



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

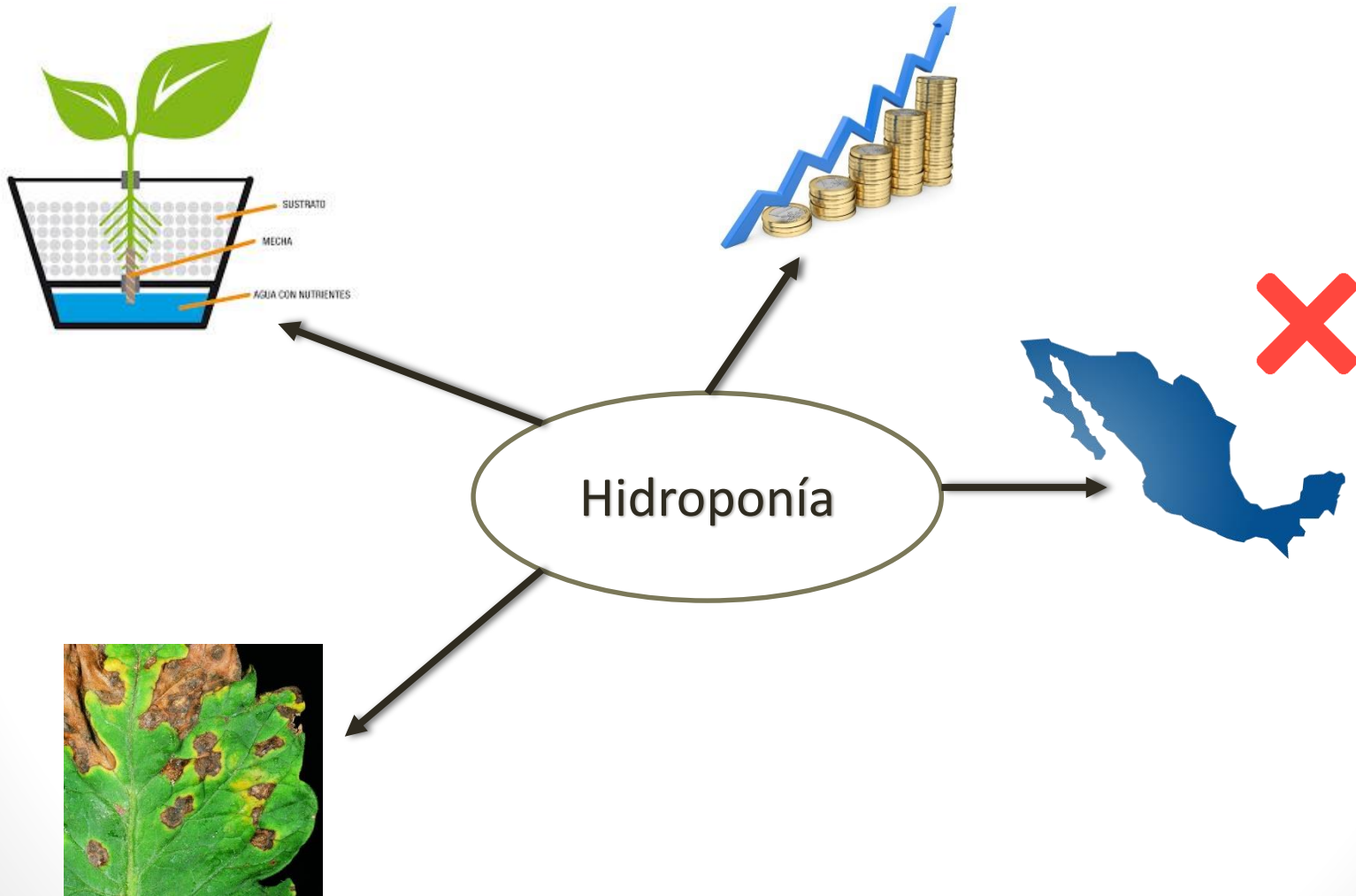
Ing. Eduardo Gutiérrez Aldana

Dr. José Félix Serrano Talamantes

I. Contenido

1. Identificación del problema
2. Objetivos
3. Introducción
4. Clasificación de enfermedades
5. Sistema Web
6. Pruebas del sistema
7. Conclusiones
8. Trabajo a futuro
9. Referencias

1. Identificación del problema



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 enfermedad, bajo un subconjunto predefinido de enfermedades del tomate

2. Objetivos

Objetivos particulares

- Entrenar el modelo de clasificación de imágenes con el conjunto de datos predefinido.
- Realizar pruebas de eficiencia del clasificador, obteniendo un resultado superior al 90%.
- Implementar un sistema web en el que se aloje el clasificador y permita a los usuarios realizar identificaciones de enfermedades.

2. Objetivos

Clase	Número de imágenes
Virus del rizado amarillo del tomate (<i>Tomato yellow leaf curl virus</i>)	4032
Virus del mosaico del tomate (<i>Tomato mosaic virus</i>)	325
Corynespora cassiicola. Mancha en forma de blanco (<i>Target spot</i>)	1,356
Araña roja (<i>Spider mites</i>)	1,628
Septoriosis (<i>Septoria spot</i>)	1,723
Passalora fulva. Moho en la hoja (<i>Leaf mold</i>)	904
Tizón tardío (<i>Lateblight</i>)	1,781
Tizón temprano (<i>Earlyblight</i>)	952
Mancha bacteriana (<i>Bacterial spot</i>)	2,127
Hoja sana	1,591
Total	16,419

Figura 2.9. Conjunto de datos.

3. Introducción

3.1 Metodología de desarrollo

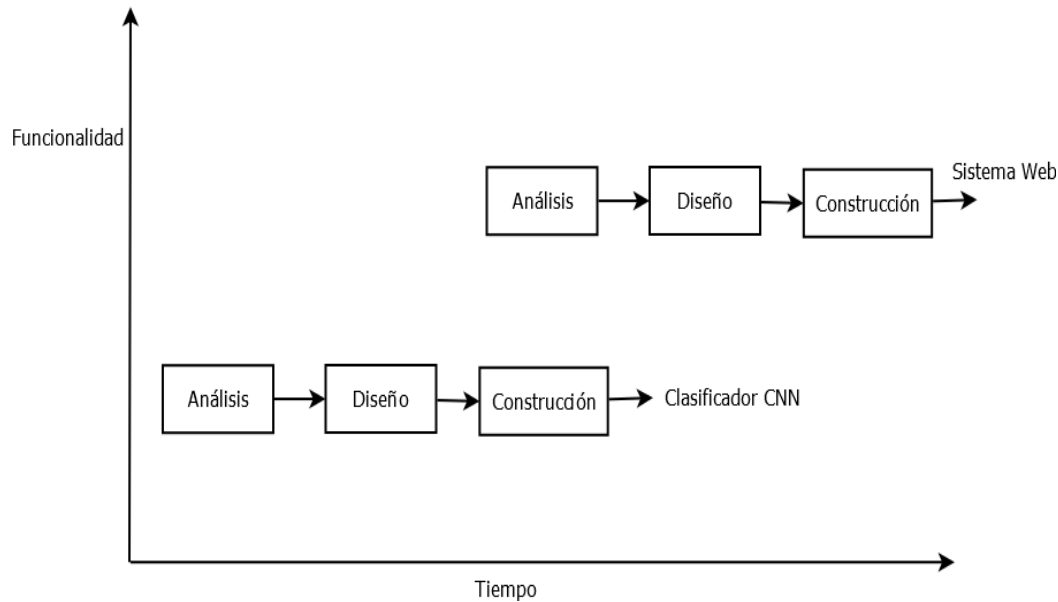


Figura 3.1. Metodología.

3. Introducción

3.2 Enfermedades identificables

- Virus del rizado amarillo del tomate.
- Virus del mosaico del tomate.
- *Corynespora cassiicola*. Mancha en forma de blanco.
- Araña roja.
- Septoriosis.
- *Passalora fulva*. Moho en la hoja.
- Tizón tardío.
- Tizón temprano.
- Mancha bacteriana.

3. Introducción

3.2 Enfermedades identificables



Figura 3.2. Virus del rizado amarillo.



Figura 3.3. Mancha en forma de blanco.



Figura 3.4. Septoriosis.



Figura 3.5. Tizón tardío.

3. Introducción

3.3 Requisitos

Identificador	Tipo	Nombre	Descripción
RF01	Funcional	Selección de imagen	El sistema permitirá seleccionar una imagen del sistema de archivos local del dispositivo del usuario.
RF02	Funcional	Clasificación	El sistema clasificará la imagen, indicando la probabilidad de que el resultado sea el correcto.
RNF01	No funcional	Eficiencia de clasificación	La eficiencia de clasificación deberá ser superior al 90%.
RNF02	No funcional	Tiempo de clasificación	El clasificador identificará la clase de la imagen de entrada en un tiempo no mayor a cinco segundos.
RNF03	No funcional	Ambiente web	El sistema será desarrollado en un ambiente Web.
RNF04	No funcional	Estándares de diseño de la interfaz web	El sistema seguirá los estándares CSS3, XHTML 1.0 Transitional y ECMAScript 5 para el desarrollo de la interfaz gráfica.

3. Introducción

3.4 Algoritmo para la clasificación de enfermedades

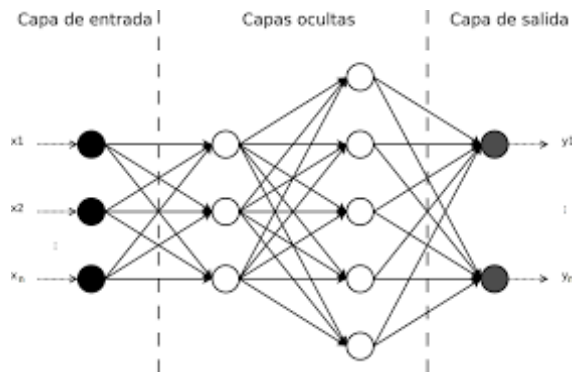


Figura 3.6. Red neuronal unidireccional.

IMAGENET

- LeNet (1990).
- AlexNet (2012)
- ZFNet (2013)
- GoogLeNet (2014)

3. Introducción

3.5 Sistema Web

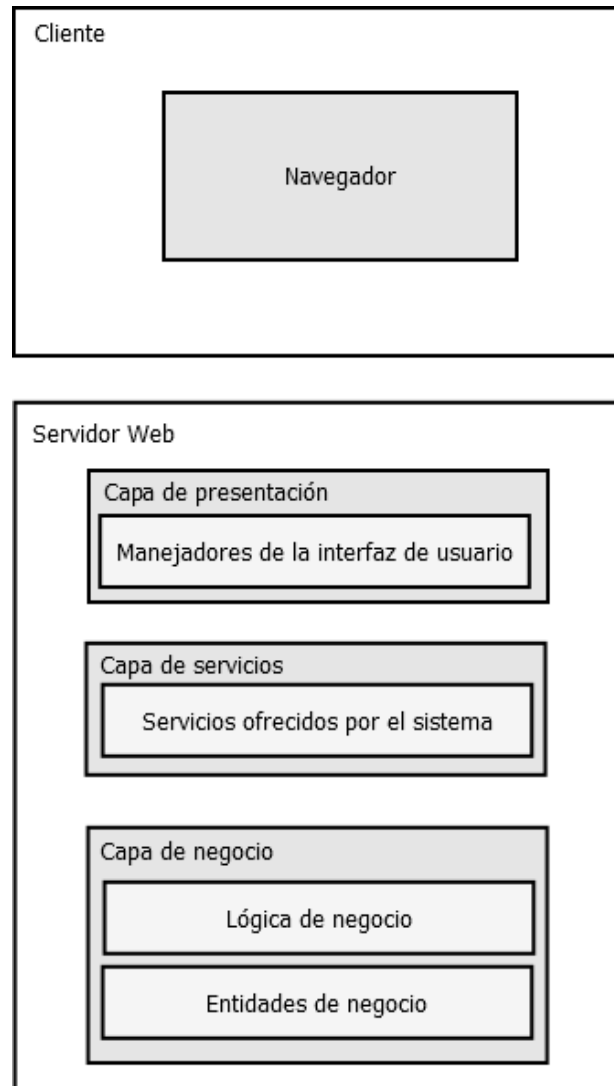


Figura 3.7. Arquitectura del sistema Web.

3. Introducción

3.5 Sistema Web

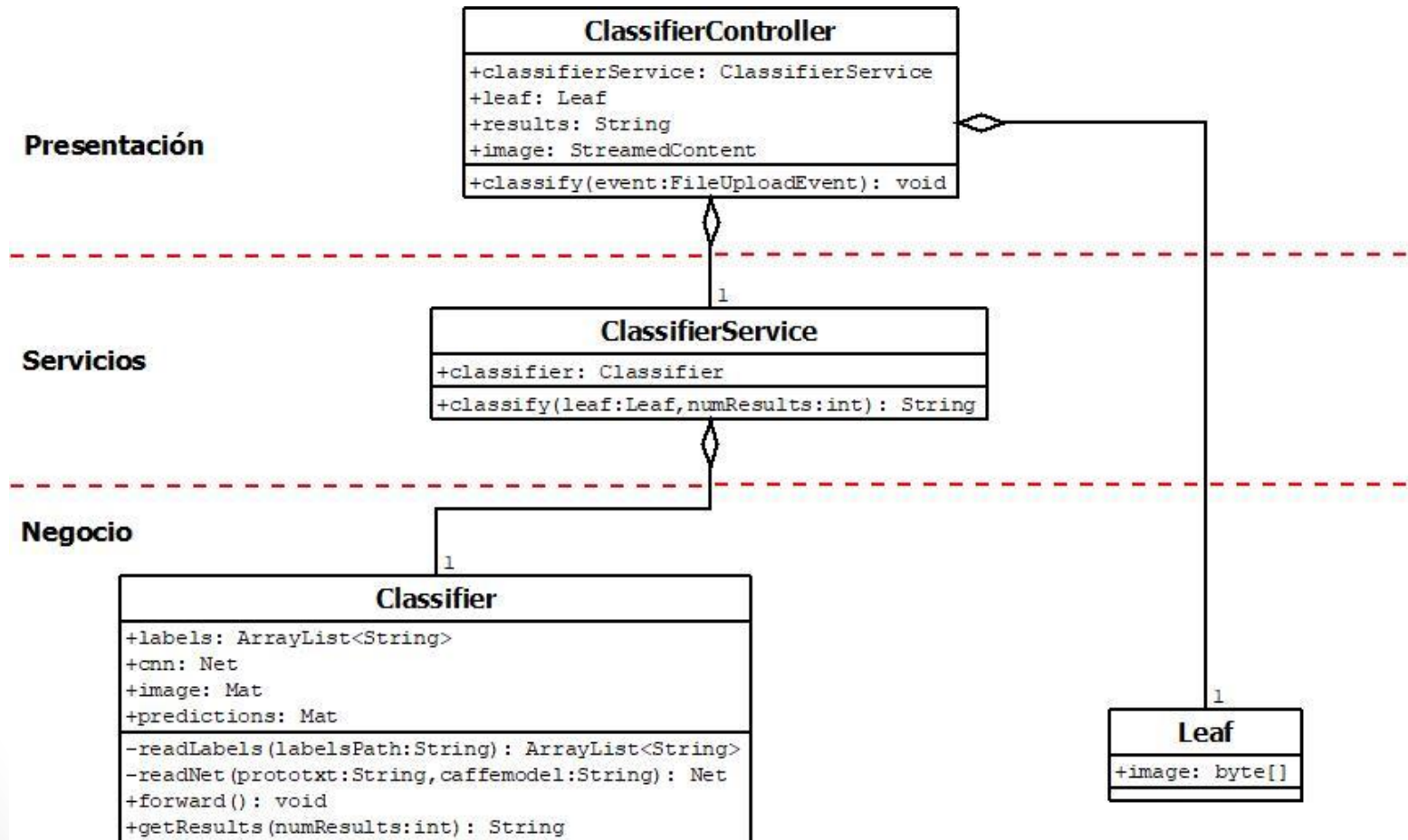


Figura 3.8. Diagrama de clases de la aplicación Web.

4. Clasificación de enfermedades.

4.1 Conjunto de datos

El conjunto de datos consiste de 16,419 imágenes divididas en las diez clases la tabla siguiente. Del conjunto total, 80% fueron usadas para la fase de entrenamiento y el restante para pruebas.

Clase	Número de imágenes
Virus del rizado amarillo del tomate (<i>Tomato yellow leaf curl virus</i>)	4032
Virus del mosaico del tomate (<i>Tomato mosaic virus</i>)	325
Corynespora cassiicola. Mancha en forma de blanco (<i>Target spot</i>)	1,356
Araña roja (<i>Spider mites</i>)	1,628
Septoriosis (<i>Septoria spot</i>)	1,723
Passalora fulva. Moho en la hoja (<i>Leaf mold</i>)	904
Tizón tardío (<i>Lateblight</i>)	1,781
Tizón temprano (<i>Earlyblight</i>)	952
Mancha bacteriana (<i>Bacterial spot</i>)	2,127
Hoja sana	1,591
Total	16,419

Figura 4.1. Conjunto de datos.

4. Clasificación de enfermedades.

4.2 Entrenamiento del clasificador

- El entrenamiento del clasificador se realizó con el framework de aprendizaje profundo Caffe.
- La máquina utilizada tiene las siguientes características: 4 GB de RAM, procesador Intel i3, sin GPU compatible con Caffe.
- Se seleccionó la arquitectura AlexNet. Esta arquitectura tuvo un tiempo estimado de entrenamiento de 7 días. En cambio la GoogLeNet tuvo un tiempo estimado de entrenamiento de 91 días!
- La AlexNet alcanzó una eficiencia del 98.82% después de 8 mil iteraciones de entrenamiento, en un periodo de una semana.

4. Clasificación de enfermedades.

4.2 Entrenamiento del clasificador

	0	1	2	3	4	5	6	7	8	9
0	336	0	0	0	0	0	0	0	0	0
1	1	350	0	0	0	0	0	0	0	4
2	0	0	425	1	0	0	0	0	0	0
3	1	0	0	107 1	0	0	0	0	0	0
4	1	0	0	0	314	0	1	0	3	0
5	0	0	0	0	0	75	0	0	0	0
6	18	1	0	0	0	1	257	3	0	1
7	1	0	0	0	0	0	1	196	2	0
8	0	0	0	0	0	0	0	0	379	3
9	0	0	0	0	0	0	0	0	0	191

0 Araña roja

1 Septoriosis

2 Mancha bacteriana

3 Virus del rizado amarillo

4 Planta sana

5 Virus del mosaico

6 Mancha en forma de blanco

7 Tizón temprano

8 Tizón tardío

9 Passalora fulva (Moho en la hoja)

4. Clasificación de enfermedades.

4.3 Identificación de enfermedades

Es posible realizar identificaciones de enfermedades usando el clasificador entrenado de la siguiente forma:



5. Sistema Web.

5.1 Implementación

Las tecnologías de desarrollo de la aplicación fueron las siguientes:

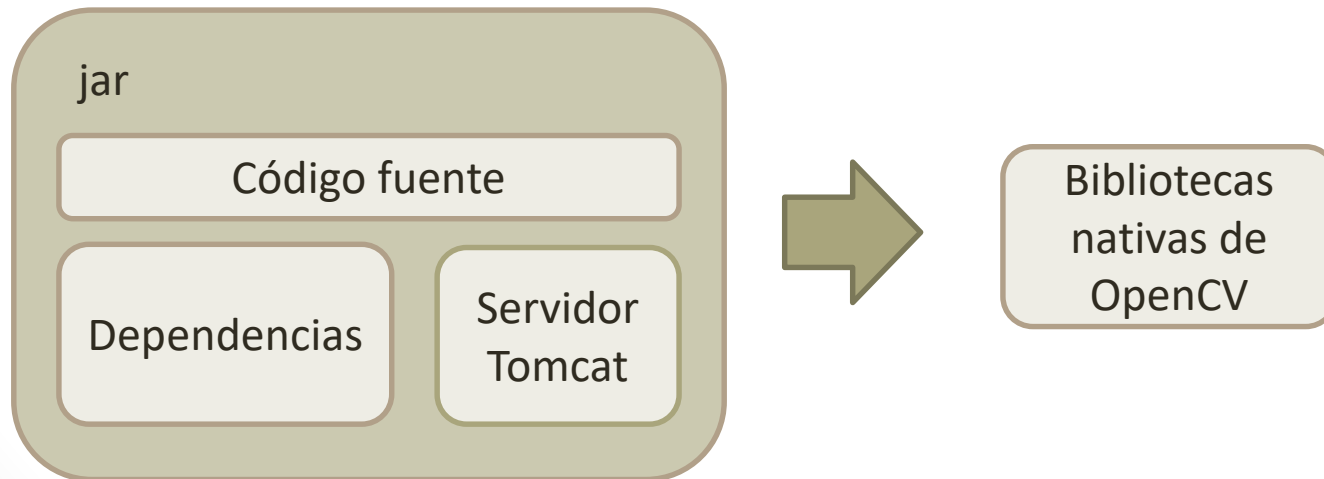


5. Sistema Web.

5.2 Implantación

El software del servidor, incluido un servidor web tomcat y las dependencias necesarias se comprimieron en un archivo jar, con excepción de las bibliotecas nativas de OpenCV (*.so, *.dll) para Java.

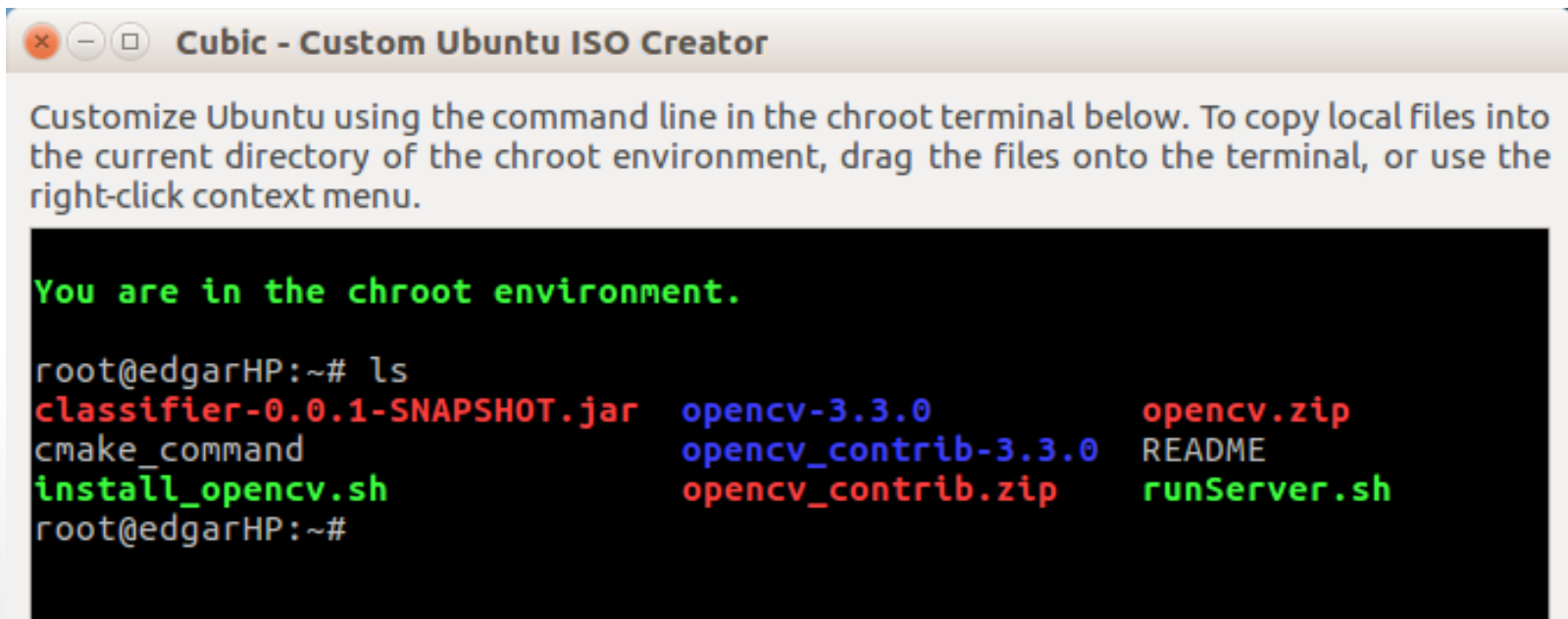
La ubicación de dichas bibliotecas deben indicarse con la bandera **Djava.library.path** del comando java al ejecutar el jar.



5. Sistema Web.

5.2 Implantación

También se modifico una imagen iso de Uubuntu 16.04 en la que se incluye el jar y scripts para la instalación de OpenCV y la ejecución del jar.



The screenshot shows a window titled "Cubic - Custom Ubuntu ISO Creator". Inside, there is a terminal window with the following text:

```
Customize Ubuntu using the command line in the chroot terminal below. To copy local files into the current directory of the chroot environment, drag the files onto the terminal, or use the right-click context menu.
```

You are in the chroot environment.

```
root@edgarHP:~# ls
classifier-0.0.1-SNAPSHOT.jar  opencv-3.3.0      opencv.zip
cmake_command                opencv_contrib-3.3.0  README
install_opencv.sh            opencv_contrib.zip  runServer.sh
root@edgarHP:~#
```

6. Pruebas del sistema.

6.1 Pruebas

Número de prueba	Requisito	Descripción de la prueba	Resultado extensión esperado
1	RF01	Seleccionar una imagen del sistema de archivos local del dispositivo.	Es posible seleccionar imágenes con *.jpg, *.png y *.bmp.
2	RF01	Seleccionar un archivo con extensión distinta a las permitidas.	No es posible seleccionar archivos con extensión distinta a las permitidas.
3	RF02	Seleccionar una imagen cuya clase sea previamente conocida (etiquetada) y clasificarla.	El sistema clasifica la imagen en alguna de las nueve enfermedades o en hoja sana.
4	RF02	Seleccionar una imagen que no corresponda con ninguna de las diez clases válidas.	El sistema debe indicar que la imagen no corresponde a ninguna de las enfermedades identificables u a una hoja sana.
5	RNF01	Medir la eficiencia de clasificación con imágenes del conjunto de prueba.	La eficiencia de clasificación es superior al 90%.
6	RF02	Medir el tiempo de respuesta del sistema Web.	El tiempo promedio de clasificación es menor a cinco segundos, para al menos 10% de imágenes del conjunto de prueba.

6. Pruebas del sistema.

6.2 Resultados

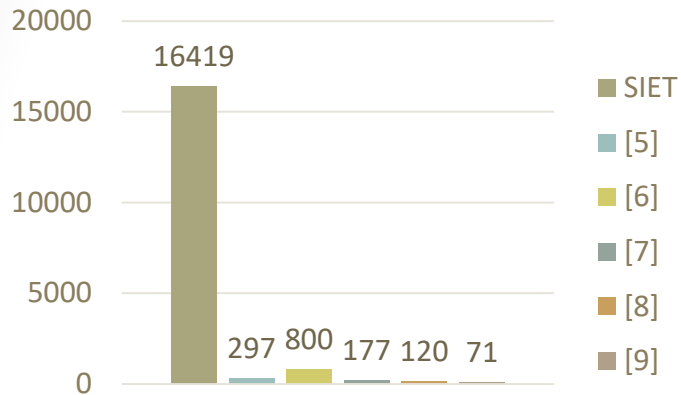
Número de prueba	Resultado	Observaciones
1	Se permitió la selección de archivos extensiones *.jpg, *.png y *.bmp.	
2	No se permitió la selección de archivos con extensión distinta a las permitidas.	
3	El sistema clasificó la imagen de forma correcta y mostró los dos resultados más probables.	
4	El sistema clasifica algunas de las imágenes en clases válidas.	Se debería agregar al clasificador una clase específica para aquellas imágenes no reconocidas.
5	La prueba de eficiencia (98.82%) fue presentada al final de la sección 5.3.	
6	El tiempo promedio de clasificación con imágenes del subconjunto de pruebas fue de 582 ms.	Solo el proceso de clasificación tarda en promedio 115 ms. El resto consiste en la transmisión de datos entre cliente y servidor.

7. Conclusiones.

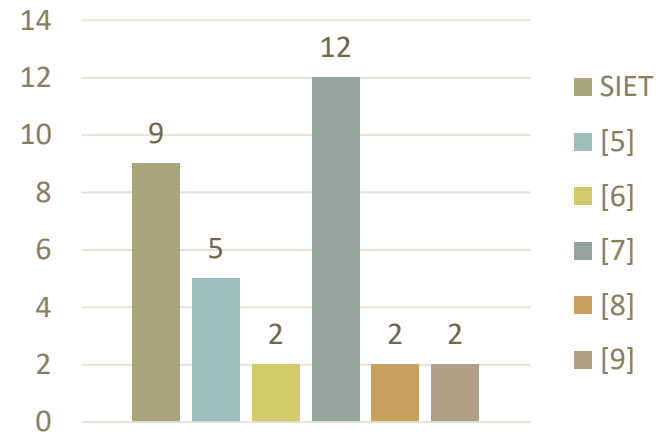
- Se desarrolló un sistema de reconocimiento de imágenes a partir de una red neuronal convolucional.
- Se entrenó la red neuronal sobre un conjunto de datos formado por 16,419 imágenes.
- Se logró identificar nueve enfermedades del tomate con una eficiencia de clasificación del 98.82%.
- Se implantó la red neuronal en un ambiente web a través del cual usuarios pueden realizar identificaciones de enfermedades del tomate.

7. Conclusiones.

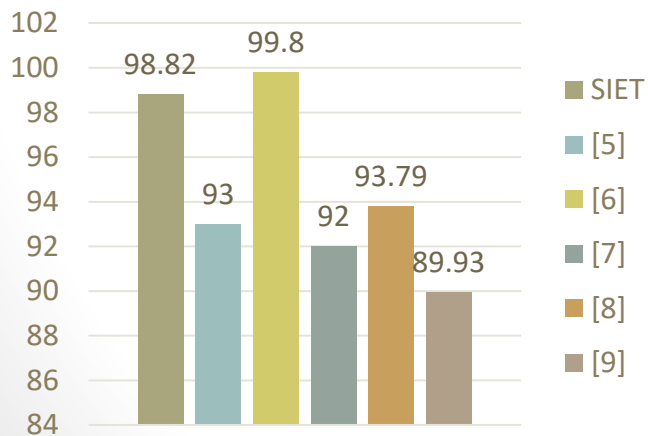
Total de imágenes



Enfermedades



Eficiencia



Estudio	Descriptor	Clasificador
Prasad, 2016 [5]	GWT, GLCM	KNN
Mokhtar et al. 2015 [6]	GLCM	SVM
Semary et al. 2015 [7]	Momentos de color, GLCM, descomposición wavelet	SVM
Dandawate 2015 [8]	SIFT	SVM
Raza et al. 2015 [9]	Estadísticas locales, globales	SVM

8. Trabajo a futuro.

- Probar y comparar arquitecturas distintas de CNN en busca de una mayor eficiencia de clasificación.
- Ampliar el número de plantas y enfermedades identificables.
- Usar las imágenes que los usuarios proporcionan para incrementar el tamaño del conjunto de datos, con el objetivo de mejorar la eficiencia de clasificación.
- Desarrollo de una aplicación móvil.

9. Referencias

- [1] SAGARPA. Hidroponía rústica. [En línea] Disponible en: <http://www.sagarpa.gob.mx>
- [2] Research and Markets. Global Hydroponics Market - Forecasts from 2017 to 2022. [En línea] Disponible en: <https://www.researchandmarkets.com>.
- [3] Sánchez F. Entrevista con Felipe Sanchez del Castillo, investigador de la Universidad Autónoma Chapingo. Recuperado de <http://www.2000agro.com.mx>.
- [4] Brahimi M. et al. Deep Learning for Tomato Disease: Classification and Symtoms Visualization. 2017.
- [5] Prasad S. et al. Multi-resolution mobile vision system for plant leaf disease diagnosis. 2015.
- [6] Mokhtar U. et al. SVM-Based detection of tomato leaves diseases. 2015.
- [7] Semary N. et al. Fruit-based tomato grading system using features fusión and support vector machine. 2015.
- [8] Dandawate Y., Kokare R. An automated approach for classification of plant diseases towards development of futuristic decision support system in Indian perspective 2015.
- [9] Raza S. et al. Automatic detection of diseased tomato plants using thermal and stereo visible light images. 2015

Gracias