# Reverse Engineering the Brain

# Project Report

**Team – Gold**

**Names:**

Tiffany

Surabhi

Sharath (ID: vp4pa)

Digital Image Processing - Experiment 1

Objective: Understand the internal structure of *Drosophila’s* brain by processing and analyzing the TIFF stacks provided.

## Methodology:

We have built our model on “SampleStack.tif “ volume stack. Each frame in the stack is read and segmented as red, blue and green images. A separate stack is created for each color. **Red stack** is used to count the Neuroblasts, **Blue stack** is to count glia cells and the **Green stack** is to estimate the surface area of the membrane. We tried to automate the process with following functionality.

1. **Main Function: (exp1\_main)**

Read each frame from the stack and segment it as red, blue and green using exp1\_cluster( ) function.

1. **Color Segmentation using k-means clustering: (exp1\_cluster)**

We have used k-means clustering technique after converting the image into L\*a\*b color space to segment red, green and blue components. All the color related information is in the chromaticity layers (a,b) of L\*a\*b space. The distance between colors is measured using Euclidean distance metric.

K-means clustering calculates distance between each object and partitions them such that objects in each cluster are close and objects in different clusters are as far as possible.

Though the clustering technique has effectively segmented colors, the challenge was to identify the clusters with colors. Being an unsupervised technique, k-means clustering arbitrarily assigns a cluster number to colors. We used mean value of cluster centers to determine the color of a cluster.

For every image, **3 clusters** are automatically obtained. From the stack of 98 images, we have built 3 different stacks one for each color red, green and blue.

1. **Counting Neuroblasts: (exp1\_neuro)**

For Neuroblasts, read each image from Red stack

1. Extract the red channel
2. Preprocess it using appropriate morphological operations
   1. Median Filtering
   2. Binary conversion
   3. Filling holes
   4. Open operation using disk with radius 3
3. After preprocessing of all images, construct a N-dimensional binary image array
4. Counting Neuroblasts in N-D image array using bwlabeln( )
5. Write the binary images to a new stack (red\_bw\_stack.tif)
6. **Glia nuclei: (exp1\_glia)**

For Glia cells, read each image from Blue stack

1. Extract the blue channel
2. Preprocess it using appropriate morphological operations
   1. Median Filtering
   2. Binary conversion – 3 \* Gray threshold level is used to remove the excess background noise
   3. Filling holes
   4. Open operation using disk with radius 3
3. Construct a N-dimensional binary images
4. Counting Glia cells in N-D image using bwlabeln( )
5. Write processed binary images to a new stack (blue\_bw\_stack.tif)
6. **Surface area of Glia membrane: (exp1\_glia\_membrane)**

For Glia cells, read each image from Blue stack

1. Extract the green channel
2. Preprocessing
   1. Preprocess it using appropriate morphological operations
   2. Open it in ImageJ
      1. A = Measure total count of pixels that constitutes the membrane
      2. T = Measure the total area of image in pixel counts
         1. Height \* Width from the size function
      3. Calculate the area in % = 100 \* (A/T)
   3. From focal length of microscope (f) and distance of measurement (Z), obtain the scale of coordinates
   4. Calculate the approximate area of membrane by converting the % area into the required scale
3. **Visualizations:**

Visualized the processed stacks in ImageJ. Added videos of segmented stacks and b/w stacks under [Results](#_Results:)

1. **Automation:**

All the functionalities are automated and the required results can be generated for any given stack with a single call to above mentioned functions.

Sources:

*1. Color segmentation using k-means clustering* <http://www.mathworks.com/help/images/examples/color-based-segmentation-using-k-means-clustering.html#zmw57dd0e2744>

*2. N-d label counting*

[*http://www.mathworks.com/help/images/ref/bwlabeln.html*](http://www.mathworks.com/help/images/ref/bwlabeln.html)

*3. CONSTRUCTING SOFTWARE FOR ANALYSIS OF NEURON, GLIAL AND ENDOTHELIAL CELL NUMBERS AND DENSITY IN HISTOLOGICAL NISSL-STAINED RODENT BRAIN TISSUE - AgataKOŁODZIEJCZYK1, MagdalenaŁADNIAK2, AdamPIO ́RKOWSKI2*

*JOURNAL OF MEDICAL INFORMATICS & TECHNOLOGIES Vol. 23/2014, ISSN 1642-6037*

4. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm>

5. <http://www.mathworks.com/matlabcentral/answers/110269-color-segmentation-of-skin-markers-hsv-or-lab-how-to-choose-thresholds>

## Results:

No. of Neuroblasts: 111

No. of Glia nuclei: 97

Surface Area: To be calculated

### Visuals

Videos to be added

### Output stacks :

3 stacks (red, blue and green) after segmentation – Attached

3 binary image stacks after processing of red, blue and green images – Attached

## Code:

#### Main function (exp1\_main):

%Read stack and process each image

delete('red\_stack.tif', 'green\_stack.tif', 'blue\_stack.tif');

stack = 'SampleStack.tif';

info = imfinfo(stack);

for k = 1:numel(info)

I = imread(stack, k);

%Call clustering function for each frame

segmented\_img = exp1\_cluster(I);

imwrite(segmented\_img{3}, 'red\_stack.tif', 'writemode', 'append');

imwrite(segmented\_img{2}, 'blue\_stack.tif', 'writemode', 'append');

imwrite(segmented\_img{1}, 'green\_stack.tif','writemode', 'append');

end

%Clustering to determine color segments

#### K-means clustering function (exp1\_cluster):

function segmented\_img = exp1\_cluster(I)

%I = imread('SampleStack.tif', 46);

%imshow(I), title('Frame 36, Original');

%Covert to L,a,b colorspace from RGB

% L = Lumonocity(Birghtness), a,b = Chromaticity (colors a = R-G, b = B-Y)

form = makecform('srgb2lab');

Ilab = applycform(I, form);

%Classify the colors using k-means

%Get only a, b space of the image

Iab = double(Ilab(:,:,2:3));

%figure, imshow(Iab), title('ab space');

rows = size(Iab,1);

cols = size(Iab,2);

Iab = reshape(Iab, rows\*cols, 2);

% find objects which has similar pixel patterns and put them into

% one cluster

%k-means clustering claculates distance between each object and partitions

%them such that objects in each cluster are close and as far as possible

%from objects in other clusters

nclusters = 3;

[clIndex, clCenter] = kmeans(Iab,nclusters,'Replicates',3);

%Reshape it back

clIndex = reshape(clIndex,rows,cols);

%figure, imshow(clIndex,[]), title('Image after clustering');

%Create the color images from segmented ones

segmented\_img = cell(1,nclusters);

% clust\_img = cell(1, nclusters);

rgb\_label = repmat(clIndex,[1 1 nclusters]); %Create 3 clusters

cluster\_means = mean(clCenter,2);

[value, id] = sort(cluster\_means);

IL = Ilab(:,:,1); %Get the L component of Ilab

% In clustering we get only clusters belong to particular color

% We need a mechanism to identify the clusters with corresponding %colors Automatically

for k = 1:nclusters

col\_idx = find(clIndex == id(k));

L\_col = IL(col\_idx);

is\_light\_col = im2bw(L\_col, graythresh(L\_col));

col\_labels = repmat(uint8(0),[rows cols]);

col\_labels(col\_idx(is\_light\_col==true)) = 1;

col\_labels = repmat(col\_labels,[1 1 3]);

col\_image = I;

col\_image(col\_labels ~= 1) = 0;

rgb = find(id==k);

segmented\_img{k} = col\_image;

clust\_img{id(k)} = col\_image;

end

% figure, imshow(segmented\_img{1}), title('Green');

% figure, imshow(segmented\_img{2}), title('Blue');

% figure, imshow(segmented\_img{3}), title('Red');

% figure, imshow(clust\_img{1}), title('1');

% figure, imshow(clust\_img{2}), title('2');

% figure, imshow(clust\_img{3}), title('3');

#### Function to count Neuroblasts (exp1\_neuro):

delete('red\_bw\_stack.tif');

stack = 'red\_stack.tif';

info = imfinfo(stack);

n = numel(info);

red\_processed = cell(1,n);

IC = imread(stack, 1);

[rows, cols, col] = size(IC);

d = zeros(rows, cols, 98);

for k = 1:n

IC = imread(stack, k);

I = IC(:,:,1);

%figure, imshow(I), title('gray image');

I = medfilt2(I);

T = graythresh(I);

Ibw = im2bw(I,T);

%Ibw = imerode(Ibw, strel('disk', 2));

Ibw = imfill(Ibw, 'holes');

Ibw = imopen(Ibw, strel('disk', 3));

%figure, imshow(Ibw), title('Processed');

[L,ccnum] = bwlabel(Ibw);

if ccnum < 10

d(:,:,k) = Ibw;

imwrite(Ibw, 'red\_bw\_stack.tif', 'writemode', 'append');

end

%end

end

[L, n] = bwlabeln(d); %count the neuroblasts in the 3D image

disp('No. of neuroblasts:');

disp(n);

#### Function to count Glia Cells (exp1\_glia):

delete('blue\_bw\_stack.tif');

stack = 'blue\_stack.tif';

info = imfinfo(stack);

n = numel(info);

I = imread(stack, 1);

[rows, cols, col] = size(IC);

d = zeros(rows, cols, 98);

for k = 1:n

IC = imread(stack, k);

I = IC(:,:,1);

%I = rgb2gray(IC);

%figure, imshow(I), title('gray image');

I = medfilt2(I);

T = graythresh(I);

Ibw = im2bw(I,3\*T);

%Ibw = imerode(Ibw, strel('disk', 2));

Ibw = imfill(Ibw, 'holes');

Ibw = imopen(Ibw, strel('disk', 3));

%figure, imshow(Ibw), title('Processed');

[L,ccnum] = bwlabel(Ibw);

%if ccnum < 10

d(:,:,k) = Ibw;

imwrite(Ibw, 'blue\_bw\_stack.tif', 'writemode', 'append');

%end

end

[L, n] = bwlabeln(d); %count the neuroblasts in the 3D image

disp('No. of Glia nuclei:');

disp(n);

#### Function to calculate surface area of Glia Membrane (exp1\_glia\_membrane):