Automated Recognition of Butterfly Species from an Image

Final Project Report

Mengu Sukan
CS 6737 Biometrics • Columbia University • December 11, 2008

ABSTRACT

My project is about identifying butterflies from an image. To that end, I made use of several techniques from computer vision and pattern classification that we discussed in class during the semester. My initial database contains a considerable amount of data: 103 species and 450+ images collected from a single website. The most challenging part of my project was finding the right features and the right segmentation methodology. My initial attempts on both fronts (segmentation in the HSV space and classification based on the RGB distribution of the image) failed to achieve decent classification results. At that point, I reverted to using pixel-wise comparison and linear dimension reduction concepts that we learned in the context of face recognition to my case. With the right mask, I was able to obtain a perfect classification rate using a simple Nearest-Neighbor classifier. Finally, I've included a few thoughts on possible next steps for further experimentation/research at the end of this report.

OBJECTIVE

The idea of working on a butterfly classification methodology came to as I was looking at a piece of art hanging on my living room wall. The project for identifying plant species was on fresh in my mind and I thought: If there is interest in a system to assist botanists/museum curators identify different species of plants, perhaps a similar system that classifies other organisms could be of use as well. Additionally, even though I am not familiar with biological classification of species per se, I thought that working with butterflies that come in many color patterns and shapes could be both challenging and interesting.

One of my first to-do's after coming up with an idea for my final project was to investigate the level of difficulty associated with the butterfly classification problem and how it is being addressed today. Various sources put the number of estimated butterfly species in the world today somewhere between 13,000 to 25,000 distinct species. Although this is a small number compared to plants (which have their species-count in 100,000s), I felt that it still is a sizable problem that might warrant an automated system for identification. Additionally, if one were to succeed in building a reliable system to identify butterflies, I thought that it might be possible for that same system to attempt classification of moths, which butterflies are a subset of, and ultimately other insects, which would obviously increase the applicable domain of such a system by orders of magnitude.

Based on the limited amount of time I had to design and build this system, I decided to put a few important constraints on the problem. One of the topics that was discussed extensively in class in the context of face recognition was the importance of lighting and pose in recognition tasks. We have seen time and time again that variability in those 2 factors makes the job of an automated recognition system much harder then the simple case with known pose and lighting. Given that I had start from scratch with no precedent in the case of butterflies, I decided to limit my system to a specific database of images, which practically guarantee pose alignment and show little variability in lighting. With these constraints in place, for the rest of this report, I will focus on the fundamental issues of pattern recognition, i.e. feature selection, preprocessing and classification. I believe that while having such a constraint helps a lot with the task of classification, it does not make the system impractical since the users of the final constraint could be instructed to take their input pictures in a certain way (i.e. butterfly lying on its back with wings spread, horizontal and vertical axes aligned with the image plane) given that the process is not designed to be used in the field on live butterflies (which would make it hard to get the pose described above).

Previous Work

Curiously, previous work which mentioned the phrases "butterfly" and "pattern recognition" is almost non-existent (at least as far as I can tell from my research using Columbia Library's CLIO system). I was able to find a 3-page paper written by a team of researchers in Korea that was published in 2006, titled "Development of a classification algorithm for butterflies and ladybugs". Although they report high classification rates in the neighborhood of 80%-90% for butterflies and ladybugs respectively, unfortunately the paper is very vague in terms of the actual methodology that their team followed other then mentioning the steps in a few sentences using key phrases such as "RGB value analysis", "Fourier analysis" and "comparison of differential coefficients".

Since the plant species identification project was the original inspiration for my project, I also read that paper to get insight on the features and methods they looked at when considering plant identification. Fortunately, that paper is much more rich in detail as compared to the previous paper I looked at. Even though many of the methods described in it (e.g. Inner Distance Shape Context) were impractical for me to implement by myself in the given amount of time, I did get some inspiration from their work, such as looking into utilizing the HSV color space for segmen-

tation. At the end, I went a different direction with my segmentation problem, but the HSV space was a good place start.

METHODS

Data Collection

As I mentioned in the Objectives section, to allow myself to focus on specific areas of this recognition problem, I made a decision to put limit my input data to a specific set of images with an aligned pose and little-to-none variability in lighting. Although there are many commercial and non-profit websites dedicated to butterflies, I found the pose- and light-invariant database of images I was looking for at a site called butterfliesofamerica.com. To emphasize, the advantage of using a single source of data was that I did not have to worry about aligning my samples and correcting their lighting. All of the images on this website have the butterfly centered in the image, horizontally- and vertically-aligned, wings fully spread and against a uniform white background. The disadvantage of this particular dataset is that it is limited to North America. The elementary reading I have done on butterflies seems to suggest that many of the known butterfly species today live and thrive on tropical climates, which means a dataset which includes other regions around the world might have been more representative of the overall problem.

The actual data collection process consisted of browsing the site for species that had multiple samples and downloading those images to my hard drive, making a note of their label. Another complicating factor that I noticed once I started fleshing out my methodology was the fact that some of the samples in the same family looking drastically different than the other images in that family. Only after a while into the project did I notice that those images were actually taken from the bottom of the butterfly displaying the patterns that are on the bottom of their wings. This become abundantly clear once paying attention to the presence of the butterfly's legs in the picture, which indicate that the butterfly is lying on its back. Given that the 2 sides of the butterfly seem to be so drastically different, I decided to treat the front and back of butterflies as separate classes in my classification algorithm. In other words, if my is presented with a sample of "Blackened Bluewing" taken from the bottom, it should return "Blackened Bluewing - bottom" as the class. Notice that ignoring the pictures taken from the bottom (i.e. excluding them from my database) would have effectively halved my dataset. Another approach that I could have taken was

to associate top- and bottom-images with each other to capture more data for each sample. I will discuss this approach in a little bit more detail in the Future Work section.

To give you an idea of the dataset, here are some descriptive statistics from my database:

Statistic			Value
Number of species			103
Number of images	Total		453
	Per species	Average	4.4
		Median	4
		Min	2
		Max	8
Resolution	Maximum Available		840 x 600 px
	Effective (used during classification)		210 x 150 px

Segmentation

Before we can start with classification, we need to start with collecting some data to describe to butterflies and before we can do that, we need to be able to separate the portion of the image that contains the butterfly from the background of the image. In the case of face recognition, we were able to achieve this goal by applying an elliptical mask to our image taking advantage of the fact that all faces more or less have the same shape. In the case of butterflies, however, masking is a little bit trickier since the butterflies each have a different shape and we would like to keep that information as a feature to help us with the classification.

A simple method for segmentation is to convert the image to grayscale and then apply a threshold to extract a part of the image that lies in a certain intensity range. Trying this approach on my dataset lead me to conclude that butterfly pictures are not good candidates for this approach, presumably because the wing can have varying intensities along its edge and if any of those intensities happens to be close to the background, we can have cases where the resulting mask is not a closed shape making the following steps in segmentation (e.g. fill, object identification, etc.) extremely difficult.

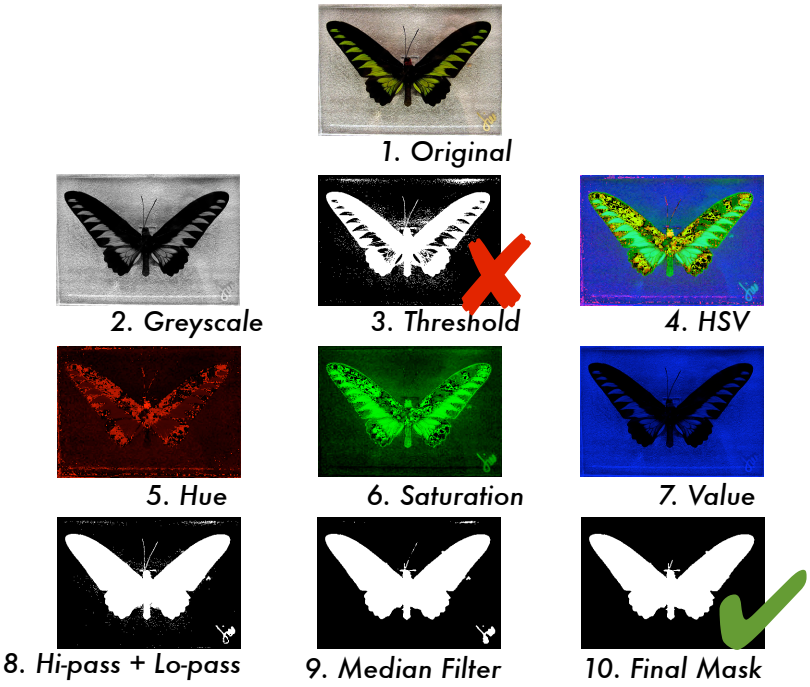
Segmentation in the HSV Space

As I mentioned in the "Previous Work" section, the segmentation methodology used in the plant identification project gave me the inspiration of trying the segmentation in the HSV instead of the grayscale space. The advantage of using the HSV space for segmentation is as follows: Colorful spots within the wing pattern are easy to identify in the saturation space as they are represented with intensity peaks in this space. The complementary (i.e. dark) parts of the wing are equally well represented in the neighboring value space. Applying a high-pass filter to the saturation space and a low-pass filter to the value space gets us 2 complimentary masks, which can be overlaid to get us a great starting point for a mask without large holes.

The initial mask obtained from the HSV space is a good starting point, but we still need a few additional operations to get the final mask that extracts our butterflies. The first of these steps is to remove the "salt & pepper" noise that passes through the filters we have applied in the previous step. A median filter seems to work really well in this case since most of the noise is contained in tiny blobs that consists of only a few pixels. After this step, we should only have 1 or more medium- to large-size blobs in our masks. Since we know that the butterflies are centered and take the most amount of space in the image, we can simply select the largest object ask our final mask.

The series of images included on the next page show the state of the image at various parts of this segmentation procedure.

Segmentation Using HSV



Segmentation using a series of binary operations

Although the segmentation algorithm described on the previous page seemed promising on a small set of samples, finding a set of threshold parameters for the S and V-channels that worked across all of the 100+ species in my initial database proved to be a difficult endeavor.

After a series of failed attempts trying to make the HSV-segmentation work, I decided to revisit the grayscale thresholding method that I had rejected earlier. The approach I tried on this second go-around was to set the initial threshold very high to obtain only a thin binary silhouette of the butterfly, and grow that silhouette using a series of morphological operations available Matlab's image processing toolbox. In fact, after a certain amount of trial-and-error, I was able to find a series of those morphological operations that gave me a satisfactory mask for all 450+ images in my database.

One of the big discoveries was using a "bottom-hat" operator with an octagonal structuring element. Because of the many nearly 45 degree angles present in the geometry of the butterfly, this mask was able to get between the crevices of the wing and clean out the artifacts that were coming through, for example due to shadows cast on the background by the wing.

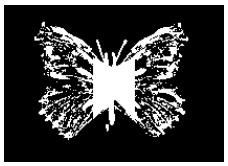
Yet another large step towards a universal segmentation algorithm came from realizing that all butterflies, no matter how distinct their wing-shapes are away from the body, share a common feature which is their body and how the wings are making an wide-angle with the body towards the head and the bottom of the animal on either side. After having this realization, I supplied my algorithm with a starting point which remedied a lot problems that were happening earlier when if the original silhouette had disconnections between the body and the wings, some of the operations were greedy enough to erase the whole wing from the mask.

Below is high-level description of my segmentation algorithm with accompanying images of a sample mask at each step.

Segmentation Using a Series of Binary Morphological Operators

1. Original

2. High Threshold



3. Silhouette + Initial Core

4. Dilate

5. Bridge

6. Median Filter

7. Dilate

8. Erode

9. Final Mask

Notice how the shadow cast by the butterfly's wings is successfully left out of the mask.

Feature Selection and Extraction

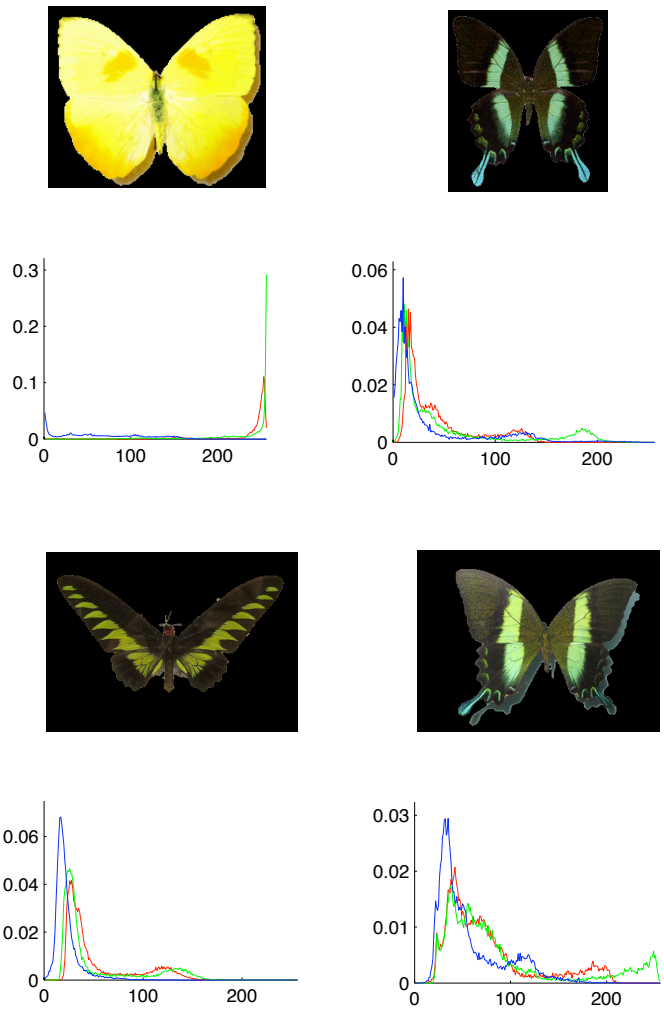
Looking over the images in my dataset 2 features jumped out at me as the most promising features of identifying a butterfly: color and shape. For the remainder of this section I will discuss how I went about extracting and processing those 2 features.

Color

What piece of information should I look at in an image if I want to tell a butterfly whose dominating color is blue from another one whose dominating color is yellow? It seemed to me that the most straight forward way of going about this would be look at the color distribution in images which is traditionally done by looking at a histogram that displays the 3-channels of the RGB space as overlaid histograms.

To compare 2 butterflies using RGB histograms, we need to come up with a similarity metric based on the RGB histogram. I tried several approaches to arrive at a good metric. One way to accomplish this is to think of the "distance" between 2 RGB histograms and take the sum of squared differences at each point of the histogram (i.e Euclidean pairwise distance). While this method got decent classification results I noticed that sometimes images that visibly had similar colors were found to have a good amount of distance between them. This is most likely due to the fact that the histogram data may be noisy and too precise for this application. Indeed, reducing the number of buckets in our RGB histogram (i.e. combining several buckets into one therefore giving us a smoother curve) resulted in improved classification rates. Other approaches I tried quickly but did not perform as well were using the Bhattacharyya and Chi Squared distances between the histograms, which take it into account that these histograms are based on a probability distribution. However, my classification rate actually suffered when using those 2 distance metrics (slightly, but still in the negative direction), so I decided to stick with the Euclidean distance as described earlier.

Color Distribution



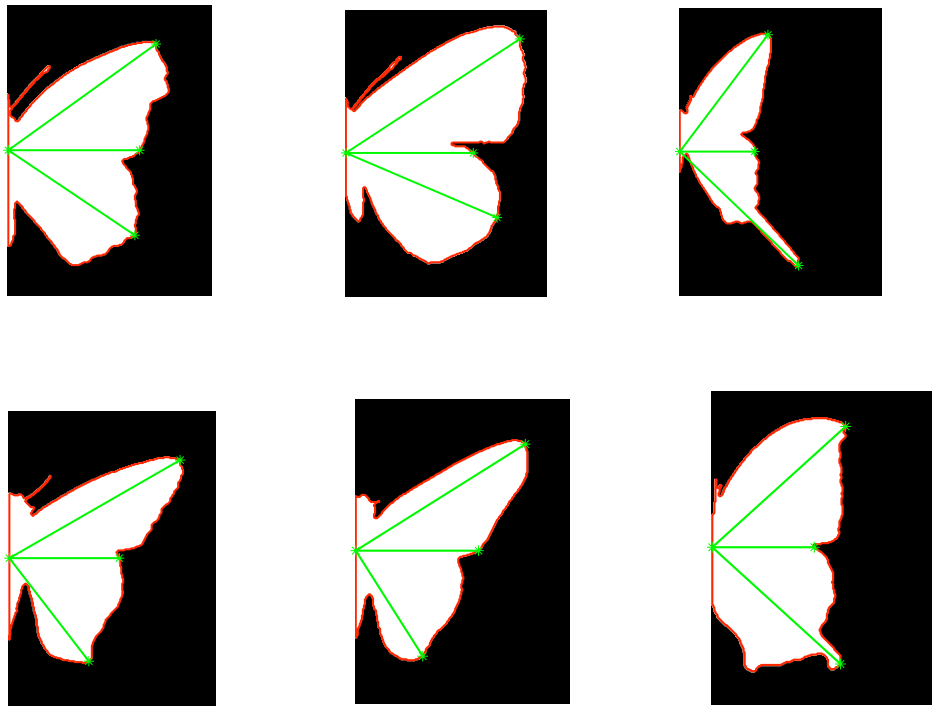
Shape

When looking at the shape of a butterfly, the aspects of their geometry that stood out to me as being discriminating between species were the relationship between the top part of the wing with the bottom part of the wing. More specifically, I noticed that some butterflies have upper wings that stick out 2-3 further than their bottom wings. Additionally, it seemed to be that even if the wings were of equal length, sometimes the angle between changes between species. With those 2 thoughts in my head, I implemented the following geometrical feature extraction algorithm: First of all, we can take advantage of the symmetry of the butterfly along the y-axis and concentrate our efforts on either left- or right-half of the image (i.e. by split image along the y-axis). The

aim of this algorithm is to find the top- and bottom-corners of the wing. If we have those 2 points, we can easily calculate the lengths of the top- and bottom-wings as well as the angle between them the horizontal axis. Since wings are complex shapes, I made the simplifying assumption that the point along contour that is furthest away from the centroid of the butterfly is the "corner" of the wing. Although one can visually confirm that this is not always the case, I felt that it provided us with a good estimate of the wing geometry, at least enough information to discriminate between butterfly species that may have the similar color distributions. Identifying the contour and finding the furthest point on the top half and then the bottom half was a simple to code taking advantage of Matlab's own boundary function, as well as a fast (i.e. vectorized) implementation of the distance finding function (i.e. to find the distance of each point along the contour from the butterfly's centroid).

Finally, once I had the coordinates of the corners, I decided to only keep the following metrics: ratio of top-wing lengths vs bottom-wing length, angle between the top-wing and the horizontal axis, and the angle between the bottom-wing and the horizontal axis. The image included below highlights the geometric features obtained from a particular sample.

Geometric Feature Extraction

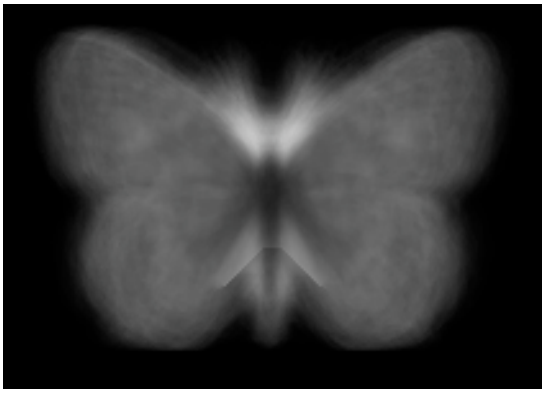


Pixel-wise Comparison with Linear Transformations

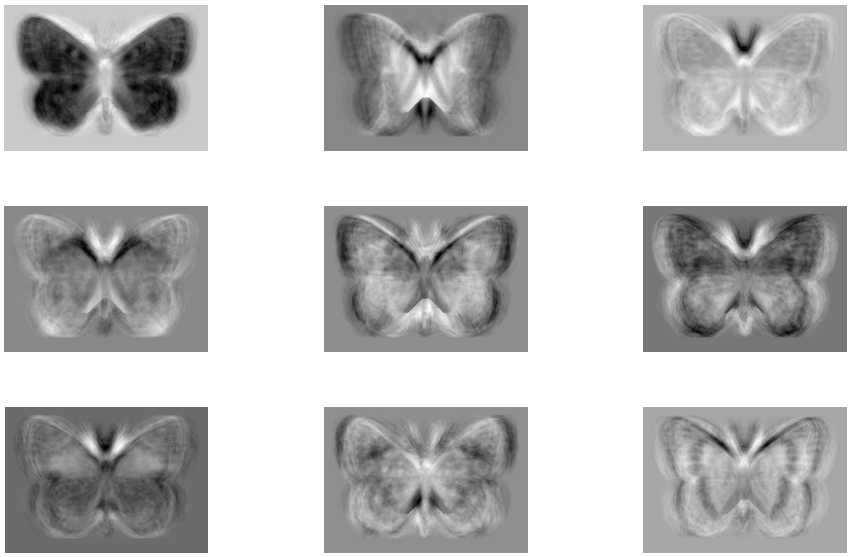
Similar to what happened with segmentation, my initial pursuit of searching for specific features that I felt were going to be discriminating in this case, turned out to be not so discriminating. Although I have not confirmed this mathematically, my intuition tells me that the reason these features did not work well in classification may be the impact of even a small amount of noise when one is dealing with smaller data sets (note that for a handful of our 103 distinct species-direction combinations, there are only 2 images available - detailed statistics on the initial data base can be found in the Appendix).

After tinkering with a few parameters with the hope of getting decent classification results using color and geometric features, I decided to take a different path and apply the tried-and-true pixel wise intensity comparison method (and the associated linear transformation) to see if they work on these butterflies and well as they do on human faces. Sure enough, even my initial attempts testing various combinations were much more successful then what I was able get using the first 2 features I looked at. For a detailed look on exact results from my experiments, please jump ahead to the "Results" section. Before we discuss the classification methods and results, I want to quickly mention that I was able to project the original 31,500-dimensional space down to a 350-dimensional space while maintaining the same level of separability as the original high-dimensional space. As I suspect that you have never seen Eigen- and Fisher-flies before, they are included below for your reference:

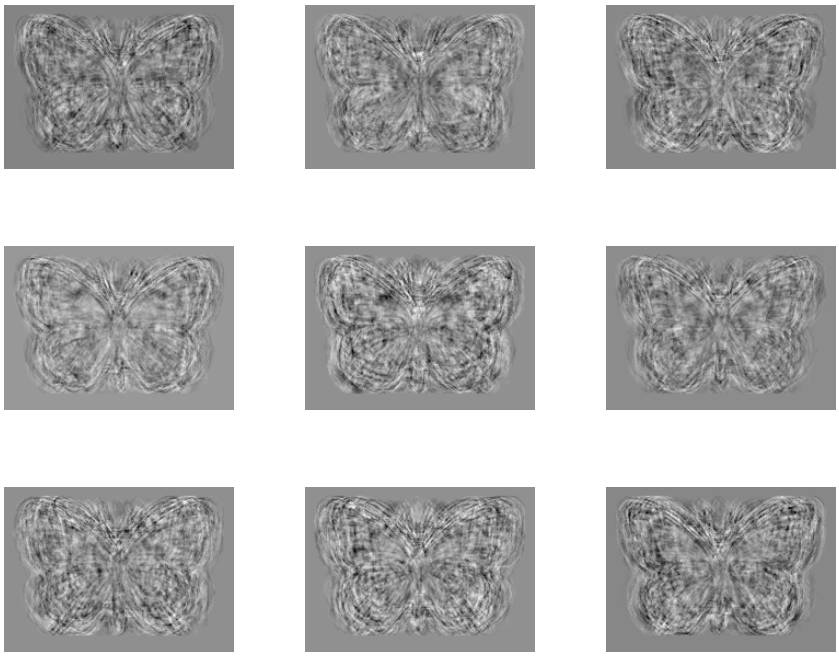
The "Average" Butterfly
from the my initial Database



"Eigen-flies"



"Fisher-flies"



Classification

Similar to Lab #6 (Face Recognition), when the "heavy-lifting" in a classification problem is done in pre-processing step, the classifier can be a simple one and that was certainly the case for since my final classifier uses a simple nearest neighbor (NN) algorithm to assign the class of a sample. In the "Results" section, you will notice that running the same classifier on pre-processed vs not pre-processed data is like night and day in terms of classification performance.

RESULTS

Impact of Various Preprocessing Steps on KNN Classifier Performance

Mask Applied?	FLD Transformation	Error Rate
No	No	57.1%
Yes	No	0.0%
No	Yes	14.3%
Yes	Yes	14.9%

Notice the large difference in error rate between having a mask and not having it in the case of running KNN on the non-transformed images. In contrast, once the data is projected to a lower dimensional space using FLD, the classifier does not seem to need the mask at all (the error rates are almost identical between using a mask and not using one when the data is projected using FLD).

FUTURE WORK

Additional Data

As with any classification task, the more data the better we can gauge its usefulness and the better can train to generalize well under various circumstances. As I noted earlier, although I was able to pull 400+ images from butterfliesofamerica.com in a short amount of time, I would have loved to work with a larger database, especially one which contained exotic species from the tropics.

Additional Features

Since my initial feature set (i.e. color and geometric points) did not work too well, I would have like to explored some additional features, some of which I'm highlighting here:

Feature	Description
Pattern (in Spectrum Space)	As much as the shape of the wing, it is also the patterns on it that make a butterfly unique. I wonder if pattern dependent features extracted from the spectrum space, similar to the features we were provided for the very first lab (e.g. gradient, LaPlacian, etc.).
Shape (curvature)	Aside from the distance of the wing top from the body, I was thinking what other geometric features could be important and realized that the curvature of the wings contour could be a distinguishing feature. Extracting that feature would require a more sophisticated calculation that the one I put together to get the furthest point from the center.
Pattern Matching	Building on the idea of using the patterns on the wing, I was wondering if a feature/pattern detector such as SIFT could be used to count similarities between 2 samples.
Local color matching	I have not heard of an instance of pixel-wise matching done on all three channels instead of just the grayscale intensity. Presumably because in most cases (e.g. faces) it doesn't add much information. I wonder if the additional information in the color space may be useful in the case of identifying something as colorful as the butterfly

Feature	Description
Matching on front & back simultaneously	As I mentioned earlier in the report, my data included pictures taken from the bottom and from the top of each butterfly and most of the time the sides of the same butterfly do not look at all similar. For simplicities sake, I chose to treat the top and the bottom picture as separate species that may classifier had to recognize separately. In the case of a tool that might be used out on the field or trying make classifications amongst collected species at a museum, I wonder if a classifier that compared the top and bottom of the sample with the top and bottoms of all known species simultaneously to get the highest level of accuracy.

Less Constraints

Last but not least, similar to the face recognition problem, once we tackle the pose/lighting constraint variety, we could move on the attempting to recognize butterflies in pictures where they can be in their natural habitat and there for not necessarily aligned with the image plane as was the case with all the pictures in my initial database. Surely, that would be a much more difficult problem and presumably more useful for other people, as well.

Appendix

List of butterfly species and number of images included in initial database:

Species-Direction	#	Species-Direction	#	Species-Direction	#	Species-Direction	#
Darkened White-bottom	8	Long-tailed Metalmark-b	5	Two-Tailed Tiger Swallowtail-	4	Blue-stiched Eighty-eight-top	4
Darkened White-top	8	Deep-blue Eyed-Metalm	5	Northern Blue-top	4	Polydamas Swallowtail-top	3
Sandhill Skipper-bottom	8	Deep-blue Eyed-Metalm	5	Northwestern Fritillary-bottor	4	Purple-topped Euselasia-botto	3
Sandhill Skipper-top	8	Emerald-patched Cattleh	5	Black-veined Leafwing-top	4	Clodius Parnassian-bottom	3
Short-tailed Swallowtail-bottor	6	Emerald-patched Cattleh	5	Blackened Bluewing-bottom	4	Clodius Parnassian-top	3
Short-tailed Swallowtail-top	6	American Snout-bottom	5	Northwestern Fritillary-top	4	Viceroy-bottom	3
Green-patch Swallowtail-top	6	American Snout-top	5	One-spotted Prepona-bottom	4	Viceroy-top	3
Hammock Skipper-bottom	6	Long-tailed Metalmark-t	5	One-spotted Prepona-top	4	Purple-topped Euselasia-top	3
Stub-tailed Morpho-bottom	6	Margined White-bottom	5	Orange Banner-bottom	4	Red Cracker-bottom	3
Deasert Orangetip-bottom	6	Margined White-top	5	White-spotted Agrias-bottom	4	Great Southern White-bottom	3
Deasert Orangetip-top	6	Mercurial Skipper-botto	5	Two-Tailed Tiger Swallowtail-	4	Great Southern White-top	3
Anna's Eighty-eight-bottom	6	Mercurial Skipper-top	5	Variable Cattleheart-bottom	4	Clouded Sulphur-bottom	3
Anna's Eighty-eight-top	6	Monarch-bottom	5	Variable Cattleheart-top	4	Clouded Sulphur-top	3
Stub-tailed Morpho-top	6	Eunoe Mimic-White-bot	5	White-spotted Agrias-top	4	Red Cracker-top	3
Theona Checkerspot-bottom	6	Eunoe Mimic-White-top	4	Blackened Bluewing-top	4	Rhesus Skipper-bottom	3
Barred Yellow-bottom	6	Monarch-top	4	Orange Banner-top	4	Common Morpho-bottom	3
Barred Yellow-top	6	Mormon Fritillary-botto	4	Orange-spotted Prepona-bott	4	Common Morpho-top	3
Theona Checkerspot-top	6	Mormon Fritillary-top	4	Blue Copper-bottom	4	Rhesus Skipper-top	3
Tiger Mimic-White-bottom	6	Mormon Metalmark-bott	4	Blue Copper-top	4	Rocky Mountain Parnassian-bo	3
Hammock Skipper-top	6	Mormon Metalmark-top	4	Orange-spotted Prepona-top	4	Green Anglewing-bottom	3
Leanira Checkerspot-bottom	6	Northern Blue-bottom	4	Pointed Leafwing-bottom	4	Green Anglewing-top	3
Tiger Mimic-White-top	5	Eurytides_marcellus-bot	4	Yellow Kite-Swallowtail-botto	4	Weidemeyer's Admiral-botton	2
True Cattleheart-bottom	5	Eurytides_marcellus-top	4	Yellow Kite-Swallowtail-top	4	Weidemeyer's Admiral-top	2
American Copper-bottom	5	Great Blue Hairstreak-bo	4	Pointed Leafwing-top	4	Rocky Mountain Parnassian-to	2
American Copper-top	5	Great Blue Hairstreak-to	4	Polydamas Swallowtail-bottom	4	Green-patch Swallowtail-botto	2
Leanira Checkerspot-top	5	True Cattleheart-top	4	Blue-stiched Eighty-eight-bot	4		