INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Ingeniería en Inteligencia Artificial

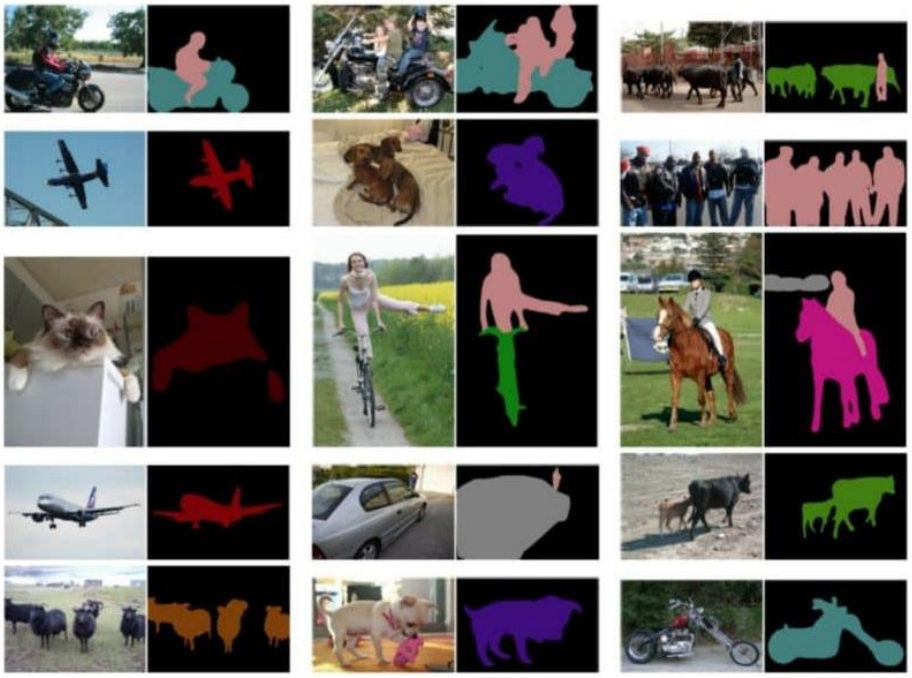
Practica 7: Segmentación por crecimiento de regio en imágenes a color

Nombre del alumno: Torres López Marco Antonio

Nombre del profesor: Saul de la O Torres

Grupo: 5BM1

Unidad de aprendizaje: Visión artificial



Desarrollo

El código implementado es similar al de la partica 5 (segmentación por regiones) con algunas modificaciones:

Creamos una semilla aleatoria dentro de la imagen o también puede escribir la semilla que desee (deberá primero verificar su tamaño de imagen para evitar errores)

```
Point semillaAleatoria = new Point(
        (int) (Math.random() * (img.width()-1)),
        (int) (Math.random() * (img.height()-1)));
```

```
Point semillaAleatoria = new Point(430,350);
Point semillaAleatoria = new Point(430,150);
Point semillaAleatoria = new Point(250,370);
```

Anteriormente se manejábamos 1 canal de grises para poder procesar la segmentación, pero en esta ocasión para manejar los 3 canales de color (RGB) eliminaremos la conversión BRG2GRAY:

```
Imgproc.cvtColor(src, src, Imgproc.COLOR_BGR2GRAY);
        motrarImagen("Grises", src);
```

Dentro del proceso de segmentación no cambiara nada, buscara los pixeles vecinos para determinar si estos pertenecen a la región dentro de los 3 canales:

```
while (crecimineto) {
            crecimineto = false;

            // 2. tomamos los vecinos de R
            tmpR = R.getR();
            for (Pixel tmpPixel : tmpR) {
                if (tmpPixel.adicionado) {
                    int x = tmpPixel.getX(),
                        y = tmpPixel.getY();
                    Point nbr;

                    // Pixel de arriba
                    nbr = new Point(incrementVector(new double[]{x, y}, new double[]{0, -1}, new double[]{0, 0}, new double[]{width-1, height-1}));
                    if (!tmpPixel.equals(nbr))
                        Vecinos.add(new Pixel(src.get((int) nbr.y, (int) nbr.x), nbr));

                    // Pixel de abajo
                    nbr = new Point(incrementVector(new double[]{x, y}, new double[]{0, 1}, new double[]{0, 0}, new double[]{width-1, height-1}));
                    if (!tmpPixel.equals(nbr))
                        Vecinos.add(new Pixel(src.get((int) nbr.y, (int) nbr.x), nbr));

                    // Pixel Derecho
                    nbr = new Point(incrementVector(new double[]{x, y}, new double[]{1, 0}, new double[]{0, 0}, new double[]{width-1, height-1}));
                    if (!tmpPixel.equals(nbr))
                        Vecinos.add(new Pixel(src.get((int) nbr.y, (int) nbr.x), nbr));

                    // Pixel Izquierdo
                    nbr = new Point(incrementVector(new double[]{x, y}, new double[]{-1, 0}, new double[]{0, 0}, new double[]{width-1, height-1}));
                    if (!tmpPixel.equals(nbr))
                        Vecinos.add(new Pixel(src.get((int) nbr.y, (int) nbr.x), nbr));

                    tmpPixel.adicionado = false;
                }
            }

```

Resultados

Imagen original:

Imagen regionalizada en los 3 canales:

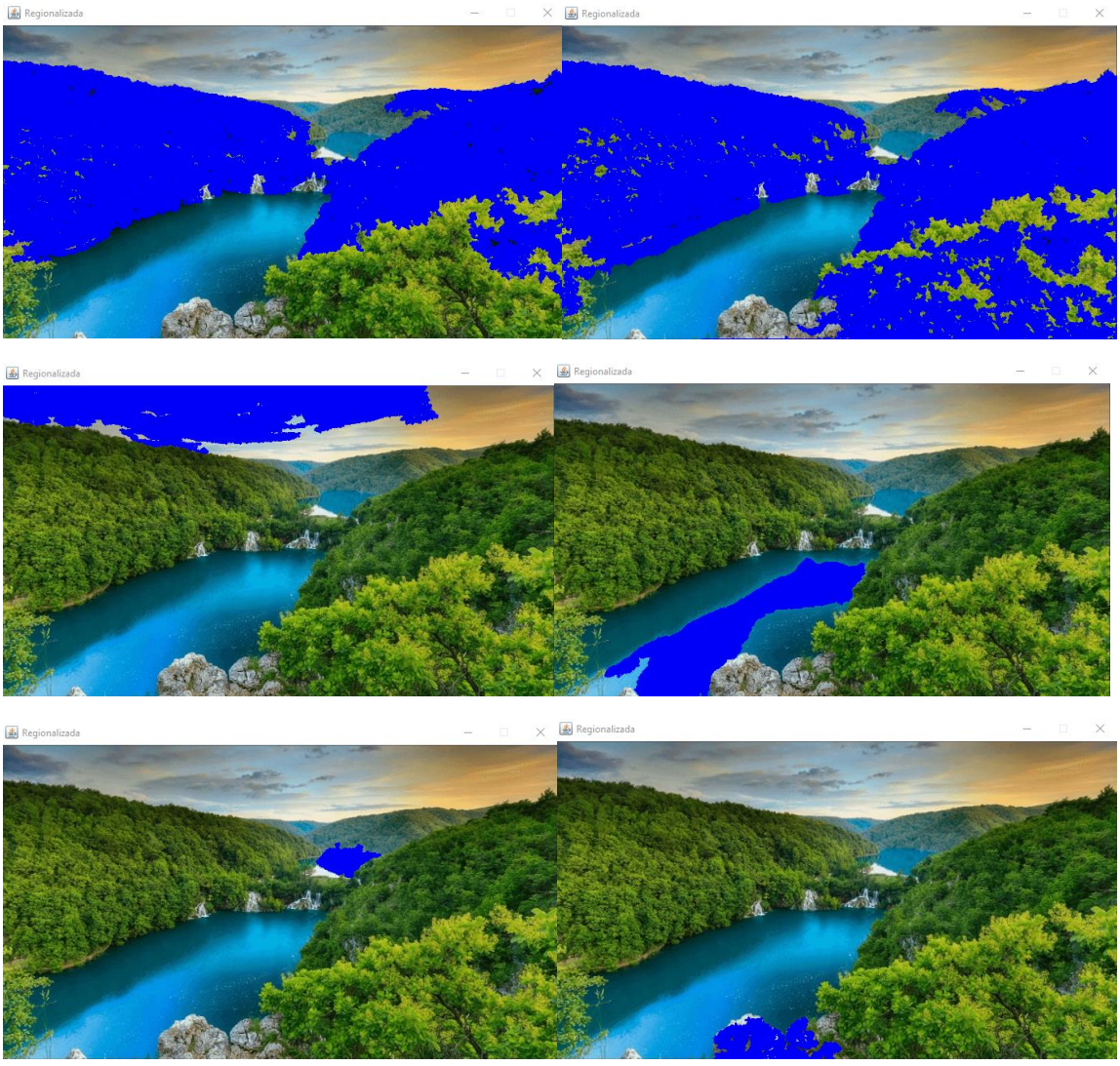


Imagen original:

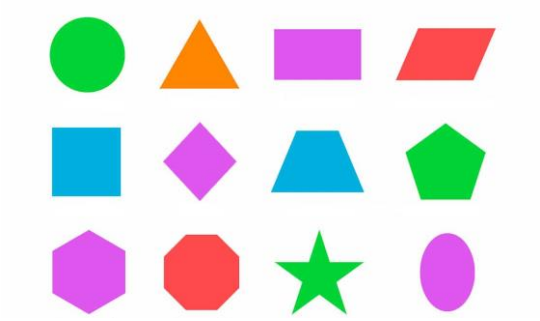
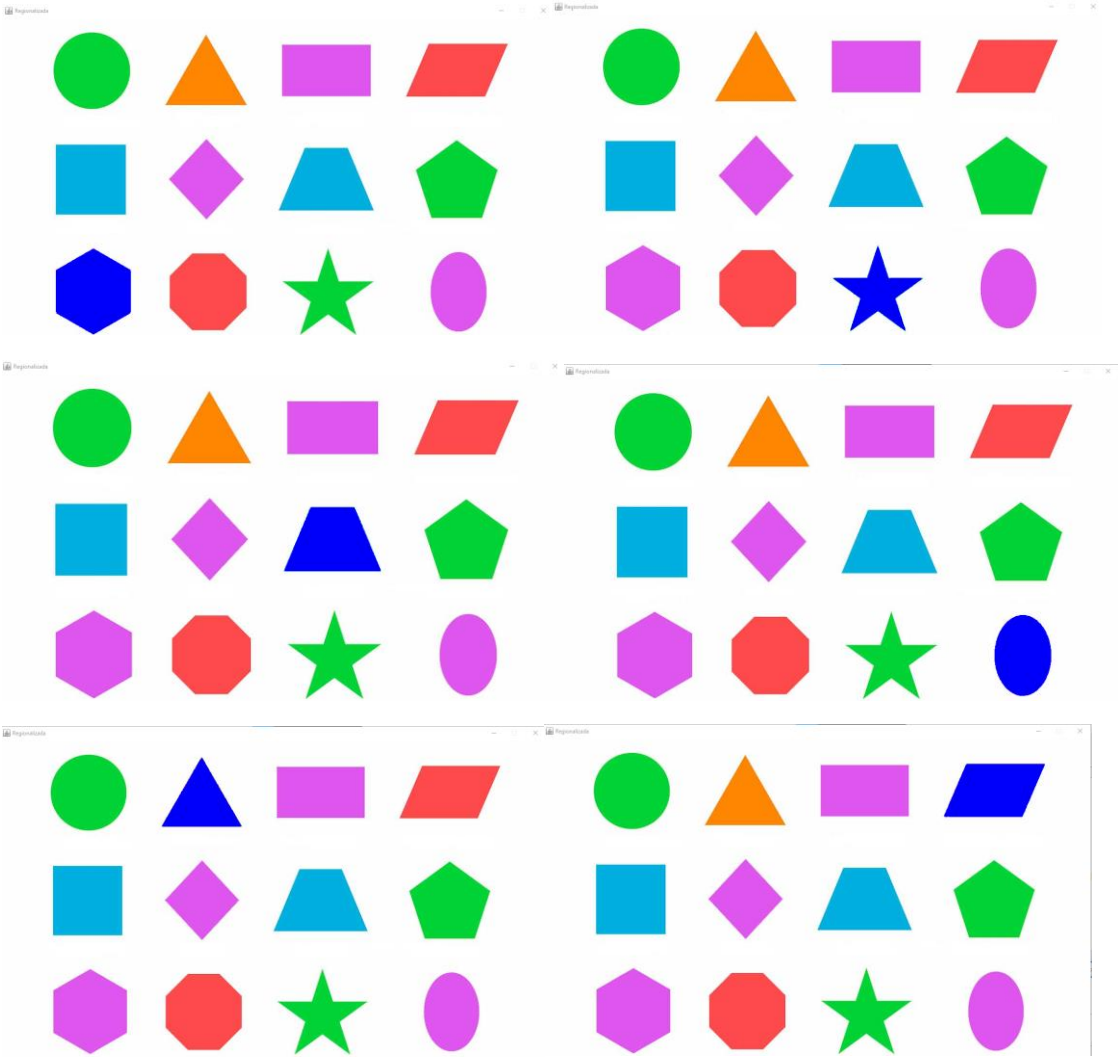


Imagen regionalizada en los 3 canales:



Conclusión

Dentro de esta práctica se logró la segmentación de regiones con los 3 canales, algo que se puede observar en el momento de ejecución es que el tiempo de ejecución incrementa considerablemente ya que se procesan los 3 canales de color llegando a tardarse de medio minuto hasta 10 minutos para arrojar la imagen regionalizada, aunque otorga mejores resultados que cuando se trabajo con solo 1 canal.