

Aplicación para obtener el área foliar desde imágenes obtenidas de un escáner

CONTRERAS-MURILLO, Sergio[`], LUGO-ESPINOSA, Oziel^{``} y AYALA-DE LA VEGA, Joel ^{```}

S. Contreras, O. Lugo y J. Ayala

[`]Universidad Autónoma del estado de México, Centro Universitario Texcoco, Alumno de la maestría en Ciencias de la Computación; Av. Jardín Zumpango s/n Fracc. El Tejocote, Texcoco, Estado de México

^{``}Universidad Autónoma del estado de México, Centro Universitario Texcoco, Profesor Investigador de tiempo completo en la Maestría en Ciencias de la Computación; Av. Jardín Zumpango s/n Fracc. El Tejocote, Texcoco, Estado de México;

^{```}Universidad Autónoma del estado de México, Centro Universitario Texcoco, Coordinador de la Maestría en Ciencias de la Computación; Av. Jardín Zumpango s/n Fracc. El Tejocote, Texcoco, Estado de México
jayalad@uaemex.mx

D. Sepúlveda, R. Salazar, F. Pérez y J. Rocha (eds.) Ciencias Químicas y Matemáticas-©ECORFAN, Texcoco de Mora-México, 2015.

5 Introducción

El tamaño de las hojas de las plantas es uno de los aspectos más importantes en el estudio de la salud de un cultivo pues está directamente relacionado con su capacidad para absorber la luz solar e intercambiar humedad y CO₂ con su ambiente.

Mediante el análisis del área foliar puede medirse el efecto de una plaga sobre una planta, la eficacia de las técnicas y sustancias empleadas en su cuidado o cuantificar su volumen de producción.

Las técnicas para determinar el área foliar se dividen en destructivas y no destructivas, dependiendo de si la hoja estudiada es separada o no de la planta. Dichas técnicas intentan contrarrestar los errores presentes en una estimación manual.

Un primer método de medición es la utilización de un integrador electrónico. Existen integradores portátiles y de laboratorio. Estos equipos son escáneres cuya principal desventaja es su costo elevado.

La segunda técnica es la obtención del área de una hoja mediante estimaciones matemáticas con la desventaja de que debe diseñarse un modelo matemático por cada cultivo distinto a estimar. La exactitud de un modelo estadístico depende de la técnica empleada y de la calidad de las muestras utilizadas para su desarrollo.

Las técnicas estadísticas son una alternativa al uso de equipo especializado de medición pero dependen a su vez de otras mediciones que, de realizarse manualmente, pueden propagar errores.

En cuanto a las técnicas destructivas, la medición en condiciones de laboratorio se centra en el uso de aplicaciones para el tratamiento digital de imágenes. Un ejemplo es la aplicación ImageJ® (imageJ) que es una herramienta de propósito general que permite aplicar transformaciones a imágenes digitales.

En este trabajo se presenta una aplicación de tratamiento digital de imágenes que a diferencia de otras opciones es específica para obtener el área foliar por lo que el usuario sólo requiere seleccionar la imagen que desee analizar.

Otras opciones requieren que el usuario aumente el contraste, elimine el ruido, aplique filtros, etc. antes de obtener el área en píxeles. Además, debe instrumentar una técnica para convertir de píxeles a cm en imágenes tomadas, por ejemplo, con cámaras digitales de una resolución y a una distancia particulares.

La interface gráfica de la aplicación fue desarrollada con lenguaje java en el IDE netbeans y el tratamiento digital de las imágenes se realizó mediante la ejecución de un script del lenguaje R.

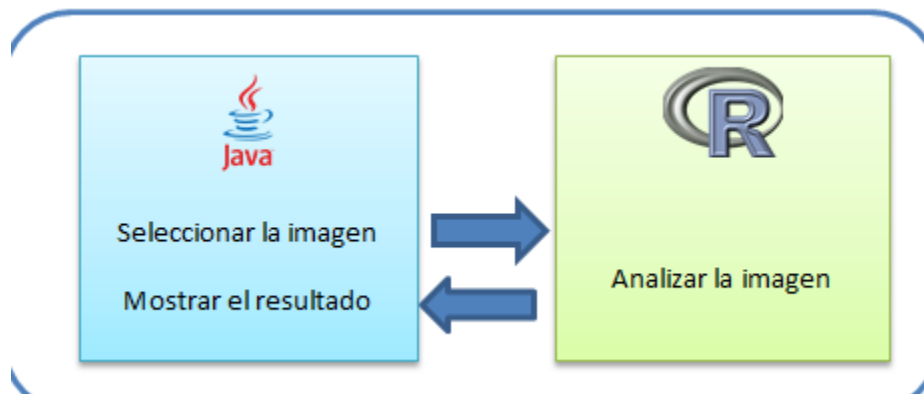
5.1 Materiales y métodos

La aplicación realiza el tratamiento digital de las imágenes mediante la ejecución de un script del lenguaje R en una instancia instalada por el usuario.

"R es un lenguaje y entorno para cómputo estadístico y graficación, similar al lenguaje S desarrollado originalmente en los laboratorios Bell. Es una solución de código abierto para el análisis de información que es mantenido por una activa y numerosa comunidad alrededor del mundo" (Kabacoff, 2011).

La interface gráfica está desarrollada con el lenguaje Java. El desarrollo en diferentes plataformas de la lógica de negocios y la interface gráfica permite que ambas secciones sean codificadas paralelamente, ahorrando tiempo y favoreciendo la integración de equipos multidisciplinarios (Figura 5).

Figura 5 módulos de la aplicación



El programa fue desarrollado para el sistema operativo Windows© pero puede utilizarse en Linux© y Macintosh© pues tanto R como Java son multiplataforma.

La aplicación presentada se desarrolló en la versión 8 del entorno de desarrollo gratuito NetBeans.

Por otro lado, los scripts del lenguaje R se desarrollaron en el entorno de desarrollo gratuito CRAN (Comprehensive R Archive Network) disponible para el sistema operativo Windows.

Al instalar "R" se publica la variable de entorno R_HOME que indica la ubicación de los archivos instalados. Las variables de entorno son una lista de claves que el sistema operativo almacena sobre la ubicación de archivos y programas.

Inicialmente, R solo incluye un conjunto base de paquetes. Los paquetes son conjuntos de scripts desarrollados por programadores alrededor del mundo cuyas funciones pueden ser utilizadas gratuitamente en las aplicaciones. Existen más de 5000 paquetes disponibles para su descarga desde servidores de universidades y centros de investigación.

El script de R que analiza las imágenes digitales requiere de la instalación de 3 de estos paquetes:

Paquete JPG:

Este paquete proporciona una manera fácil y sencilla de leer, escribir y mostrar imágenes de mapa de bits almacenados en el formato JPEG. Puede leer y escribir vectores en memoria o hacia archivos (jpeg, 2014).

Paquete BMP:

Lee imágenes en formato BMP de Windows. Actualmente se limita a imágenes en escala de grises a 8 bits e imágenes (A) RGB de 24 y 32 bits. Aplicación R sin dependencias externas (bmp, 2014).

Paquete PIXMAP:

Funciones para la importación, exportación, impresión y demás manipulaciones de imágenes de mapas de bits (pixmap, 2014).

La aplicación supone que el usuario digitalice sus muestras como imágenes en formato jpg con la ayuda de un escáner convencional (Figura 5.1). La aplicación no está acotada a ninguna marca o modelo particular de escáner. Tampoco requiere que las imágenes tengan una resolución o tamaño específico pero debe tenerse en cuenta que su calidad incide directamente en la rapidez del análisis.

Figura 5.1 Digitalización de las hojas



Las imágenes en formato JPG obtenidas desde un scanner tienen como parte de sus metadatos su densidad, es decir, el número de píxeles que representan una pulgada cuadrada. Este valor es utilizado por las impresoras para imprimir las imágenes en su tamaño correcto. También es utilizado por una computadora para desplegar las imágenes en un monitor en su tamaño real y es utilizado por la aplicación para convertir los resultados de píxeles a centímetros.

Figura 5.2 Parte de los metadatos de una imagen en formato JPG

```

<javax_imageio_jpeg_image_1.0>
  <JPEGvariety>
    <app0JFIF
      majorVersion="1"
      minorVersion="1"
      resUnits="1"
      Xdensity="96"
      Ydensity="96"
      thumbWidth="0"
      thumbHeight="0"/>
    </JPEGvariety>
  <markerSequence>
    <dqt></dqt>
    <sof
      process="0"
      samplePrecision="8"
      numLines="513"
      samplesPerLine="931"
      numFrameComponents="3">
    </sof>
    <dht></dht>
    <sos></sos>
  </markerSequence>
</javax_imageio_jpeg_image_1.0>

```

La figura 5.2 muestra un fragmento de los metadatos de una imagen en formato JPG en la que las propiedades "numLines" y "samplesPerLine" de la etiqueta "sof" hacen referencia al alto y ancho de la imagen en pixeles. Por su parte, las propiedades "Xdensity" e "Ydensity" de la etiqueta "app0JFIF" muestran los pixeles que, vertical y horizontalmente, dibujan una pulgada cuadrada en su escala original.

La clase "ImageIO" del paquete javax.imageio permite tener acceso a los metadatos de una imagen desde un programa de java. La estructura XML de los metadatos de las imágenes es diferente para cada tipo de archivo. Actualmente la aplicación solo soporta el formato JPG.

La estrategia de funcionamiento de la aplicación es la siguiente (Figura 5.3):

Habiendo digitalizado la muestra utilizando un escáner:

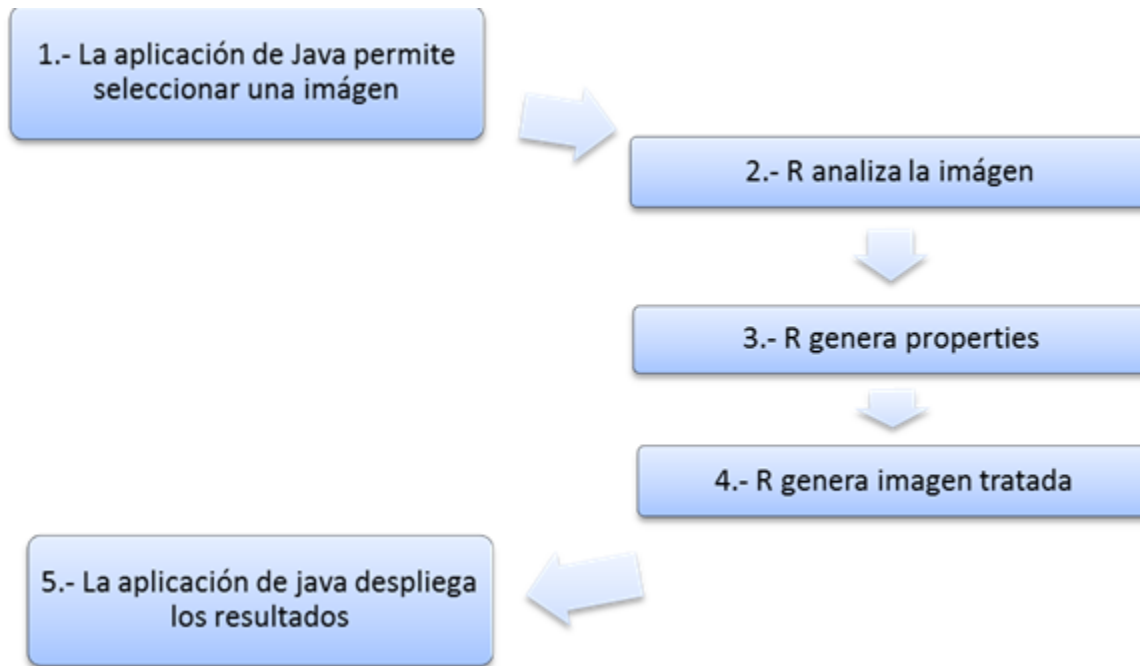
- 1.- La aplicación le solicita al usuario que seleccione una imagen a analizar.
- 2.- La aplicación ejecuta un script del lenguaje R para analizar el archivo seleccionado.
- 3.- Las dimensiones de la hoja se guardan en un archivo de texto con el mismo nombre y ubicación que la imagen pero de tipo "properties".
- 4.- También se guarda un archivo de imagen con el mismo nombre y ubicación que la imagen original pero en formato de mapa de bits de 256 colores que contiene la versión tratada digitalmente.
- 5.- La aplicación despliega los resultados al usuario.

Los resultados son almacenados para que cuando el usuario revise una imagen nuevamente no sea necesario volver a analizarla.

La aplicación no requiere que las carpetas e imágenes tengan una estructura o nomenclatura específica. El usuario puede almacenarlas como prefiera.

El contenido del archivo properties y la imagen resultante del análisis pueden ser revisados posteriormente por el usuario, incluso sin utilizar esta herramienta.

Figura 5.3 Funcionamiento de la aplicación



Internamente R convierte la imagen a blanco y negro. Debido a que las imágenes son obtenidas de un escáner no se consideró eliminar el ruido pues las condiciones de iluminación suelen ser óptimas y el proceso elevaría el tiempo de ejecución de la aplicación.

La imagen es tratada como matriz numérica donde cada pixel es representado con un número que va del 0 al 256 y que representa tonos de oscuros a claros respectivamente (Figura 5.3).

Figura 5.4 Imagen de una hoja en blanco y negro.



A partir de la imagen en blanco y negro se detectan los bordes de la hoja con el operador de Roberts Cross (Figura 5.4). Este método le aplica dos kernels a la matriz numérica mediante una operación de convolución.

Al implementarlo [El detector de bordes], el valor máximo arrojado por la aplicación de dichos patrones es guardado como el valor del borde en ese punto. El punto $E_{x,y}$ en el borde es entonces el máximo de los dos valores derivados de la convolución de los dos kernels en un punto $P_{x,y}$ de la imagen. (Nixon, 2008).

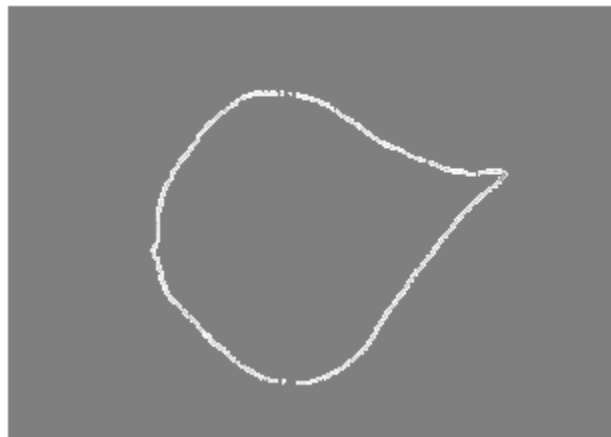
Operador de Roberts Cross:

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Operación de convolución:

$$E_{x,y} = \max\{|M^+ * P_{x,y}|, |M^- * P_{x,y}|\} \quad \forall x, y \in 1, N - 1 \quad (1)$$

Figura 5.5 Aplicación del detector de bordes



Las coordenadas de los pixeles que componen el borde son aislados para ser presentados al usuario dibujados sobre la imagen original (Figura 5.5).

Figura 5.6 Borde de la hoja

Para encontrar el ápice (punta) de la hoja se busca la esquina del borde que esté más alejada de cualquier otro pixel. Una esquina es el punto en el que la dirección del borde cambia más allá de cierto número de grados. Dependiendo del tipo y estado de la muestra podría haber varias esquinas pero solo una es el ápice. El ápice es necesariamente una esquina.

Las matrices de la Figura 5.7 son un ejemplo de secciones que pueden encontrarse en la matriz numérica que representa la imagen, Los valores reales van del 0 al 256 pero se presenta binarizada para su mejor análisis.

Figura 5.7 Ejemplos de secciones presentes en la imagen.

$$\begin{array}{ccc}
 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} &
 \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} &
 \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\
 \text{Color sólido} & \text{Borde} & \text{Esquina}
 \end{array}$$

La interface gráfica permite que el usuario reubique la base y la punta propuestas en el caso de que la las condiciones o taxonomía de la hoja hubiera dificultado su análisis.

Finalmente se realiza un barrido perpendicular al eje formado por la base y la punta para obtener la parte más ancha de la hoja.

El método “exec” de la clase `java.lang.Runtime` permite ejecutar comandos del sistema operativo desde una aplicación Java.

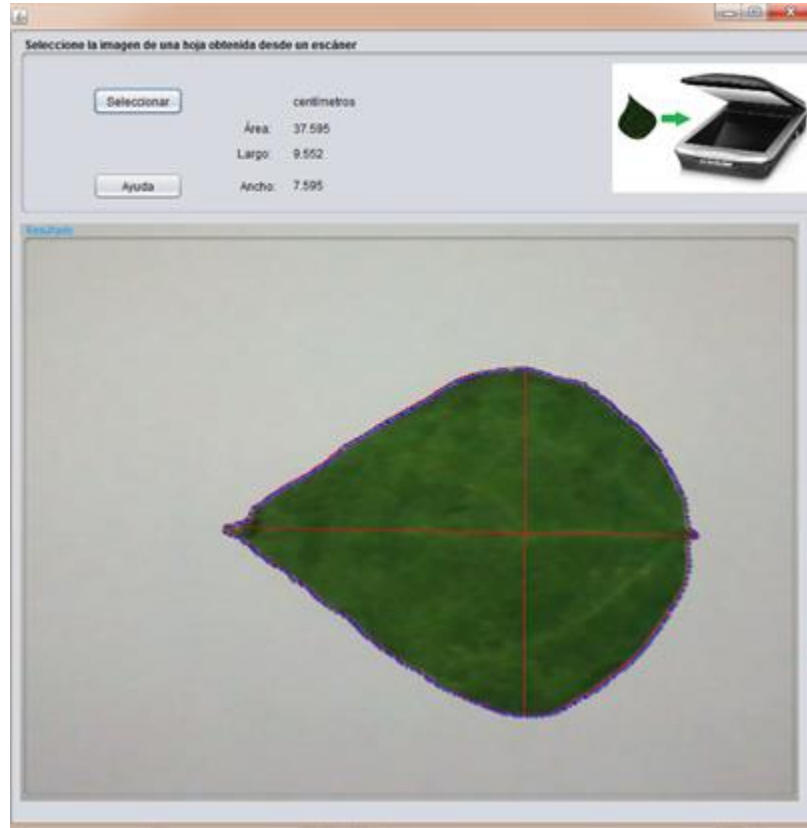
El comando para solicitarle a R la ejecución de un script es "R CMD BATCH script.R log.txt" solo accesible si la variable de entorno `R_HOME` está correctamente establecida.

5.2 Resultados

La aplicación requiere que el usuario tenga a Java y a R apropiadamente instalados. La aplicación consta de un archivo ejecutable que se guarda junto con los scripts de R en una ubicación específica. La aplicación se diseñó para que sea el usuario el que defina la estructura de carpetas y nombres para sus imágenes. La figura 9 muestra la única ventana de la aplicación.

Para iniciar el análisis de una imagen el usuario selecciona un archivo en formato JPG obtenido desde un escáner presionando el botón "seleccionar". La imagen seleccionada es tratada y el resultado es presentado como texto en la parte superior y gráficamente debajo (Figura 5.8).

Figura 5.8 Ejecución de la aplicación



Los resultados son almacenados en un archivo de propiedades con el mismo nombre y ubicación que la imagen seleccionada. Si la imagen ya había sido tratada, entonces sólo despliega los resultados previamente obtenidos; del mismo modo se guarda la imagen tratada digitalmente. Ambos archivos pueden ser revisados por el usuario posteriormente incluso sin utilizar esta herramienta.

5.3 Conclusiones

Existen opciones gratuitas y comerciales de aplicaciones que permiten realizar un tratamiento digital a las imágenes y obtener las medidas de las figuras que muestran, pero al ser de propósito general requieren que el usuario realice cada vez configuraciones, pasos y conversiones que se evitan al utilizar una herramienta de uso específico como ésta.

Las imágenes obtenidas desde un escáner sí guardan en sus metadatos su densidad, imágenes obtenidas de otras fuentes podrían no contener esta información. Para esta técnica tampoco se requiere un fondo de color específico o con patrones. Incluso no depende de una marca o tipo específicos de escáner en tanto que se configure para obtener las imágenes en formato JPG.

La aplicación supone que solo aparezca una hoja simple por imagen.

La aplicación se diseñó para no ser una herramienta intrusiva, es el usuario el que establece su propia estructura de carpetas para almacenar sus archivos digitales y la imagen original no es alterada.

Los archivos no están acotados a un tamaño y resolución específicos pero el usuario debe considerar que la velocidad de ejecución de la herramienta depende de las dimensiones de las imágenes seleccionadas y de las especificaciones del equipo de cómputo; cuanto menor sea la resolución y tamaño de las imágenes, menor será también el tiempo que tome su análisis. El usuario puede configurar su escáner para obtener los archivos con las propiedades que requiera su investigación.

Tanto el lenguaje R como el lenguaje Java son distribuidos gratuitamente y son ampliamente utilizados en el ámbito académico y de investigación.

5.4 Referencias

Jefferis, G. (2014) read windows Bitmap (BMP) images. Consultado el 2 de septiembre de 2014 en <http://cran.r-project.org/web/packages/bmp/>

Rasband, W. (2014). ImageJ features .Consultado el 2 de septiembre de 2014 en <http://imagej.nih.gov/ij/features.html>

Urbanek, S. (2014) read and write JPEG images. Consultado el 2 de septiembre de 2014 en <http://cran.r-project.org/web/packages/jpeg/>

Kabacoff, R, R in action, data analysis and graphics with R 1a ed. Nueva York. Manning, 2011. p. 5. ISBN: 9781935182399

Nixon M. y Aguado A. Feature Extraction and image Processing, 2a ed., Londres, Elsevier, 2008, p. 120. ISBN: 9780123725387

Bivand, R; Leisch, F. y Maechler M. (2014). pixmap: Bitmap Images ("Pixel Maps"). Consultado el 2 de septiembre de 2014 en <http://cran.r-project.org/web/packages/pixmap>

Hornik K, (2013). Frequently asked questions on R. Consultado el 2 de septiembre de 2014 en <http://cran.r-project.org/doc/FAQ/R-FAQ.html>

Venables W. N. y Smith D. M. (2014)., An Introduction to R, Notes on R: A Programming Environment for Data Analysis and Graphics, Version 3.0.1, R Core Team, 2014.