



Instituto Politécnico Nacional

Escuela Superior de Cómputo



Prototipo de aplicación para la detección de deficiencia
de nutrientes en cultivos de hidroponía.

Presenta:

Edgar Rodrigo Arredondo Basurto

Directores

Ing. Eduardo Gutiérrez Aldana

Dr. José Félix Serrano Talamantes

1. Identificación del problema

- La hidroponía permite cultivar especies para el consumo humano en regiones donde no existen tierras de cultivo.
- Los nutrientes se proporcionan disueltos en agua.
- Si existe una deficiencia de algún nutriente en la solución, se genera una anomalía dentro de la estructura fisiológica de la planta.



Figura 1.1 Daño causado por deficiencia de calcio.



Figura 1.2 Daño causado por deficiencia de potasio.

2. Objetivos

Objetivo general

Diseñar y desarrollar el prototipo de una aplicación de visión por computadora que analiza imágenes de hojas de tomate con una anomalía visible y realiza un diagnóstico de una posible deficiencia de nutrientes, en un subconjunto definido de nutrientes.



Figura 2.1. Deficiencia de magnesio



Figura 2.2. Deficiencia de potasio



Figura 2.3. Deficiencia de nitrógeno



Figura 2.4. Deficiencia de azufre

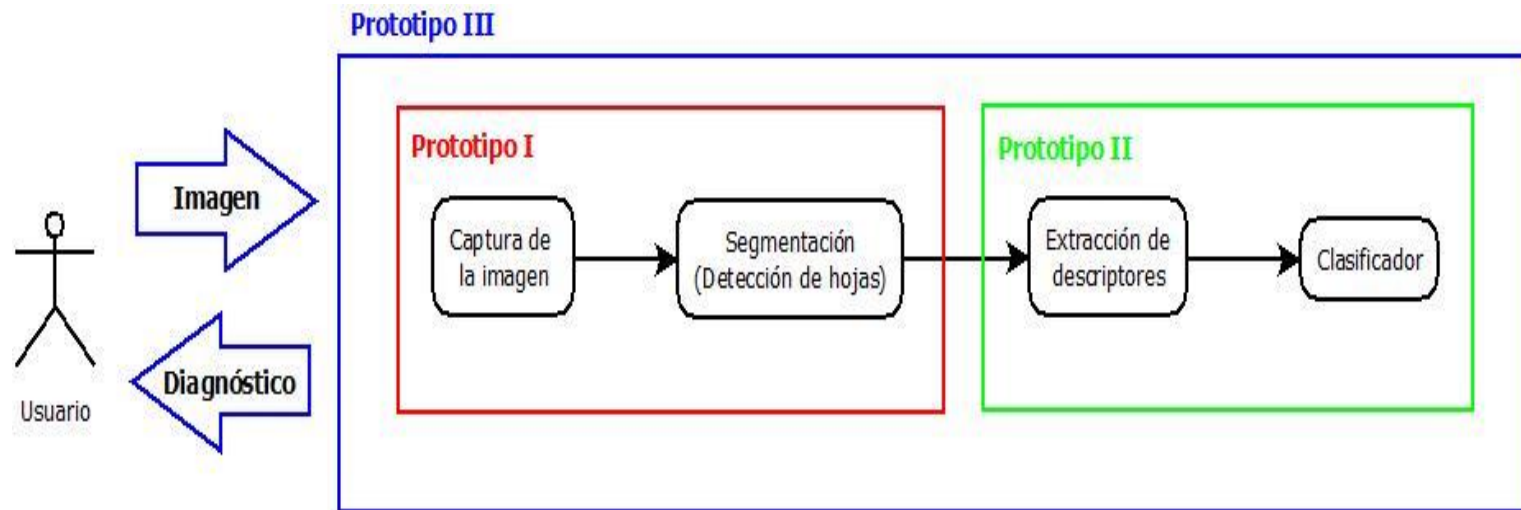
2. Objetivos

Objetivos particulares



- Definir y diseñar el proceso de segmentación de una hoja, sin considerar restricciones en el fondo de la imagen.
- Definir los descriptores adecuados para la clasificación de las hojas y diseñar el método de extracción de los mismos.
- Análisis y selección del lenguaje y entorno de desarrollo de la aplicación para el usuario.

3. Resumen



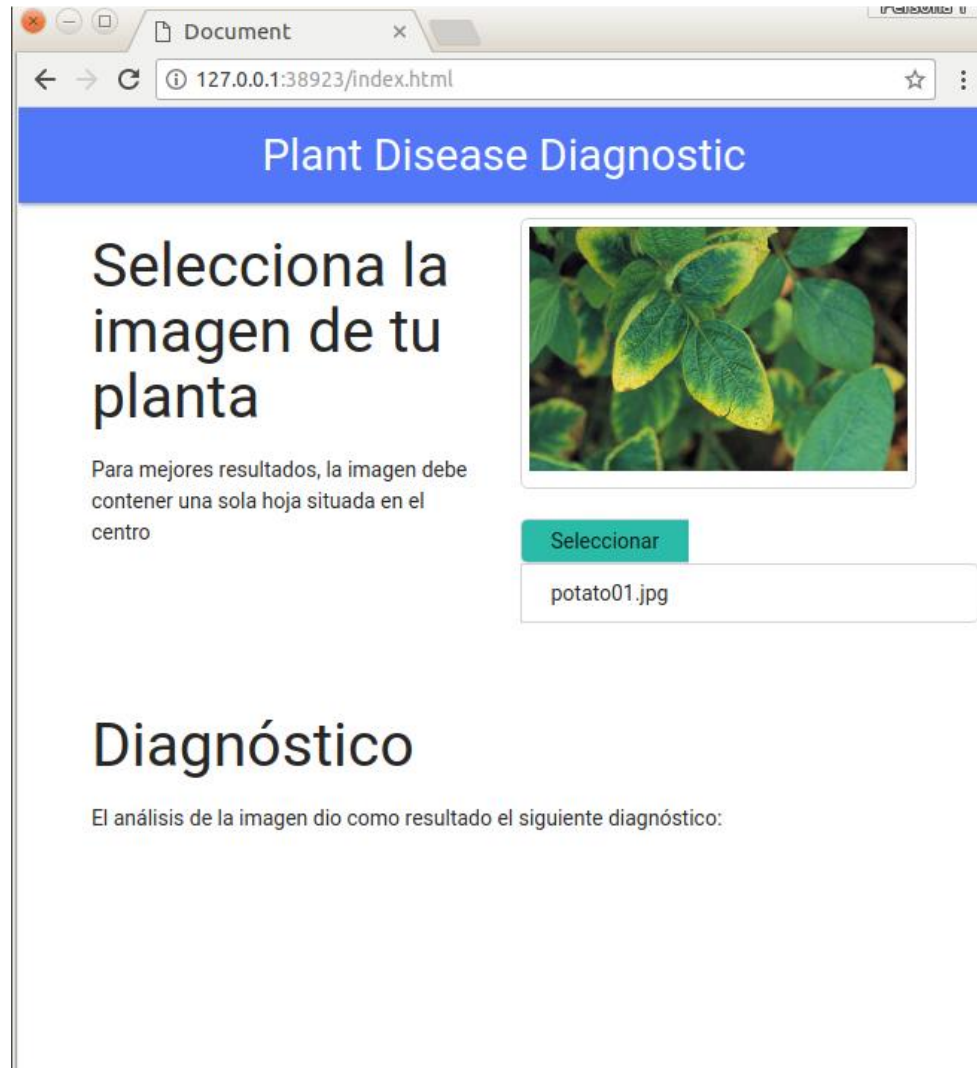


Figura 3.1. Diseño preliminar

4. Captura y segmentación de la imagen.

La imagen de entrada tiene las siguientes restricciones.

- Contener hojas de planta de tomate (figura 4.1).
- Capturar preferentemente solo una hoja con síntomas de la deficiencia (figura 4.2).



Figura 4.1. Forma de hoja de una planta de tomate.



Figura 4.2. Ejemplo de imagen de entrada.

4. Captura y segmentación de la imagen.

Los objetos de interés son las hojas, por lo que deben segmentarse del resto de la imagen (figura 4.3).

- Para segmentar las hojas se emplea un algoritmo de detección de objetos.
- Se optó por una Máquina de Vectores de Soporte (SVM) entrenada con descriptores HOG* (Histograma de Gradientes Orientados) [3].

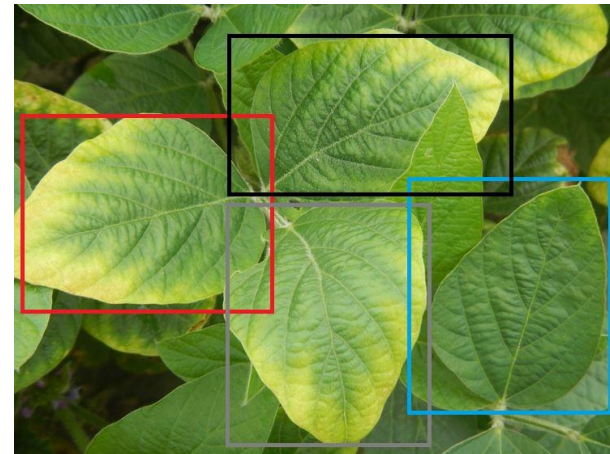
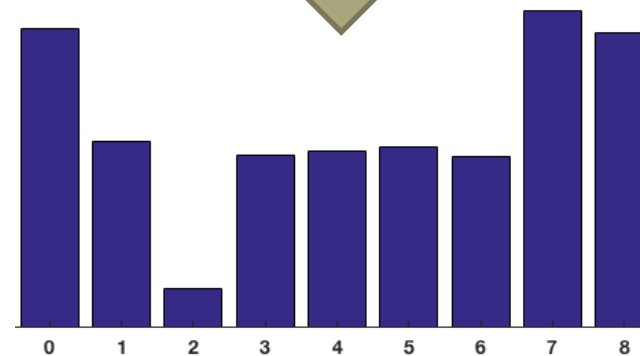
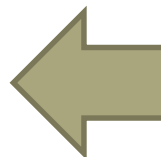
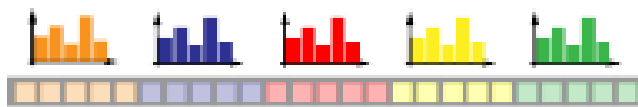
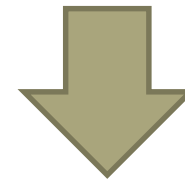
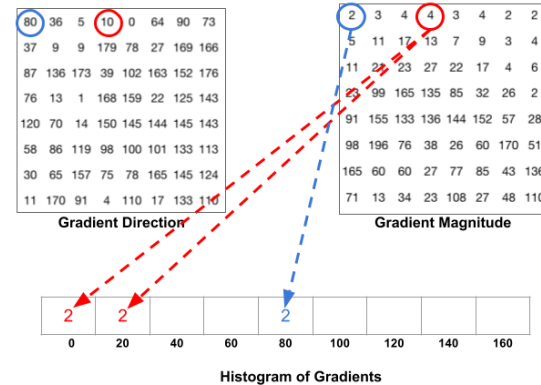
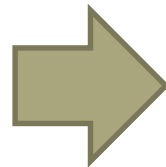
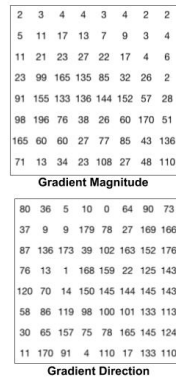
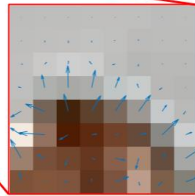
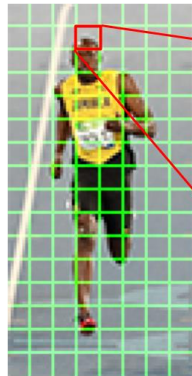


Figura 4.3. Detección de hojas.

* Dalal, Triggs. *Histograms of Oriented Gradients for Human Detection*, 2005

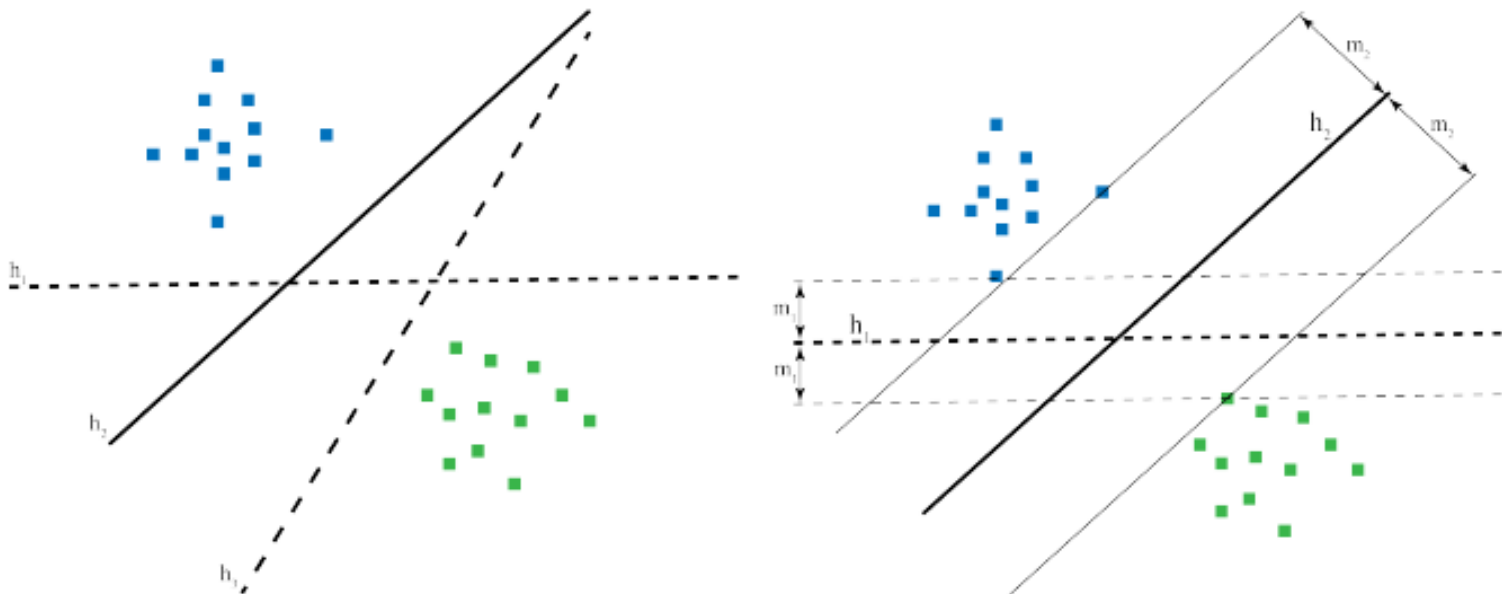
4. Captura y segmentación de la imagen.

Cálculo del descriptor HOG



4. Captura y segmentación de la imagen.

Clasificador SVM



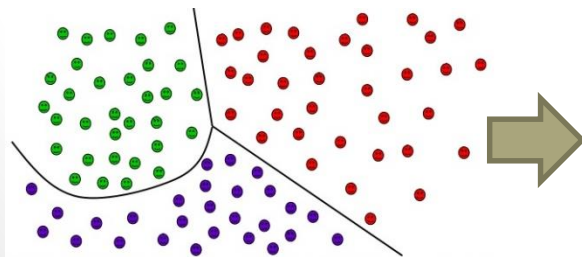
5. Descriptor y clasificación de la hoja

Descriptores de textura de Haralick

- Momento angular
- Contraste
- Correlación
- Varianza
- Momento de diferencias inverso
- Suma de promedios
- Suma de varianzas
- Suma de entropías
- Entropía
- Diferencia de varianzas
- Diferencia de entropías
- Medida de correlación 1
- Medida de correlación 2
- Máximo coeficiente de correlación

5. Descriptor y clasificación de la hoja

- Las clases en las que una hoja puede ser clasificada son: 1) Hoja sana, 2) Hoja con deficiencia de potasio y 3) Hoja deficiencia de nitrógeno/azufre, 4) Hoja con deficiencia de magnesio y 5) Hoja con deficiencia no identificada.
- El clasificador será seleccionado con base en pruebas de eficiencia durante la implementación del segundo prototipo.



Clasificador

	C1	C2	C3
C1	195	4	1
C2	20	150	30
C3	0	0	200

6. Implementación

La funcionalidad relativa al sistema de visión por computadora será implementada con la biblioteca de OpenCV 3, que ofrece soporte para C++, java y python.

Resultado de un análisis de tiempo de ejecución, el resto de la implementación se realizará en C++ (tabla 6.1).

Lenguaje de programación	Java	Python	C++
Tiempo de respuesta [ms]	188.8	195.58	112.48

Tabla 6.1 Comparativa en el desempeño de un mismo detector de objetos implementado en diferentes lenguajes de programación.

Características de la prueba:

- Imagen de 500x374 px
- SO Ubuntu 16.04 64 bits
- Intel Core i3 2.53 GHz
- RAM 4 GB

6. Implementación

La presentación del prototipo será como un servidor en una red local mediante la arquitectura de la figura 6.2.

Los casos de uso del usuario son dos, seleccionar imagen y obtener el diagnóstico (figura 6.3).



Figura 6.2 Arquitectura del sistema.

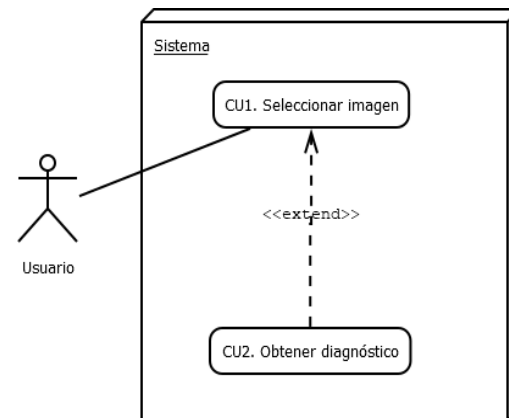


Figura 6.3 Casos de uso.

6. Implementación

La interfaz web propuesta se muestra en la figura 6.4.

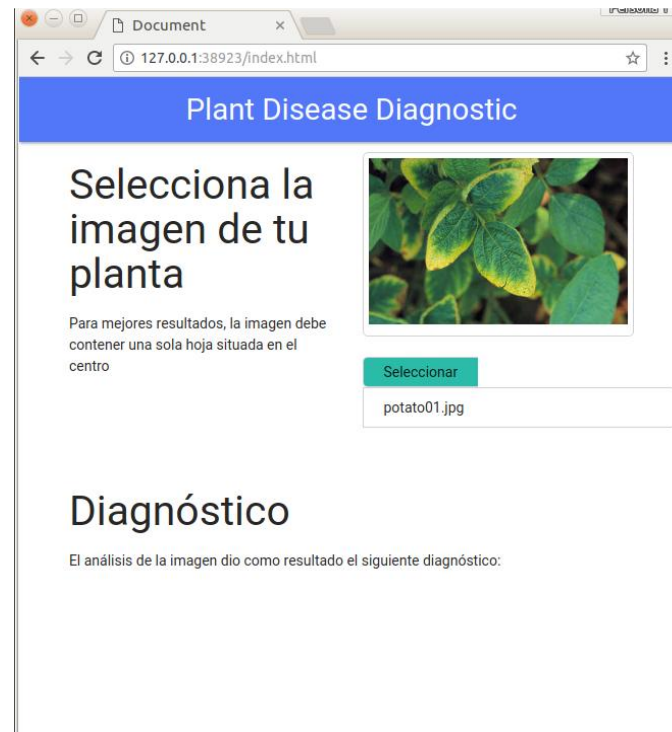


Figura 6.4. Interfaz web propuesta.

7. Referencias

- [1] Daren Mueller, Iowa State University, Bugwood.org.
- [2] Comercializadora Hydro Environment S.A. DE C.V. Deficiencia de los nutrientes en las plantas. [En línea] Disponible en: <http://hydroenv.com.mx/>
- [3] Rosenbrock A. Histogram of Oriented Gradients and Object Detection. Pyimagesearch. [En línea] Disponible en <http://www.pyimagesearch.com/>
- [4] Cruza, J., Elcio, H., Shiguemorib y Lamartine N. F. Guimaraes. A comparison of Haar-like, LBP and HOG approaches to concrete and asphalt runway detection in high resolution imagery. 2015.
- [5] Mallick, S. Histogram of Oriented Gradients. [En línea] Disponible en: <https://www.learnopencv.com/>

Gracias