國 立 交 通 大 學

統計學研究所

碩 士 論 文

果蠅嗅覺神經影像之三維結構分析及統計分類

Statistical Analysis and Classification for 3D Structure of Drosophila Calyx Images

研 究 生：陳瑜達

指導教授：盧鴻興 教授

中 華 民 國 九 十 七 年 六 月

果蠅嗅覺神經影像之三維結構分析及統計分類

Statistical Analysis and Classification for 3D Structure of Drosophila Calyx Images

研 究 生：陳瑜達　　　　Student：Yu-Da Chen

指導教授：盧鴻興 教授　　Advisor：Horng-Shing Lu

國 立 交 通 大 學

統計學研究所

碩 士 論 文

A Thesis
Submitted to Institute of Statistics
College of Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in

Statistics

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

果蠅嗅覺神經影像之三維結構分析及統計分類

研究生：陳瑜達　　　　指導教授：盧鴻興 教授

國立交通大學統計學研究所 碩士班

摘要

本研究的主要目的是建構一個自動化流程的三維果蠅嗅覺影像的分類器。此次研究的資料是來自國立清華大學腦科學中心江安世博士所提供的 125張高解析度LSM影像檔，這些影像皆是以果蠅的嗅覺腦區(Antennal Lobe)的方位來命名分成六種類別，分別是DA1、DL1、DL3、DM1、DM2和VL2a。經由扣除可能是實驗染色錯誤的神經影像，剩餘113張影像。由於影像資料有太多雜訊，在此我們藉由對每張神經影像裡的每個物件做標記來達到影像分割的自動化，取出我們需要的神經來達到去雜訊的目的。之後我們針對神經影像取出數種較穩健的特徵值，藉由這些特徵值來區別各種神經影像在空間上的分佈情形。對影像取完特徵值之後，我們對這些特徵值使用逆分層回歸（Sliced inverse regression）可以幫助我們提升分類的正確率。最後使用Weka及R中的SVM,J48,IBk,OneR做統計分類及預測。在此各種分類器的分類結果皆以leave-one-out的cross-validation正確率當做評估的標準。

關鍵詞：去雜訊、擷取特徵值、逆分層回歸、Weka。

Statistical Analysis and Classification for 3D Structure of Drosophila Calyx Images

Student：Yu-Da Chen　　Advisor：Horng-Shing Lu

Institute of Statistics

National Chiao Tung University

Abstract

The goal of our research is to construct an automated process to classify 3D Drosophila calyx images. The 125 high resolution LSM images were administered by Ann-Shyn Chiang from the Department of Life Science at National Tsing Hua University. Those images are classified into six categories that are named by their position in the Antennal Lobe. The six categories are named DA1,DL1,DL3,DM1,DM2 and VL2a. By removing some wrong images that may be caused by experimental errors, there remain 113 images, so we just do a classification on those 113 images. Because the images have too much noise, here we use volume filter to extract useful neurons from images to remove noise automatically. Furthermore, we calculate many robust features based those neuron images. Then we can distinguish different spatial circumstances relative to their dissemination by using those features. After extracting features from images, we use sliced inverse regression on feature data which can help us to increase accuracy. Finally, we use SVM, J48, IBk, and OneR classifiers in Weka and R. Here are different ways to classify results all use leave-one-out cross-validation to evaluate correctness.

Key Words : Remove noise; Features extraction; Sliced inverse regression; Weka.

誌謝

這篇論文的完成，最要感謝盧鴻興教授一年來不倦不悔的指導跟辛苦的批閱和指導，在老師的指點之下，讓我逐漸能將統計應用於生活上，並且引導我學習與研究的方向。除了課業上的知識以外，老師也讓我學到許多求知與做研究的方法，讓我在統計這片領域中，找到我的興趣跟做出研究的成就感。還要感謝博班泰賓學長的大力幫助，楝鴻學長的有效的教導，感謝享屏學長在我遇到問題時能給我意見並且教導我使用 VTK，感謝珮伶學姊帶領我進入影像處理的領域，沒有學姐的幫忙我的論文沒辦法這麼快就進入狀況。感謝珮琦、政言和健鈞同學給予我鼓勵，除了課業上的切磋討論外，閒暇之餘能跟他們閒話家常，交換各種意見與心得，讓我的生活充滿樂趣，在互相鼓勵之下度過研究所的生涯，還有郃嵐、夙吟、香菱、怡玲、佩蓁、筱娟、姿蒨、文廷、重耕、泰佐、翁賢、仲竹、昱緯、彥銘，感謝這些研究所的好朋友們，因為他們我才能順利完成學業，讓我研究所的生活過的多彩多姿。另外也感謝所上每位老師在授課所以給予的觀念，讓我受益良多。加上心愛的家人-爸爸、媽媽、瑜琇、瑞堂、佳華的支持與關懷，讓我非常順利的完成論文。

同時感謝口試委員許文郁教授很認真的看完了我的論文，幫我挑出了論文的許多錯誤讓我非常感動，感謝黃冠華教授和陳君厚教授提供諸多建議，使的本論文更加完善。

剛開始抱著期待與學習的心來到交大統研，系上提供良好的學習環境。現在，我將懷著感謝與難忘的心情離開，在這充滿了班上同學跟老師的回憶，包含了迎新、家聚和送舊等等。我很開心自己是交大統研所的一員，未來也會努力工作為所上爭光。

最後，僅將論文獻給敬愛的父母親和盧鴻興教授，感謝他們的辛勞，我愛他們，就因為有他們，才有現在信心滿滿，不怕未來挑戰的我。所有在我周圍關心我的人，一同分享此篇論文完成之喜悅與榮耀謝謝大家。

陳瑜達 謹 誌于

國立交通大學統計研究所

中華民國九十七年六月

Contents

List of figures

1.Introduction

An organism's sensory system must pass through three procedures: (1) Sensory receptors are stimulated externally and then send information; (2) Information passes through nerves to the cerebrum; (3) The cerebrum accepts information from nerves and analyzes the information. Generally speaking, the hardest part to understand sensory system is the mechanism of central brain. But regarding the olfactory system, there has been no exact conclusion for many years as to the procedure of how the sensory receptors are stimulated externally and then send information. The mechanism of a sensory system has always been a mystery. Not until Richard Axel and Linda B. Buck used their accomplished molecular biology technique in neuroscience did they discover that the sensory receptor on a cell bound the odor, and they finally solved the mystery and won the Nobel Prize in Physiology or Medicine in 2004.

The Life Science Department of National Tsing Hua University uses FocusClearTM, invented by the department, to make the cerebral organization easily observable. Additionally, they use a special genetic engineering technique to pigment the projection neuron fluorescent green in a fly cerebrum. It uses a confocal microscope to scan the organization slice by slice, then uses those slice data to reconstruct a 3D image. Finally, they complete the olfactory neuron circuit of a fly cerebrum, understand the mechanisms how the cerebrum analyzes an olfactory signal and smells odors. They complement the region that was unknown previously.

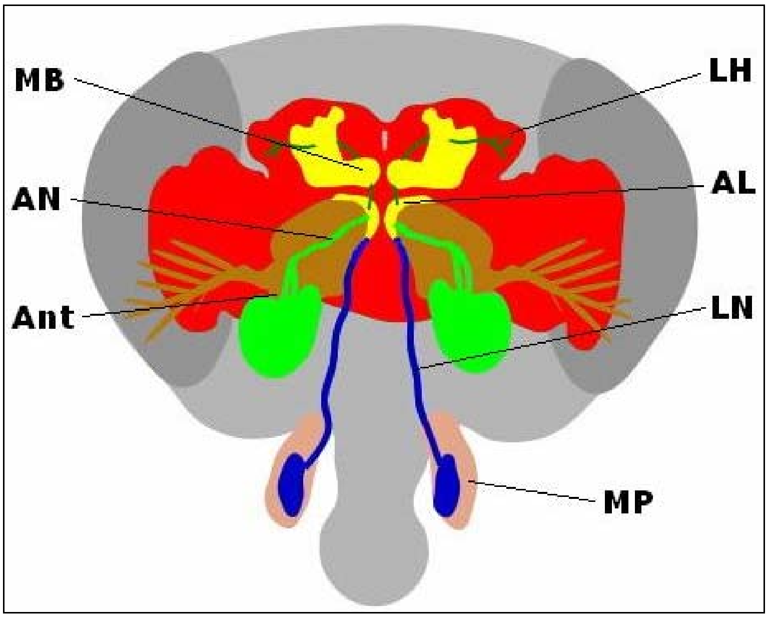


Figure 1. Organization of the *Drosophila* olfactory system.

1.1 Olfactory sensory receptors to antenna lobes

In a fly's olfactory system, odors detected by the receptor distributed on the olfactory sensory nerve ending on the Antenna(Ant) and Maxillary Palp(MP). The Ant and MP have three different types of receptors：club-shaped basiconic sensilla, spine-shaped trichoid sensilla, and small cone-shaped coeloconic sensilla [Stocker, 1994; Couto et al., 2005]. When olfactory receptors bind odors, they first activate a couple of G-proteins (heterotrimeric G-protein), promote to generate the cyclic adenosine monophosphate(cyclic AMP, cAMP), and then open an ion channel to activate the whole olfactory cell [Firestein, 2001;Buck et. al, 1991]. The information sent by the receptor on the MP will be received by ciliated endings of a nerve called the Labial nerve (LN), and the axon will converge to the ipsilateral antennal lobe (AL) (like the olfactory bulb in the mammalians); similarly, the olfactory receptor on the antenna sends information to the glomerulus through the Antennal nerve (AN).

The Ant and MP have about 1200 and 120 olfactory sensory neurons, respectively [Hallem et al., 2004] and have about 62 types of olfactory receptors. Almost one olfactory nerve expresses just one type of olfactory receptor [Hallem et al., 2004; Couto et al., 2005], but there are still some tiny exceptions [Goldman et al., 2005]. When olfactory sensory neuron dendrites receive information of odor, the axon expressing the same olfactory receptor will converge to a single or a few glomeruli in the AL [Kreher et al., 2005; Marin et al., 2002; Jefferis et al., 2001]. From this pattern, the fly's olfactory system has a high degree of specificity.

1.2 Antenna lobes to mushroom body and to lateral horn

Antenna lobes have about 50 glomeruli [Marin et al., 2002; Jefferis et al., 2001] and have about 150 projection neurons peripherally. Each glomerulus sends odor information to about 3-7 projection neurons (equivalent to mammalian mitral/tufted cells), and here one projection neuron just receives information from one glomerulus [Lin et al., 2007]. Then, the axon through three different tracts projects to the Lateral Horn(LH), the inner antenna-cerebrum track(iACT), the medial ACT(mACT), and the outer ACT(oACT) [Marin et al., 2002; Wong et al., 2002]. The majority of the projection neurons project through the iACT to the Mushroom body(MB) calyx and then to the Lateral Horn, and few projection neurons project to the Lateral Horn through the mACT or oACT directly. Even if different projection neurons receive information from the same glomerulus, the patterns of different tracks in the Lateral Horn are very different [Wong et al., 2002].

Table 1-1. The biological terminologies and the corresponding abbreviations used in this study are listed.

OSR	Olfactory Sensory Receptor
Ant	Antenna
MP	Maxillary Palp
cAMP	cyclic Adenosine MonoPhosphate
LN	Labial Nerve
AN	Antennal nerve
AL	Antennal Lobe
OB	Olfactory Bulb
iACT	inner Antenna-Cerebrum Track
mACT	medial Antenna-Cerebrum Track
oACT	outer Antenna-Cerebrum Track
MB	Mushroom Body
LH	Lateral Horn

1.3 Data description

In our research, the LSM images administered by Ann-Shyn Chiang of the Department of Life Science at National Tsing Hua University show projection neuron dendrites receiving information from the glomerulus in the antennal lobe, then converging upon axons and passing through the iACT to the Mushroom body, and finally arriving at the Lateral horn. Because those images do not have information on which projection neurons receive information from which glomeruli, we wish to find some robust features that can represent the pattern of those projection neurons and use those features to do statistical classification.

Table 1-2. The numbers of flies and after removing wrong images in 6 categories are listed, which can be combined as 2 or 3 categories.

6 categories	**3 categories**	**2 categories**	**Number of flies**	**After removing wrong images**
DL1	ab	ab	40	35
VL2a	ac	ac-or-at	25	24
DM1	ab	ab	22	20
DM2	ab	ab	13	10
DL3	at	ac-or-at	13	13
DA1	at	ac-or-at	12	11
Total			125	113

2.Methodology

The goal of our research is to find a statistical method. Then we can utilize different patterns of projection neurons in the Lateral Horn of flies to calculate some features that are able to represent the character of an image and use statistical classification to classify our LSM images. The images are classified into six categories that are named by their position in the Antennal Lobe. The six categories are named DA1,DL1,DL3,DM1,DM2 and VL2a. By removing some wrong images which may be caused by experimental errors, there remain 113 images, so we just do a classification on those 113 images.

Because we hope our method can be utilized to classify any type of neuron, and because we can easily see the colors of projection neuron from the green channel, therefore in this research, our analysis just depends on the green channel and does not consider the respective spatial position in red channel simultaneously. Using the green channel and the red channel simultaneously, we can see the projection neuron dendrites via MB and LH with different spatial circumstances relative to their dissemination.

Because we need to calculate some features that are able to represent the pattern of spread spatially, we must to do preprocessing of those images, otherwise noise will influence the accuracy of the features. Here we use RST-invariant features and sliced inverse regression to preprocess our data features and then utilize SVM and classification trees or some other method to do statistical classification. Then we use leave-one-out cross-validation to evaluate accuracy.

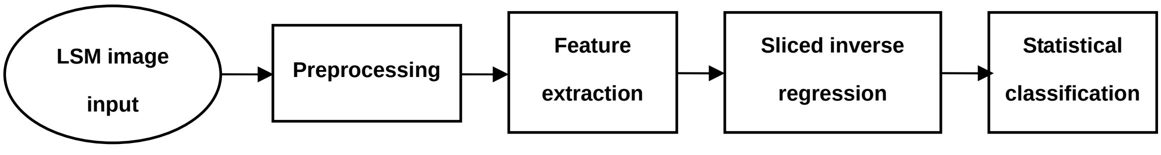


Figure 2. The overall flowchart of our process.

2.1 Image Pre-Processing

Image preprocessing usually includes spatial quantization (or size reduction), gray level quantization (reduce the nymber of bits per pixel), and spatial filter to remove noise or transform color space. Here, we use the flowchart below to do a spatial filter on 3D fly images directly. In conclusion, the projection neuron's axon and dendrites of fly images may be too thin, so if we use a 3x3x3 structure element to do 3D median filter, we can find a fly projection neuron axon cut into many pieces, while we denoise and simultaneously amputate the axon which we do not wish to delete (Figure 4). 3D median filters are not applicable to our fly images. Then we try another method. We use a mean filter to replace the median filter. In images with less noise, it seems suitable (Figure 5). But most images do not alter for the better after the 3D mean filter. Furthermore, in the median filter, we can change to use any quantile to replace the median, or in the mean filter, we can change weight of the 3x3x3 structure element, but all of those changes ameliorate restrictedly, so we don't amplify here.

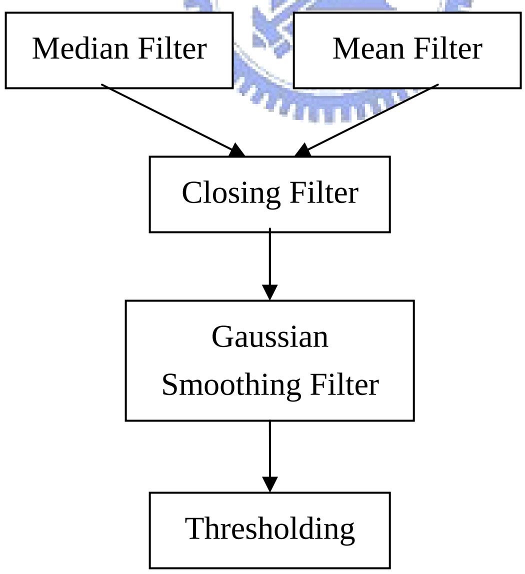


Figure 3. The flowchart of denoising and morphological operations.

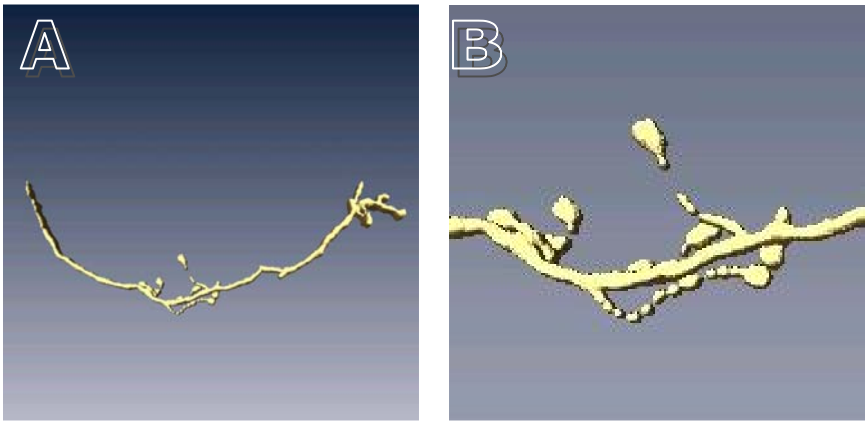


Figure 4(A). The conclusion after the median filter, closing filter, and Gaussian smooth filter, having a threshold at DA1 GH146-singlePN80calyx image.

Figure 4(B). Zoom in of projection neuron passed through the Mushroom body in Figure 4(A), we can see dendrite cut off in many pieces.

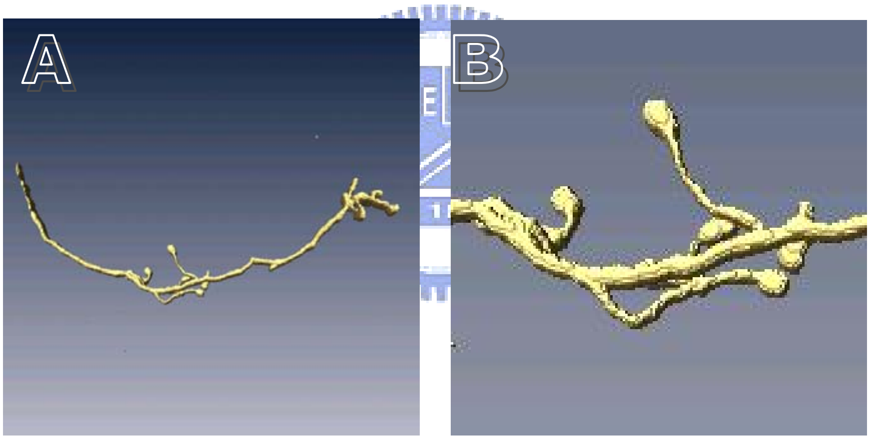


Figure 5(A). The conclusion after the mean filter, closing filter, and Gaussian smooth filter, having a threshold at DA1 GH146-singlePN80calyx image.

Figure 5(B). Zoom in of projection neuron passed through Mushroom body in Figure 5(A), we can see that dendrites are still connected. This result is good than using median filter.

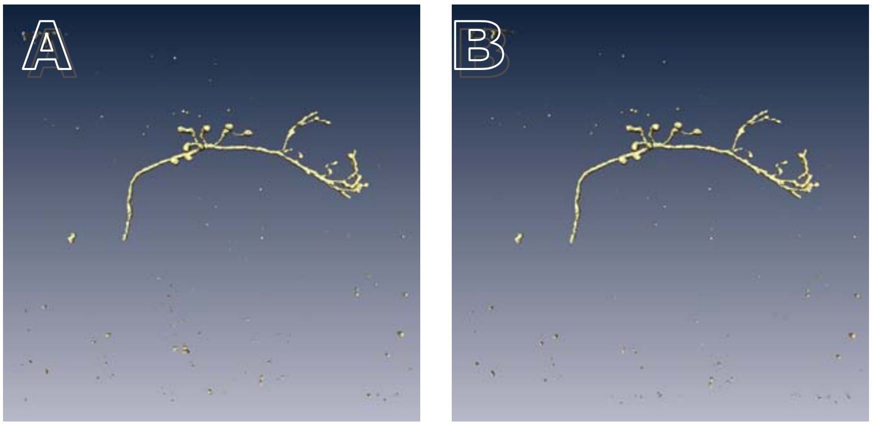


Figure 6(A). The conclusion after the median filter, closing filter, and Gaussian smooth filter, having a threshold at DL1 GH146-singlePN37calyx image.

Figure 6(B). The conclusion after the mean filter, closing filter, and Gaussian smooth filter, having a threshold at DL1 GH146-singlePN37calyx image. Here we can see that denoise methods are usually helpless.

Because common methods are helpless in our 3D LSM images, we utilize a gut concept (which calculates effectively) to denoise of our LSM images. We call this method a volume filter, which is based on object labeling and segmentation. The main idea of our method is to do object labeling when an image is inputted. When every object in the image is labeled , then we can easily calculate the area or volume of each object. After we calculate the volume, we can keep the maximum-volume object, or we can set a threshold, just keep objects with volume greater than the threshold. We interpret our method at length below. Before account for volume filter, we introduce two fundmental definitions about how two objects are connected. In our application, if any two pixels have 26-adjacency relationship, then we consider that the two pixels are connected.

2.1.1 Definition : Neighbors of a pixel

Assume p is a pixel at coordinate (x,y). In a two-dimension plane, p has two horizontal and two vertical neighbors, whose coordinates are (x+1,y), (x-1,y), (x,y+1), and (x,y-1). The set is assembled of four neighbors of p, represented by $N_4(p)$. Observe that the pixels around p still have other four diagonal neighbors whose coordinates are (x+1,y+1), (x-1,y+1), (x-1,y-1), and (x+1,y-1). The set assembled by four diagonal neighbors of p is represented by $N_D(p)$. Assemblimg $N_4(p)$ and $N_D(p)$ is called 8-neighbor and it can be represented by $N_8(p)$, which contains the 8 pixels except p in the 3×3 structure.

2.1.2 Definition : Adjacency of a pixel

Consider the 3×3 structure in a 2D plane with p at the center of the structure element. If there exists a q whose intensity is not zero, and q belongs to $N_4(p)$, then we can say q and p are 4-adjacency. Similarity, if q belongs to $N_8(p)$, q and p are 8-adjacency.

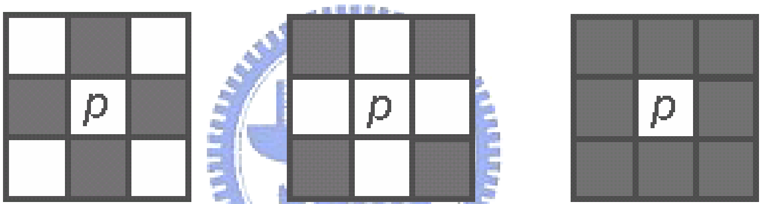


Figure 7(A). ***Figure 7(B).*** ***Figure 7(C).***.

Figure 7(A)(B)(C) are sketch of $N_4(p)$, $N_D(p)$ and $N_8(p)$ respectively.

2.1.3 Algorithm of connected components (Object labeling)

Here we make a example of 8-adjacency.

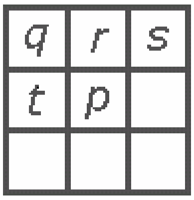


Figure 8. The structure element of connected components algorithm. In each pixel, we just need to check target pixels p q, r, s, and t.

1. If $p = 0$, check next point.

$$2.\ \text{if } p \neq 0,\ \begin{cases} \text{if all } \{q, r, s, t\}=0, \text{ set } p \text{ a new label,} \\ \text{if only one of } \{q, r, s, t\} \neq 0, \text{ assume } q \neq 0, \text{ set label of } p = \text{label of } q, \\ \text{if more than one of } \{q, r, s, t\} \neq 0, \text{ assume } q,\ r \neq 0, \text{ set label of } p = \text{label of } q \text{ or } r, \\ \quad \text{and mark label of others is equal to label of } p. \end{cases}$$

3. merge the same label object.

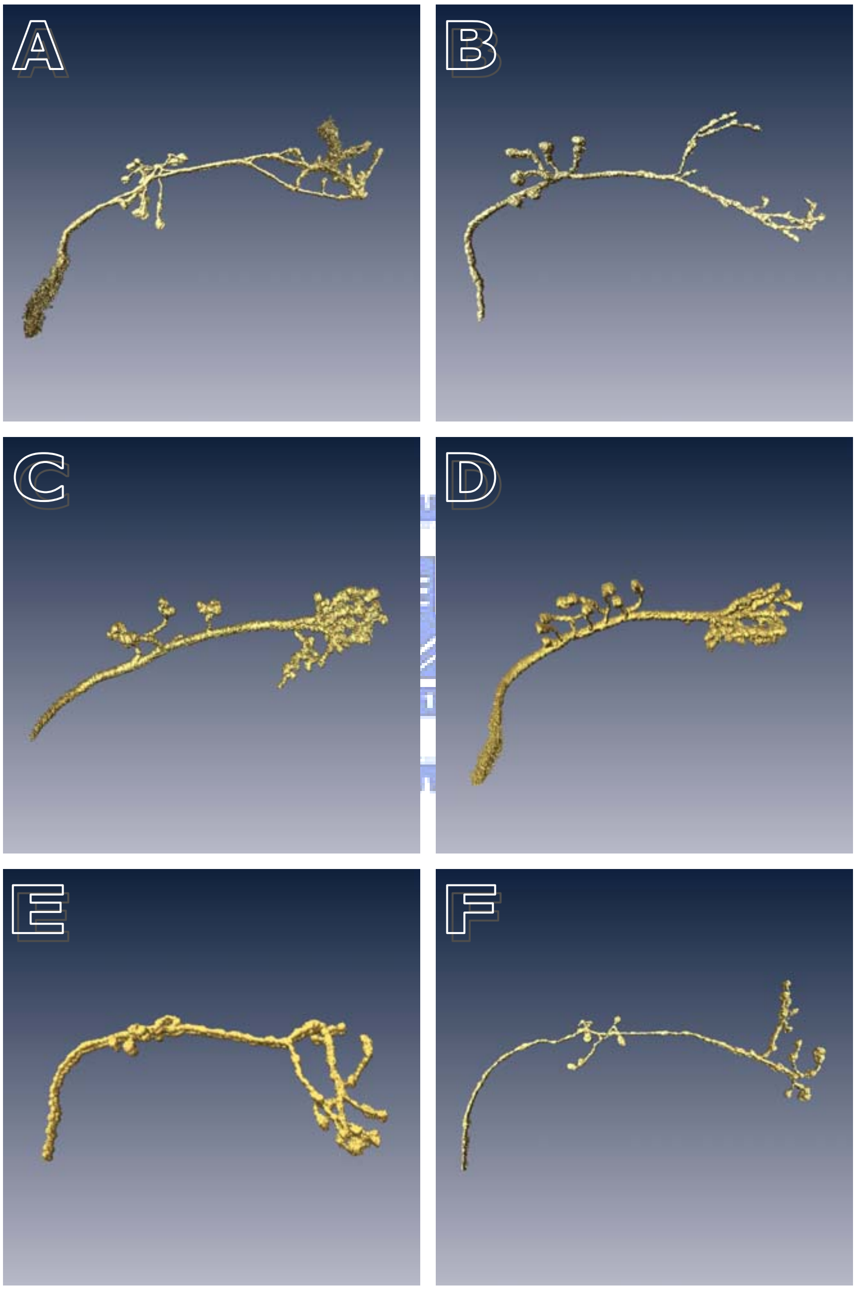


Figure 9. *(A)(B)(C)(D)(E)(F)* are images that after object labeling, have just kept maximum-volume object: GH114-singlePN37calyx in DA1, GH146-singlePN37calyx in DL1, GH193- singlePN37calyx in DM1, GH146-singlePN73calyx in DM2, GH118-singlePN37calyx in VL2a, GH146-singlePN139calyx in DL3 respectively.

3 Feature Extraction

In general in image analysis, we need to extract some high-level important information from an abundance of low-level pixel data. Those procedures are called feature extraction, and sometimes also called data reduction. In our research, the goal is to develop a statistical classification to classify fly calyx images. But every image is not standardized, so they have different size, direction and position. So we have to use more robust statistics to spatially represent the spread of projection neuron. Those features we need are called RST-invariant features. RST means rotation, size, and translation..

3.1 Relative frequency vector of histogram

Here we utilize a histogram to calculate the relative frequency vector (first order probability) [Umbaugh et al., 1997]. Because we focus on the green channel, we can take green channel as gray-level image. In usual gray-level image, each pixel needs one byte, so each pixel has 2^8 combinations and ranges from 0 to 255. If a pixel has 0 intensity, then this pixel seem a black point. If intersity get increase, then the pixel get more bright, and become white point until intensity achieves 255. Here, an image size is about $1024\times1024\times80$, and every image slice may be different, so most of the relative frequency vectors will be close to zero. This will make features between images not have flair and be hard to classify them into six categories. So we take a log transform at the relative frequency vector to help us increase accuracy.

Relative frequency is computed as following:

$$P_{r.f.}(x) = \frac{N(x)}{M} \qquad (3.1\text{-}1)$$

where M is number of pixels. In this research, it is about $1024\times1024\times80$, and $N(x)$ is numbers of gray-level pixel at x intensity.

3.2 Histogram features

After we calculate the relative frequency vector we can compute eight features based on the relative frequency vector. These eight features are mean, standard deviation , coefficient of variation (CV), skewness, kurtosis, energy, entropy, and volume. We can utilize the relative frequency vector to calculate mean efficiently.

$$\bar{x} = \sum_{x=0}^{255} x P_{r.f.}(x) = \sum_{s} \sum_{r} \sum_{c} \frac{I(c,r,s)}{M} \qquad (3.2\text{-}2)$$

where c, r, and s are column, row and slice, respectively. *M* is the numbers of pixels in the image. Mean tells us the overall brightness of the image. If the mean is big, it represents the image is more bright; similarily, if mean is small, then the image is more dark.

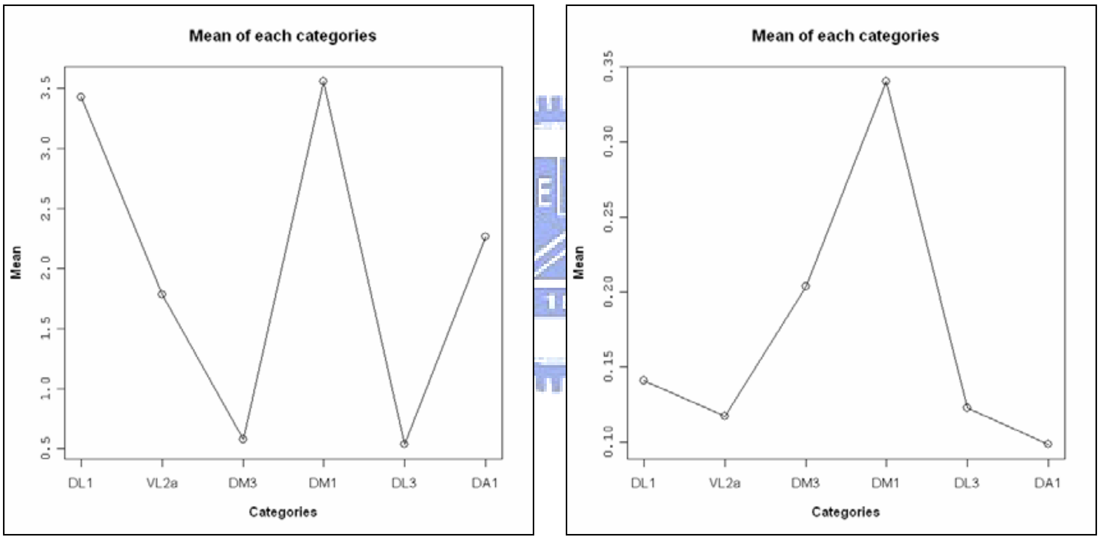


Figure 3-1. Mean of raw (left) and revised (right) data of six categories.

$$\sigma_x = \sqrt{\sum_{x=0}^{255} (x - \bar{x})^2 P(x)} \qquad (3.2\text{-}3)$$

The standard deviation σ_x is root of variance. It can tell us the contrast of image. If the s.d. is big, it represents the contrast is big, the foreground has more difference with background, and the object can be recognized easily.

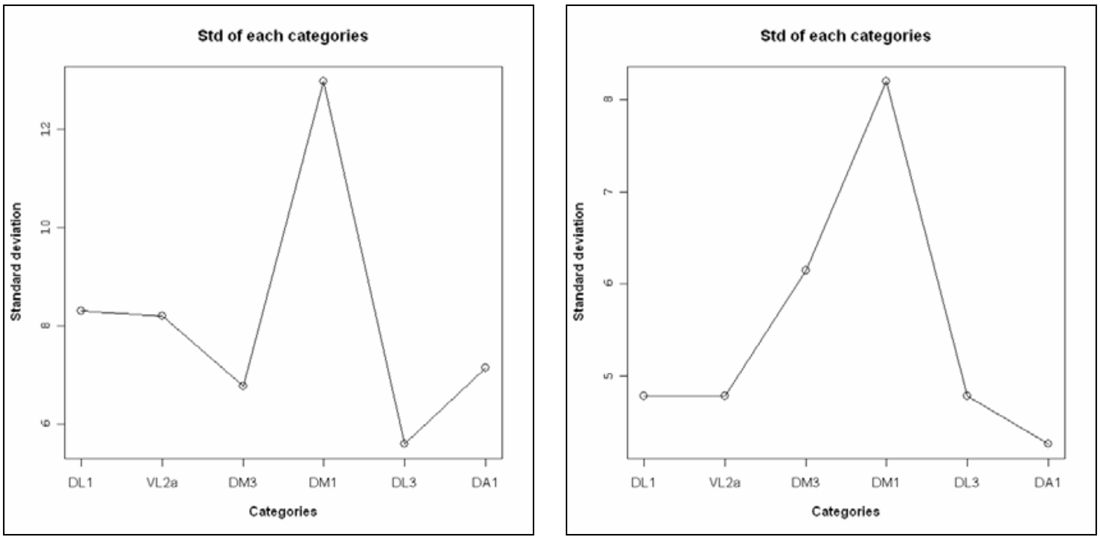


Figure 3-2. Standard deviation of raw (left) and revised (right) data of six categories.

$$CV = \frac{\sigma}{\mu} \tag{3.2-4}$$

Coefficient of variation (CV) is a standard measurement to calculate the spread of data.

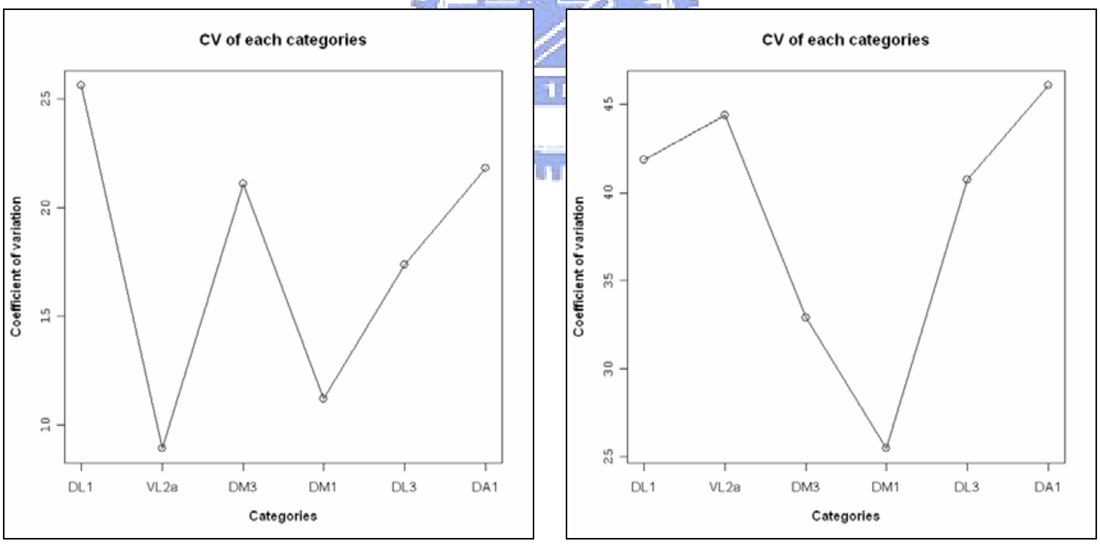


Figure 3-3. CV of raw (left) and revised (right) data of six categories.

$$SKEW = \frac{1}{\sigma_x} \sum_{x=0}^{255} (x - \bar{x})^3 P(x) \tag{3.2-5}$$

Skew can help us to measure the asymmetry of the gray-level distribution. If skew = 0, then the distribution is symmetric. If skew > 0, then the distribution is skewed to the right and has more extreme values on right side. If skew < 0, then the distribution is skewed to the left and has more extreme values on left side. There are two other methods to measure skew, provided by Pearson that calculate more efficiently.

$$SKEW' = \frac{\bar{x} - mode}{\sigma_x} \tag{3.2-6}$$

$$SKEW'' = \frac{3(\bar{x} - median)}{\sigma_x} \tag{3.2-7}$$

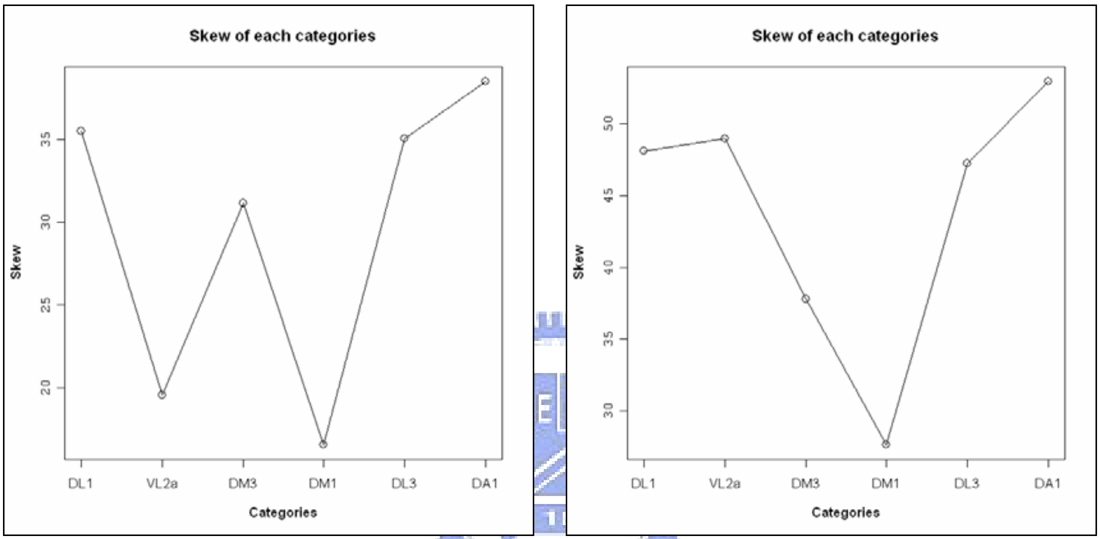


Figure 3-4. Skew of raw (left) and revised (right) data of six categories.

$$KURTOSIS = \frac{\mu_4}{\sigma_x^4} \tag{3.2-8}$$

Kurtosis helps us to measure the peakedness of gray-level distribution. Higher kurtosis tell us that more of the variance is due to infrequent extreme deviations.

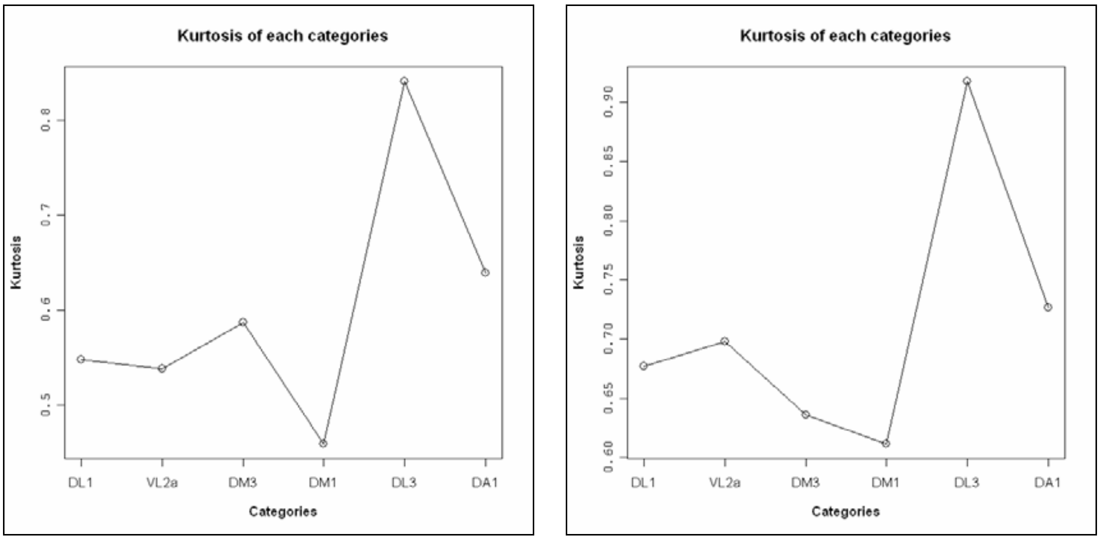


Figure 3-5. Kurtosis of raw (left) and revised (right) data of six categories.

$$ENERGY = \sum_{x=0}^{255} \left[p(x) \right]^2 \qquad (3.2\text{-}9)$$

Energy reaches the maximum value 1 when whole image just has one intensity. If energy is high, it means pixels value just center at some intensity, and the image can easily be compressed. If gray-level pixels spread widely, then energy decreases rapidly.

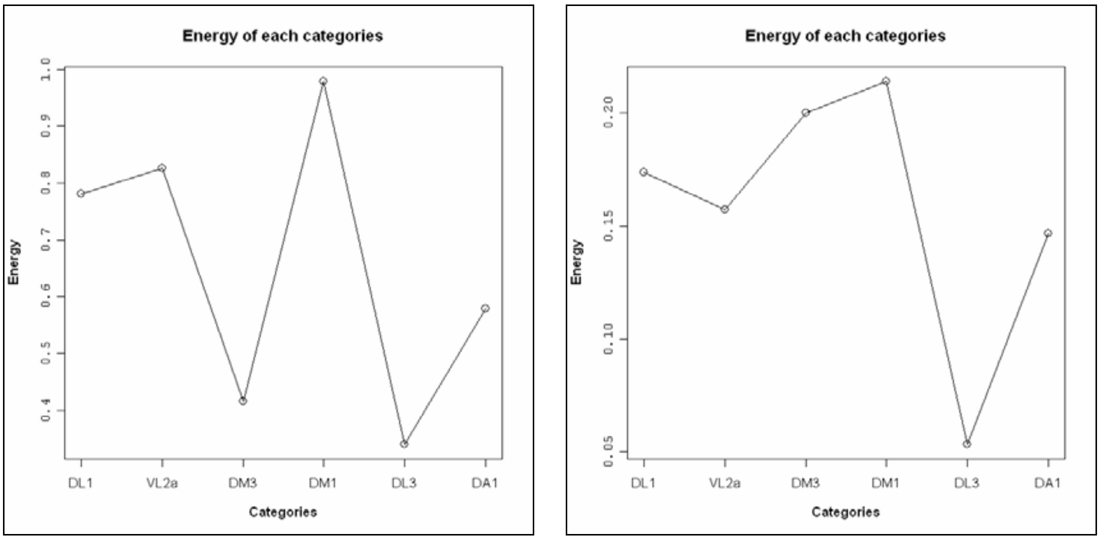


Figure 3-6. Energy of raw (left) and revised (right) data of six categories.

$$ENTROPY = -\sum_{x=0}^{255} P(x)\log_2\left[P(x)\right] \qquad (3.2\text{-}10)$$

Entropy is a measure that tells us how many bits we need to code the image data. If gray-level pixels spread more wildly, entropy gets bigger, contrary to energy.

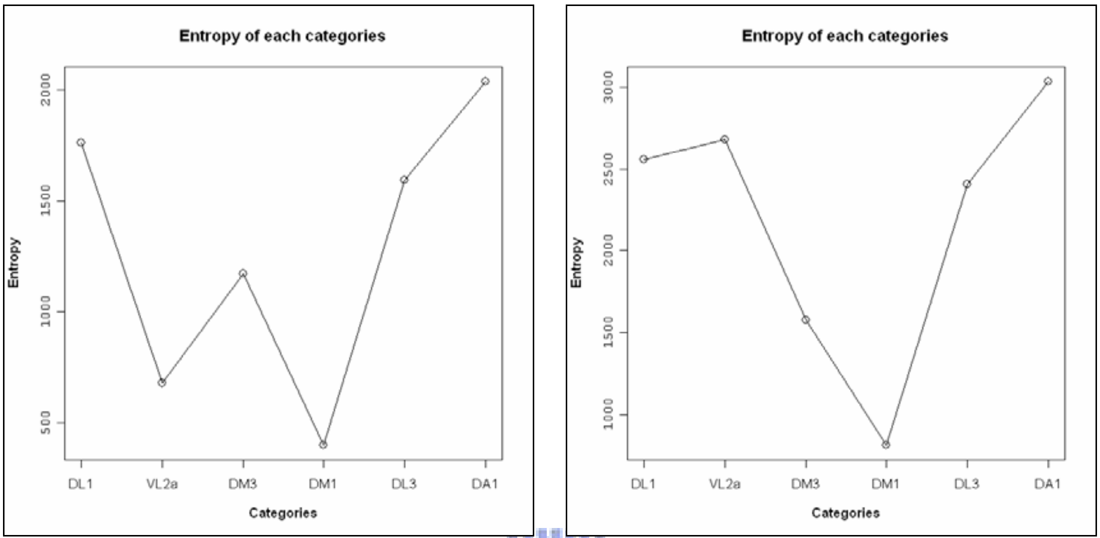


Figure 3-7. Entropy of raw (left) and revised (right) data of six categories.

3.3 skeleton

Skeleton neurons can help us to see the nerve networks clearly. Furthermore, extracting features on skeleton neurons can help us to improve the accuracy. Therefore, we calculate RST-invariant features and end points on skeleton neurons. We tried two kinds of skeleton algorithms. One is an improved fully parallel 3D thinning algorithm [Wang], and another is parameter controlled skeletonization of 3D objects [Gagvani et al., 1997].

3.3.1 An improved fully parallel 3D thinning algorithm

The improved fully parallel 3D thinning algorithm which was proposed by Tao Wang is like a Z-S algorithm in 2 dimensions [Zhang, 1984]. It’s like excoriating target objects layer by layer. Then we can get skeleton neuron finally. The paper defines four classes of about 52 templates to delete every non tail-point simultaneously. The following is a sketch of one of the deleting templates. P is target pixel ,“ • ” is a an object point, and “ ◦ ”is a background point. The unmarked point is arbitrary.

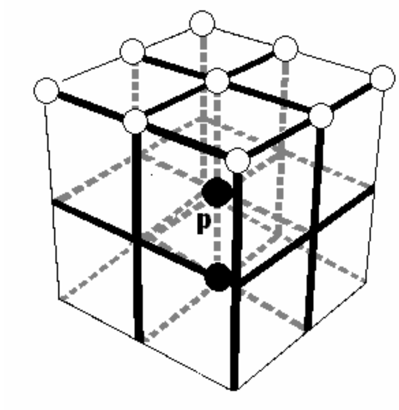
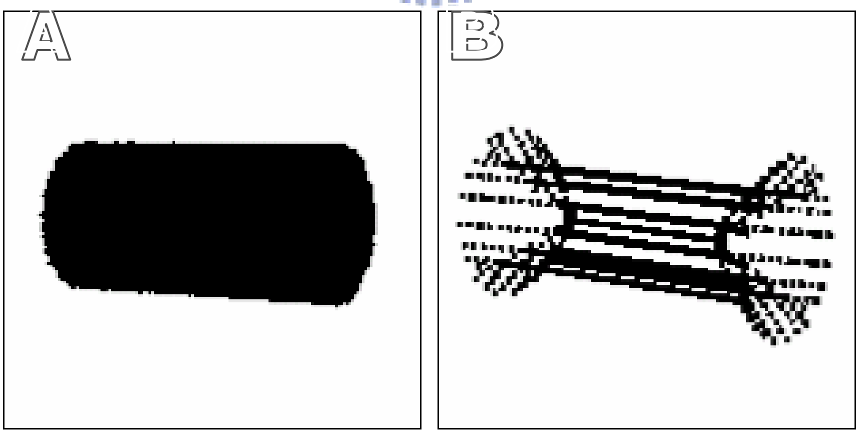


Figure 10. **A sketch of a deleting template.**

3.3.2 Parameter Controlled Skeletonization of 3D Objects

Parameter Controlled Skeletonization is also a method that helps us to create skeleton neurons [Nikhil et al., 1997]. Furthermore, the method has more flexibility. It can control the thickness of neurons by selecting a threshold.



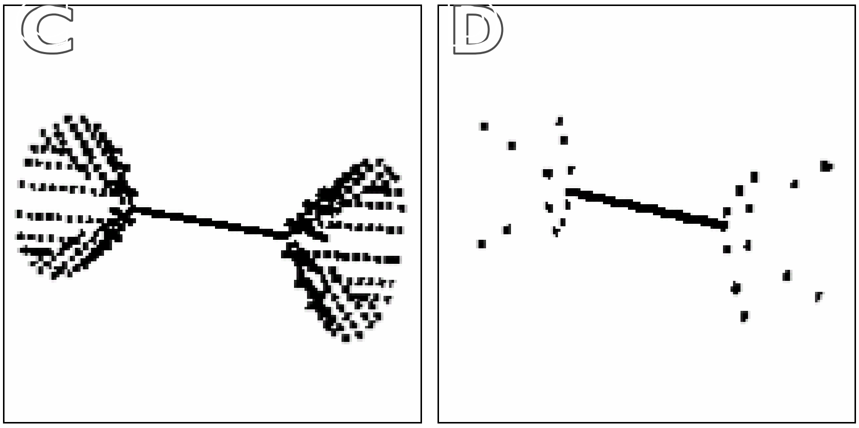


Figure 11. Sketch of Parameter Controlled Skeletonization on a cylinder with threshold = 0, 0.5, 0.6, 0.8.

4. Sliced inverse regression

In ordinary regression, given a response variable *Y* and explanatory variables *X* with *p* dimensions, we usually want to find a function of *X* that can estimate *Y* well. But if *X*'s dimensions are too large, then we may fall into the curse of dimensionality. To avoid this problem, usually have to do variable selection (feature selection) or dimension reduction. A commonly used method of dimension reduction is PCA. PCA is based on choosing eigenvalues to help us compress variable to a few variables which are linear combination of ordinary variables. Furthermore, PCA can also help us to visualize our data by using factor scores.

After feature extraction and before statistical classification, we use slice inverse regression (SIR) to preprocess our feature data which was proposed by Ker-Chau Li (1991). SIR is not focus on estimating the regression function, but effectively reducting dimensions. SIR can help us get information from our high-dimension data by low-dimension projection without losing information. This is different from PCA.

SIR is considered a more general model:

$$Y = f\left(\beta_1 X, \beta_2 X, \ldots, \beta_k X, \varepsilon\right) \qquad (4-1)$$

where, $\beta_1, \ldots, \beta_k$ are unknown projection vectors called effective dimension reduction directions (EDR-directions) [Li, 1991], *k* is unknown and less then p, *Y* is six categories of neurons named by positions in antenna lobe, and *X* is our features extracted from images. In this general model, it needn't assume $\varepsilon \sim N(0, \sigma^2)$ and *f* might be any function of *X*. The relationship of *X* and *Y* just through *p* linear combinations $\beta_1 X, \beta_2 X, \ldots, \beta_k X$. So the goal of SIR is to estimate $\beta_1, \ldots, \beta_k$, and then we can project our feature data into *k* dimensions.

SIR divides *Y* into *k* slices and use those slices to estimate a centered regression curve *E(Z|Y)*, where *Z* is standardized of *X*. Then we transform (4-1) to a new model (4-2), and this new model help us to estimate EDR-directions easily.

$$Y = f\left(\eta_1 Z, \eta_2 Z, \ldots, \eta_k Z, \varepsilon\right) \qquad (4-2)$$

Where $\eta_1, \eta_2, \ldots, \eta_k$ are called standardized EDR-directions. Afterwards, it utilizes the property that *E(Z|Y)* is spanned by standardized EDR-directions to perform weighed principle component analysis on *Cov[E(Z|Y)]*. Because *Cov[E(Z|Y)]* degenerates in any direction that is orthogonal to the standardized EDR-directions, the largest *K* eigenvectors corresponding to largest *k* eigenvalues of *cov[E(Z|Y)]* are the standardized EDR-directions. Finally, we can estimate EDR-directions by transforming standardized EDR-directions to the original scale.

4.1 Algorithm of Sliced inverse regression

1. Arrange our data in the form as shown below, where *Y* in our data are neuron categories, and *X* are features extracted from fly calyx images. In our data, n is 113 and *p* is number of features (histogram 256 + RST-invariant 7 + volume 1 + skeleton neuron (RST-invariant 7 + volume 1) = 272).

Y_1	$X_1 = (X_{11}, X_{12}, \ldots, X_{1p})'$
Y_2	$X_2 = (X_{21}, X_{22}, \ldots, X_{2p})'$
. . .	. . .
Y_n	$X_n = (X_{n1}, X_{n2}, \ldots, X_{np})'$

Table 4.1 Data frame

2. Standardize x.

$$z_i = \hat{\Sigma}_{xx}^{-1/2}(x_i - \bar{x}) \quad (i = 1, \ldots, n) \qquad (4-1)$$

3. Divide *Y* into *k* non-overlapping slices according to categories of neurons and compute the proportion of Y_i in slice s; denote $\hat{p}_s$, that

$$\hat{p}_s = \frac{\#\ of\ \ Y\ \ in\ slice\ s}{n}, \quad s = 1, \ldots, k \qquad (4-2)$$

Let I_s be the indicate function for slice *s*. Then $\hat{p}_s$ can be rewritten as

$$\hat{p}_s = \frac{1}{n}\sum_{i=1}^{n} I_S\left(Y_i\right), \quad s = 1,\ldots,k \tag{4-3}$$

4. Compute the sample mean z_i within each slice.

$$\bar{z}_s = \frac{1}{n_s}\sum_{i=1}^{n} z_i I_s\left(Y_i\right) \quad s = 1,\ldots,k \tag{4-4}$$

5. Form the weighted covariance matrix $\hat{V}$.

$$\hat{V} = \sum_{s=1}^{k} \hat{p}_s \bar{z}_s \bar{z}_s' \tag{4-5}$$

6. Find the eigenvalue $\hat{\lambda}_i$ and eigenvector $\hat{\eta}_i$ of $\hat{V}$. $\hat{\eta}_i$ are the standardized EDR-directions. The maximum numbers of eigenvalues unequal to zero are just dependent on the number of slices-1.

7. Transform $\hat{\eta}_i$ back to the original scale.

$$\hat{\beta}_i = \hat{\Sigma}_{xx}^{1/2} \hat{\eta}_i \tag{4-6}$$

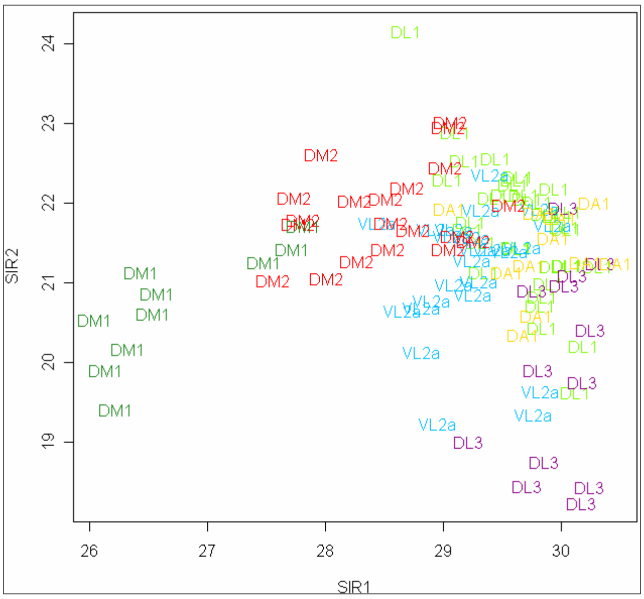
8. Product $\hat{\beta}_1,\ldots,\hat{\beta}_{k-1}$ with *X*.

5. Results

Table 7-1~7-16 in appendix are predicted and classification results using WEKA and R. SVM is one of the classifier functions in WEKA called SMO. J48 is one of the classifier trees in WEKA which is used the C4.5 decision tree algorithm. IBk and OneR are lazy learners and rule learners in WEKA that are also utilized frequently.

First, we find that features extracted from skeleton neurons on revised images can help us to improve accuracy by observing table 7-1~7-16. Second, if we extract RST-invariant on red channel, it can also help us to improve accuracy. By observing table 7-2, it receives 59.29 % accuracy which is bigger than table 7-4 55.75 %. In table 7-2 and table 7-4, we can conclude that if we just want to classifer neuron images and don’t need analysis of 3d structures, then we can only use ordinary image without noise removal. So, when we remove noise, we might also remove useful informations. Nonetheless, our noise removal filter can help us to visualize neuron clearly.

We use $\hat{\beta}_1 X$, $\hat{\beta}_2 X$ and $\hat{\beta}_3 X$ to plot 2D and 3D scatter plots of six categories. We can find that DL1, DA1 and VL2a are so close. Therefore, we combine them into one group. After that, we classify new data into about four groups. By combining groups, we can get higher accuracy. We can see at length in appendix table 7-9~7-16.



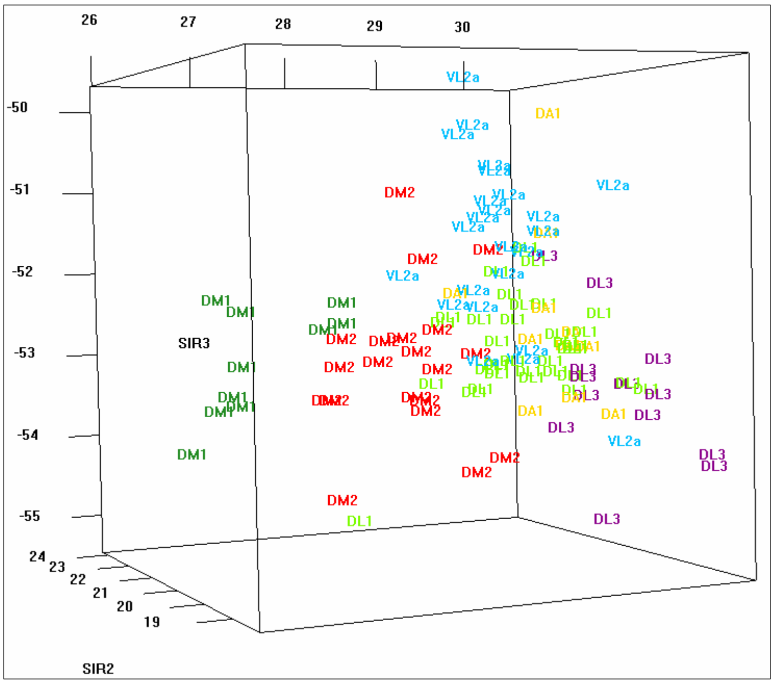


Figure 12. 2D and 3D scatter plots of six categories using $\hat{\beta}_1 X$, $\hat{\beta}_2 X$ and $\hat{\beta}_3 X$.

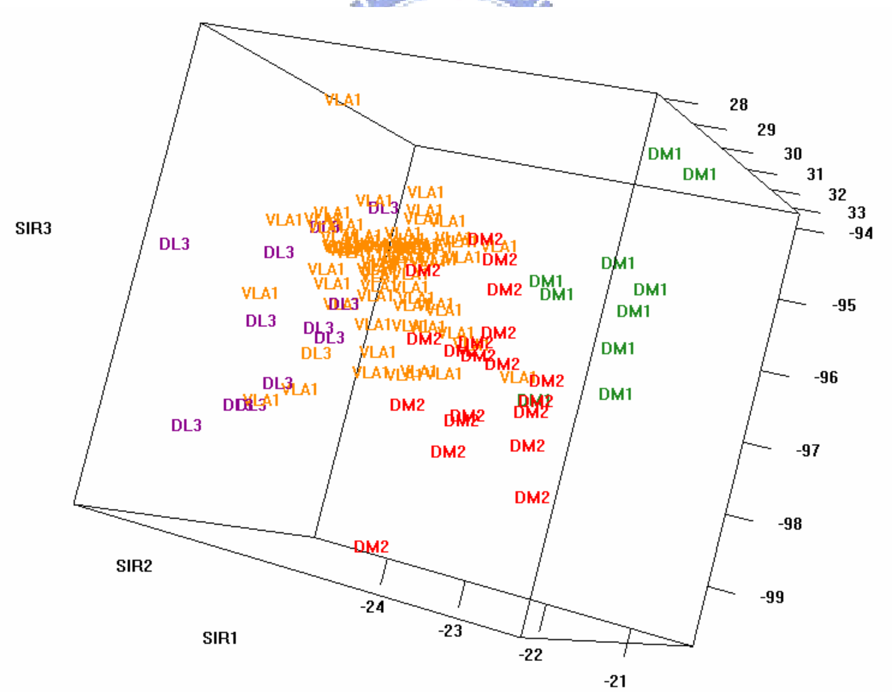


Figure 13. 3D scatter plots of combined four categories using $\hat{\beta}_1 X$, $\hat{\beta}_2 X$ and $\hat{\beta}_3 X$. Here we combine DL1, DA1 and VL2a into a new group and call the new group VLA1.

6. Conclusion

By observing table 7-1 ~ table 7-16, we find the accuracies after our noise removal method are sometimes lower than using raw image directly. But in table 7-6 and table 7-8, if we also consider red channel and skeleton neuron, then our predicted result have a better behaviors about 58.4071 % and 59.292 % respectively. In table 7-9 ~ table 7-16, the highest accuracy of revised data is 70.8 % and it's lower than 77.8761 % of raw data. If we use sir on extracted features can help us to increase the accuracy but not on raw data.

Our volume filter might still not good enough. So, when we remove noise, we might also remove useful informations that make our accuracies are sometimes lower than using raw images. Nonetheless, our noise removal filter can help us to visualize neurons clearly. Besides, we can try more other features and methods to improve accuracy in the future.

Every animal's behavior is controlled by its central nervous system. Neuroscientists believe that much of mankind's abnormal behavior is caused by genetic errors. Modern research in this area has improved to the point where scientists can construct an olfactory nerve network. Furthermore, we can learn more about how nerve networks express which genes. Although current research in olfactory systems have been done only on Drosophila and focus on only parts of the cells, the brain's olfactory nerve network can already be constructed, and this technology can later be utilized to construct taste, visual, auditory, or higher-level images, or even on mammals. Hopefully these research results can be used to cure humanity's sicknesses one day.

7. Appendix

Table 7- 1. Classification results in R/WEKA without using sliced inverse regression and without using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	75 66.3717 %	103 91.1504 %	113 100 %	66 58.4071 %
Raw data (no log)	72 63.7168 %	91 80.531 %	113 100 %	66 58.4071 %
Revised data (take log)	80 70.7965 %	103 91.1504 %	113 100 %	64 56.6372 %
Revised data (no log)	75 66.3717 %	95 84.0708 %	113 100 %	64 56.6372 %
Revised data (take log + skeleton)	86 76.1062 %	104 92.0354 %	113 100 %	66 58.4071 %
Revised data (no log + skeleton)	75 66.3717 %	98 86.7027 %	113 100 %	66 58.4071 %

Table 7-2. Predicted results in R/WEKA without using sliced inverse regression and using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	65 57.5221 %	58 51.3274 %	67 59.292 %	47 41.5929 %
Raw data (no log)	65 57.5221 %	46 40.708 %	63 55.7522 %	47 41.5929 %
Revised data (take log)	37 32.7434 %	59 52.2124 %	20 17.6991 %	37 32.7434 %
Revised data (no log)	51 45.1327 %	56 49.5575 %	55 48.6726 %	39 34.5133 %
Revised data (take log + skeleton)	38 33.6283 %	51 45.1327 %	27 23.8938 %	56 49.5575 %
Revised data (no log + skeleton)	58 51.3274 %	55 48.6726 %	54 47.7876 %	56 49.5575 %

Table 7-3. Classification results in R/WEKA using sliced inverse regression and without using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	109 96.4602 %	108 95.5752 %	113 100 %	75 66.3717 %
Raw data (no log)	60 53.0973 %	89 78.7611 %	113 100 %	55 48.6726 %
Revised data (take log)	113 100 %	113 100 %	113 100 %	113 100 %
Revised data (no log)	61 53.9823 %	100 88.4956 %	113 100 %	57 50.4425 %
Revised data (take log + skeleton)	113 100 %	113 100 %	113 100 %	113 100 %
Revised data (no log + skeleton)	74 65.4867 %	95 84.0708 %	113 100 %	63 55.7522 %

Table 7-4. Predicted results in R/WEKA using sliced inverse regression and using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	47 41.59292 %	43 38.0531 %	54 47.78761 %	38 33.62832 %
Raw data (no log)	47 41.59292 %	53 46.90265 %	53 46.90265 %	44 38.93805 %
Revised data (take log)	33 29.20354 %	22 19.46903 %	35 30.97345 %	17 15.04425 %
Revised data (no log)	59 52.21239 %	43 38.0531 %	47 41.59292 %	43 38.0531 %
Revised data (take log + skeleton)	30 26.54867 %	22 19.46903 %	33 29.20354 %	18 15.92920 %
Revised data (no log + skeleton)	61 53.9823 %	49 43.36283 %	63 55.75221 %	43 38.0531 %

Table 7-5. Classification results added red channel in R/WEKA and without using sliced inverse regression and without using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	76 67.2566 %	106 93.8053 %	113 100%	66 58.4071 %
Raw data (no log)	72 63.7168 %	95 84.0708 %	113 100%	66 58.4071 %
Revised data (take log)	87 76.9912 %	106 93.8053 %	113 100%	64 56.6372 %
Revised data (no log)	99 87.6106 %	110 97.3451 %	113 100 %	72 63.7168 %
Revised data (take log + skeleton)	92 81.4159 %	106 93.8053 %	113 100 %	66 58.4071 %
Revised data (no log + skeleton)	80 70.7965 %	104 92.0354 %	113 100 %	66 58.4071 %

Table 7-6. Predicted results added red channel in R/WEKA and without using sliced inverse regression and using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	66 58.4071%	57 50.4425%	66 58.4071%	47 41.5929%
Raw data (no log)	64 56.6372%	45 39.823 %	62 54.8673%	47 41.5929%
Revised data (take log)	39 34.5133 %	54 47.7876 %	29 25.6637 %	37 32.7434 %
Revised data (no log)	63 55.7522 %	60 53.0973 %	54 47.7876 %	39 34.5133 %
Revised data (take log + skeleton)	42 37.1681 %	48 42.4779 %	32 28.3186 %	56 49.5575 %
Revised data (no log + skeleton)	66 58.4071 %	60 53.0973 %	51 45.1327 %	56 49.5575 %

Table 7-7. Classification results added red channel in R/WEKA and using sliced inverse regression and without using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	109 96.4602 %	111 98.2301 %	113 100 %	75 66.3717 %
Raw data (no log)	68 60.177 %	98 86.7257 %	113 100 %	52 46.0177 %
Revised data (take log)	113 100 %	113 100 %	113 100 %	89 78.7611 %
Revised data (no log)	71 62.8319 %	101 89.3805 %	113 100 %	58 51.3274 %
Revised data (take log + skeleton)	113 100 %	113 100 %	113 100 %	106 93.8053 %
Revised data (no log + skeleton)	84 74.3363 %	106 93.8053 %	113 100 %	69 61.0619 %

Table 7-8. Predicted results added red channel in R/WEKA and using sliced inverse regression and using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	54 47.78761 %	41 36.28319 %	49 43.36283 %	33 29.20354 %
Raw data (no log)	54 47.78761 %	51 45.13274 %	51 45.13274 %	51 29.20354 %
Revised data (take log)	30 26.54867 %	27 23.89381 %	27 23.89381 %	19 16.81416 %
Revised data (no log)	56 49.55752 %	48 42.47788 %	47 47.78761 %	33 29.20354 %
Revised data (take log + skeleton)	34 30.08850 %	30 26.54867 %	33 29.20354 %	26 23.00885 %
Revised data (no log + skeleton)	67 59.29204 %	59 52.21239 %	59 52.21239 %	50 44.24779 %

Table 7-9. Classification results on combined groups in R/WEKA without using sliced inverse regression and without using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	89 78.7611 %	108 95.5752 %	113 100 %	90 79.646 %
Raw data (no log)	76 67.2566 %	99 87.6106 %	113 100 %	90 79.646 %
Revised data (take log)	94 83.1858 %	111 98.2301 %	113 100 %	90 79.646 %
Revised data (no log)	88 77.8761 %	98 86.7257 %	113 100 %	90 79.646 %
Revised data (take log + skeleton)	97 85.8407 %	112 99.115 %	113 100 %	90 79.646 %
Revised data (no log + skeleton)	92 81.4159 %	107 94.6903 %	113 100 %	90 79.646 %

Table 7-10. Predicted results on combined groups in R/WEKA without using sliced inverse regression and using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	87 76.9912 %	81 71.6814 %	88 77.8761 %	86 76.1062 %
Raw data (no log)	76 67.2566 %	73 64.6018 %	81 71.6814 %	86 76.1062 %
Revised data (take log)	69 61.0619 %	71 62.8319 %	49 43.3628 %	79 69.9115 %
Revised data (no log)	76 67.2566 %	79 69.9115 %	77 68.1416 %	80 70.7965 %
Revised data (take log + skeleton)	73 64.6018 %	69 61.0619 %	57 50.4425 %	79 69.9115 %
Revised data (no log + skeleton)	80 70.7965 %	80 70.7965 %	76 67.2566 %	80 70.7965 %

Table 7-11. Classification results on combined groups in R/WEKA using sliced inverse regression and without using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	111 98.2301 %	112 99.115 %	113 100 %	104 92.0354 %
Raw data (no log)	70 61.9469 %	91 80.531 %	113 100 %	80 70.7965 %
Revised data (take log)	113 100 %	113 100 %	113 100 %	113 100 %
Revised data (no log)	80 70.7965 %	93 82.3009 %	113 100 %	84 74.3363 %
Revised data (take log + skeleton)	113 100 %	113 100 %	113 100 %	113 100 %
Revised data (no log + skeleton)	86 76.1062 %	98 86.7257 %	113 100 %	93 82.3009 %

Table 7-12. Predicted results on combined groups in R/WEKA using sliced inverse regression and using leave-one -out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	65 57.52212 %	66 58.40708 %	66 58.40708 %	57 50.44248 %
Raw data (no log)	70 61.9469 %	66 58.40708 %	73 64.60177 %	52 46.0177 %
Revised data (take log)	33 29.20354 %	45 39.82301 %	39 34.51327 %	28 24.77876 %
Revised data (no log)	75 66.37168 %	83 73.45133 %	72 63.71681 %	78 69.02655 %
Revised data (take log + skeleton)	32 28.31858 %	44 38.93805 %	38 33.62832 %	22 19.46903 %
Revised data (no log + skeleton)	77 68.14159 %	77 68.14159 %	74 65.48673 %	73 64.60177 %

Table 7-13. Classification results on combined groups added red channel in R/WEKA and without using sliced inverse regression and without using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	89 78.7611 %	110 97.3451 %	113 100 %	90 79.646 %
Raw data (no log)	76 67.2566 %	99 87.6106 %	113 100 %	90 79.646 %
Revised data (take log)	93 82.3009 %	110 97.3451 %	113 100 %	90 79.646 %
Revised data (no log)	92 81.4159 %	103 91.1504 %	113 100 %	90 79.646 %
Revised data (take log + skeleton)	97 85.8407 %	112 99.115 %	113 100 %	90 79.646 %
Revised data (no log + skeleton)	93 82.3009 %	110 97.3451 %	113 100 %	90 79.646 %

Table 7-14. Classification results on combined groups added red channel in R/WEKA and without using sliced inverse regression and using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	87 76.9912 %	74 65.4867 %	87 76.9912 %	86 76.1062 %
Raw data (no log)	85 75.2212 %	80 70.7965 %	84 71.6814 %	72 63.7168 %
Revised data (take log)	70 61.9469 %	77 68.1416 %	55 48.6726 %	79 69.9115 %
Revised data (no log)	78 69.0265 %	83 73.4513 %	77 68.1416 %	80 70.7965 %
Revised data (take log + skeleton)	70 61.9469 %	70 61.9469 %	59 52.2124 %	79 69.9115 %
Revised data (no log + skeleton)	79 69.9115 %	70 61.9469 %	76 67.2566 %	80 70.7965 %

Table 7-15. Classification results on combined groups added red channel in R/WEKA and using sliced inverse regression and without using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	110 97.3451 %	111 98.2301 %	113 100 %	94 83.1858 %
Raw data (no log)	70 61.9469 %	89 78.7611 %	113 100 %	79 69.9115 %
Revised data (take log)	113 100 %	111 98.2301 %	113 100 %	107 94.6903 %
Revised data (no log)	77 68.1416 %	101 89.3805 %	113 100 %	80 70.7965 %
Revised data (take log + skeleton)	111 98.2301 %	112 99.115 %	113 100 %	106 93.8053 %
Revised data (no log + skeleton)	80 70.7965 %	107 94.6903 %	113 100 %	94 83.1858 %

Table 7-16. Classification results on combined groups added red channel in R/WEKA and using sliced inverse regression and using leave-one-out cross-validation to evaluate correctness.

	SVM	J48	IBk	OneR
Raw data (take log)	69 61.06195 %	65 57.52212 %	73 64.60177 %	61 53.9823 %
Raw data (no log)	64 56.63717 %	59 52.21239 %	64 56.63717 %	52 46.0177 %
Revised data (take log)	37 32.74336 %	37 32.74336 %	42 37.16814 %	25 25.66372 %
Revised data (no log)	72 63.71681 %	73 64.60177 %	72 63.71681 %	68 60.17699 %
Revised data (take log + skeleton)	49 43.36283 %	47 41.59292 %	50 44.24779 %	37 32.74336 %
Revised data (no log + skeleton)	75 66.37168 %	80 70.79646 %	77 68.14159 %	78 69.02655 %

8. Reference

Bozza, Thomas., Feinstein, Paul., Zheng, Chen. and Mombaerts, Peter. “Odorant Receptor Expression Defines Functional Units in the Mouse Olfactory System.” The Journal of Neuroscience, April 15, 22(8):3033–3043, 2002.

Buck, Linda. and Axel, Richard. “A novel multigene family may encide odorant receptors: A molecular basis for odor recognition.” Cell. Vol. 65, 175-187, 1991.

Couto, Africa., Alenius, Mattias. and J.Dickson, Barry. “Molecular, Anatomical, and Functional Organization of the Drosophila Olfactory System.” Current Biology, Vol.15, 1535-1547, 2005.

Firestein, Stuart, “How the olfactory system makes sense of scents.” Nature 413(6852) 211-218, 2001.

Gagvani, Nikhil. and Silver, Deborah . “Parameter controlled skeletonization of three dimensional objects.” Technical report CAIP-TR-216, 3 June, 1997.

Goldman, A.L., Van der Goes van Naters, W., Lessing, D., Warr, C.G., and Carlson, J.R. “Coexpression of two functional odor receptors in one neuron.” Neuron, Vol. 45, 661–666, March 3, 2005.

Gonzalez and Woods. “Digital Image Processing.“ Addison Wesley.

Hallem, Elissa A., Ho, Michael G. and Carlson, John R. “The Molecular Basis of Odor Coding in the Drosophila Antenna.” Cell, Vol. 117, 965–979, June 25, 2004,

Hallem, Elissa A. and Carlson, John R. “The odor coding system of drosophila.” TRENDS in Genetics Vol.20 No.9 2004

Huang, T.S., Yang, G.J. and Tang, G.Y. “A fast two-dimensional median filtering algorithm.” IEEE Transactions on Acoustics, Speech and Signal Processing, vol.27, 13-18, 1979.

Jefferis, Gregory S.X.E., Marin, Elizabeth C., Stocker, Reinhard F. and Luo, Liqun. “Target neuron prespecification in the olfactory map of drosophila.” Nature, Vol414, 2001.

Kreher, Scott A., Kwon, Jae Young. and Carlson, John R. “The molecular basis of odor coding in the drosophila larva.” Neuron, Vol, 46, 445-456, 2005.

Li, Ker-Chau. “Sliced Inverse Regression for Dimension Reduction.” Journal of the American Statistical Association, Vol. 86, No. 414, Jun., pp. 316 -327, 1991.

Liaw, Joen-Woei., Shih, Tian-Yuan., Chang, Kuen-Tzung. “A Comparison of Thinning Algorithms.”

Lin, Hui-Hao., Lin, Chih-Yung. and Chiang, Ann-Shyn. “Internal representations of smell in the drosophila brain.” Journal of Biomedical Science, 14:453-459, 2007.

Marin, Elizabeth C., Jefferis, Gregory S.X.E., Komiyama, Takaki., Zhu, Haitao. and Luo, Liqun. “Representation of the glomerular olfactory map in the drosophila brain.” Cell, vol. 109, 243-255, 2002.

Pye, C. Jeremy. and Bangham, J.A. “A fast algorithm for morphological erosion and dilation.”

Stocker, Reinhard F. “The organization of the chemosensory system in Drosophila melanogast- er: a review.” Cell Tissue Res, 275:3-26, 1994.

Umbaugh, Scott E., Wei, Yansheng. and Zuke, Mark. “Feature Extraction in Image Analysis.” IEEE Engineering in medicine and biology, July/August, 1997.

Vosshall, Leslie B., Amrein, Hubert., Morozov, Pavel S., Rzhetsky, Andrey. and Axel, Richard. “A spatial map of olfactory receptor expression in the drosophila antenna.” Cell, Vol. 96, 725-736, 1999.

Wang, Tao. and Basu, Anup. “An improved fully parallel 3D thinning algorithm.”

Wong, Allan M. and Axel, Richard. “Spatial respresentation of the glomerular map in the Drosophila protocerebrum.” Cell 109(2) 229-241, 2002.

Zhang, S. and Fu, K.S. “A thinning Algorithm for Discrete Binary Images.” Proceedings of the International Conference on Computers and Application. Beijing, China. 98, 1984.