

# CS5542 Big Data Apps and Analytics

## LAB ASSIGNMENT #6

### 1. Spark Programming:

The spark programming is run on a video containing wildlife imagery. The data set contains the images of different wild animals, possibly which are present in the video. The first program is feature extraction/ Object detection. The main features are extracted from the video with training images provided. The images are of different classes or categories. For this lab, I have taken images from four categories: Elephant, Gazelle, Rhino, Zebra.

These are the images that are used for training :









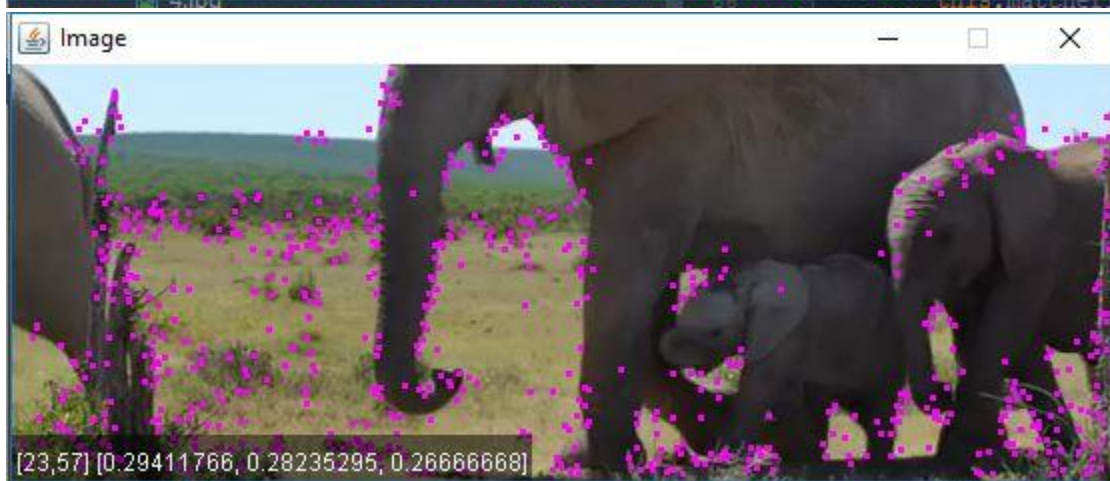






Now the program is run on these images and the video and here are the screens:

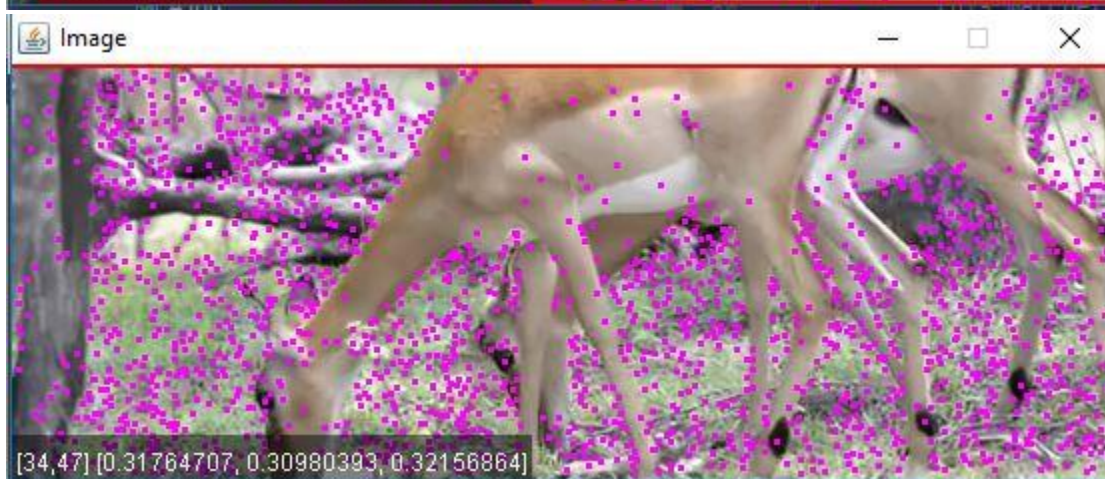












FeatureExtraction - [F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction] - [demos] - ...src\main\java\ObjectDetection.java - IntelliJ IDEA 2016.3.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

FeatureExtraction src main java ObjectDetection.java

Project FeatureExtraction [demos] F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction

data 1.jpg 2.jpg 3.jpg 4.jpg elephant.jpg gazelle.jpg rhino.jpg sample2.mkv zebra.jpg

output elephant.txt features.txt gazelle.txt rhino.txt zebra.txt

src main java ObjectDetection.java ObjectMainDetection ObjectFeatureExtraction resources test java target Classification.xml

All files are up-to-date (24 minutes ago)

69:47 CRLF UTF-8

```
39 }
40 }
41
42 class ObjectMainDetection {
43     private ConsistentLocalFeatureMatcher2d<Keypoint> matcher1;
44     private ConsistentLocalFeatureMatcher2d<Keypoint> matcher2;
45     private ConsistentLocalFeatureMatcher2d<Keypoint> matcher3;
46     private ConsistentLocalFeatureMatcher2d<Keypoint> matcher4;
47     final DoGSIFTEngine engine;
48
49     private MBFImage modelImage1, modelImage2, modelImage3, modelImage4;
50
51
52     public ObjectMainDetection() throws IOException {
53         this.engine = new DoGSIFTEngine();
54         this.engine.getOptions().setDoubleInitialImage(true);
55         this.matcher1 = new ConsistentLocalFeatureMatcher2d<Keypoint>(
56             new FastBasicKeypointMatcher<Keypoint>(threshold: 8));
57         this.matcher2 = new ConsistentLocalFeatureMatcher2d<Keypoint>(
58             new FastBasicKeypointMatcher<Keypoint>(threshold: 8));
59         this.matcher3 = new ConsistentLocalFeatureMatcher2d<Keypoint>(
60             new FastBasicKeypointMatcher<Keypoint>(threshold: 8));
61         this.matcher4 = new ConsistentLocalFeatureMatcher2d<Keypoint>(
62             new FastBasicKeypointMatcher<Keypoint>(threshold: 8));
63         final RobustHomographyEstimator ranst = new RobustHomographyEstimator(threshold: 0.5, iterations: 1500,
64             new RANSAC_PercentageInitialStoppingCondition(percentagelimit: 0.6), HomographyRefinement.NONE,
65             new TransformMatrixConditionCheck<HomographyModel>(threshold: 10000));
66         this.matcher1.setFittingModel(ranst);
67         this.matcher2.setFittingModel(ranst);
68         this.matcher3.setFittingModel(ranst);
69         this.matcher4.setFittingModel(ranst);
70         LoadReferenceObject();
```

FeatureExtraction - [F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction] - [demos] - ...src\main\java\ObjectDetection.java - IntelliJ IDEA 2016.3.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

FeatureExtraction src main java ObjectDetection.java

Project FeatureExtraction [demos] F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction

data 1.jpg 2.jpg 3.jpg 4.jpg elephant.jpg gazelle.jpg rhino.jpg sample2.mkv zebra.jpg

output elephant.txt features.txt gazelle.txt rhino.txt zebra.txt

src main java ObjectDetection.java ObjectMainDetection ObjectFeatureExtraction resources test java target Classification.xml

All files are up-to-date (25 minutes ago)

69:47 CRLF UTF-8

```
70 LoadReferenceObject();
71 StartVideo();
72 }
73
74 public void StartVideo() throws IOException {
75     Video<MBFImage> video = new XuggleVideo(new File(pathname: "data/sample2.mkv"));
76     int count1 = 0, count2 = 0, count3 = 0, count4 = 0;
77     String ol = "output/features.txt";
78     FileWriter fw = new FileWriter(ol);
79     BufferedWriter bw = new BufferedWriter(fw);
80     for (MBFImage mbfImage : video) {
81         final LocalFeatureList<Keypoint> kp1 = this.engine.findFeatures(Transforms.calculateIntensityWTSC(mbfImage));
82         final MBFImageRenderer renderer = mbfImage.createRenderer();
83         renderer.drawPoints(kp1, RGBColour.MAGENTA, (size: 3));
84
85         if (this.matcher1.findMatches(kp1)
86             && ((MatrixTransformProvider) this.matcher1.getModel()).getTransform().cond() < 1e6) {
87             try {
88                 final Matrix boundsToPoly = ((MatrixTransformProvider) this.matcher1.getModel()).getTransform()
89                     .inverse();
90
91                 if (modelImage1.getBounds().transform(boundsToPoly).isConvex()) {
92                     renderer.drawShape(this.modelImage1.getBounds().transform(boundsToPoly), thickness: 3, RGBColour.RED);
93
94                     if (count1 <= 10) {
95                         List<Point2d> vertices = this.modelImage1.getBounds().transform(boundsToPoly).asPolygon().getVertices();
96                         int x[] = new int[4], y[] = new int[4];
97                         for (int i = 0; i < vertices.size(); i++) {
98                             x[i] = (int) vertices.get(i).getX();
99                             y[i] = (int) vertices.get(i).getY();
100                         }
101                     }
```



```
FeatureExtraction - [F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction] - [demos] - ...src\main\java\ObjectDetection.java - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FeatureExtraction src main java ObjectDetection.java
Project FeatureExtraction [demos] F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction
  .idea
  data
    1.jpg
    2.jpg
    3.jpg
    4.jpg
    elephant.jpg
    gazelle.jpg
    rhino.jpg
    sample2.mkv
    zebra.jpg
  output
    elephant.txt
    features.txt
    gazelle.txt
    rhino.txt
    zebra.txt
  src
    main
      java
        ObjectDetection.java
        ObjectMainDetection.java
        ObjectFeatureExtraction.java
      resources
      test
        java
        target
        Classification.xml
  All files are up-to-date (25 minutes ago)

ObjectMainDetection ObjectMainDetection()
101
102 Polygon polygon = new Polygon(x, y, npoints: 4);
103 for (int i = 0; i < kpl.size(); i++) {
104     if (polygon.contains(kpl.get(i).getX(), kpl.get(i).getY())) {
105         double c[] = kpl.get(i).getFeatureVector().asDoubleVector();
106         bw.write( String.format("%d\t", i));
107         for (int j = 0; j < c.length; j++) {
108             bw.write( String.format("%d\t", c[j]));
109         }
110         bw.newLine();
111     }
112 }
113 count1++;
114 }
115 }
116 }
117 }
118 } catch (Exception e) {
119     System.out.println(e.getMessage());
120 }
121 }
122 }
123 if (this.matcher2.findMatches(kpl))
124     && ((MatrixTransformProvider) this.matcher2.getModel()).getTransform().cond() < 1e6 ) {
125     try {
126         final Matrix boundsToPoly = ((MatrixTransformProvider) this.matcher2.getModel()).getTransform()
127             .inverse();
128         renderer.drawShape(this.modelImage2.getBounds().transform(boundsToPoly), thickness: 3, RGBColour.RED);
129     }
130     if (count2 <= 10) {
131         if (modelImage2.getBounds().transform(boundsToPoly).isConvex()) {
132             List<Point2d> vertices = this.modelImage2.getBounds().transform(boundsToPoly).asPolygon().getVertices();
133             int x[] = new int[4], y[] = new int[4];
134             for (int i = 0; i < vertices.size(); i++) {
135                 x[i] = (int) vertices.get(i).getX();
136                 y[i] = (int) vertices.get(i).getY();
137             }
138             Polygon polygon = new Polygon(x, y, npoints: 4);
139             for (int i = 0; i < kpl.size(); i++) {
140                 if (polygon.contains(kpl.get(i).getX(), kpl.get(i).getY())) {
141                     double c[] = kpl.get(i).getFeatureVector().asDoubleVector();
142                     bw.write( String.format("%d\t", i));
143                     for (int j = 0; j < c.length; j++) {
144                         bw.write( String.format("%d\t", c[j]));
145                     }
146                     bw.newLine();
147                 }
148             }
149             count2++;
150         }
151     }
152 } catch (Exception e) {
153     System.out.println(e.getMessage());
154 }
155 }
156 }
157 }
158 }
159 }
```

```
FeatureExtraction - [F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction] - [demos] - ...src\main\java\ObjectDetection.java - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FeatureExtraction src main java ObjectDetection.java
Project FeatureExtraction [demos] F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction
  .idea
  data
    1.jpg
    2.jpg
    3.jpg
    4.jpg
    elephant.jpg
    gazelle.jpg
    rhino.jpg
    sample2.mkv
    zebra.jpg
  output
    elephant.txt
    features.txt
    gazelle.txt
    rhino.txt
    zebra.txt
  src
    main
      java
        ObjectDetection.java
        ObjectMainDetection.java
        ObjectFeatureExtraction.java
      resources
      test
        java
        target
        Classification.xml
  All files are up-to-date (25 minutes ago)

ObjectMainDetection ObjectMainDetection()
128
129 renderer.drawShape(this.modelImage2.getBounds().transform(boundsToPoly), thickness: 3, RGBColour.RED);
130
131 if (count2 <= 10) {
132     if (modelImage2.getBounds().transform(boundsToPoly).isConvex()) {
133         List<Point2d> vertices = this.modelImage2.getBounds().transform(boundsToPoly).asPolygon().getVertices();
134         int x[] = new int[4], y[] = new int[4];
135         for (int i = 0; i < vertices.size(); i++) {
136             x[i] = (int) vertices.get(i).getX();
137             y[i] = (int) vertices.get(i).getY();
138         }
139         Polygon polygon = new Polygon(x, y, npoints: 4);
140         for (int i = 0; i < kpl.size(); i++) {
141             if (polygon.contains(kpl.get(i).getX(), kpl.get(i).getY())) {
142                 double c[] = kpl.get(i).getFeatureVector().asDoubleVector();
143                 bw.write( String.format("%d\t", i));
144                 for (int j = 0; j < c.length; j++) {
145                     bw.write( String.format("%d\t", c[j]));
146                 }
147                 bw.newLine();
148             }
149         }
150         count2++;
151     }
152 } catch (Exception e) {
153     System.out.println(e.getMessage());
154 }
155 }
156 }
157 }
158 }
159 }
```

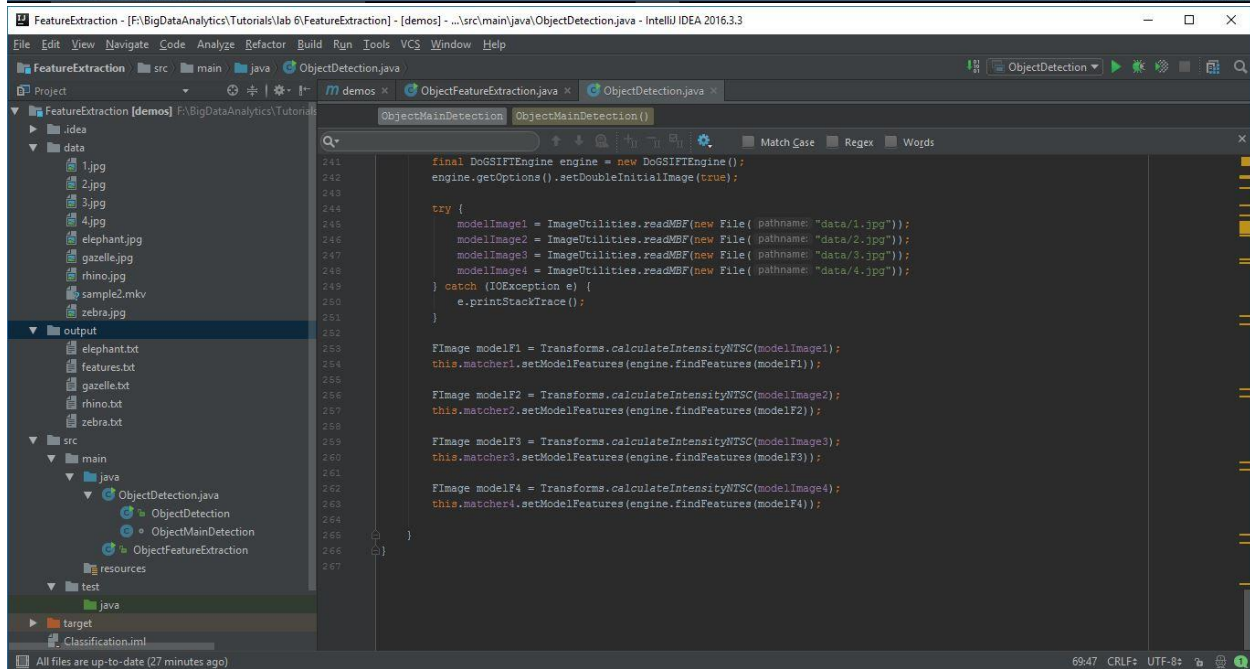
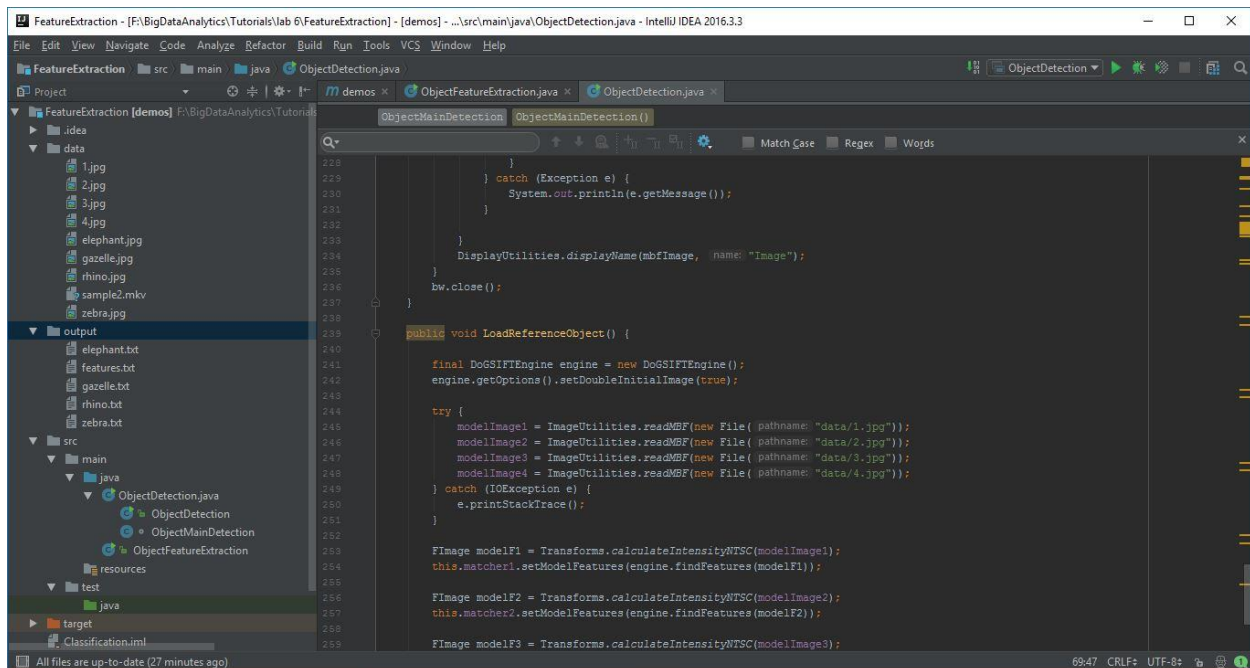
```
FeatureExtraction - [F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction] - [demos] - ...src\main\java\ObjectDetection.java - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FeatureExtraction src main java ObjectDetection.java ObjectDetection
Project FeatureExtraction [demos] F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction
  .idea
  data
    1.jpg
    2.jpg
    3.jpg
    4.jpg
    elephant.jpg
    gazelle.jpg
    rhino.jpg
    sample2.mkv
    zebra.jpg
  output
    elephant.txt
    features.txt
    gazelle.txt
    rhino.txt
    zebra.txt
  src
    main
      java
        ObjectDetection.java
        ObjectMainDetection.java
        ObjectFeatureExtraction.java
      resources
      test
        java
        target
        Classification.xml
  All files are up-to-date (25 minutes ago)

154
155
156 } catch (Exception e) {
157     System.out.println(e.getMessage());
158 }
159
160 }
161 if (this.matcher3.findMatches(kpl))
162     ss ((MatrixTransformProvider) this.matcher3.getModel()).getTransform().cond() < 1e6) {
163     try {
164         final Matrix boundsToPoly = ((MatrixTransformProvider) this.matcher3.getModel()).getTransform()
165             .inverse();
166         if (modelImage3.getBounds().transform(boundsToPoly).isConvex()) {
167             renderer.drawShape(this.modelImage3.getBounds().transform(boundsToPoly), thickness: 3, RGBColour.RED);
168             if (count3 <= 10) {
169                 List<Point2D> vertices = this.modelImage3.getBounds().transform(boundsToPoly).asPolygon().getVertices();
170                 int x[] = new int[4], y[] = new int[4];
171                 for (int i = 0; i < vertices.size(); i++) {
172                     x[i] = (int) vertices.get(i).getX();
173                     y[i] = (int) vertices.get(i).getY();
174                 }
175                 Polygon polygon = new Polygon(x, y, npoints: 4);
176                 for (int i = 0; i < kpl.size(); i++) {
177                     if (polygon.contains(kpl.get(i).getX(), kpl.get(i).getY())) {
178                         double c[] = kpl.get(i).getFeatureVector().asDoubleVector();
179                         bw.write( "2,");
180                         for (int j = 0; j < c.length; j++) {
181                             bw.write( " " + c[j] + " ");
182                         }
183                         bw.newLine();
184                     }
185                 }
186             }
187         }
188     }
189 }
```

```
FeatureExtraction - [F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction] - [demos] - ...src\main\java\ObjectDetection.java - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FeatureExtraction src main java ObjectDetection.java ObjectDetection
Project FeatureExtraction [demos] F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction
  .idea
  data
    1.jpg
    2.jpg
    3.jpg
    4.jpg
    elephant.jpg
    gazelle.jpg
    rhino.jpg
    sample2.mkv
    zebra.jpg
  output
    elephant.txt
    features.txt
    gazelle.txt
    rhino.txt
    zebra.txt
  src
    main
      java
        ObjectDetection.java
        ObjectMainDetection.java
        ObjectFeatureExtraction.java
      resources
      test
        java
        target
        Classification.xml
  All files are up-to-date (26 minutes ago)

197
198
199 if (this.matcher4.findMatches(kpl))
200     ss ((MatrixTransformProvider) this.matcher4.getModel()).getTransform().cond() < 1e6) {
201     try {
202         final Matrix boundsToPoly = ((MatrixTransformProvider) this.matcher4.getModel()).getTransform()
203             .inverse();
204         if (modelImage4.getBounds().transform(boundsToPoly).isConvex()) {
205             renderer.drawShape(this.modelImage4.getBounds().transform(boundsToPoly), thickness: 3, RGBColour.RED);
206             if (count3 <= 10) {
207                 List<Point2D> vertices = this.modelImage4.getBounds().transform(boundsToPoly).asPolygon().getVertices();
208                 int x[] = new int[4], y[] = new int[4];
209                 for (int i = 0; i < vertices.size(); i++) {
210                     x[i] = (int) vertices.get(i).getX();
211                     y[i] = (int) vertices.get(i).getY();
212                 }
213                 Polygon polygon = new Polygon(x, y, npoints: 4);
214                 for (int i = 0; i < kpl.size(); i++) {
215                     if (polygon.contains(kpl.get(i).getX(), kpl.get(i).getY())) {
216                         double c[] = kpl.get(i).getFeatureVector().asDoubleVector();
217                         bw.write( "2,");
218                         for (int j = 0; j < c.length; j++) {
219                             bw.write( " " + c[j] + " ");
220                         }
221                         bw.newLine();
222                     }
223                 }
224             }
225             count4++;
226         }
227     }
228 }
```





FeatureExtraction - [F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction] - [demos] - ..\output\elephant.txt - IntelliJ IDEA 2016.3.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

FeatureExtraction output elephant.txt

Project

- FeatureExtraction [demos] F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction
  - .idea
  - data
    - 1.jpg
    - 2.jpg
    - 3.jpg
    - 4.jpg
    - elephant.jpg
    - gazelle.jpg
    - rhino.jpg
    - sample2.mkv
    - zebra.jpg
  - output
    - elephant.txt
    - features.txt
    - gazelle.txt
    - rhino.txt
    - zebra.txt
  - src
    - main
      - java
        - ObjectDetection.java
        - ObjectDetection
        - ObjectMainDetection
        - ObjectFeatureExtraction
      - resources
    - test
      - java
    - target
    - Classification.xml

All files are up-to-date (30 minutes ago)

```
1 1,-126.0 -126.0 -126.0 -116.0 -79.0 -122.0 -128.0 -128.0 23.0 -118.0 -128.0 -120.0 -9.0 -104.0 -127.0 -122.0 50.0 -86.0 -123.0 -116.0
2 0,-125.0 -125.0 -127.0 -126.0 -123.0 -124.0 -119.0 -123.0 -22.0 -115.0 -126.0 -114.0 -100.0 -126.0 -126.0 -125.0 40.0 -111.0 -128.0 -128.0
3 0,-88.0 -30.0 -69.0 -127.0 -127.0 -114.0 -118.0 -102.0 -106.0 -67.0 -69.0 -76.0 -97.0 -124.0 -128.0 -124.0 -95.0 -124.0 -122.0 -102.0
4 0,-109.0 -5.0 -121.0 -128.0 -128.0 -112.0 -124.0 -113.0 -32.0 -98.0 -122.0 -116.0 -84.0 -94.0 -125.0 -121.0 -82.0 -91.0 -117.0 -128.0
5 0,-125.0 -127.0 -127.0 -115.0 -109.0 -128.0 -128.0 -127.0 -36.0 -128.0 -128.0 -120.0 -111.0 -128.0 -127.0 -35.0 -82.0 -128.0 -119.0 -99.0
6 0,-119.0 -127.0 -127.0 -128.0 -126.0 -126.0 -105.0 -82.0 -99.0 -125.0 -128.0 -128.0 -126.0 -127.0 -102.0 -8.0 -110.0 -127.0 -128.0 -128.0
7 0,-3.0 -82.0 -126.0 -128.0 -128.0 -128.0 -125.0 -106.0 -49.0 -95.0 -126.0 -128.0 -128.0 -128.0 -117.0 -64.0 -47.0 -45.0 -111.0 -128.0 -128.0
8 0,-98.0 -128.0 -128.0 -128.0 -128.0 -128.0 -104.0 35.0 -127.0 -128.0 -127.0 -127.0 -127.0 -124.0 -33.0 -63.0 -124.0 -126.0 -123.0 -80.0 -128.0
9 0,-128.0 -128.0 -128.0 -115.0 -88.0 -60.0 -83.0 -128.0 -125.0 -128.0 -128.0 -125.0 -124.0 -89.0 -11.0 -112.0 -87.0 -83.0 -127.0 -128.0 -128.0
10 0,-91.0 -126.0 -123.0 -124.0 -127.0 -126.0 -88.0 -4.0 -4.0 -124.0 -122.0 -107.0 -120.0 -127.0 -122.0 -4.0 -88.0 -123.0 -115.0 -4.0 -70.0 -128.0
11 0,-106.0 -55.0 -115.0 -124.0 -118.0 -121.0 -128.0 -128.0 -116.0 -95.0 -119.0 -119.0 -24.0 -60.0 -92.0 -115.0 -68.0 -15.0 -96.0 -113.0 -128.0
12 0,-116.0 -94.0 -107.0 -110.0 -97.0 -117.0 -127.0 -128.0 3.0 -60.0 -123.0 -128.0 -128.0 -128.0 -128.0 -107.0 -38.0 -109.0 -127.0 -63.0 -128.0
13 0,-104.0 -88.0 -123.0 -124.0 -102.0 -75.0 -126.0 -125.0 -89.0 -93.0 -126.0 -127.0 -118.0 -86.0 -119.0 -113.0 -127.0 -118.0 -95.0 -122.0 -128.0
14 0,-98.0 -123.0 -127.0 -126.0 -107.0 -108.0 -121.0 -115.0 13.0 -109.0 -127.0 -127.0 -122.0 -110.0 -122.0 -85.0 -70.0 -113.0 -125.0 -126.0 -128.0
15 0,-115.0 -84.0 -122.0 -124.0 -95.0 -97.0 -126.0 -127.0 -117.0 -81.0 -71.0 -114.0 -71.0 -99.0 -128.0 -128.0 -10.0 -59.0 -102.0 -126.0 -128.0 -128.0
16 0,-124.0 -126.0 -122.0 -63.0 -6.0 -108.0 -126.0 -125.0 -127.0 -128.0 -128.0 -70.0 -11.0 -120.0 -107.0 -115.0 -128.0 -126.0 -128.0 -128.0 -128.0
17 0,-78.0 -107.0 -128.0 -128.0 -128.0 -128.0 -128.0 -128.0 0.0 -96.0 -128.0 -126.0 -119.0 -126.0 -127.0 -121.0 -108.0 -95.0 -115.0 -93.0 -128.0
18 0,-123.0 -104.0 -111.0 -128.0 -128.0 -128.0 -126.0 -124.0 -21.0 -75.0 -96.0 -127.0 -128.0 -128.0 -128.0 -126.0 57.0 -59.0 -128.0 -128.0 -128.0
19 0,-118.0 -114.0 -113.0 -93.0 -86.0 -99.0 -120.0 -123.0 -99.0 -79.0 -55.0 -9.0 -85.0 -127.0 -127.0 -120.0 -76.0 -74.0 -121.0 -89.0 -80.0 -128.0
20 0,-98.0 -123.0 -128.0 -128.0 -72.0 -80.0 -109.0 -100.0 -95.0 -122.0 -126.0 -113.0 -103.0 -30.0 -94.0 -95.0 -7.0 -87.0 -125.0 -123.0 -128.0 -128.0
21 0,-56.0 -95.0 -128.0 -128.0 -128.0 -128.0 -128.0 -127.0 5.0 -63.0 -125.0 -128.0 -128.0 -128.0 -128.0 -111.0 -87.0 -83.0 -87.0 -124.0 -128.0 -128.0
22 0,-83.0 -104.0 -49.0 -110.0 -119.0 -116.0 -123.0 -119.0 -26.0 -100.0 -105.0 -122.0 -127.0 -128.0 -101.0 -46.0 -126.0 -127.0 -128.0 -128.0 -128.0
23 0,-33.0 -93.0 -128.0 -128.0 -128.0 -128.0 -128.0 -127.0 41.0 23.0 -118.0 -128.0 -128.0 -128.0 -128.0 -128.0 -82.0 -83.0 -96.0 -185.0 -126.0 -128.0
24 0,-122.0 -126.0 -128.0 -128.0 -128.0 -128.0 -128.0 -127.0 25.0 -113.0 -128.0 -128.0 -128.0 -128.0 -128.0 -128.0 -128.0 -125.0 44.0 -99.0 -128.0 -128.0 -128.0
25 0,-107.0 -98.0 -115.0 -108.0 -103.0 -123.0 -123.0 -128.0 -120.0 -87.0 -102.0 -119.0 -116.0 -94.0 -118.0 -120.0 -117.0 -126.0 -123.0 -123.0 -111.0 -128.0
26 0,-111.0 -7.0 -48.0 -126.0 -128.0 -128.0 -128.0 -107.0 -111.0 -5.0 -4.0 -96.0 -128.0 -128.0 -128.0 -127.0 -96.0 -28.0 -53.0 -123.0 -125.0 -118.0 -128.0
27 0,-128.0 -124.0 -123.0 -111.0 -95.0 -88.0 -102.0 -127.0 -71.0 -125.0 -128.0 -124.0 -113.0 -96.0 -116.0 -121.0 3.0 -117.0 -128.0 -128.0 -128.0 -128.0
28 0,-86.0 -117.0 -113.0 -97.0 -125.0 -128.0 -128.0 -95.0 -122.0 -126.0 -121.0 1.0 -105.0 -128.0 -128.0 -125.0 -56.0 -120.0 -116.0 -38.0 -128.0 -128.0
29 0,-68.0 -120.0 -126.0 -123.0 -112.0 -86.0 -98.0 -53.0 -127.0 -104.0 -118.0 -104.0 -39.0 -111.0 -127.0 -128.0 -128.0 -128.0 -128.0 -128.0 -119.0 -128.0
30 0,-51.0 -108.0 -127.0 -125.0 -126.0 -128.0 -128.0 -126.0 3.0 -30.0 -117.0 -118.0 -123.0 -125.0 -127.0 -116.0 -121.0 -109.0 -114.0 -128.0 -128.0
31 0,-117.0 -92.0 -123.0 -120.0 -102.0 -95.0 -127.0 -128.0 -107.0 -92.0 -117.0 -97.0 -111.0 -128.0 -128.0 -128.0 -128.0 -128.0 -128.0 -128.0 -128.0
32 0,-114.0 -109.0 -124.0 -111.0 -111.0 -117.0 -125.0 -125.0 -34.0 -108.0 -124.0 -118.0 -115.0 -112.0 -123.0 -108.0 -106.0 -119.0 -127.0 -128.0 -128.0
33 0,-104.0 -115.0 -127.0 -125.0 -76.0 -67.0 -113.0 -121.0 -94.0 -110.0 -122.0 -103.0 -84.0 -121.0 -128.0 -128.0 -39.0 -97.0 -126.0 -125.0 -128.0 -128.0
34 0,-69.0 -86.0 -117.0 -124.0 -127.0 -128.0 -123.0 -79.0 -106.0 -107.0 -93.0 -76.0 -92.0 -109.0 -112.0 -108.0 -20.0 -76.0 -92.0 -94.0 -10.0 -128.0 -128.0
35 0,-84.0 -123.0 -110.0 -108.0 -107.0 -122.0 -119.0 -54.0 -108.0 -127.0 -127.0 -124.0 -118.0 -91.0 -61.0 -62.0 -17.0 -118.0 -128.0 -126.0 -89.0 -49.0 -128.0 -128.0
```

FeatureExtraction - [F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction] - [demos] - ..\output\gazelle.txt - IntelliJ IDEA 2016.3.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

FeatureExtraction output gazelle.txt

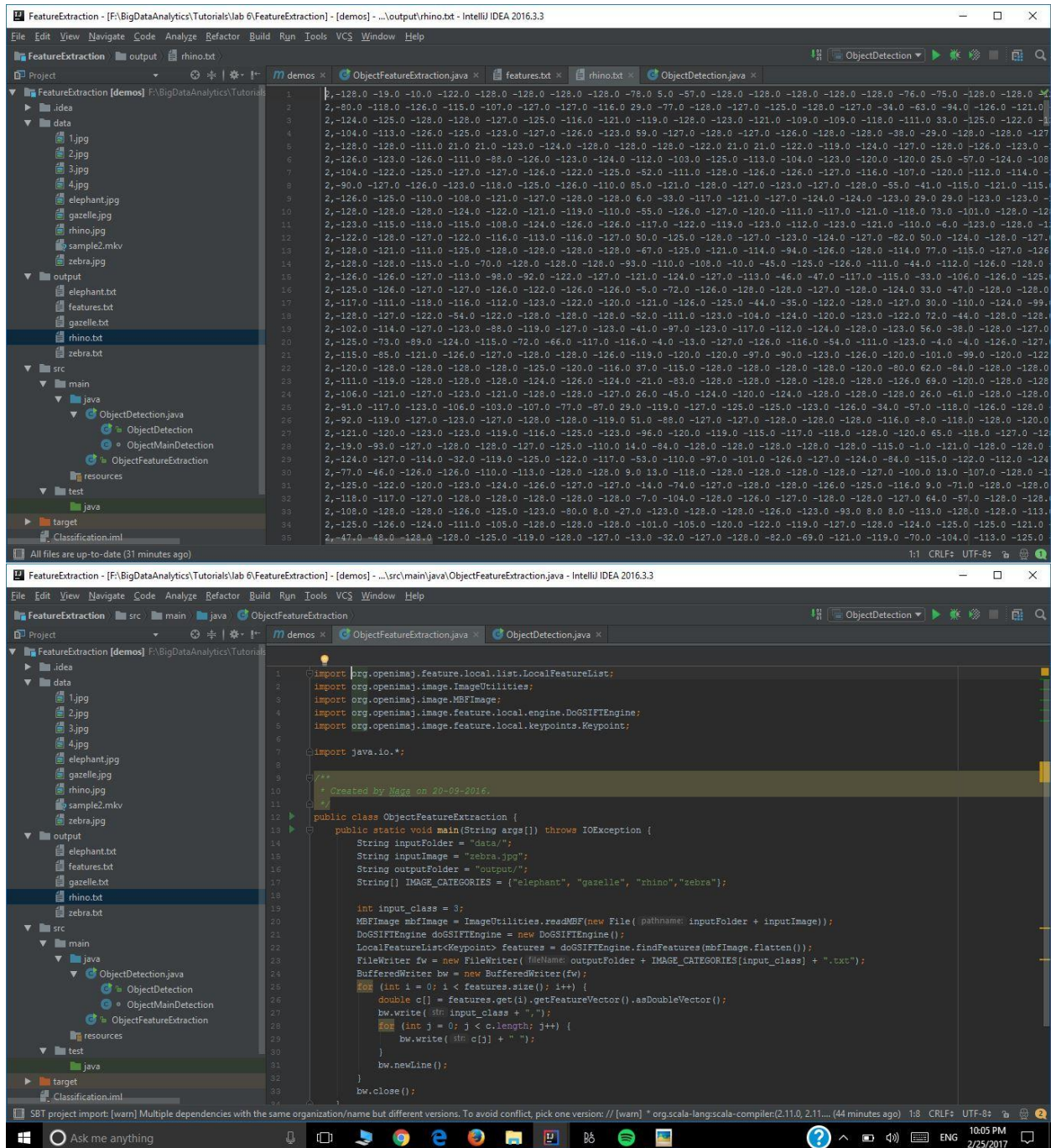
Project

- FeatureExtraction [demos] F:\BigDataAnalytics\Tutorial\lab 6\FeatureExtraction
  - .idea
  - data
    - 1.jpg
    - 2.jpg
    - 3.jpg
    - 4.jpg
    - elephant.jpg
    - gazelle.jpg
    - rhino.jpg
    - sample2.mkv
    - zebra.jpg
  - output
    - elephant.txt
    - features.txt
    - gazelle.txt
    - rhino.txt
    - zebra.txt
  - src
    - main
      - java
        - ObjectDetection.java
        - ObjectDetection
        - ObjectMainDetection
        - ObjectFeatureExtraction
      - resources
    - test
      - java
    - target
    - Classification.xml

All files are up-to-date (30 minutes ago)

```
1 1,-128.0 -128.0 -128.0 -128.0 -123.0 -103.0 -123.0 -128.0 -128.0 -127.0 -118.0 -115.0 -121.0 -113.0 -122.0 -128.0 -126.0 -119.0 -116.0 -128.0 -128.0
2 1,-128.0 -128.0 -128.0 -128.0 -122.0 -112.0 -128.0 -128.0 -86.0 -127.0 -128.0 -108.0 36.0 -126.0 -128.0 -120.0 36.0 -126.0 -128.0 -128.0 -128.0
3 1,-125.0 -126.0 -128.0 -128.0 -117.0 -127.0 -127.0 -121.0 -77.0 -126.0 -128.0 -128.0 14.0 -115.0 -128.0 -123.0 14.0 -106.0 -128.0 -128.0 -128.0
4 1,-96.0 -126.0 -128.0 -127.0 -24.0 -117.0 -128.0 -127.0 8.0 -104.0 -128.0 -127.0 -59.0 -109.0 -128.0 -123.0 -50.0 -108.0 -128.0 -128.0 -128.0
5 1,-105.0 -128.0 -128.0 -128.0 -24.0 -121.0 -128.0 -123.0 9.0 -119.0 -128.0 -127.0 -80.0 -117.0 -128.0 -103.0 -66.0 -117.0 -128.0 -126.0 -128.0
6 1,-122.0 -120.0 -128.0 -127.0 -60.0 -105.0 -128.0 -128.0 -41.0 -91.0 -128.0 -128.0 -13.0 -75.0 -128.0 -128.0 -1.0 -15.0 -128.0 -128.0 -128.0 -128.0
7 1,-119.0 -103.0 -123.0 -116.0 -31.0 -119.0 -128.0 -128.0 -81.0 -39.0 -121.0 -112.0 -111.0 -128.0 -128.0 -128.0 -81.0 -117.0 -123.0 -128.0 -128.0
8 1,-78.0 -126.0 -128.0 -110.0 -6.0 -123.0 -126.0 -88.0 -85.0 -126.0 -127.0 -108.0 -5.0 -81.0 -120.0 -94.0 16.0 -127.0 -128.0 -125.0 -99.0 -128.0
9 1,-85.0 -124.0 -127.0 -110.0 -15.0 -99.0 -123.0 -100.0 39.0 -119.0 -128.0 -121.0 -68.0 -117.0 -127.0 -92.0 -26.0 -128.0 -128.0 -128.0 -128.0 -128.0
10 1,-128.0 -128.0 -61.0 -4.0 -76.0 -127.0 -128.0 -128.0 -125.0 -120.0 -4.0 -49.0 -46.0 -117.0 -126.0 -125.0 -128.0 -102.0 -4.0 -79.0 -39.0 -128.0 -128.0
11 1,-127.0 -115.0 -114.0 -127.0 -128.0 -128.0 -128.0 -128.0 -125.0 -122.0 -122.0 -127.0 -125.0 -122.0 -117.0 -114.0 -128.0 -121.0 -116.0 -128.0 -128.0
12 1,-57.0 -117.0 -127.0 -101.0 -36.0 -126.0 -127.0 -107.0 -79.0 -117.0 -124.0 -70.0 14.0 -122.0 -127.0 -121.0 14.0 -88.0 -125.0 -125.0 -117.0 -128.0 -128.0
13 1,-117.0 -114.0 -116.0 -116.0 -95.0 -77.0 -5.0 -100.0 -114.0 -119.0 -123.0 -88.0 -84.0 -71.0 -98.0 -121.0 -6.0 -81.0 -122.0 -120.0 -126.0 -128.0 -128.0
14 1,-11.0 -108.0 -85.0 -114.0 -128.0 -128.0 -128.0 -73.0 -106.0 -65.0 -26.0 -112.0 -115.0 -120.0 -126.0 -122.0 -123.0 -120.0 -120.0 -123.0 -128.0 -128.0
15 1,-128.0 -126.0 -124.0 -122.0 -121.0 -124.0 -127.0 -128.0 -56.0 -110.0 -125.0 -128.0 -128.0 -124.0 -120.0 -112.0 -14.0 -120.0 -120.0 -128.0 -128.0 -128.0
16 1,-125.0 -115.0 -102.0 -122.0 -127.0 -128.0 -128.0 -128.0 -124.0 -115.0 -89.0 -81.0 -117.0 -128.0 -128.0 -128.0 -113.0 -119.0 -119.0 -106.0 -8.0 -128.0 -128.0
17 1,-128.0 -128.0 -114.0 -5.0 -71.0 -128.0 -128.0 -128.0 -128.0 -123.0 -121.0 -3.0 -61.0 -125.0 -110.0 -124.0 -122.0 -73.0 -100.0 -121.0 -128.0 -128.0
18 1,-124.0 -113.0 -86.0 -123.0 -123.0 -111.0 -127.0 -112.0 -97.0 -36.0 -113.0 -124.0 -121.0 -110.0 -113.0 -79.0 -85.0 -120.0 -127.0 -128.0 -128.0 -128.0
19 1,-128.0 -128.0 -128.0 -105.0 2.0 -111.0 -128.0 -128.0 -98.0 -113.0 -127.0 -94.0 2.0 -109.0 -127.0 -122.0 -4.0 -81.0 -128.0 -126.0 -126.0 -122.0 -128.0 -128.0
20 1,-110.0 -107.0 -107.0 -126.0 -127.0 -128.0 -128.0 -128.0 -128.0 -105.0 -110.0 -124.0 -81.0 -96.0 -116.0 -128.0 -127.0 -79.0 -124.0 -126.0 -111.0 -128.0 -128.0
21 1,-98.0 -91.0 -94.0 -78.0 -108.0 -128.0 -128.0 -126.0 27.0 -46.0 -106.0 -116.0 -123.0 -128.0 -128.0 -98.0 -83.0 -115.0 -126.0 -94.0 -86.0 -128.0 -128.0
22 1,-124.0 -128.0 -123.0 -75.0 -128.0 -128.0 -128.0 -112.0 -123.0 -128.0 -126.0 -99.0 -107.0 -117.0 -98.0 -89.0 -108.0 -124.0 -117.0 -112.0 -128.0 -128.0
23 1,-107.0 -125.0 -128.0 -128.0 -128.0 -125.0 -115.0 -58.0 -96.0 -128.0 -128.0 -128.0 -128.0 -122.0 -92.0 -6.0 -70.0 -125.0 -127.0 -126.0 -128.0 -128.0
24 1,-108.0 -58.0 -83.0 -116.0 -100.0 -96.0 -125.0 -124.0 -127.0 -123.0 -123.0 -122.0 -49.0 -6.0 -116.0 -128.0 -119.0 -128.0 -128.0 -128.0 -128.0 -128.0
25 1,-104.0 -125.0 -127.0 -128.0 -128.0 -128.0 -128.0 -124.0 -61.0 -38.0 -111.0 -128.0 -128.0 -128.0 -128.0 -123.0 -52.0 -99.0 -127.0 -122.0 -128.0 -128.0
26 1,-127.0 -122.0 -112.0 -117.0 -127.0 -128.0 -128.0 -127.0 -123.0 -123.0 -114.0 -43.0 -89.0 -128.0 -128.0 -126.0 -1.0 -119.0 -120.0 -82.0 -128.0 -128.0
27 1,-29.0 -86.0 -121.0 -128.0 -128.0 -128.0 -128.0 -128.0 -124.0 -61.0 -38.0 -111.0 -128.0 -128.0 -128.0 -128.0 -123.0 -52.0 -99.0 -127.0 -122.0 -128.0 -128.0
28 1,-108.0 -128.0 -128.0 -128.0 -128.0 -124.0 -48.0 -6.0 -121.0 -126.0 -127.0 -125.0 -110.0 -18.0 -6.0 -89.0 -123.0 -127.0 -126.0 -127.0 -128.0 -128.0
29 1,-6.0 -117.0 -127.0 -121.0 -124.0 -127.0 -126.0 -84.0 -5.0 -54.0 -128.0 -128.0 -128.0 -128.0 -128.0 -103.0 -5.0 -76.0 -124.0 -128.0 -128.0 -128.0 -128.0
30 1,-128.0 -127.0 -109.0 -68.0 -69.0 -100.0 -128.0 -128.0 -96.0 -102.0 -112.0 -110.0 -119.0 -128.0 -128.0 -128.0 -115.0 -113.0 -116.0 -74.0 -128.0 -128.0
31 1,-124.0 -126.0 -111.0 -93.0 -71.0 -124.0 -126.0 -126.0 -80.0 -45.0 -94.0 -104.0 -128.0 -128.0 -125.0 -126.0 -7.0 7.0 -119.0 -122.0 -128.0 -128.0 -128.0
32 1,-125.0 -128.0 -115.0 18.0 -126.0 -128.0 -128.0 -126.0 -24.0 -93.0 -122.0 -101.0 -128.0 -128.0 -128.0 -118.0 18.0 -35.0 -128.0 -128.0 -128.0 -128.0 -128.0
33 1,-66.0 -81.0 -93.0 -105.0 -123.0 -126.0 -117.0 -114.0 38.0 -39.0 -124.0 -124.0 -127.0 -128.0 -128.0 -127.0 -71.0 -122.0 -128.0 -128.0 -128.0 -128.0 -128.0
34 1,-119.0 -98.0 -81.0 -120.0 -128.0 -128.0 -128.0 -128.0 -69.0 -64.0 -86.0 -120.0 -128.0 -128.0 -128.0 -123.0 -5.0 -20.0 -108.0 -127.0 -128.0 -128.0 -128.0
35 1,-111.0 -118.0 -120.0 -102.0 -124.0 -128.0 -128.0 -127.0 -128.0 -127.0 -97.0 3.0 -92.0 -121.0 -128.0 -128.0 -126.0 -89.0 -49.0 -105.0 -128.0 -128.0 -128.0
```





The features extracted from these are divided into test and train features, the image classification algorithm is run using this test and train features. Here are the screens:





```
ImageClassification - [F:\BigDataAnalytics\Tutorials\lab 6 screens\ImageClassification] - [imageclassification] - ...src\main\scala\ImageClassification_DT.scala - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ImageClassification src / main scala ImageClassification_DT.scala
Run ImageClassification_DT
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

2.0 : (0,12.718505123566415);
(1,10.608800482218204);
(2,76.67269439421338)

0.0 : (0,14.597970335675253);
(1,12.256049960967994);
(2,73.14597970335676)

3.0 : (0,33.74050841862001);
(1,13.04060746120832);
(2,53.21888412017167)

1.0 : (0,15.340442986054143);
(1,14.438063986874488);
(2,70.22149302707138)
(2,0,2,0)
(2,0,0,0)
(2,0,3,0)
(2,0,1,0)
Accuracy:0.25
Confusion Matrix:
0.0 0.0 1.0 0.0
0.0 0.0 1.0 0.0
0.0 0.0 1.0 0.0
0.0 0.0 1.0 0.0
Process finished with exit code 0
```

```
ImageClassification - [F:\BigDataAnalytics\Tutorials\lab 6 screens\ImageClassification] - [imageclassification] - ...src\main\scala\ImageClassification_RF.scala - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ImageClassification src / main scala ImageClassification_RF.scala
Run ImageClassification_RF
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

2.0 : (0,10.729355033152501);
(1,1.567209162145871);
(2,87.70343590470163)

0.0 : (0,13.817330210772834);
(1,2.107728337236534);
(2,84.07494145199063)

3.0 : (0,30.57114559260482);
(1,3.1363486299108616);
(2,66.29250577748432)

1.0 : (0,11.238720262510254);
(1,2.7891714520098443);
(2,85.9721082854799)
(2,0,2,0)
(2,0,0,0)
(2,0,3,0)
(2,0,1,0)
Accuracy:0.25
Confusion Matrix:
0.0 0.0 1.0 0.0
0.0 0.0 1.0 0.0
0.0 0.0 1.0 0.0
0.0 0.0 1.0 0.0
Process finished with exit code 0
```

```
ImageClassification - [F:\BigDataAnalytics\Tutorials\lab 6 screens\ImageClassification] - [imageclassification] - ...src\main\scala\ImageClassification_DT.scala - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ImageClassification src main scala ImageClassification_DT.scala
Project ImageClassification_DT.scala ImageClassification_RF.scala
ImageClassification [imageclassification] F:\BigDataAr
  .idea
  data
    test
      testfeatures.txt
    train
      features.txt
  project [imageclassification-build] sources root
  src
    main
      java
      resources
      scala
        ImageClassification_DT
        ImageClassification_RF
      scala-2.11
    test
    target
    build.sbt
  External Libraries
  1 import org.apache.log4j.{Level, Logger}
  2 import org.apache.spark.mllib.evaluation.MulticlassMetrics
  3 import org.apache.spark.mllib.linalg.Vectors
  4 import org.apache.spark.mllib.regression.LabeledPoint
  5 import org.apache.spark.mllib.tree.DecisionTree
  6 import org.apache.spark.rdd.RDD
  7 import org.apache.spark.{SparkConf, SparkContext}
  8
  9 object ImageClassification_DT {
 10   def main(args: Array[String]) {
 11     val IMAGE_CATEGORIES = Array("elephant", "gazelle", "rhino", "zebra")
 12     System.setProperty("hadoop.home.dir", "F:\\winutils")
 13     // Turn off Info Logger for Consolexxx
 14     Logger.getLogger("org").setLevel(Level.OFF)
 15     Logger.getLogger("akka").setLevel(Level.OFF)
 16     val sparkConf = new SparkConf().setAppName("ImageClassification").setMaster("local[*]")
 17     val sc = new SparkContext(sparkConf)
 18     val train = sc.textFile("data/train")
 19     val test = sc.textFile("data/test")
 20     val parsedData = train.map { line =>
 21       val parts = line.split(',')
 22       LabeledPoint(parts(0).toDouble, Vectors.dense(parts(1).split(' ').map(_.toDouble)))
 23     }
 24     val testDatal = test.map(line => {
 25       val parts = line.split(',')
 26       LabeledPoint(parts(0).toDouble, Vectors.dense(parts(1).split(' ').map(_.toDouble)))
 27     })
 28
 29     val trainingData = parsedData
 30
 31
 32
  Run ImageClassification_RF
  Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
```

```
ImageClassification - [F:\BigDataAnalytics\Tutorials\lab 6 screens\ImageClassification] - [imageclassification] - ...src\main\scala\ImageClassification_DT.scala - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ImageClassification src main scala ImageClassification_DT.scala
Project ImageClassification_DT.scala ImageClassification_RF.scala
ImageClassification [imageclassification] F:\BigDataAr
  .idea
  data
    test
      testfeatures.txt
    train
      features.txt
  project [imageclassification-build] sources root
  src
    main
      java
      resources
      scala
        ImageClassification_DT
        ImageClassification_RF
      scala-2.11
    test
    target
    build.sbt
  External Libraries
  33 val numClasses = 4
  34 val categoricalFeaturesInfo = Map[Int, Int]()
  35 val impurity = "gini"
  36 val maxDepth = 5
  37 val maxBins = 32
  38
  39 val model = DecisionTree.trainClassifier(trainingData, numClasses, categoricalFeaturesInfo,
  40   impurity, maxDepth, maxBins)
  41
  42 val classifyf1 = testDatal.map { line =>
  43   val prediction = model.predict(line.features)
  44   (line.label, prediction)
  45 }
  46
  47 val prediction1 = classifyf1.groupBy(_._1).map(f => {
  48   var fuzzy_Pred = Array(0, 0, 0)
  49   f._2.foreach(ff => {
  50     fuzzy_Pred(ff._2.toInt) += 1
  51   })
  52   var count = 0.0
  53   fuzzy_Pred.foreach(f => {
  54     count += f
  55   })
  56   var i = -1
  57   var maxIndex = 4
  58   val max = fuzzy_Pred.max
  59   val pp = fuzzy_Pred.map(f => {
  60     val p = f * 100 / count
  61     i = i + 1
  62     if(f == max)
  63       maxIndex=i
  64     (i, p)
  65   })
  66
  Run ImageClassification_RF
  Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
```



ImageClassification - [F:\BigDataAnalytics\Tutorials\lab 6 screens\imageClassification] - [imageClassification] - ...src\main\scala\imageClassification\_DT.scala - IntelliJ IDEA 2016.3.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

ImageClassification src / main scala ImageClassification\_DT.scala

Project ImageClassification\_DT.scala ImageClassification\_RF.scala

ImageClassification [imageClassification] F:\BigDataAr

```
59 -----
60 val pp = fuzzy.Fred.map(f => {
61     val p = f * 100 / count
62     i = i + 1
63     if (f == max)
64         maxIndex=i
65     (i, p)
66 })
67 (f._1, pp, maxIndex)
68 })
69 prediction1.foreach(f => {
70     println("\n\n" + f._1 + " : " + f._2.mkString("\n"))
71 })
72 val y: RDD[(Double, Double)] = prediction1.map(f => {
73     (f._3.toDouble, f._1)
74 })
75 y.collect().foreach(println(_))
76
77 val metrics = new MulticlassMetrics(y)
78
79 println("Accuracy:" + metrics.accuracy)
80
81 println("Confusion Matrix:")
82 println(metrics.confusionMatrix)
83 }
84 }
85
```

Run ImageClassification\_RF

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

ImageClassification - [F:\BigDataAnalytics\Tutorials\lab 6 screens\imageClassification] - [imageClassification] - ...src\main\scala\imageClassification\_RF.scala - IntelliJ IDEA 2016.3.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

ImageClassification src / main scala ImageClassification\_RF.scala

Project ImageClassification\_DT.scala ImageClassification\_RF.scala

ImageClassification [imageClassification] F:\BigDataAr

```
1 import org.apache.log4j.{Level, Logger}
2 import org.apache.spark.mllib.evaluation.MulticlassMetrics
3 import org.apache.spark.mllib.linalg.Vectors
4 import org.apache.spark.mllib.regression.LabeledPoint
5 import org.apache.spark.mllib.tree.{DecisionTree, RandomForest}
6 import org.apache.spark.{SparkConf, SparkContext}
7
8 object ImageClassification_RF {
9     def main(args: Array[String]) {
10         val IMAGE_CATEGORIES = Array("elephant", "gazelle", "rhino", "zebra")
11         System.setProperty("hadoop.home.dir", "E:\\Windows\\Hadoop")// Turn off Info Logger for Consolexxxx
12         Logger.getLogger("org").setLevel(Level.OFF)
13         Logger.getLogger("akka").setLevel(Level.OFF)
14         val sparkConf = new SparkConf().setAppName("ImageClassification").setMaster("local[*]")
15         val sc = new SparkContext(sparkConf)
16         val train = sc.textFile("data/train")
17         val test = sc.textFile("data/test")
18         val parsedData = train.map { line =>
19             val parts = line.split(',')
20             LabeledPoint(parts(0).toDouble, Vectors.dense(parts(1).split(' ').map(_.toDouble)))
21         }
22         val testDatat = test.map(line => {
23             val parts = line.split(',')
24             LabeledPoint(parts(0).toDouble, Vectors.dense(parts(1).split(' ').map(_.toDouble)))
25         })
26
27         val trainingData = parsedData
28
29         val numClasses = 4
30         val categoricalFeaturesInfo = Map[Int, Int]()
31
32
```

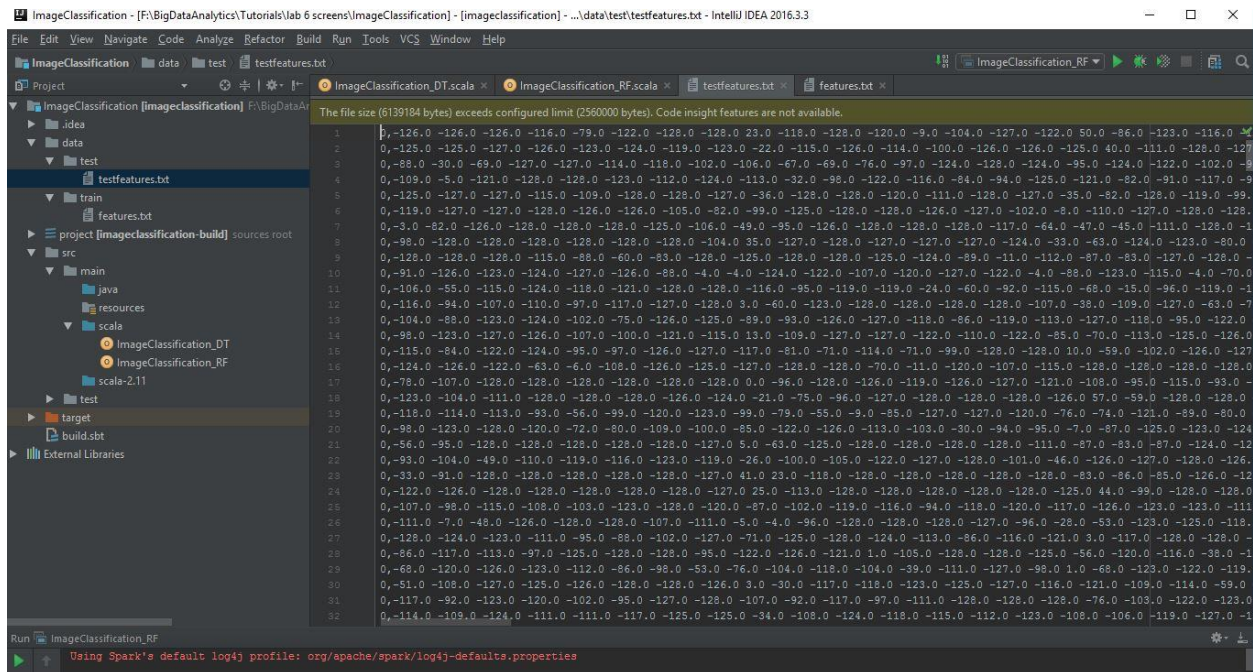
Run ImageClassification\_RF

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

```
ImageClassification - [F:\BigDataAnalytics\Tutorials\lab 6 screens\ImageClassification] - [imageclassification] - ...src\main\scala\ImageClassification_RF.scala - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ImageClassification src main scala ImageClassification_RF.scala
Project ImageClassification_DT.scala ImageClassification_RF.scala
ImageClassification [imageclassification] F:\BigDataAr
.idea
  data
    test
      testfeatures.txt
    train
      features.txt
  project [imageclassification-build] sources root
  src
    main
      java
      resources
      scala
        ImageClassification_DT
        ImageClassification_RF
        scala-2.11
      test
    target
    build.sbt
  External Libraries
Run ImageClassification_RF
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
```

```
ImageClassification - [F:\BigDataAnalytics\Tutorials\lab 6 screens\ImageClassification] - [imageclassification] - ...src\main\scala\ImageClassification_RF.scala - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ImageClassification src main scala ImageClassification_RF.scala
Project ImageClassification_DT.scala ImageClassification_RF.scala
ImageClassification [imageclassification] F:\BigDataAr
.idea
  data
    test
      testfeatures.txt
    train
      features.txt
  project [imageclassification-build] sources root
  src
    main
      java
      resources
      scala
        ImageClassification_DT
        ImageClassification_RF
        scala-2.11
      test
    target
    build.sbt
  External Libraries
Run ImageClassification_RF
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
```





The accuracy came out as 25%