Cloud Recognition in the Miami Skyline

Senior Design Project, ECS193

Eric Gribkoff
University of California, Davis
eagribkoff@ucdavis.edu

Saerorm Park
University of California, Davis
saepark@ucdavis.edu

Liz Shigetoshi
University of California, Davis
eashigetoshi@ucdavis.edu

Ashish Subedy
University of California, Davis
asubedy@ucdavis.edu

1. INTRODUCTION

The goal of our project was to build a system capable of automatically classifying cloud types present in images taken by a camera observing the Miami skyline. The ability to automatically classify the clouds present in a large set of images, without human involvement, enables easily compiling statistics on the weather conditions of a region over large periods of time. Climate scientists can mine this data for statistics useful in characterizing weather patterns and modeling climate.

Previous cloud classification systems extract features from the color distribution and textures present in images containing clouds. Our method differs from existing systems by incorporating the following techniques:

- Cloud pixel identification based on Minimum Cross-Entropy (MCE)
- Color-Invariant edge detection combined with Histogram of Oriented Gradients (HOG)

These additions improve classifier performance. This is demonstrated by tests reporting the accuracy of classifiers with and without the above enhancements.

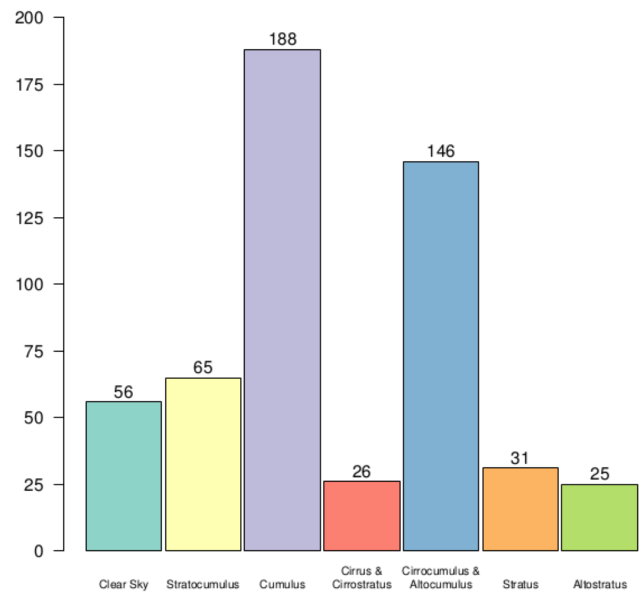


Figure 1: Category Distribution

2. DATA

We received a large number of images taken by a netcam overlooking the Miami skyline. For our system, we considered images that contained only one category of cloud. The seven classified cloud categories were as follows:

1. Cirrocumulus and Altocumulus
2. Stratocumulus
3. Cumulus
4. Stratus
5. Clear Sky
6. Cirrus and Cirrostratus
7. Altostratus

Figure 1 describes the cloud category distribution of our 537 images.

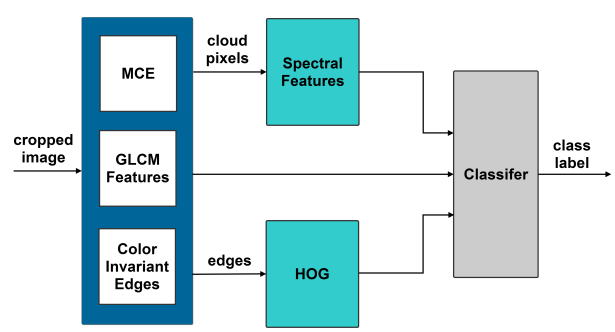
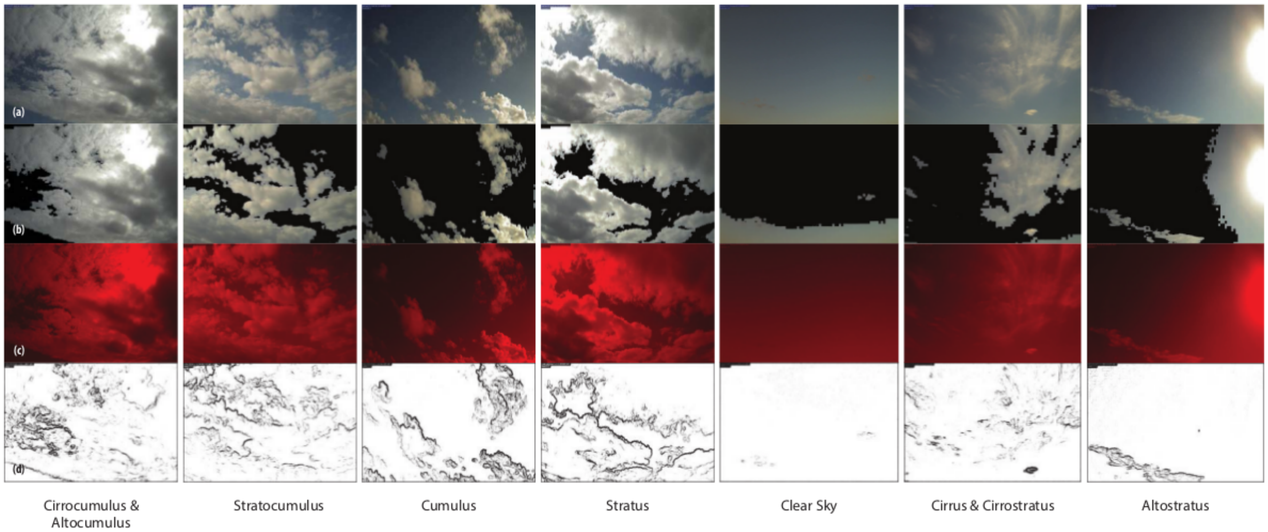


Figure 2: System Overview

3. FEATURE EXTRACTION

The feature extraction is performed in Matlab. A file containing the image file names and corresponding class labels is passed to our Matlab code. The features extracted fall



(a) Original images; (b) Segmented images; (c) Color invariant images; (d) Edge detected images.

Figure 3: Feature Extraction

into the following categories:

- Spectral Features
- Textural Features
- Edges
- Cloud Cover

3.1 Spectral Features

3.1.1 Image Segmentation

Prior to computing the spectral features for an image, an attempt is made to segment the image into cloud and non-cloud pixels. Two techniques were independently used for this: fixed thresholding and minimum cross-entropy (MCE).

Fixed thresholding classifies a pixel as cloud if the color intensity of the red channel divided by the color intensity of the blue channel is greater than 0.9. This number is highly dependent upon the particular camera in use and the lighting intensity in the image, as discussed in [3] and [4]. The value 0.9 was chosen based on a simple examination of the results using a variety of threshold values; based on a subjective evaluation of the results, 0.9 was found to offer good performance for our image set.

The other technique, MCE, is discussed in detail in [4]. MCE does not rely on a fixed threshold value. Instead, a value $\lambda = (b-r)/(b+r)$, where b is the blue channel intensity and r is the red channel intensity, is computed for each pixel. The minimum cross-entropy is then computed over the entire image, using λ. The returned threshold is used to decide if a pixel is a cloud.

In [4], a difference in the standard deviation of λ across an entire image is used to determine whether fixed thresholding or MCE would better function to segment a given image. No such distinction in the standard deviations of λ was found to hold for image set. As such, we ran trials of our feature extraction and classification using first fixed thresholding, then MCE, independently. Results for each approach are presented below.

3.1.2 Use of Segmented Image

After the image has been segmented by either of the above techniques, the following seven spectral features are computed over the cloud pixels only:

1. Mean of red channel
2. Mean of blue channel
3. Standard deviation of blue channel
4. Skewness of blue channel
5. Difference between red and green channels
6. Difference between red and blue channels
7. Difference between blue and green channels

No spectral features are computed for pixels identified as non-cloud.

3.2 Textural Features

Textural features are represented by the following measurements derived from the grey level co-occurrence matrix (GLCM) of the original image's blue channel:

1. Energy
2. Entropy
3. Contrast
4. Homogeneity

These four GLCM features are computed over all pixels of the entire image, irrespective of the results of the image segmentation process.

3.3 Edges

Edge information is extracted by making use of the color-invariant edge detection method presented in [1]. We made use of the W color invariant set, which is invariant to the image's illumination intensity, to identify the edges present.

This color invariant set was found, by a subjective analysis of the results, to correctly identify the cloud edges regardless of the sun's presence in an image.

The edge-detected image is then converted to a Histogram of Oriented Gradients (HOG), which returns an 81-element vector representing the distribution of edges over the image. Like the textural features, the HOG is calculated over the entire cropped image, ignoring the results of the image segmentation step.

3.4 Cloud Cover

The final feature, cloud cover, is the percentage of pixels in the cropped image identified as cloud pixels during the image segmentation step.

4. CLASSIFICATION RESULTS

4.1 Weka

Weka was used to perform all classification.

4.2 Results

Four classifiers were found to offer performance near the 90% accuracy mark, based off a set of trials performed within the Weka GUI. Ensemble of Nested Dichotomies used the default Weka configuration. KNN is evaluated using the IB-K algorithm, with $K = 1$. The multilayer perceptron was trained with the default Weka options, except the number of hidden layers was set to 7. The random forests are generated using 10 random trees, each based off 7 randomly chosen attributes.

Accuracy measurements reported by Weka as a result of 10-fold cross validation are presented in Figure 4. The classifiers tested were:

1. Ensemble of Nested Dichotomies
2. KNN
3. Multilayer Perceptron
4. Random Forest

Results show accuracy using fixed thresholding or MCE for image segmentation and including or excluding the HOG features.

KNN was used in the cloud classification systems presented in [2] and [3]. It was also found to offer the greatest accuracy in our own evaluations. KNN required less than a second to compute the model and perform the 10-fold cross validation. The detailed results generated by Weka for the KNN classifier, using MCE for image segmentation and incorporating the HOG features, are presented in Figures 5 and 6.

5. ACKNOWLEDGEMENTS

Special thanks to Prabhat, Rusen Oktem, and David Romps at the Lawrence Berkeley National Laboratory, and Ken Joy at University of California, Davis.

6. REFERENCES

[1] J.-M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(12):1338–1350, 2001.

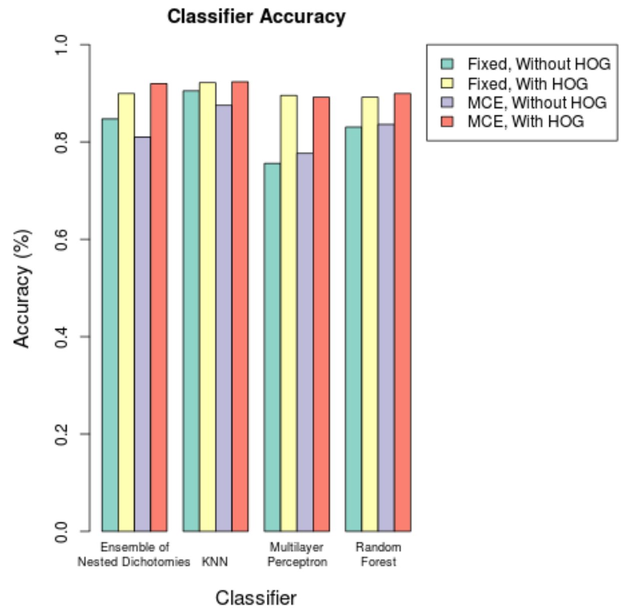


Figure 4: Classifier Results

KNN (K-Nearest Neighbors)	
Correctly Classified Instances: 92.365% (496/537)	
Incorrectly Classified Instances: 7.635% (41/537)	
Time Taken to Build Model: 0 seconds	
Total number of Instances: 537 images	

Class	TP Rate	FP Rate	Precision	F-Measure	ROC Area
Clear Sky	0.911	0.012	0.895	0.903	0.952
Stratocumulus	0.846	0.028	0.809	0.827	0.900
Cumulus	0.931	0.032	0.941	0.936	0.943
Cirrus & Cirrostratus	0.932	0.013	0.965	0.948	0.950
Cirrocumulus & Altocumulus	0.960	0.002	0.960	0.960	0.985
Stratus	1	0.004	0.939	0.969	0.999
Altostratus	0.923	0.006	0.889	0.906	0.970

Figure 5: KNN Results

Confusion Matrix

a	b	c	d	e	f	g
51	2	3	0	0	0	0
0	55	6	3	0	0	1
5	5	175	2	0	1	0
0	6	2	136	0	1	1
0	0	0	0	24	0	1
0	0	0	0	0	31	0
1	0	0	0	1	0	24

Classified as

a = Clear Sky
b = Stratocumulus
c = Cumulus
d = Cirrocumulus & Altocumulus
e = Altostratus
f = Stratus
g = Cirrus & Cirrostratus

Figure 6: KNN Confusion Matrix

[2] A. Heinle, A. Macke, and A. Srivastav. Automatic cloud classification of whole sky images. *Atmos. Meas. Tech*, 3(3):557–567, 2010.

[3] A. Kazantzidis, P. Tzoumanikas, A. Bais, S. Fotopoulos, and G. Economou. Cloud detection and classification with the use of whole-sky ground-based images. *Atmospheric Research*, 2012.

[4] Q. Li, W. Lu, and J. Yang. A hybrid thresholding algorithm for cloud detection on ground-based color images. *Journal of Atmospheric and Oceanic Technology*, 28(10):1286–1296, 2011.

[5] O. Ludwig, D. Delgado, V. Gonçalves, and U. Nunes. Trainable classifier-fusion schemes: an application to pedestrian detection. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, pages 1–6. IEEE, 2009.