



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
Ingeniería en Inteligencia Artificial



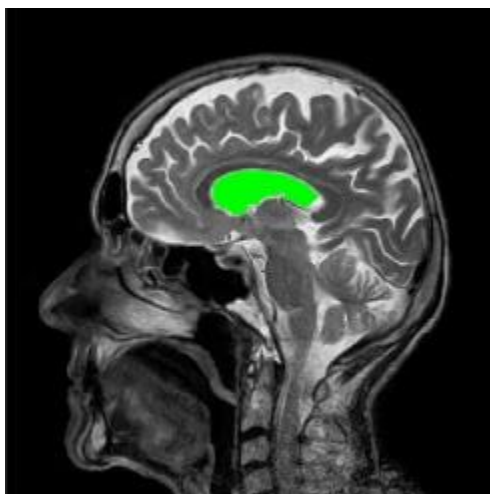
Practica 5: Segmentación por crecimiento de regiones

Nombre del alumno: Torres López Marco Antonio

Nombre del profesor: Saul de la O Torres

Grupo: 5BM1

Unidad de aprendizaje: Visión artificial



Desarrollo

Para la segmentación por crecimiento de regiones se realizaron ciertas modificaciones, primero se le agrego una semilla aleatoria dentro de la imagen:

```
1
2 Point semillaAleatoria = new Point((int) (Math.random()
3 * (newImg.width()-1)),
4 * (newImg.height()-1));
```

Agregamos unos parámetros para la búsqueda de la segmentación (como una variable para las vecindades) y agregamos la semilla dentro de la imagen:

```
1 boolean growing = true;
2     int width = src.width(),
3       height = src.height();
4     Region R = new Region();
5     List<Pixel> neighborhood = new ArrayList<>();
6     List<Pixel> tmpR;
7
8     Imgproc.cvtColor(src, src, Imgproc.COLOR_BGR2G
9 RAY);
10    mostrarImagen("Grises", src);
11
12    // Toma la semilla de la imagen
13    double[] pixelD = src.get((int) randSeedPos.y,
14 (int) randSeedPos.x);
15    int pixValue = (int) pixelD[0];
16    double r = pixValue;
17    Referencia r -> pixColor
18
19    // 1. Iniciamos la region R con la semilla p
20    Pixel pixel = new Pixel(pixValue, randSeedPo
21 s);
22    pixel.adicionado = true;
23    R.add(pixel);
```

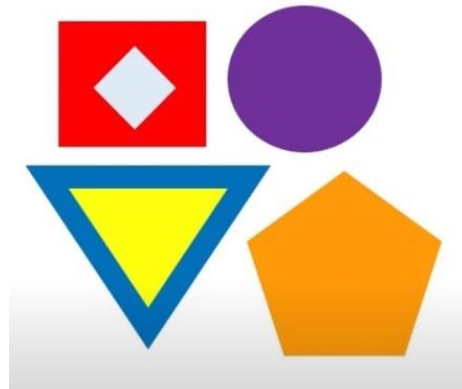
Realizamos la búsqueda de los pixeles vecinos que se encuentran dentro de la semilla:

```
1
2 // Pixel arriba
3 nbr = new Point(IncrementVector(new double[]{x, y}, new double[]{0, -1}, new double[]{0, 0}, new double[]{width-1, height-1}));
4 if (!tmpPixel.equals(nbr))
5     neighborhood.add(new Pixel(src.get((int) nbr.y, (int) nbr.x)[0], nbr));
6
7 // Pixel abajo
8 nbr = new Point(IncrementVector(new double[]{x, y}, new double[]{0, 1}, new double[]{0, 0}, new double[]{width-1, height-1}));
9 if (!tmpPixel.equals(nbr))
10     neighborhood.add(new Pixel(src.get((int) nbr.y, (int) nbr.x)[0], nbr));
11
12 // Right Pixel derecho
13 nbr = new Point(IncrementVector(new double[]{x, y}, new double[]{1, 0}, new double[]{0, 0}, new double[]{width-1, height-1}));
14 if (!tmpPixel.equals(nbr))
15     neighborhood.add(new Pixel(src.get((int) nbr.y, (int) nbr.x)[0], nbr));
16
17 // Pixel izquierdo
18 nbr = new Point(IncrementVector(new double[]{x, y}, new double[]{-1, 0}, new double[]{0, 0}, new double[]{width-1, height-1}));
19 if (!tmpPixel.equals(nbr))
20     neighborhood.add(new Pixel(src.get((int) nbr.y, (int) nbr.x)[0], nbr));
21
22 tmpPixel.adicionado = false;
```

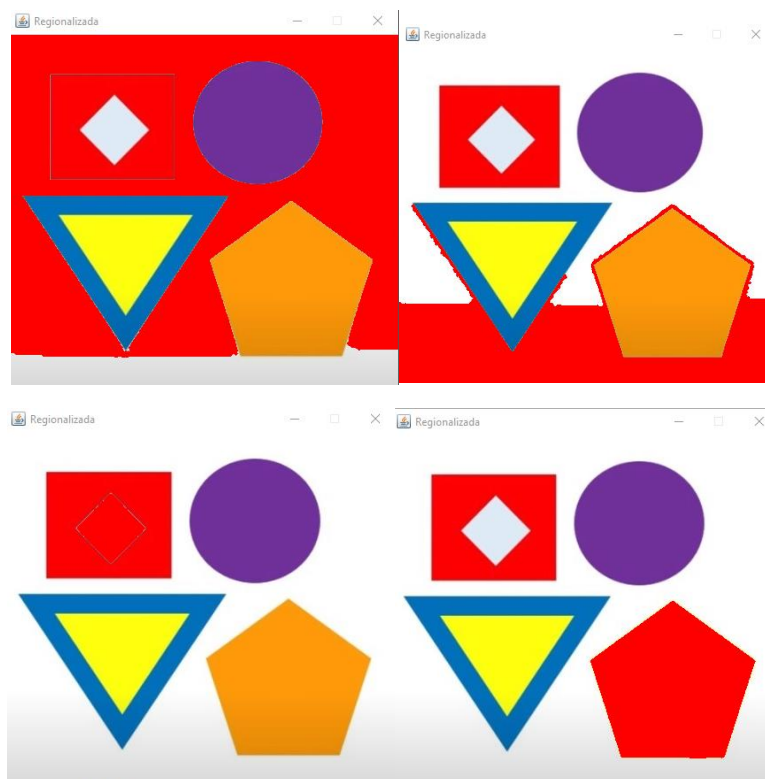
Resultados

Prueba 1:

Imagen original



Regionalizada:

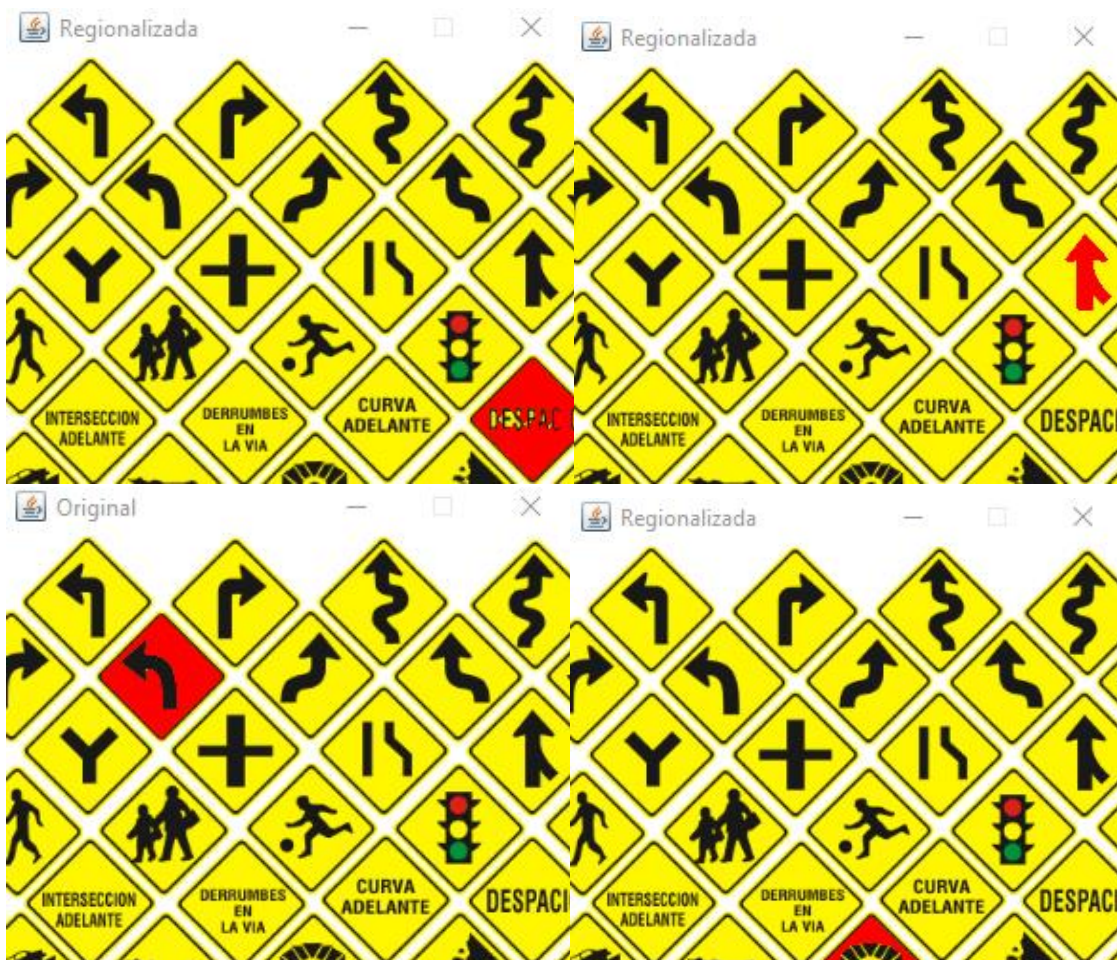


Prueba 2:

Original



Regionalizada



Conclusiones

Podemos observar que realiza bien el trabajo de segmentación por regiones, además este método es uno de los más eficaces y fáciles de implementar, aunque puede demorar un poco si la región cuenta con demasiados píxeles.

Es importante la segmentación de regiones ya que se usa tanto para localizar objetos como para encontrar sus bordes dentro de una imagen.