Image Detection: Large and Small Feature Extraction

GOUCHER college

Sam Shapiro, Isabelle Pardew, and Phong Le (Faculty Advisor)
Center for Data, Mathematical and Computational Sciences

Abstract

We present our progress on the comparison of different image recognition algorithms. We studied large feature extraction with a variation of the Viola-Jones face detection algorithm. This algorithm attempts to identify faces by comparing them to certain "**Haar features**." The most useful features are selected by a training algorithm called **Adaboost.** The detection time is then decreased through a specially trained "**cascaded classifier**." In some initial tests, our version of this algorithm was able to distinguish between faces and non-faces with 92% accuracy. Small feature extraction was studied with **principal component analysis** and the **Weyl representation**. Our goal is to compare how these two algorithms perform at identifying cancerous cells.

Haar Features

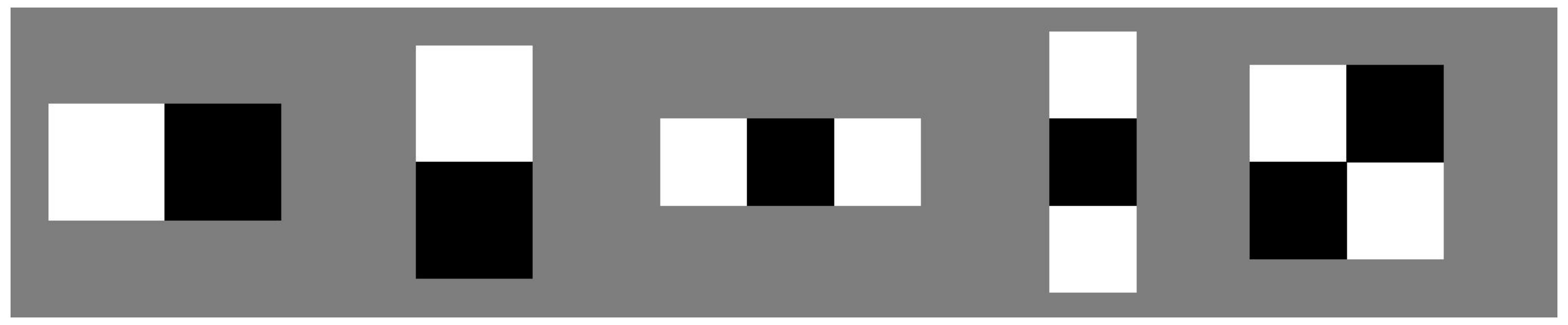

Figure 1: The five basic features we used. All other features can be created from these.

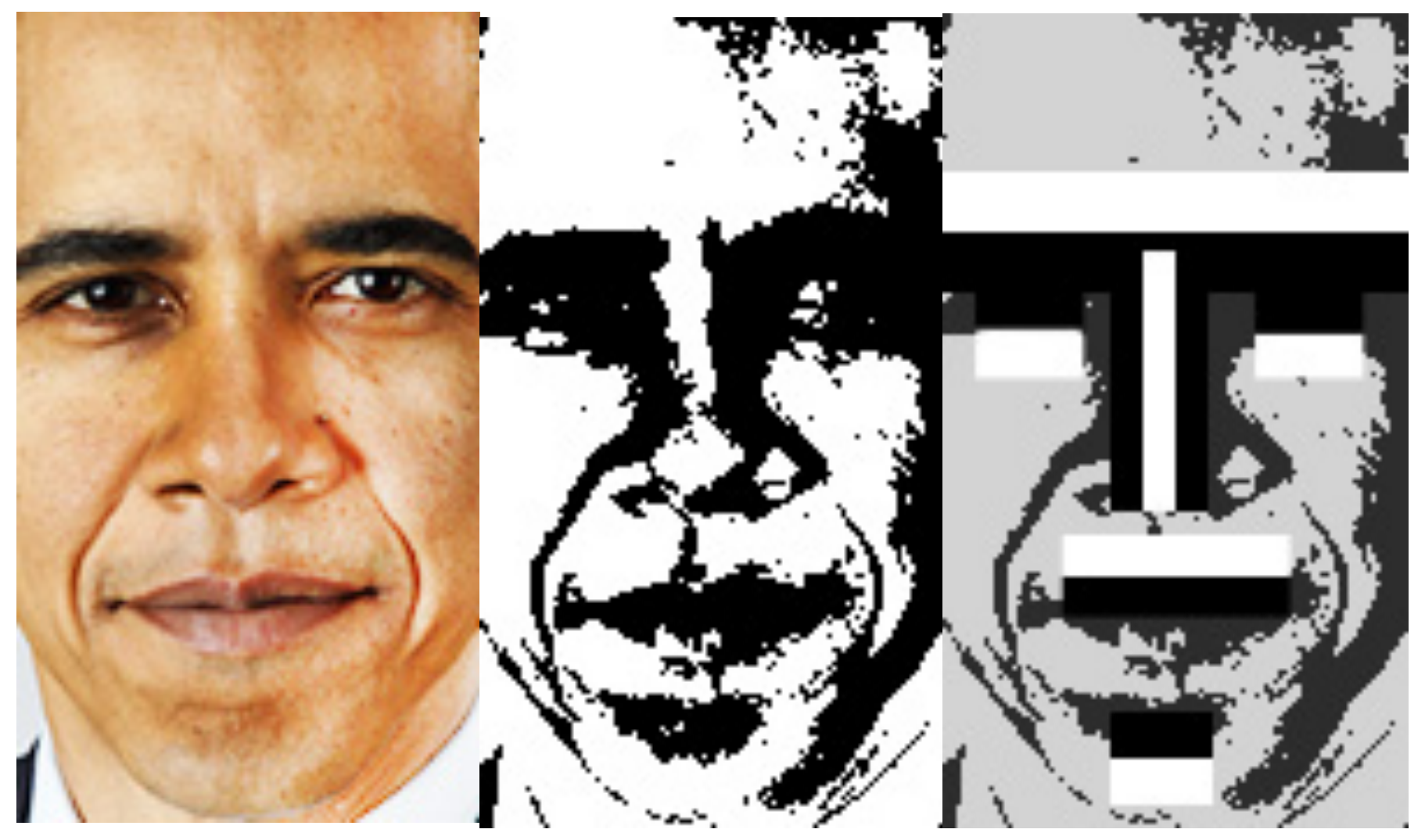

Figure 2: An example of how Haar features are used to represent a face. [1]

Many face detection algorithms use **Haar features** to identify large facial features. Identifying major features this way is the basic idea behind **large feature extraction**. All of the Haar features that are needed to identify a face may be derived by stretching and inverting the colors of the five features shown in figure 1, In a typical face detection application there are about 160,000 different features to consider![2] To narrow this down to a reasonable number we use **Adaboost**.

References and Databases

1. Arubas, Eyal. "Face Detection and Recognition (Theory and Practice)." *Eyal's Technical Blog*. N.p., 6 Apr. 2013. Web. <http://eyalarubas.com/face-detection-and-recognition.html>.
2. Viola, Paul, and Michael J. Jones. "Robust Real-Time Face Detection." *International Journal of Computer Vision* 57.2 (2004): 137-154.
3. Jensen, Ole Helvig. "Implementing the Viola-Jones Face Detection Algorithm." Diss. Technical U of Denmark, 2008. Web.
4. Tutorial-haartraining (GitHub database from user Sonots)
5. NIST Mugshot Identification Database (MID)
6. Yale Face Database
7. AT&T "The Database of Faces"

Adaboost

Adaboost is a machine learning algorithm that selects the best features for face detection. We start out with a large number of **weak classifiers**. A weak classifier is given by $h(x, f, p, \theta) = 1\ if\ pf(x) < p\theta, and = 0\ otherwise$. Where x is an image, f is a Haar feature, θ is a threshold, $f(x)$ is how well the feature matches the image and p is polarity, 1 or -1. $h(x, f, p, \theta) = 1$ means there is a face, and $h(x, f, p, \theta) = 0$ means there is no face. A weak classier should be able to identify faces with slightly above 50% accuracy.

We want to create a **strong classifier** ($C(x)$), a linear combination of the best weak classifiers. This is more accurate than a single weak classifier. To do this we use **Adaboost**, which runs for T training rounds. For $1 \leq t \leq T$ we find the weak classifier which minimizes the weighted error: $\epsilon_t = \sum_i w_i |h(x_i, f_t, p_t, \theta_t) - y_i|$ where f_t, p_t, θ_t minimize ϵ_t, and y_i=1,0 for faces and non-faces respectively. The weights w_i are updated for the next training round. We define $h_t(x) = h(x, f_t, p_t, \theta_t)$ and $\alpha_t = log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$. Our strong classifier is then: $C(x) = 1$ if $\sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} a_t$, and $= 0$ otherwise. [2]

Cascaded Classifiers

Evaluating a single **strong classifier** at every location on an image is inefficient, so we use a **cascaded classifier**. A cascaded classifier consists of many strong classifiers, called **layers**. The goal of each layer is to determine whether a sub-window is "definitely not a face" or "maybe a face." [3] Each layer is trained to have a very high detection rate, with the trade-off of a significant false positive rate.

The early layers contain only a few **weak classifiers**, so they are quick to evaluate. If a layer determines that a sub-window is possibly a face, it continues to the next layer. If a layer determines that a sub-window is definitely not a face, then the detector immediately proceeds to the next sub-window. So the vast majority of sub-windows are evaluated quickly

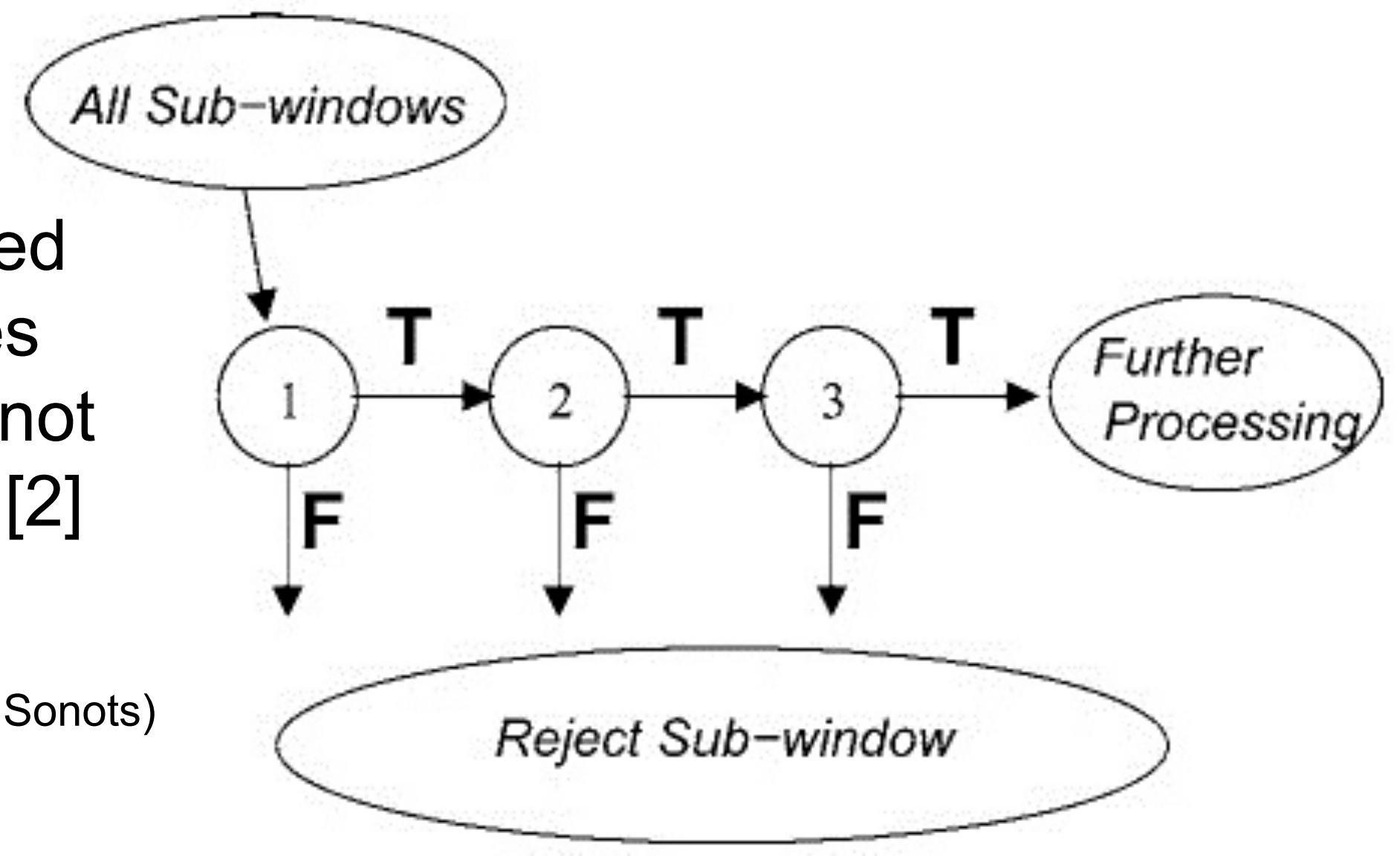

Figure 3: A three layer Cascaded classifier. If any layer determines that a sub-window is "definitely not a face" that window is rejected. [2]

Detection Results

Here we present the results from a detector trained by **Adaboost**. It was trained on 575 images from databases [4], [5], and [6], with five training rounds. The people in every positive training example were directly facing the camera. The four test data sets contained 100 – 150 images each. None of the test images were in the training set.

Test Image Source	Detection Rate	False Positive Rate	Total Accuracy
Sonots [4]	100%	16%	92%
NIST MID [5]	78.7%	16%	81.3%
Yale [6]	100%	24%	88%
AT&T [7]	9%	5.3%	45.7%

Note the very low detection rate for the AT&T test set. This is because many of the people in that dataset are not directly facing the camera, unlike the training set. One way to improve the accuracy of our detector is to use a training set that contains pictures of faces from more angles.

Principal Component Analysis (PCA)

PCA is commonly used for dimension reduction on large data sets, but also plays a key role in the **Weyl representation**. **PCA** takes a data set, and finds the most variance across all the data. The dimensions with the most variance can recreate the data set with lower dimensions, making it simpler to analyze. In effect, the new graph displays how much variance there is in the data set, which is very useful for analyzing patterns on small features.

PCA can be paired with an algorithm for even better pattern recognition. The small-feature analysis of **PCA** is used concurrently with the **Weyl representation**, which prioritizes superstructure over substructure. This is useful when we are trying to represent data differently. Coordinate indices allow us to break down binary groups, or expand them as needed, which is useful when dealing with images.

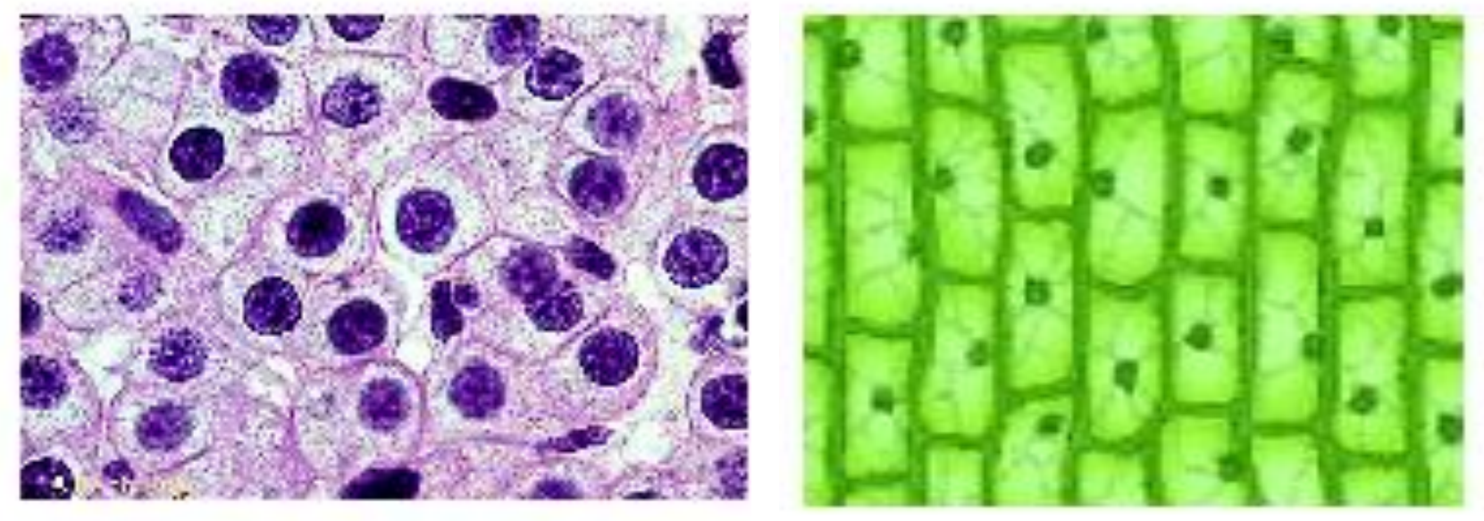

Figure 4

The images shown in **Fig. 4** are human skin cells, and plant cells, respectively. When we compared their patterns and structures using the **Weyl representation** and **PCA, the** algorithm was able to effectively distinguish human cells and those of a plant. **Fig. 5** is a graph showing that the distinction between these two patterns is fairly successful.

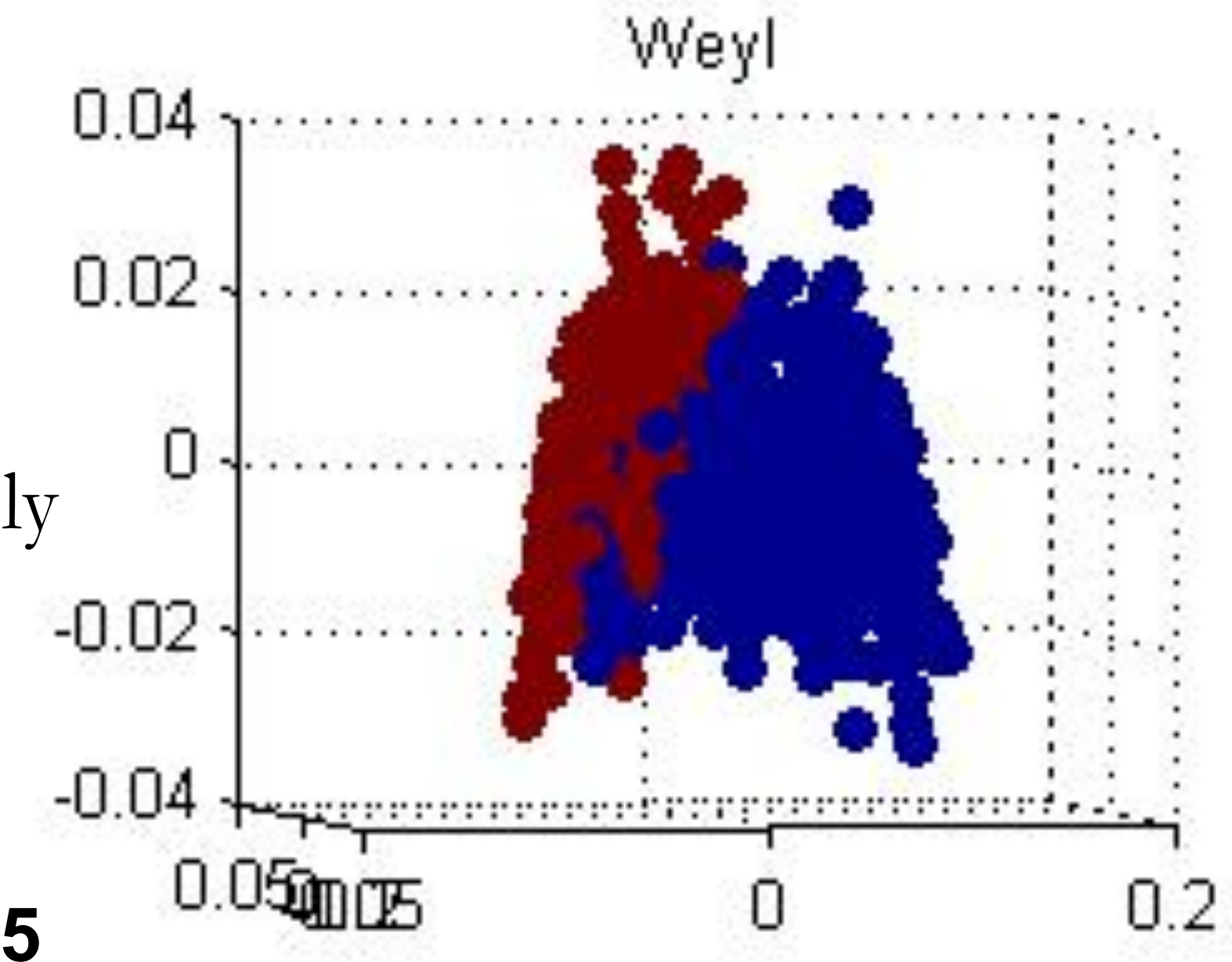

Figure 5