$$ is the watermark strength).

To detect the presence of the watermark, form the “similarity” or “normalized correlation” score

$$nc(d^*, w) = \frac{d^* \cdot w}{\sqrt{d^* \cdot d^*}}.$$ 

which is high when this particular watermark is present and low otherwise.

This watermark conveys no information except its presence (“zero bit watermark”).
Watermark detection

Watermarked image

![Graph showing similarity of 50 random watermarks]
Watermark attacks

- *the counter-discipline to watermarking, destroying hidden data.*

(The opponent wants to remove the watermark without destroying the cover.)

Note that embedding a watermark MUST degrade the cover a bit, otherwise it can be painlessly overwritten.
Common attacks

- Noise attack
  \textit{Not targeted: simply aim to reduce watermark fidelity.}

- Collusion attack
  \textit{Take the average of many copies of the same cover, with different watermarks.}

- Desynchronization attack
  \textit{Distort the cover spatially so that the watermark is no longer detectable.}

- Watermark estimation
  \textit{EITHER use many objects with the same watermark, OR treat the detector as an oracle, to estimate (and then subtract) the watermark.}
Noise attack

*Added Gaussian noise (standard deviation=20).*

**Watermarked+attacked image**
Collusion attack

*Formed a new image by averaging three copies of the same cover marked with different watermarks.*
Desynchronization attack

Removed pixel column 1 and pixel duplicated column 512.

Watermarked+attacked image
Steganography & watermarking

- Steganography
  *Embed message so that it cannot be detected*
  Steganalysis
  *Detect hidden information*

- Digital watermarking
  *Embed message so that it cannot be removed*
  Watermark attacks
  *Remove information (without destroying cover)*
JPEG

- *lossy compression using quantization in the DCT domain.*

1. Split into $8 \times 8$ blocks.
2. Discrete Cosine Transform.
3. Quantize DCT coefficients.
4. Compress conventionally and store.
1. Split into 8×8 blocks.
2. Discrete Cosine Transform (to 8×8 coefficients).
3. Divide coefficients by “quantization table” and round to nearest.
4. Compress quantized coefficients conventionally and store.
JPEG

In practice, JPEG compression reduces file size with relatively little loss in visual quality.

Bitmap 768KB

JPEG 66KB
JPEG steganography

It is at this level that the steganographic payload is (usually) embedded.

e.g. "F5" steganography:
uses the LSBs of the nonzero quantized coefficients.
F5 steganography

Alters LSBs of nonzero quantized coefficients (there are some slight difficulties with avoiding zeros).

JPEG cover 66KB

Stego object with 8.1KB payload
Can hide information in...

- uncompressed images,
- compressed images,
- audio,
- movies,
- 3D meshes,
- screen savers,
- fonts,
- source code,
- byte code,
- text,
- DNA (!),
- ...
Part 2: Steganalysis

- Aims & general overview
  - “Chi-Square” detector for LSB replacement in uncompressed images
- Detection using combinatorial analysis of embedding
  - “Couples” detector for LSB replacement in uncompressed images
- Detection using machine learning
  - “Extended DCT” detector for F5 embedding in JPEG images
Steganalysis

Aim: to detect whether an object contains a covert payload or not.

Steganalysis can be...
- **targeted** at a particular embedding method *(most common)*, or
- **blind**, with potential to unmask even unknown embedding methods *(rare, usually weak)*.

The output can be...
- **simple binary**: yes or no to the presence of payload, or
- **quantitative**, estimating the size of the payload.

Most steganalysers use one of two methodologies:
1. Combinatorial analysis of embedding operation *(must be targeted)*.
2. An application of machine learning techniques.
Chi-Square detector
Chi-Square detector

Measure closeness of pairs by the “Chi-Square” statistic:

\[ X^2 = \sum_{i=0}^{127} \frac{(f[2i] - f[2i + 1])^2}{f[2i] + f[2i + 1]} \]

(Have to exclude terms with zero or very small denominator.)

High value of \( X^2 \) → no payload

Low value of \( X^2 \) → suspect payload

Steganalysis

*Aim: to detect whether an object contains a covert payload or not.*

Steganalysis can be...

- **targeted** at a particular embedding method (*most common*), or
- **blind**, with potential to unmask even unknown embedding methods (*rare, usually weak*).

The output can be...

- **simple binary**: yes or no to the presence of payload, or
- **quantitative**, estimating the size of the payload.

Most steganalysers use one of two methodologies:
1. **Combinatorial analysis of embedding operation** (*must be targeted*).
2. An application of machine learning techniques.
Measuring performance

Binary detectors are benchmarked by their false positive / false negative tradeoff “Receiver Operating Characteristic” curve.
Image library

In the absence of a perfect model for covers, we must estimate the ROC empirically.

Here we will use a library of 1600 cover images, each 3 Mpixels, taken in RAW format using a digital camera.

NB: performance might be highly dependent on the characteristics of the covers. Good researchers test on multiple, independent, sets of covers.

Performance of Chi-Square
Chi-Square is only a weak detector for LSB replacement steganography, if
• the payload size is smaller than maximum, and
• the payload is spread pseudorandomly through the cover.
Couples detector

“Couples” is a more recent detector for LSB replacement in uncompressed images. It differs from Chi-Square in that:

1. It has a specific model for certain statistical properties of cover images.
2. It is quantitative (estimates the size of payload).

The detector uses properties of adjacent pairs of pixels, to estimate the proportionate payload size.

---


Steganalysis

Aim: to detect whether an object contains a covert payload or not.

Steganalysis can be...

- **targeted** at a particular embedding method (most common), or
- **blind**, with potential to unmask even unknown embedding methods (rare, usually weak).

The output can be...

- **simple binary**: yes or no to the presence of payload, or
- **quantitative**, estimating the size of the payload.

Most steganalysers use one of two methodologies:

1. **Combinatorial analysis of embedding operation** (must be targeted).
2. An application of machine learning techniques.
Couples detector

We look at *adjacent pairs of pixels*, and the effects of LSB operations on them.

Definitions (classification of pixel pairs)

\[ \mathcal{P} \quad \text{all adjacent pixel value pairs } (x, y). \]

\[ \mathcal{C}_m \quad \text{pairs with values } (x, y) \text{ such that } \lfloor x/2 \rfloor - \lfloor y/2 \rfloor = m. \]

\[ \mathcal{E}_m \quad \text{pairs with values } (2k, 2k + m). \]

\[ \mathcal{O}_m \quad \text{pairs of the form } (2k + 1, 2k + 1 + m). \]

e.g. if 66 and 72 are the values of two adjacent pixels then this pair is in \( \mathcal{P}, \mathcal{C}_3, \text{ and } \mathcal{E}_6. \)
Trace sets

\[ \mathcal{P} \] all adjacent pixel value pairs \((x, y)\).

\[ \mathcal{C}_m \] pairs with values \((x, y)\) such that \(\lfloor x/2 \rfloor - \lfloor y/2 \rfloor = m\).

\[ \mathcal{E}_m \] pairs with values \((2k, 2k + m)\).

\[ \mathcal{O}_m \] pairs of the form \((2k + 1, 2k + 1 + m)\).
Trace sets

**Structural Property:**

*LSB replacement moves pairs between trace subsets, but the trace sets are fixed.*
Embedding transitions

Fix $m$. How are the trace subsets of $C_m$ affected by LSB operations?
Embedding transitions

Example: some pairs for $m = 3$
Embedding transitions

When proportion $p$ LSBs are flipped (at random).
Embedding transitions

Fix a cover of size $N$. Embed a random message of length $2pN$.

Define

\[ e_m = \text{#pairs in } \mathcal{E}_m \text{ in cover} \]
\[ o_m = \text{#pairs in } \mathcal{O}_m \text{ in cover} \]
\[ e'_m = \text{#pairs in } \mathcal{E}_m \text{ after embedding} \]
\[ o'_m = \text{#pairs in } \mathcal{O}_m \text{ after embedding} \]

Then

\[ e'_{2m} \approx (1 - p)^2 e_{2m} + p(1 - p) o_{2m - 1} + p(1 - p) e_{2m + 1} + p^2 o_{2m}. \]

(this is really the expectation of a random variable)
Embedding transitions

Fix a cover of size $N$. Embed a random message of length $2pN$.

Define

$e_m = \text{#pairs in } \mathcal{E}_m \text{ in cover}$

$o_m = \text{#pairs in } \mathcal{O}_m \text{ in cover}$

$e'_m = \text{#pairs in } \mathcal{E}_m \text{ after embedding}$

$o'_m = \text{#pairs in } \mathcal{O}_m \text{ after embedding}$

Then

$$e'_{2m} \approx (1 - p)^2 e_{2m} + p(1 - p) o_{2m-1} + p(1 - p) e_{2m+1} + p^2 o_{2m}.$$
Embedding transitions

Fix a cover of size $N$. Embed a random message of length $2pN$.

Define

$e_m = \#\text{pairs in } \mathcal{E}_m \text{ in cover}$

$o_m = \#\text{pairs in } \mathcal{O}_m \text{ in cover}$

$e'_m = \#\text{pairs in } \mathcal{E}_m \text{ after embedding}$

$o'_m = \#\text{pairs in } \mathcal{O}_m \text{ after embedding}$

Then

$$e'_m \approx (1 - p)^2 e_m + p(1 - p) o_{2m - 1} + p(1 - p) e_{2m + 1} + p^2 o_{2m}.$$
Embedding transitions

Fix a cover of size $N$. Embed a random message of length $2pN$.

Define

$e_m = \text{#pairs in } \mathcal{E}_m \text{ in cover}$

$o_m = \text{#pairs in } \mathcal{O}_m \text{ in cover}$

$e_m' = \text{#pairs in } \mathcal{E}_m \text{ after embedding}$

$o_m' = \text{#pairs in } \mathcal{O}_m \text{ after embedding}$

Then

$$e_m' \approx (1 - p)^2 e_m + p(1 - p) o_{2m - 1} + p(1 - p) e_{2m + 1} + p^2 o_{2m}.$$
Inverting the transitions

We derive:

\[
\begin{pmatrix}
\epsilon'_{2m} \\
\delta'_{2m-1} \\
\epsilon'_{2m+1} \\
\delta'_{2m}
\end{pmatrix} \approx
\begin{pmatrix}
(1-p)^2 & p(1-p) & p(1-p) & p^2 \\
p(1-p) & (1-p)^2 & p^2 & p(1-p) \\
p(1-p) & p^2 & (1-p)^2 & p(1-p) \\
p^2 & p(1-p) & p(1-p) & (1-p)^2
\end{pmatrix}
\begin{pmatrix}
\epsilon_{2m} \\
\delta_{2m-1} \\
\epsilon_{2m+1} \\
\delta_{2m}
\end{pmatrix}
\]

Inverting,

\[
\begin{pmatrix}
\epsilon_{2m} \\
\delta_{2m-1} \\
\epsilon_{2m+1} \\
\delta_{2m}
\end{pmatrix} \approx \frac{1}{(1 - 2p)^2}
\begin{pmatrix}
(1-p)^2 & -p(1-p) & -p(1-p) & p^2 \\
-p(1-p) & (1-p)^2 & p^2 & -p(1-p) \\
-p(1-p) & p^2 & (1-p)^2 & -p(1-p) \\
p^2 & -p(1-p) & -p(1-p) & (1-p)^2
\end{pmatrix}
\begin{pmatrix}
\epsilon'_{2m} \\
\delta'_{2m-1} \\
\epsilon'_{2m+1} \\
\delta'_{2m}
\end{pmatrix}
\]
Inverting the transitions

We derive:

\[
\begin{pmatrix}
  e_{2m}' \\
  o_{2m-1}' \\
  e_{2m+1}' \\
  o_{2m}'
\end{pmatrix}
\approx
\begin{pmatrix}
  (1-p)^2 & p(1-p) & p(1-p) & p^2 \\
  p(1-p) & (1-p)^2 & p^2 & p(1-p) \\
  p(1-p) & p^2 & (1-p)^2 & p(1-p) \\
  p^2 & p(1-p) & p(1-p) & (1-p)^2
\end{pmatrix}
\begin{pmatrix}
  e_{2m} \\
  o_{2m-1} \\
  e_{2m+1} \\
  o_{2m}
\end{pmatrix}
\]

Inverting,

\[
\begin{align*}
  e_m &\approx \phi_m(p, e', o') \\
  o_m &\approx \psi_m(p, e', o')
\end{align*}
\]
Cover model

In natural images, we believe that $e_m \approx o_m$.

(Why? The difference between the values of each pair should be independent of the parity.)
Cover model

In natural images, we believe that $e_m \approx o_m$. 
Cover model

In natural images, we believe that \( e_m \approx o_m \).
Creating the estimator

For each $m$, we have

\[ e_m \approx o_m \]
\[ e_m \approx \phi_m(p, e', o') \]
\[ o_m \approx \psi_m(p, e', o') \]

and

\[ 0 \approx e_m - o_m \approx \phi_m(p, e', o') - \psi_m(p, e', o') \]

The Couples estimator for $p$ is the (lower) root of the equation

\[ 0 = \sum_m \phi_m(p, e', o') - \psi_m(p, e', o') \]
Estimator performance

Estimates from 150 images: some had zero LSB payload, some 0.5bpp, some 1bpp.
Detector performance

![Graph showing detector performance with probability of correct classification vs. payload size (bpp)].
Other detectors for LSB embedding

“Histogram Characteristic Function”  Harmsen, 2002; Ker, 2005;...
“Higher Order Statistics”          Lyu & Farid, 2002
“Chi-Square”                   Westfeld & Pfitzmann, 1999
“Raw Quick Pairs”            Fridrich et al., 2000
“RS”                               Fridrich et al., 2001
“Difference Histogram”       Zhang & Ping, 2003
“Pairs” (for palette images) Fridrich et al., 2003
“Triples”                      Ker, 2005
“Couples/ML”                  Ker, 2007
“2Couples” (for embedding in 2 LSBs) Ker, 2007
“WS”                                Fridrich & Goljan, 2004; Ker & Böhme, 2008; Böhme, 2008
Detector performance

[Graph showing the probability of correct classification vs. payload size (bpp) for different methods: Chi-Square, Couples, WS]
F5 steganography

... uses the LSBs of the nonzero quantized coefficients.

\[ d^{(i,j,k)}[I] = \text{Quantized coefficient at mode} (i, j) \text{ in } 8 \times 8 \text{ block } k \]

F5 detector

- a simplification of the “Extended DCT Feature” classifier due to Pevný & Fridrich.

Rather than examine the F5 embedding operation in detail, this detector uses machine learning (supervised learning for classification) techniques.

Steganalysis

Aim: to detect whether an object contains a covert payload or not.

Steganalysis can be...

- **targeted** at a particular embedding method (**most common**), or
- **blind**, with potential to unmask even unknown embedding methods (**rare, usually weak**).

The output can be...

- **simple binary**: yes or no to the presence of payload, or
- **quantitative**, estimating the size of the payload.

Most steganalysers use one of two methodologies:

1. Combinatorial analysis of embedding operation (**must be targeted**).
2. An application of **machine learning techniques**.
Supervised learning for classification

Suppose a universe of objects which fall into discrete, disjoint, classes.

Key elements:

- Select feature vector
  
  *Each object is projected onto a vector of (hopefully) relevant features*

- Training phase

  *Separate feature space into class regions based on known objects*

- Application

  *Predict class of new objects, based on their features*
Example
Example
Ideal

Stego object features

Cover object features
DCT features

For features we use the histogram of coefficients (for each DCT mode separately):

\[
h_{(i,j)}^n[I] = \frac{\#\{d_{(i,j,k)}[I] = n \mid 1 \leq k \leq B\}}{B}
\]

And also the “dual histogram”:

\[
g_{(i,j)}^n[I] = \frac{\#\{d_{(i,j,k)}[I] = n \mid 1 \leq k \leq B, 1 \leq i', j' \leq 8\}}{\#\{d_{(i,j,k)}[I] = n \mid 1 \leq k \leq B\}}
\]

To keep the dimensionality down, we consider only

\[(i, j) \in \{(2, 1), (1, 2), (2, 2), (3, 1), (1, 3)\}\]

\[-5 \leq n \leq 5\]

for a total of 110 features.
Classification engines

Popular methods to determine the class regions from the training data include:

- Fisher Linear Discriminator
- Multi-layer Perceptron (a.k.a. Neural Network)
- Support Vector Machine
- k-Nearest Neighbours

In most cases the classes are separated by hyperplanes, but the “kernel trick” allows certain types of nonlinear classification at little extra cost.
Support Vector Machine

A SVM finds a separating hyperplane with maximum margin between the classes.

– *When no such hyperplane exists “soft margin” SVMs can be used.*
– *The “kernel trick” allows nonlinear boundaries.*
Performance

To make a steganography detector,

1. Take a set of cover images, and create a set of stego images.
2. Compute the 110 features for every image.
3. Train a SVM on this data (also optimizing the learning parameters).

*Test the trained SVM on fresh images. Result... hopeless performance.*
Calibration

We need a rough estimate for feature values of the cover, given the stego object.

Decompress stego object

Crop 4 rows & columns, recompress with same JPEG parameters

“Calibration image”
Calibration

We need a rough estimate for feature values of the cover, given the stego object.

Decompress stego object

Crop 4 rows & columns, recompress with same JPEG parameters

“Calibration image”

I

\( \tilde{I} \)
Calibrated features

We use the calibrated histogram:

$$h_{(i,j)}^n[\mathbf{I}] - h_{(i,j)}^n[\mathbf{\tilde{I}}]$$

And also the calibrated dual histogram:

$$g_{(i,j)}^n[\mathbf{I}] - g_{(i,j)}^n[\mathbf{\tilde{I}}]$$

For

$$(i, j) \in \{(2, 1), (1, 2), (2, 2), (3, 1), (1, 3)\}$$

$$-5 \leq n \leq 5$$
Performance

To make a steganography detector,

1. Take a set of cover images, and create a set of stego images.
2. Compute the 110 calibrated features for every image.
3. Train a SVM on this data (also optimizing the learning parameters).

Test the trained SVM on fresh images. Result...
Performance

![Graph showing performance improvement with payload size for different DCT features and SVM methods. The graph plots the probability of correct classification against payload size (bits per nonzero coefficient). Two lines are shown: one for 110 DCT features / SVM and another for 274 DCT features / SVM. The graph indicates a clear improvement in classification accuracy as payload size increases.]
Other detectors for F5

(most work for other JPEG embedders too)

Category attack

“Binary Similarity Measures”
“Higher Order Statistics”
“KFD”
23 “DCT features”
“Markov features”

“Merged features” (Markov + DCT)

Lee & Westfeld, 2006 & 2007

Avcibas et al., 2001
Lyu & Farid, 2002
Harmsen & Pearlman, 2004
Fridrich, 2004
Shi et al., 2005
Pevný & Fridrich, 2007
Steganalysis

Most steganalysers use one of two methodologies:

1. Combinatorial analysis of embedding operation.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>often highly sensitive</td>
<td>difficult to find cover properties</td>
</tr>
<tr>
<td>usually of low computational</td>
<td>can be complex to derive a detector</td>
</tr>
<tr>
<td>complexity</td>
<td></td>
</tr>
<tr>
<td>applicable to many cover types</td>
<td></td>
</tr>
</tbody>
</table>

2. An application of machine learning techniques.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>embedding need not be fully</td>
<td>easy to include too many useless</td>
</tr>
<tr>
<td>understood</td>
<td>features</td>
</tr>
<tr>
<td>can utilize standard techniques</td>
<td>often computationally expensive</td>
</tr>
<tr>
<td>easy to add new features</td>
<td>different cover types need separate</td>
</tr>
<tr>
<td></td>
<td>training</td>
</tr>
</tbody>
</table>