

**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

---



*Una universidad incluyente y comprometida con el desarrollo integral*

---

**DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS.**

Autor

**Omar Esneider Alvarez Acosta**

Director

**Marleny Fernández Sandoval**

**INGENIERÍA ELECTRÓNICA  
DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA SISTEMAS Y  
TELECOMUNICACIONES  
FACULTAD DE INGENIERÍAS Y ARQUITECTURA  
UNIVERSIDAD DE PAMPLONA  
PAMPLONA  
JULIO 18 DE 2015  
UNIVERSIDAD DE PAMPLONA**



Autor: Omar Esneider Alvarez Acosta  
Director: Marleny Fernández Sandoval

**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

---

**FACULTAD DE INGENIERÍAS Y ARQUITECTURA  
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA,  
SISTEMAS Y TELECOMUNICACIONES  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
TRABAJO PRESENTADO PARA OPTAR POR EL TITULO DE  
INGENIERO ELECTRÓNICO**

**TEMA:**

**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

**FECHA DE INICIO DEL TRABAJO: 1 de Septiembre 2020  
FECHA DE TERMINACIÓN DEL TRABAJO: 18 de Diciembre 2020**

**NOMBRES Y FIRMAS DE AUTORIZACIÓN PARA LA SUSTENTACIÓN:**

\_\_\_\_\_  
Omar Esneider Alvarez Acosta  
**AUTOR**

\_\_\_\_\_  
MARLENY FERNÁNDEZ  
SANDOVAL  
**DIRECTORA**

\_\_\_\_\_  
JOSÉ DANIEL RAMÍREZ CORZO  
**DIRECTORA DE PROGRAMA**

**JURADO CALIFICADOR:**

\_\_\_\_\_  
ABELARDO MEJÍA BUGALLO

\_\_\_\_\_  
MILLER SÁNCHEZ BALAGUERA

\_\_\_\_\_  
MARLENY FERNÁNDEZ SANDOVAL

**PAMPLONA NORTE DE SANTANDER  
COLOMBIA  
18 de Diciembre del 2020**



**Autor:** Omar Esneider Alvarez Acosta  
**Director:** Marleny Fernández Sandoval

**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE  
HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES  
EN AMBIENTES CONTROLADOS**

---

*Para ustedes;*

*Omar Alvarez Pedraza*

*Gloria Aydet Acosta García*

*Laura García Arias*

*Sara Salome Martínez García*

*Ludy Arias Villamizar*

*Nelly Arias Villamizar*

*Eterna Gratitud*



**Autor:** Omar Esneider Alvarez Acosta  
**Director:** Marleny Fernández Sandoval

## **DEDICATORIA**

La presente tesis está dedicado a Dios, ya que gracias a el he logrado concluir mi carrera, a mis padres Omar Alvarez y Aydet Acosta pues sin ellos no lo había logrado, por todo el sacrificio que tuvieron que pasar para ayudarme a cumplir este sueño, gracias por haberme forjado como la persona que soy en la actualidad; muchos de mis logros se los debo a ustedes entre los que se incluye este, no ha sido sencillo el camino hasta ahora, pero gracias a sus aportes, a su amor, a sus valiosos consejos y apoyo, lo complicado de lograr esta meta se ha notado menos.

A mi esposa y mi hija que han sido las personas que me han sido mi apoyo incondicional, gracia a ellas he tenido las fuerza en los momentos mas difíciles para sobresalir y nunca rendirme, gracias por ayudarme a construir nuestro hermoso destino.

Gracias a mis hermanos Haware, Fabian y Cristian Alvarez que han sido mi mayor orgullo, gracias por el apoyo incondicional y por creer en mí, prometo nunca defraudarlos, este triunfo es de ustedes.

A todos los profesores que con sus conocimientos y experiencias me ayudaron en ser un gran profesional, por todos los consejos y la paciencia brindada.

A todas esas personas que de una u otra manera han aportado su granito en la realización de este proyecto, a la ingeniera Helena Castro por su apoyo y aporte en la parte investigativa y a todos mis familiares que creyeron en mí, MIL GRACIAS



# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

---

## RESUMEN

Este proyecto está enfocado en implementar un sistema de visión artificial automático, para detectar tres clases de hongos como son el oídio, mildiu y la roya ya que son los más frecuentes en esta clase de cultivos, mediante una cámara instalada dentro de un invernadero, previamente construido, capturará imágenes en tiempo real de los cambios que se presentan en las hojas, estos pueden ser tipo: cambio de color, cambio de forma o imperfecciones dependiendo la clase del hongo. El sistema se desarrollará mediante 4 fases principales las cuales son: adecuación del sitio de desarrollo, adquisición de datos, procesamiento de imágenes y reconocimiento de patrones. En la adecuación del sitio de desarrollo se llevará a cabo la construcción de un ambiente controlado que cumplirá con los estándares básicos que permitan garantizar el ambiente para el cultivo, ya que es una planta que requiere cuidado, en este ambiente se podrán visualizar la humedad y la temperatura parámetros claves para su crecimiento y desarrollo. En la fase de adquisición de datos se recolectará información fotográfica de las distintas clases de hongos que puede padecer la planta, entre ellos: oídios, mildiu y roya entre otros, esta base de datos se obtendrá partir de consultas bibliográficas y de páginas web la cual pretende contribuir al posterior entrenamiento de las diferentes redes neuronales a utilizar. Una vez determinada la base de datos con las imágenes de los hongos se procede a realizar la fase 3, que consiste en el procesamiento de estas, donde se extraerán los patrones que correspondan a características específicas relacionadas con cada enfermedad y plaga todo esto se realizara por medio de un ordenador de placa reducida conocido como raspberry pi. Por último, se pondrá en marcha la fase 4 la cual, básicamente consiste en la utilización de una cámara (raspicam) instalada en invernadero que realizará un recorrido por el cultivo, el objetivo es reconocer patrones que identifique la presencia de algún imperfecto presente en el cultivo, al detectar alguna anomalía enviara un mensaje de texto automáticamente al estudiante mediante un módulo sim808 reportando así la presencia del hongo y enviara su posición actual.

Palabras Claves: visión artificial, hongos, cannabis medicinal, redes neuronales, Python.



**Autor:** Omar Esneider Alvarez Acosta  
**Director:** Marleny Fernández Sandoval

# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

---

## ABSTRACT

This project is focused on implementing an automatic artificial vision system to detect three types of fungi such as powdery mildew, mildew and rust as they are the most common in this class of crops, using a camera installed inside a greenhouse, previously built, capture images in real time of the changes that occur in the leaves, these can be type: change of color, change of shape or imperfections depending on the class of fungus. The system will be developed through 4 main phases which are: adaptation of the development site, data acquisition, image processing and pattern recognition. In the adequacy of the development site will be carried out the construction of a controlled environment that will meet the basic standards to ensure the environment for cultivation, as it is a plant that requires care, in this environment you can visualize the humidity and temperature key parameters for growth and development. In the data acquisition phase, photographic information will be collected on the different types of fungi that the plant may suffer, among them: oidium, botrytis, rust, Pythium, Fusarium, among others. This database will be obtained from bibliographic consultations and web pages, which will contribute to the subsequent training of the different neuronal networks to be used. Once the database with the images of the fungi is determined, we proceed to phase 3, which consists in the processing of these, where the patterns that correspond to specific characteristics related to each disease and pest will be extracted. All this will be done by means of a computer with a reduced plate known as raspberry pi.

Finally, phase 4 will be started, which basically consists on the use of a camera (raspicam) installed in the mini-greenhouse that will make a tour through the crop. The objective is to recognize patterns that identify the presence of some imperfect present in the crop, when detecting some anomaly it will send a text message automatically to the student through a sim808 module reporting the presence of the fungus and sending its current position.

Keywords: artificial visión, mushrooms, medicinal cannabis, neuronal networks, Python



**Autor:** Omar Esneider Alvarez Acosta  
**Director:** Marleny Fernández Sandoval

## Contenido

RESUMEN .....	6
ABSTRACT.....	7
INTRODUCCIÓN .....	12
JUSTIFICACIÓN .....	13
DELIMITACIONES .....	14
OBJETIVO GENERAL .....	14
OBJETIVOS ESPECÍFICOS .....	14
<b>CAPÍTULO 1 .....</b>	<b>1</b>
1.REFERENTES TEÓRICOS.....	2
1.1    LA PLANTA DE CANNABIS.....	2
1.1.1    ¿QUÉ SON LOS HONGOS? .....	3
1.1.2    MILDIU .....	4
1.1.3    OÍDIO.....	5
1.1.4    ROYA .....	6
1.2    INTELIGENCIA ARTIFICIAL.....	8
1.2.1    CLASIFICACIÓN DE INTELIGENCIA ARTIFICIAL.....	8
1.2.2    PRINCIPALES RAMAS DE LA INTELIGENCIA ARTIFICIAL.....	9
1.3    PROCESAMIENTO DIGITAL DE IMÁGENES .....	14
1.3.1    ESPACIO DE COLOR .....	14
1.3.2    PIXEL.....	15
1.3.3    ETAPAS EN EL PROCESAMIENTO DE IMAGEN .....	16
1.4    REDES NEURONALES .....	22
1.4.1    REDES NEURONALES CONVOLUCIONALES (CNN).....	23
1.4.2    REDES NEURONALES CONVOLUCIONALES BASADAS EN REGIÓN (R-CNN) 26	
1.5    YOLO V3.....	26
1.5.1    YOLO V3-TINY.....	27
1.5.2    PREDICCIÓN DE BOUNDING BOX.....	28
1.5.3    PREDICCIÓN DE LAS CLASES.....	29



**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

---

1.5.4	PREDICCIÓN A TRAVÉS DE ESCALAS .....	30
1.5.5	ARQUITECTURA DE LA RED .....	31
1.6	RASPBERRY PI 4 .....	31
1.6.1	¿QUÉ ES? .....	32
1.6.2	VENTAJAS Y DESVENTAJAS .....	32
1.6.3	CARACTERÍSTICAS RPI4 .....	33
<b>1.</b>	<b>2.METODOLOGÍA .....</b>	<b>35</b>
2.1.	EXPLORACIÓN PREVIA:.....	36
2.2.	ADECUACIÓN DEL SITIO DE DESARROLLO: .....	37
2.3.	ADQUISICIÓN DE DATOS .....	40
2.4.	PRE-PROCESAMIENTO DE IMÁGENES .....	40
2.5.	RECONOCIMIENTO DE PATRONES .....	45
2.6.	ENTRENAMIENTO YOLOV3 PARA DETECCIÓN DE HONGOS .....	48
2.6.1	CREACIÓN DE LA DATA PARA ENTRENAMIENTO Y VALIDACIÓN .....	48
2.6.2	ETIQUETADO DE IMÁGENES .....	49
2.6.3	CONVERSIÓN DE XML A YOLO V3.....	50
2.6.4	ENTRENAMIENTO MODELO DETECCIÓN DE HONGOS.....	51
2.7.	MODELOS DE PRUEBA.....	67
2.7.1	Modelo Propio .....	67
2.7.2	Modelo ResNet .....	68
2.7.3	Modelo VGG19 .....	70
2.8.	SISTEMA DE VISION ARTIFICIAL.....	72
2.9.	IMPLEMENTACIÓN .....	72
3.	VALIDACIÓN Y RESULTADOS .....	76
4.	CONCLUSIONES.....	87
5.	REFERENCIAS.....	88





## ÍNDICE DE FIGURAS

Ilustración 1. Hoja de Cannabis [1] .....	3
Ilustración 2 Hojas de Cannabis infectadas con hongos .....	4
Ilustración 3. Hoja de cannabis con Mildiu[4].....	5
Ilustración 4. Hoja de cannabis con oídio .....	6
Ilustración 5. Hojas de cannabis con Roya[6].....	7
Ilustración 6. Algoritmo Random Forest [12] .....	12
Ilustración 7. K-Vecinos [10].....	13
Ilustración 8. Imagen en espacio RGB-BGR-HSV [16].....	15
Ilustración 9. Imagen Binarizada [16].....	17
Ilustración 10. Rango de grises de una imagen [16] .....	18
Ilustración 11. Imagen binarizada. [21].....	18
Ilustración 12. Imagen difuminada [16] .....	20
Ilustración 13.Histograma RGB imagen A [16].....	21
Ilustración 14. Histograma RGB imagen B [16] .....	21
Ilustración 15. Estructura red neuronal multicapa[23].....	23
Ilustración 16. Capas de una neurona convolucional [24] .....	24
Ilustración 17. Aplicación de capa de un filtro convolucional [4] .....	25
Ilustración 18. Matriz maxpooling [25] .....	25
Ilustración 19. Arquitectura R-CNN [24] .....	26
Ilustración 20.Detección YoloV3 y YoloV3-tiny. [26].....	28
Ilustración 21: Bounding boxes con dimensiones priors y localización de predicción .....	29
Ilustración 22 Arquitectura YoloV3 [26].....	31
Ilustración 23. Placa Raspberry pi4 [27].....	32
Ilustración 24. Diagrama de conexión de RPI4 [27] .....	34
Ilustración 25. Germinación de semillas de cannabis [Autor].....	37
Ilustración 26. Plántulas de cannabis en crecimiento .....	38
Ilustración 27. Construcción del invernadero en Pamplona [Autor] .....	39
Ilustración 28. lampara Fluorecente .....	39
Ilustración 29. Hoja infectada con Mildiu .....	40
Ilustración 30. Logo de base de datos Kaggle .....	40
Ilustración 31. Importación de librerías OpenCv [Autor].....	41
Ilustración 32.Redimensionamiento imagen de entrada.....	41
Ilustración 33. (A) Imagen de entrada (B) Histograma RGB.....	41
Ilustración 34. Conversión BGR-GRAY imagen [Autor] .....	42
Ilustración 35. (A) Imagen escala de gris (B) Histograma color gris.....	42
Ilustración 36. Código de binarización [Autor].....	42
Ilustración 37. Imagen binarizada .....	43
Ilustración 38. Función de detección de contornos [Autor] .....	43
Ilustración 39. Dibujar Contorno.....	43



**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE  
HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES  
EN AMBIENTES CONTROLADOS**

---

Ilustración 40. Imagen con contorno dibujado .....	44
Ilustración 41. Función para recortar contorno .....	44
Ilustración 42. Guardar imágenes en Carpetas .....	45
Ilustración 43. imagen recortada .....	45
Ilustración 44. Rango do colores HSV .....	46
Ilustración 45.(A) imagen binarizada (B) Imagen con contorno (C) Imagen recortada[propio] [Autor] .....	46
Ilustración 46. Rango de colores HSV [Autor] .....	47
Ilustración 47. Conversión BGR-HSV [Autor].....	47
Ilustración 48. Mascara de Mildiu [Autor] .....	47
Ilustración 49. Mascara binarizada Mildiu [Autor].....	47
Ilustración 50.(A)imagen color (B)mascara de mildiu [Autor] .....	48
Ilustración 51. Entorno de etiquetado Labelimg.....	49
Ilustración 52. Documento con extensión XML .....	50
Ilustración 53. Código conversor XML-TXT .....	50
Ilustración 54 Arquitectura Darknet53 .....	51
Ilustración 55. Arquitectura Yolo V3 .....	53
Ilustración 56. Estructura YoloV3.....	54
Ilustración 57. función YoloV3 .....	55
Ilustración 58. Función Modulo Residual.....	55
Ilustración 59. Modulo residual .....	56
Ilustración 60. Extracción de características .....	56
Ilustración 61. Vector de características .....	57
Ilustración 62. Región de interés.....	57
Ilustración 63.Área de ROI mapeada en la red CNN .....	58
Ilustración 64. Imágenes con mallas de identificación .....	59
Ilustración 65.Dimensionamiento de caja de predicción.....	59
Ilustración 66. Función limitadora de objetos .....	60
Ilustración 67.Procesamiento NMS.....	61
Ilustración 68. Código función NMS.....	61
Ilustración 69. Antecedente GIOU .....	64
Ilustración 70. Código función GIOU .....	65
Ilustración 71. Tasa de Aprendizaje .....	66
Ilustración 72. Modificación ResNet50 .....	69
Ilustración 73. Estructura ResNet-50 .....	69
Ilustración 74. Capas Modificadas ResNet-50.....	69
Ilustración 75 Estructura modificada VGG19 .....	71
Ilustración 76. Grafica entrenamiento para 100 imagen [Autor] .....	79
Ilustración 77. Grafica validación 1000 imágenes [Autor] .....	80
Ilustración 78. Grafica modelo de Data de Entrenamiento .....	80
Ilustración 79. Matriz de Confusion YoloV3 .....	81
Ilustración 80. Perdidas de entrenamiento y validación del modelo propio [propio].....	82
Ilustración 81. Porcentaje Accuracy de entrenamiento y validación [propia].....	83
Ilustración 82. Matriz de decisión modelo propio [propio] .....	83
Ilustración 83. perdidas y Accuracy ResNet-50.....	84
Ilustración 84. Matriz de confusión ResNet-50.....	84



## INTRODUCCIÓN

En Colombia cuenta con una reglamentación que permite la fabricación, uso de semillas y cultivo de cannabis para fines medicinales y científicos, así como para la fabricación de derivados de cannabis, los que mayormente son exportados. De la normativa se desprende no sólo la regulación sobre el cannabis y la emisión de licencias, también la responsabilidad social con pequeños y medianos productores.[1]

Las aplicaciones de visión artificial y el análisis de imágenes pueden resolver varios problemas en los campos industriales, científicos o de seguridad. Para realizar la tarea de clasificación de imágenes, es necesario extraer las características de cada tipo de imagen, como color, forma y textura. En el trabajo actual, es necesario implementar un algoritmo para la construcción de un detector de imágenes de hongos en plantas de cannabis medicinal basado en la extracción de características de color de imágenes en determinadas regiones de interés.

Para desarrollar el entrenamiento de redes neuronales y la clasificación de imágenes de hongos, se utiliza la extracción de patrones y mediante la realización de algoritmos de inteligencia artificial las imágenes se clasifican automáticamente. El conjunto de datos utilizado incluye: tres tipos de hongos (Mildiu, Oídio y Roya), cada tipo de hongo tiene un número variable de imágenes, y luego se usa la tecnología de máscara para preparar la imagen seleccionando la región de interés y dividiéndolas en dos conjuntos de datos: datos de entrenamiento y datos de prueba. Después de entrenar el sistema de red neuronal, se realizará una prueba de clasificación para evaluar la efectividad. [2]



## JUSTIFICACIÓN

Cultivar plantas de Cannabis medicinal no es fácil. Como todos los seres vivos, nuestras plantas pueden tener problemas: hongos, plagas y virus pueden atacar nuestro cultivo y extenderse en pocos días destruyendo toda la plantación si no se ponen medidas a tiempo. La prevención y el control diario del cultivo son fundamentales para evitar el inicio de un problema que nos puede costar mucho esfuerzo erradicar si cultivamos nuestro Cannabis medicinal en exterior, tendrá que convivir rodeado de esporas, insectos de todo tipo y bacterias. Con unas buenas labores de cultivo y una buena nutrición, las plantas cultivadas al exterior crecerán sanas y robustas. En el cultivo en interior, podemos evitar que cualquier insecto u hongo entre en nuestro cuarto de cultivo con medidas preventivas, los hongos más frecuentes son el oídio, mildiu y la roya. La prevención es muy importante: tanto si te decantas por los insecticidas ecológicos como por los sistémicos, para tener éxito en el cultivo del Cannabis es conveniente “blindar” las plantas. Nos referimos a tratar sistemáticamente con insecticidas durante la fase vegetativa para evitar pulverizar durante la fase de floración. En caso de infestación durante esta última fase, recomendamos luchar contra los hongos solo con medios orgánicos, para evitar contaminar las flores con algún agente químico potencialmente tóxico para la salud.[3]



## DELIMITACIONES

### OBJETIVO GENERAL

- ✚ Desarrollar un sistema de visión artificial para la detección de hongos en plantas de cannabis

### OBJETIVOS ESPECÍFICOS

- Realizar un análisis documental sobre el tema de investigación con base de datos bibliográficos que permita la identificación de los diferentes hongos presentes en los cultivos del cannabis.
- Construir un invernadero para la siembra y cultivo de cannabis medicinal.
- Desarrollar el sistema de visión artificial para la toma de muestras en las plantas de cannabis medicinal.
- Implementar un sistema de inteligencia artificial basado en redes neuronales para la identificación de los hongos en las plantas de cannabis medicinal en una raspberry pi 4.
- Validar el sistema mediante la detección de hongos en el cultivo de cannabis medicinal con la ayuda de un experto en el área quien desde su punto de vista dará un visto bueno a la metodología implementada teniendo en cuenta características visuales de la afectación de los mismos sobre La planta dentro de un ambiente controlado.



# CAPÍTULO 1



*Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval*

## 1. REFERENTES TEÓRICOS

### 1.1 LA PLANTA DE CANNABIS

Las plantas de cannabis han vivido en este planeta durante miles, incluso millones de años, y lo han estado haciendo durante mucho tiempo antes y después de la intervención humana. Siempre que la temperatura no sea demasiado baja y haya suficiente luz solar y alimentos para que la planta florezca, el cáñamo puede crecer en casi cualquier lugar y en cualquier lugar. En Asia, puede viajar a varias partes de Mongolia y visitar las plantas de cannabis que crecen de forma natural en las laderas y vastas llanuras, que a veces cubren colinas enteras y se extienden por los valles allí. Las plantas de cannabis son plantas muy adaptables que pueden crecer tanto en interior como en exterior.[1]

La planta de cannabis se utilizó por miles de años en distintas culturas alrededor del mundo para distintos fines, entre ellos, la medicina. La planta posee distintos principios activos, denominados Cannabinoides. Estos Cannabinoides, en un número de más de 60, tienen distintas funciones, algunas descubiertas, y en su mayoría sin tener en claro aún cual es propósito. Estos Cannabinoides, por estar dentro de la planta se denominan Fitocannabinoides.

Los hongos son la enfermedad más temida por los cultivadores, ya que pueden perjudicar de forma irreversible toda la producción, teniendo que tirar tantos meses de trabajo a la basura, literalmente. No solo debemos ocuparnos de la nutrición de las plantas, sino que también debemos tener muy presente los factores como la temperatura, la humedad y la ventilación del lugar de cultivo. LOS HONGOS se encuentran inactivos en forma de esporas esperando ese ambiente perfecto para poder desarrollarse y enfermar a todas las plantas. Hay muchas especies de hongos, pero estas variedades son las que más afectan al cultivo de cannabis.[2]





*Ilustración 1. Hoja de Cannabis [1]*

### 1.1.1 ¿QUÉ SON LOS HONGOS?

Los hongos son eucariotas muy primitivos pertenecientes al reino de los hongos, se reproducen a través de esporas y suelen aparecer en lugares cálidos, oscuros y húmedos.

Las esporas del hongo permanecen en el medio ambiente hasta que se encuentran las condiciones ideales para su crecimiento y desarrollo. El problema es que las condiciones de crecimiento de los hongos son muy similares a las condiciones de crecimiento de las plantas, ambientes oscuros, cálidos y húmedos, por lo que no es de extrañar que este tipo de organismos se reproduzcan en los cultivos.[2]

Hay algunos hongos que se pueden reproducir muy rápidamente, en pocos días podemos contagiar todo el cultivo con estos desagradables huéspedes, por eso hay que tomar una serie de precauciones para evitarlos. Si las medidas preventivas no han funcionado y los hongos han invadido nuestros cultivos, detectarlos lo antes posible y actuar con rapidez para eliminarlos es fundamental para evitar mayores daños y salvar nuestra cosecha.[3]







*Ilustración 2 Hojas de Cannabis infectadas con hongos*

### 1.1.2 MILDIU

El Mildiu (Mildew) es una enfermedad fúngica que hace que las hifas crezcan en los tejidos de las plantas. Al igual que el mildiú polvoriento, solo permanece en la superficie de hojas, tallos y flores, penetra en los tejidos de las plantas y causa graves daños, generalmente en el aire de las plantas. Aunque existe y puede atacar muchos tipos de árboles y plantas, no es uno de los hongos que solemos ver en los cultivos de cáñamo, por lo que las enfermedades infecciosas pueden provenir de cualquier parte. Como la mayoría de los hongos, funciona mejor en climas húmedos, templados y con poca ventilación.[4]

El mildiú es un hongo extremadamente agresivo que prospera en lugares donde la temperatura es inferior a 26°C y la humedad supera el 50%. Aunque tiene un buen sentimiento por las hojas jóvenes, puede afectar a las plantas tanto durante el crecimiento como durante la floración. El moho aparece en forma de manchas amarillentas y tenues en las hojas, y no siempre es posible ver un polvo blanco similar al mildiú polvoriento en la parte inferior de las hojas. Cuando la condición empeora, hará que las hojas se deshojen y luego mueran.



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

El hongo que causa el mildiú polvoriento tiene dos formas de reproducción, una es la reproducción asexual y la otra es la reproducción sexual. La etapa sexual se caracteriza por la formación de pequeños cuerpos negros, esféricos y completamente encerrados denominados pequeñas pieles, que son raros en áreas tropicales, pero muy comunes en áreas con estaciones despejadas (primavera, verano, otoño e invierno). La retracción de la vaina es una estructura que permite que el hongo sobreviva en condiciones adversas y también cambia su composición genética. [3]

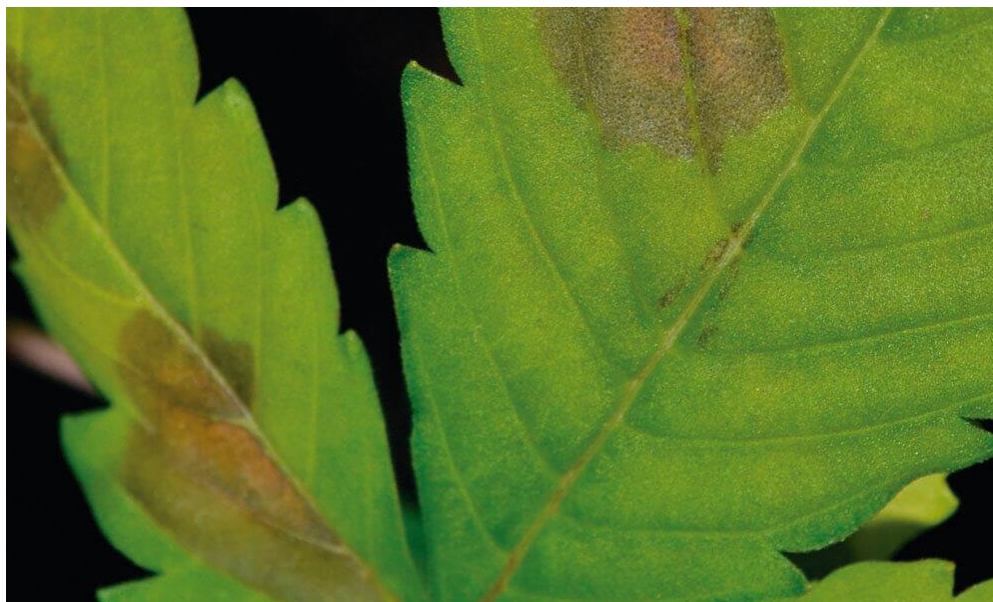


Ilustración 3. Hoja de cannabis con Mildiu[4]

### 1.1.3 OÍDIO

El mildiú polvoroso es un hongo que afecta a muchas especies. Quizás el más famoso es el mildiú polvoroso de la vid (*Vitis vinifera* L.) causado por la especie *Uncinula necator*, aunque el más común que se encuentra en los jardines es el mildiú polvoriento causado por *Podosphaera pannosa*. [5]

A veces, durante el cultivo de cannabis, especialmente en cultivos de exterior y cultivos de invernadero, y menos común en cultivos de interior, un hongo terrible llamado Oídio polvoriento aparecerá en su planta de cannabis durante la noche. El Oídio puede desarrollarse rápidamente y destruir por completo su cosecha, por lo que le recomendamos que siempre evite el mildiú polvoroso, especialmente en cultivos de exterior e invernadero. [3]

Es fácil de detectar porque al principio suele aparecer como un polvo blanco en las hojas y poco a poco se va extendiendo hasta llegar a los tallos y flores. Cuando el mildiú polvoriento se vuelve más común, puede causar amarillamiento y defoliación de las hojas. Es un hongo al que le gusta la humedad y las altas



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

temperaturas, se deposita en las hojas y transporta las hifas para absorber los nutrientes de las plantas. No es un hongo particularmente dañino, pero sí muy común, si no se puede controlar a tiempo y llega a los cogollos, se debe desechar porque, como las esporas infectadas con Botrytis, no se pueden comer de ninguna manera por sus efectos nocivos. Por nuestra salud

Salvo que los cultivos se cultiven en un ambiente cerrado donde se pueda controlar la temperatura, la humedad y la ventilación, nos será difícil evitar que aparezca si no utilizamos fungicidas. Especialmente cuando aparecieron hongos en años anteriores. Una de las pocas medidas que podemos tomar contra el mildiú polvoriento es asegurar una buena ventilación, evitar sombras constantes, regular el riego y fertilizar adecuadamente. [5]



Ilustración 4. Hoja de cannabis con oídio

### 1.1.4 ROYA

La roya, causada por El patógeno *Hemileia vastratrix* (Berk et Br.) Es un Hongos pertenecientes a la familia Umbelliferae, orden Uredinales, basidiomicetos. Todas las setas Los que pertenecen a la orden son parásitos, Solo se desarrolla en el tejido vivo de la planta. El anfitrión, en este caso, es la hoja del cafeto.[6]

Inicialmente, los síntomas son Aparecen pequeñas lesiones o manchas redondas, Amarillo claro, 1-3 mm de diámetro, Si busca una mancha, la mancha es translúcida Ligeras y similares a las manchas de aceite. A medida que este punto aumenta gradualmente Esporulación (formación de esporas La superficie inferior de la hoja), puede alcanzar 2 cm El diámetro, tomado como naranja, la superficie es Conviértete en polvo. En esta etapa de desarrollo, si El dedo toca la lesión y las esporas se adhieren a ella. Cuando estos son abundantes, se caen fácilmente. 6 Se forma una nube de color naranja claro. si existiera Muchas



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

lesiones o manchas, crecen juntas Cubra toda la hoja de trabajo entre sí y haga que disminución.[7]

También puedes observar alrededor de la mancha. Borde marrón, generalmente amarillo Si produce esporas de óxido El clima es propicio para la esporulación. Esta mancha se aprecia más Frecuencia al final de la epidemia, Después de un período de sequía en nuestro país Cosecha de café. Esta herida representa La principal fuente de infección en las primeras etapas de la infección.

Las esporas producidas por la roya del café son Llamadas uredospores, que produce Un gran número, es el medio más adecuado. Haz que la enfermedad sea perpetua. Estas esporas pueden Puede sobrevivir por poco tiempo en condiciones secas. Pero cuando Las hojas permanecen húmedas durante varias horas. Las esporas de óxido pueden sobrevivir más tiempo. Germinar bien y penetrar las hojas, haciendo infección.[6]

La germinación de esporas solo ocurre bajo las siguientes condiciones Agua libre, la humedad relativa alta no es suficiente Se produjo la germinación de esporas. por La investigación sobre este tema es bien conocida Fácil de germinar en comparación con las hojas jóvenes. Hojas con vida media o vieja. Pero cuando Empezó a llover y se localizó la enfermedad La hoja más interior y más antigua de Bandola es Fuente de esporas para nuevas infecciones.

El hongo de la roya produce otras estructuras Se llama Udaria, teglial y basidial pero este último es desconocido Obviamente funciona.[5]



Ilustración 5. Hojas de cannabis con Roya[6]



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

## 1.2 INTELIGENCIA ARTIFICIAL

La inteligencia artificial es un amplio campo de investigación dedicado a la creación de sistemas y máquinas con comportamiento inteligente. Ahora, los expertos en este campo de investigación parten de las siguientes dificultades: no existe una definición precisa del concepto de inteligencia humana, y la función del cerebro humano aún no se comprende del todo. La robótica y la tecnología informática son los dos componentes básicos de este campo técnico, que existe desde hace más de 50 años.

Inicialmente, la IA se centró en resolver teoremas, fórmulas y problemas geométricos, pero pronto se centró en los llamados problemas de razonamiento de sentido común. Los tres procesos básicos de la inteligencia humana que la inteligencia artificial intenta imitar están en orden ascendente:[8]

- ✓ **Aprendizaje:** incluida la adquisición de conocimientos e información, y sus reglas de aplicación. Comprender la inteligencia requiere conocimientos previos es uno de los primeros descubrimientos de la IA.
- ✓ **Razonamiento:** se trata de utilizar estas reglas y conocimientos para sacar conclusiones aproximadas o definitivas.
- ✓ **Autocorrección:** un genio con inteligencia artificial debe finalmente ser capaz de reconocer sus propios errores y descartar el conocimiento, las conclusiones y las acciones que lo llevaron.

### 1.2.1 CLASIFICACIÓN DE INTELIGENCIA ARTIFICIAL

La escuela clásica dentro de la IA, utiliza representaciones simbólicas basadas en un número finito de primitivas y de reglas para la manipulación de símbolos (por ejemplo, redes semánticas, lógica de predicados, etc.), los cuales fueron y siguen siendo parte central de dichos sistemas.[9]

La característica del método de IA sub-simbólico es crear un sistema con las siguientes funciones: Aprender. Esto se puede lograr a nivel personal imitando el cerebro (reelaborado). Neuro), imitando la evolución a nivel de especie. Hasta hace poco era común Habitualmente se habla de algoritmos genéticos (GA) en



# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

---

lugar de identificar diferentes tipos, Porque los algoritmos restantes pueden interpretarse como variantes o mejoras Acciones de la compañía.[9]

Hoy en día, la inteligencia artificial comienza a expandir sus campos de investigación en todas las direcciones, y trata de integrar diferentes métodos en sistemas a gran escala para encontrar soluciones. Aprovechando estas ventajas Dominio del conocimiento porque la aplicación está en Conocimientos de medicina, biología, ingeniería, educación, etc. [8]

## 1.2.2 PRINCIPALES RAMAS DE LA INTELIGENCIA ARTIFICIAL

### 1.2.2.1 APRENDIZAJE AUTOMATICO (MACHINE LEARNING)

El aprendizaje automático consiste en una asignatura de informática relacionada con el desarrollo de la inteligencia artificial. Como se mencionó anteriormente, se puede utilizar para crear sistemas que puedan aprender por sí mismos. Es una tecnología que puede automatizar una serie de operaciones para reducir la necesidad de intervención humana. Esto puede ser una gran ventaja a la hora de controlar grandes cantidades de información de forma más eficaz.[10]

El llamado aprendizaje incluye la capacidad del sistema para reconocer una serie de patrones complejos determinados por una gran cantidad de parámetros. En otras palabras, la máquina no aprende por sí misma, sino que aprende su propio algoritmo de programación y lo modifica ingresando continuamente los datos en la interfaz, para que pueda predecir la situación futura o tomar medidas automáticamente en función de determinadas situaciones. Dado que estas acciones son realizadas de forma autónoma por el sistema, se puede decir que el aprendizaje es automático sin intervención humana.

En la informática clásica, la única forma en que un sistema informático puede realizar una operación es escribir un algoritmo que defina el contexto y los detalles de cada operación. Por otro lado, los algoritmos utilizados en el desarrollo del aprendizaje automático realizan una gran parte de estas operaciones por sí mismos. Obtienen sus propios cálculos basados en los datos recopilados en el sistema, y cuantos más datos obtengan, mejores y más precisas serán sus acciones.[11]

Las computadoras usan este algoritmo para programarse a sí mismas hasta cierto punto. Como ingenieros, pueden diseñar nuevas respuestas informáticas en respuesta a la información proporcionada a través de su interfaz u otros medios. Todos los datos nuevos se convierten en un nuevo algoritmo, y cuantos más datos, mayor complejidad y eficiencia computacional puede proporcionar el sistema informático.



## **TIPOS DE MACHINE LEARNING**

Un sistema de aprendizaje automático computarizado utiliza la experiencia y la evidencia en forma de datos, a través de los cuales puede comprender patrones o comportamientos por sí mismo. De esta manera, puede construir pronósticos de proyectos o iniciar acciones como soluciones a tareas específicas. A partir de una gran cantidad de ejemplos de casos, se puede desarrollar un modelo que pueda inferir y generalizar el comportamiento que se ha observado, y hacer predicciones para nuevos casos en base a esto. Por ejemplo, podemos considerar la predicción del valor de ciertas acciones en el futuro en función del comportamiento de ciertas acciones en el período anterior.[11]

Existen tres tipos principales de Aprendizaje Automático:

**APRENDIZAJE SUPERVISADO:** Este tipo de aprendizaje se basa en la denominada información de formación. Entrene al sistema proporcionándole una cierta cantidad de datos y utilizando etiquetas para definir el sistema en detalle. Por ejemplo, proporcione una computadora con imágenes de gatos y perros con etiquetas que los definan. [9]

Otro método de desarrollo del aprendizaje automático implica el uso de diferentes parámetros para predecir valores continuos, que combinados con la introducción de nuevos datos pueden predecir un resultado determinado. Este método se llama regresión. Lo que hace que el aprendizaje supervisado sea diferente es que se utilizan diferentes ejemplos para resumir nuevos casos.[17]

Recordemos los 6 pasos principales en el desarrollo de un modelo de machine Learning:

- I. Recopilación de datos.
- II. Elección de una medida o indicador del éxito.
- III. Establecimiento de un protocolo de evaluación.
- IV. Preparación de los datos.
- V. Desarrollo de un punto de referencia del modelo.
- VI. Desarrollo de un buen modelo y ajuste fino de sus parámetros.



### *A. Regresión logística*

Cuando el algoritmo nunca converge al peso actualizado durante el proceso de entrenamiento, ocurrirá uno de los principales problemas de clasificación. Esto sucede cuando las clases no se pueden separar linealmente por completo. Por tanto, para hacer frente a problemas de clasificación binaria, la regresión logística es uno de los algoritmos más utilizados. La regresión logística es un algoritmo de clasificación simple pero poderoso (a pesar de su nombre). Funciona muy bien en clases lineales separables y se puede ampliar a la clasificación de clases múltiples mediante la tecnología.[10]

### *B. Máquinas de soporte vectorial (SVM)*

Este algoritmo puede ser considerado una extensión del algoritmo “perceptrón”. En SVM el objetivo de la optimización es establecer una línea de decisión que separe las clases maximizando el margen entre esta línea y los puntos de muestra cercanos a este hiperplano. Estos puntos se llaman vectores soporte.[9]

### *C. Algoritmo de árbol de decisión*

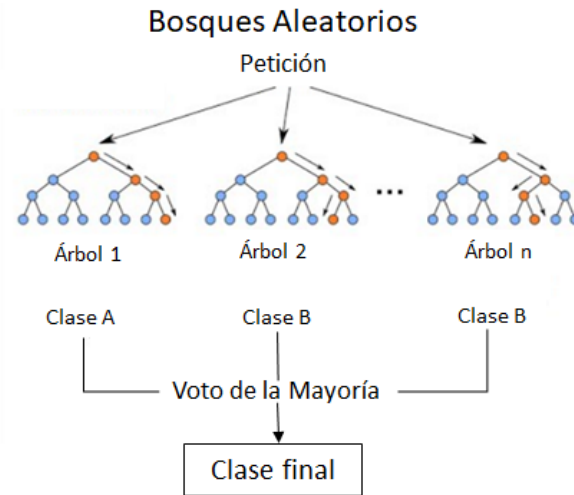
Los árboles nos brindan muchos beneficios necesarios para la supervivencia, como aire limpio, agua filtrada, sombra y comida. Incluso en las peores condiciones, nos brindan esperanza y conocimiento, así como el coraje para perseverar. Los árboles nos enseñan a seguir enraizando mientras volamos alto. Como todos sabemos, este árbol nos sirve de muchas formas diferentes, pero recientemente su estructura nos inspiró un algoritmo para resolver problemas y hacer que las máquinas aprendan lo que queremos que aprendan. Sí, estoy hablando de "algoritmos basados en árboles".[12]

### *D. Bosques Aleatorios (Random Forests)*

Si tenemos un conjunto de datos con muchas características (columnas), el algoritmo del árbol de decisión tiende a sobre ajustar, añadiendo complejidad al modelo y el proceso de aprendizaje. Podemos solventar este punto seleccionando cada columna de forma aleatoria y realizando árboles de decisión para cada conjunto de columnas. [12]







*Ilustración 6. Algoritmo Random Forest [12]*

El algoritmo realizará los siguientes pasos:

- ✓ Diseñar una muestra de arranque de tamaño  $n$ .
- ✓ Desarrollar un árbol de decisión desde la muestra de arranque. En cada nodo habrá características seleccionadas aleatoriamente sin reemplazamiento y el nodo se cortará maximizando la ganancia de información.
- ✓ El proceso previo se repetirá  $K$  veces.
- ✓ Agregar la predicción hecha para cada árbol, asignando la etiqueta de clase por votación mayoritaria.

### E. Los K-vecinos Más Cercanos (K Nearest Neighbors or KNN)

K El vecino más cercano (KNN) es un tipo especial de modelo de aprendizaje automático, generalmente llamado "algoritmo perezoso". Se les llama así porque no aprenden a distinguir conjuntos de datos mediante funciones de optimización, sino que recuerdan conjuntos de datos.

El nombre también se refiere a una categoría de algoritmos denominados "no paramétricos". Estos son algoritmos basados en ejemplos, que se caracterizan



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

por memorizar conjuntos de datos de entrenamiento, y el aprendizaje lento es un caso especial de estos algoritmos, que se asocia con un costo computacional cero durante el aprendizaje.[10]

El proceso que sigue el algoritmo es:

- ✓ Escoge el número de  $k$  y la distancia.
- ✓ Encuentra el  $k$  vecino más cercano de la muestra que se pretende clasificar.
- ✓ Asigna la etiqueta de clase por votación mayoritaria.

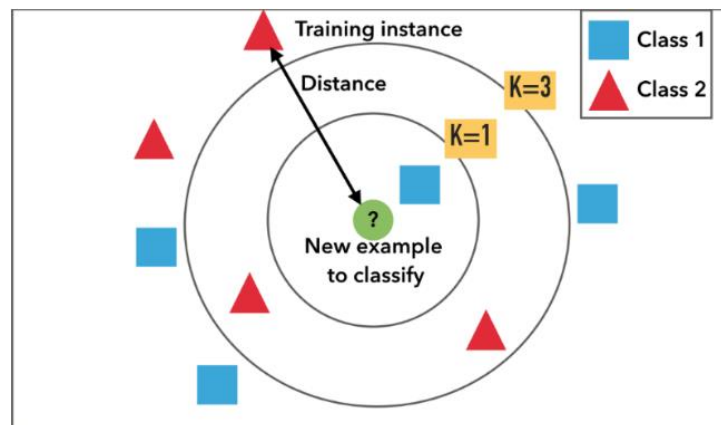


Ilustración 7. K-Vecinos [10]

El algoritmo predice las  $k$  muestras más cercanas al punto a clasificar en función de la distancia métrica. Su principal ventaja es que puede adaptarse a nuevos datos de entrenamiento y es un algoritmo basado en memoria. La desventaja es que el costo computacional aumenta linealmente con el tamaño de los datos de entrenamiento.

### 1.2.2.2 APRENDIZAJE SUPERVISADO

Cuando los datos "etiquetados" no están disponibles para la capacitación, se producirá un aprendizaje sin supervisión. Solo conocemos los datos de entrada, pero no hay datos de salida que correspondan a una determinada entrada. Por lo tanto, solo podemos describir la estructura de los datos en un intento de encontrar una determinada organización que simplifique el análisis. Por lo tanto, son de naturaleza exploratoria.[1]

Por ejemplo, las tareas de agrupación en clústeres encuentran grupos en función de la similitud, pero no hay garantía de que tengan algún significado o utilidad. A



veces, cuando navega por datos sin un objetivo claro, encontrará asociaciones falsas extrañas, pero poco realistas.[13]

El aprendizaje no supervisado se usa comúnmente para:

- Problemas de clustering
- Agrupamientos de co-ocurrencias
- Perfilado o profiling.

### 1.3 PROCESAMIENTO DIGITAL DE IMÁGENES

El procesamiento de imágenes digitales es Esta herramienta es ampliamente utilizada en la automatización de procesos industriales por las siguientes razones Demuestre confiabilidad, eficiencia y velocidad Procesando. La industria agrícola tiene Comienza a usar esta técnica Monitorear los aspectos relevantes de los cultivos.[14]

El desarrollo de algoritmos de procesamiento de imágenes. El campo de la ciencia, especialmente las aplicaciones relacionadas con la visión por computadora en el campo de la inteligencia artificial. La digitalización de datos nutre este conocimiento Capture videos, fotos, etc. en el dispositivo. De esta forma, la imagen se vuelve El núcleo de la descripción de la realidad. Del mismo modo, el uso de comerciales y su digitalización en el mercado minorista también brindan oportunidades para la aplicación de técnicas analíticas. Imágenes lógicas para identificar negocios, operaciones y oportunidades comerciales Generación de conocimiento.[15]

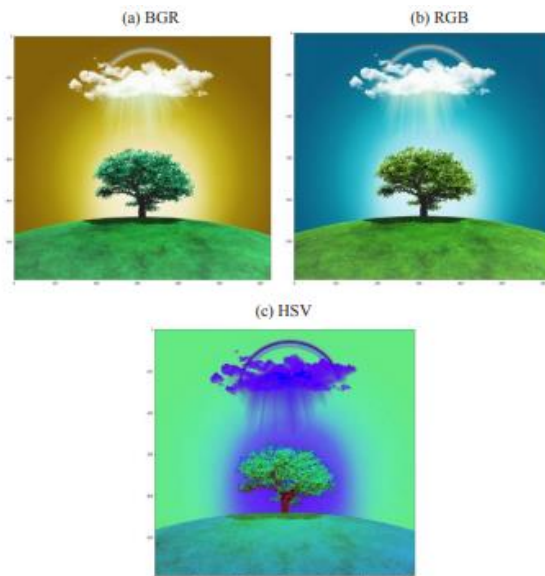
#### 1.3.1 ESPACIO DE COLOR

En el espacio de color más común utilizado para el procesamiento de imágenes, el rojo es Verde y azul (RGB). Según los estándares técnicos de la Comisión Internacional de Iluminación. Según los estándares técnicos de la Comisión Internacional de Iluminación CIE 1931-RGB, el color se compone de tres tamaños monocromáticos perceptibles Visión. Sin embargo, estos no son los únicos espacios de color. Entre otras cosas Espacio HSV (valor de saturación de tono) y espacio (L \* a \* b), más inclinado a describir La percepción del color ante ciertas dimensiones de una imagen cambia.[16]

elige uno El espacio de color depende del uso previsto de la imagen (si Es necesario analizar la superficie a utilizar. En este tipo La aplicación utiliza combinaciones CMYK (cian, magenta y clave), de las cuales hay muchas Una



utilidad para comparar patrones impresos. Para el campo de procesamiento El análisis de píxeles en la imagen utiliza especialmente el espacio de color RGB. En la imagen.[16]



*Ilustración 8. Imagen en espacio RGB-BGR-HSV [16]*

### 1.3.2 PIXEL

Cada imagen consta de un conjunto de píxeles. Pixel es la unidad básica de píxel. Analiza la imagen. No hay elementos más pequeños y refinados. La imagen es mejor que los píxeles. Suele estar relacionado con el "color" o la "intensidad". Si pensamos en una imagen, actúa como una cuadrícula y cada cuadrado de la cuadrícula contiene un píxel. Por ejemplo, una imagen con una resolución de 500 x 300 significa que la imagen consiste en una matriz de píxeles con 500 filas y 300 columnas.[17]

La mayoría de los píxeles se representan de dos formas: escala de grises y color. En una imagen en escala de grises, el valor de cada píxel está entre 0 y 255, donde cero corresponde al negro, 255 corresponde al blanco. Valor entre 0 y 255 significa cambios en los tonos de gris, donde los valores cercanos a 0 son más oscuros, y el valor cerca de 255 es más ligero. Los píxeles de color suelen estar en el espacio de color; el más utilizado es el denominado RGB, que es el espacio de color. El valor de la parte roja, el valor de la parte verde, el valor de la parte azul. Cada uno de estos tres valores de color está representado por un número entero entre 0 y 255. Los píxeles solo deben estar en el rango de [0, 255], normalmente se utilizan números enteros sin signo 8 bits que representan la intensidad de cada color.



# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Al formar un vector RGB (rojo, verde, azul), el vector representa el color; por lo tanto, para Construir blanco, el vector tendrá rojo, verde y azul respectivamente, Deje: (255, 255, 255), luego, para el negro, obtendremos el vector (0, 0, 0).

De esta forma, cada píxel se define en toda la cuadrícula de la imagen. Esta cuadrícula es Convertir a sistema de coordenadas, siempre comenzando desde la esquina superior Izquierda, con las coordenadas del punto (0, 0). A partir de ahora, El área de configuración de los ejes x e y, varios cálculos numéricos que se pueden realizar Adecuado para fotografías.[18]

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0, n-1) \\ f(1,0) & f(1,1) & f(1, n-1) \\ f(m-1,0) & f(m-1,1) & f(m-1, n-1) \end{bmatrix}$$

Ecuación 1

## 1.3.3 ETAPAS EN EL PROCESAMIENTO DE IMAGEN

La etapa de procesamiento comienza en el orden en que se capturan las imágenes, y luego Reprocesa esto. Si el análisis es para el proceso de aprendizaje Imágenes continuas (por ejemplo, videos) para aprender La inteligencia artificial captura la carga eléctrica con tecnología de mayor enfoque y claridad Relacionado.[18]

En cualquier caso, se requiere un procesamiento previo, y el objetivo de esta etapa Consiste en reducir el "ruido" modificando la relación de la imagen y limitando el valor, Arriba o abajo, la intensidad de cada píxel en su área correspondiente. como siempre Aplicar difusa y binarización, pero determinar si es necesario Las características de la imagen deben posicionarse para su preprocesamiento. Busque y extraiga una lista de detectores o características, que son regiones de píxeles Tiene un valor que satisface la función de umbral y puede Si busca alta similitud (homografía geométrica) o alta similitud (por ejemplo, El detector realiza una "coincidencia sólida".[15]

Si el análisis es para comparar dos imágenes, se requerirá un paso de recorte O subdividido. Entre las técnicas de pretratamiento más utilizadas, El procesamiento de binarización y tramado se describen en detalle a continuación.

Los métodos de mejora de imágenes se dividen en dos categorías: En el dominio del espacio y la frecuencia, algunos se basan en Combinación de las dos primeras categorías



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Los métodos en el dominio espacial se aplican al plano de la imagen en sí, estos se basan en la manipulación directa de píxeles de imagen. El principal procesamiento del dominio espacial es la intensidad actúa sobre un solo píxel (especialmente para Contraste y umbral) y en los píxeles que pertenecen a la imagen. Generalmente, la tecnología de dominio espacial es más computacional se puede implementar de manera eficiente y requiere menos recursos de procesamiento. Métodos para realizar operaciones en el dominio de la frecuencia. La transformada de Fourier de la imagen, no la imagen en sí.[14], [19]

### 1.3.3.1 ESCALA DE GRIS

Antes de procesar la imagen, generalmente busca un área en la imagen. Puede considerarse como un pico, o los píxeles se clasifican como miembros con el mismo patrón y área o miembros que no pertenecen al mismo patrón y área. Una de estas técnicas de preparación es la binarización, que se realiza mediante cada píxel recibe una imagen en escala de grises con un valor entre 0 y 255.[20]

Utilice esta técnica para crear una representación binaria de la imagen, buscando enfatice áreas específicas. La ventaja de utilizar esta técnica es a cada píxel que considera el umbral se le asigna un valor de cero o 255, en el caso de que el valor de píxel sea menor o mayor que el umbral.[20]



Ilustración 9. Imagen Binarizada [16]



# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Utilice esta técnica para crear una representación binaria de la imagen, buscando Enfatice áreas específicas. La ventaja de utilizar esta técnica es A cada píxel que considera el umbral se le asigna un valor de cero o 255, En el caso de que el valor de píxel sea menor o mayor que el umbral.

220	226	237	183	153	170	155	74	154	230	228	222	222	229	228	228	228	227	227	227	228
226	223	227	183	164	138	159	87	202	226	226	227	224	227	229	229	227	226	227	228	228
218	210	206	129	120	92	143	74	196	226	231	226	226	227	230	230	226	228	222	220	228
174	181	203	169	117	178	120	61	119	224	224	226	228	223	220	225	223	228	226	222	219
164	189	183	168	168	171	158	95	183	223	226	226	225	226	223	228	226	222	228	225	176
173	191	232	199	141	142	151	211	225	228	232	223	208	140	192	225	224	221	222	160	120
153	167	224	176	113	89	86	218	220	227	224	211	128	127	224	226	227	224	140	150	226
205	232	227	228	204	152	151	206	208	187	233	215	152	180	226	224	229	165	164	208	181
218	232	226	229	241	175	179	167	130	128	181	217	172	201	222	177	182	135	154	111	126
229	224	227	187	197	176	227	225	164	128	187	126	140	206	212	167	151	120	224	169	121
212	223	221	223	168	229	205	203	224	184	117	137	142	186	176	188	159	186	200	185	119
184	207	222	220	206	217	208	115	125	124	206	149	164	129	199	195	200	124	163	146	121
168	224	224	169	211	190	185	152	125	122	189	117	87	81	183	180	111	104	128	124	120
120	120	170	140	127	121	120	202	124	183	187	172	183	187	162	180	121	97	87	91	110
127	140	124	140	133	142	89	124	100	165	126	146	124	123	144	177	89	98	112	92	98
115	128	129	121	204	154	167	240	111	124	120	146	107	76	88	83	101	63	78	112	112
167	126	120	127	98	124	98	88	120	122	122	120	120	122	163	128	107	88	80	92	144
117	83	88	123	103	83	88	109	112	128	124	121	81	128	88	167	91	100	90	91	126
162	87	82	82	87	84	83	82	82	82	119	164	175	142	76	140	73	202	72	84	78

Ilustración 10. Rango de grises de una imagen [16]

### 1.3.3.2 SEGMENTACIÓN POR BINARIZACIÓN

La binarización de fotos se usa ampliamente en reproducción de imágenes, visión artificial, segmentación de caracteres y reconocimiento de entropía (OCR). de Utilice el sistema de puntos para dibujar pizarrones en blanco y negro, Además de habilidades, paciencia y largo tiempo. Es por eso que intentar acortar En el proceso, analizamos el uso de algoritmos en este trabajo. Binarización y difuminado. El resultado es obviamente no El entusiasmo y la calidad de la exquisita artesanía del exquisito reloj hecho a mano, pero muy acertado. Tiempo de procesamiento.[21]

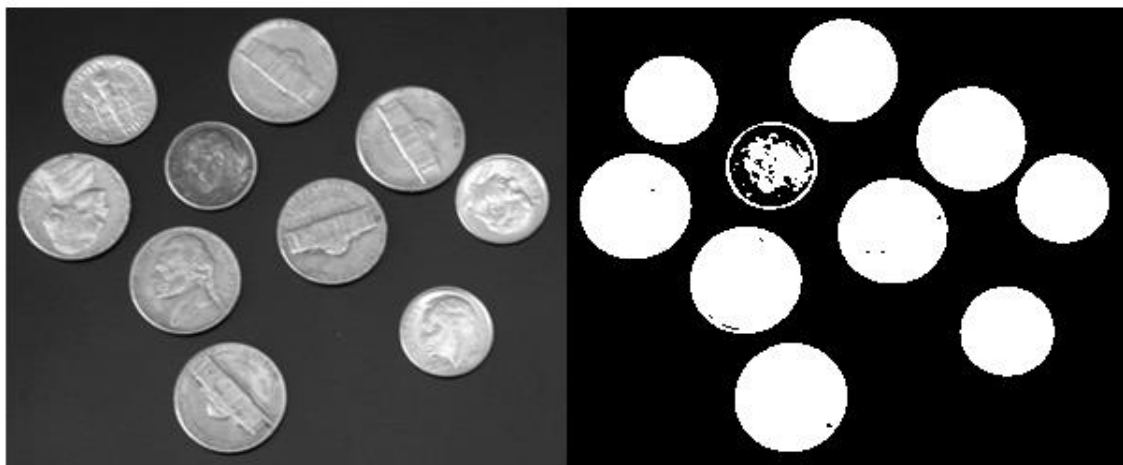


Ilustración 11. Imagen binarizada. [21]



Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval

El proceso de binarización es una técnica de segmentación. Cuando hay una gran diferencia entre el objeto de análisis y el fondo de la imagen, la técnica se utiliza para evaluar la similitud y la diferencia (nivel de gris) entre píxeles, y luego se establece el umbral para optimizar. Para lograr la mejor separación de objetos. El proceso de binarización finaliza comparando la intensidad con el umbral seleccionado, si la intensidad del píxel es mayor que la intensidad del umbral, se asigna como blanco, en caso contrario se asigna como negro.[22]

Al seleccionar el umbral, ocurren los siguientes problemas: la sombra causada por la posición del objeto en relación con la fuente de luz, el ruido en la imagen, la rugosidad en el borde y la eliminación del área causada por el área de baja iluminación. Todos estos problemas pueden hacer que el borde se mueva, lo que lleva a la modificación de la posición del ROI, lo que puede cambiar el resultado final en el sistema de clasificación. [20]

### **1.3.3.3 DIFUMINADO**

Una técnica de procesamiento es Difuminado, que incluye Valor de píxel medio con valor de intensidad de píxel alrededor. Esta mezcla de intensidades produce una imagen borrosa. Diferente a Tecnología de desenfoque para fotografías que desea que se vean más nítidas Ayudar al procesamiento de imágenes en tareas como el etiquetado de las dimensiones del área Sobre la imagen y la detección de bordes. Entre varios métodos de desenfoque, existen Puede mencionar el método gaussiano, que utiliza Píxeles que rodean el píxel central para que esté más cerca del píxel central El peso del centro es mayor.[14]







*Ilustración 12. Imagen difuminada [16]*

El promedio se realiza convolucionando la imagen con el filtro de caja normalizado. Por lo tanto, el filtro obtiene el valor promedio de todos los píxeles debajo del área del núcleo y reemplaza el elemento del medio con este valor promedio. Una alternativa es usar las funciones `cv2.blur ()` o `cv2.boxFilter ()`. Al usar estas funciones, debemos especificar el ancho y alto del kernel. El filtro de caja normalizado  $3 \times 3$  es el siguiente:

$$K = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

*Ecuación 2*

#### **1.3.3.4 HISTOGRAMA DE GRADIENTES**

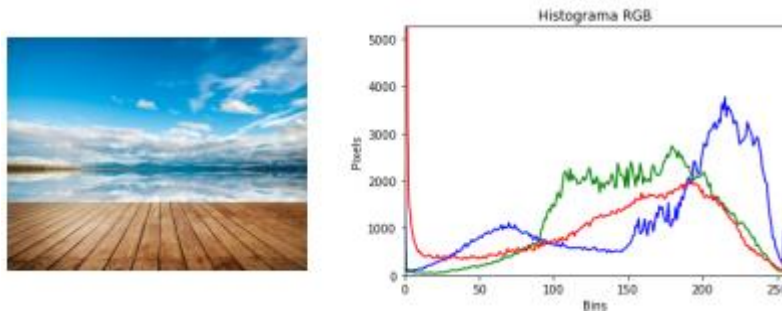
Es común que las imágenes de entrada tengan demasiada información adicional sin necesidad de clasificación y análisis adicional. Entonces el primer paso en la clasificación de imágenes, la imagen se simplifica extrayendo información importante que contiene, el resto se deja de lado. Este paso se llama extracción característica. En los métodos tradicionales de visión por computadora, estas funciones son fundamentales para el rendimiento del algoritmo. Algunas técnicas utilizadas en la visión por ordenador son los detectores de características de Haar (Wang, 2014), Histograma de Gradientes Orientados (HOG), Transformación de Caracteres Invariantes (SIFT), Función Robusta Acelerada (SURF), entre otras.[16]



**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

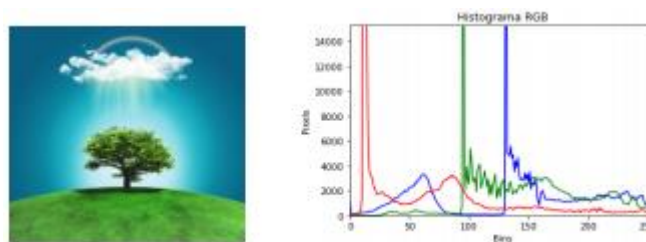
El histograma representa la distribución de la intensidad de los píxeles (color o escala de grises). s nivel La intensidad del color (valor de píxel). En el espacio de color RGB de los valores de píxeles El rango es de 0 a 255. Al dibujar un histograma, el eje X tiene 256 bins y luego Efectivamente, estamos contando el número de veces que cada píxel produce este valor. Por el contrario, si solo se utilizan dos segmentos (espaciados uniformemente), entonces Contará el número de veces que el píxel está en el rango de [0, 128] o [128, 255]. Solo revisa el histograma de la imagen para entender Hablando en general de distribución de contraste, brillo e intensidad.[16]

Por ejemplo, al examinar el histograma de color en las figuras A y B se observa un pico agudo en el histograma azul alrededor de bin 210. Esto indica un valor azul más claro debido a la imagen del cielo.



*Ilustración 13. Histograma RGB imagen A [16]*

Por otro lado, en la Figura B (respectivamente el histograma), se observan tres picos de color. El pico rojo obedecía a la sombra del árbol. Y los picos en los colores Verde y azul, porque hay uniformidad en la misma intensidad de color.[16]



*Ilustración 14. Histograma RGB imagen B [16]*

$$\nabla f = \begin{bmatrix} gx \\ gy \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix}$$

*Ecuación 3*



# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

La derivada de una imagen puede ser aproximada por diferencias finitas. Para evitar distorsionar la imagen, se aplica un filtro 3 x 1:

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

Ecuación 4

La dirección de la gradiente y la magnitud se obtiene de las fórmulas

$$\theta = \tan^{-1} \left[ \frac{gy}{gx} \right], \theta \in [0^\circ, 180^\circ] \text{ y } \sqrt{gy^2 + gx^2}$$

Ecuación 5

En cada píxel de una celda de 8 x 8, se conoce el gradiente (tamaño y dirección), y Por lo tanto, hay 64 tamaños y 64 direcciones, que son 128 números. Histograma Al asignar estos 128 números al histograma de 9 bits (es decir, 9 digital). El intervalo del histograma corresponde a la dirección del gradiente y su valor es 0, 20, 40 ... 160 grados. Si la dirección del degradado en el píxel es exactamente 0, 20, 40 ... o 160 grados, el histograma asigna esta puntuación al bin correspondiente. Un píxel El voto donde la dirección del gradiente no es exactamente 0, 20, 40 ... 160 grados Cambie entre los dos contenedores más cercanos según la distancia desde el contenedor inicial.[16]

## 1.4 REDES NEURONALES

En el contexto del aprendizaje automático, una de las tecnologías más famosas es La más famosa hoy en día es la red neuronal. Artificial (ARN). Estos algoritmos se remontan a las décadas de 1940 y 1950, Hasta ahora, debido a Los recursos informáticos que necesitan y la gran cantidad de datos que contienen Depende de su función normal. Actualmente, son Considerada una de las mejores tecnologías en muchas aplicaciones. Incluyendo visión por computadora o procesamiento del lenguaje natural.[12]

Como sugiere el nombre, la idea de una red neuronal es intentar Imita el cerebro humano. Aunque no sé cómo El funcionamiento del cerebro humano, ciertas características, como La estructura producida e implícitamente inspirada por estas observaciones Estos algoritmos. Específicamente, el algoritmo consta de un conjunto de Las unidades llamadas neuronas están conectadas entre sí para la transmisión. Información sobre los demás. Al ingresar datos / información transversal Esta red neuronal eventualmente producirá los siguientes resultados o valores Nos interesa la salida. Para que puedas entender la red La neurona es



# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

una función importante, producirá algún resultado relacionado Ingrese datos, pero esta función depende de una variedad de Inventa sus subfunciones.[23]

En su versión más básica, una sola red neuronal puede ser Imagina un modelo de clasificación general, como La sección anterior. Sin embargo, cuando hay varias neuronas en la red El modelo es algo complicado. Ahora, la salida de la red ya no es Solo tenemos una neurona, pero como hay más de una, necesitamos otra Calcule la neurona de salida en función de la salida de otras neuronas. está, Es decir, se agrega una neurona, que se conecta al resto de neuronas. red. Se dice que esta neurona pertenece a otra capa de la red porque necesita Toma lo que otras personas aportan como entrada y luego sigue.[23]

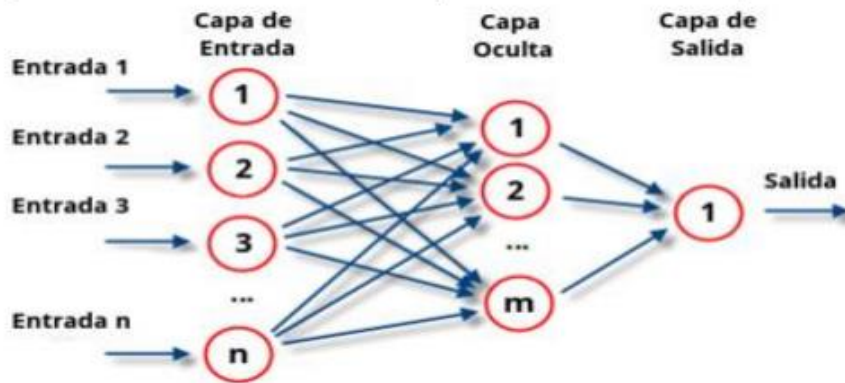


Ilustración 15. Estructura red neuronal multicapa[23]

## 1.4.1 REDES NEURONALES CONVOLUCIONALES (CNN)

Una vez explicadas las operaciones más básicas de las redes neuronales, comenzamos a comentar ¿Qué es una red neuronal convolucional (CNN), y luego enfatizar la red neuronal convolucional basada en regiones (R-CNN), sentando así las bases de la red neuronal? Utilizado en nuestro proyecto YoloV3.



**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

CNN es una red neuronal muy eficaz para la detección de objetos y las tareas de clasificación y segmentación de imágenes. Tienen una arquitectura similar a las redes neuronales. Ordinario, la principal diferencia es la diferencia en la apariencia de la capa dedicada Extrae funciones automáticamente. Para estudiar imágenes, CNN Contiene varias capas ocultas dedicadas a detectar diferentes características Nivel de detalle. Esto significa que la primera capa puede detectar formas simples. Líneas y curvas hasta llegar a niveles más profundos Reconocer formas complejas, como caras o formas de animales. La capa oculta es Dividido en las siguientes categorías: [24]

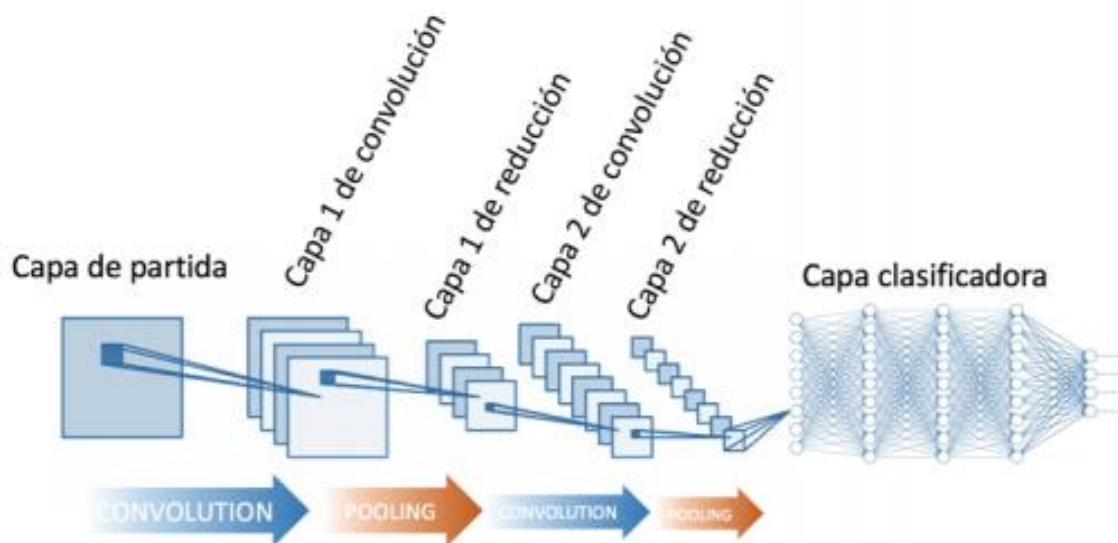
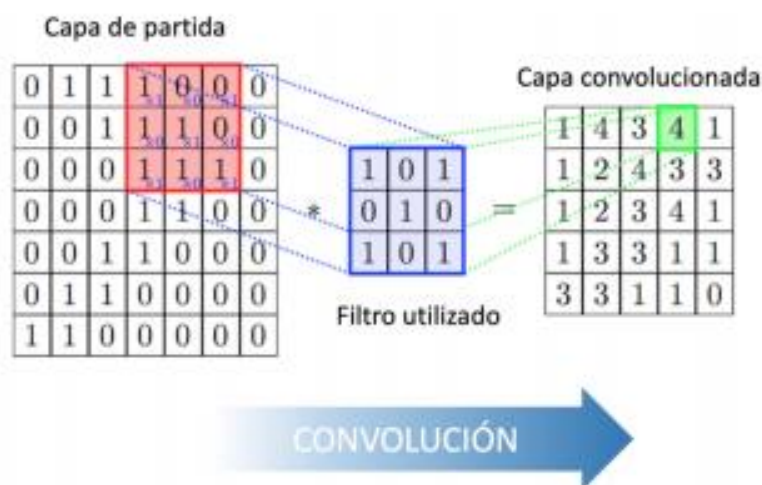


Ilustración 16. Capas de una neurona convolucional [24]

**1.4.1.1 CAPAS CONVOLUCIONALES.**

Esta capa es la diferencia entre CNN y cualquier otra red neuronal, esto se debe a que Aplicación de la operación de convolución. La convolución implica aplicar filtros o núcleos Ingrese la imagen para obtener el mapa de características de la imagen original. Una especie El kernel se deslizará en la imagen de entrada, por



lo que se desplazará de izquierda a derecha y de arriba a abajo. Examine la imagen completa de arriba a abajo. Producto entre cada Elementos de imagen y elementos de filtro que se superponen y agregan en esta posición resultado.[24]

*Ilustración 17. Aplicación de capa de un filtro convolucional [4]*

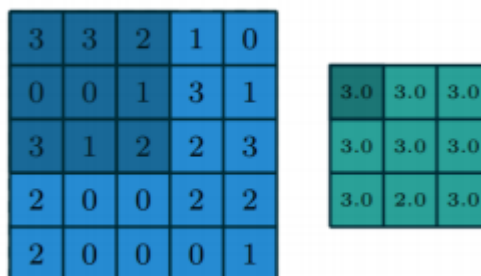
Luego de aplicar este operador, transforma la imagen en una forma con ciertas características 12 más avanzados Al tener valores más altos, se vuelven más ventajosos en imágenes convolucionales. En los píxeles que los representan. Estas características vendrán determinadas por la forma del filtro.[24]

### 1.4.1.2 CAPAS DE POOLING:

La capa de convergencia es otra parte importante de CNN. Capa de agrupación Reducir el tamaño de los mapas de características mediante el uso de determinadas funciones Resume la partición.[25]

Existen tres formas de principales de aplicar el pooling.

- a. Max-pooling: devuelve el máximo valor de entrada de todos los pixeles cubiertos por el kernel
- b. Min-pooling: devuelve el valor mínimo de entre todos los píxeles cubiertos por el kernel
- c. . Average pooling: se calcula el valor medio de entre todos los píxeles cubiertos por el kernel



*Ilustración 18. Matriz maxpooling [25]*

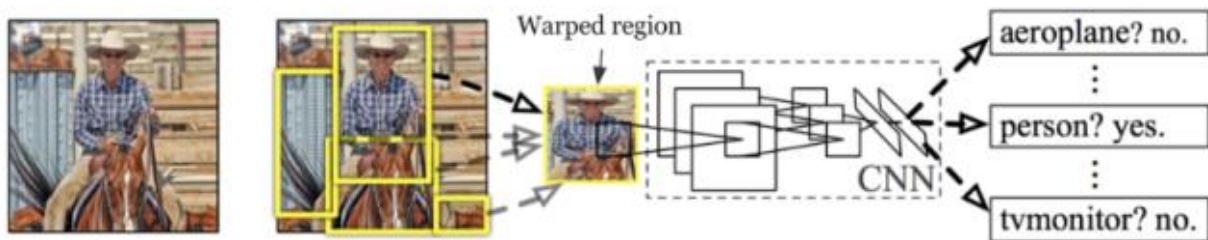
### 1.4.1.3 CAPA CLASIFICADORA

Después de aplicar capas sucesivas de convolución y agrupación, la red utiliza Normalmente, las capas completamente conectadas se utilizan para realizar inferencias. El número de neuronas en la última capa completamente conectada es el mismo que Quiere predecir. [25]



### 1.4.2 REDES NEURONALES CONVOLUCIONALES BASADAS EN REGIÓN (R-CNN)

La idea principal de R-CNN es detectar objetos en subregiones de la imagen principal. Primero, realice una búsqueda selectiva para identificar el área La región de interés, finalmente, extrayendo Las características de CNN de cada región son independientes. [24]



*Ilustración 19. Arquitectura R-CNN [24]*

## 1.5 YOLO V3

You Only Look Once (Yolo) es un tipo de R-CNN que, tal y como su nombre indica, Solo es necesario "ver" la imagen una vez para el intercambio de alta velocidad. Pérdida de precisión. YoloV3 es la última versión de Yolo, la fecha de lanzamiento es 2018. En comparación con el algoritmo de reconocimiento, el algoritmo de detección no solo puede predecir la categoría del objeto, sino también la posición del objeto en la imagen. YoloV3 es capaz de detectar múltiples objetos en una imagen a través de una sola red. Neuronas, Además, es uno de los algoritmos de detección más rápidos que puede posicionarse en La mejor opción para la inspección en tiempo real. [26]

Por tanto, las características que lo hacen de interés en relación a nuestro proyecto, son:

- Elevado porcentaje de acierto.
- Proporciona la posición en la imagen del objeto detectado.
- Integración sencilla en OpenCv.



- Gran velocidad de detección.

Las tareas específicas de YoloV3 se explican a continuación para comprender cómo funciona.

En el artículo “Recent advances in image processing techniques for automated leaf pest and disease recognition – A review” (Avances recientes en las técnicas de procesamiento de imágenes para la infección automatizada de plagas de hojas y la enfermedad de reconocimiento – Una revisión) nos da una idea de la capacidad de este algoritmo para la detección de hongos usando redes neuronales profundas y YoloV3 son capaces de detectar plagas en hojas.

En el artículo “*Uso del algoritmo de aprendizaje profundo YOLO v4 basado en poda de canales para la detección precisa y en tiempo real de flores de manzana en entornos naturales*” nos hablan en esta investigación, el conjunto de datos de flores de manzana etiquetado manualmente se utilizó en primer lugar para ajustar la red de detección de objetos YOLO v4 mediante el aprendizaje por transferencia para realizar la detección rápida y precisa de flores de manzana. Luego, los canales redundantes y los parámetros de peso en el modelo de detección de flores de manzana entrenado YOLO v4 se recortaron mediante el algoritmo de poda de canales, para simplificar la estructura y los parámetros del modelo con la premisa de garantizar la precisión del modelo.

### 1.5.1 YOLO V3-TINY

En YoloV3-tiny se realiza downsampling en las capas Max-Pooling. En cambio, en YoloV3 se lleva a cabo en las capas de convolución.

El primer parámetro a anotar es la velocidad de detección de cada imagen. Aumentar la imagen que se detecta depende en cierta medida de las características de la computadora donde realizar las pruebas, ahora, para calcular la velocidad de detección, use el siguiente script. Estudie la imagen de Python y los parámetros del modelo (cfg y peso). El script calcula el tiempo promedio requerido para realizar el descubrimiento sin considerar el tiempo necesario para cargar los pesos de la red, porque este paso es ejecutar solo una vez. De esta forma, el resultado es relativamente alto. Detección. Para obtener una medición de tiempo confiable, Después de mil pruebas, estos dos modelos.[26]

Usando yoloV3-tiny, la velocidad de detección es 10 veces más rápida. A esa velocidad será capaz de detectar a una velocidad de 27 imágenes por segundo, que es mejor de lo esperado en este sentido, mientras que yoloV3 solo puede procesar 2.5 imágenes cada segundo.





**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

Modelo	Tiempo de detección medio(s)
YoloV3-tiny	0.037
YoloV3	0.401

*Tabla 1. Velocidad de detección media YoloV3 y YoloV3-tiny. [26]*

Dado que la velocidad es fundamental, la calidad de estas pruebas aún debe comprobarse. Pero no sirve de nada sin precisión.



*Ilustración 20. Detección YoloV3 y YoloV3-tiny. [26]*

La precisión de detección cuando se usa YoloV3-tiny [ilustración 20] es demasiado baja, lo que Vehículos no identificados, incluso vehículos diferentes en la misma zona. Por eso, a pesar de la increíble velocidad de este modelo, no se puede utilizar para nuestro propósito.

### 1.5.2 PREDICCIÓN DE BOUNDING BOX

YoloV3 no puede predecir directamente el ancho y la altura del borde, porque de hecho Causará pendientes inestables durante el entrenamiento. La red neuronal devuelve 4 Las coordenadas de cada cuadro delimitador tx, ty, tw, th. Si el píxel está fuera de ángulo La distancia en la esquina superior izquierda es cx y cy, y



*Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval*

el ancho y alto del cuadro delimitador anterior son  $p_w$  y  $p_y$ , entonces la predicción se corresponde con la Ecuación. [26]

$$bx = \sigma (tx) + cx$$

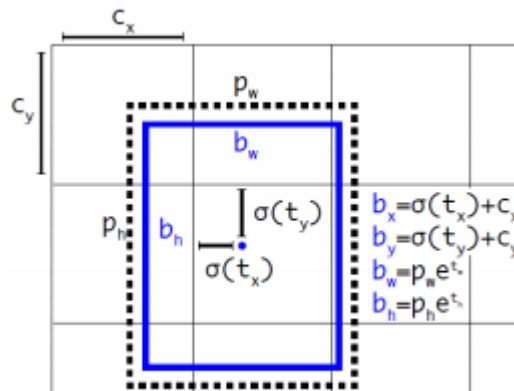
$$by = \sigma (ty) + cy$$

$$bw = pwe^{tw}$$

$$bh = phe^{th}$$

*Ecuación 6*

Donde  $b_x$ ,  $b_y$ ,  $b_w$  y  $b_h$  el centro de coordenadas  $(x,y)$ , y el ancho y alto de la predicción respectivamente.



*Ilustración 21: Bounding boxes con dimensiones priors y localización de predicción*

En la Figura 20, puede ver el ancho y el alto de la caja. Detectado como el desplazamiento del centroide del grupo. Previsión central Las coordenadas de la caja se basan en la ubicación de la aplicación del filtro. Utiliza la función sigmoidea.[26]

### 1.5.3 PREDICCIÓN DE LAS CLASES

Cada cuadro usa uno para predecir la clase a la que puede pertenecer Clasificador con múltiples etiquetas. La función softmax ya no se utilizará como en Yolo Arriba, porque se encontró que es innecesario hasta cierto punto excepto por un buen desempeño Suponiendo que cada caja tiene solo una clase, generalmente No dado. Por ahora, solo se utilizarán clasificadores lógicos. Durante este período En el entrenamiento, la función de pérdida de entropía cruzada binaria se usa para predecir la categoría. Esta fórmula ayuda a trabajar



Tener un conjunto de datos complejo, que puede tener etiquetas superpuestas, como mujeres y personas, O coches y camiones.[26]

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

*Ecuación 7*

Donde

- N es el número de clases
- y es la etiqueta de la muestra (1 correcto, 0 no)
- pi es la probabilidad de pertenencia

#### **1.5.4 PREDICCIÓN A TRAVÉS DE ESCALAS**

YoloV3 puede predecir tres proporciones diferentes de fotogramas. Extracción del sistema Utilice conceptos similares a las redes piramidales de características para determinar las características de estas escalas

Lo que significa que con una imagen entrada de 416x416 (esta es la entrada en nuestro proyecto), se hacen detecciones en escalas de 13x13, 26x26 y 52x52.

La red reduce el tamaño de la imagen de entrada a la primera capa de detección, donde Utilice el paso 32 para realizar la detección utilizando el mapa de características de la capa (primero escala). Luego, cada capa es muestreada por un coeficiente x2 y se compara con El mapa de características de la capa anterior. Ahora usa Paso 16 (segunda escala). Finalmente, repita el muestreo y use Paso 8.

El autor afirma que esto ayuda a la detección de objeto pequeños, mejorando este tipo de detecciones frente a las versiones previas de Yolo.



### 1.5.5 ARQUITECTURA DE LA RED

Yolo está formado únicamente por capas convolucionales, lo que es apodado como Fully Convolutional Network (FCN).

En YoloV3 se utiliza una nueva arquitectura más profunda que la de su predecesor, llamada Darknet-53, la cual, como su nombre indica está formada por 53 capas convolucionales. [26]

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	128 × 128
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	64 × 64
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	32 × 32
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	16 × 16
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	8 × 8
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

*Ilustración 22 Arquitectura YoloV3 [26]*

## 1.6 RASPBERRY PI 4



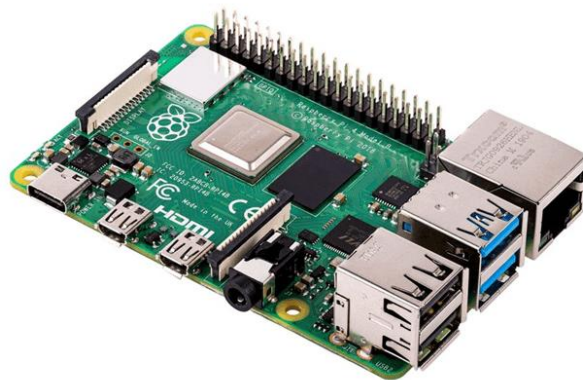
*Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval*

### 1.6.1 ¿QUÉ ES?

Este sistema informático es casi del tamaño de una tarjeta de crédito y consta de una computadora de placa única. Esta computadora no solo está compuesta por el hardware básico (procesador, unidad de almacenamiento, etc.) de la computadora, sino que también incluye varios tipos de computadoras. Conexión (USB, HDMI, video, sonido, etc.). Tiene una ranura para insertar una tarjeta microSD, que actúa como un disco duro y contiene un sistema operativo. En este sentido, aunque se pueden utilizar otras distribuciones de Linux o versiones especiales de Windows, se recomienda utilizar un sistema Raspbian basado en Debian. La energía es proporcionada por un cargador microUSB (como un teléfono inteligente) y se puede establecer una conexión a Internet a través de una interfaz Ethernet a través de un cable de red.[27]

Otros elementos como un mouse, teclado o disco duro externo también se pueden conectar a través de la ranura USB, y el puerto HDMI es la forma más fácil de conectar el monitor a la Raspberry Pi. También hay varios pines (pines de contacto) que se pueden programar con otras funciones.

El nombre de la computadora es la palabra inglesa *gameplay*, en referencia a la Raspberry Pi. La primera parte de su nombre sigue la tradición de las empresas informáticas, que se caracterizan por los nombres de frutas como manzanas, moras o bellotas. La segunda parte Pi se refiere a la abreviatura de "intérprete de Python", porque Python es el principal lenguaje de programación de Raspberry Pi.[19]



*Ilustración 23. Placa Raspberry pi4 [27]*

### 1.6.2 VENTAJAS Y DESVENTAJAS

Algunas de las ventajas del uso de Raspberry son:

- ❖ Trabaja con software libre, lo cual ahorra el pago de licencias.
- ❖ Bajo consumo de energía y bajo costo.
- ❖ Diseño compacto con salidas digitales y analógicas para visualizar su entorno gráfico: RCA o HDMI.



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

---

- ❖ Capacidad de navegación por internet y puede funcionar como un pequeño servidor dependiendo de la aplicación.

Y algunas de sus limitaciones son:

- ❖ Si bien no es costoso, su hardware es limitado. Se necesita conseguir dispositivos de comunicación como el teclado, monitor y ratón.
- ❖ No posee receptor WiFi, ni de bluetooth.
- ❖ No puede alimentar gran cantidad de periféricos en caso de una alta demanda de corriente.
- ❖ Es algo lento al usarse en procesos que necesiten mucho procesamiento.
- ❖

### 1.6.3 CARACTERÍSTICAS RPI4

- Procesador Broadcom BCM2711, cuatro núcleos Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz.
- Gigabit Ethernet.
- x puertos USB 3.0 y 2 x USB 2.0.
- Conector estándar Raspberry Pi GPIO de 40 pines (compatible con placas previas)
- 2GB LPDDR4-2400 SDRAM (según modelo).
- 2.4 GHz y 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE.
- Puertos micro-HDMI x 2 ports (soporte pantallas hasta resolución 4K y 60 fps)
- Conector MIPI DSI display port
- Conector 2 líneas MIPI CSI camera port
- Sonido estéreo y vídeo compuesto
- Soporte para vídeo H.265 (resolución 4k a 60 fps decodificación), H264 (1080p y 60 fps decodificación, 1080p 30 fps codificación)
- Gráficos OpenGL ES 3.0
- Lector de tarjetas Micro-SD



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

- Conector USB tipo C para alimentación 5V DC

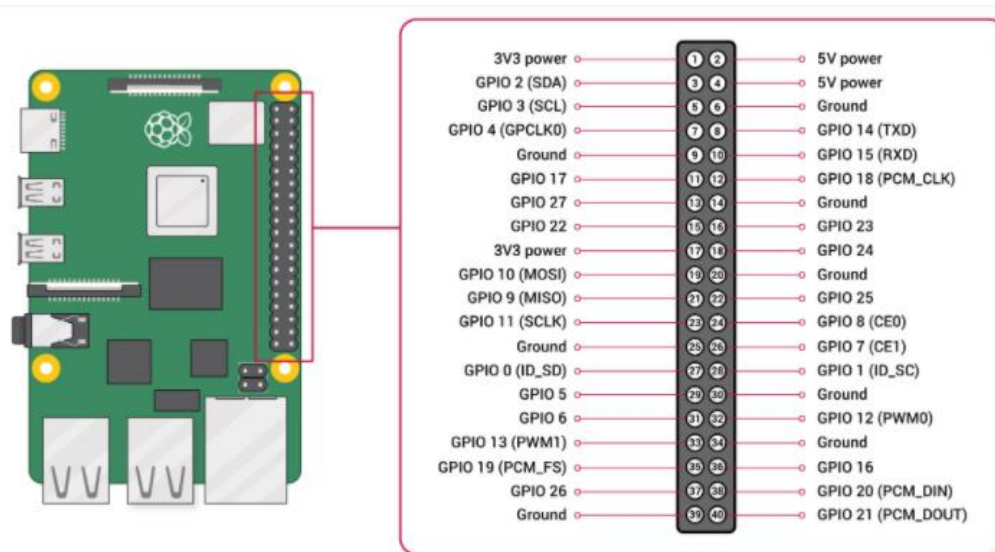


Ilustración 24. Diagrama de conexión de RPI4 [27]

## CAPÍTULO 2



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

## 2. METODOLOGÍA





# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

En la propuesta original del presente trabajo de grado se planteó la siguiente metodología:



A continuación se expondrá el trabajo realizado en cada una de las fases de la metodología

## 2.1. EXPLORACIÓN PREVIA:

Durante el desarrollo de esta fase se consultaron varias fuentes como artículos científicos, proyectos que involucraran la inteligencia artificial con la agricultura y



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

se tomaron en cuenta la opinión de varias personas con experiencia en el cultivo de cannabis medicinal para así poder tener en claro las actividades que se llevarían a cabo y las herramientas necesarias para ello.

Algunos temas que se consultaron fueron, la siembra correcta de la planta de cannabis en climas frío y cálido que es nuestro objeto a estudiar, los parámetros necesarios para la construcción de un lugar adecuado y necesario para la siembra de la planta, en este lugar se realizara la validación y puesta en marcha del proyecto, se consultaron todo lo relacionado con machine Learning para la detección de distintos hongos que se puedan desarrollar en la planta.

### 2.2. ADECUACIÓN DEL SITIO DE DESARROLLO:

para el desarrollo de esta fase se tuvieron en cuenta varios aspectos, desde la correcta germinación de la semilla de cannabis estas nacen normalmente en un ambiente adecuado. Esta situación en la naturaleza suele ocurrir en primavera. A medida que aumenta el tiempo de sol y la inclinación de la tierra, el terreno húmedo comienza a calentarse. Otro dato a tener en cuenta es que germinan en la tierra, protegidos de la luz en áreas ventiladas (aire rico en oxígeno), y una vez que germinan comienzan a echar raíces.

Siempre que se controlen los siguientes parámetros, se pueden utilizar diferentes métodos para simular estas condiciones de germinación en casa:

- **Humedad:** Junto con el calor, activa la semilla y reblandece la cascara que la protege. Una vez germinada, será indispensable para sobrevivir.
- **Oscuridad:** Las raíces son sensibles a la luz, por lo que deberán estar a oscuras si queremos garantizar su germinación.
- **Calor:** Junto con la humedad, activa la semilla indicando condiciones favorables en el exterior. Es importante mantenerlo entre 20 y 26°C.
- **Oxígeno:** es importante abrir el germinador cada día para renovar el aire y proporcionar el oxígeno necesario.



Ilustración 25. Germinación de semillas de cannabis [Autor]

Para la germinación se utilizo un frasco donde se dejaron en agua las semillas durante dos días para luego ser depositadas en servilletas donde culminaran el proceso, es el método más común y de bajo coste, su efectividad es del 100%.



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Al pasar 5 días aproximadamente, la semilla presentara una raíz y será el momento de trasplantarla a su primera matera donde observaremos el crecimiento de la plántula esto tardara más o menos 7 días, se observará como van naciendo las primeras hojas.



*Ilustración 26. Plántulas de cannabis en crecimiento*

En esta etapa necesitara abono y riego constante que ayudara a estimular las raíces y garantizar un crecimiento más fuerte, es una etapa fundamental donde necesita mucho cuidado, solo queda cuidarla para que crezca para ser estudiada.

Otra actividad a desarrollar es la construcción de un pequeño invernadero que nos facilitara tanto para el crecimiento de la planta como para el proceso de detección de los hongos presentes en ella ya que nos ayuda a controlar un parámetro muy importante como es la iluminación, el terreno donde se construyo es de 4 x 1 mts de longitud, perfecto para construcción y adaptación del invernadero



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS



Ilustración 27. Construcción del invernadero en Pamplona [Autor]

Uno de las constantes que se desea controlar y que es muy importante es la iluminación del lugar. En muchos aspectos de nuestra vida diaria, consideraremos este problema: encontrar una casa, un espacio de aprendizaje, tomar fotografías, etc. Una buena iluminación, o una iluminación adecuada, es fundamental para todo.

El principal problema con las aplicaciones de visión artificial en entornos industriales es que las condiciones bajo niveles de luz ambiental nunca son uniformes. En algunos lugares, debido a la iluminación natural o al encender las luces auxiliares durante el día, la luz cambia a lo largo del día. Esto da como resultado cambios mínimos de luz, pero de una forma u otra afectará los procedimientos de nuestra cámara de visión artificial. Para ello, solo hay una solución: aplicar iluminación artificial para que sea constante y uniforme en todo momento del día.

Por eso se tomó la decisión de adecuarle una lampara fluorescentes capaz de poder mantener una iluminación constante facilitando la toma de muestras de las diferentes imágenes necesarias para la detección de los hongos.



Ilustración 28. lampara Fluorescente



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

### 2.3. ADQUISICIÓN DE DATOS

Esta fase es la más importante del proyecto ya que si se tienen una base de datos extensa y que cumpla con las características que se requieren para una buena detección de los hongos, al final se obtendrá resultados favorables. Los datos disponibles para este proyecto, y cuyo procesamiento posteriormente serán tratados en diversas aplicaciones son extraídas principalmente de bases de datos y de páginas web, debido a la pandemia del Covid-19 es la forma más fácil de adquirirlas, en el transcurso de la investigación se tomó la decisión de trabajar con imágenes de otras plantas y características similares ya que no se encontraron un volumen adecuado para que al momento de realizar un entrenamiento se pueda garantizar un resultado eficaz.



*Ilustración 29. Hoja infectada con Mildiu*

Kaggle es una plataforma popular que proporciona un excelente recurso para las personas que desean aprender Machine Learning o incluso ciencia de datos. permite a los usuarios encontrar y publicar conjuntos de datos, cuenta con usuarios de varios dominios como estadistas, científicos de datos y practicantes de aprendizaje automático.



*Ilustración 30. Logo de base de datos Kaggle*

Para el desarrollo de este trabajo se utilizaron 500 imágenes de cada hongo para un total de 1500, las imágenes deben ser con extensión de JPG o PNG para que puedan ser usadas en el entrenamiento.

### 2.4. PRE-PROCESAMIENTO DE IMÁGENES

Esta parte del proyecto está enfocado en resaltar zonas de interés tanto de imágenes digitales como en videos mediante diferentes métodos que nos brinda la librería libre de OpenCv y Python y eliminar fragmentos no deseados como ruidos y brillo.



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Iniciamos importando las librerías necesarias como OpenCv que se utilizan para procesamiento de imágenes y numpy que trae integrada muchas funciones de cálculo matricial de N dimensione, luego se procede a capturar o leer la imagen que deseamos tratar como se muestra en el siguiente código.

Esta parte del proyecto está enfocado en resaltar zonas de interés tanto de imágenes digitales como en videos mediante diferentes métodos que nos brinda la librería libre de OpenCv y Python y eliminar fragmentos no deseados como ruidos y brillo.

Iniciamos importando las librerías necesarias como OpenCv que se utilizan para procesamiento de imágenes y numpy que trae integrada muchas funciones de cálculo matricial de N dimensione, luego se procede a capturar o leer la imagen que deseamos tratar como se muestra en el siguiente código.

```
1 import cv2
2 import numpy as np
3
4
5 imgr= cv2.imread("Mildiu.jpg")
6
```

Ilustración 31. Importación de librerías OpenCv [Autor]

La mayoría de las imágenes son de gran tamaño por eso se opta por redimensionar cada imagen para facilitar el tratamiento y obtener mejores resultados.

```
8 img=cv2.resize(imgr, (250,250))
9 cv2.imshow("Imagen", img)
```

Ilustración 32.Redimensionamiento imagen de entrada

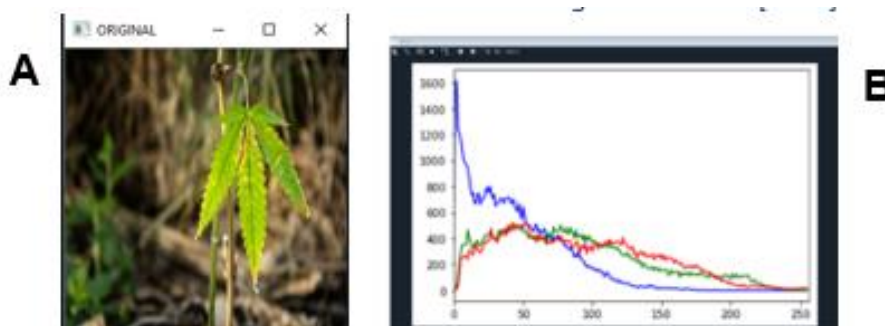


Ilustración 33. (A) Imagen de entrada (B) Histograma RGB

Uno de los métodos más utilizados para facilitar el procesamiento de la imagen es convertir la imagen RGB a escala de gris.



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

```
11 Gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
12 cv2.imshow('Gris', Gray)
```

Ilustración 34. Conversión BGR-GRAY imagen [Autor]

La imagen en escala de gris consta de cada componente multiplicado por 3 constantes Definición: Alfa, Beta y Gamma. Este proceso le resta toda la información El color contenido en cada píxel.

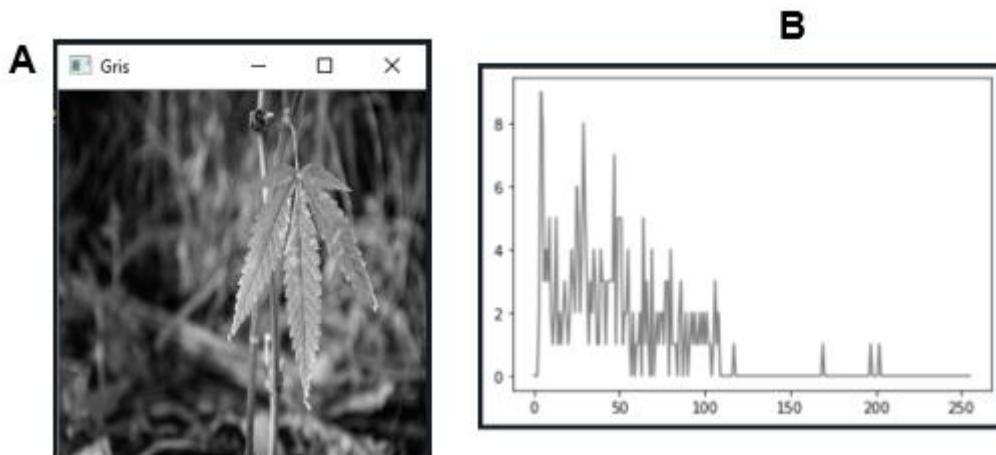


Ilustración 35. (A) Imagen escala de gris (B) Histograma color gris

Como podemos observar en las ilustraciones ( ) pasamos de una imagen de 3 canales (B, G, R) a una imagen de 1 solo canal con intensidades de 0 a 255 donde los valores cercanos a cero son más oscuros llegando al color negro y los más cercanos a 255 son más claros llegando al color blanco, facilitando así el procesamiento y ocupando menos espacio.

Otro proceso importante que utilizamos es la binarización donde pasamos de tener una matriz con píxeles de 0 a 255 a una matriz con dos valores 0 y 255 creando una imagen de dos únicos colores, lo importante es poder detectar los rangos que necesita la imagen para la correcta binarización.

```
28 th = 150
29 max_val = 255
30 _, binario = cv2.threshold(Gray, th, max_val, cv2.THRESH_BINARY)
```

Ilustración 36. Código de binarización [Autor]

La función threshold se utiliza para aplicar el umbral, el primer argumento (Gray) es la imagen de origen, que debe ser una imagen en escala de gris, el segundo argumento (th) Es el valor del umbral que se utiliza para clasificar los valores de los píxeles. El tercer argumento (max\_val) es el valor máximo que se asigna a los valores de píxeles que superan el umbral.



Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval

## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Analizando la ilustración () nos damos cuenta que nuestra zona de interés es de 0 a 150 el cual es el valor máximo para el umbral, los valores mayores se igualarán a 0, mostrando la siguiente imagen.

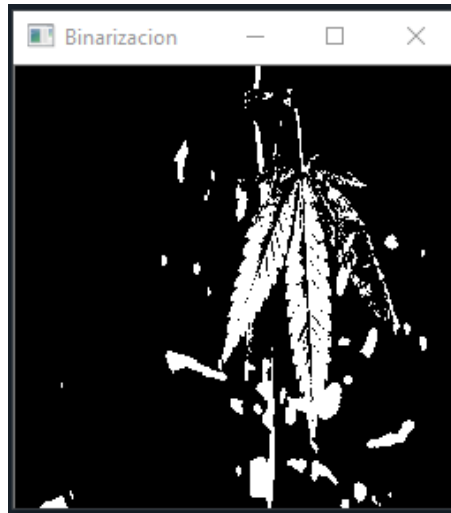


Ilustración 37. Imagen binarizada

Luego de que tenemos nuestra imagen binarizada procedemos a extraer solo nuestra zona de interés que en este caso son las hojas de la planta para eso utilizamos la función de detección de contornos (`cv2.findContours`). el primer argumento es la imagen de origen (Gray), el segundo es el modo de recuperación de contorno, el tercer argumento es el método de aproximación de contorno.

```
33  _,contornos,_ = cv2.findContours(binario,cv2.RETR_EXTERNAL,  
34  cv2.CHAIN_APPROX_SIMPLE)
```

Ilustración 38. Función de detección de contornos [Autor]

Cada contorno individual es una matriz Numpy de coordenadas (x, y) de los puntos fronterizos del objeto. Notamos en la imagen binarizada que hay trozos blancos que no son necesarios y deseamos eliminar para ello usamos una función que se encarga de filtrar la imagen por áreas, solo los contornos con áreas mayores a un umbral establecido quedaran en la imagen.

```
36  for c in contornos:  
37      area = cv2.contourArea(c)  
38      if area > 900:  
39          font = cv2.FONT_HERSHEY_SIMPLEX  
40          nuevoContorno = cv2.convexHull(c)  
41          cv2.drawContours(img, [nuevoContorno], 0, (0,255,100), 3)
```

Ilustración 39. Dibujar Contorno



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

La función `convexHull` sigue todos los puntos extremos de un contorno y crea una envolvente a su alrededor.

En este caso el umbral para detectar el área de nuestro interés es de 900, toda área menor a este valor se eliminará mostrándonos como resultado la siguiente imagen:

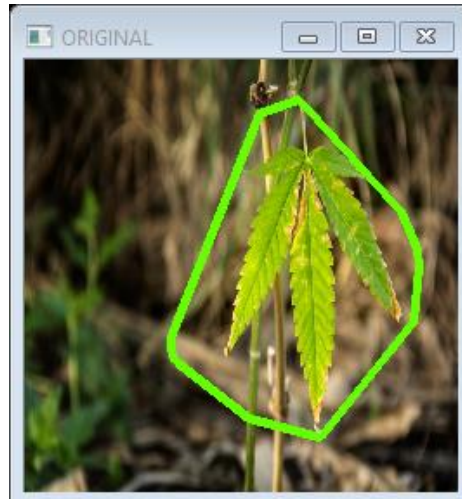


Ilustración 40. Imagen con contorno dibujado

Como la finalidad de esta fase de la metodología es poder detectar de una imagen la zona más adecuada eliminando ruido o zonas que no son importantes, en este caso solo nos interesa la planta, la siguiente actividad es recortar el entorno que se detectó y esa nueva imagen la guardaremos en una carpeta que luego utilizaremos en otra fase.

```
6  ▼ for c in contornos:
7      area = cv2.contourArea(c)
8      ▼ if area > 900:
9          font = cv2.FONT_HERSHEY_SIMPLEX
10         nuevoContorno = cv2.convexHull(c)
11         xymax = np.max(nuevoContorno, axis=0, out=None, initial=0)
12         |
13         xmax=xymax[0,0]
14         ymax=xymax[0,1]
15
16         xymin = np.min(nuevoContorno, axis=0, out=None, initial=255)
17
18         xmin=xymin[0,0]
19         ymin=xymin[0,1]
20
21         crop_img = img[ymin:ymax, xmin:xmax]
22         cv2.imshow('Recorte', crop_img)
23         cv2.drawContours(img, [nuevoContorno], 0, (0,255,100), 3)
```

Ilustración 41. Función para recortar contorno



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

En esta oportunidad utilizaremos dos funciones (np.max y np.min) que nos ayudaran a hallar los puntos maximos y minimos del contorno que nos servira para hacer un recorte y obtener una nueva imagen.

```
cv2.imwrite("images/img" + str(img_index)+'.png',crop_img)  
img_index = img_index +1
```

Ilustración 42. Guardar imágenes en Carpetas

Esta línea de código nos permite guardar las nuevas imágenes en una carpeta llamada “images” con la extensión .png



Ilustración 43. imagen recortada

Esta es la parte final del procesamiento de imágenes, con esta fase lograremos crear carpetas que nos ayudaran al entrenamiento.

### 2.5. RECONOCIMIENTO DE PATRONES

Si examinamos detalladamente las imágenes de los 3 hongos Mildiu, Oídio y roya notamos que una de sus mayores características que los diferencian es el color, Mildiu es de color café el Oídio es de color blanco y la roya es de color amarillo por eso hemos tomado como patrón principal el color, otros aspectos que se pueden analizar es la detección de formas basadas en los contornos o la detección de esquinas, orillas, haar (volumen invariante) etc. pero no sería tan relevantes.

Para esto usaremos la segmentación de imágenes según el modo de color que esto incluye la comparación de diferentes grupos de píxeles que componen las imágenes con una serie de patrones previamente entrenados.

Teniendo las imágenes de la fase anterior “Procesamiento de imágenes” procedemos a hacer el análisis para la segmentación. Para esto tenemos que tener en claro varios temas como los diferentes modelos de color RGB (Red, Green, Blue) y HSV (Hue, Saturation, value) que nos sirve para establecer los umbrales necesarios para detectar los colores.



# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

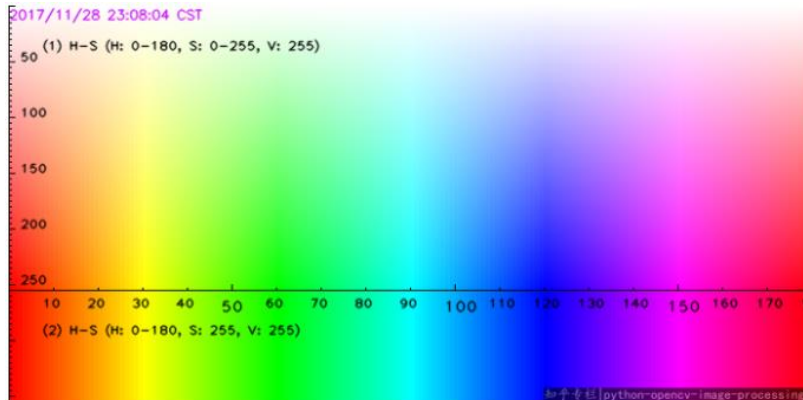


Ilustración 44. Rango de colores HSV

Para ello, necesitamos definir una máscara que se usará como filtro para los colores restantes que no queremos detectar. Necesitamos el valor de cambio de cada canal y cubrimos un cierto rango de espacio de color HSV. Es decir, por ejemplo, H varía de 0 a 10, S varía de 100 a 255 y V varía de 100 a 255 para detectar rojo / naranja.

En este caso hallamos el color amarillo y el blanco representando la Roya y Oídio.

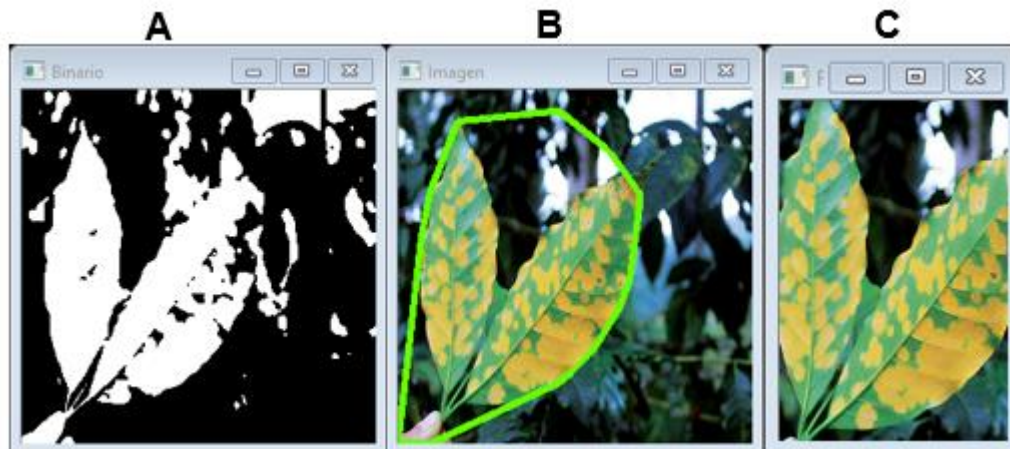


Ilustración 45.(A) imagen binarizada (B) Imagen con contorno (C) Imagen recortada[propio] [Autor]

En la ilustración () nos muestra el procesamiento de la imagen para extraer la zona de más interés.

De acuerdo con la ilustración () los rangos para cada color son:



```
verde_bajos = np.array([35,100,100],np.uint8)
verde_altos = np.array([70 ,255,255],np.uint8)

Mildiu_bajos = np.array([10,100,100],np.uint8)
Mildiu_altos = np.array([15 ,255,255],np.uint8)

Roya_bajos = np.array([20,100,100],np.uint8)
Roya_altos = np.array([35 ,255,255],np.uint8)
```

*Ilustración 46. Rango de colores HSV [Autor]*

Como sabemos OpenCv por defecto representa las imágenes en BGR y para la segmentación esa imagen debe pasar a rangos de HSV con la siguiente línea.

```
HSVimg = cv2.cvtColor(crop_img, cv2.COLOR_BGR2HSV)
cv2.imshow("Imagen_HSV", HSVimg)
```

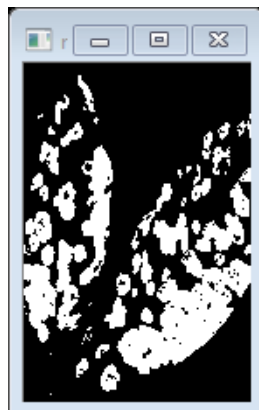
*Ilustración 47. Conversión BGR-HSV [Autor]*

y la conversión de BGR-HSV como lo mencionaba anteriormente. Luego se procede a definir una máscara que funcionara como filtro del resto de los colores que no queremos detectar.

```
mask = cv2.inRange(HSVimg, Roya_bajos, Roya_altos)
maskHoja = cv2.bitwise_and(crop_img, crop_img, mask=mask )
```

*Ilustración 48. Mascara de Mildiu [Autor]*

la función inRange utiliza 3 parámetros, la imagen y dos umbrales, esto devuelve una máscara con valores 1 donde detecto el color elegido y 0 donde no hay presencia de este como se muestra en la siguiente ilustración.



*Ilustración 49. Mascara binarizada Mildiu [Autor]*



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

La imagen original es filtrada con la función `bitwise_and`, esta realiza una operación AND con los valores originales de la imagen. Se sabe que las imágenes están representadas en arreglos numpu, la función devuelve la imagen original filtrada con la máscara anteriormente establecida.

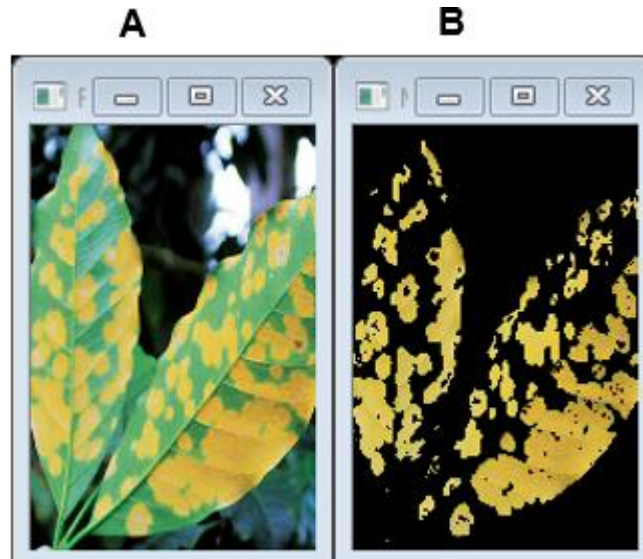


Ilustración 50.(A)imagen color (B)mascara de mildiu [Autor]

De esta manera se puede visualizar cualquier otro color que nos indica la presencia de del hongo

### 2.6. ENTRENAMIENTO YOLOV3 PARA DETECCIÓN DE HONGOS

Para realizar un buen entrenamiento se deben cumplir una serie de pasos que iniciara con la recolección de imágenes que se usaran tanto para entrenar como para validar, pasando con la creación de un modelo hasta llegar a la última fase que es la detección de los hongos

#### 2.6.1 CREACIÓN DE LA DATA PARA ENTRENAMIENTO Y VALIDACIÓN

Para creación de la data necesitamos gran cantidad de imágenes, esta recolección es muy importante para el desarrollo de nuestro sistema, ya que gracias a esto dependerá el éxito de nuestro proyecto.

Como sabemos la planta de cannabis no es tan comercial al contrario es una planta que en la mayoría de lugares es ilegal y que hasta ahora se está



*Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval*

## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

convirtiéndola en una alternativa medicinal, imágenes de esta planta son muy escasas en la web, no se encontraron archivos de las hojas enfermas o con presencia de hongos suficientes para crear una data lo suficientemente grande para entrenar nuestro modelo, entonces se tomó la decisión de adquirir imágenes de otras plantas como la hoja de café que contaban con la presencia de dichos hongos, su similitud con la hoja de cannabis y los hongos es muy grande y fue así como creamos nuestra data, con imágenes de cannabis mezcladas con otras plantas muy parecidas.

### 2.6.2 ETIQUETADO DE IMÁGENES

No es suficiente tener una gran cantidad de imágenes, también se necesita saber en qué parte de la imagen se encuentra nuestro objeto a detectar y para eso utilizamos un software externo usamos uno de los más populares "labelimg", que se encargará de etiquetar las imágenes encerrando en un rectángulo nuestra zona de interés para luego crear un documento con extensión .XML que contiene las coordenadas y su respectiva etiqueta como se muestra en la imagen.

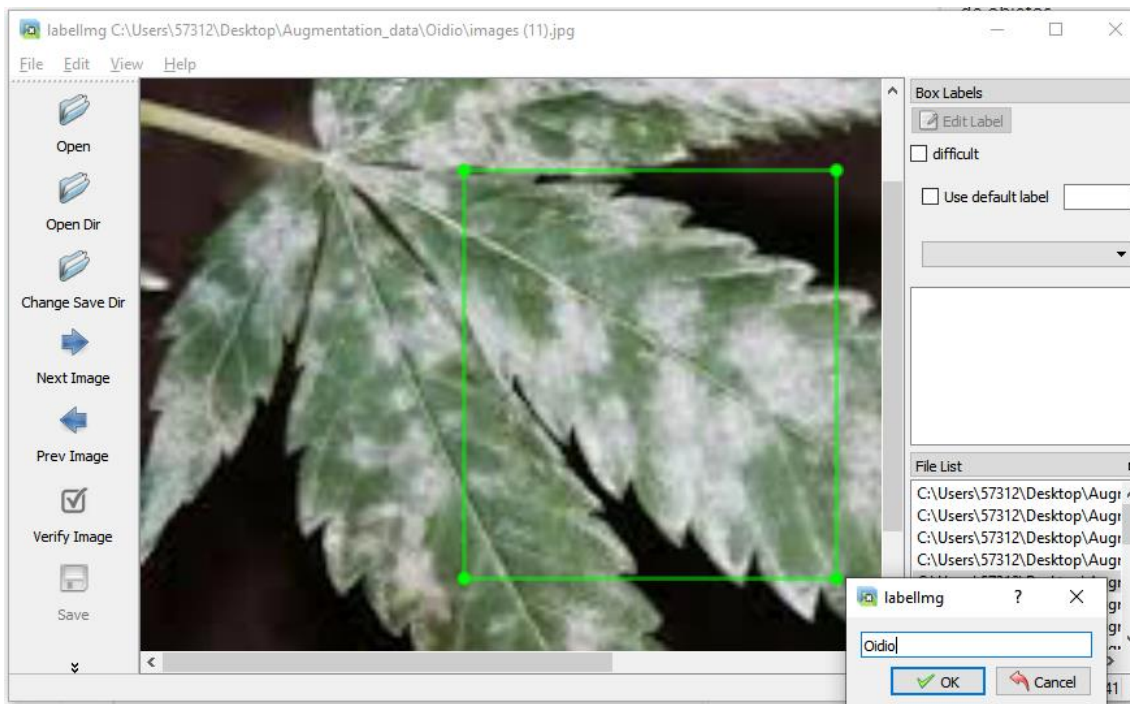


Ilustración 51. Entorno de etiquetado Labelimg



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

```
<annotation>
  <folder>imagenes</folder>
  <filename>Oidio_0_5211.png</filename>
  <path>C:\Users\57312\Desktop\TensorFlow-2.x-YOLOv3-master\imagenes\Oidio_0_5211.png</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>128</width>
    <height>128</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Oidio</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>72</xmin>
      <ymin>14</ymin>
      <xmax>89</xmax>
      <ymax>24</ymax>
    </bndbox>
  </object>
  <object>
    <name>Oidio</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
```

Ilustración 52. Documento con extensión XML

## 2.6.3 CONVERSIÓN DE XML A YOLO V3

hora debemos convertir las etiquetas .XML al formato de entrenamiento de Yolo apropiado. Esta conversión convertirá la carpeta de entrenamiento y la carpeta de validación en dos archivos con extensión .txt estos archivos son utilizados para el entrenamiento del modelo.

```
import xml.etree.ElementTree as ET
import os
import glob

foldername = os.path.basename(os.getcwd())
if foldername == "tools": os.chdir("..")

data_dir = '/custom_dataset/'
Dataset_names_path = "model_data/fungus_names.txt"
Dataset_train = "model_data/fungus_train.txt"
Dataset_test = "model_data/fungus_test.txt"
is_subfolder = False
```

Ilustración 53. Código conversor XML-TXT



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

- data\_dir es la ubicación de nuestra carpeta de imágenes, la configuramos en / custom\_dataset / ;
- Dataset\_names\_path es una ubicación donde se guardará la ruta de clases generada, la configuramos en model\_data / fungis\_names.txt ;
- Dataset\_train es una ubicación donde se guardará la ruta de anotación del tren generada, la configuramos en model\_data / Dataset\_train.txt ;
- Dataset\_test es una ubicación donde se guardará la ruta de anotación de prueba generada, la configuramos en model\_data / Dataset\_test.txt ;
- la última línea es is\_subfolder , esto se cambia cuando no tenemos subcarpetas en las rutas de tren o de prueba. En un tutorial anterior, usábamos subcarpetas al descargar el conjunto de datos personalizado, ahora no tenemos subcarpetas, así que lo configuramos en is\_subfolder = False.

### 2.6.4 ENTRENAMIENTO MODELO DETECCIÓN DE HONGOS

Antes de empezar con la explicación de como entrenar, se tiene que saber con claridad que es yoloV3, se debe saber que es un algoritmo que utiliza redes neuronales convolucionales para la detección de objetos, es un algoritmo bastante rápido pero muy poco preciso, comparándolo con algoritmos de reconocimiento YoloV3 predice las etiquetas y además muestra la ubicación del objeto en este caso el Hongo.

Este algoritmo utiliza un extractor de características llamado Darknet 53, que como su nombre lo indica contiene 53 capas de convolución cada una seguida por una capa de normalización por lotes y la activación de Leaky ReLU, no utiliza función de agrupación.

	Type	Filters	Size	Output
	Convolutional	32	3 x 3	256 x 256
	Convolutional	64	3 x 3 / 2	128 x 128
1x	Convolutional	32	1 x 1	
	Convolutional	64	3 x 3	
	Residual			128 x 128
	Convolutional	128	3 x 3 / 2	64 x 64
2x	Convolutional	64	1 x 1	
	Convolutional	128	3 x 3	
	Residual			64 x 64
8x	Convolutional	256	3 x 3 / 2	32 x 32
	Convolutional	128	1 x 1	
	Convolutional	256	3 x 3	
8x	Residual			32 x 32
	Convolutional	512	3 x 3 / 2	16 x 16
	Convolutional	256	1 x 1	
4x	Convolutional	512	3 x 3	
	Residual			16 x 16
	Convolutional	1024	3 x 3 / 2	8 x 8
	Convolutional	512	1 x 1	
	Convolutional	1024	3 x 3	
	Residual			8 x 8
	Avgpool		Global	
	Connected		1000	

Ilustración 54 Arquitectura Darknet53



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

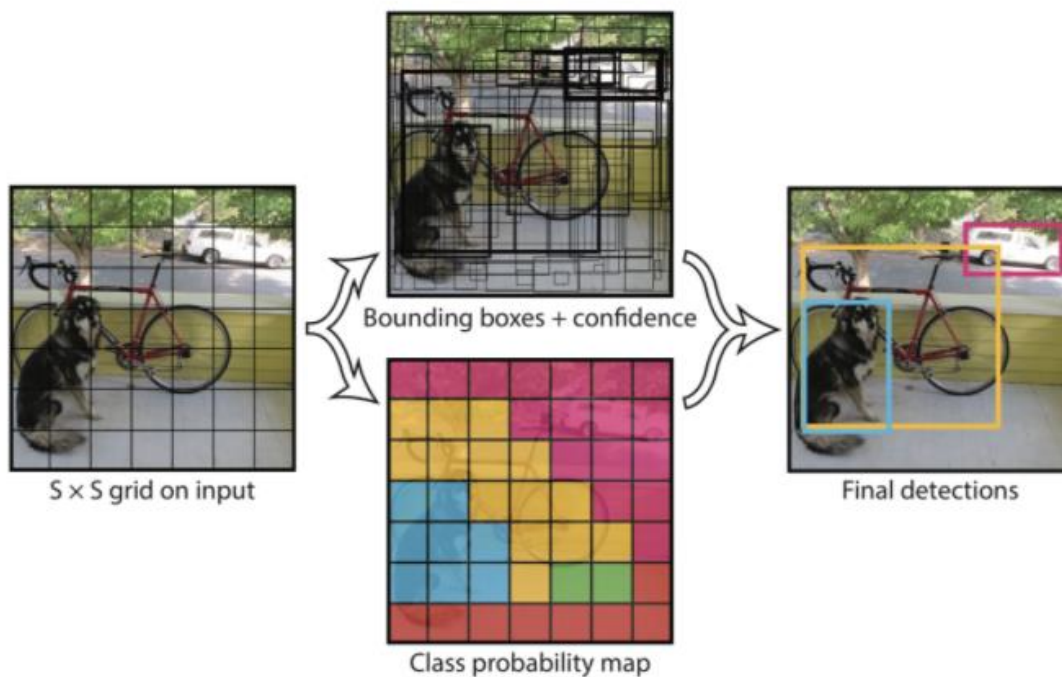


## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Una vez las imágenes han sido previamente etiquetadas con la herramienta labeling, misma que se localiza en la raíz de Darknet. Previo a que las imágenes hayan sido movidas, se debe crear 2 archivos, uno que contendrá el nombre del objeto a reconocer llamado fungi.names y el otro con las ubicaciones de donde se guardaran pesos y las ubicaciones de las imágenes para crear el set de entrenamiento llamado object.data. Una vez que se tiene todos estos pasos completos se debe descargar un archivo de la página oficial de Darknet YOLOv3 con los pesos para del modelo de machine Learning. El archivo contiene los pesos predeterminados antes de empezar a entrenar.

La salida del algoritmo YoloV3 genera cuadros delimitadores cada cuadro predicho muestra una puntuación de confianza, toda casilla inferior a al umbral de confianza más alto será descartado para su procesamiento.

YoloV3 divide la imagen en cuadrículas  $s \times s$ , si el centro de un objetivo cae en una cuadrícula, la cuadrícula es responsable de detectar el objetivo



El cuadro delimitador contiene 4 valores:  $x, y, w, h$ ,  $(x, y)$  representa el centro del cuadro.  $(W, h)$  representa el ancho y la altura de la caja, La confianza indica la probabilidad de contener objetos en este cuadro de predicción, que en realidad es el valor de IoU entre el cuadro de predicción y el cuadro real, La probabilidad de clase indica la probabilidad de clase del objeto y YOLOv3 usa un método de dos clases.



# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

```
def darknet53 ( input_data ):  
    input_data = convolutional ( input_data , ( 3 , 3 , 3 , 32 ) )  
    input_data = convolutional ( input_data , ( 3 , 3 , 32 , 64 ) , downsample = True )  
  
    para i en el rango ( 1 ) :  
        input_data = residual_block ( input_data , 64 , 32 , 64 )  
  
    input_data = convolutional ( input_data , ( 3 , 3 , 64 , 128 ) , downsample = True )  
  
    para i en el rango ( 2 ) :  
        input_data = residual_block ( input_data , 128 , 64 , 128 )  
  
    input_data = convolutional ( input_data , ( 3 , 3 , 128 , 256 ) , downsample = True )  
  
    para i en el rango ( 8 ) :  
        input_data = residual_block ( input_data , 256 , 128 , 256 )  
  
    route_1 = input_data  
    input_data = convolutional ( input_data , ( 3 , 3 , 256 , 512 ) , downsample = True )  
  
    para i en el rango ( 8 ) :  
        input_data = residual_block ( input_data , 512 , 256 , 512 )  
  
    route_2 = input_data  
    input_data = convolutional ( input_data , ( 3 , 3 , 512 , 1024 ) , downsample = True )  
  
    para i en el rango ( 4 ) :  
        input_data = residual_block ( input_data , 1024 , 512 , 1024 )  
  
    return route_1 , route_2 , input_data
```

Para entender mejor la arquitectura de Darknet-53 usamos la siguiente imagen:

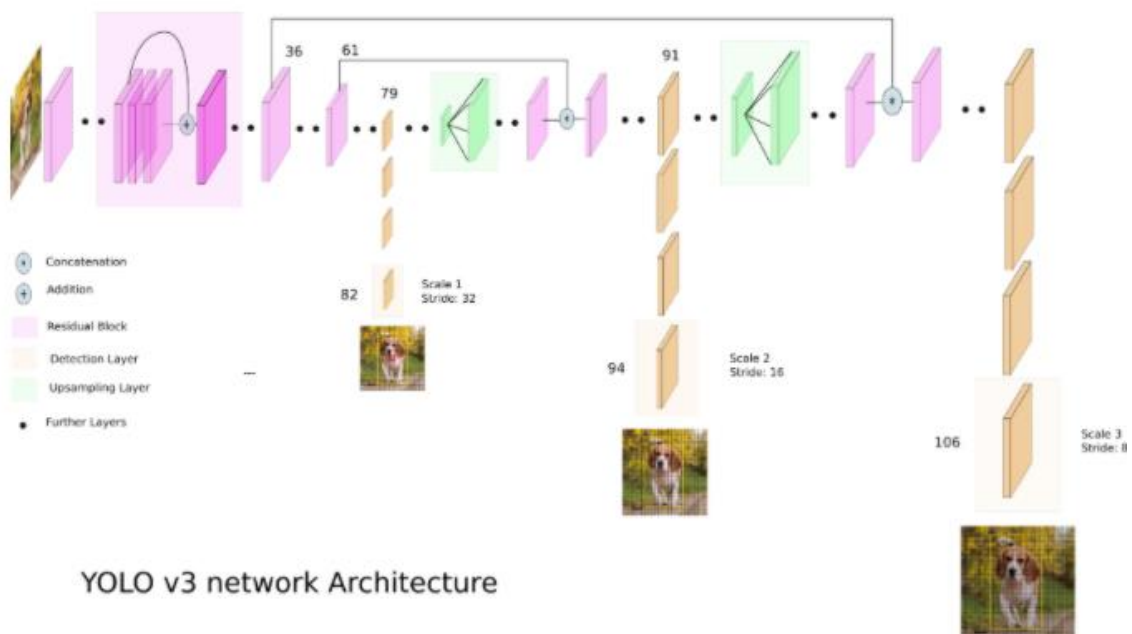


Ilustración 55. Arquitectura Yolo V3



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

En la imagen anterior se puede observar que yolo realiza la detección en 3 escalas diferentes para modificar en diferentes tamaños las imágenes mediante pasos de 32,16 y 8, eso quiere decir que, si el tamaño de la imagen es de 416\*416, el algoritmo realizara la detección en la escala de 13\*13, 26\*26 y 52\*52

Para la primera escala YoloV3 reduce la resolución de la imagen de entrada 13\*13 y hace una predicción en la capa 82, la primera escala de detección produce un tensor 3D de tamaño 13\*13\*255

Después el algoritmo toma el mapa de características de la capa 79 aplica una capa convolucional antes de realizar un muestreo superior en un factor de 2 para tener un tamaño de 26 x 26. Este mapa de características muestreado se concatena con el mapa de características de la capa 61. El mapa concatenado Luego, el mapa de características se somete a unas pocas capas convolucionales más hasta que se realiza la segunda escala de detección en la capa 94. La segunda escala de predicción produce un tensor 3D de tamaño 26 x 26 x 255.

El mismo diseño se vuelve a realizar una vez más para predecir la 3ª escala. Al mapa de características de la capa 91 se le agrega una capa convolucional y luego se concatena con un mapa de características de la capa 36. La capa de predicción final se realiza en la capa 106, lo que produce un tensor 3D de tamaño 52 x 52 x 255. En resumen, Yolo predice la detección de más de 3 escalas diferentes, por lo que si alimentamos una imagen de tamaño 416x416, produce 3 tensor de forma de salida diferente, 13 x 13 x 255, 26 x 26 x 255 y 52 x 52 x 255.

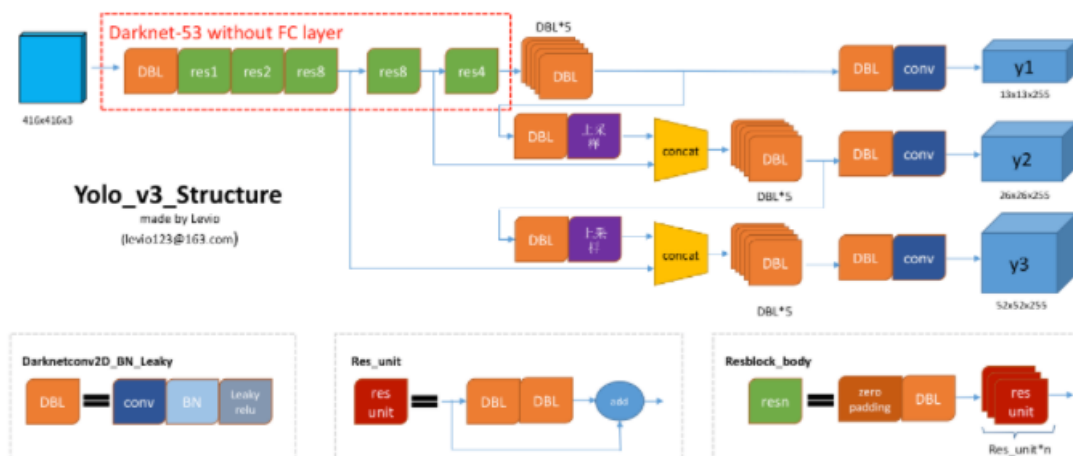


Ilustración 56. Estructura YoloV3



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

```
def YOLOv3(input_layer, NUM_CLASS):
    # After the input layer enters the Darknet-53 network, we get three branches
    route_1, route_2, conv = darknet53(input_layer)
    # See the orange module (DBL) in the figure above, a total of 5 Subconvolution operation
    conv = convolutional(conv, (1, 1, 1024, 512))
    conv = convolutional(conv, (3, 3, 512, 1024))
    conv = convolutional(conv, (1, 1, 1024, 512))
    conv = convolutional(conv, (3, 3, 512, 1024))
    conv = convolutional(conv, (1, 1, 1024, 512))
    conv_lobj_branch = convolutional(conv, (3, 3, 512, 1024))

    # conv_lbbox is used to predict large-sized objects , Shape = [None, 13, 13, 255]
    conv_lbbox = convolutional(conv_lobj_branch, (1, 1, 1024, 3*(NUM_CLASS + 5)), activate=False, bn=False)

    conv = convolutional(conv, (1, 1, 512, 256))
    # upsample here uses the nearest neighbor interpolation method, which has the advantage that the
    # upsampling process does not need to learn, thereby reducing the network parameter
    conv = upsample(conv)

    conv = tf.concat([conv, route_2], axis=-1)
    conv = convolutional(conv, (1, 1, 768, 256))
    conv = convolutional(conv, (3, 3, 256, 512))
    conv = convolutional(conv, (1, 1, 512, 256))
    conv = convolutional(conv, (3, 3, 256, 512))
    conv = convolutional(conv, (1, 1, 512, 256))
    conv_mobj_branch = convolutional(conv, (3, 3, 256, 512))

    # conv_mbbox is used to predict medium-sized objects, shape = [None, 26, 26, 255]
    conv_mbbox = convolutional(conv_mobj_branch, (1, 1, 512, 3*(NUM_CLASS + 5)), activate=False, bn=False)

    conv = convolutional(conv, (1, 1, 256, 128))
    conv = upsample(conv)

    conv = tf.concat([conv, route_1], axis=-1)
    conv = convolutional(conv, (1, 1, 384, 128))
    conv = convolutional(conv, (3, 3, 128, 256))
    conv = convolutional(conv, (1, 1, 256, 128))
    conv = convolutional(conv, (3, 3, 128, 256))
    conv = convolutional(conv, (1, 1, 256, 128))
    conv_sobj_branch = convolutional(conv, (3, 3, 128, 256))

    # conv_sbbox is used to predict small size objects, shape = [None, 52, 52, 255]
    conv_sbbox = convolutional(conv_sobj_branch, (1, 1, 256, 3*(NUM_CLASS + 5)), activate=False, bn=False)

    return [conv_sbbox, conv_mbbox, conv_lbbox]
```

Ilustración 57. función YoloV3

## Residual Modul

El modulo residual es el uso del mecanismo de atajo que alivia el problema de desaparición del gradiente causado al aumentar la profundidad en la red neuronal, para optimizar utiliza el mapeo de identidad para establecer un canal de correlación directa entre la entrada y la salida y facilitar que la red se concentre en aprender el residuo entre a identidad y la salida.

```
def residual_block ( input_layer , input_channel , filter_num1 , filter_num2 ):
    short_cut = input_layer
    conv = convolutional ( input_layer , filter_shape = ( 1 , 1 , input_channel , filter_num1 )
    conv = convolutional ( conv , filter_shape = ( 3 , 3 , filter_num1 , filter_num2 )

    residual_output = short_cut + conv
    return residual_output
```

Ilustración 58. Función Modulo Residual



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

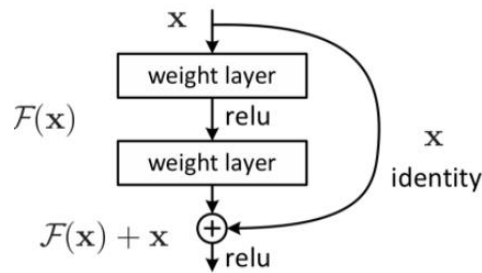


Ilustración 59. Módulo residual

### Extraer características

Para conocer el proceso de predicción de YOLO en detalle, es necesario comprender primero qué son los mapas de características y las incrustaciones.

- **Mapa de caracterización**

Cuando hablamos de las redes de CNN, siempre escuchamos mapas de características. También se llama mapeo de características. En resumen, la imagen de entrada se convoluciona con el núcleo de convolución para obtener las características de la imagen.

En términos generales, cuando la red CNN extrae características de la parte inferior de la imagen hacia arriba, aumentará la cantidad de mapas de características (en realidad, la cantidad de núcleos de convolución), la información espacial disminuirá y sus características también aumentarán. Más abstracto. Por ejemplo, la famosa red VGG16, su diagrama característico es el siguiente:

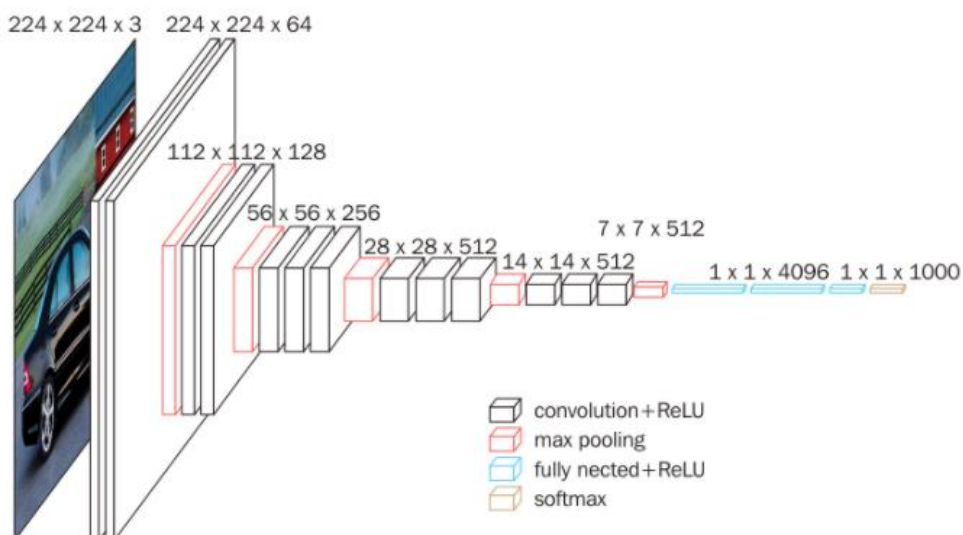


Ilustración 60. Extracción de características



- **Vector de caracterización**

Cuando se trata de mapas de características, a menudo podemos escuchar los mapas de características mencionados en el campo del reconocimiento facial. En términos generales, en realidad es un mapa de características extraído de la última capa completamente conectada en el vector de características.



Ilustración 61. Vector de características

Cuando la red de CNN extrae características de la imagen de abajo hacia arriba, el tamaño espacial del mapa de características generado generalmente se volverá cada vez más pequeño, y el tamaño del canal será cada vez más profundo:

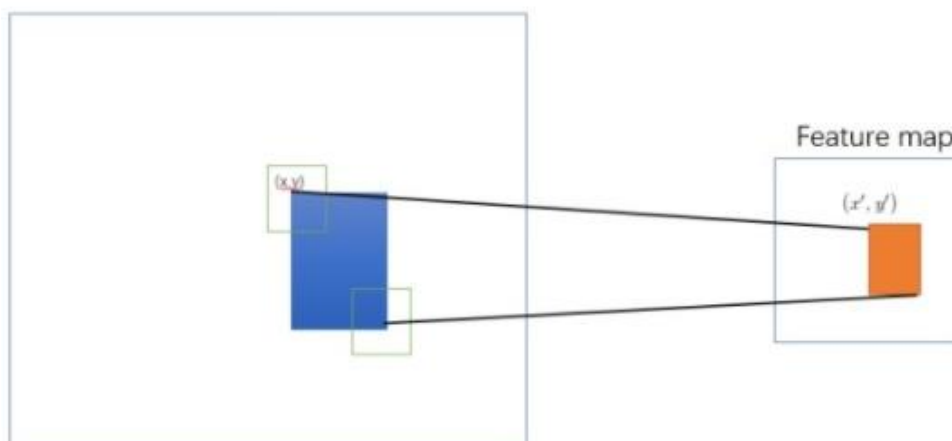


Ilustración 62. Región de interés

Esto está relacionado con el mapeo del ROI (región de interés) al mapa de funciones. En la figura anterior: después de que el ROI se asigna a la imagen original en el espacio de la red CNN, el tamaño del espacio en el mapa de



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

características se reducirá, incluso un punto, pero los puntos de información del canal serán muy ricos. Esta información representa las características de la información de la imagen en el área de ROI mapeada en la red CNN. Debido a que los píxeles de la imagen están estrechamente conectados en el espacio, provocará mucha redundancia espacial. Por tanto, normalmente eliminamos esta redundancia reduciendo el tamaño del espacio y aumentando el tamaño del canal, e intentamos obtener sus características más básicas en el tamaño más pequeño:

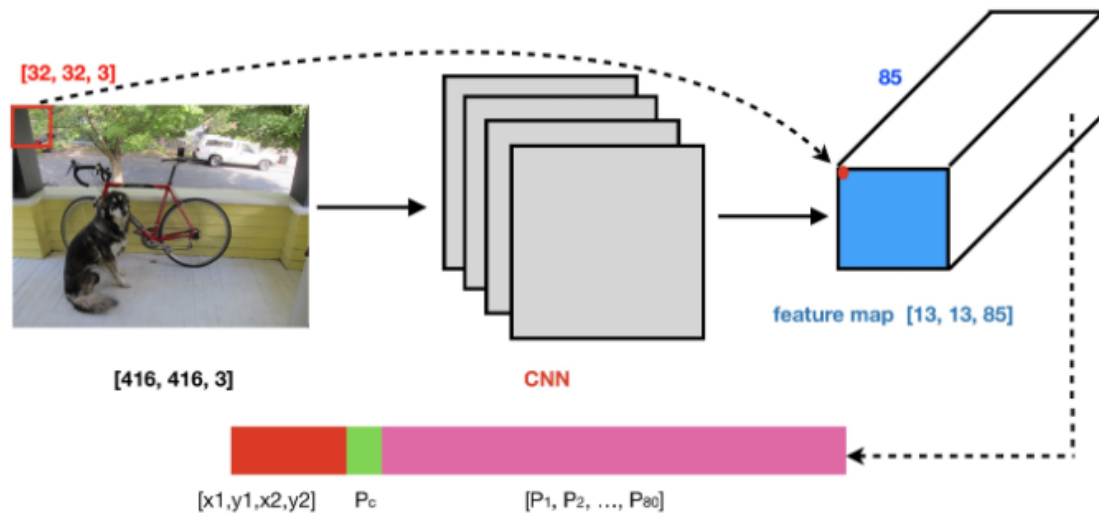


Ilustración 63. Área de ROI mapeada en la red CNN

Por ejemplo, CNN asigna el ROI rojo a la esquina superior izquierda de la imagen original y solo obtiene un punto en el espacio de mapeo de características, pero este punto tiene 85 canales. Por lo tanto, el tamaño del ROI se ha cambiado del tamaño original [32,32,3] al tamaño actual de 85.

De hecho, este es el vector de características de 85 dimensiones obtenido después de que la red CNN realiza la extracción de características en el ROI. Las primeras cuatro dimensiones del vector de características representan información del marco candidato, la dimensión promedio representa la probabilidad de la existencia o no del objeto de juicio y las siguientes 80 dimensiones representan la información de probabilidad de clasificación de 80 categorías.

### Detección de YoloV3

YOLO divide las mallas gruesa, media y fina de la imagen de entrada para predecir objetos grandes, medianos y pequeños, respectivamente. Si el tamaño de la imagen de entrada es 416X416, los tamaños de cuadrícula gruesa, media y fina son 13x13, 26x26 y 52x52, respectivamente. De esta forma, se escala 32 veces, 16 veces y 8 veces en longitud y ancho respectivamente:



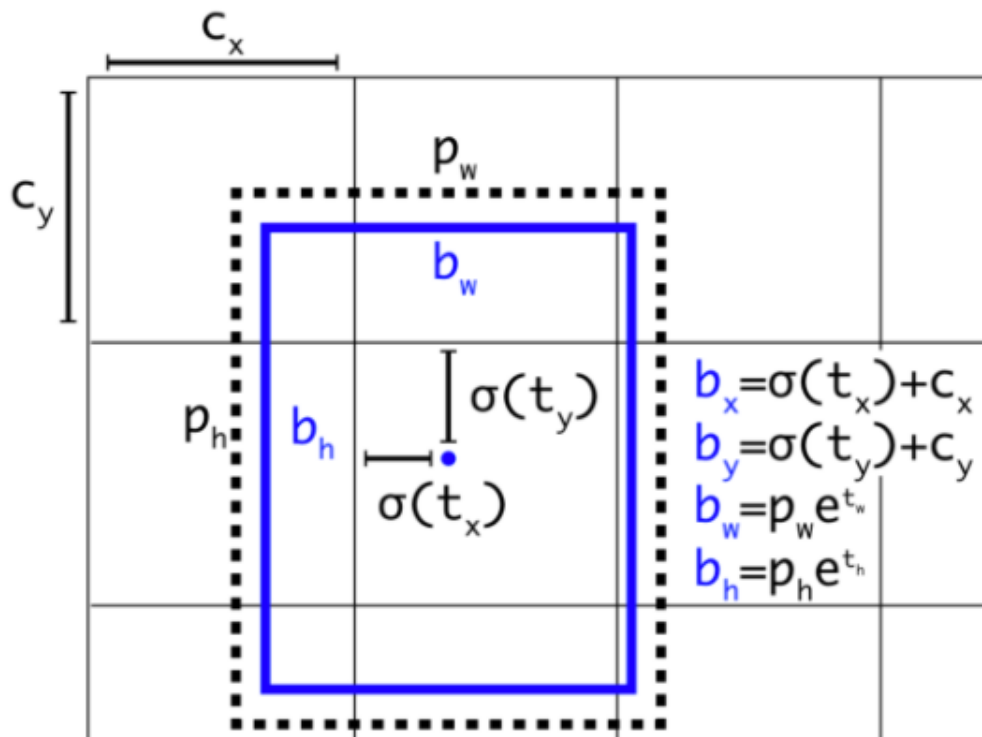
Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval



*Ilustración 64. Imágenes con mallas de identificación*

### Dimensiones de la caja delimitadora

La salida de las tres ramas de la red YOLOv3 se enviará a la función de decodificación para decodificar la información del canal del mapa de funciones. En la siguiente figura: el cuadro de trazos negros representa el cuadro anterior (ancla) y el cuadro azul representa el cuadro de predicción. Al aplicar una transformación de espacio logarítmico a la salida y luego multiplicar por el punto de anclaje, se puede predecir el tamaño del cuadro delimitador:



*Ilustración 65. Dimensionamiento de caja de predicción*





## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

- **b** denota la longitud y el ancho del marco de predicción respectivamente, y **P** denota la longitud y el ancho del marco a priori respectivamente.
- **t** representa el desplazamiento del centro del objeto desde la esquina superior izquierda de la cuadrícula y **C** representa las coordenadas de la esquina superior izquierda de la cuadrícula.

```
def decode ( conv_output , NUM_CLASS , i = 0 ): # donde i = 0, 1 o 2 para corresponder a las tres escalas de cua
conv_shape      = tf . shape ( conv_output )
batch_size      = conv_shape [ 0 ]
output_size     = conv_shape [ 1 ]
conv_output = tf . remodelar ( conv_output , ( batch_size , output_size

, output_size , 3 , 5 + NUM_CLASS ))
conv_raw_dx dy = conv_output [ :, :, :, :, 0 : 2 ] # desplazamiento de la posición central
conv_raw_dwdh = conv_output [ :, :, :, :, 2 : 4 ] # Cuadro de predicción offset de longitud y ancho
conv_raw_conf = conv_output [ :, :, :, :, 4 : 5 ] # confianza del cuadro de predicción

conv_raw_prob = conv_output [ :, :, :, :, 5 : ] # categoría probabilidad del cuadro de predicción # siguiente
y = tf . rango ( tamaño_salida , dtype = tf . int32 )
y = tf . expand_dims ( y , - 1 )
y = tf . mosaico ( y , [

1 , tamaño_salida ])
x = tf . rango ( tamaño_salida , dtype = tf . int32 )
x = tf . expand_dims ( x , 0 )
x = tf . mosaico ( x , [ tamaño_de_salida , 1 ])
cuadrícula_xy = tf . concat ([ x [ :, :, tf . newaxis
], y [ :, :, tf . eje nuevo ]], eje = - 1 )
cuadrícula_xy = tf . azulejo ( xy_grid [ tf . newaxis , :, :, tf . newaxis , :], [ batch_size , 1 , 1 , 3 ,
xy_grid = tf . emitir ( xy_grid , tf . float32 )

# Calcular la posición del centro de la caja de predicción:
pred_xy = ( tf . Sigmoide ( conv_raw_dx dy ) + xy_grid ) * zancadas [ i ] # Calcular la longitud y anchura de
pred_wh = ( tf . Exp ( conv_raw_dwdh ) * ANCLAJES [ i ]) * ZANCAS [ i ]
pred_xywh = tf . concat ([ pred_xy

, pred_wh ], eje = - 1 )
pred_conf = tf . sigmoide ( conv_raw_conf ) caja # objeto calcula la confianza predicho
pred_prob = tf . sigmoide ( conv_raw_prob ) # cálculo de la categoría del compartimento probabilidad objeto
```

Ilustración 66. Función limitadora de objetos

### PROCESAMIENTO NMS

Como sugiere el nombre, la supresión no máxima puede suprimir los términos de supresión no máxima. NMS eliminará los cuadros delimitadores con alta superposición y puntuaciones bajas. El algoritmo NMS es muy simple y el proceso de iteración es el siguiente:

Proceso 1: Determine si el número de cuadros delimitadores es mayor que 0, si no, finalice la iteración;

Proceso 2: seleccione el cuadro delimitador A con la puntuación más alta de acuerdo con el orden de puntuación y elimínelo;



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Proceso 3: Calcule el IoU de este cuadro delimitador A y todos los cuadros delimitadores restantes y elimine los cuadros delimitadores cuyo valor de IoU sea mayor que el umbral, repita los pasos anteriores;



Ilustración 67. Procesamiento NMS

Implementación en código:

```
def nms ( bboxes , iou_threshold , sigma = 0.3 , method = 'nms' ) :
    classes_in_img = list ( set ( bboxes [: , 5 ] ) )
    best_bboxes = [] para cls en classes_in_img :
        cls_mask = ( bboxes [: , 5 ] == cls )
        cls_bboxes = bboxes [ cls_mask ]

        # Proceso 1: Determine si el número de cuadros delimitadores es mayor que 0 mientras len ( cls_bboxes )
        max_ind = np . argmax ( cls_bboxes [: , 4 ] )
        best_bboxes = cls_bboxes [ max_ind ]
        best_bboxes . añadir ( best_bbox )
        cls_bboxes = np . concatenar ( [

            cls_bboxes [: max_ind ] , cls_bboxes [ max_ind + 1 : ] ] ) # Proceso 3: Calcule este cuadro delimitador
            iou = bboxes_iou ( best_bbox [ np . newaxis , : 4 ] , cls_bboxes [: , : 4 ] )
            de peso = np . unos ( ( len ( iou ) , ) ,

            dtype = np . float32 ) afirmar método en [ 'nms' , 'soft-nms' ] if método == 'nms' :
                iou_mask = iou > iou_threshold
                peso [ iou_mask ] = 0.0 if método == 'soft-nms' :
                    peso = np . exp ( - ( 1.0 * iou ** 2 / sigma

                ))

            cls_bboxes [: , 4 ] = cls_bboxes [: , 4 ] * weight
            score_mask = cls_bboxes [: , 4 ] > 0.
            cls_bboxes = cls_bboxes [ score_mask ] return best_bboxes
```

Ilustración 68. Código función NMS



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

---

Al final, todo el cuadro delimitador A es lo que queremos. Tomemos un ejemplo simple: si los cinco cuadros delimitadores y las puntuaciones son: A: 0.8, B: 0.05, C: 0.9, D: 0.5, E: 0.6, el umbral de puntuación establecido es 0.3, los pasos de cálculo son los siguientes sigue:

**Paso 1:** el número de cuadros delimitadores es 5, satisfaciendo las condiciones de iteración;

**Paso 2:** Seleccione el cuadro delimitador A con la puntuación más alta y ordénelo según el orden de puntuación;

**Paso 3:** Calcule el IoU del cuadro delimitador A y los otros cuatro cuadros delimitadores. Suponga que los valores de IoU obtenidos son: B: 0.2, C: 0.7, D: 0.01, E: 0.08, y elimine el cuadro delimitador C;

**Paso 4:** Ahora solo quedan los cuadros delimitadores B, D, E, satisfaciendo las condiciones de iteración;

**Paso 5:** Seleccione el cuadro delimitador D con la puntuación más alta según el orden de puntuación y elimínelo;

**Paso 6:** Calcule el IoU del cuadro delimitador D y los otros dos cuadros delimitadores. Suponga que los valores de IoU obtenidos son: B: 0.06, E: 0.8 y elimine el cuadro delimitador E;

**Paso 7:** Ahora solo queda el cuadro delimitador B, que cumple las condiciones de iteración;

**Paso 8:** Seleccione el cuadro delimitador B con la puntuación más alta de acuerdo con el orden de puntuación y elimínelo;

**Paso 9:** En este momento, el número de cuadros delimitadores es cero y finaliza la iteración.

Finalmente, obtenemos los cuadros delimitadores A, B y D, pero la puntuación del cuadro delimitador B es muy baja, lo que indica que el cuadro delimitador no tiene objetos, por lo que debería descartarse. En el algoritmo YOLO, hay dos situaciones de procesamiento NMS: una es procesar todas las tramas predichas junto con la NMS, y la otra es realizar el procesamiento NMS por separado para cada tipo de trama predicha. En el último caso, se producirá el fenómeno de que el cuadro de predicción pertenece a la categoría A y a la categoría B, lo que es más adecuado para situaciones en las que hay varios objetos en una unidad pequeña al mismo tiempo



---

## **FUNCIÓN DE PERDIDA**

En YOLO v3, el autor considera la tarea de detección de objetivos como un problema de regresión de la predicción del área objetivo y la predicción de categorías, por lo que su función de pérdida también es diferente.

Sin embargo, a través de la interpretación del código fuente de Darknet, la función de pérdida de YoloV3 se puede resumir de la siguiente manera:

- Pérdida de confianza, determinar si hay objetos en el marco de predicción;
- Pérdida de regresión de caja, calculada solo cuando la caja de predicción contiene objetos;
- Pérdida de clasificación, determina a qué categoría pertenecen los objetos en el marco de predicción.

### **Pérdida de confianza**

YOLO v3 optimiza directamente la pérdida de confianza, de modo que el modelo pueda aprender a distinguir las regiones de primer plano y de fondo de la imagen, que es similar a la función RPN en Faster rcnn: La regla de determinación es simple: si el IoU del cuadro predicho y de todos los cuadros reales es menor que un cierto umbral, se determina que es el fondo; de lo contrario, es el primer plano (incluidos los objetos).

### **Pérdida de clasificación**

La pérdida de clasificación utilizada aquí es la entropía cruzada de las dos clasificaciones, es decir, los problemas de clasificación de todas las categorías se atribuyen a si pertenecen a la categoría, por lo que la clasificación múltiple se considera un problema de dos clases. La ventaja de este método es eliminar la exclusión mutua de categorías, especialmente para resolver el problema de la falta de detección causada por la superposición de objetos de múltiples categorías.

```
respond_bbox = label[:, :, :, :, 4 : 5 ]
prob_loss = respond_bbox * tf.nn.sigmoid_cross_entropy_with_logits ( etiquetas = label_prob , logits = conv_raw_prob )
```

### **Pérdida de regresión de caja**

```
respond_bbox = label[:, :, :, :, 4:5]
bbox_loss_scale = 2.0 - 1.0 * label_xywh[:, :, :, :, 2:3] * label_xywh[:, :, :, :, 3:4] / (input_size ** 2)
giou_loss = respond_bbox * bbox_loss_scale * (1 - giou)
giou_loss = tf.reduce_mean(tf.reduce_sum(giou_loss, axis=[1,2,3,4]))
```



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

- Cuanto menor sea el tamaño del cuadro delimitador, mayor será el valor de `bbox_loss_scale`. Sabemos que los anclajes en YOLO v1 han procesado la raíz y el ancho en la pérdida, que es para debilitar el impacto del tamaño del cuadro delimitador en el valor de la pérdida.
- `Respond_bbox` significa que, si la celda de la cuadrícula contiene objetos, se calculará la pérdida del cuadro delimitador.
- Cuanto mayor sea el valor de GloU entre los dos cuadros delimitadores, menor será el valor de pérdida de GloU, por lo que la red se optimizará hacia la dirección de mayor superposición entre el cuadro de predicción y el cuadro real.

la pérdida de IoU original se reemplazó con la pérdida de GloU, esto mejoró la precisión de detección en aproximadamente un 1%. La ventaja de GloU es que mejora el método de medición de distancia entre el cuadro de predicción y el cuadro de anclaje.

### INTRODUCCIÓN A LOS ANTECEDENTES DE GIOU

Este es un método propuesto bastante nuevo para optimizar el cuadro delimitador GioU (Universal IoU). El cuadro delimitador suele estar representado por las coordenadas de la esquina superior izquierda y la esquina inferior derecha, a saber  $(x_1, y_1, x_2, y_2)$ . Bueno, encuentras que este es un vector. La distancia del vector generalmente se puede medir usando el estándar L1 o el estándar L2. Sin embargo, cuando las normas L1 y L2 toman el mismo valor, el efecto de detección será muy diferente. El resultado directo es que el valor de IoU predicho y el marco de detección real han cambiado mucho, lo que indica que los estándares L1 y L2 no reflejan muy bien el efecto de detección.

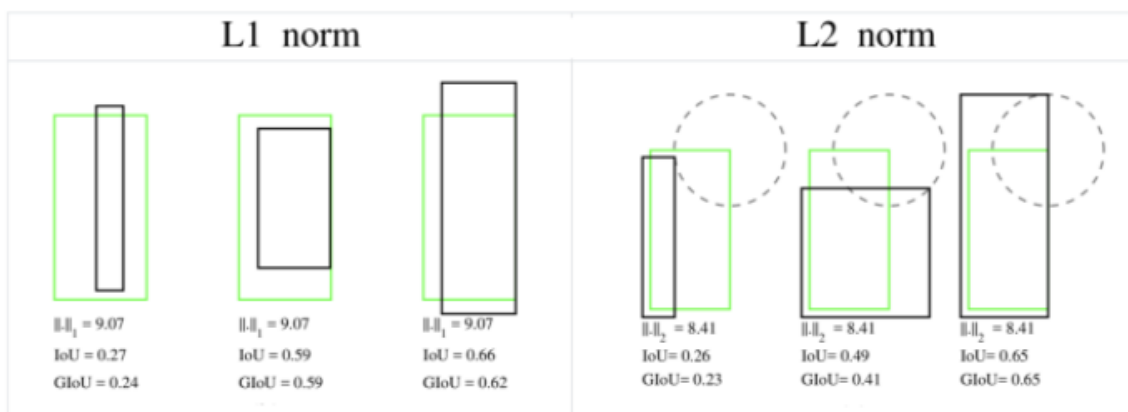


Ilustración 69. Antecedente GIOU



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Cuando los estándares L1 o L2 son iguales, se encuentra que los valores de IoU y GloU son muy diferentes, lo que indica que no es apropiado usar el estándar L para medir la distancia del cuadro delimitador. En este caso, la comunidad académica suele utilizar IoU para medir la similitud entre dos cuadros delimitadores. El autor descubrió que usar IoU tiene dos desventajas que lo hacen inadecuado para funciones de pérdida:

Cuando hay una discrepancia entre el marco predicho y el marco real, el valor de IoU es 0, lo que da como resultado un gradiente de 0 al optimizar la función de pérdida, lo que significa que no se puede optimizar. Por ejemplo, los valores de IoU de la escena A y la escena B son ambos 0, pero el efecto de predicción de la escena B es mejor que el de A porque la distancia entre los dos cuadros delimitadores es más cercana (la norma L es más pequeña):

```
def bbox_giou (boxes1, boxes2):
    ...
    # Calculate the iou value between the two bounding boxes
    iou = inter_area / union_area

    # Calculate the coordinates of the upper left corner and the lower right corner of the smallest closed convex surface
    enclose_left_up = tf.minimum (boxes1 [..., :2], boxes2 [..., :2])
    enclose_right_down = tf.maximum (boxes1 [..., 2:], boxes2 [..., 2:])
    enclose = tf.maximum(enclose_right_down-enclose_left_up, 0.0 )

    # Calculate the area of the smallest closed convex surface C
    enclose_area = enclose [..., 0 ] * enclose [..., 1 ]

    # Calculate the GIoU value according to the GIoU formula
    giou = iou- 1.0 * (enclose_area-union_area) / enclose_area

    return giou
```

Ilustración 70. Código función GIOU

### Entrenamiento de modelos

#### Inicialización de peso

Uno de los problemas que enfrenta el entrenamiento de redes neuronales, especialmente las redes neuronales profundas, es la desaparición o explosión del gradiente, lo que significa que cuando entrenamos redes profundas, la derivada o pendiente a veces se vuelve muy grande, o pequeña, o incluso disminuye exponencialmente. . En este punto, la pérdida que vemos se convertirá a NaN. Supongamos que estamos entrenando una red neuronal tan profunda. Por simplicidad, la función de activación  $g(z) = z$ , y el parámetro de sesgo se ignora.

La comprensión intuitiva aquí es: si el peso  $W$  es solo un poco mayor que 1 o la matriz unitaria, la salida de la red neuronal profunda explotará. Y si  $W$  es ligeramente menor que 1, puede ser 0,9, el valor de salida de cada capa de la red disminuirá exponencialmente. Esto significa que la inicialización adecuada del valor de peso es particularmente importante. El siguiente es un código simple para mostrar esto:



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
%matplotlib inline
plt.figure(figsize=(20,10))

x = np.random.randn(2000, 800) * 0.01 # Create input data
stds = [0.25, 0.2, 0.15, 0.1, 0.05, 0.04, 0.03, 0.02] # Try to use different standard deviations so that the initial weights are

for i, std in enumerate(stds):
    # First Layer - fully connected layer
    dense_1 = tf.keras.layers.Dense(750, kernel_initializer=tf.random_normal_initializer(stddev=std), activation='tanh')
    output_1 = dense_1(x)
    # Second Layer - fully connected layer
    dense_2 = tf.keras.layers.Dense(700, kernel_initializer=tf.random_normal_initializer(stddev=std), activation='tanh')
    output_2 = dense_2(output_1)
    # Third Layer - fully connected layer
    dense_3 = tf.keras.layers.Dense(650, kernel_initializer=tf.random_normal_initializer(stddev=std), activation='tanh')
    output_3 = dense_3(output_2).numpy().flatten()

    plt.subplot(1, len(stds), i+1)
    plt.hist(output_3, bins=600, range=[-1, 1])
    plt.xlabel('std = %.3f' %std)
    plt.yticks([])
plt.show()
```

## TASA DE APRENDIZAJE

La tasa de aprendizaje es uno de los hiperparámetros que más afectan al rendimiento. Si solo podemos acomodar un hiperparámetro, entonces la mejor opción es la tasa de aprendizaje. De hecho, en la mayoría de los casos, la conversión de la pérdida a NaN se debe a una elección incorrecta de la tasa de aprendizaje. En la imagen de abajo, puede ver que si la tasa de aprendizaje es demasiado pequeña, el descenso del gradiente será muy lento. Si la tasa de aprendizaje es demasiado grande, el descenso del gradiente puede exceder el mínimo. Puede que no converja o incluso diverja.

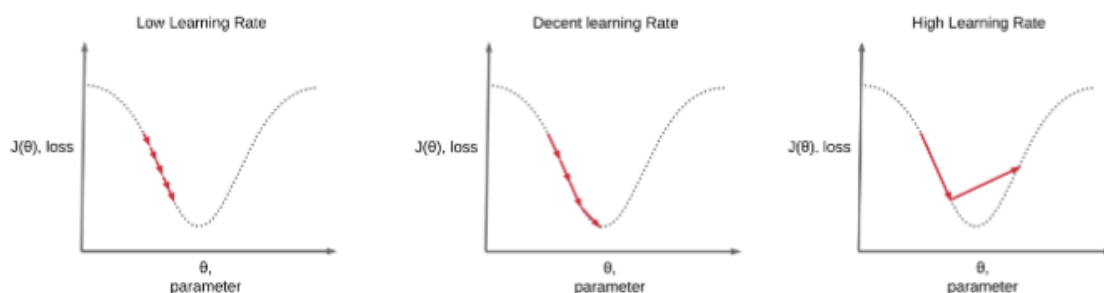


Ilustración 71. Tasa de Aprendizaje

La tasa de aprendizaje es uno de los hiperparámetros que más afectan al rendimiento. Si solo podemos acomodar un hiperparámetro, entonces la mejor opción es la tasa de aprendizaje. De hecho, en la mayoría de los casos, la conversión de la pérdida a NaN se debe a una elección incorrecta de la tasa de aprendizaje. En la imagen de abajo, puede ver que si la tasa de aprendizaje es demasiado pequeña, el descenso del gradiente será muy lento. Si la tasa de aprendizaje es demasiado grande, el descenso del gradiente puede exceder el mínimo. Puede que no converja o incluso diverja.



Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval

## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

---

Dado que la red neuronal es muy inestable al comienzo del entrenamiento, la tasa de aprendizaje al principio debe establecerse muy baja para garantizar que la red pueda tener una buena convergencia. Pero una tasa de aprendizaje más baja hará que el proceso de entrenamiento sea muy lento, por lo que aquí utilizaremos un método para aumentar gradualmente la tasa de aprendizaje más baja a una tasa de aprendizaje más alta para alcanzar la etapa de entrenamiento de "calentamiento". En la red, se llama fase de calentamiento.

Sin embargo, si minimizamos la pérdida de entrenamiento de la red, no se recomienda utilizar siempre una tasa de aprendizaje más alta, porque esto hará que el gradiente de peso oscile hacia adelante y hacia atrás, y la pérdida de entrenamiento es difícil de alcanzar el mínimo global. . Por lo tanto, el método de desintegración del coseno se adoptó finalmente en el mismo documento. Esta etapa puede denominarse etapa de decaimiento del consenso.

### 2.7. MODELOS DE PRUEBA

#### 2.7.1 Modelo Propio

El primer modelo que se probó para detección de hongos esta creado por decisión propia, donde con los conocimientos obtenidos sobre operaciones convolucionales se creó un modelo secuencial el cual nos permite que las capas se agreguen en secuencia, seguido de operaciones convolucionales, y maxpooling, se usó función de activación ReLU y softmax.

Se usaron 2474 imágenes para entrenamiento y 304 imágenes de validación.





# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

```
nb_train_samples = 2474
nb_validation_samples = 304

model = Sequential()

inputShape = (height_shape, width_shape, 3)
model.add(Conv2D(32,(3,3), input_shape=inputShape))
model.add(Conv2D(32,(3,3)))
model.add(MaxPool2D())

model.add(Conv2D(64,(3,3)))
model.add(Conv2D(64,(3,3)))
model.add(Conv2D(64,(3,3)))
model.add(MaxPool2D())

model.add(Conv2D(128,(3,3)))
model.add(Conv2D(128,(3,3)))
model.add(Conv2D(128,(3,3)))
model.add(MaxPool2D())

model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dense(32,activation='relu'))
model.add(Dense(num_classes,activation='softmax', name='output'))

model.summary()

model.compile(loss='categorical_crossentropy',optimizer='adadelta',metrics=['accuracy'])

model_history = model.fit_generator(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator,
    steps_per_epoch=nb_train_samples//batch_size,
    validation_steps=nb_validation_samples//batch_size)
```

## 2.7.2 Modelo ResNet

ResNet-50 es una red neuronal convolucional que tiene 50 capas de profundidad. Puede cargar una versión preentrenada de la red entrenada en más de un millón de imágenes de la base de datos ImageNet . La red previamente entrenada puede clasificar imágenes en 1000 categorías de objetos, como teclado, mouse, lápiz y muchos animales. Como resultado, la red ha aprendido representaciones de características ricas para una amplia gama de imágenes. La red tiene un tamaño de entrada de imagen de 224 por 224.



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

```

from keras.applications.resnet50 import ResNet50
image_input = Input(shape=(width_shape, height_shape, 3))
mResNet50 = ResNet50(input_tensor=image_input, include_top=False, weights='imagenet')
#mResNet50.summary()

last_layer=mResNet50.layers[-1].output

x=Flatten(name='flatten')(last_layer)
x=Dense (128, activation='relu', name='fc1')(x)
x=Dropout(0.3)(x)
out = Dense(num_classes, activation='softmax', name='output')(x)
custom_model = Model(image_input,out)

custom_model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
custom_model.summary()

```

*Ilustración 72. Modificación ResNet50*

En esta ocasión se modificó la estructura de la red neuronal ResNet-50 Para adaptarla a nuestra conveniencia, con el fin de aprovechar sus capas ocultas preentrenadas.

conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	conv5_block3_2_relu[0][0]
conv5_block3_3_bn (BatchNormali	(None, 7, 7, 2048)	8192	conv5_block3_3_conv[0][0]
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	conv5_block2_out[0][0] conv5_block3_3_bn[0][0]
conv5_block3 out (Activation)	(None, 7, 7, 2048)	0	conv5_block3 add[0][0]

=====  
Total params: 23,587,712  
Trainable params: 23,534,592  
Non-trainable params: 53,120

*Ilustración 73. Estructura ResNet-50*

En la image anterior vemos como está construida las ultimas capas de la red neuronal las cuales por medio las modificaremos cambiando

conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	conv5_block3_add[0][0]
flatten (Flatten)	(None, 100352)	0	conv5_block3_out[0][0]
fc1 (Dense)	(None, 128)	12845184	flatten[0][0]
dropout_3 (Dropout)	(None, 128)	0	fc1[0][0]
output (Dense)	(None, 3)	387	dropout_3[0][0]

=====  
Total params: 36,433,283  
Trainable params: 36,380,163  
Non-trainable params: 53,120

*Ilustración 74. Capas Modificadas ResNet-50*



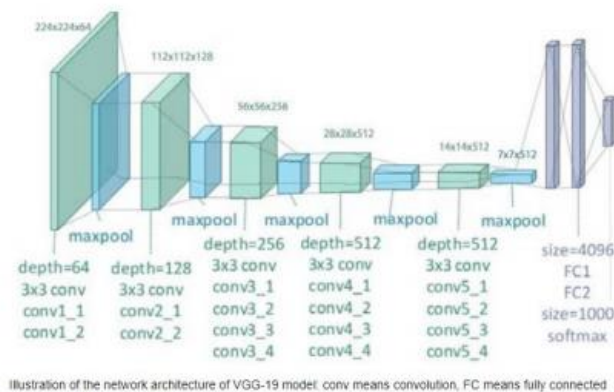
**Autor: Omar Esneider Alvarez Acosta**  
**Directora: Marleny Fernández Sandoval**

# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Se le agrego una capa densa con dimensiones 128x12, a la ca conv5\_block3\_out (Activación) se le aplico flatten para aplanar su salida y solo clasificara 3 clases Oídio, Mildiu y Roya, con un total de 36.433.283 parámetros en total donde 36.380.163 son entregables.

## 2.7.3 Modelo VGG19

esta arquitectura ha sido muy influyente, y el razonamiento intuitivo detrás de las CNN para que estas redes logre una representación jerárquica de los datos suficiente buena, la red convolucional tiene que tener una estructura de capas profunda. Además, esta red también se destacó por su simplicidad: está formada por 19 capas convolucionales (detrás de cada cual se aplica la función ReLU) con filtros de tamaño 3x3, con stride y pad a 1, y las capas de Max pooling (2x2) con stride a 2. La configuración de los pesos está disponible de manera pública, y se ha utilizado en muchas otras aplicaciones y retos como base de extractor de características. Consiguió una tasa de error de 7.2%. En la figura se puede ver el dibujo de esta arquitectura.



En la red neuronal convolucional VGG19 se modificaron las últimas 6 capas como se muestra en la siguiente imagen:



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

```

from keras.applications.vgg19 import VGG19
image_input = Input(shape=(width_shape, height_shape, 3))
mVGG19 = VGG19(input_tensor=image_input, include_top=False, weights='imagenet')
mVGG19.summary()

last_layer=mVGG19.layers[-1].output

x=Flatten(name='flatten')(last_layer)
x=Dense (128, activation='relu', name='fc1')(x)
x=Dropout(0.3)(x)
x=Dense (128, activation='relu', name='fc2')(x)
x=Dropout(0.3)(x)
out = Dense(num_classes, activation='softmax', name='output')(x)
custom_model = Model(image_input,out)

for layer in custom_model.layers[:-6]:
    layer.trainable = False

custom_model.summary()
custom_model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

```

Se uso la función flatten para aplanar las salidas, se agregaron 2 capas densas para conectar todas las salidas de la red neuronal, cada una con función de activación ReLU, ya que esta función nos brinda mejores resultados en la clasificación de imágenes obteniendo la siguiente estructura:

block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 128)	3211392
dropout_6 (Dropout)	(None, 128)	0
fc2 (Dense)	(None, 128)	16512
dropout_7 (Dropout)	(None, 128)	0
output (Dense)	(None, 3)	387
=====		
Total params: 23,252,675		
Trainable params: 3,228,291		
Non-trainable params: 20,024,384		

*Ilustración 75 Estructura modificada VGG19*



*Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval*

# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

## 2.8. SISTEMA DE VISION ARTIFICIAL

Para nuestro sistema de vision artificial se implementó un soporte de madera el cual se instaló la raspberry y la cámara Raspicam que tiene una resolución de 5Mp a 7.5 FPS, que será la encargada de tomar el video en tiempo real, estas imágenes serán analizadas en la raspberry pi 4 mediante el modelo previamente entrenado el cual se encargara de detectar características que identifican la presencia del hongo en las hojas.

El funcionamiento de modelo consiste en analizar la imagen de entrada y dividirla en SxS recuadros y cada recuadro será analizado para la extracción de características que previamente están establecidas en el entrenamiento para luego identificar a que clase o tipo de hongo pertenece, cuando el modelo identifica por completo el objetos este lo encierra en un rectángulo brindándonos una coordenadas y su respectiva posición en la imagen, y como resulta obtenemos la identificación de la enfermedad presente en la hoja mediante una etiqueta, finalmente mediante control remoto entre la Raspberry y un computador portátil donde se podrá visualizar las imágenes de salida y llevar un seguimiento en vivo de las detecciones de los hongos en las plantas.

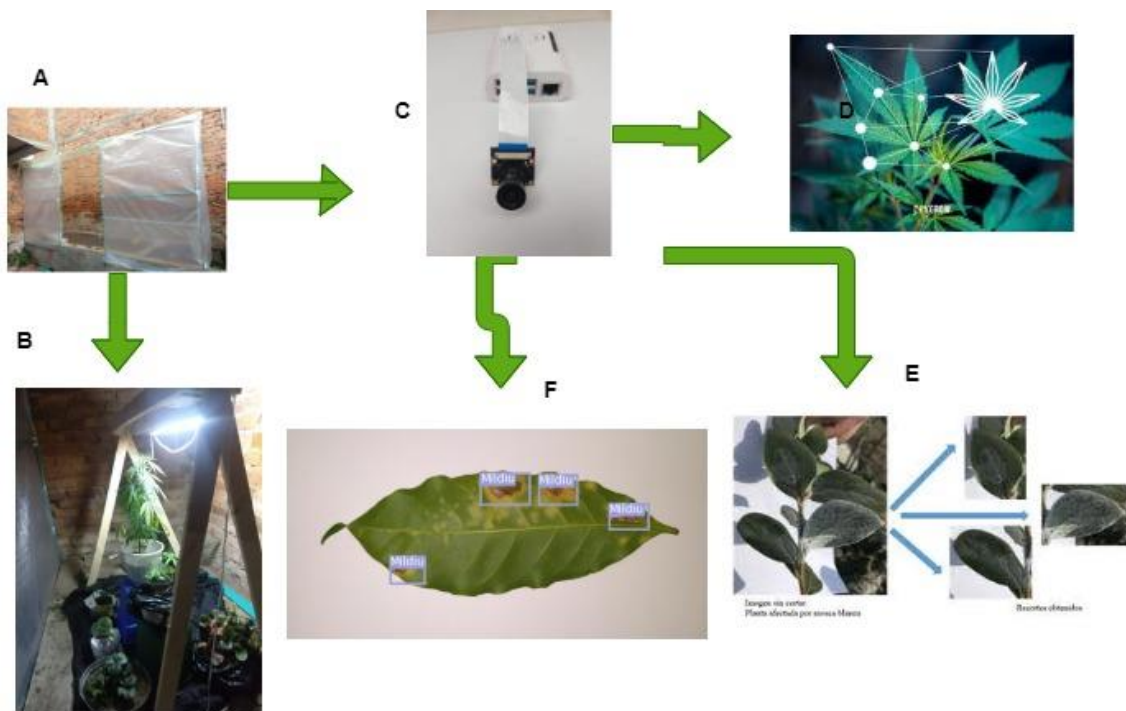


Ilustración 76. A) construcción de invernadero, B) Soporte raspberry, iluminación, C) Cámara Raspicam, D) Análisis imagen de entrada, E) extracción de características, F) Etiquetado imagen de salida

## 2.9. IMPLEMENTACIÓN



Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval

## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Para la detección de hongos se construyó un invernadero que nos ayudara a tener un crecimiento favorable de la planta y mejorara la adquisición de imágenes en la detección de los hongos, de igual manera cuenta con un sistema que se encarga de la toma de imágenes para luego ser procesadas e identificadas, este sistema consta de una raspberry pi 4, cámara 5mp OV5647 distancia focal 3.6mm ajustable, un soporte de rieles que ayudara al desplazamiento de la cámara para poder supervisar en totalidad el invernadero este sistema cuenta con un sistema de iluminación que ayudara al procesamiento de imágenes.

Después de tener nuestro modelo entrenado y listo procedemos a instalar el programa en la raspberry pi4 que será la encargada de capturar las imágenes que luego serán procesadas para detectar la presencia de hongos. la cámara será configurada para que las tomas sean más precisas, de la presencia de una planta enferma, el enlace de estos dos sistemas es por conexión serie, al vincular el sistema de vision artificial con el sistema de comunicación creara un proyecto más confiable y llamativo.



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

---



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

# CAPÍTULO 3



*Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval*



### **3. VALIDACIÓN Y RESULTADOS**

Al comprobar que nuestro sistema de visión artificial nos arroja resultados favorables, es decir nos detecta lo que para nosotros es la presencia de un hongo en la planta ya sea Mildiu, Oídio o Roya y para tener más certeza sobre el objetivo cumplido contamos con la asesoría de una ingeniera agrónoma que con sus conocimientos sobre estas enfermedades nos dio la aprobación del sistema, todo esto se llevara a cabo en varios pasos

1. se tomaron varias imágenes de hojas infectadas con hongos, fueron analizadas por el sistema, dando como resultado la detección de la enfermedad, este resultado se le enviaron a la ingeniera para que ella los revisara y diera un veredicto positivo.
2. Otra forma de validar nuestro proyecto y comprobar que realmente se está ejecutando de forma correcta es tomando unas imágenes de plantas infectadas por los hongos, estos datos son enviados por la persona encargada de la validación del sistema, luego son evaluadas por el algoritmo de vision artificial dando como resultado la detección correcta de los hongos en cada planta.

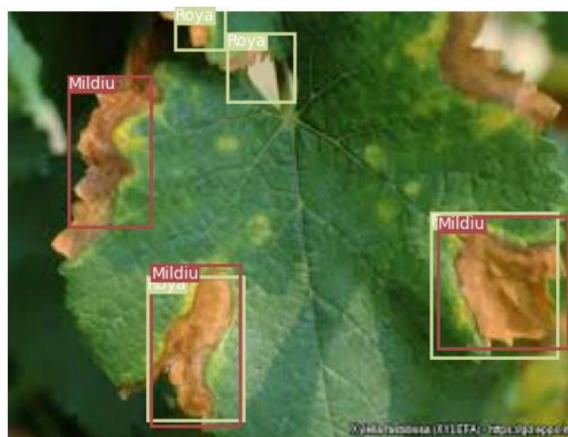


# DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

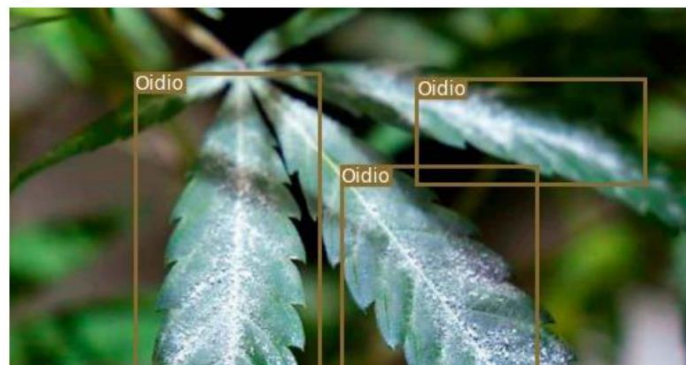
Pamplona, 09 de Diciembre de 2020

Yo, María Helena Castro Pineda Ingeniera agrónoma egresada de la universidad de Pamplona con tarjeta profesional numero 1246, puedo constatar que los síntomas de las siguientes enfermedades producidas por hongos patógenos, detectadas por el sistema son:

**Mildeo:** Los síntomas característicos de esta enfermedad son manchas de color café con bordes amarillos y en el envés de la hoja forma una especie de moho.



**Oídio:** Este hongo produce una capa de aspecto harinoso o algodonoso de color blanco o gris.



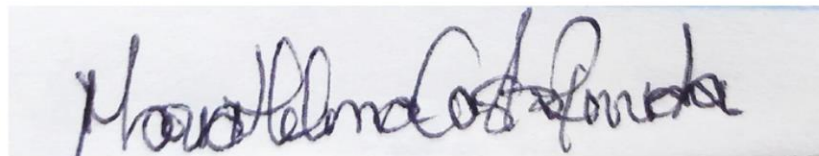
*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

**Roya:** Este hongo proporciona a la hoja una lesión clorótica inicialmente con decoloración de aéreas de la hoja y polvillo amarillo o naranja ubicado en el envés de esta



Cada uno de estos corresponde a síntomas típicos que producen estos hongos en las plantas.



**MARÍA HELENA CASTRO PINEDA**  
C.C. No.1.052.313.543 de Belén  
Ingeniera Agrónoma  
T.P. No. 121019-0529027 NTS



*Autor: Omar Esneider Alvarez Acosta*  
*Directora: Marleny Fernández Sandoval*

## RESULTADOS

Después de probar varios modelos ResNet-50, VGG19, Modelo Propio y YoloV3 se pudieron analizar sus resultados donde se pudo comprobar que el mejor método para detección de hongos es sin duda YoloV3 es más rápido y preciso dando resultados favorables en el sistema, a continuación, se demostraran los resultados de cada modelo mediante matriz de confusión y las graficas de entrenamiento

Empezamos con la graficas de entrenamiento de YoloV3 que es el modelo que escogimos como referente para el sistema de vision artificial



Ilustración 77. Grafica entrenamiento para 100 imagen [Autor]

En esta grafica se puede observar como la perdida va disminuyendo y la precisión aumenta a medida que avanza las iteraciones, es un comportamiento que es favorable mas no quiere decir que sea preciso esto ocurre para los datos de entrenamiento, cada clase cuenta con 1000 imágenes.



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

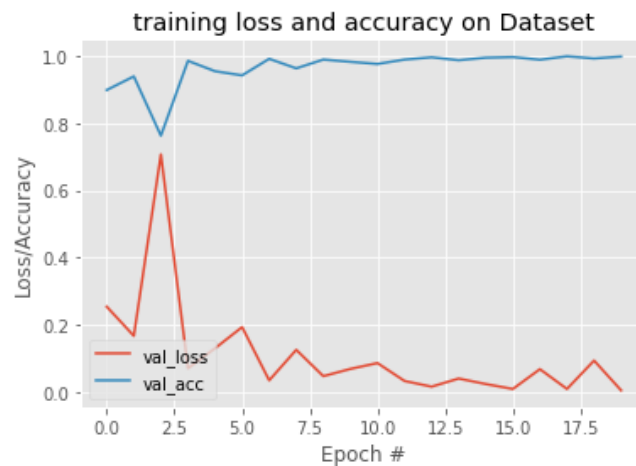


Ilustración 78. Grafica validación 1000 imágenes [Autor]

En esta ocasión muestra el avance del modelo con los datos de validación que son 250 por clase en total 750 imágenes, la gráfica nos muestra el comportamiento de la red neuronal convolucional, muestra un comportamiento más acertado pero que puede mejorar agregándole más capas o quizás con más número de épocas, se está viendo mejorías, pero todavía falta unos detalles para lograr lo esperado



Ilustración 79. Grafica modelo de Data de Entrenamiento

Esta grafica demuestra que con más datos de entrenamiento se puede llegar a un mejor resultado en menos épocas, demostrando que la cantidad de imágenes es muy importante para obtener un modelo bastante estable, pero para poder trabajar con una cantidad abundante de imágenes, se necesitaría un procesador robusto el cual en este momento no es posible trabajar.



Autor: Omar Esneider Alvarez Acosta  
Directora: Marleny Fernández Sandoval

### MATRIZ DE CONFUSIÓN

En esta matriz 3x3 podemos observar en la diagonal principal todas las evaluaciones correctas y son denominados verdaderos positivos (V.P), los valores que no correspondan a una misma clase y que se encuentren en la diagonal principal son llamados verdaderos negativos (V.N), los falsos positivos son los valores que encontramos en el eje principal pero ese encuentran en la parte derecha de este (F.P) y los falsos negativos se encuentran en la parte izquierda (F.N), en los ejes (x, y) se encuentran las clases de hongos del proyecto 0 = Mildiu, 1 = Oídio, 2 = Roya.

$$Precision = \frac{(V.P)}{V.P+F.P} [8]$$

$$Recall = \frac{V.P}{V.P+F.N} [9]$$

$$F1 - Score = 2 * \frac{Precision*Recall}{Precision+Recall} [10]$$

$$Accuracy = \frac{V.P+V.N}{V.P+F.P+V.N+F.N} [11]$$

	precision	recall	f1-score	support
0	1.0000	0.9604	0.9798	101
1	1.0000	1.0000	1.0000	102
2	0.9619	1.0000	0.9806	101
accuracy			0.9868	304
macro avg	0.9873	0.9868	0.9868	304
weighted avg	0.9873	0.9868	0.9868	304

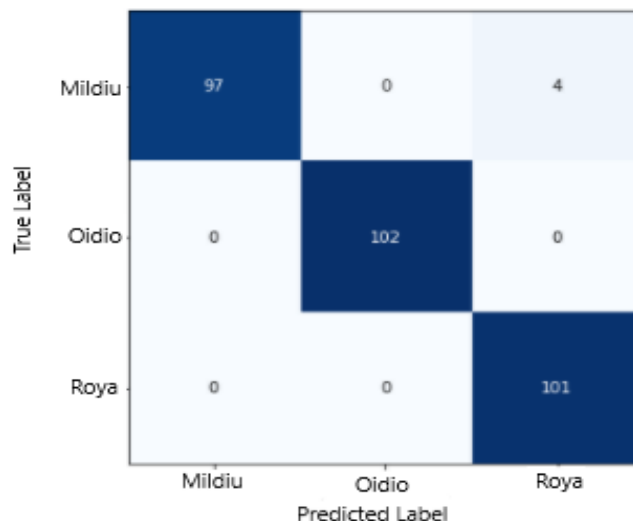


Ilustración 80. Matriz de Confusión YoloV3



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

Como se muestra en la ilustración 79 se puede observar en la clase 0 "Mildiu" su entrenamiento nos marca 97 aciertos de 101 con una precisión de 100% pero con un recall del 96% eso quiere decir que para esa clase las probabilidades de clasificar hongos de otra clase como mildiu es cero, en cambio existe la posibilidad de que esas mismas imágenes sean clasificadas como Oídio o Roya

Para la clase de Oídio denotamos que fue la clase mejor entrenada ya que cuenta con una precisión y un recall de 100% este resultado nos muestra que de 102 imágenes de prueba fueron todos exitosos y que no existe posibilidades de que otra clase sea clasificada como oídio.

En la clase de la Roya se obtuvo una precisión de 96% de precisión y 100% de recall, lo que quiere decir que de las 101 imágenes de roya todas son clasificadas como roya, pero existe una posibilidad de que otras clases sean clasificadas como esta.

Las matrices resultantes de los modelos de ResNet-50, VGG19, y el modelo son los siguientes

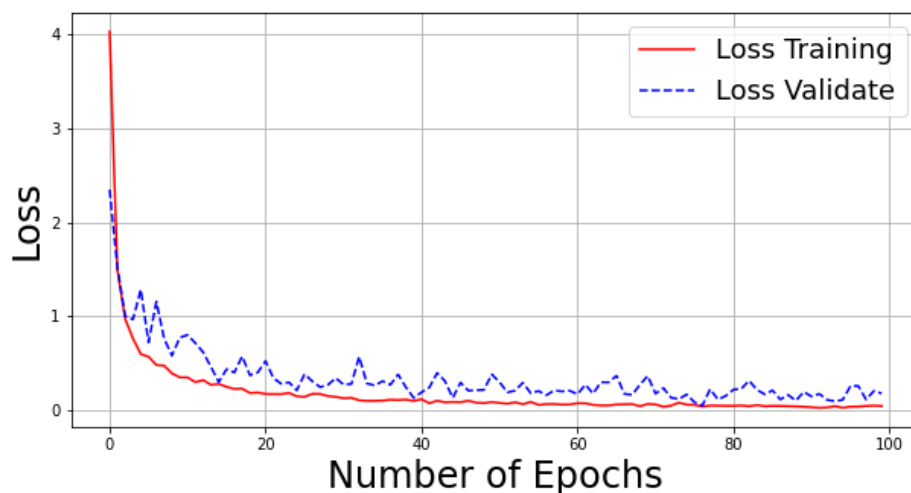


Ilustración 81. Pérdidas de entrenamiento y validación del modelo propio [propio]



**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

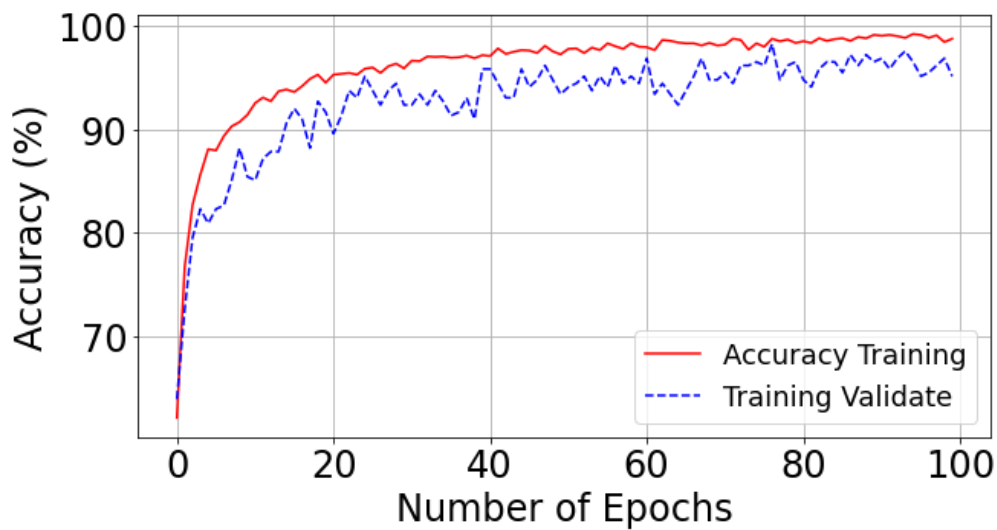


Ilustración 82. Porcentaje Accuracy de entrenamiento y validación [propia]

En la ilustración 80-81 se observa el avance del entrenamiento y como la perdida va disminuyendo a medida que avanza las épocas, se puede visualizar un comportamiento deseado.

0	0.9737	0.3663	0.5324	101
1	0.7083	0.3333	0.4533	102
2	0.4220	0.9109	0.5768	101
accuracy			0.5362	304
macro avg	0.7013	0.5369	0.5208	304
weighted avg	0.7014	0.5362	0.5206	304

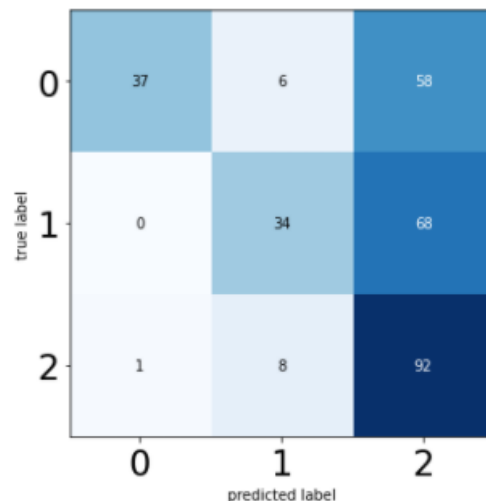


Ilustración 83. Matriz de decisión modelo propio [propio]

Este modelo nos brinda una precisión de **53%** el cual es muy bajo para el sistema de clasificación

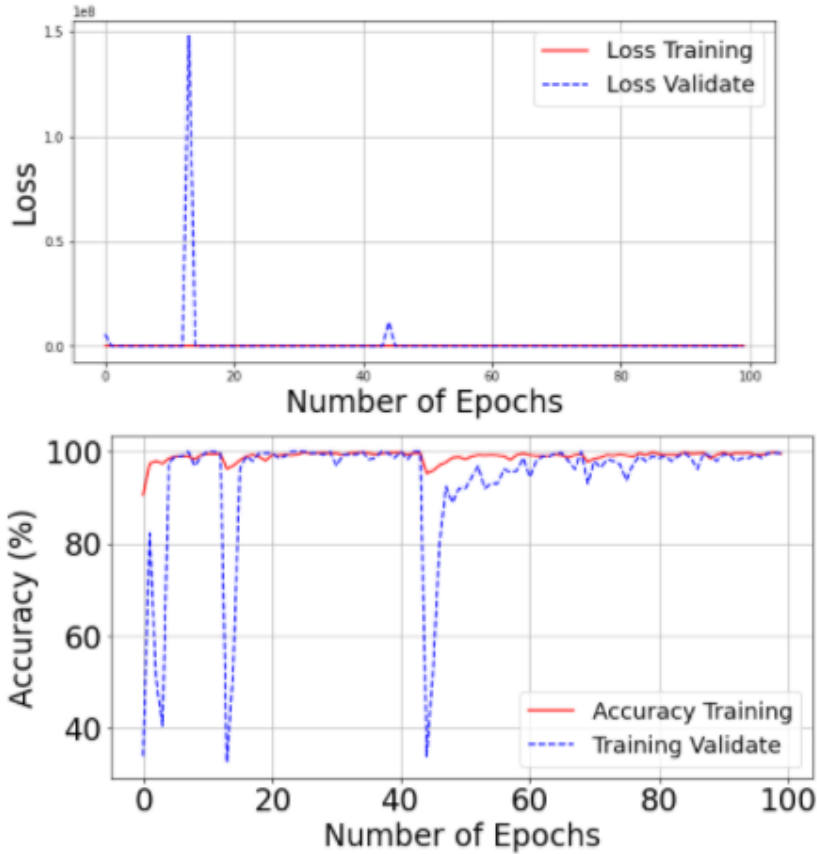




**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

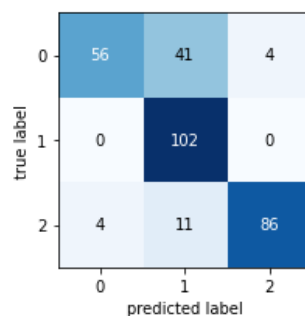
**RESNET-50**

Las gráficas que se muestran a continuación son las referentes a la red neuronal convolucional ResNet-50, se observa un comportamiento no muy preciso ya que la gráfica de pérdidas se observa prácticamente sin pérdidas.



*Ilustración 84. perdidas y Accuracy ResNet-50*

	precision	recall	f1-score	support
0	0.9333	0.5545	0.6957	101
1	0.6623	1.0000	0.7969	102
2	0.9556	0.8515	0.9005	101
accuracy			0.8026	304
macro avg	0.8504	0.8020	0.7977	304
weighted avg	0.8498	0.8026	0.7977	304



*Ilustración 85. Matriz de confusión ResNet-50*



## DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS

La precisión de este modelo es tan solo del **80%** es un buen porcentaje, pero no es el esperado, por eso se toma la decisión de buscar un modelo con más precisión.

### VGG19

Para el modelo VGG19 se usaron los mismos datos que los anteriores obteniendo los siguientes gráficos:

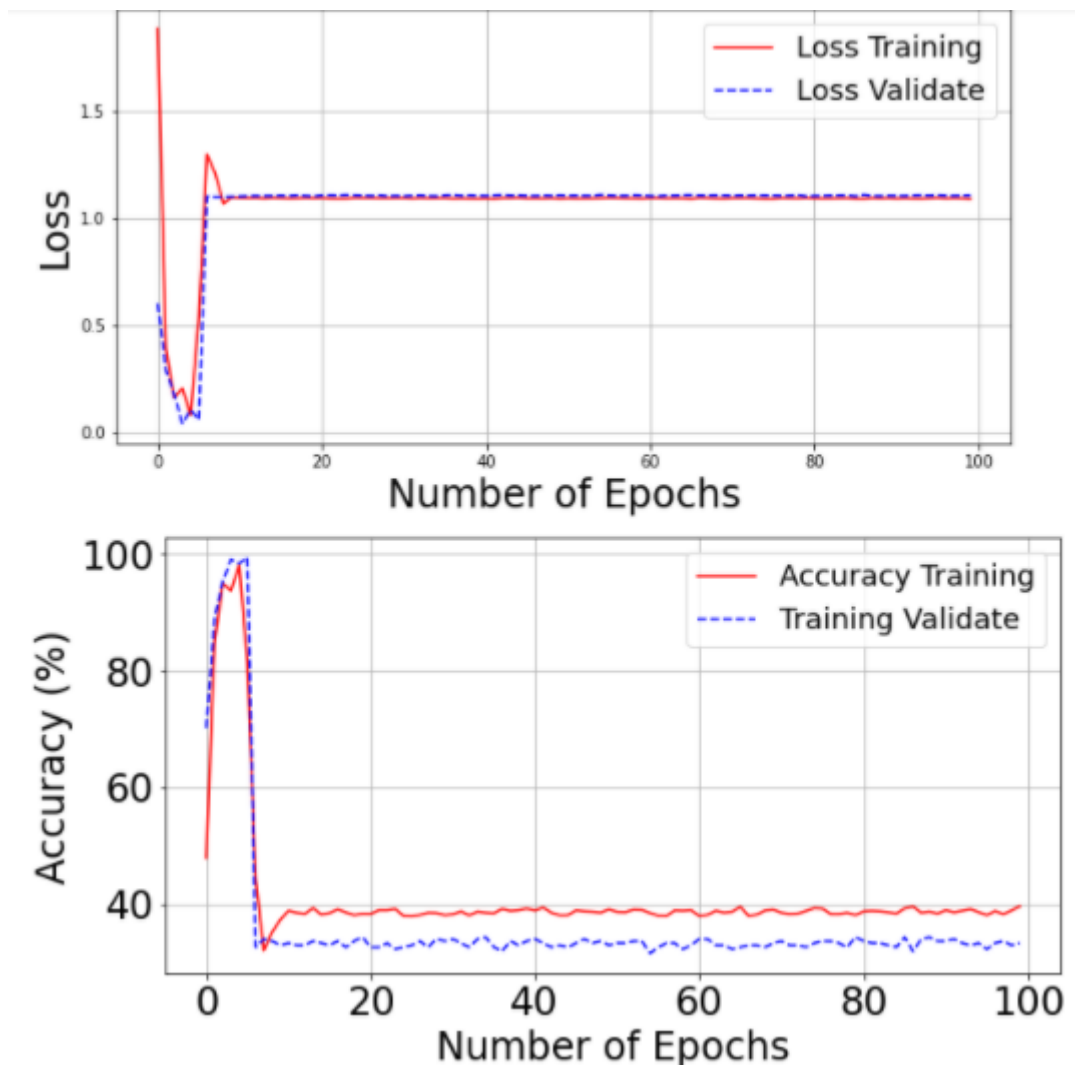


Ilustración 86. perdida y accuracy VGG19

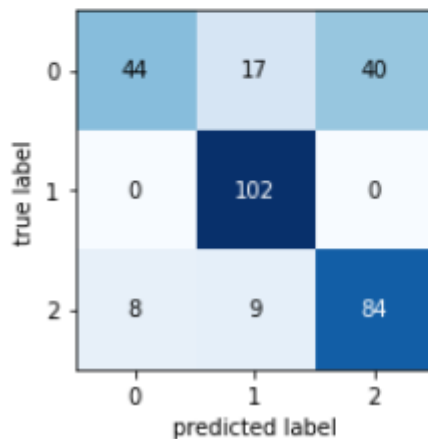


**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

---

Los resultados de este modelo no fueron los esperados, nos dan una precisión de **75%** el cual sigue siendo muy bajo para obtener buenos resultados en la clasificación.

	precision	recall	f1-score	support
0	0.8462	0.4356	0.5752	101
1	0.7969	1.0000	0.8870	102
2	0.6774	0.8317	0.7467	101
accuracy			0.7566	304
macro avg	0.7735	0.7558	0.7363	304
weighted avg	0.7736	0.7566	0.7368	304



En esta última matriz podemos observar que la clase de mildiu cuya cantidad es de 101 imágenes solo 44 son acertadas 17 fueron clasificadas como Oídio y 40 como Roya, en la clase de Oídio se presenta una precisión y un recall de 100% es la única clase que clasifica bien, en cambio en la roya de 101 imágenes solo 84 son correctas, 9 de oídio y 8 de Mildiu.



## 4. CONCLUSIONES

Los sistemas de inteligencia artificial basados en redes neuronales están presentes en diferentes campos de la producción agrícola y la producción industrial, se pueden usar en sistemas de clasificación de frutas según su madurez, detección de enfermedades en cultivos, robótica agrícola entre otros, su uso está justificado por que han demostrado resultados beneficiosos, gracias al uso de los sistemas de vision artificial es posible automatizar tareas laboriosas de forma más segura, la construcción de un invernadero nos facilitó el proceso de detección de características en las hojas ya que se controla el medio y lo más importante la iluminación.

Los resultados de este proyecto demuestran la posibilidad de usar el algoritmo como YoloV3 – YoloV4 como herramienta de precisión para la detección de hongos en plantas de cualquier especie facilitando al agricultor el control del mismo, garantizando su cosecha. Durante el desarrollo del proyecto se identificó gran falencia al momento de crear nuestro modelo ya que no contábamos con un poder computacional capaz de sostener los requerimientos que nos exigía el sistema para el entrenamiento y así poder obtener un aprendizaje del modelo con eficiencia mayor a 80% de igual manera no contábamos con la cantidad de imágenes de nuestra planta base como lo es el cannabis ya que por ser una especie no tan legal en el mundo se tomó la decisión de entrenar con imágenes de distintas plantas como la hoja de fresa, café y manzana creando así un proyecto que beneficiara además del cultivo del cannabis otros cultivos que tengan la presencia de los hongos.

Finalmente, gracias a la colaboración de una ingeniera agrónoma logramos validar que el sistema detecta de manera eficaz los hongos presentes en las plantas, y para poder llegar hasta allí se necesitó mucha información acerca de las características que presenta cada enfermedad, y así se logro cumplir los objetivos que se plantearon.



## 5. REFERENCIAS

- [1] C.-T. Wang *et al.*, "Ambient measurements of monoterpenes near Cannabis cultivation facilities in Denver, Colorado," *Atmos. Environ.*, vol. 232, p. 117510, 2020, doi: <https://doi.org/10.1016/j.atmosenv.2020.117510>.
- [2] G. R. Potter *et al.*, "Global patterns of domestic cannabis cultivation: Sample characteristics and patterns of growing across eleven countries," *Int. J. Drug Policy*, vol. 26, no. 3, pp. 226–237, 2015, doi: <https://doi.org/10.1016/j.drugpo.2014.12.007>.
- [3] G. Green, "The Cannabis Grow Bible," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.
- [4] M. Integrado, "Mildiu en cebolla."
- [5] S. Takamatsu, S. Limkaisang, S. Kom-un, A. Bolay, and C. To-anun, "Identity of a powdery mildew fungus occurring on Paeonia and its relationship with Erysiphe hypophylla on oak," *Mycoscience*, vol. 47, no. 6, pp. 367–373, 2006, doi: <https://doi.org/10.1007/S10267-006-0317-5>.
- [6] U. E. Junta de Andalucía, "Roya (Puccinia spp.)," *Red alerta e Inf. Fitosanit. Cons. Agric. y pesca.*, pp. 1–4, 2010.
- [7] H. vastatrix Berk, B. Actualizado aumentado por, and I. Miguel Barquero Miranda, *Instituto Del Cafe De Costa Rica Recomendaciones Para El Combate De La Roya Del Cafeto*. 2013.
- [8] N. Muthukrishnan, F. Maleki, and K. Ovens, "Brief History of Artificial Intelligence," p. 5821.
- [9] E. Anderson, *Inteligencia artificial y sistemas expertos*, no. 26. 1996.
- [10] P. Suresh Kumar, H. S. Behera, A. K. K, J. Nayak, and B. Naik, "Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades," *Comput. Sci. Rev.*, vol. 38, p. 100288, 2020, doi: <https://doi.org/10.1016/j.cosrev.2020.100288>.
- [11] Ahmed Fawzy Gad, *Practical Computer Vision Applications Using Deep Learning with CNNs: With Detailed Examples in Python Using TensorFlow and Kivy*. 2018.
- [12] A. E. Gaona Barrera and D. M. Ballesteros Larrotta, "Selección eficiente de arquitecturas neuronales empleando técnicas destructivas y de regularización," *Rev. Tecnura*, vol. 16, no. 33, p. 158, 2012, doi: [10.14483/udistrital.jour.tecnura.2012.3.a11](https://doi.org/10.14483/udistrital.jour.tecnura.2012.3.a11).
- [13] C. Peñafiel and R. Ávila, "Inteligencia Artificial," *Intel. Artif.*, vol. 2, no. 6, pp. 1–33, 2007, doi: [10.13140/2.1.3720.0960](https://doi.org/10.13140/2.1.3720.0960).
- [14] M. McLoughlin, "Synoptic pericope order," *Ephemer. Theol. Lovan.*, vol. 85, no. 1, pp. 71–97, 2009, doi: [10.2143/ETL.85.1.2040696](https://doi.org/10.2143/ETL.85.1.2040696).
- [15] "Procesamiento digital de imágenes," *Perfiles Educ.*, no. 72, 1996.



**DESARROLLO DE UN SISTEMA DE VISION ARTIFICIAL PARA LA DETECCIÓN DE HONGOS EN PLANTAS DE CANNABIS MEDICINAL MEDIANTE REDES NEURONALES EN AMBIENTES CONTROLADOS**

- [16] M. Biasoni, "Métodos de umbralización de imágenes digitales basados en entropía de shannon y otros," vol. XXX, pp. 1–4, 2011.
- [17] L. S. B. A. & Wolfman, "No Title No Title," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013, doi: 10.1017/CBO9781107415324.004.
- [18] J. A. Taquía-Gutiérrez, "El procesamiento de imágenes y su potencial aplicación en empresas con estrategia digital," *Interfases*, vol. 0, no. 010, p. 11, 2017, doi: 10.26439/interfases2017.n10.1767.
- [19] G. Viera-Maza, "Procesamiento de imágenes usando OpenCV aplicado en Raspberry Pi para la clasificación del cacao," *Thesis*, p. 136, 2017, [Online]. Available: [https://pirhua.udep.edu.pe/bitstream/handle/11042/2916/IME\\_218.pdf?sequence=1&isAllowed=y](https://pirhua.udep.edu.pe/bitstream/handle/11042/2916/IME_218.pdf?sequence=1&isAllowed=y).
- [20] R. Magro, "Binarización de imágenes digitales y su algoritmia como herramienta aplicada a la ilustración entomológica," *Boletín la SEA*, vol. 53, no. 53, pp. 443–464, 2013.
- [21] B. R. Soc, E. Hist, and N. Sec, "pr en s a En En pr en En en pr sa," vol. 00, pp. 1–14, 2012.
- [22] J. A. Taquía-Gutiérrez, "Técnicas de procesamiento digital de imágenes y reconocimiento de patrones," pp. 11–17, 2017, [Online]. Available: [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/nieto\\_b\\_d/capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/nieto_b_d/capitulo2.pdf).
- [23] E. Básica *et al.*, "REDES NEURONALES ARTIFICIALES- Xabier Bosogain Olabe.pdf," [Online]. Available: [http://cvb.ehu.es/open\\_course\\_ware/castellano/tecnicas/redes\\_neuro/contenidos/pdf/libro-del-curso.pdf](http://cvb.ehu.es/open_course_ware/castellano/tecnicas/redes_neuro/contenidos/pdf/libro-del-curso.pdf).
- [24] Ó. Picazo Montoya, "Redes Neuronales Convolucionales Profundas para el reconocimiento de emociones en imágenes," p. 45, 2018, [Online]. Available: [http://oa.upm.es/51441/%0Ahttp://oa.upm.es/51441/1/TFM\\_OSCAR\\_PICAZO\\_MONTOYA.pdf](http://oa.upm.es/51441/%0Ahttp://oa.upm.es/51441/1/TFM_OSCAR_PICAZO_MONTOYA.pdf).
- [25] J. Sebastian and G. Prieto, "Redes neuronales convolucionales y redes neuronales recurrentes en la transcripción automática," no. July, 2019, doi: 10.13140/RG.2.2.10855.39843.
- [26] T. F. De Grado, "Aplicación de percepción mejorada para personas con problemas visuales usando deep learning y dispositivos vestibles," 2020.
- [27] R. P. Sanchez, "Reconocimiento de imágenes mediante Raspberry Pi," p. 91, 2016, [Online]. Available: <https://www.semanticscholar.org/paper/Deep-Learning-para-reconocimiento-de-imágenes-en-Pi-Vlaeminch/10f5f265496d32dd1ae52c846497378792a681c2#paper-header>.

