

Técnicas de Procesamiento de Imágenes, para la detección o diagnóstico de enfermedades en Imágenes del sector Agrícola.

Autor: Fernando R. Aramayo

Universidad Nacional de Jujuy, Facultad de Ingeniería. Universidad Nacional de Jujuy,
Facultad de ciencias Agrarias
fernando.ruben.aramayo@gmail.com

Ingeniero Informática (Universidad Nacional de Jujuy), Licenciado en Sistemas (Universidad Nacional de Jujuy), Profesor Adjunto en la Facultad de Ciencias Agrarias de la UNJU.

Resumen

El presente trabajo propone una descripción/clasificación sobre el manejo de imágenes y el desarrollo de un prototipo de Sistema que logre identificar enfermedades en las imágenes que provee el usuario al Sistema. La clasificación de imágenes debe ser interpretada como una clasificación de píxeles, un proceso en el cual cada uno de los píxeles en una imagen es asignado a una clase o categoría. Este proceso puede ser utilizado para realizar preguntas tales como ¿Que píxeles en la imagen corresponden a un bosque, arena o en este caso a una enfermedad determinada? pero no preguntas como ¿Esta imagen contiene un Auto? Las personas piensan acerca del contenido de una imagen, sin tener en cuenta, cuan complejo es el procesamiento del sistema visual y del cerebro de un ser humano para poder interpretar las imágenes.



Comparar el contenido de una imagen no es una tarea sencilla, pero se la puede simplificar reduciendo el problema a uno más simple (dividiéndolo en problemas más sencillos), en vez de preguntarse ¿Los objetos de estas imágenes son iguales? se podría preguntar ¿Alguna región de esta imagen es similar, en algún aspecto, a algunas regiones de la otra imagen? Ahora se podría tratar el problema con técnicas de procesamiento de imágenes. Para que una imagen pueda ser manipulada primero se debe procesar la misma, los pasos realizados son:

- Transformar sus colores a escala de grises
- Convertir la imagen a una imagen binaria
- Resaltar sus bordes
- Obtener sus contornos (coordenadas (x, y) de la imagen binaria)

Palabras clave

Procesamiento de imágenes, diagnóstico por imágenes, clasificación de píxeles, diagnóstico de enfermedades.

Image Processing Techniques, for the detection or diagnosis of diseases in Images of the Agricultural sector.

Abstract

The present work proposes a description / classification on the management of images and the development of a prototype of System that manages to identify diseases in the images that provide the user to the System. Image classification should be interpreted as a pixel classification, a process in which each pixel in an image is assigned to a class or category. This process can be used to ask questions such as ¿What pixels in the image correspond to a forest, sand or in this case a certain disease? But not questions like ¿This image contain a Car?

People think about the content of an image, regardless of how complex is the processing of the visual system and the brain of a human being to be able to interpret the images.

Comparing the content of an image is not a simple task, but it can be simplified by reducing the problem to a simpler one (dividing it into simpler problems), instead of asking yourself ¿The objects of these images are the same? you might ask ¿Is any region of this image similar in some aspect to some regions of the other image? The problem could now be dealt with image processing techniques. In order for an image to be manipulated first, it must be

processed, the steps performed are:

- *Transform your colors to grayscale*
- *Convert the image to a binary image*
- *Highlight your edges*
- *Get its contours (coordinates (x, y) of the binary image)*

Key Words

Image processing, diagnosis by images, classification of pixels, diagnosis of diseases

Introducción

La clasificación de imágenes debe ser interpretada como una clasificación de píxeles, un proceso en el cual cada uno de los píxeles en una imagen es asignado a una clase o categoría. Este proceso puede ser utilizado para preguntar cosas como ¿Que píxeles en la imagen corresponden a un bosque, arena o en nuestro caso a una enfermedad determinada? pero no preguntas como ¿Esta imagen contiene un Auto?

Las personas piensan acerca del contenido de una imagen, sin tener en cuenta, cuan complejo es el procesamiento del sistema visual y del cerebro de un ser humano para poder interpretar las imágenes. Un experimento sencillo, es mostrar la imagen 1 a diferentes personas y preguntarles ¿Qué hay en esta imagen?, probablemente la mayoría va a describir al animal (esto puede ser debido a que es considerado el “sujeto” en la imagen), pero la descripción en si puede variar por completo (si se tiene en cuenta otros objetos relevantes en la imagen) y puede variar según los detalles que se tengan en cuenta (conjunto de información relevante acerca de cada objeto). Si ni siquiera los seres humanos pueden ponerse de acuerdo con una simple y concisa descripción acerca de que hay en la imagen, como se puede esperar que una computadora haga lo mismo con un simple algoritmo.



Imagen 1: Interpretación de una Imagen



El problema es que las computadoras no pueden “ver” una imagen y “comparar” con otra para determinar si son iguales, donde con “iguales” se refiere al contenido de la imagen. Sin embargo los avances en el procesamiento de imágenes pueden volver algún día esta tarea factible, bajo condiciones controladas, pero no con un simple algoritmo.

Objetivos

- Objetivos Generales: Desarrollar un prototipo de Sistema que sea capaz de procesar las imágenes provistas por los usuarios y en base a técnicas de procesamiento de Imágenes detectar enfermedades en las mismas.
- Objetivos específicos:
 1. Aplicar las técnicas de procesamiento de imágenes para determinar las áreas en las cuales se realizarán los controles en busca de enfermedades.
 2. Clasificar los pixeles correspondientes a una enfermedad y su posterior comparación con las áreas de la imagen a diagnosticar.

Comparar imágenes en cuanto a su contenido

Si se desea comparar imágenes, se enfrentara una tarea muy difícil que no puede ser resuelta con algoritmos simples. Posiblemente se deba procesar la imagen para obtener un conjunto de características o rasgos que puedan ser consideradas como objetos. Comparar el contenido de una imagen no es una tarea sencilla, pero se la puede simplificar si decidimos reducir el problema a uno más simple (dividiéndolo en problemas más sencillos), en vez de preguntarnos ¿Los objetos de estas imágenes son iguales? nos podríamos preguntar ¿Alguna región de esta imagen es similar, en algún aspecto, a algunas regiones de la otra imagen? En base a ello, ahora se podría tratar el problema con algunas técnicas de procesamiento de imágenes. El proceso para extraer objetos en el cerebro humano es muy, muy complejo y flexible, siendo capaz de reconocer no solo el objeto en sí, sino también su comportamiento general en la escena, como así también a la categoría a la cual pertenece. Es muy sencillo para los seres humanos observar la Imagen 1 que contiene una “ardilla mirando hacia arriba” o “un pequeño roedor colgando de una reja”, pero esta tarea no puede ser realizada con solo algoritmos de procesamiento de imágenes.

Herramientas

Para la implementación del sistema se trabajó sobre la especificación de java JAI (Java Advance Imaging). El API de JAI extiende de la plataforma de Java 2 y permite aplicar mediante applets y aplicaciones de escritorio un alto grado de “performance” en el



procesamiento de imágenes. JAI maneja operadores que son de gran utilidad para el procesamiento de imágenes, entre las cuales podemos mencionar los siguientes: operadores de área, operadores de color, operadores geométricos, operadores de frecuencia, operadores estadísticos, operadores de extracción de bordes, etc.

Entre algunas de las principales características de JAI se puede mencionar las siguientes:

- Un API orientado a objetos.
- Imágenes distribuidas.
- Flexible y extensible.
- Poderoso.
- Alto performance.

Preprocesamiento

Para que una imagen pueda ser manipulada más fácilmente es necesario que antes sea preprocesada. Los procesos que se utilizaron sobre la imagen para lograr los objetivos fueron los siguientes:

- Transformar sus colores a escala de grises.
- Convertir la imagen a una imagen binaria.
- Resaltar sus bordes.
- Y finalmente obtener sus contornos, esto es, obtener sus coordenadas (x, y) de la imagen binaria.

Transformar sus colores a escala de grises

El proceso que se implementó fue el siguiente: En una primera instancia se creó una matriz con los siguientes valores que se especifican a continuación, los cuales se utilizarán para preprocesar la imagen y transformarla en una imagen en escala de grises.

$$\text{Matriz} = \{ \{0.114D, 0.587D, 0.299D, 0.0D\} \}$$

Luego se deben realizar las operaciones lógicas necesarias, para que los valores de la "Matriz" sean añadidos a la imagen mediante el operador "bandcombine" (Manual de Usuario JAI 2017) que maneja JAI en sus "operadores de punto". Una vez realizado el paso anterior, la imagen es transformada a escala de grises, principalmente se debe controlar que el número de Bandas sea igual a 3 que hacen referencia a los colores RGB (red, green, blue – rojo, verde, azul), en caso de que la imagen lo necesite aplicar los algoritmos necesarios para transformarla en una imagen RGB. El pseudocódigo para este subproceso es el siguiente:

En las Imagen 2 e Imagen 3 se muestran ejemplos de este proceso.

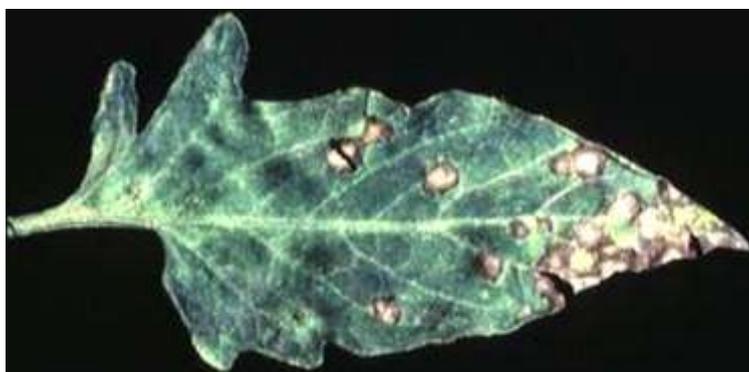


Imagen 2: Imagen a analizar

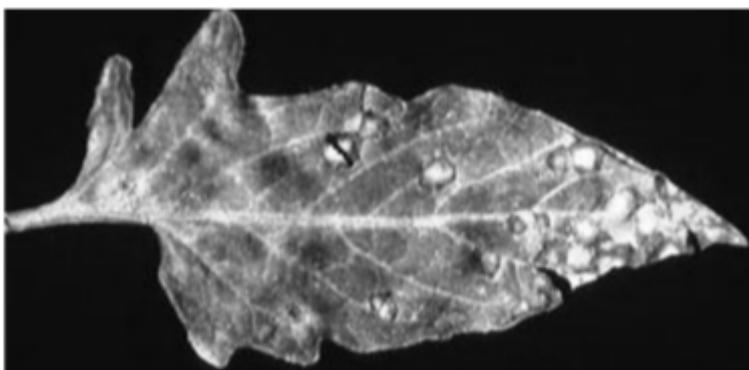


Imagen 3: Imagen en escala de Grises

Convertir la imagen a una imagen binaria

El proceso que se debe realizar tras haber aplicado la transformación a escala de grises es el de umbralizar (Cattaneo, Larcher, Ruggeri, Herrera, BIASONI, 2011) la imagen, es decir, convertir una imagen a una imagen binaria, donde únicamente tenga dos valores, 0's y 1's, ceros para el fondo y unos para los bordes de la imagen o viceversa. Esto se hace mediante un thresholding, también conocido como realce binario de contraste, básicamente a lo que se refiere es definir los límites medios de los objetos que aparece en un fondo contrastado. Al igual que el proceso anterior, JAI cuenta con un operador en específico que se encarga de hacer esto, este operador se denomina del mismo nombre thresholding (Manual de Usuario JAI 2017).

La operación toma una imagen y mapea todos los píxeles de esta imagen que caigan dentro de un rango de valores específicos. El rango es especificado por un valor alto y un valor bajo. Para definir estos valores de forma óptima se puede hacer uso del histograma (Cattaneo, Larcher, Ruggeri, Herrera, BIASONI, 2011) de la imagen, en especial con el operador extrema.

En las Imagen 2 e Imagen 3 se pueden visualizar los ejemplos realizados con el proceso descrito.



Imagen 4: Imagen Binaria

Resaltar los bordes

Una vez realizado el proceso de umbralizado en la imagen, el proceso siguiente es resaltar sus bordes para que sea más fácil extraer sus propiedades y posteriormente sea manipulada de manera eficiente. Para lograr dicho objetivo existen muchos métodos, los cuales son básicamente mascarar que resaltan los bordes de las imágenes, esto con el fin de encontrar el contorno de una imagen de forma más fácil. Un borde de una imagen usualmente ocurre en los límites de un objeto dentro de la imagen, donde la amplitud del objeto cambia drásticamente a la amplitud del fondo u otro objeto. Para poder aplicar los diversos métodos de reconocimiento de bordes debemos tener conceptos acerca de diferentes métodos matemáticos.

Gradiente de una imagen: El gradiente de una imagen $f(x,y)$ se define como un vector bidimensional dado por la Ecuación 1, el cual es un vector perpendicular al borde de la imagen que se desea analizar.

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix}$$

Ecuación 1

Sus componentes miden la rapidez en que los valores de los pixeles cambian en la distancia y en las direcciones x e y. El vector G apunta en la dirección de máxima variación de f en el punto (x,y) por unidad de distancia, con la magnitud y dirección dadas por las Ecuación 2 y Ecuación 3.

Para detectar la presencia de una discontinuidad en el gradiente, se debe calcular el cambio en el gradiente en el punto (x,y). Esto se puede hacer referenciando la medida aportada por la magnitud del gradiente y su dirección.

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Ecuación 2

$$\theta(x, y) = \text{Tg}^{-1} \frac{G_y}{G_x}$$

Ecuación 3

En una práctica habitual es aconsejable aproximar la magnitud del gradiente con valores absolutos, representados por la Ecuación 4.

$$|G| = |G_x| + |G_y|$$

Ecuación 4

Se decide si un punto es de borde, si la magnitud del gradiente supera o no un determinado umbral, se debe ajustar dicho umbral para que el resultado de la extracción de bordes sea el mismo tanto si se calcula la magnitud del gradiente mediante Ecuación 2 y Ecuación 3 como si se hace por Ecuación 4, esta última ecuación resulta mucho más fácil de implementar. Para calcular la derivada en la Ecuación 1 se puede utilizar las diferencias de primer orden entre dos píxeles adyacentes, como se ejemplifica en la Ecuación 5 .

$$G_x = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} \quad G_y = \frac{f(y + \Delta y) - f(y - \Delta y)}{2\Delta y}$$

Ecuación 5

Esta es la forma más elemental de obtener el gradiente en un punto. La magnitud del gradiente puede tomar cualquier valor real y el ángulo también cualquier valor entre 0º y 360º. No obstante se estudiarán otros operadores para implementar el concepto de derivada en un punto y que consideran una vecindad de 3x3 entornos al punto, podemos obtener una imagen binaria utilizando la Ecuación 6.

$$G(x, y) = \begin{cases} 1 & \text{si } G [f(x, y)] > T \\ 0 & \text{si } G [f(x, y)] \leq T \end{cases}$$

Ecuación 6

Donde T es un valor de umbral no negativo (en este caso T=30). Solo los píxeles de borde cuyo gradiente excedan el valor de T se consideran importantes. Así la ecuación anterior se puede ver como un procedimiento que extrae solo aquellos píxeles caracterizados por

transiciones de intensidad significativas (dependiendo de T).

Método de Sobel (Ferraro, Hector P, 2007): La operación de Sobel extrae todos los bordes en una imagen sin importar la dirección. La imagen resultante aparece como un contorno omnidireccional de los objetos en la imagen original. Se destacan las regiones constantes de brillo. Como se indicó en párrafos anteriores los valores de G_x y G_y de la Ecuación 1 pueden implementarse por convolucion de la imagen con las máscaras 3x3 dadas en la Ecuación 7 conocidas como mascarar de Sobel.

$$\begin{matrix} \begin{bmatrix} Z_1 & Z_2 & Z_3 \\ Z_4 & Z_5 & Z_6 \\ Z_7 & Z_8 & Z_9 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \\ \text{(a)} & \text{(b)} & \text{(c)} \end{matrix}$$

Ecuación 7

- (a) Región de la imagen de dimensión 3x3
- (b) Mascara usada para obtener G_x en el punto central de la región 3x3
- (c) Mascara usada para obtener G_y en el mismo punto

Los operadores gradiente en general tienen el efecto de magnificar el ruido (Ferraro, Hector P, 2007) subyacente en la imagen, tanto los operadores de Sobel como el resto de los operadores de vecindad, tienen la propiedad añadida de suavizar la imagen, eliminando parte del ruido y por consiguiente, minimiza la aparición de falsos bordes. A partir de la Ecuación 7 las derivadas basadas en los operadores de Sobel se calculan mediante la Ecuación 8.

$$\begin{aligned} G_x &= (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7) \\ G_y &= (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3) \end{aligned}$$

Ecuación 8

Donde los distintos valores de Z en la región de la Ecuación 7 (a) son los niveles de gris de los píxeles solapados por las máscaras en cualquier localización de la imagen. Para obtener los valores de los componentes del vector gradiente en el punto definido por el píxel central de la región se utiliza la Ecuación 8 con lo que la magnitud y el ángulo se pueden obtener a partir de la Ecuación 2, Ecuación 3, Ecuación 4, es decir obtenemos el valor del gradiente en dicho punto. Para obtener el siguiente valor, las máscaras se mueven a la siguiente posición del nuevo píxel y se repite el proceso, después de haber barrido todas las posibles



posiciones, el resultado es una imagen gradiente. Una vez que se ha obtenido la magnitud del gradiente, se puede decidir si un determinado punto es de borde o no aplicando la Ecuación 6 obteniendo así una imagen binaria como resultado.

$$\begin{bmatrix} 0_0 & 8_0 & 8_0 & 8 & 8 & 8 \\ 1_0 & 8_0 & 9_0 & 7 & 6 & 8 \\ 2_0 & 3_0 & 4_0 & 8 & 8 & 8 \\ 2 & 2 & 2 & 8 & 8 & 8 \\ 2 & 2 & 2 & 2 & 8 & 8 \end{bmatrix}$$

Ecuación 9

Realicemos un ejemplo práctico sencillo para aclarar los conceptos anteriores, supongamos la siguiente imagen dada en la Ecuación 9. Se pretende calcular el gradiente en el píxel marcado con el punto negro. La región de la imagen que interviene está marcada con los puntos en blanco. Para calcular el gradiente aplicamos la Ecuación 8 obteniendo $G_x = 30 - 4 = 26$ y $G_y = 12 - 24 = -12$.

Utilizando la Ecuación 4 se obtiene $|G| = 48$, si fijamos un valor de umbral $T = 30$, el píxel marcado con el punto negro sería un punto de borde y si por el contrario, el umbral es mayor que 48, dicho punto no sería considerado de borde.

A continuación se desplazaran las máscaras a la siguiente posición, en este caso al píxel con valor 9 situado a la derecha del que acabamos de procesar y así sucesivamente hasta llegar al píxel con valor 6, con lo que se completaría esta fila, ya que no podemos llegar hasta el final por el tamaño de 3×3 de las máscaras, tras lo cual el siguiente píxel a procesar sería el marcado con valor 3 en la siguiente fila y así sucesivamente.

Método de Prewitt (Escalante, 2006): El operador de Prewitt es similar al de Sobel diferenciándose en los coeficientes.

$$\begin{matrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \\ \text{(a)} & \text{(b)} \end{matrix}$$

Ecuación 10

- (a) Mascara usada para obtener G_x en el punto central de la región 3×3
- (b) Mascara usada para obtener G_y en el mismo punto.

La magnitud y dirección del gradiente se obtienen como en el caso de sobel por medio de las



Ecuación 2, Ecuación 3, Ecuación 4 en la Imagen 5 se muestra la imagen binarizada obtenida aplicando el metodo de Prewitt.

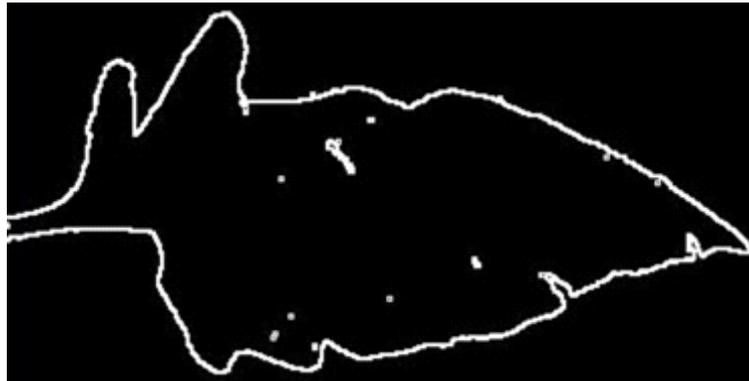


Imagen 5: Imagen binarizada con método de Prewitt

Método de Roberts (Ramos Rivas, 2003): El operador de Roberts a diferencia de los anteriores, marca solamente los puntos de borde, sin informarnos sobre la orientación de los mismos. Es un operador muy simple que trabaja muy bien en imágenes binarias. Opera según las dos diagonales perpendiculares mostradas en la Ecuación 11 y definidas por la Ecuación 12.

$$\begin{array}{ccccc}
 f(x-1,y-1) & \swarrow & D2f(x,y-1) & f(x+1,y-1) & \\
 f(x-1,y) & & \nearrow D1f(x,y) & f(x+1,y) & \\
 f(x-1,y+1) & & f(x,y-1) & f(x+1,y+1) &
 \end{array}$$

Ecuación 11

$$D_1 = f(x, y) - f(x - 1, y - 1)$$

$$D_2 = f(x, y - 1) - f(x - 1, y)$$

Ecuación 12

Existen dos formas del operador de Roberts dadas por la Ecuación 13 1) raíz cuadrada de la suma de las diferencias de los vecinos diagonales al cuadrado y 2) suma de la magnitud de las diferencias de los vecinos diagonales.

$$R = |D_1| + |D_2|$$

$$R = \sqrt{D_1^2 + D_2^2}$$

Ecuación 13

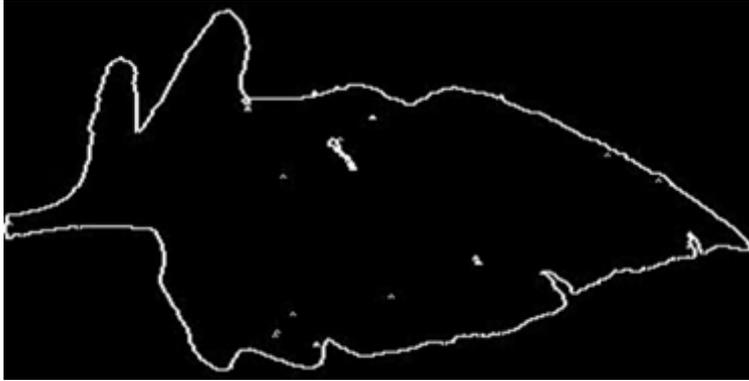


Imagen 6: Imagen binarizada con método de Roberts

Método de Kirsch: Las máscaras de Kirsch se denominan también de brújula por que se definen considerando una máscara simple y rotándola en las ocho direcciones principales de la brújula: Norte, Noroeste, Oeste, Suroeste, Sur, Sureste, Este y Noroeste. Las máscaras se definen como sigue en la Ecuación 14.

$$\begin{aligned}
 K_0(0^\circ) &\equiv \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & K_1(45^\circ) &\equiv \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & K_2(90^\circ) &\equiv \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
 K_3(135^\circ) &\equiv \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & K_4(225^\circ) &\equiv \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & K_5(180^\circ) &\equiv \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \\
 K_6(315^\circ) &\equiv \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & K_7(270^\circ) &\equiv \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}
 \end{aligned}$$

Ecuación 14

Para cada punto de la imagen se obtienen ocho valores, resultantes de la convolucion con cada una de las máscaras, el valor del módulo del gradiente resulta ser el máximo de esos ocho valores, mientras que la dirección queda determinada por el ángulo asociado a la máscara que ha generado dicho valor máximo.

Método de Freichen (Ramos Rivas, 2003): Este método usa las siguiente nueve mascararas

$$f_1 \equiv \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & 1 \end{bmatrix} \quad f_2 \equiv \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} \quad f_3 \equiv \frac{1}{2\sqrt{2}} \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix}$$

$$f_4 \equiv \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} \quad f_5 \equiv \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad f_6 \equiv \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

$$f_7 \equiv \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad f_8 \equiv \frac{1}{6} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} \quad f_9 \equiv \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Ecuación 15

En esta operación, cuando se compara con otro realce de borde, es más sensible a la configuración de valores independientes de píxeles. Dentro de JAI se pueden aplicar estas máscaras de forma fácil usando el operador “gradientmagnitud” (Manual de Usuario JAI 2017). Para esto es necesario crear una matriz con los respectivos valores de las máscaras. La fórmula empleada de este operador se puede definir como sigue:

Donde SH (x, y, b) y SV (x, y, b) son las imágenes gradiente horizontal y vertical respectivamente de la banda b de la imagen fuente correlacionada con las máscaras (horizontales y verticales) ortogonales provistas por el gradiente.

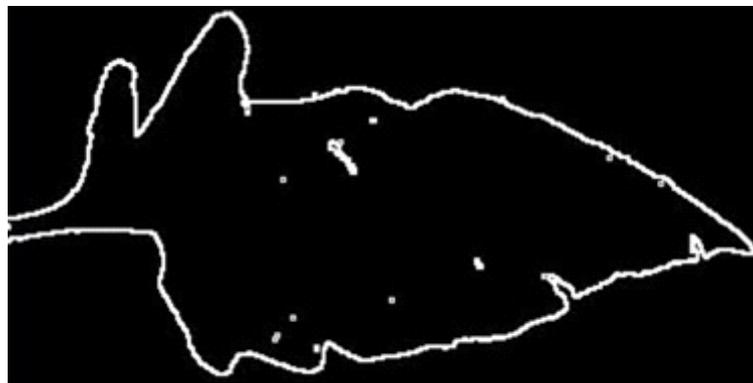


Imagen 7: Imagen binarizada con método de Freichen

En este proyecto la máscara empleada fue la de Roberts, esto debido a que este método resalta los bordes de los objetos en una forma no muy ancha y esto nos ayuda para que los contornos de las imágenes no sean tan “gruesos” cosa que a futuro será de gran ayuda.

Obtener las coordenadas de los píxeles que conforman el contorno

Luego de aplicar el método de resalto de contorno con el método de Roberts, el paso siguiente es obtener las coordenadas de los píxeles donde sus valores son 1's. Para esto se usa el método PxlGrabber (Manual de Usuario JAI 2017) donde se obtiene un arreglo con los valores de todos los píxeles de la imagen, a continuación en otro arreglo se guardan los píxeles que sólo tengan valores a uno.



Ya teniendo la información necesaria de los valores de los píxeles lo siguiente es aplicar un algoritmo para obtener las coordenadas del contorno. Para eso se usó el método de rastreo de contorno de Moore utilizando los denominados 8 vecinos de un píxel. Se conoce rastreo de contorno como al seguimiento de los bordes o límites de un objeto. Esto se realiza mediante los límites de un patrón dado, donde estos límites lo conforman los píxeles del borde del patrón. Existen dos formas de un píxel límite o borde los píxeles de 8 bordes y los píxeles de 4 bordes. Un píxel negro es considerado de borde 4 si comparte en sus límites al menos un píxel blanco, por otra parte, un píxel negro es de borde 8 si comparte en sus límites o en sus esquinas un píxel blanco. De esta forma un píxel de borde 4 también es un píxel de borde 8 pero un píxel de borde 8 puede o no ser un píxel de borde 4.

Una vez obtenidas las coordenadas del contorno de la imagen se usan diferentes técnicas para coleccionar sus características y puedan ser usadas para otros procesos. Los vecinos de Moore de un píxel P es un conjunto de 8 píxeles que comparten los límites y esquinas con dicho píxel. Como se muestra en la Ecuación 16.

$$\begin{array}{ccc}
 P_1 & P_2 & P_3 \\
 P_8 & P & P_4 \\
 P_7 & P_6 & P_5
 \end{array}$$

Ecuación 16

El procedimiento de esta técnica es simple, dado un patrón digital, por ejemplo, un conjunto de píxeles negros en un fondo blanco o viceversa, se localiza un primer píxel negro, denominado píxel de inicio, en este caso este píxel de inicio se encuentra avanzando de la esquina izquierda superior cada columna y renglón hasta encontrar un primer píxel negro. Una vez localizado este píxel de inicio avanzamos en dirección de las agujas del reloj, sus ocho vecinos, hasta que encontremos otro píxel negro. Así cada vez que encontremos un píxel negro P retrocedemos al píxel blanco de dónde venimos y a partir de ahí avanzamos a los demás píxeles vecinos de P. Este algoritmo termina cuando regresemos al píxel de inicio. Como se podrá observar con el algoritmo se viajara por todo el contorno de un objeto y vamos obteniendo sus coordenadas. Sin embargo si se aplica este criterio de 20 paradas existen algunos casos donde no recorre todo el contorno. Para ello basta con escoger otro criterio de parada. Existen dos opciones:

- Parar después de visitar el píxel de inicio más de una vez.
- Parar después de visitar el píxel de inicio una segunda vez pero en la misma forma de cómo entramos a este píxel.

A este criterio se le conoce como criterio de parada de Jacob. Ya que fue propuesto por Jacob Eliosoff.

Usando el criterio de Jacob mejorará de forma notable el algoritmo de Moore



convirtiéndolo en el mejor algoritmo de rastreo de contorno de un objeto. Una vez realizado el rastreo de contorno de los 8 vecinos de Moore, es decir que una vez que se haya identificado el objeto, se procederá a realizar la búsqueda de la enfermedad solamente dentro de las coordenadas encontradas por el método de Moore.

Clasificación de imágenes basadas en píxel

La siguiente tarea es identificar para cada píxel en una imagen, a que clase debe ser asignado ese píxel, con clase nos referimos a las diferentes enfermedades que se pueden presentar por ejemplo clase1=enfermedad1, clase2=enfermedad2, etc. Existen dos métodos principales para la clasificación de imagen: supervisada y no supervisada.

- En una clasificación supervisada, debemos enseñarle al algoritmo de clasificación como diferenciar una clase de otra, usualmente dando ejemplos de píxeles que sabemos que deberían ser asignados a una clase particular. El algoritmo usara la información que se le provee para clasificar los otros píxeles en la imagen.
- En una clasificación no supervisada se le provee al algoritmo información básica de cómo se espera que las clases sean representadas en la imagen, y el algoritmo busca identificar esas clases. Ambos métodos de clasificación (supervisado y no supervisado) utilizan alguna medida de distancia la cual es calculada sobre los valores de los píxeles (no sobre sus coordenadas en la imagen)

Distancia en un espacio de características

La idea principal es que si un píxel es asignado a la clase A, píxeles similares a él deberían ser asignados a la misma clase. El concepto de similaridad para la clasificación de píxeles es muy importante, los píxeles son similares entre sí, no en relación a su posición o contexto, pero si en relación a sus valores.

En el caso de imágenes RGB, cada píxel es representado por tres coordenadas, usualmente en el rango [0-255], que representa un punto en el espacio de características tridimensional. Como ejemplo, el píxel cuyo valores RGB son (128, 128, 128) puede ser ubicado en el centro del cubo RGB (o espacio de características), mientras el píxel (0, 0, 255) y (255, 0, 0) puede ser ubicado en vértices diferentes del cubo RGB o espacio de características. Píxeles vecinos pueden estar muy distantes en el espacio de características, y píxeles distantes en la imagen pueden estar muy cerca en el espacio de características.

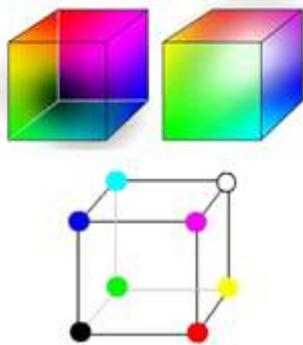


Imagen 8: Cubo RGB o Espacio de Características

Los dos conceptos basados en la distancia son muy importantes en este trabajo, los cuales son intervalos en el espacio de características y distancias en el espacio de características.

Un intervalo entre dos píxeles es una región en el espacio de características cercado por dos valores de píxeles, por ejemplo la región cercada por $(18, 30, 50)$ y $(23, 48, 90)$ en el espacio de características RGB contiene todos aquellos píxeles que cumplen con la regla “ $18 \leq R \leq 23$ ”, “ $30 \leq G \leq 48$ ” y “ $50 \leq B \leq 90$ ”. Un simple clasificador puede ser construido para crear y aplicar esta regla.

La distancia entre dos píxeles en un espacio de características puede ser calculada de muchas maneras diferentes. En este trabajo se utiliza solamente la distancia Euclideana, la cual es la distancia más corta entre píxeles en un espacio de características, y es calculada como la raíz cuadrada de la suma de las diferencias entre los valores de dos píxeles al cuadrado. La distancia entre píxeles en el espacio de características se utiliza para comparar. Para la mayoría de los algoritmos se tiene que calcular la distancia de un píxel a otros píxeles (siempre en función del espacio) y decidir a qué clase pertenece basada en la distancia más pequeña. Por ejemplo, considere la posibilidad de que se sabe que los píxeles de la clase A tienen valores en torno a $(21, 71, 40)$ y los píxeles de la clase B tienen píxeles con valores en torno a $(66, 75, 70)$. Si se presenta un píxel para el cual la clase no se conoce y con los valores $(37, 77, 53)$ se puede calcular las distancias entre este píxel y el píxel de la clase A (21.47) y el píxel con el píxel de la clase B (33.67), y decidir que, dado que el píxel está más cerca (en función del espacio) para la clase A que la clase B probablemente debería ser asignado a la clase A. Este método de clasificación se llama clasificador de distancia mínima y es utilizado en la propuesta del prototipo de Sistema.

La extracción de muestras para la clasificación basada en píxeles

Muchos métodos de clasificación supervisada requieren como entrada, muestras de todas las clases que se utilizarán para la clasificación.



La mayoría de los algoritmos no utilizan los valores de los píxeles de la muestra directamente, en su lugar calculan algunas firmas de los píxeles que se utilizan para representar las clases correspondientes. Las firmas pueden ser consideradas como descriptores de las clases, a menudo contienen información estadística sobre los píxeles utilizados como muestras.

Las firmas son específicas de un método de clasificación, cada uno requiere información diferente acerca de los píxeles para la etapa de clasificación. Para crear las firmas (o entrenar el algoritmo de clasificación supervisada) se debe identificar las regiones en una imagen prototipo que contiene los píxeles para una clase particular (las muestras) y usar esos píxeles para el cálculo de las firmas. Este proceso debe repetirse para cada clase (o expresada en términos diferentes para cada enfermedad), para los que se puede utilizar una o más regiones de la muestra. La identificación de las muestras debe ser realizada por un experto en las características de la imagen (para los ejemplos presentados, una persona con la formación de Agrónomo).

La se usara para la clasificación en el Sistema. Se trata de una hoja de la planta de Tomate que presenta signos de una enfermedad (Bacterias, hongos). Las clases que se utilizaran para su clasificación son las correspondientes a la enfermedad que afecta a la planta, algunas regiones de la imagen se deben seleccionar para las muestras que se utilizaran en el Sistema, las cuales representaran a las clases. Los pasos a realizar son los siguientes:

- Identificar Muestras: Identificar en la Imagen Original aquellas regiones donde se presenta la enfermedad. El sistema necesitara 3 datos muy importantes para lograr realizar una adecuada creación de la firma representativa de esa clase o enfermedad, los datos necesarios son: a) la posición (x,y) de la esquina superior izquierda, b) el ancho del rectángulo que abarcara la muestra c) el alto del rectángulo que abarcara la muestra. En la Imagen 9 se puede observar las regiones seleccionadas como muestras en la Imagen Origen, para luego poder realizar las firmas que nos permitirán clasificar la Imagen.



Imagen 9: Selección de Regiones donde se presenta la enfermedad



En este caso solo seleccionamos 3 regiones representativas de la enfermedad que padece la Imagen Origen, las cuales se encargaran de simbolizar la enfermedad que padece la planta.

- Identificar Clases: El Sistema utiliza clases para representar las enfermedades que presenta la planta, es decir que el sistema una vez que se encuentra funcionando recorrerá toda la imagen en busca de regiones similares a las que se encuentran presente en recuadros rojos en la Imagen 9. Una vez localizadas reemplazara las regiones encontradas con los colores correspondientes que se establecieron en la clase de esa enfermedad, es decir que para nuestro caso los píxeles de las regiones encontradas serán reemplazados por el color RGB [255 80 80] establecido en la clase de esa enfermedad.

- Creación de Firmas: Como se explicó con anterioridad se deben crear firmas las cuales representaran a las regiones encontradas y servirán para realizar la búsqueda de enfermedades en las nuevas imágenes.

El prototipo de Sistema utilizara dos tipos de métodos que serán utilizados por el clasificador. Uno de ellos representara a la definición de clases, la cual contiene la/las definición/es de las clases para el clasificador, lo que se trata de explicar es que este método contendrá 3 valores muy importantes los cuales utiliza como color de referencia para esa clase.

Por ejemplo la clase utilizada en nuestro Sistema (255 80 80) la cual representa un color, se reemplazara en la imagen original si es que encuentra la enfermedad buscada. Como se mencionó anteriormente las regiones seleccionadas se utilizaran como muestra para cada clase las cuales nos ayudaran a poder crear las firmas, para ello de cada región seleccionada debemos ingresar la posición (x,y) de la esquina superior izquierda, seguida por el ancho de ese rectángulo y por el alto. En el Sistema para lograr calcular la firma introducimos: la posición de la esquina superior izquierda (217, 116), ancho del rectángulo: 9 y por ultimo su alto: 6, con el cual el sistema procederá a crear la firma correspondiente.

El clasificador

Es un clasificador supervisado muy simple, que utiliza intervalos de píxeles para determinar si un píxel pertenece a una clase o no. Los puntos de delimitación de los intervalos, se obtienen de los valores de los píxeles de las muestras para la clase. Dado que este clasificador es supervisado, hay dos pasos necesarios para su utilización: creación de la firma (entrenamiento) y la clasificación.

Creación de la firma (entrenamiento): Los pasos para la creación de una firma utiliza como entrada los píxeles de la imagen original, la descripción de clase y las muestras de los archivos de datos (imágenes) para calcular el mínimo y máximo límite para cada clase.

1. Para cada clase se crean un par de estructuras de datos para representar los límites (un límite mínimo $\min K$ y un límite máximo $\max K$, para cada clase K). Esta estructura de datos



debe tener el mismo dominio y dimensiones que los píxeles de la imagen que se clasificarán.

2. Para cada clase inicializar los valores mínimos del límite en $\min K$ con un valor lo suficientemente grande y los valores máximos del límite en $\max K$ con un valor lo suficientemente pequeño. Esta es una forma sencilla de asegurar que sus valores se cambiaran para reflejar los límites de los valores de los píxeles.

3. Para cada píxel en cada muestra de esa clase se debe hacer lo siguiente:

4.

A) Si alguno de los valores de los píxeles es menor que el valor correspondiente en $\min K$, establecer el valor correspondiente en $\min K$ con el valor del píxel.

B) Si alguno de los valores de los píxeles es mayor que el valor correspondiente en $\max K$, establecer el valor correspondiente en $\max K$ con el valor del píxel.

5. Guardar el resultado de los valores de los límites. Este sencillo procedimiento calculará los límites mínimos y máximos para cada categoría, que se utilizará como las firmas de las clases.

Clasificación: El proceso de clasificación es muy simple, todo lo que se necesita es la imagen original, la descripción de las clases (de forma que se pueda utilizar diferentes colores para representar las diferentes clases, son los colores que reemplazaran a la enfermedad en la imagen que se esté analizando) y las firmas creadas por la aplicación. Los pasos son los siguientes:

1. Leer la imagen original, la descripción de las clases y las firmas, almacenado su información en las estructuras adecuadas.

2. Crear una imagen con las mismas dimensiones que la original, la cual será "pintada", es decir aquellos píxeles clasificados serán pintados con los colores correspondientes a su clase.

3. Para cada píxel de la imagen original, obtener sus valores RGB.

4.

A) Inicializar una estructura para representar la clase de ese píxel (en este ejemplo se inicializa con el color negro ya que es diferente de todos los colores utilizados en la descripción de las clases de la imagen.)

B) Para cada clase, ver si sus límites (desde el archivo de firmas) contienen los valores RGB del píxel que está siendo considerado. Si es así, almacenar el color para esa clase como la nueva clase para el píxel.

C) Pintar el píxel de la imagen de salida con el color almacenado

5. Guardar la imagen resultante.

El algoritmo de clasificación permite el rechazo de los píxeles, lo que significa que los valores RGB del píxel analizado se encontraban fuera de los límites para todas las clases, dejando a los píxeles como no clasificados. Existen maneras de evitar el rechazo, la más

evidente es conseguir más muestras, preferentemente extraídas de las regiones con los píxeles sin clasificar.

La Imagen 10 se obtuvo como resultado de la aplicación del algoritmo de clasificación. Los colores de los píxeles son los mismos que se especifican en las definiciones de clases.



Imagen 10: Imagen resultante de la clasificación

Conclusiones

La mayoría de las personas se preguntan cómo pueden comparar imágenes (probablemente se refieren al contenido de las imágenes) para ver si son iguales, esperando por una simple función u operador que devuelva TRUE o un valor entre 0 y 1 donde 0 significa diferente, 1 significa igual y valores intermedios puede considerarse como diferentes grados de similitud. Muchas aplicaciones se pueden beneficiar de este alto nivel de comparación, por ejemplo, para detectar violaciones de copyright en imágenes, identificación de rostro en multitudes con propósitos de seguridad, etc.

Desafortunadamente comparar contenidos de imágenes no es una simple tarea, y en la mayoría de los casos, no está definido. Se debe considerar con cuántas variables se va a tener que tratar: cada objeto en la imagen tiene una forma, tamaño, orientación, color (el cual puede ser más que uno), textura, etc; el mismo objeto puede aparecer diferente en otras imágenes, por ejemplo puede aparecer en una diferente posición, escala, orientación o iluminación o bien una combinación de todas las anteriores ¿Se debe considerar estas variaciones como irrelevantes para la comparación o son importantes? ¿Se debería asignar diferentes “pesos” a los objetos, debido a que si las comparaciones son similares de esta forma la medida de similitud en algunos es más importante que en otros? ¿Los objetos pequeños se pueden perder en las imágenes, los debemos ignorar por ello? ¿Es el “fondo” un objeto y debería ser considerado en la comparación? Y por último una pregunta, que en muchas ocasiones se dificulta responder ¿Y qué es exactamente un “fondo”?

Una imagen digital está compuesta por píxeles, y estos píxeles representan un valor en un espacio multi-dimensional. Cada píxel puede ser representado como: x_i el cual es el valor del píxel x (vector de características) en el espacio de características “ i ”.

El método de clasificación asume que todos los píxeles que pertenecen a una clase tienen valores que se encuentran dentro de los límites definidos por cada clase o enfermedad. Un píxel "x" es clasificado a una clase "A" si todos los valores del vector "x" se encuentran dentro de los límites de la clase "A". La Imagen 11 presenta un ejemplo en un espacio de características bi-dimensional.

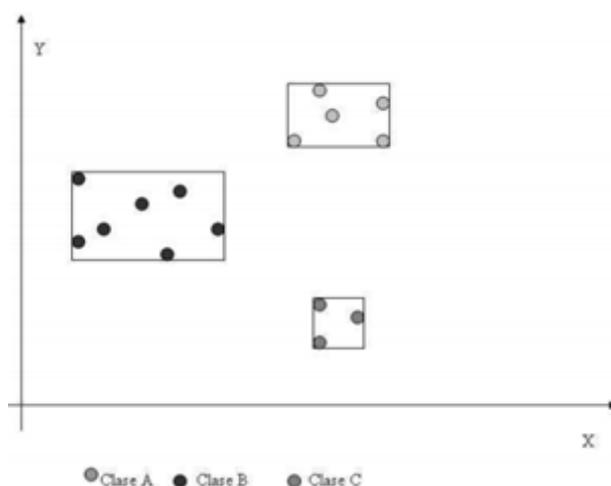


Imagen 11

La clasificación se realiza comprobando si cada píxel se encuentra dentro de los límites de alguna clase, devolviendo un valor verdadero o falso, si el valor es falso el píxel queda sin clasificar o bien clasificado como rechazado.

Unos de los aspectos a mejorar en el sistema de reconocimiento de imágenes serian en el caso de que existiera sobre posición de las clases (enfermedades) en todos los espacios de características que se estén teniendo en cuenta para el diagnóstico. En el ejemplo siguiente se puede observar que se puede encontrar el sobre posicionamiento de 2 clases (A y C) en el espacio de características bi-dimensional sobre el eje de las "X" como lo ilustra la Imagen 12. En el espacio de características tridimensional (haciendo referencia a los planos R, G, B) se podría utilizar el tercer plano para resolver el sobre posicionamiento. Cuando el problema se presenta en todos los espacios de características, no existe una forma simple de determinar a qué clase debería ser asignada el píxel.

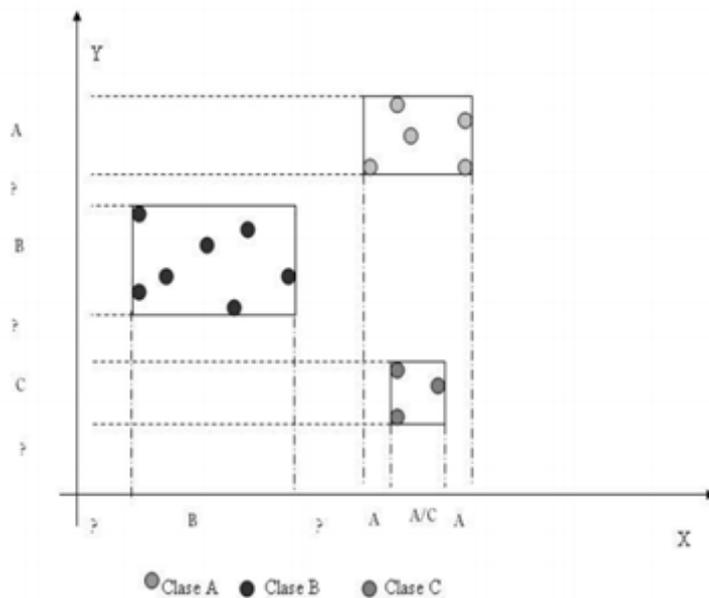


Imagen 12: Sobre posicionamiento de Clases

Otro punto a mejorar en el sistema es que el mismo permita reconocer formas, es decir la silueta del “objeto” a diagnosticar. Para permitir una clasificación más exacta de las enfermedades no solamente basadas en un descriptor de la enfermedad, sino también en comparar la silueta del objeto de estudio.

Actualmente el sistema lo realiza, pero se encuentra muy dependiente de la “calidad” de la imagen. Esto quiere decir que si la imagen no tiene la calidad necesaria, presenta sombras que cortan la silueta o si la imagen no toma el “objeto entero” a analizar, los bordes de la imagen a analizar no coincidirán con los bordes de la imagen que se tiene como base de comparación.

Otro problema que puede surgir es que el sistema implementa la opción de recortar la imagen por parte del usuario, esto sucede cuando el usuario desea acotar a una determinada porción de la imagen, el procesamiento de la misma.

Son puntos que a medida que se fue resolviendo el problema de clasificar enfermedades en base a imágenes, se pudieron observar, si bien aparecieron otros a lo largo de la investigación y el desarrollo del sistema, se fueron solucionando con el tiempo. Estos que se mencionaron son los que se podría implementar en una nueva versión del sistema, para una mejor clasificación de enfermedades en un diagnóstico de imagen.

Bibliografía

- Cattaneo, Carlos; Larcher, Ledda; Ruggeri, Ana; Herrera, Andrea; Biasoni, Enrique. (2011). Métodos de umbralización de imágenes digitales basados en entropía de shannon y otros. Extraído el 25 de septiembre del 2017, de <http://www.cimec.org.ar/ojs/index.php/mc/article/viewFile/3951/3868>
- Escalante, Boris. (2006). Procesamiento Digital de Imágenes. Extraído el 25 de septiembre del 2017, de <http://verona.fi-p.unam.mx/boris/teachingnotes/Capitulo4.pdf>
- Ferraro, Hector P. (2007). Reconocimiento automático de texto braille. Extraído el 25 de septiembre del 2017, de <http://sedici.unlp.edu.ar/bitstream/handle/10915/3978/Tesis.%20P.%20Reconocimiento%20autom%C3%A1tico%20de%20texto%20braille.pdf-PDFA.pdf?sequence=1>
- Manual de Usuario JAI. Extraído el 25 de septiembre del 2017, de http://download.oracle.com/otn-pub/java/jai-guide/1.0.1/jai1_0_1-guide.pdf?AuthParam=1506382581_6aea9c56067a5e9b80b27bbc328130d1.
- Ramos Rivas, Melva. (2003). Sistema de pre-procesamiento de imágenes electrocardiográficas en telemedicina. Extraído el 25 de septiembre del 2017, de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/ramos_r_m/