California State University Channel Islands

Fashion Data Image Classifier

12/20/2019

MATH-546 SEC 001 - PATTERN RECOGNITION ASSIGNMENT

Vasanthi Lingamdinne
MSCS GRADUATE -2019

Contents

1. Abstract

The goal of this project is to classify **Fashion clothing images**, I have chosen 6 different classifier categories such as **T-shirt, Trousers, Hoodie, Sneakers, Sandals and Backpack**. For this project, I started with segmenting each category from a set of training images (Proper segmentation of feature is done by filling the holes, removing the noise). With the help of regionprops, different properties (eg: **area, eccentricity, extent and circularity**) of the images are collected and a decision is made by partitioning the data based on the properties of training images. A classifier is created with the decision tree and tested the classifier with set of testing images set and a confusion matrix is evaluated. The classifier created will identify the proper category for the fashion clothing images. The objective of this classifier project is for the new business in fashion industry, to better to give clothing or fashion recommendation to customers by analyzing their search data and their interests. A fashion clothing classifier here will evaluate the customers search data and give buying recommendation based on their interest. I used MATLAB to segment, classify and evaluate the images and JMP for decision tree with the help of partition algorithm.

2. Introduction & Background:

2.1 Image Classification:

Image classification refers to a process that can classify an image according to its visual content. For example, an image classification algorithm may be designed to tell what category the image belongs to. Robust image classification is still a challenge in computer vision applications.

Image classification is very important part in large number of areas in multimedia applications and other fields. Content Based Image Retrieval plays an important role in the field of medical imaging, business, marketing, social network analysis, computer science, mathematical analysis etc. By Image similarity measurement one can retrieve a range of similar images which are relevant with the query image. When the image database size is high, it is essential to have an efficient image retrieval technique.

For image similarity measurement features are extracted first. Here the features that are extracted from the images are color feature, shape and texture. Shape feature comes under middle level feature. Color and texture come under low level feature.

For Shape features, regionprops are used for extracting the property of image region as the shape feature. Many region properties are there such as Area, Centroid, Circularity, Convexlength, Convexareas, Eccentricity, Extent etc. For this fashion clothing image classification, regionprops used are Area, Circularity, Extent and Eccentricity.

2.2 Regionprops Properties:

2.2.1 Area:

Actual number of pixels in the region, returned as a scalar. (This value might differ slightly from the value returned by bwarea, which weights different patterns of pixels differently.)

To find the equivalent to the area of a 3-D volume, use the 'Volume' property of regionprops3.

2.2.2 Circularity:

Circularity that specifies the roundness of objects, returned as a struct with field Circularity. The struct contains the circularity value for each object in the input image.

The circularity value is computed as (4*Area*pi)/(Perimeter2).

For a perfect circle -> the circularity value = 1.

2.2.3 Extent:

Ratio of pixels in the region to pixels in the total bounding box, returned as a scalar. Computed as the Area divided by the area of the bounding box.

2.2.4 Eccentricity:

Eccentricity of the ellipse that has the same second-moments as the region, returned as a scalar. The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length.

The value is between 0 and 1 (0 and 1 are degenerate cases).

An ellipse whose eccentricity is 0 is a circle,

An ellipse whose eccentricity is 1 is a line segment.

2.3 Image Datasets

2.3.1 Training dataset used:

Dataset comprised of 97 images of 6 types of clothing, such as sneakers, t-shirts, trousers, sandals and hoodies.

2.3.2 Test dataset:

Dataset comprised of 30 images of 6 types of clothing, such as sneakers, t-shirts, trousers, sandals and hoodies.

3. Process Flow

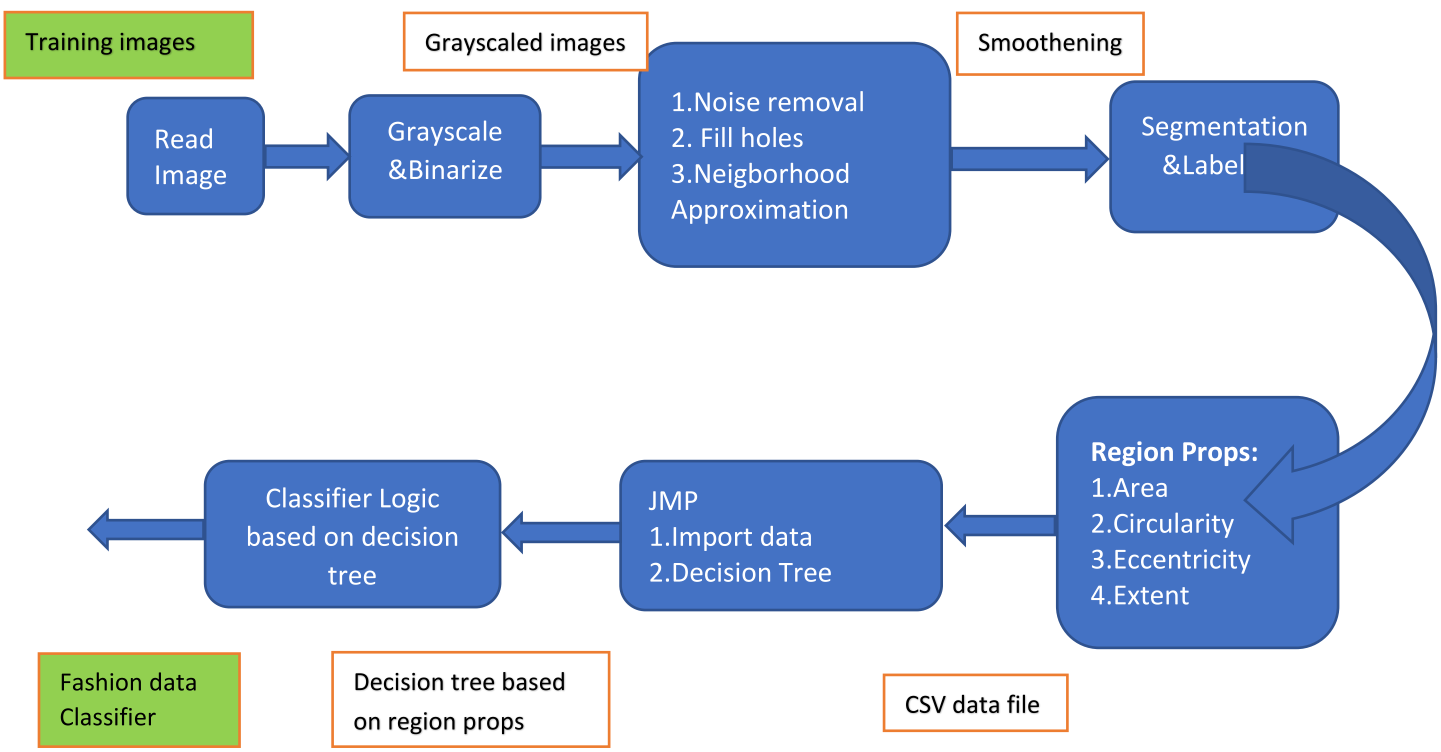

Figure 3.1 Process flow/Algorithm

3.1 Steps/Algorithm

3.1.1 Step 1: Read the image file.

Imread(), all the images from training dataset are read into MATLAB**.**

3.1.2 Step 2: Convert the RGB images into Grayscale images.

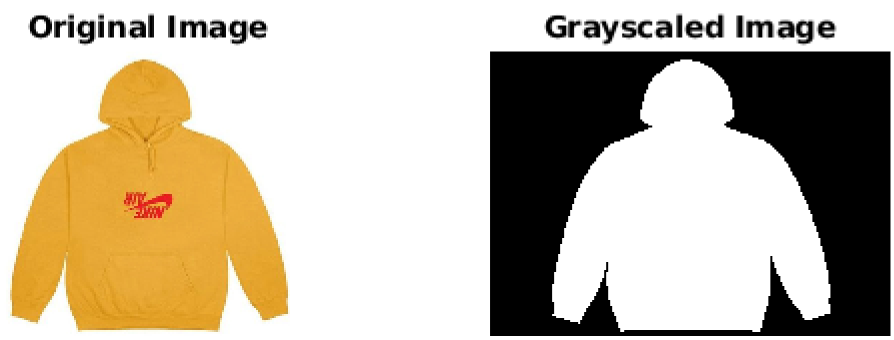

Figure 3.1.2.1 Image Grayscaling

3.1.3 Step 3: Smoothening:

- Training images consists various types of noises which will impact the segmentation and labelling of objects.
- Also specific image type for example, sneaker image has holes for laces hence labelling is difficult in that case, so I have used imfill() to fill the holes hence the segmentation of sneakers is done correctly.
- Neighborhood approximation is done by strel(), this converts the image to flat structuring image.

1.Noise removal

2.Fill holes

3. Neighborhood approximation

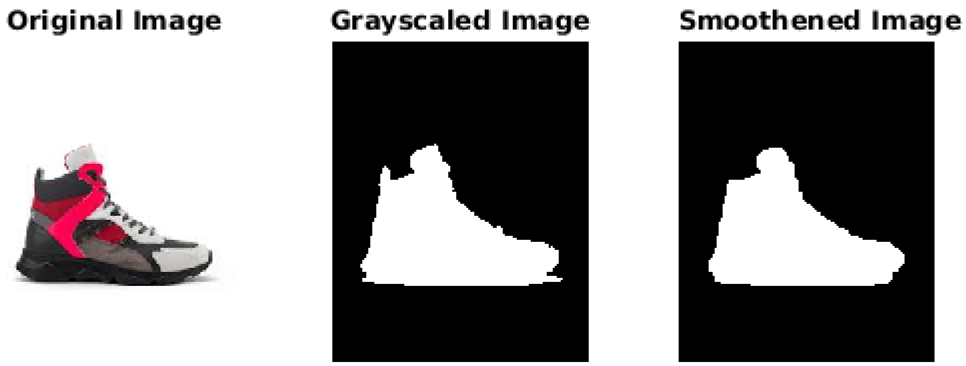

Figure 3.1.3.1 Image Smoothening

3.1.4 Step 4: Image segmentation and labeling

The main goal of segmentation is to simplify the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images. Image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image by regionprops.

- bwlabel();
- L1 = label2rgb(L, @jet, [.5 .5 .5]);

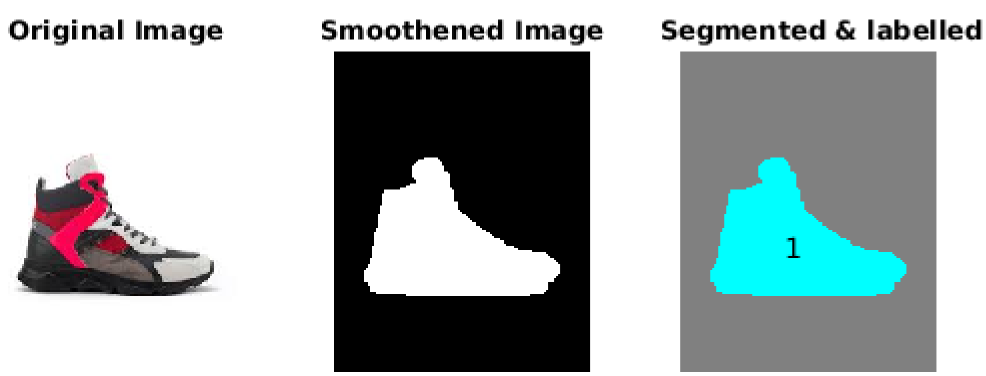

Figure 3.1.4.1 Image Segmentation & Labelling

3.1.5 Step 5: Calculate the region property using regionprops function

Properties chosen for this classifier are Area, Circularity, Extent and Eccentricity.

stats = regionprops(L,properties)

All the regionprops stats for each image from training data set are loaded into a csv file with proper labelling of classifier names as shown in the Figure 3.1.51.

Category

	A	B	C	D	E	F
1	Category	Area	Eccentricity	Circularity	Extent	
2	backpack	30096	0.749464012	0.677506558	0.81241733	
3	backpack	14	0.954684476	0.688427614	0.583333333	
4	backpack	252591	0.779144416	0.66332513	0.732742516	
5	backpack	17614	0.735703967	0.248265256	0.833996212	
6	backpack	33277	0.717735091	0.816402137	0.864068342	
7	backpack	207864	0.810974526	0.671474528	0.852002902	
8	backpack	50625	0	0.825784772	1	
9	backpack	30313	0.623878329	0.644341356	0.789359929	
10	backpack	311711	0.768223367	0.519316591	0.795002652	
11	backpack	323279	0.651658607	0.774026371	0.804809253	
12	backpack	1	0	0.77	1	
13	backpack	23120	0.732451616	0.655809332	0.734784681	
14	backpack	415063	0.782544697	0.845000702	0.872189978	
15	backpack	22209	0.754708922	0.690918978	0.860613811	
16	backpack	281066	0.753708438	0.778535722	0.789014777	
17	backpack	204748	0.734670003	0.794349162	0.843250634	
18	backpack	20390	0.714463323	0.882586758	0.870474727	
19	backpack	653859	0.723205128	0.736516656	0.878134569	
20	Hoodie	620326	0.132084263	0.463160377	0.653306204	
21	Hoodie	104750	0.606124362	0.087188078	0.616597208	
22	Hoodie	49012	0.609250233	0.53675491	0.678686164	
23	Hoodie	168555	0.418226493	0.327126095	0.679027515	
24	Hoodie	24488	0.689682951	0.502145262	0.692867046	
25	Hoodie	477777	0.426290851	0.369032419	0.613190257	
26	Hoodie	2	0.866025404	6.542258754	1	

datafile (2)

Figure 3.1.5.1 Regionprops data

3.2 Step 5: JMP decision tree:

3.2.1 JMP Software: JMP is a suite of computer programs for statistical analysis developed by the JMP business unit of SAS Institute. JMP software is partly focused on exploratory data analysis and visualization. It is designed for users to investigate data to learn something unexpected, as opposed to confirming a hypothesis. JMP links statistical data to graphics representing them, so users can drill down or up to explore the data and various visual representations of it. It's primary applications are for designed experiments and analyzing statistical data from industrial processes.

3.2.2 Partition Models in JMP:

The Partition platform recursively partitions data according to a relationship between the predictors and response values, creating a decision tree. The partition algorithm searches all possible splits of predictors to best predict the response. These splits (or partitions) of the data are done recursively to form a tree of decision rules. The splits continue until the desired fit is reached. The partition algorithm chooses optimum splits from a large number of possible splits, making it a powerful modeling, and data discovery tool.

- Recursively partitions data between the predictors and responses.
- **Partition Algorithm:**
  - Searches all possible splits of predictors to predict the responses.
  - Create a decision tree with decision rules

3.2.3 Building the Predictive model in JMP:

The csv data file with region props data of training dataset is fed into JMP. Performed below steps on JMP for the predictive analysis to achieve the decision tree.

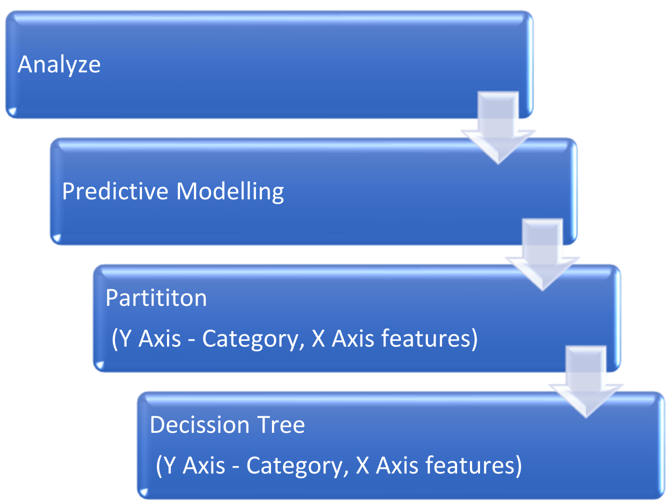

Figure 3.2.3 Partition Model

3.2.3.1 Analyze

Features data extracted from training images is fed into JMP software for partition model.

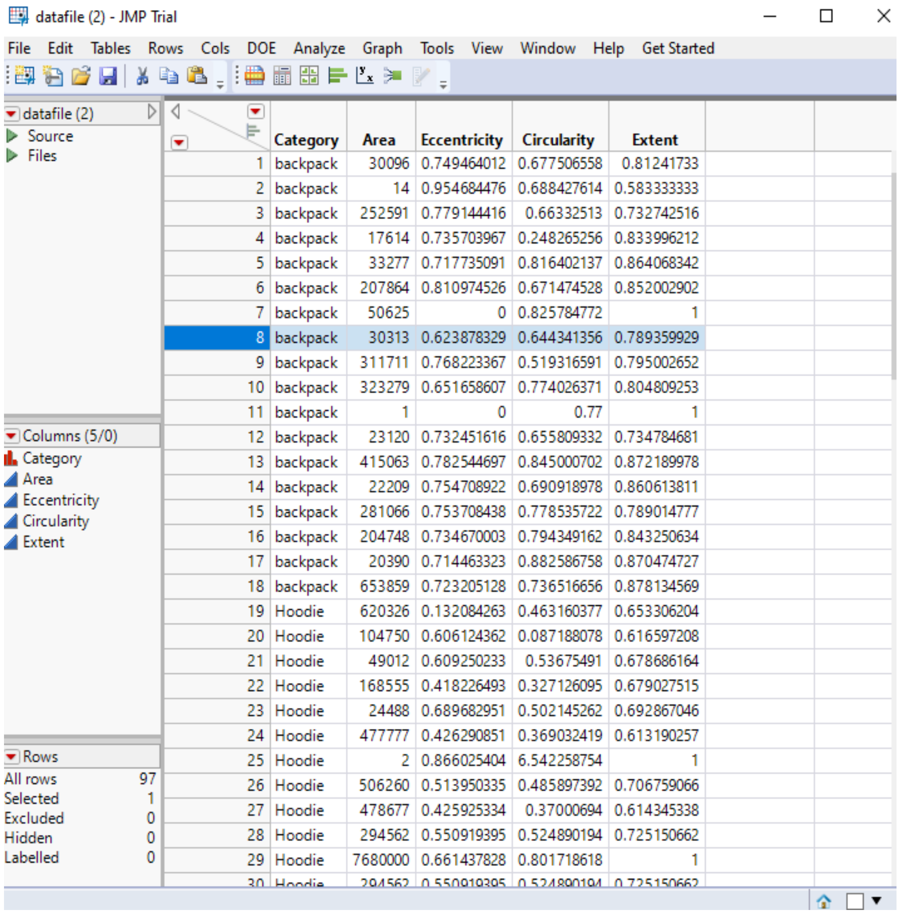

Figure 3.2.3.1 JMP Training Data Import

3.2.3.2 Predictive Modelling

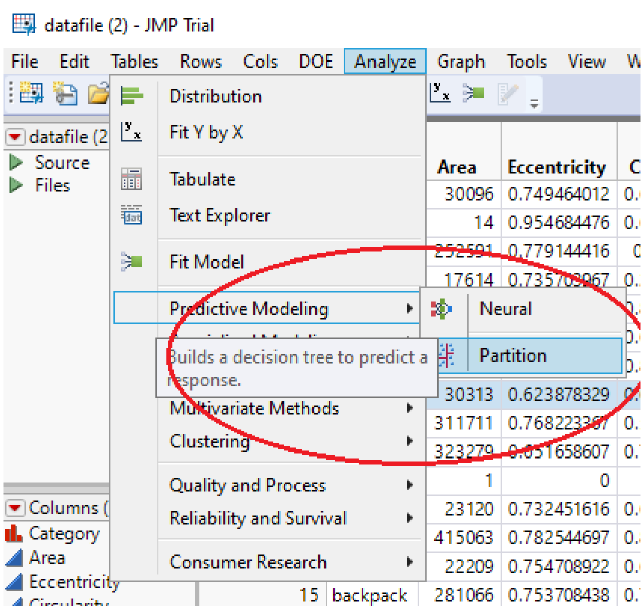

Figure 3.2.3.2 Predictive Modelling in JMP

3.2.3.3 Partition

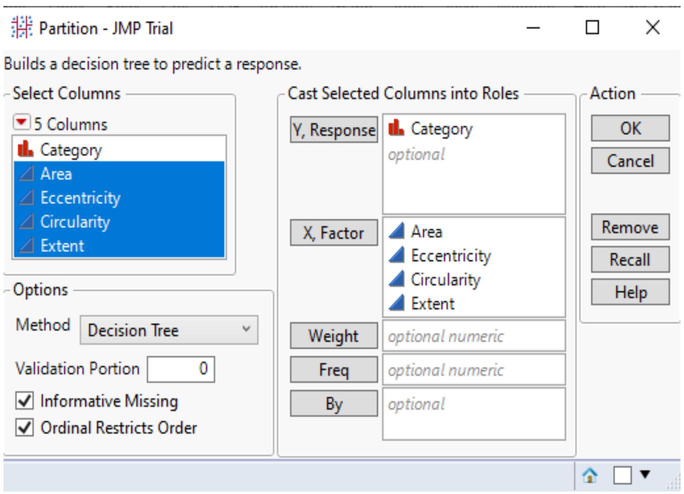

Figure 3.2.3.3 Partition in JMP

Partition is a recursive process, and the user has multiple options for continuing and controlling the process. Split button was clicked recursively to get the best split possible.

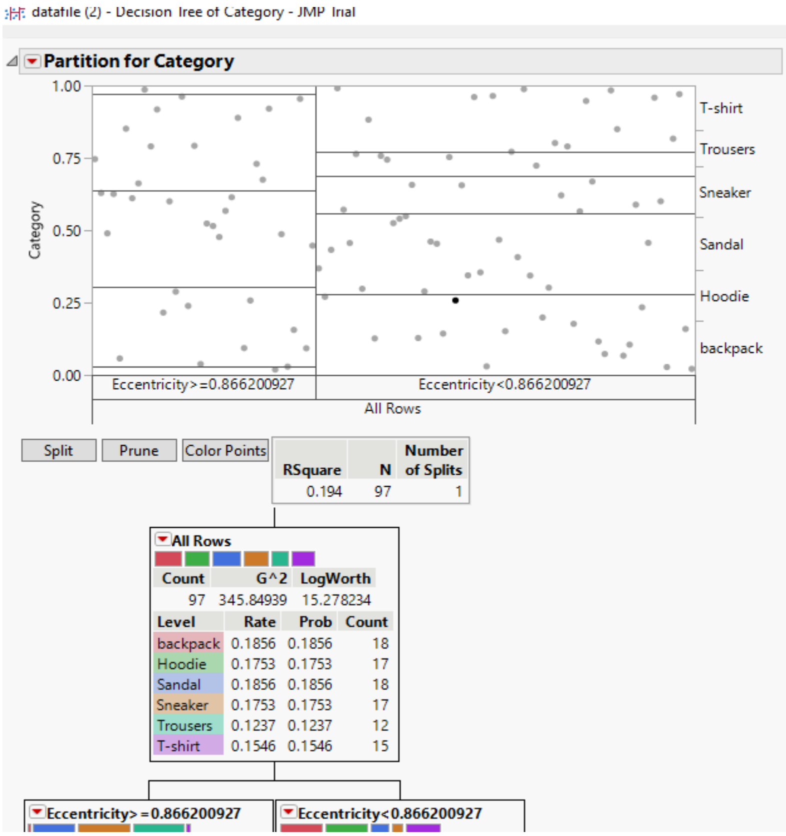

Figure 3.2.3.3 Partition by Category in JMP

3.2.3.4 Decision Tree by Partition Model

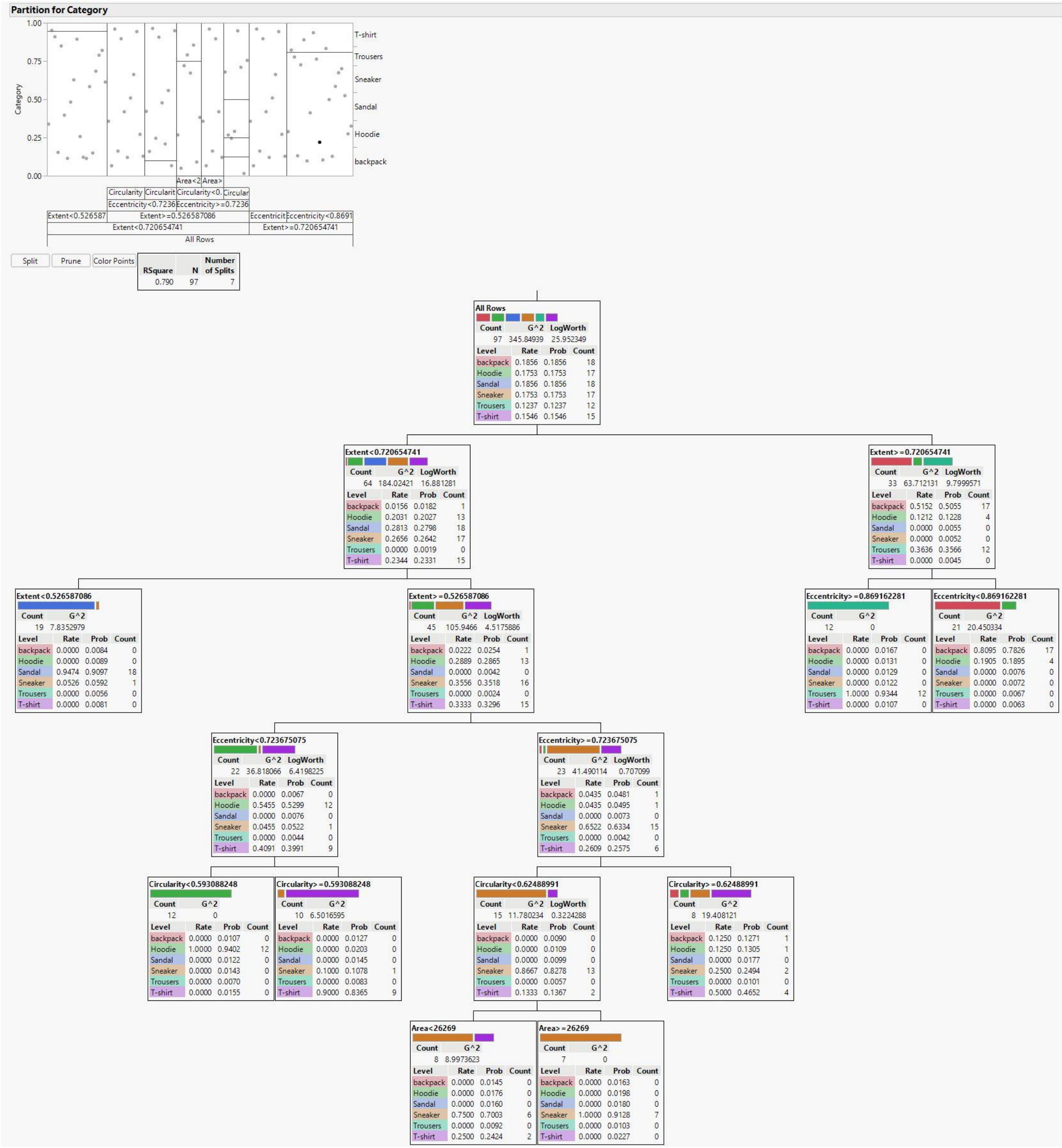

Figure 3.2.3.4 Decision Tree

3.3 Step 6: Classifier Logic based on decision tree

A classifier logic is created in MATLAB based on the decision tree built by partition model in JMP.

As shown in the below image, classifier logic is built based on the features of images and each image is classified to a specific category based on the classifier logic.

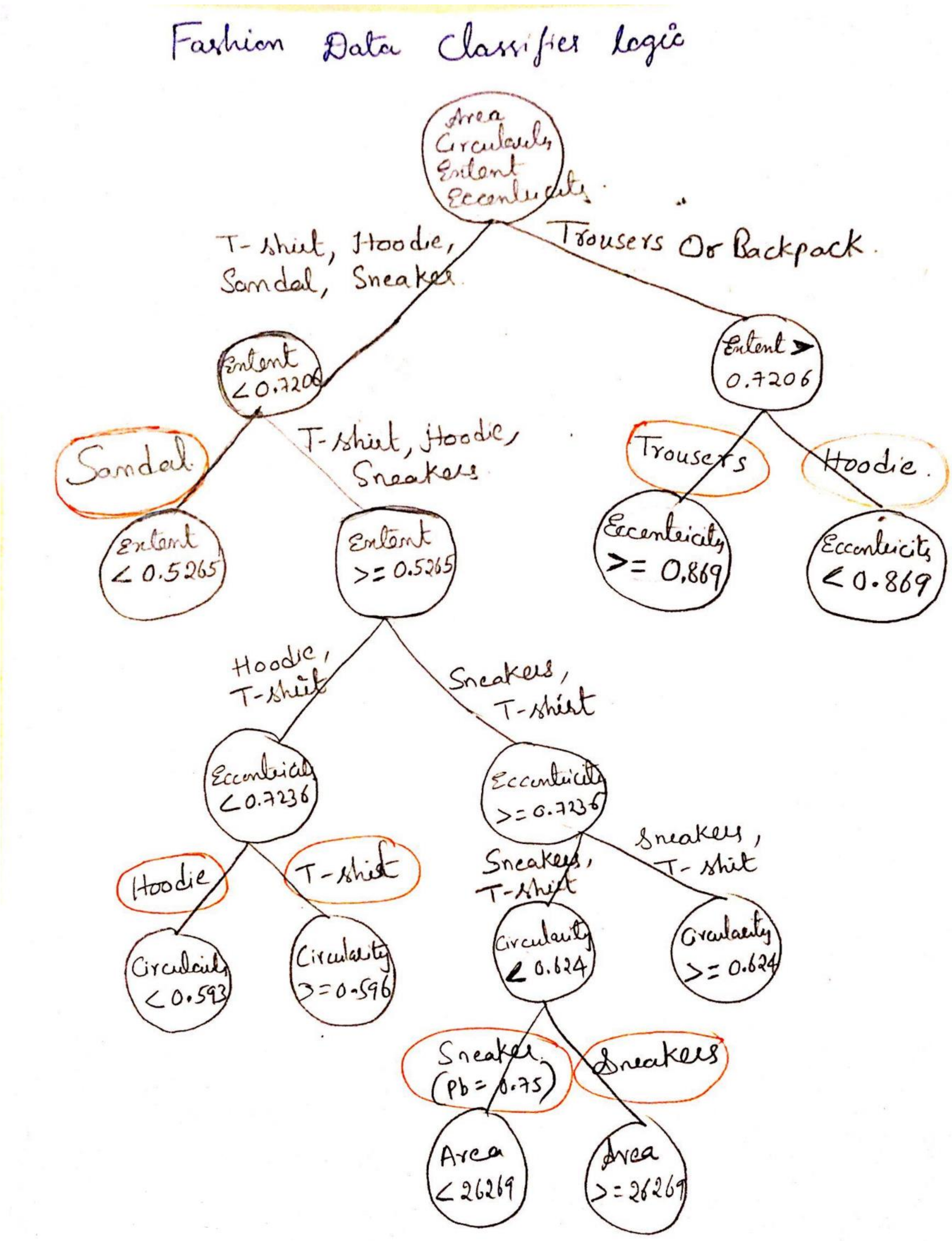

Figure 3.3.1 Classification Logic drawn based on decision tree

How I classified each image based on its regionprops features. Below is the ideology that I worked on to come up with this specific regionprops and classification logic.

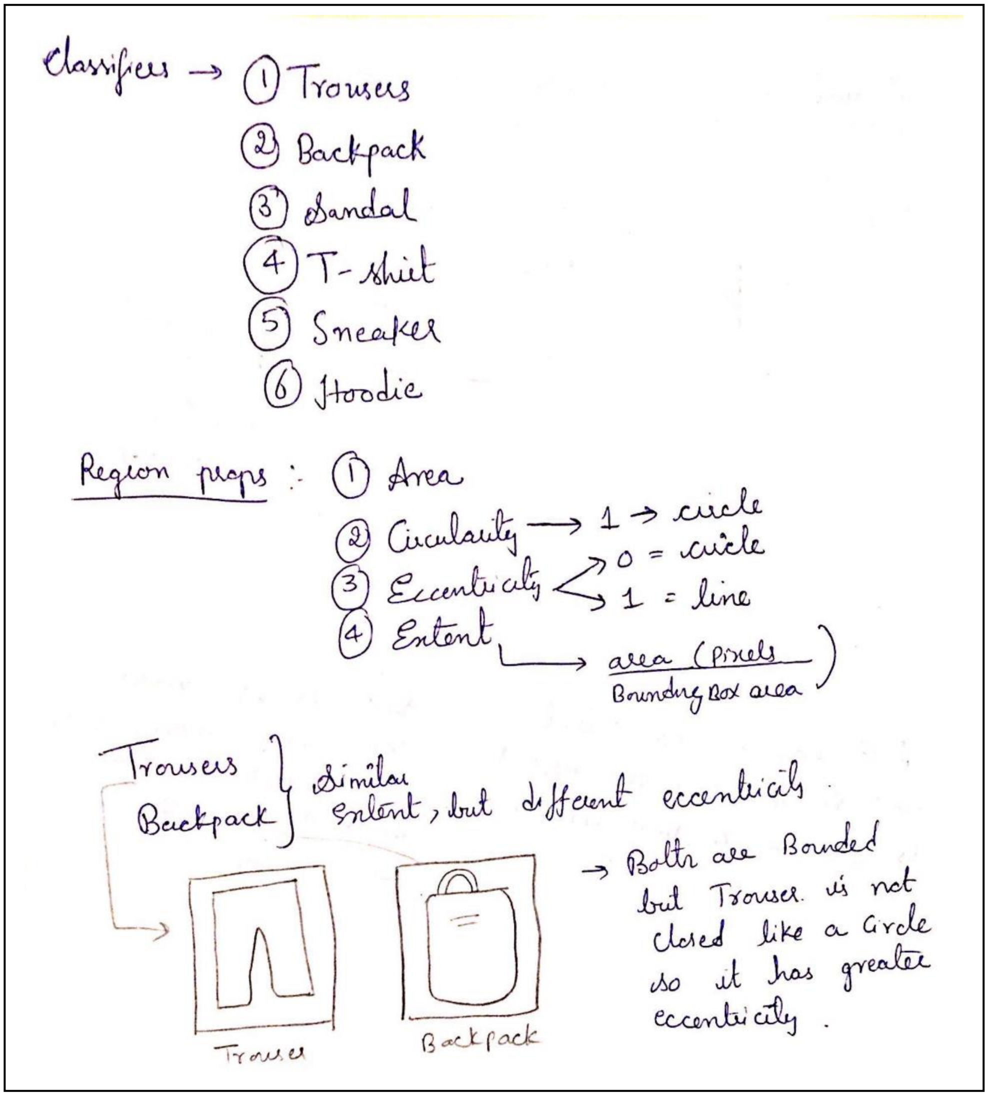

Figure 3.3.2 Classification and Image features Ideology

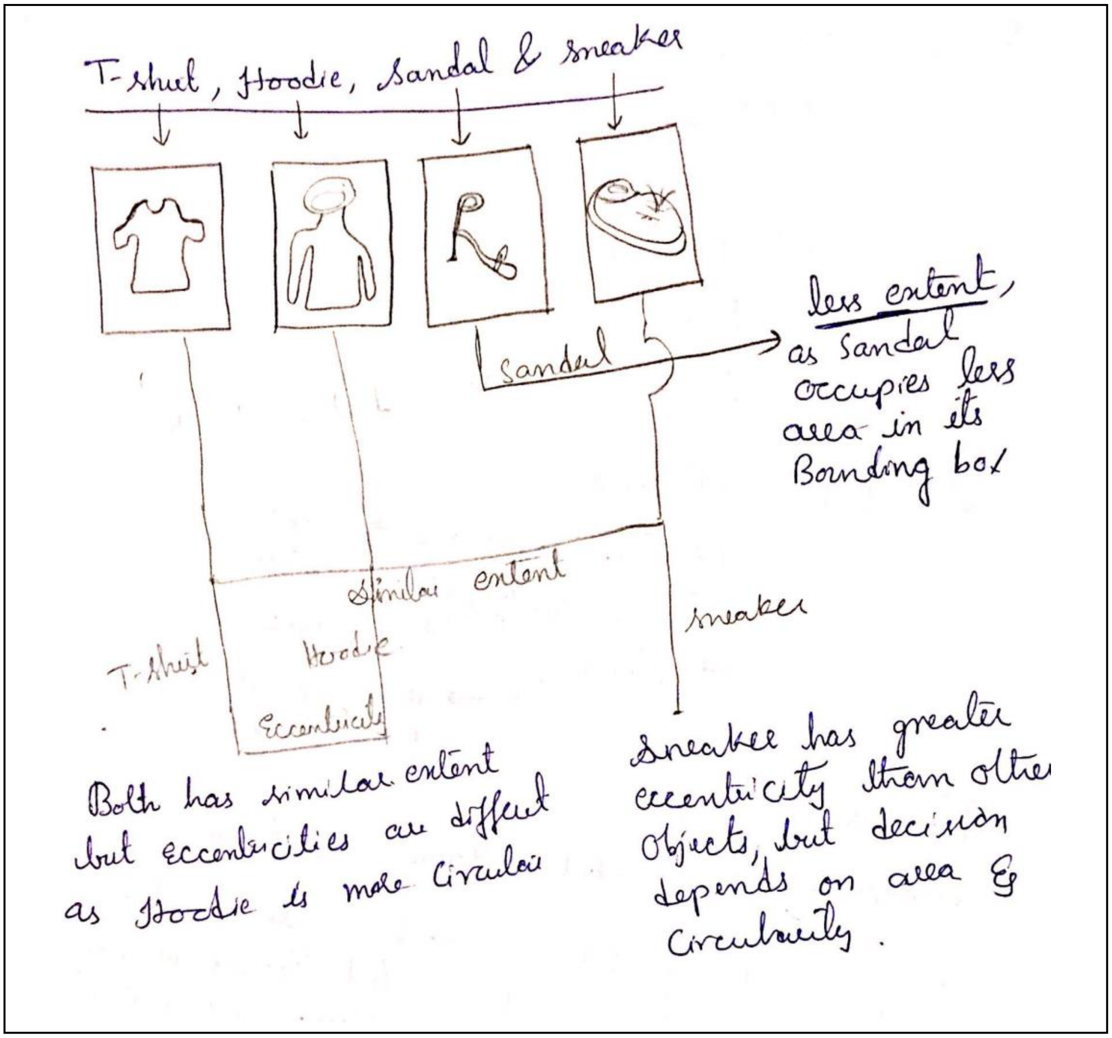

Figure 3.3.3 Classification and Image features Ideology

```
stats=regionprops(bw2,'Area','Eccentricity','Circularity','Extent');
area=stats(1).Area;    eccen=stats(1).Eccentricity;
circu=stats(1).Circularity;
ext=stats(1).Extent;
if ext<0.72
    if ext<0.5265
        figure,imshow(i1);
        title('Sandal');
    else
       if eccen<0.7236
           if circu < 0.59308
                figure,imshow(i1);
                title('Hoodie');
           else
                figure,imshow(i1);
                title('T-shirt');
           end
       else
           if circu < 0.6248
               if area >= 26269
                    figure,imshow(i1);
                     title('Sneaker');
               else
                    figure,imshow(i1);
                    title('Most Likely sneaker with pb 0.75 and T-shirt with 0.25')
               end
           else
               figure,imshow(i1);
                title('Most Likely T-shirt with pb 0.46 or sneaker with pb 0.249');
           end
       end


    end
else
    if eccen >= 0.869
        figure,imshow(i1);
        title('Trousers');
    else
        figure,imshow(i1);
        title('Backpack with probability of 0.78 and hoodie with 0.18');
    end
end
```

Figure 3.3.2 Classification Logic in MATLAB

4 Confusion Matrix

Below is the confusion matrix evaluated for the training and test data set on JMP.

The training dataset of 97 images which consists of images with 6 different classifiers such as 'Hoodie, Backpack, Sandal, Sneakers, Trousers, T-shirt'.

From the confusion matrix it is found that

1. Out of 18 backpack images, 17 are predicted as backpack and 1 is predicted as sneaker
2. Out of 17 Hoodie images, 13 are predicted as Hoodie and 4 are predicted as T-shirt, this is because Hoodies and T-shirts have similar eccentricity and circularity features.
3. Out of 18 Sandals images 16 are predicted as sandals and 2 are predicted as sneakers, because sandals and sneakers have similar extent.

4. Out of 17 sneakers, 13 are predicted as sneakers and other images are randomly predicted as other categories, as sneakers are of different types in shapes, hence it is incorrectly predicted for some type of images.
5. All 12 Trousers are predicted correctly, mainly based on the extent and eccentricity.
6. Out of 15 T-shirt's, 13 are predicted as T-shirts and 2 are classified as Hoodie and sneakers respectively.

Confusion Matrix

Training

Actual	Predicted Count					
Category	**backpack**	**Hoodie**	**Sandal**	**Sneaker**	**Trousers**	**T-shirt**
backpack	17	0	0	1	0	0
Hoodie	0	13	0	0	0	4
Sandal	0	0	16	2	0	0
Sneaker	1	1	1	13	0	1
Trousers	0	0	0	0	12	0
T-shirt	0	1	0	1	0	13

Figure 4.1 Confusion Matrix for the input dataset

5 Classification results on test images

5.1 Trouser Classifier

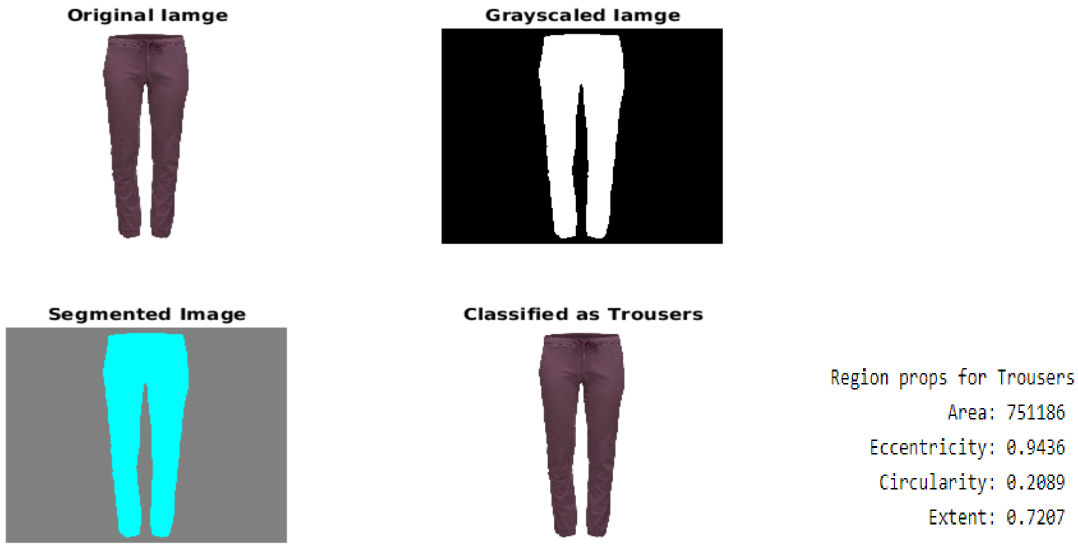

Figure 5.1.1 Trousers Classification

5.2 Sandals Classifier

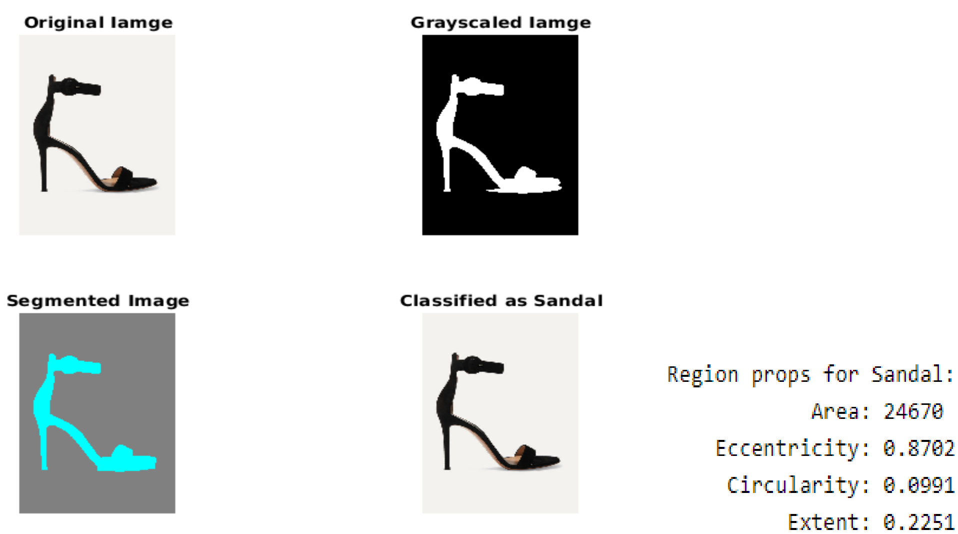

Figure 5.2.1 Sandals Classification

5.3 Hoodie Classifier


```
Region props for Hoodie:
         Area: 477777
 Eccentricity: 0.4263
  Circularity: 0.3690
       Extent: 0.6132
```

Figure 5.3.1 Hoodie Classification

5.4 T-shirt Classifier

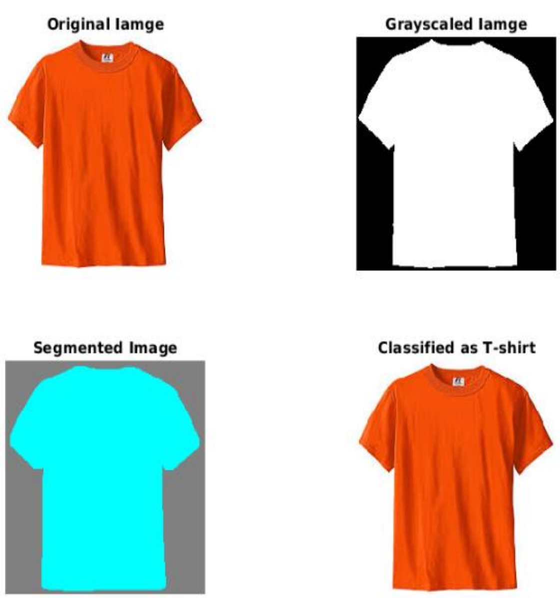

Figure 5.4.1 T-Shirt Classification

5.5 Sneakers Classifier

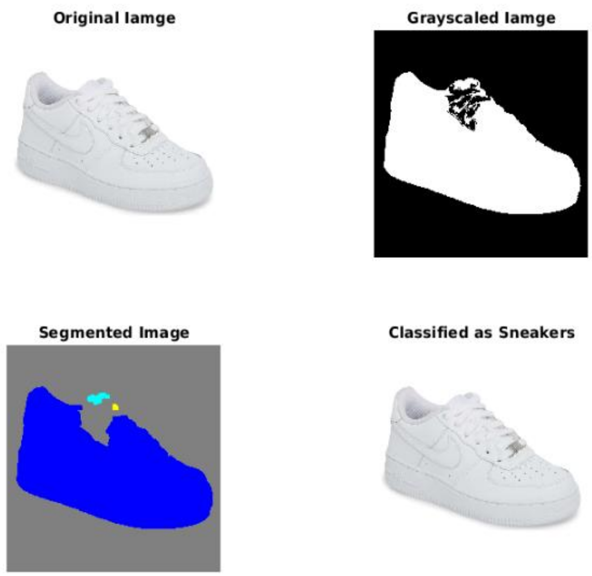

Figure 5.5.1 Sneakers Classification

5.6 Backpack Classifier

Figure 5.6.1 Backpack Classification

5.7 Classifier with probabilities

Figure 5.7.1 Backpack Classification with probability

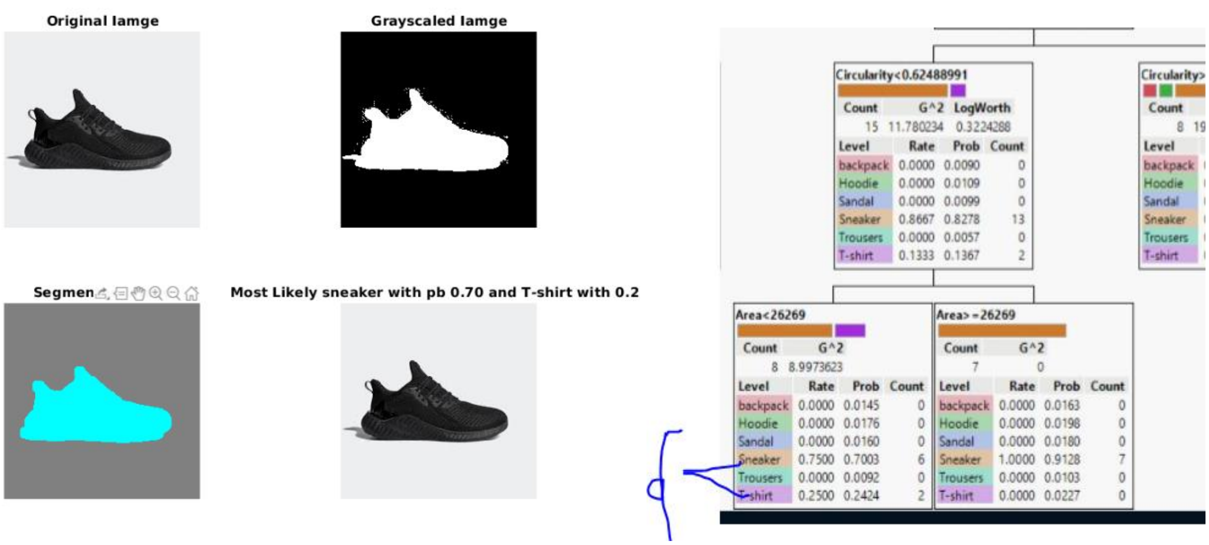

Figure 5.7.2 Sneaker Classification with probability

6 Future Work

I would like to explore how adding images from different angels impact the model performance, also I will plan to work on improving the classifier for more types of fashion-clothing and achieve more accurate results.

7 References

[1] Mathworks: https://www.mathworks.com/help/images/ref/regionprops.html

[2] Wikipedia: https://en.wikipedia.org/wiki/JMP_(statistical_software)

[3] Mira Shapiro: https://analytics.ncsu.edu/sesug/2013/JMP-04.pdf

[4] Luca Chuang: https://medium.com/luca-chuangs-bapm-notes/decision-tree-using-jmp-d61f0f9fd149