

COLEGIO DE POSTGRADUADOS  
CAMPUS MONTECILLO

POSGRADO EN SOCIOECONOMÍA, ESTADÍSTICA, E INFORMÁTICA

CÓMPUTO APLICADO



MANUAL DE USUARIO

IDENTO V1.0

<sup>1</sup>JUAN PABLO AMBROSIO AMBROSIO, \*<sup>1</sup>JUAN MANUEL GONZÁLEZ CAMACHO

---

<sup>1</sup> Campus Montecillo, Colegio de Postgraduados. Carretera México-Texcoco km. 36.5 P.O. Box 56230, Montecillo, México, México. Tel: 595 9520200 ext. 1431 (jmgc@colpos.mx)

## CONTENIDO

PROPÓSITO .....	1
ALCANCE Y LIMITACIONES .....	2
CARACTERÍSTICAS DE IDENTO .....	2
AMBIENTE DE DESARROLLO .....	3
GUÍA DE INSTALACIÓN .....	4
PROCESAMIENTO DE IMÁGENES .....	7
ESPACIO DE COLOR HSV .....	7
ESPACIO DE COLOR ExR .....	10
MÁQUINA DE SOPORTE VECTORIAL .....	11
SELECCIÓN DE MUESTRAS .....	12
VARIABLES PREDICTORAS .....	13
CRECIMIENTO POR REGIÓN .....	13
SELECCIÓN MANUAL .....	15
PREDICCIÓN .....	15
SOPORTE TÉCNICO .....	16
CONCLUSIONES .....	17
REFERENCIAS BIBLIOGRÁFICAS .....	17

## INTRODUCCIÓN

El desarrollo de la tecnología computacional en las últimas décadas ha permitido avances exponenciales en la implementación de modelos de inteligencia artificial en las diferentes áreas de la actividad humana. En la agricultura se han implementado modelos para tareas de monitoreo, diagnóstico, cuantificación, clasificación y predicción. En muchas investigaciones no se llega a la etapa de producción y distribución de las herramientas desarrolladas al usuario final.

Los cultivos en su gran mayoría son afectados por agentes fitopatógenos, en particular el tomate (*Solanum lycopersicum l.*) es sensible a enfermedades causadas por virus, hongos, e insectos, que disminuyen el rendimiento del cultivo y generan un impacto directo sobre la producción. El diagnóstico oportuno permite desarrollar acciones de prevención y control. Esta tarea se ha abordado ampliamente por las técnicas de visión artificial, y de aprendizaje automático.

En este manual se describen los componentes y opciones básicas de uso de la herramienta informática IDENTO desarrollada para clasificar e identificar tres enfermedades en el cultivo de tomate (*Solanum lycopersicum l.*) y hoja sana a partir de imágenes digitales de las hojas. Esta herramienta incorpora el uso de modelos de inteligencia artificial y visión por computadora para su aplicación de manera versátil por medio de una interfaz gráfica de fácil utilización.

## PROPÓSITO

Describir los componentes y el uso de la interfaz gráfica para facilitar el procesamiento, selección de muestras de imágenes y predicción de tres clases de enfermedades en el cultivo de tomate (*Solanum lycopersicum l.*) con base en tres modelos de aprendizaje automático optimizados: máquina de soporte vectorial (SVM, support vector machine), de ensamble potenciación del gradiente (HGB, histogram gradient boosting) y una red neuronal artificial, perceptrón multicapa (MLP, multilayer perceptron). Las clases objetivo son V: virus *Tomato yellow leaf curl*, H: hongo *Septoria lycopersici*, A: ácaro *Tetranychus urticae* y S: Hojas sanas.

## ALCANCE Y LIMITACIONES

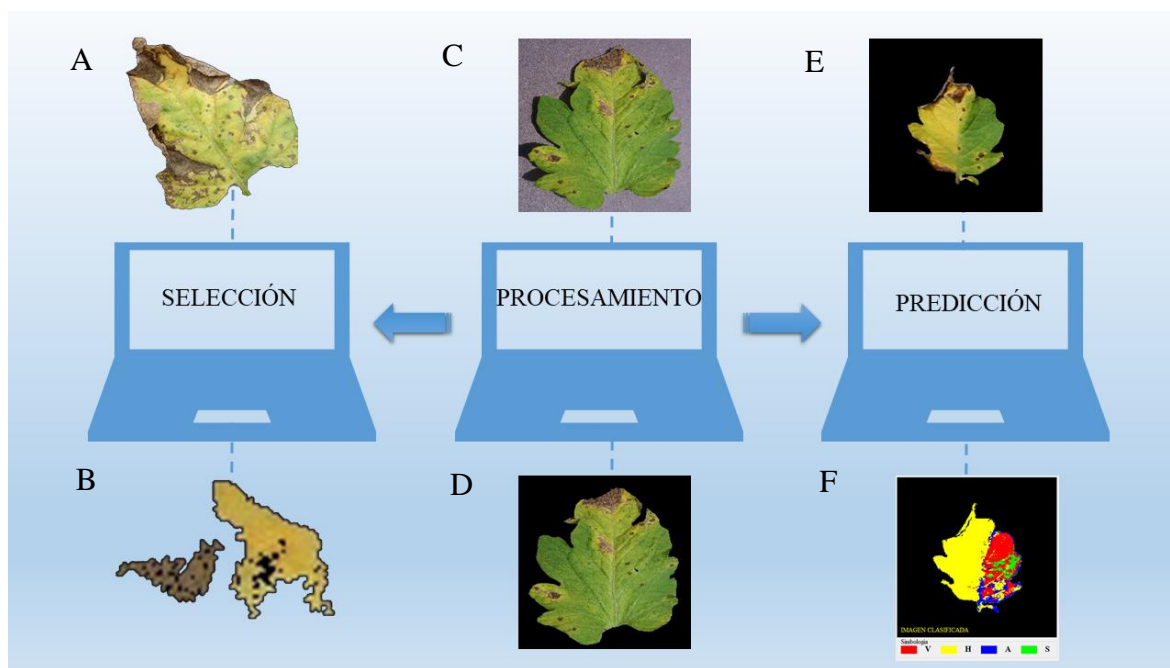
La interfaz gráfica está probada para el uso de imágenes en formato gráfico JPG transformadas al espacio de color RGB y con dimensiones de 256 píxeles por 256 píxeles. La herramienta procesa una imagen o grupos de imágenes en forma secuencial; y habilita tres enfoques de segmentación, dos métodos de selección de muestras y tres modelos de aprendizaje automático entrenados y optimizados para clasificar e identificar tres enfermedades y hojas sanas del cultivo de tomate.

Los clasificadores utilizan como características o variables de entrada, píxeles de imagen transformados al espacio de color HSV y exceso de rojo (ExR). El entrenamiento y evaluación de los modelos, así como los métodos de selección y segmentación se detallan en (Ambrosio, 2021).

## CARACTERÍSTICAS DE IDENTO

IDENTO tiene una interfaz gráfica versátil e intuitiva que consta de tres ventanas de trabajo: **Procesamiento**, **Selección** y **Predicción**. La ventana **Procesamiento** es la principal y facilita el acceso a las ventanas **Selección** y **Predicción**. **Procesamiento** cuenta con diferentes operaciones para segmentar el objeto (hoja) de interés de una imagen de entrada. En la Figura 1C se muestra una imagen de entrada; y una imagen segmentada con el fondo homogéneo (Figura 1D). La ventana **Selección** facilita la selección de una muestra de píxeles con una característica específica de interés. A partir de una imagen de entrada previamente procesada, la selección de una muestra de píxeles se realiza de dos formas: i) selección de crecimiento por región y ii) selección manual; luego el subconjunto de píxeles seleccionados se agrupa y se exporta a un archivo texto (.txt o .csv) (Figura 1 A y B). El subconjunto de píxeles exportados integra el conjunto de datos para entrenar y evaluar el desempeño de los modelos de aprendizaje automático por medio de los módulos habilitados de Scikit Learn (Pedregosa et al., 2011). La ventana **Predicción** facilita la ejecución de los modelos entrenados; la transformación de la imagen preprocesada en un conjunto de características o predictores de entrada; la ejecución de los modelos para realizar la predicción; y por último, la transformación de las respuestas predichas para su descripción gráfica en cuatro categorías de píxeles asociados a, V: virus, H: hongo, A: ácaro y S: hojas sanas. La clasificación e

identificación de una hoja afectada por hongo *Septoria lycopersici* se ilustra en la Figura 1 E y F.



**Figura 1. Descripción de la interfaz gráfica principal de IDENTO representada por tres opciones de ventana: Selección, Procesamiento y Predicción.**

### AMBIENTE DE DESARROLLO

La interfaz gráfica de IDENTO se desarrolló con el diseñador gráfico PyQt5 (PyQt5, 2021). El procesamiento de imágenes, datos y entrenamiento de los modelos de aprendizaje se realizó en un sistema de cómputo bajo ambiente Windows 10 de 64 bits, procesador Intel Core i5 7th Gen @2.50 GHz, 500 SSD, memoria RAM instalada de 16 GB. La selección de hiperparámetros óptimos por medio de una búsqueda por retícula se realizó en una máquina virtual proporcionada por el servidor de Google Colaboratory. La máquina tiene las siguientes especificaciones de GPU: 1. Modelo Nvidia K80/T4, 2. Frecuencia: 0.82/1.59 GHz, 3. Núcleos: 2, 4. Memoria RAM: 12.69 GB, 5. Memoria del disco: 68.35 GB, 6. Año de lanzamiento: 2014/2018. Google Colaboratory ofrece dos modelos de GPU diferentes desde 2019: K80 y T4.

La aplicación IDENTO se ejecuta en diferentes plataformas siempre y cuando se utilice la versión de los módulos que la integran y que se describen en la siguiente sección.

## GUÍA DE INSTALACIÓN

La instalación de IDENTO consiste en proporcionar al usuario las funciones personalizadas y los módulos propios y de terceros para la ejecución correcta del programa.

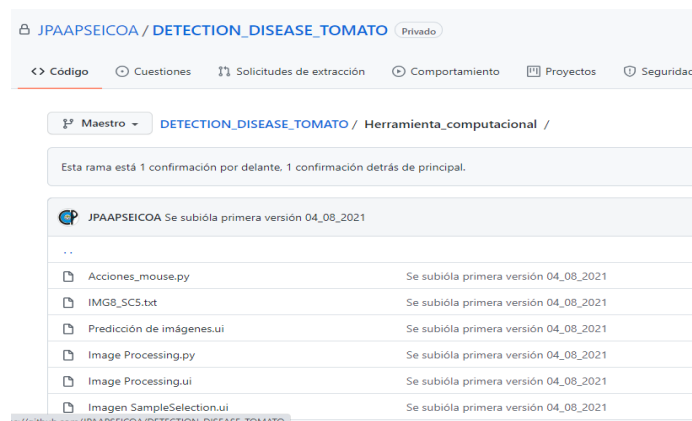
En la Tabla 1 se enlistan los archivos necesarios para la ejecución de la interfaz gráfica de IDENTO y se describe de manera breve su función.

**Tabla 1. Descripción de archivos que son requisito para la correcta ejecución de IDENTO.**

Número	Archivo	Descripción
1	Acciones_mouse.py	Script Python que controla las acciones del mouse (clic) para una selección manual
2	configurationSis.pkl	Archivo que guarda las configuraciones por defecto de algunas funciones que requieren de parámetros previos
3	encabezado_temp.pkl	Archivo que guarda la configuración de la tabla a exportar durante el proceso de muestreo de una imagen de entrada
4	Image Prediction.ui	Diseño gráfico de la ventana para predicción
5	Image Processing.py	Script Python que integra la ventana para procesamiento, de igual forma carga los archivos y módulos necesarios para ejecutar el proceso de segmentación
6	Image Processing.ui	Archivo de diseño gráfico de la ventana para procesamiento
7	Image SampleSelection.ui	Archivo de diseño y configuración gráfica de la ventana de selección de muestras
8	Image Settings.ui	Configuración del diseño de la ventana que guarda las configuraciones del programa
9	Image_Prediction.py	Script Python que permite la integración del diseño y funciones para operar la ventana de predicción

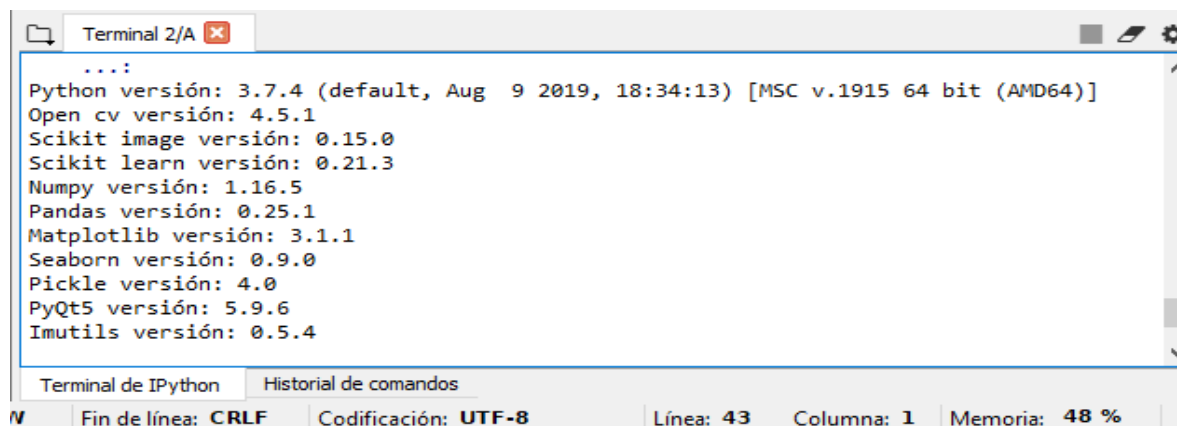
10	Image_Sample_Selection.py	Script Python que integra el diseño de la ventana y las funciones para la selección de muestras
11	Image_setting.py	Script Python que permite cargar la ventana de configuración y manipular los valores que se emplean por defecto
12	Manejador_archivos.py	Script Python que contiene funciones que permite abrir y guardar archivos de imagen y txt
13	model_SVM_FONDO.pkl, model_HGB.pkl, model_MLP.pkl y model_SVM.pk	Archivos que almacenan los modelos de aprendizaje automático optimizado
14	Prediction_background.py	Script Python que permite llevar a cabo una clasificación binaria (0 fondo y 1 hoja) con una máquina de soporte vectorial
15	ROI_by_Manual.py	Script Python que permite llevar a cabo una selección para adición y sustracción de áreas en el proceso de segmentación
16	Selection_manual_temp.png	Archivo en formato gráfico que permite guardar una selección y poder transferir a otras funciones que la requieran
17	Selection_methods.py	Script Python que implementa el método de selección crecimiento por región
18	Selection_sample_manual.py	Script Python que permite hacer la selección manual de muestras a exportar
19	temp_smdo.pkl	Archivo de configuración que guarda resultados de una función y la dispone para un tercero

Estos módulos, archivos, funciones, clases están disponibles en el siguiente repositorio [GitHub](https://github.com/JPAAPSEICOA/DETECTION_DISEASE_TOMATO) [https://github.com/JPAAPSEICOA/DETECTION\\_DISEASE\\_TOMATO](https://github.com/JPAAPSEICOA/DETECTION_DISEASE_TOMATO) con previa solicitud de colaboración al correo: [jpaa.colpos.pseicoa@gmail.com](mailto:jpaa.colpos.pseicoa@gmail.com). La carpeta Herramienta computacional contiene específicamente los archivos necesarios para la ejecución de la interfaz (Figura 2).



**Figura 2. Descripción del repositorio GitHub para la descarga de los archivos requisito para la correcta ejecución de IDENTO.**

Para ejecutar IDENTO es necesario descargar los archivos descritos previamente y almacenarlos en una carpeta de trabajo. Posteriormente instalar las versiones indicadas de las librerías y módulos (Figura 3). Por último, desde un editor de código Python (Spider, por ejemplo) se ejecuta el script principal **Image Processing.py** (número 5 de la Tabla 1). Si la instalación es correcta se visualiza la ventana de trabajo mostrada en la Figura 4.



**Figura 3. Versiones requeridas de módulos para ejecutar la herramienta IDENTO.**



## PROCESAMIENTO DE IMÁGENES

La ventana **Procesamiento** facilita la segmentación de una hoja con un fondo homogéneo en el espacio de color RGB (0, 0, 0). Esta ventana tiene tres menús de trabajo y ocho salidas de visualización de imágenes (Figura 4)



**Figura 4. Descripción de la ventana Procesamiento para realizar la segmentación de objetos de interés, se ejecuta el script Image Processing.py.**

La segmentación tiene tres opciones; opción 1 y opción 2 utilizan la imagen transformada al espacio de color HSV y exceso de rojo (ExR). La opción 3 utiliza un modelo de clasificación binaria (0: fondo, 1: hoja); esto es una máquina de soporte vectorial por medio de las variables predictoras: coordenadas del pixel (y, x), espacio de color HSV tripleta (H, S, V) y exceso de rojo (ExR).

La opción **File** de la barra de tareas gestiona las opciones de trabajo. **Open File** y **Save File** abren y/o guardan una imagen. **Open Files** manipula imágenes por lotes o grupos; abre un conjunto de imágenes y almacena los directorios en una lista. **Next File** procesa una imagen de la lista y **Save It** guarda una imagen de la lista. Este proceso se repite hasta segmentar todas las imágenes de la lista. [Descargar video](#).

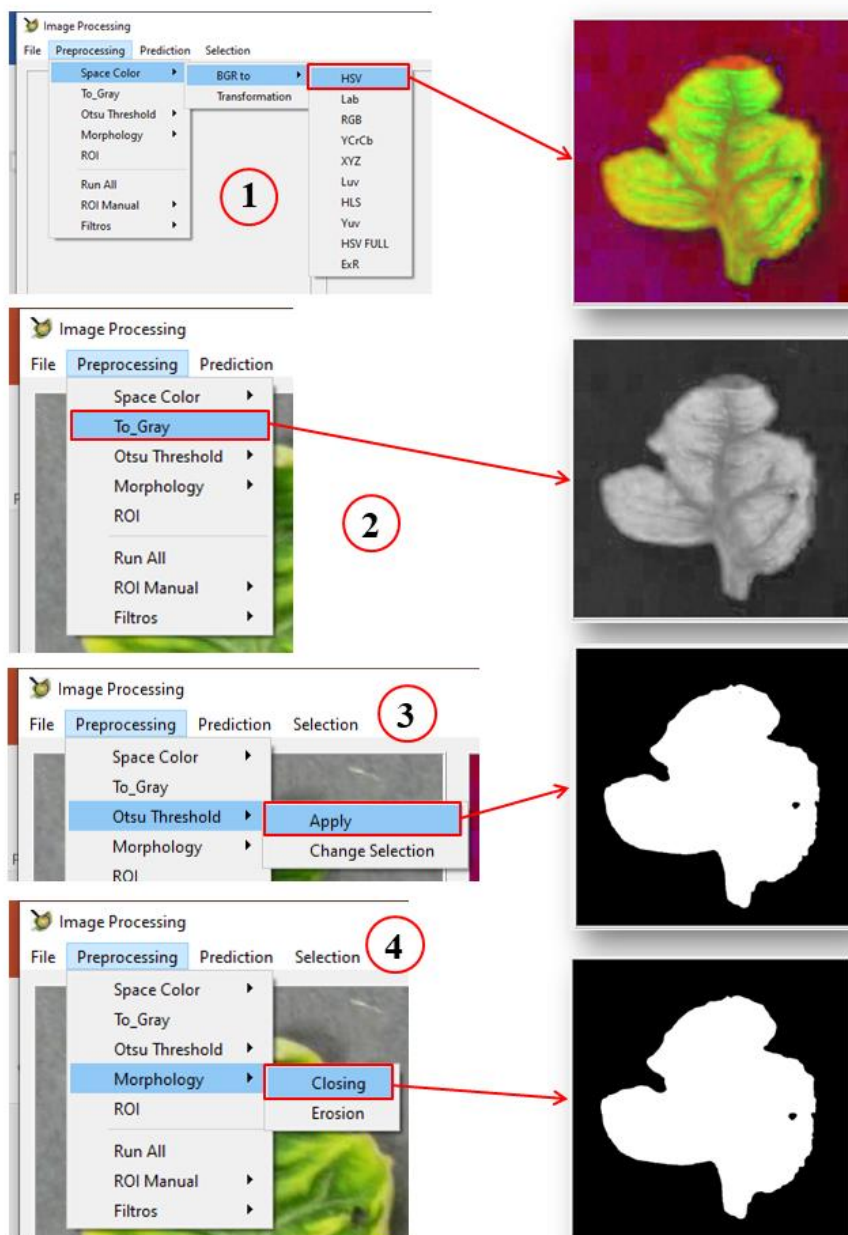
### ESPACIO DE COLOR HSV

El flujo de trabajo se puede visualizar a través de las Figuras 5 y 6. El proceso inicia con la apertura de un nuevo archivo mediante las funciones habilitadas en la opción **File**. La segmentación se lleva a cabo como sigue: 1. Se transforma la imagen RGB al espacio de color HSV de tripletas (H, S, V). 2. Se transforma la imagen del paso anterior a través de la ecuación  $I_t = 0.2989 \times H + 0.5870 \times S + 0.1140 V$ , donde  $I_t$  es la imagen transformada de un solo canal. 3. Se analiza el histograma y mediante el método de Otsu se determina el umbral óptimo para la segmentación binaria (0: fondo negro, 1: hoja, blanco).

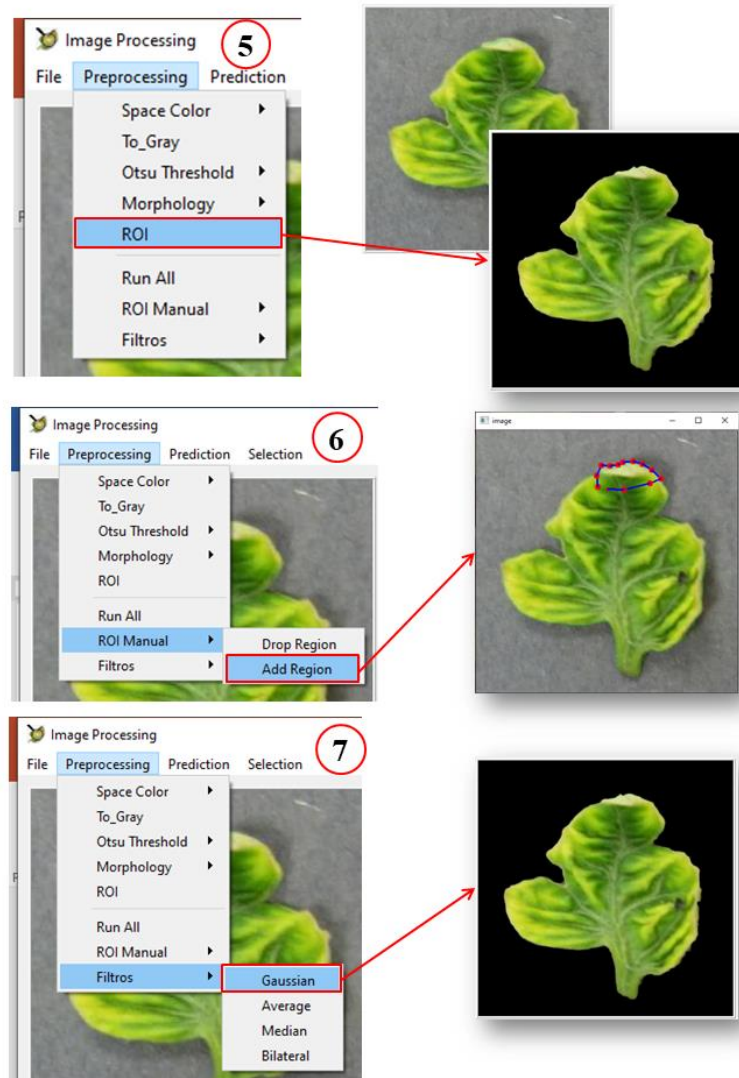
En ocasiones por la naturaleza de la imagen la selección resulta ser inversa, por tal motivo se tiene la opción **Change Selection** para intercambiar la selección binaria. 4. En este paso se llevan a cabo dos operaciones morfológicas para mejorar la segmentación. El cierre (**Closing**) permite adicionar pixeles que son 0, pero que forman parte de la hoja y erosión (**Erosion**) permite quitar aquellos pixeles que no son parte de la hoja. 5. Se detectan los bordes externos de la imagen binaria y se crea una máscara con los valores (255, 255, 255) que permite recuperar los pixeles originales RGB por diferencia de matrices. 6.

6. El submenú **ROI Manual** permite agregar (**Add Region**) y quitar (**Drop Region**) áreas de interés que no fueron procesadas correctamente. Esta operación se lleva a cabo en una ventana emergente que redimensiona la imagen para su correcta visualización. Para la selección manual se usa clic izquierdo para adicionar un punto, clic derecho para eliminar el último punto, y scroll (presionado) para finalizar, seguido la tecla Esc (Escape) para cerrar la ventana de edición. 7. La opción **Filters** le permite agregar filtros (Gaussiana, promedio, mediana y bilateral) a la selección final en caso de necesitarse.

Note que el flujo de trabajo es secuencial y adicionalmente se muestran las salidas de cada operación en los diferentes paneles de visualización que tiene provista **IDENTO**. Alternativamente la función **Run All** permite correr todas las funciones del proceso de manera automática.



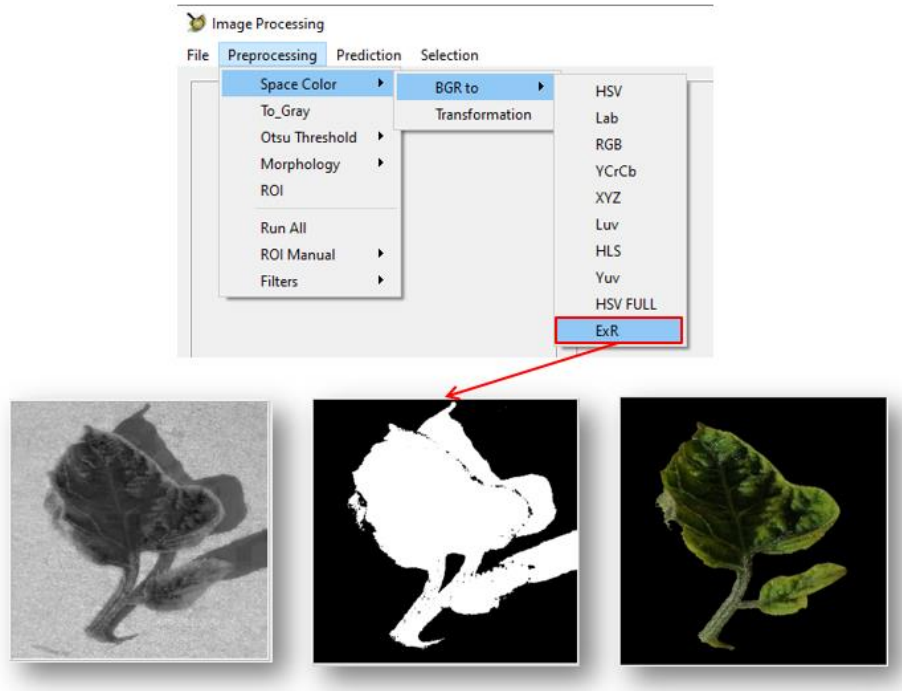
**Figura 5. Etapas de procesamiento de imágenes RGB con transformación al espacio de color HSV.**



**Figura 6. Etapas de segmentación de hojas con base en el espacio de color HSV (continuación).**

### **ESPACIO DE COLOR ExR**

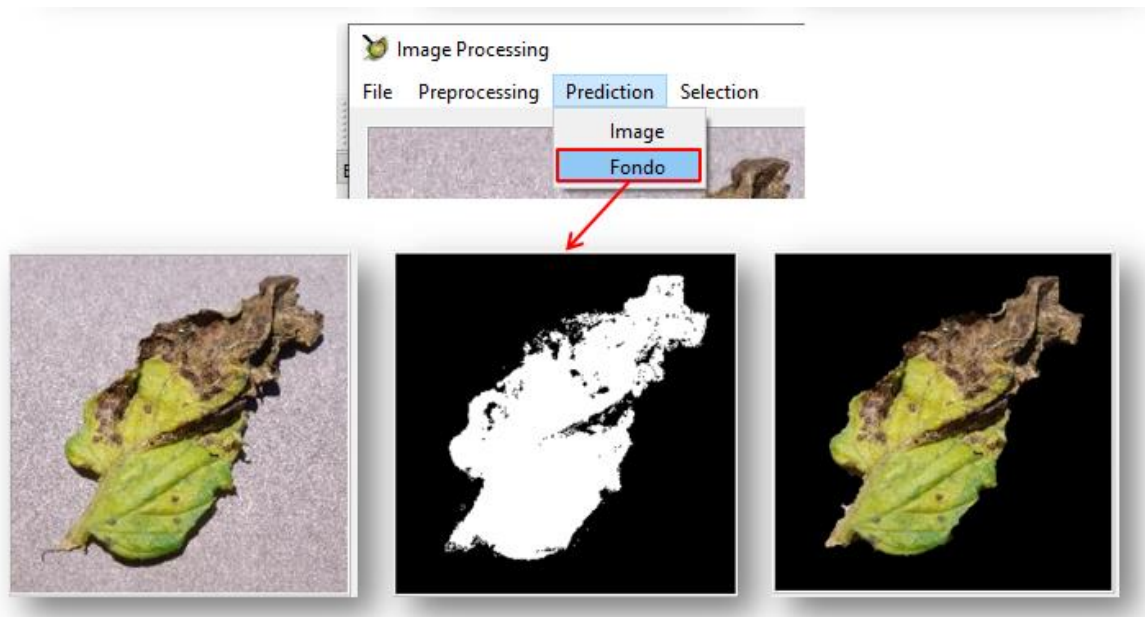
El proceso de segmentación es exactamente igual al descrito en la sección anterior (Figura 5 y 6). La diferencia es que en lugar de HSV se emplea la imagen de un solo canal que representa el exceso de rojo (Figura 7). Esta alternativa es un complemento al método anterior ya que permite en algunas imágenes desarrollar una mejor segmentación. Al ejecutar ExR automáticamente ejecuta el resto de las funciones secuencialmente, sin embargo, se puede usar las funciones de **ROI Manual** para mejorar el resultado de la segmentación.



**Figura 7. Segmentación de la imagen RGB al espacio de color ExR.**

## **MÁQUINA DE SOPORTE VECTORIAL**

Esta opción implementa una máquina de soporte vectorial con previo entrenamiento, validación y prueba. El clasificador binario (0: fondo negro, 1: hoja, blanco) usa las variables o características de entrada: X, Y, H, S, V, y ExR para realizar la predicción de un nuevo pixel. El proceso de trabajo es similar al espacio de color HSV (Figura 5 y 6), a diferencia que la binarización Otsu se reemplaza por el clasificador SVM y se sigue con los pasos 4, 5, 6 y 7 respectivamente. Este método es más lento que las dos anteriores, debido a la predicción que se lleva a cabo a nivel pixel y si este número se incrementa el tiempo de procesamiento lo hace de igual forma.

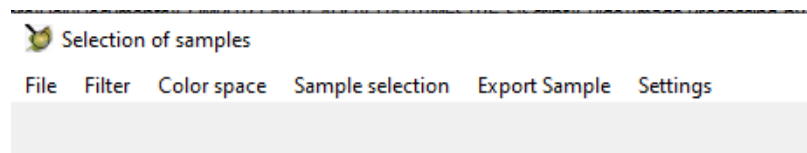


**Figura 8. Segmentación de la hoja empleando un clasificador binario basado en una máquina de soporte vectorial.**

### SELECCIÓN DE MUESTRAS

El objetivo de esta ventana es permitir extraer muestras que permiten entrenar, validar y probar modelos de aprendizaje automático. Para acceder a ella es necesario ejecutar el comando **Selection>Samples** desde la venta de **Procesamiento de imágenes**. Cuenta con seis menús de trabajo (Figura 9). El flujo de trabajo es secuencial, primero se abre una imagen en formato grafico (File), luego se le aplica un filtro en caso de ser necesario (**Filter**). Se dispone de filtros: gaussiana, promedio y mediana, además de restablecer la imagen y exportar en formato grafico de preferencia. Posteriormente se selecciona el conjunto de características que se desea exportar (**Color space**), luego se selecciona la muestra por el método de crecimiento por región o selección manual (**Sample selection**) y por último se exporta el muestreo en formato txt o csv (**Export Sample**). La función **Settings** permite configurar la disimilaridad que se emplea para el proceso de selección con crecimiento por región.

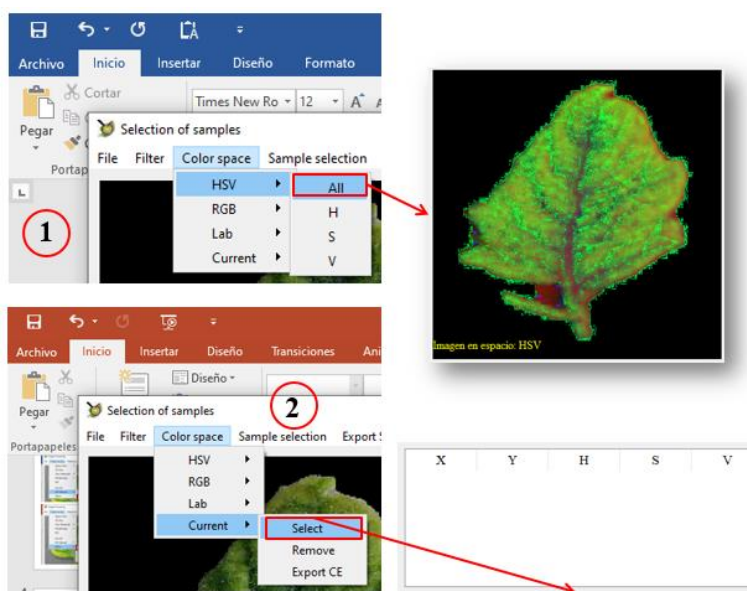
[Descargar video.](#)



**Figura 9. Barra de tareas de la ventana selección de muestras.**

## VARIABLES PREDICTORAS

Las variables predictoras o características de entrada (píxeles de color) se exportan a los siguientes espacios de color: espacio de color HSV (completo o por canal), espacio de color RGB (tripleta), RGB normalizado (tres canales), exceso de rojo (ExR), exceso de verde (ExG), la diferencia de las dos anteriores (ExG-ExR), CIVE y espacio de color Lab (completo o por canal). Una vez que se selecciona un espacio de color, este se muestra en el segundo panel de visualización (Figura 10.1), posteriormente mediante el menú **Current** se puede adicionar (**Select**) la selección (Figura 10.2) o quitar (**Remove**) y en caso de ser necesario exportar (**Export CE**) en formato gráfico de preferencia (jpg, png). Se muestra en la ventana una tabla con los encabezados actualizados de las variables seleccionadas actualmente.



**Figura 10. Selección de variables predictoras a exportar, 1. Selección del espacio de color y 2. Selección actual.**

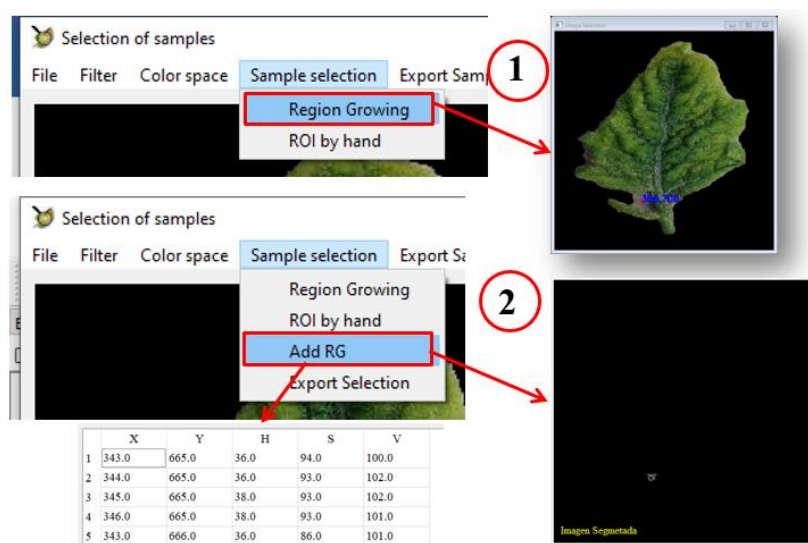
## CRECIMIENTO POR REGIÓN

Este método de selección consiste en definir una semilla de arranque, luego el algoritmo compara los píxeles vecinos (6 o 8) con la semilla y mediante un valor de disimilaridad decide que píxeles son parte de la región, y la define como la nueva semilla. Este proceso se repite hasta que ya no exista píxeles vecinos similares a la región o se ha recorrido la imagen en su totalidad.



La opción **Region Growing** permite hacer la selección con este enfoque. Al hacer clic, la aplicación abre una ventana emergente en la que se pide la semilla (Figura 11.1), con la ayuda del mouse (clic izquierdo) se puede definir un punto de la región de interés y de manera inmediata se muestran las coordenadas del punto seleccionado. Se puede seleccionar las veces que sea necesario el punto y una vez que se tenga certeza, utilizar la tecla Escape (Esc) para cerrar la ventana y ejecutar el algoritmo. Los resultados del algoritmo se muestran a través de una imagen RGB en el tercer panel de visualización y las diez primeras filas de los valores de las variables predictoras seleccionadas en la sección anterior (Figura 11.2). Después de la primera ejecución del algoritmo se activan dos opciones de gran ayuda, la primera permite ejecutar de nuevo el algoritmo y el resultado se adiciona a la selección anterior y la segunda permite exportar la selección actual en formato grafico de preferencia (JPG, png).

En el menú Settings se puede configurar el valor de disimilaridad (0.45 por defecto), un valor cercano a 1 permite mayor diferencia, por lo tanto, mayor número de pixeles serán considerados parte de la región. Los valores más usados varían entre 0.3 y 0.6 en función de la homogeneidad de la región a muestrear.

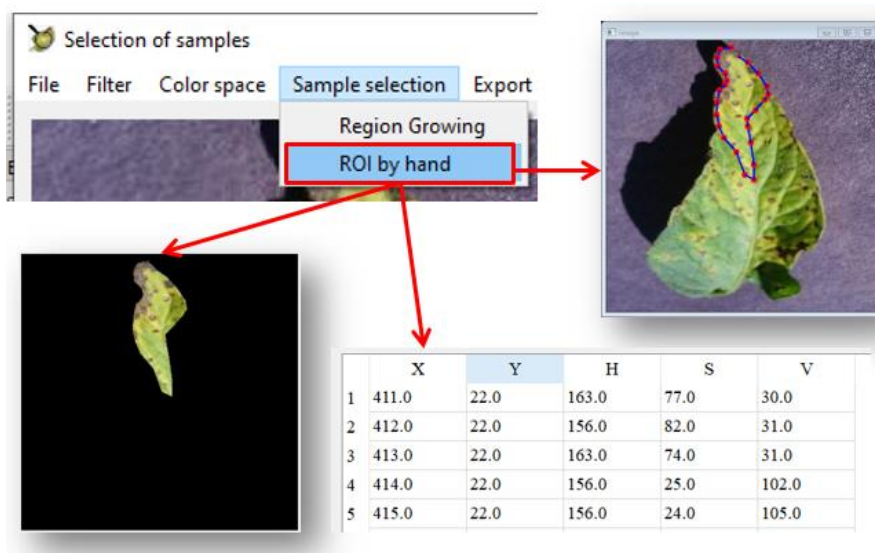


**Figura 11. Selección de muestras mediante el método crecimiento por región: 1. Selección del pixel semilla por coordenadas (x, y), 2. Se muestran los resultados de la selección, adicionalmente se puede agregar una nueva selección.**



## SELECCIÓN MANUAL

La Selección **manual** es similar al mostrado en la sección de procesamiento de imágenes. La selección se lleva a cabo en una ventana emergente, clic izquierdo para adicionar un punto, clic derecho para eliminar el último punto y scroll (presionado) para finalizar. Note que la selección no debe cerrar completamente ya que el algoritmo internamente cierra el polígono uniendo el primer y último punto seleccionado. Los resultados se muestran en una imagen segmentada RGB y una tabla que muestra los primeros diez filas de valores.



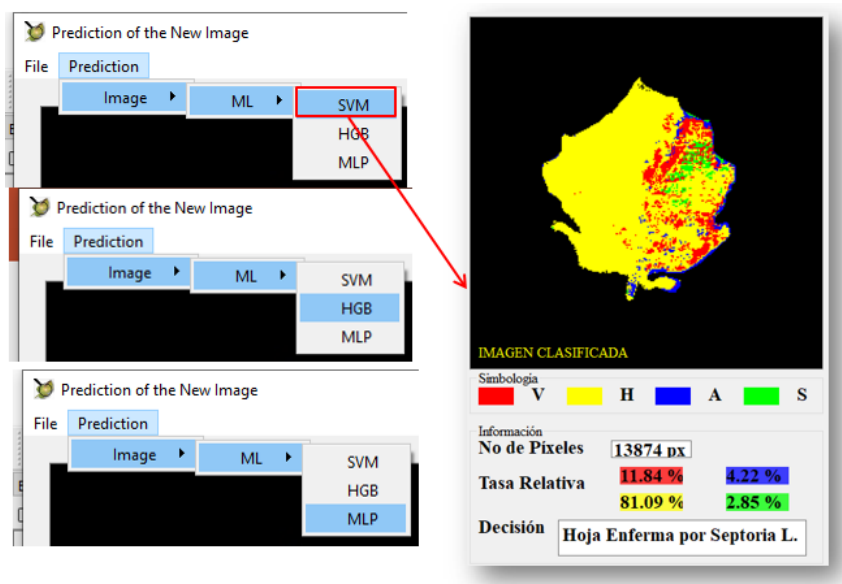
**Figura 12. Selección manual de la región de interés con la visualización de los resultados en formato de imagen y de tabla.**

En el menú **Export Sample>to txt** se puede exportar en formato txt o csv delimitado por coma el resultado de la selección.

## PREDICCIÓN

La predicción se lleva a cabo sobre una imagen preprocesada con fondo homogéneo. El menú **File** permite definir la forma de trabajo, procesar únicamente un archivo o un conjunto de imágenes de la misma manera que en la ventana principal de procesamiento de imágenes. La opción **Prediction>Image>ML** permite seleccionar el modelo con la que se lleva a cabo la clasificación multiclase: máquina de soporte vectorial (SVM), de ensamble potenciación del gradiente (HGB) y perceptrón multicapa (MLP). La salida es una imagen con simbología de colores que permite visualizar la distribución de píxeles para cada clase

(Figura 13). Además, en etiquetas se muestran la tasa relativa por clase y mediante reglas heurísticas establecidas mediante el análisis de imágenes se identifica la enfermedad de mayor impacto en la hoja. [Descargar video](#).



**Figura 13. Clasificación e identificación de tres enfermedades y hoja sana en el cultivo de tomate a través de tres modelos de aprendizaje automático: SVM, HGB y MLP**

### SOPORTE TÉCNICO

El manual pretende cubrir todas las etapas para el correcto funcionamiento de IDENTO. Sin embargo, por alguna circunstancia de configuración del ambiente de trabajo o razón extraordinaria, éste puede presentar inconvenientes. En caso necesario contactar al autor al correo electrónico [jpaa.colpos.pseicoa@gmail.com](mailto:jpaa.colpos.pseicoa@gmail.com) para atender sus preguntas o dudas.

## **CONCLUSIONES**

IDENTO V1.0 desarrolla una solución a un problema en específico, sin embargo, incorpora el paradigma de aprendizaje automático que permite escalar la solución al problema complejo de detección de enfermedades en cultivos agrícolas. Debido a que se dispone al usuario los módulos y dependencias, se vuelve una herramienta muy potente, ya que se pueden modificar, y adicionar mejoras en los modelos con facilidad.

Esto permite que futuros investigadores con proyectos similares colaboren y generen una versión más robusta y de mayor generalidad para atender el diagnóstico de enfermedades con la visión artificial. IDENTO V1.0 permite mostrar la potencia y versatilidad del lenguaje Python para el desarrollo de software.

## **REFERENCIAS BIBLIOGRÁFICAS**

- Ambrosio, J.P. 2021. Modelos De Inteligencia Artificial Para Identificación De Enfermedades En Hojas De Tomate A Partir De Imágenes Digitales. Tesis de grado. Postgrado En Socioeconomía, Estadística, E Informática. Colegio de Postgraduados, Montecillo, México.
- Pedregosa, F. and et al. 2011. Scikit-learn: Machine Learning in Python. JMLR 12. 2825-2830 p.
- PyQt5. 2021. copyright (c) Riverbank Computing Limited. Disponible en: <https://www.riverbankcomputing.com/software/pyqt/> (Consulta: Septiembre 2021).