



DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO  
PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR  
EN LOS CULTIVOS DE FRESA.



ACREDITACIÓN INSTITUCIONAL  
*Avanzamos... ¡Es nuestro objetivo!*

FERNANDO RODRÍGUEZ DÁVILA.

Trabajo de grado.

Director.  
Diego Alfonso Peláez Carrillo.

UNIVERSIDAD DE PAMPLONA.  
FACULTAD DE INGENIERÍAS Y ARQUITECTURA.  
PROGRAMA DE INGENIERÍA ELECTRÓNICA.  
PAMPLONA – NORTE DE SANTANDER.  
COLOMBIA.  
2020

**Nota de aceptación.**

---

---

---

---

---

---

---

---

---

Firma de jurado

---

Firma de jurado

---

Firma de jurado

## **DEDICATORIA.**

Quiero dedicarle este logro a Dios, pues en él encontré la fortaleza para llevar a cabo todo este proceso, a mi abuelo que no se encuentra en estos momentos conmigo pero que durante su estancia en este mundo fue mi amigo, mi papá y un ser maravilloso que marcó mi vida, a toda mi familia que con su apoyo y amor hicieron que esto sea posible, a mi hermano Haider que ha sido muy especial conmigo apoyándome desde niño en todas las metas que me propuse hasta llegar a este punto, a mi pareja quien me brindó su apoyo, motivación y amor durante todo este proceso.

## **AGRADECIMIENTOS.**

Doy gracias a Dios por estar en estos momentos aquí, respirando, viviendo este día, logrando una meta más. Gracias a todas esas personas que me apoyaron a llegar hasta este punto, a mi familia por su amor, a mis hermanos y hermanas que los amo demasiados, a mis papas por todo ese amor brindado, a mi abuelo quien me brindó su apoyo desde niño, aunque no esté conmigo en estos momentos esto es para él, a mi pareja que me ha apoyado con este proyecto, a mis amigos de la universidad que fueron mi apoyo cuando me encontraba solo, a los docentes quienes dieron lo mejor de ellos para brindarme su conocimiento y sabiduría. Además, quiero agradecerle al señor Ángel que por cosas del destino lo encontré durante la búsqueda de los cultivos de fresa y quien fue, el quien me ayudó a encontrar el cultivo de la vereda el totumo. ¡Profundamente agradecidos con todos, los aprecio demasiado!

## **RESUMEN DEL PROYECTO.**

Países desarrollados han implementado tecnologías innovadoras en el sector agrícola, que le permiten al campesino detectar cuando hay presencia de anomalías en sus cultivos; a diferencia de países en vía de desarrollo como es el caso de Colombia, donde estamos en ese proceso de crecimiento y desarrollo de nuevas tecnologías asequibles al campesino del común o de bajos recursos, que le permitan optimizar la producción de los cultivos, permitiendo la detección de enfermedades por medio de herramientas y procesamientos basados en visión artificial.

La visión artificial acompañado de un procesamiento digital de imágenes son los fundamentos principales en este proyecto. Por tanto, el objetivo principal de este trabajo es monitorear sectores previamente definidos que se acoplan a características específicas dentro de los cultivos de fresa, por medio de un dron que posee una cámara RGB se realizan las capturas de imágenes a un sector del cultivo seleccionado que luego pasaran por un preprocesamiento con el fin de adaptarlo al método de inteligencia artificial. Luego de la adaptación de los datos al método de inteligencia artificial se realizará el entrenamiento y por último la clasificación cualitativa de la enfermedad Antracnosis, por consiguiente, se realiza el proceso de validación del resultado del método de inteligencia artificial incluyendo un factor de confusión con el fin de garantizar una eficiencia mayor del 75%.

Desde la adquisición de los Datos donde se estableció un plan de vuelo que cumplió con las condiciones previamente definidas, permitió adquirir la elaboración de un algoritmo de preprocesamiento que llevó cabo el recorte de zonas de interés del cultivo de fresa con el fin de extraer las imágenes individuales de los arbustos de fresa. Estas capturas individuales se redimensionaron a un tamaño estándar de 400x400 y con esas alteraciones se creó el set de entrenamiento y validación. Estos fueron proporcionados a una red neuronal convolucional de 4 capas convolucionales y 2 capas densa para su posterior entrenamiento con un número de épocas de 3 y posteriormente la validación con el set de validación obteniendo un porcentaje del 78.08% superando las expectativas anteriormente mencionadas.

## CONTENIDO.

INTRODUCCIÓN. ....	12
PLANTEAMIENTO DEL PROBLEMA. ....	14
JUSTIFICACIÓN. ....	16
OBJETIVO GENERAL. ....	17
OBJETIVOS ESPECÍFICOS. ....	17
CAPÍTULO I. ....	18
1. REFERENTES TEÓRICOS. ....	19
1.1. FRESA O FRUTILLA. ....	19
1.1.1. Corona. ....	19
1.1.2. Sistema radicular. ....	19
1.1.3. Hojas. ....	19
1.1.4. Flores y frutos. ....	20
1.1.5. Estolones. ....	20
1.1.6. Fresa camarosa. ....	21
1.1.7. Fresa ventana. ....	21
1.1.8. Fresa camino real. ....	22
1.1.9. Fresa palomar. ....	22
1.1.10. Fresa Albión. ....	22
1.1.11. Fresa san Andreas. ....	23
1.1.12. Fresa Monterrey. ....	24
1.1.13. Fresa portola. ....	24
1.1.14. Área y producción de la fresa. ....	25
1.2. ANTRACNOSIS. ....	25
1.2.1. En las hojas, ....	26
1.2.2. En pecíolos y estolones. ....	26
1.2.3. En la corona, ....	26
1.2.4. En el fruto, ....	27
1.2.5. Ciclo. ....	27
1.3. PALLIDOSIS. ....	27

1.4. DRONE.....	28
1.4.1. DJI MATRICE 100.....	28
1.4.2. DJI Mavic 2 Pro/Zoom.....	31
1.5. PYTHON.....	33
1.5.1. Numpy.....	34
1.5.2. TensorFlow.....	34
1.5.3. Keras.....	35
1.5.4. OpenCV (Open source computer vision library).....	38
1.6. INTELIGENCIA ARTIFICIAL.....	44
1.6.1. REDES NEURONALES.....	44
1.6.2. APRENDIZAJE PROFUNDO.....	45
1.6.3. APRENDIZAJE AUTOMÁTICO O MACHINE LEARNING.....	46
1.7. VISIÓN ARTIFICIAL.....	51
1.8. REDES NEURONALES CONVOLUCIONALES.....	51
1.8.1. Maxpooling.....	52
1.8.2. Softmax.....	53
1.8.3. Dropout.....	53
1.8.4. Overfitting y Underfitting.....	54
1.9. ESTADO DEL ARTE.....	55
CAPÍTULO II.....	60
2. VARIABLES FAVORABLES PARA UN AMBIENTE CONTROLADO QUE DEFINA EL PROTOCOLO PARA LA ADQUISICIÓN DE IMÁGENES.....	61
CAPÍTULO III.....	64
3. ADQUISICIÓN DE LAS IMÁGENES DE PLANTAS SANAS Y PLANTAS AFECTADAS POR ANTRACNOSIS PARA CONSTRUCCIÓN DE UNA BASE DE DATOS.....	65
CAPÍTULO IV.....	72
4. IMPLEMENTACIÓN DE UN ALGORITMO DE PREPROCESAMIENTO A LA BASE DE DATOS QUE PERMITA DEPURAR LA INFORMACIÓN ADQUIRIDA.....	73
CAPÍTULO V.....	79
5. IMPLEMENTACIÓN DE DIFERENTES MÉTODOS DE CLASIFICACIÓN INTELIGENTE PARA LA IDENTIFICACIÓN DE ANTRACNOSIS EN LOS CULTIVOS DE FRESA.....	80

CAPÍTULO VI.....	84
6. VALIDACIÓN DEL MÉTODO DE INTELIGENCIA ARTIFICIAL SELECCIONADO INCLUYENDO FACTORES DE CONFUSIÓN Y RESULTADOS. ....	85
7. CONCLUSIONES.....	88
8. TRABAJOS FUTUROS.....	90
REFERENCIAS BIBLIOGRÁFICAS.....	90



## LISTA DE FIGURAS.

Figura 1 Planta de fresa .....	20
Figura 2 fresa camarosa .....	21
Figura 3 Fresa ventana .....	21
Figura 4 Fresa camino real.....	22
Figura 5 Fresa palomar .....	22
Figura 6 Fresa Albión .....	23
Figura 7 Fresa San Andreas .....	23
Figura 8 Fresa Monterrey .....	24
Figura 9 Fresa Portola.....	24
Figura 10 Compartimientos doble de batería .....	29
Figura 11 Cámara Zenmuse Z3 .....	31
Figura 12 Cámara multiespectral RedEdge.MX .....	31
Figura 13 Drone Mavic 2 Pro.....	33
Figura 14 Numpy.....	34
Figura 15 TensorFlow .....	35
Figura 16 Marco principal.....	36
Figura 17 Marco primarios y auxiliares.....	36
Figura 18 PyPI descargas por día .....	37
Figura 19 Artículos referenciados o mencionados de aprendizaje profundo indexado en Google Scholar por semana.....	37
Figura 20 Imagen original, imagen binarizada e imagen erosionada de radio 2 .....	40
Figura 21 Imagen de entrada, imagen binarizada e imagen dilatada.....	41
Figura 22 Imagen original e imagen con gradiente .....	41
Figura 23 Métodos de aproximación de contornos.....	43
Figura 24 imagen original en escala de grises e imagen con el filtro gaussiano .....	44
Figura 25 Esquema de una red neuronal artificial con tres capas.....	44
Figura 26 Representación de una red profunda .....	45
Figura 27 Sigmoideo .....	46
Figura 29 Capas de una red neuronal.....	49
Figura 30 Capas de las redes neuronales convolucionales .....	52
Figura 31 Reducción matricial con Maxpooling .....	52
Figura 32 Capa de Softmax en una red neuronal.....	53
Figura 33 Red neuronal estándar vs red neuronal con dropout .....	54
Figura 34 Overfitting y Underfitting.....	54
Figura 35 Etapas del proyecto.....	61
Figura 36 El círculo amarillo encierra la zona Monte Adentro donde se adquirió una parte de la data .....	65
Figura 37 El círculo amarillo encierra la vereda el Totumo donde se adquirió la otra parte de la data.....	66

Figura 38 Trayectoria de vuelo en forma de cuadrícula .....	67
Figura 39 misiones disponibles de la aplicación pix4Capture .....	69
Figura 40: Características de vuelo en el sector de Monte Adentro .....	70
Figura 41. Visión del área desde la aplicación móvil.....	70
Figura 42 Preprocesamiento de imágenes para zona de interés 1 .....	73
Figura 43 Preprocesamiento de imágenes para zona de interés 2 .....	73
Figura 44 Imagen original de la data. ....	74
Figura 45 Imagen suavizada y filtrada (filtro gaussiano) .....	75
Figura 46 El contorno verde encierra la primera zona de interés .....	76
Figura 47 Doble dilatación.....	77
Figura 48 Gradiente morfológico .....	77
Figura 49. Dibujo de contornos .....	77
Figura 50 Carpetas de entrenamiento y de validación. ....	78
Figura 51 Carpetas de las tres etiquetas.....	78
Figura 52 Capas de la red convolucional .....	82
Figura 53: Resultados de entrenamiento y validación de la mejor configuración con factor de confusión .....	88

## LISTA DE TABLAS.

Tabla 1. Relación entre áreas sembradas, cosechadas y producción de fresa y los años .....	25
Tabla 2. Especificaciones del Matrice 100 .....	30
Tabla 3. Comparación entre drones: Mavic 2 pro, Matrice 100 y Parrot Bebop 2 .....	61
Tabla 4. Clase Normal, Clase Antracnosis y Clase Pallidosis (Otras).....	71
Tabla 5. Entrenamiento y validación de la red de aprendizaje por transferencia .....	83
Tabla 6: Entrenamiento y validación de la red convolucional sin factor de confusión ....	85
Tabla 7.entrenamiento y validación con patrón de confusión. Fuente: Autor .....	86

## INTRODUCCIÓN.

En la actualidad los países a nivel mundial ven la importancia de tecnificar los procesos agroindustriales con el fin de generar mayor capital económico que permitan una sociedad más sostenible. Dada esta razón durante los procesos de producción implementan tecnologías desarrolladas como es la visión por computadora para el control de enfermedades y además el monitoreo de los cultivos. Estos cultivos optimizados y controlados permiten al agricultor obtener las ganancias esperadas. Por este constante desarrollo en las tecnologías nos vemos obligado a estar en la vanguardia para mejorar los procesos de producción de los campesinos del país. Este proyecto fue elaborado con el fin de implementar algoritmos de inteligencia artificial, que permitan desarrollar procesos de clasificación, para aplicaciones en el ámbito agrícola que logren detectar la enfermedad de Antracnosis en los cultivos de fresa, ya que dichos cultivos representan un porcentaje considerable en la economía de los campesinos del municipio de Pamplona.

Para realizar este proyecto, se organizó la temática por capítulo, estos capítulos describen paso a paso la elaboración del proyecto. En el primer capítulo se generó una base de datos fundamentales para la elaboración del proyecto que comienza con los referentes teóricos y luego con el estado del arte, este dio soporte para la elaboración del proyecto, segundo se describen todas las variables favorables para la adquisición de las imágenes, estas variables están relacionadas con las condiciones climáticas y las características del dispositivo de adquisición, en el tercer capítulo se describe el plan de vuelo realizado en las zonas de monte adentro y el totumo, donde se establece la ruta de adquisición del dron por medio de la aplicación pix4Dcapture, en este aplicativo se establecen la altura, la velocidad, la misión de vuelo en forma de polígono y por ende la totalidad de imágenes capturadas para el preprocesamiento. En el cuarto capítulo se realizó el preprocesamiento por medio de las funciones de OpenCV, comenzando con el acondicionamiento del entorno instalando las librerías de opencv, numpy, Python 3.7.9, TensorFlow y Keras, seguido se cargó las imágenes con una función llamada get\_data1 que recibe como atributo la dirección local donde se encuentra la data, esta data fue ajustada y luego pasada por tres filtros gaussianos y por un proceso de morfología como son la dilatación y el gradiente morfológico, permitiendo definir la primera zona de interés (plástico envolvente) por medio de contornos utilizando las funciones de cv2.findContours, luego se calculó los máximos y mínimos del arreglo de contorno para el recorte de la primera zona de interés.

Este mismo proceso se desarrolló para extraer la segunda zona de interés (arbustos individuales de fresa), la diferencia radica en el método de extracción de contornos que para la primera zona fue cv2.RETR\_EXTERNAL y para la segunda zona fue cv2.RETR\_LIST. Una vez realizado los recortes se redimensionan todas las imágenes

con la función `get_data`, esta función redimensiona las imágenes a un tamaño de 400x400.

Una vez redimensionada las imágenes se clasifican de manera manual las clases normal, antracnosis y pallidosis (factor de confusión) para el set de validación y entrenamiento de la red.

En el quinto capítulo se desarrollaron dos algoritmos de inteligencia artificial, el primer algoritmo fue una red neuronal convolucional que cuenta con 5 capas convolucionales que extraen las características más importantes de las imágenes y que además reduce de manera muy importante las neuronas de entrada, estas características son proporcionadas a una red clásica que tiene como salida 3 neuronas para cada una de las clases. El segundo algoritmo es una red de aprendizaje por transferencia en donde se utiliza una red pre entrenado por medio de una aplicación llamada MobileNet2D de Google, en donde se creó una capa de maxpooling global que reduce al máximo la data de entrada, se genera una capa de salida con una neurona y con esto se estableció la cabeza para el modelo.

Luego se definió el modelo llamando al método `Keras.model.Model` al cual se le proporcionó los inputs que es igual al modelo base y outputs que son equivalente a la capa de salida del modelo MobileNet2D. Luego se compila el modelo utilizando el método `compile`, que tiene como optimizador `RMSprop` (0.0001) que para este tipo de aplicación es mejor que `adam`. Seguido se generó la normalización de las imágenes con la función `ImageDataGenerator` estableciéndole un reescalado de 1/255 para entrenamiento como validación, luego se establece por medio de la función `Flow_from_directory` la dirección donde se encuentra la data de entrenamiento y validación, además se estableció el tamaño de la imagen de 224x224. Luego de todo este proceso se entrena el modelo con la función `fit` asignándole el set de entrenamiento, el número de épocas=5 y por último el set de validación, por último, se evalúa el modelo con el método `evaluate`.

Durante este proceso se seleccionó la red convolucional ya que presentó mejor rendimiento en comparación a la red con aprendizaje por transferencias.

En el último capítulo se realizó la validación y el resultado de la red seleccionada, en el cual se hizo la comparación de la red convolucional con y sin factor de confusión, En donde se obtuvo excelentes resultados sin el factor de confusión arrojando como mejor resultado un porcentaje del 98.11% mientras que con el factor de confusión se generó un porcentaje aceptable obteniendo como resultado un 78.08%, siendo este porcentaje el mejor para este tipo de condición.

## PLANTEAMIENTO DEL PROBLEMA.

La fresa es un cultivo de alto valor económico nutricional y medicinal, que en todas sus etapas es susceptible al ataque de patógenos. El cultivo de fresa es atacado por una gran cantidad de enfermedades entre las cuales las más importantes son: Antracnosis, Mildeo polvoso, phytophthora fragarie, Verticillium, Rizotonia, Mycosphaerella y una de gran relevancia es Botrytis cinérea. La Botrytis cinérea es una de las enfermedades más peligrosas ya que afecta a flores y frutos y también afecta la post cosecha. Los productores tanto como los comercializadores se ven afectados por la insuficiente calidad de la fresa al presentar enfermedades como la Brotritys cinérea. Las pérdidas en el campo pueden estar cerca del 40% y en la post cosecha pueden estar alrededor del 85% [1]. Uno de los grandes retos para el campesino es manejar de manera eficiente las enfermedades, que en su gran mayoría son de carácter fungoso y, además enfermedades ocasionadas por bacterias, nematodos y en un porcentaje muy pequeño ocasionadas por virus. En el manejo y control de estas patologías utilizan ampliamente productos denominados pesticidas, los cuales generan altos riesgos para la salud del ser humano y para el medio ambiente, además afectando la inocuidad de la fresa e incrementando el valor de producción del cultivo [2]. Actualmente en pamplona uno de los grandes problemas es el uso inapropiado de productos como los plaguicidas y fertilizantes químicos que son proporcionados a los cultivos de fresa, también se desconoce los niveles de residuos de los pesticidas lo que ocasiona una contaminación en la fruta y en efecto producir enfermedades dado por este alimento [3]

Durante muchos años se viene utilizando funguicidas sintéticos para controlar los patógenos presentes en los cultivos de fresa, pero se ha demostrado que estos microorganismos se vuelven cada vez más resistentes a este tipo de productos. [2]En cultivos no tan tecnificados los campesinos realizan monitoreos semanales, para identificar de manera personal las plagas y enfermedades que estén afectando sus cultivos. Este monitoreo lo realizan dos o tres veces por semana recorriendo con atención e identificando en las plantas algún síntomas o daños presentes en la misma. Como no es posible realizar la observación de todas las plantas de determinado cultivo entonces, se realiza un muestreo a un número determinado de plantas del cultivo, entre más grande es el cultivo mayor será el muestreo que se realice [4].

El monitoreo requiere dedicación y entrenamiento. La persona que realiza el monitoreo debe familiarizarse y tener el conocimiento suficiente con respecto a las enfermedades presentes en el cultivo. Hay que tener presente que todas las enfermedades no son fáciles de identificar a simple vista y que necesitan un exhaustivo análisis para determinarlas [4]

La importancia económica que tiene la fresa a nivel nacional como internacional es fundamental para aportar al crecimiento de la economía de los países, de esta manera

es de vital interés controlar eficientemente todas las enfermedades que se puedan presentar en los cultivos con métodos más eficientes y de bajo costos.

## JUSTIFICACIÓN.

Uno de los grandes retos de la electrónica y de la computación es darle solución a problemas presentes en las actividades primordiales que permiten el desarrollo económico de un país. En Colombia la agricultura es una de las actividades que más ingresos económicos produce; por lo tanto, este proyecto va dirigido a dar solución o mejorar el problema de las enfermedades que se presentan en los cultivos, específicamente los cultivos de fresa, los cuales se ven afectados principalmente por una enfermedad conocida como antracnosis. El objetivo primordial es la identificación de determinada enfermedad en el momento oportuno con el fin de que no se vea afectado en su totalidad el cultivo. Al identificar la enfermedad estaremos logrando mitigar el uso inapropiado de funguicidas que afectan la salud del ser humano y que producen la contaminación de los suelos y del medio ambiente en general; además, logrando un gran beneficio para el sector agrícola, puesto que no tendrían grandes pérdidas gracias a esta tecnología, logrando así, mayores ganancias en la producción de cultivos. Esta identificación se da por medio del monitoreo constante por medio de una plataforma móvil autónoma con una cámara que le permite realizar capturas a cada una de las plantas en el cultivo, seguido de un procesamiento y por último el desarrollo de un método de visión artificial por medio del cual se va a identificar si presenta de manera cualitativa la enfermedad Antracnosis. Dado que los métodos artesanales de identificación son poco prácticos, esta tecnología permite que el campesino optimice los procesos de producción de la fresa logrando una mayor calidad y eficiencia en estos cultivos.

La importancia de la inteligencia artificial utilizando librerías de TensorFlow y Keras permite desarrollar algoritmos de alto nivel y esto se ve reflejado en la implantación que tiene empresas como Google en la búsqueda por voz, Facebook en el reconocimiento de personas y su recomendación de etiquetas, y además la Nasa que están implementando estos algoritmos en sus proyectos. No solamente estas empresas implementan estas librerías de inteligencia artificial, hay un sin número de empresas que ve el potencial de estos algoritmos y actualmente están implantándolos. La inteligencia artificial ha transformado muchas industrias desde la automotriz hasta lo más mínimo tiene inteligencia artificial. Por esta razón este proyecto tiene como listón la inteligencia artificial acompañada de técnicas de preprocesamiento de imágenes que con ayuda de un drone permita la adquisición de imágenes en los cultivos de fresa. Esto con el fin de crear una base de datos donde se clasifiquen las etiquetas de las enfermedades para luego entrenar red neuronal convolucional que cuenta con las mejores características para la clasificación de imágenes permitiendo identificar la presencia de antracnosis sin la importancia de parámetros de confusión.



## **OBJETIVO GENERAL.**

Desarrollar un sistema de procesamiento para la detección de antracnosis o mancha foliar en los cultivos de fresa.

## **OBJETIVOS ESPECÍFICOS.**

- Definir las variables favorables para un ambiente controlado que defina el protocolo para la adquisición de imágenes.
- Adquirir las imágenes de plantas sanas y plantas afectadas por Antracnosis para la construcción de una base de datos.
- Implementar un algoritmo de preprocesamiento a la base de datos que permita depurar la información adquirida.
- Implementar diferentes métodos de clasificación inteligente para la identificación de antracnosis en los cultivos de fresa.
- Validar la eficiencia del método de inteligencia artificial incluyendo factores de confusión.

# **CAPÍTULO I.**

## 1. REFERENTES TEÓRICOS.

### 1.1. FRESA O FRUTILLA.

Científicamente conocida como *fragaria Vulgaria sp. Fragaria chiloensis L.* Es una planta que se reproduce por medio de sus estolones (quienes forman las plantas hijas), por división de la corona y por la micro propagación, estos tres tipos de reproducción se conocen como vegetativa [5]. A nivel mundial existen diversas especies de frutillas; aunque no se sabe exactamente el origen de esta fruta, algunos científicos han llegado a la conclusión de que algunas especies de éstas, tienen orígenes en Europa, específicamente en los Alpes europeos y otras especies en Sur América en el país de Chile. [6]

En los cultivos de fresa, las plantas por lo general se ven afectadas por plagas y enfermedades que atacan todas las partes que las componen, por ello, es necesario saber e identificar que la fresa o frutilla se constituye por diversos órganos: Corona, sistema radicular, hojas, flores, frutos y estolones:

**1.1.1. Corona.** En la fresa o frutilla, la corona es un tallo corto, el cual contiene unos nudos de donde se originan las hojas, a su vez en la parte de las axilas se encuentran las yemas que dan origen a nuevas coronas, estolones o flores. Por lo general, el número de coronas que contenga una planta es directamente proporcional con la cantidad de frutos y flores que esta pueda producir, por tanto, si una planta se compone por un gran número de coronas, es mayor la cantidad de inflorescencias y frutos, pero a su vez, el tamaño de los mismos tiende a disminuir, por el contrario, si una planta tiene menor cantidad de coronas, menor será la cantidad de frutos y flores, pero mayor será el tamaño de estos. [7]

**1.1.2. Sistema radicular.** Compuesto por el conjunto de raíces que contiene la planta, el cual se origina a partir de la corona. Por lo general, gran parte de este sistema se encuentra en los primeros 15cm del suelo, pues la parte final o la profundidad de éste, dependerá de la influencia de diversos factores como lo son: la forma en que se cultiven, el clima, la clase de suelo y disponibilidad de nutrientes que haya en el suelo para la planta. [7]

**1.1.3. Hojas.** Se originan o se forman en los nudos de la corona, las características físicas de las hojas pueden variar en cada planta dependiendo de muchos factores que pueden influir; por lo que podremos encontrar plantas que posean hojas de un tinte claro o un poco más oscuro, también hojas gruesas o toscas o por el contrario delgadas y delicadas y finalmente hojas que sean de bordes lisos o bordes aserrados. Las hojas son vitales en una planta, por tanto, cualquier afección que

estás puedan tener provocarán efectos en cuanto al funcionamiento y rendimiento de la misma. [7]

**1.1.4. Flores y frutos.** Los frutos y flores se forman en cada corona de la frutilla. Las flores se componen o se integran por: un péndulo floral, sépalos, pétalos, pistilos y estambres. Los pistilos de la flor se hayan sobre un tejido conocido como receptáculo, el cual al crecer constituye la parte consumible de la fresa. La semilla de la frutilla la podemos encontrar en los aquenios los cuales son frutos secos superficiales que se encuentran en el receptáculo, estos son los encargados de producir reguladores de crecimiento que influirán en el desarrollo del receptáculo, por ello para que el fruto adquiera su forma y tamaño normal es importante que se desarrollen un gran número de aquenios, pues de lo contrario si estos no se desarrollan en gran cantidad puede conllevar a la deformación de los frutos. Por otro lado, cuando ocurre la polinización, el fruto empieza a desarrollarse, este es un proceso que puede verse afectado por factores externos como lo son las bajas temperaturas, la humedad, y principalmente las enfermedades y plagas. [7]

**1.1.5. Estolones.** Son órganos vegetativos, que, en la fresa, se forman a partir de las yemas axilares de las hojas, en condiciones específicas, el estolón consta de unos nudos que son los encargados de la formación nuevas plantas o plantas hijas. Cuando se forma la planta hija, ésta recibe los nutrientes y los líquidos vitales de la planta madre, hasta que en ella se desarrolla el sistema radicular lo cual permite que pueda ser independiente y absorber por sí misma los elementos vitales para su pervivencia. [7]



**Figura 1** Planta de fresa. **Fuente:** Hogarmanía.com, enlace: <https://www.hogarmania.com/jardineria/tecnicas/reproduccion/como-reproducir-fresas-26240.html>

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

---

En este orden de ideas, cabe resaltar que existen diferentes tipos de frutilla o fresa, las cuales se deben tener en cuenta para que los cultivos sean productivos y de calidad, pues no toda variedad prospera en todos los ambientes o climas, por lo que me permitiré mencionar las variedades de fresa principalmente cultivadas en Colombia y sus características importantes:

**1.1.6. Fresa camarosa.** Esta variedad de fresa es sensible a la antracnosis, aun así, es aclamada por los agricultores colombianos, puesto que tiene gran productividad y se adapta fácilmente a distintos climas, desde zonas subtropicales húmedas hasta zonas templadas. [6]



**Figura 2** fresa camarosa. **Fuente:** <https://lacuerva.com/producto/fresa-camarosa/>

**1.1.7. Fresa ventana.** Es una variedad sensible a las enfermedades del suelo. Es de clima frío por lo que solo se produce en zonas de esta temperatura, es bastante productiva y de buena calidad en comparación con la camarosa e incluso su color tanto interno como externo es un poco más claro que esta última, es un fruto grande que no suele sufrir deformaciones. [6]



**Figura 3** Fresa ventana. **Fuente:** <http://www.eurosemillas.com/es/variedades/fresa/item/5-ventana.html>

**1.1.8. Fresa camino real.** Esta variedad es interesante puesto que es tolerante a problemas de antracnosis. Su fruto es algo grande parecido al tamaño de la camarosa, pero con un color tanto interno como externo más oscuro, su planta es pequeña y fácil de tratar cuando ya está establecida puesto que sus etapas iniciales son de difícil manejo para los campesinos. [6]



**Figura 4** Fresa camino real. **Fuente:** <https://www.ppfam.com/FresaCamino%20Real/Mexico/49/>

**1.1.9. Fresa palomar.** Las plantas son pequeñas a comparación de la fresa ventana, por lo que son menos productivas por plantas que esta última (fresa ventana), pero, por otra parte, al ser de tamaño reducido, estas plantas pueden sembrarse en mayor cantidad por lo que podrían llegar a ser productivas. [6]



**Figura 5** Fresa palomar. **Fuente:** <https://ucanr.edu/blogs/alimentos/index.cfm?start=10&tagname=fresas>

**1.1.10. Fresa Albión.** Es resistente a la antracnosis. En Colombia es el tipo de fresa de crecimiento considerable, puesto que cubre grandes áreas sembradas, se

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

---

destaca por ser una variedad resistente a zonas o regiones entre los 2500 y 2800 metros sobre el nivel del mar. Su fruto es reconocido por ser de gran calidad, tanto en tamaño como en sabor e incluso porque sus producciones son periódicas durante la cosecha. [6]



**Figura 6** Fresa Albión. **Fuente:** <https://foro.infojardin.com/threads/plantas-de-fresas-variedad-albion.64609/>

**1.1.11. Fresa san Andreas.** Es una variedad resistente a las enfermedades, al igual que la fresa Albión, su fruto es de calidad tanto por su tamaño como su sabor. Su producción por ciclos es estable. [6]



**Figura 7** Fresa San Andreas. **Fuente:** <https://es.punchalo.com/1059-strawberry-san-andreas>

- 1.1.12. **Fresa Monterrey.** Su principal característica radica en que, a diferencia de la mayoría de frutillas, esta variedad es muy dulce por la poca acidez que contiene, esto la hace más llamativa para los consumidores. [6]



**Figura 8** Fresa Monterrey. **Fuente:** <http://www.eurosemillas.com/es/variedades/fresa/item/26-monterrey.html>

- 1.1.13. **Fresa portola.** Al ser demasiado resistente a las enfermedades no necesita de muchos cuidados a comparación de las otras variedades, es considerada como la variedad más productiva, al ser consistente, firme y de gran sabor. [6]



**Figura 9** Fresa Portola. **Fuente:** <https://es.punchalo.com/2413-strawberry-portola>



Estudios realizados por el ministerio de agricultura, demuestran que Colombia es el tercer país latinoamericano, que cuenta con mayor área sembrada de cultivos frutales, entre los cuales se destaca la fresa, fruto que se produce durante todo el año. La producción de fresa en Colombia tiene una influencia importante en cuanto a la economía nacional, pues es muy movido tanto en el mercado nacional como en el internacional en cuanto a las exportaciones que se realizan.

De acuerdo a los datos aportados por el ministerio de agricultura, en el año 2018 en Colombia se produjo aproximadamente 85.000 toneladas de fresas, de esas toneladas el 36% del fruto se dirigió a centrales de abastos, siendo las más destacadas: Corabastos en Bogotá (58%), Centroabastos en Bucaramanga (10%), Central mayorista de abastos de Medellín (10%) y Cenabastos de Cúcuta (4%). Se dice además que aproximadamente el 55% de la producción se comercializa en el fruto fresco, 25% se comercializa en pulpas y deshidratados y el otro 20% se comercializa en mermeladas y procesados.

En cuanto al comercio exterior, en 2015, las toneladas netas exportadas de fresa fueron de 101,33 toneladas, en 2016, 100,77 toneladas, 2017, 86,51 toneladas, 2018, 71,79 toneladas, y hasta el mes de junio de 2019, 11,00 toneladas. Siendo Panamá el principal país consumidor de la fresa producida en Colombia, seguido de Holanda y Curazao. [8]

### 1.1.14. Área y producción de la fresa.

Entre los años 2015 y 2018 el área sembrada en Colombia de cultivos de fresa tuvo un aumento considerable del 61,5%, lo cual equivale a 2,600 hectáreas cultivadas. Con dicho comportamiento anterior, se estimaba que para el año 2019 el área alcanzara un crecimiento de 3000 hectáreas. En la siguiente tabla se presenta la relación de las áreas cultivadas respecto los años:

	2015	2016	2017	2018	2019
Área sembrada (Has)	1,656	1,959	2,393	2,675	3,194
Área cosechada (Has)	1,503	1,623	1,960	2,161	2,472
Producción (Ton)	55,719	61,468	8,293	85,011	99,586

**Tabla 1** Relación entre áreas sembradas, cosechadas y producción de fresa y los años. **Fuente:**[8]

## 1.2. ANTRACNOSIS.

El colletotrichum es un grupo de hongos fitopatógenos que genera enfermedades tales como: cánceres, pudrición y principalmente antracnosis en diferentes plantas. Presenta lesiones semicirculares con hundimiento en forma de anillos concéntricos. Dentro de este gran grupo de hongos de colletotrichum, hay más de 100 especies encargadas de generar antracnosis en los diferentes cultivos que existen. Por otro lado, aparte del

colletotrichum, existen diversos tipos de hongos capaces de generar igualmente la antracnosis en las plantas, dichos hongos son: discula, septoria, venturia, phyllosticta y marssonina.

La antracnosis genera grandes pérdidas en el pre y postcosecha ya que reduce la calidad como la cantidad de los diversos cultivos. Las perdidas pueden llegar hasta un 100% dependiendo del patógeno y las condiciones ambientales.

De forma tradicional el grupo colletotrichum se le han identificado ciertas características morfológicas en donde se incluyen: el tamaño y forma de conidios; presencia y ausencia de setas, esclerocios, acérvulos, estado teleomorfo; así como características culturales como el color de la colonia, textura y tasa de crecimiento.

La antracnosis es una enfermedad que ocurre en las regiones productoras de fresa. La enfermedad ocurre esporádicamente y su importancia puede variar. En algunas temporadas, esta enfermedad es muy destructiva, resultando en la reducción de la productividad, fruta con lesiones que no pueden ser comercializada, e incluso mortalidad de la planta. La dureza de la antracnosis depende del grado en el que las plantas están contaminadas con el patógeno y la cantidad de riego por aspersion y de lluvia que cae sobre el cultivo. [9] La antracnosis en el cultivo de fresa produce daños en todos los órganos de la planta y es causada por tres especies de colletotrichum: *C.fragariae*, *C.acutatum*, *C.gloeosporioides*, estos hongos pueden infectar las hojas, pecíolos, flores y pedúnculos, donde se presentan manchas oscuras, circulares, inicialmente pequeñas, pero que pueden unirse de tal forma que generan lesiones de mayor tamaño. [10]

**1.2.1. En las hojas**, se produce una mancha color marrón oscura y seca, en los márgenes y puntas de las mismas, que puede extenderse hacia el centro de la hoja. La forma y tamaño de las manchas es variable, pero tienden a ser alargadas con bordes irregulares. Ocasionalmente pueden observarse pequeñas manchas circulares, de color gris o negras, que son los síntomas iniciales de antracnosis en las hojas. [7]

**1.2.2. En pecíolos y estolones** se manifiesta como manchas oscuras, secas, alargada, hundidas en el tejido vegetal, con una clara demarcación entre el tejido sano y el enfermo. Estas manchas pueden agrandarse hasta afectar todo el pecíolo o estolón, provocando la muerte de las hojas o de las puntas de guía. En condiciones de altas temperaturas y humedad se puede observar una coloración anaranjada en el centro de las manchas que corresponde a las esporas, producidas en las estructuras fructíferas del hongo denominadas acérvulos. [7]

**1.2.3. En la corona**, los síntomas se pueden apreciar realizando un corte longitudinal de la misma. Al inicio se observa una coloración anaranjada, con puntos marrones pequeños. Al progresar la infección, las áreas afectadas se tornan de un color

marrón rojizo, y pueden ser homogéneas o en estrías. La podredumbre es firme, no afecta el sistema vascular, y ocasiona el marchitamiento y la muerte de la planta. Si bien el color rojizo es distintivo de esta enfermedad, el diagnóstico de laboratorio permite diferenciarla de otros patógenos. [7]

**1.2.4. En el fruto,** produce manchas redondeadas, hundidas, firmes. Son de color marrón claro al inicio de la infección y puede permanecer así en condiciones de alta humedad. Posteriormente las manchas se oscurecen, llegando al color negro. Sobre las manchas puede observarse una coloración naranja, que corresponde a las esporas del hongo. El fruto puede ser afectado en su totalidad, siguiendo un proceso de deshidratación y momificación. usualmente ataca frutos maduros, pero en condiciones muy favorables al desarrollo de la enfermedad, pueden ocurrir ataques a frutos verdes. [7]

### **1.2.5. Ciclo.**

El hongo puede permanecer en el suelo y en los restos de hojas, estolones, frutos momificados y plantas de frutilla atacadas. Algunas malezas, así como otras especies de las familias de solanáceas, cucurbitáceas y rosáceas pueden también ser fuente de inóculo, por ser hospederas de algunas especies de *colletotrichum*.

A partir de esas fuentes de inóculo y en conjunción con condiciones ambientales favorables a la enfermedad, como alta temperatura y humedad, se da inicio a las infecciones en las hojas, los frutos y los estolones. A partir de esas infecciones, las esporas del hongo pueden alcanzar la corona de las plantas.

Altas temperaturas, mayores a 21°C, con un óptimo de 26-28°C en condiciones de alta humedad relativa, favorecen el desarrollo de la enfermedad y la esporulación del hongo. Las esporas pueden ser dispersadas por la lluvia y el viento, pero también pueden ser dispersadas por la mano de obra, en la ropa y calzado y por herramientas utilizadas para las tareas del cultivo. [7]

## **1.3. PALLIDOSIS.**

Esta enfermedad es originada por los virus SpaV y BPYV, que es transmitida por la mosca blanca y por otros virus no transmitidos por este animal.

Los síntomas que presenta se pueden confundir con la deficiencia nutricional de las plantas o también por trastornos abióticos. Esta enfermedad genera retraso en el crecimiento de las plantas, follaje rojo o púrpura y raíces frágiles. Además de la mosca blanca la enfermedad también es transmitida por virus como el *trialeurodes vaporariorrum* y por áfidos. Esta enfermedad no afecta a todos los cultivos sino a un número determinado de cultivos dentro de los cuales se encuentra el cultivo de fresa. [11]

### 1.4. DRONE.

Un drone es un objeto volador no tripulado capaz de ser manipulado a distancia o trazar su propia ruta mediante GPS.

Se puede diferenciar entre dos tipos de drones, uno es de forma de avión, los cuales tienen la ventaja del planeo por lo tanto tienen un consumo menor, y el otro en forma de cuadricóptero, propulsado por cuatro hélices y con la posibilidad de moverse en todas las direcciones.

Las partes básicas de un drone son las siguientes:

- Motores, hélices y ESCs que son componentes fundamentales para mantener el drone en el aire. Los ESCs (Electronic Speed Control) regulan la potencia eléctrica suministrada a los motores y por lo tanto la velocidad de giro del rotor, que al momento de girar a altas velocidades suspende el drone en el aire gracias a las hélices que se mueven sólidamente.
- Controlador de vuelo: Se trata del cerebro de la máquina, ya que detecta y controla todos los aspectos de esta. La mayoría de todos los componentes electrónicos van conectados al controlador de vuelo.
- Control remoto: Es un dispositivo con dos joysticks a través del cual se puede maniobrar el drone. [12]

**1.4.1. DJI MATRICE 100.** Es una plataforma de vuelo estable, flexible y potente. Cuenta con amplia variedad de aplicaciones en las áreas de investigación, negocios e incluso en la recreación, esto debido a que, cuenta con una plataforma abierta y un diseño que tiene todas las propiedades para ser personalizado.

El marco central de Matrice 100, cuenta con la cualidad de expandirse permitiendo la facilidad de monturas de componentes y dispositivos adicionales para lograr una mayor funcionalidad y efecto. Además, cuentan con una sección para adicionar una segunda batería de vuelo inteligente para lograr así un vuelo prolongado de hasta 40 minutos. [13]

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.



**Figura 10** Compartimientos doble de batería. **Fuente:** <https://www.dji.com/matrice100/features#sub-feature>

### 1.4.1.1. Especificaciones del Matrice 100.

<b>Estructura.</b>	
Distancia entre ejes diagonal.	650 mm
Peso (con batería TB47D).	2355 g
Peso (con batería TB48D).	2431 g
Peso máximo en el despegue.	3600 g
<b>Accesorios opcionales.</b>	
Peso del puerto de expansión	45 g
Peso del compartimento de la batería	160 g
Peso de la cámara y el estabilizador Zenmuse X3	247 g
<b>Rendimiento</b>	
Precisión de vuelo estacionario	Vertical: $\pm 0,5$ m, horizontal: 2,5 m
Velocidad angular máxima	Cabeceo: 300°/s, guiñada: 150°/s
Ángulo de cabeceo máximo	35°
Velocidad de ascenso máxima	5m/s
Velocidad de descenso máxima	4m/s
Resistencia máxima al viento	10m/s
Velocidad máxima	22m/s (modo ATTI, sin carga útil, sin viento) 17m/s (modo GPS, sin carga útil, sin viento)
Tiempo de vuelo estacionario (con batería TB47D)	Sin carga: 22min, 500g de carga útil: 17 min; 1 kg de carga: 13 min
Tiempo de vuelo estacionario (con batería TB48D)	Sin carga: 28min, 500g de carga útil: 20 min; 1 kg de carga: 16 min

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

Tiempo de vuelo estacionario (con dos baterías TB47D)	Sin carga: 33 min
Tiempo de vuelo estacionario (con dos baterías TB48D)	Sin carga: 40 min
Tiempo de vuelo estacionario (con batería TB47D y Zenmuse X3)	Sin carga: 19 min
Tiempo de vuelo estacionario (con batería TB48D y Zenmuse X3)	Sin carga: 23 min

<b>Sistema de propulsión.</b>	
Modelo de motor	DJI 3510
Modelo de hélice	DJI 1345s
Modelo de ESC	DJI E SERIES 620D
<b>Sistema de control de vuelo.</b>	
Modelo	N1
<b>Otro.</b>	
Temperatura de funcionamiento	-10 a 40 °C

<b>Cargador.</b>	
Modelo	A14-100P1A
Salida de tensión	26,3 V
Potencia nominal	100 W
<b>Batería (estándar).</b>	
Nombre	Batería de vuelo inteligente
Modelo	TB47D
Capacidad	4500 mAh

**Tabla 2** Especificaciones del Matrice 100. **Fuente:** [13]

**1.4.1.2. Cámara Zenmuse Z3.** Esta es una cámara aérea con zoom integrado que se basa en el modelo Zenmuse X3, este es el producto líder de DJI en su clase, optimizada para fotografía, posee un zoom de 7X creado mediante un zoom de 3.5X y un zoom digital sin pérdidas de 2X que logra una distancia focal de 22mm y 77mm, esto permite que sea ideal para aplicaciones industriales. Con respecto a la estabilidad posee un volante de inercia que añade un nivel más de control al control de viraje de la Z3. Su funcionamiento junto con el control de viraje normal, le permite tener movimientos más finos y con un excelente control.

Con respecto a la calidad, proporciona imágenes de buena resolución, ya que incorpora el sensor Sony altamente refinado de 1/2.3 pulgadas presente en el DJI Espire 1 2.0 y en el DJI Phantom 4, la calidad de la imagen es líder en su clase, las capturas instantáneas se pueden tomar con una resolución de 12 megapíxeles mientras que con la aplicación DNG RAW que permite la visión

avanzada de las imágenes, se pueden grabar videos de 30 fotogramas por segundos con una nitidez en 4K. [14]



**Figura 11** Cámara Zenmuse Z3. **Fuente:** <https://www.djioficial.com/zenmuse-z30-2/>

**1.4.1.3. Cámara multispectral RedEdge.MX.** La RedEdge.MX es un sensor multispectral profesional y resistente, que captura las bandas espectrales requeridas para el índice básico de la salud de los cultivos, además, las bandas adicionales necesarias para análisis más complejos. Este sensor, captura más datos en menor tiempo, lo que permite que los investigadores comprendan cómo reaccionan las diferentes razas en ciertos entornos y modificar las características que se desean para mejorar el rendimiento. También, captura una zona crítica del espectro de luz para el estudio de la salud de las plantas, la banda de orden rojo, es en esta banda donde se presentan los primeros signos de estrés.



**Figura 12** Cámara multispectral RedEdge.MX. **Fuente:** [13]

**1.4.2. DJI Mavic 2 Pro/Zoom.** Cuenta con un sistema de visión omnidireccional y sistemas de detección por infrarrojo. Gracias a las tecnologías de la firma DJI captura fotos complejas con facilidad, además, cuenta con detección de

obstáculos y con modos de vuelos inteligente como Hyperlapse. El Mavic 2 Pro cuenta con un estabilizador en sus ejes y una cámara con un sensor CMOS, desarrollada en forma colectiva entre DJI y Hasselblad, que permite grabar videos 4K, realizar fotos de 20 megapíxeles. Este dron de la empresa DJI plegable y portátil posee una cámara Hasselblad L1D-20C que fue creado exclusivamente para este dron con focal equivalente a 28mm.

Este dron permite que las fotografías tengan una silueta de color Dlog-M de 10 bits con el que se obtiene un rango dinámico más amplio, lo mismo que para video HDR con 10 bits en 4k lo que permite conectar el Mavic 2 pro a un televisor 4k con HLG logrando así obtener de inmediato el video con los tonos de color correctos. Esto logra que tenga un mayor rango dinámico y le permite tener una mayor flexibilidad al momento de la edición.

El Mavic pro logra mantenerse a una distancia de transmisión de video hasta un límite de 8km a una resolución de 1080p con una duración de hasta 31 min con cambios de frecuencia automático entre 2.4 y 5.8 GHz. Además, posee baja latencia equivalente a 120ms, con velocidad máxima de 72km/h en el modo sport y con tecnologías DJI que reducen el rudo generado.

El estabilizador en tres ejes del Mavic 2 pro permite realizar videos y capturas estables y fluidas en casi cualquier situación, siguiendo con las especificaciones de la cámara, este producto Hasselblad es conocido por su diseño ergonómico que gracias a tan arduo trabajo en colaboración con esta empresa y DJI, el Mavic 2 pro posee la a nueva tecnología de cámara Hasselblad L1D-20C que, gracias a la tecnología característica Hasselblad Natural colour solution(HNCS), ayuda a capturar increíbles fotos aéreas de 20 megapíxeles con detalles de color increíbles. Además, posee un sensor CMOS de 1 pulgada con una resolución de 20 megapíxeles con apertura ajustable de f/2.8 a f/11. Esta cámara admite enfoque automático, que puede enfocar desde 1m hasta el infinito, puede tomar fotos en modo de disparo como disparo único, disparo en ráfaga, intervalo, panorámica, slow Motion y HDR mejorado.

Con respecto al almacenamiento, cuenta con 8 GB de almacenamiento interno, además, admite tarjeta microSD para almacenar fotos y videos, una de las necesidades fundamentales de esta tarjeta micro SD es la velocidad UHS-I clase 3, ya que cuenta con una elevada velocidad de lectura y escritura, que requiere los datos de vides de alta resolución.

El Mavic 2 Pro, posee un peso de 907 gramos, con dimensiones en condiciones plegables de 214x91x84mm (largo por ancho por alto) y en condiciones desplegadas 322x242x84mm (largo por ancho por alto). [15]





**Figura 13** Drone Mavic 2 Pro. **Fuente:** <https://www.dji.com/mavic-2>

### 1.5. PYTHON.

Es un lenguaje de programación de alto nivel, que es utilizado para la programación de propósito general, este lenguaje fue creado por Guido Van Rossum y fue lanzado por primera vez en 1991. Python soporta múltiples paradigmas en programación, dentro de los cuales se encuentran: imperativos, orientado a objetos, programación funcional y estilos procesales; además cuenta con un sistema de tipo dinámico y gestión automática de memoria. Python dispone de una biblioteca estándar, enorme y completa.

Python está constituido con el propósito de incrementar, es decir, que se le pueden incorporar módulos y librerías según las necesidades actuales. Una de las características que se resalta de este lenguaje es que en su sintaxis no usa llaves ({ }), ni punto y comas (;) como la mayoría de lenguajes; no obstante Python, se basa en indentación para establecer su estructura. [16]

Actualmente este lenguaje cuenta con dos versiones en funcionamiento, estas son: Python 3.X y Python 2.X, donde la más utilizada actualmente es Python 3.X. Python 2.7 es compatible con versiones anteriores de Python y puede ejecutar códigos ya sea de la versión 1.X y 2.X; esta versión se considera obsoleta por los desarrolladores de CPython y sus actualizaciones solamente son seguridad y desarrollo de correcciones de errores. Estos desarrolladores de CPython tienen la intención de abandonar esta versión para el año 2020. Por su parte, Python 2.X después del 25 de junio de 2016 no genera nuevas versiones, pero las correcciones de errores, actualizaciones de seguridad, solo serán compatibles hasta el 2020. Python 3.X a diferencia de Python 2.X, no es compatible con versiones anteriores, además recibe nuevos desarrollos y características, es la versión de lenguaje con la que los desarrolladores se sienten más cómodos por tanto estos tienen la intención de continuar trabajando por medio de esta versión. En el presente proyecto será desarrollado utilizando la versión 3.8 de este lenguaje. [17]

Industrias como Google, Mozilla Firefox, Instagram, La NASA, La CIA, Disney, entre otras, utilizan en gran medida este lenguaje ya que es versátil y fácil de implementar, además permite llevar a cabo proyectos de machine learning como es el caso de este proyecto logrando así que este pueda aplicarse en diversos campos. [16] Dentro de Python podemos encontrar las siguientes librerías fundamentales:

**1.5.1. Numpy.** Es una librería del lenguaje de programación Python, que da soporte para crear vectores y matrices de múltiples dimensiones. Cuando realizamos cálculos con listas o vectores es fundamental utilizar arrays. Numpy posee estos módulos de arrays, además una cantidad considerable de métodos y funciones que se pueden aplicar a determinados arrays; Numpy al igual que el módulo math incluye funciones matemáticas básicas, no obstante, son más elaboradas y completas que incluyen ajuste lineal de funciones, entre otras utilidades más. [18]



**Figura 14** Numpy. **Fuente:** <https://en.m.wikipedia.org/wiki/NumPy>

Para utilizar las funciones de Numpy importamos el módulo de la siguiente manera:

```
Import numpy           #cargar el módulo numpy o bien
Import numpy as np     #Cargar el módulo numpy, con alias de np o bien
From numpy import      # cargar todas funciones de numpy [18]
```

**1.5.2. TensorFlow.** Es una librería desarrollada por Google para construir y entrenar redes neuronales, utilizada por determinada organización en áreas de investigación para generar sus propios productos. [16] Es una plataforma de código abierto de extremo a extremo para el aprendizaje automático, está constituida por un entorno integral y flexible de herramientas, bibliotecas y recursos de comunidad, permitiéndole a investigadores y desarrolladores llevar a cabo proyectos innovadores de aprendizaje automático, además de implementar y compilar de una manera fácil aplicaciones con tecnología de AA.

TensorFlow, nos permite entrenar e implementar fácilmente módulos en la nube como de manera local, navegadores o dispositivos sin importar el lenguaje a utilizar. En la compilación sencilla de módulos, se utilizan API intuitivas de alto nivel como Keras, emitiendo iteraciones de módulos de una forma inmediata con una depuración sencilla.

Cabe resaltar que, la arquitectura de TensorFlow es flexible, lo que permite que se puedan ejecutar en diferentes plataformas como son: GPUs, CPUs, OTPU, en ordenadores y también en dispositivos móviles. [16] Por ello, actualmente,

empresas como, Airbnb, Coca-Cola, Google, Intel, Twitter, Xiaomi, AMD, Lenovo, ebay, Uber, Paypal, nvidia, Snapchat y muchas más implementan TensorFlow.

Uno de los problemas que solucionan los desarrolladores en el mundo real, es la detección de enfermedades respiratorias mediante el aprendizaje automático. [19]



*Figura 15 TensorFlow. Fuente: <https://www.tensorflow.org/?hl=es-419>*

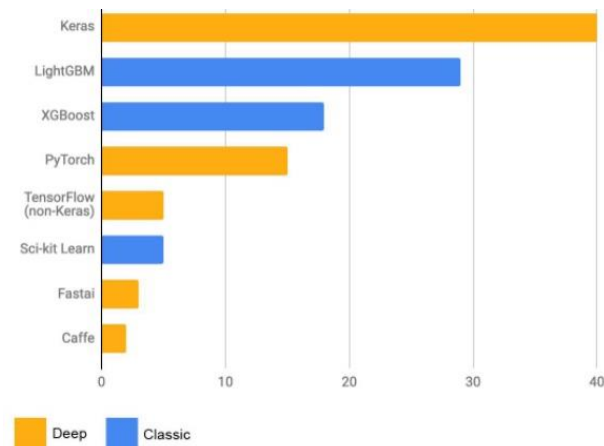
**1.5.3. Keras.** Es un API de aprendizaje profundo escrita en Python, que se ejecuta en plataformas de aprendizaje automático como TensorFlow. Keras fue desarrollada enfocándose principalmente en permitir una experimentación rápida.

Keras es un API diseñado para humanos, no para máquinas, que sigue las mejores prácticas para que los programadores no tengan un excesivo desgaste cognitivo. Ofrece API consistentes y simples, minimizando la cantidad de acciones que los programadores deben ejecutar. Esto permite que Keras sea fácil de aprender y utilizar, lo que conlleva a tener una mayor productividad.

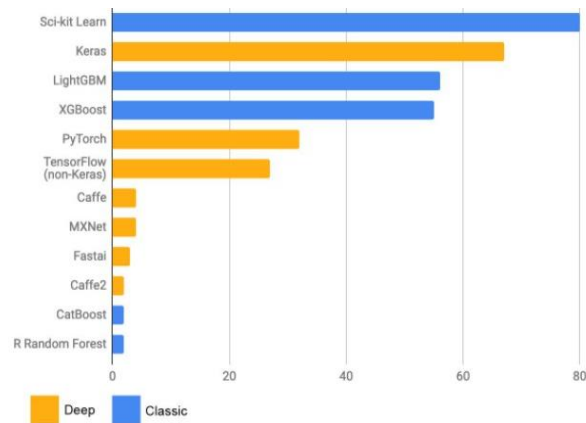
La facilidad de uso de Keras no se genera a costa de una flexibilidad reducida ya que Keras constituye profundamente las funcionalidades de TensorFlow de bajo nivel, lo que permite desarrollar flujos de trabajo altamente reciclables donde cualquier pieza se pueda personalizar.

El equipo de Keras en 2019 realizaron una encuesta en la cual terminaron entre los cinco (5) primeros en las competencias de Kaggle. Keras se ubicó como número 1 en aprendizaje profundo tanto entre los marcos primarios como entre los marcos utilizados. [20]

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.



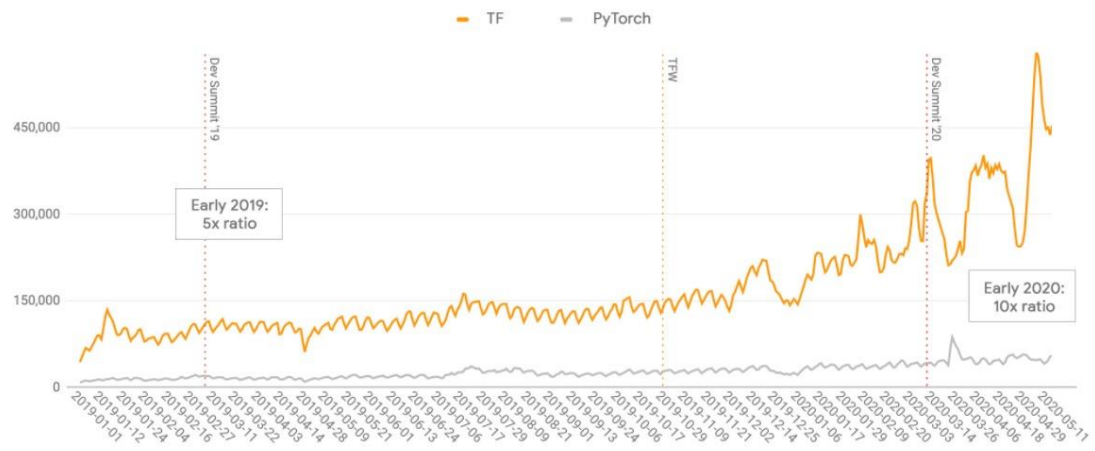
**Figura 16** Marco principal **Fuente:** [https://keras.io/why\\_keras/](https://keras.io/why_keras/)



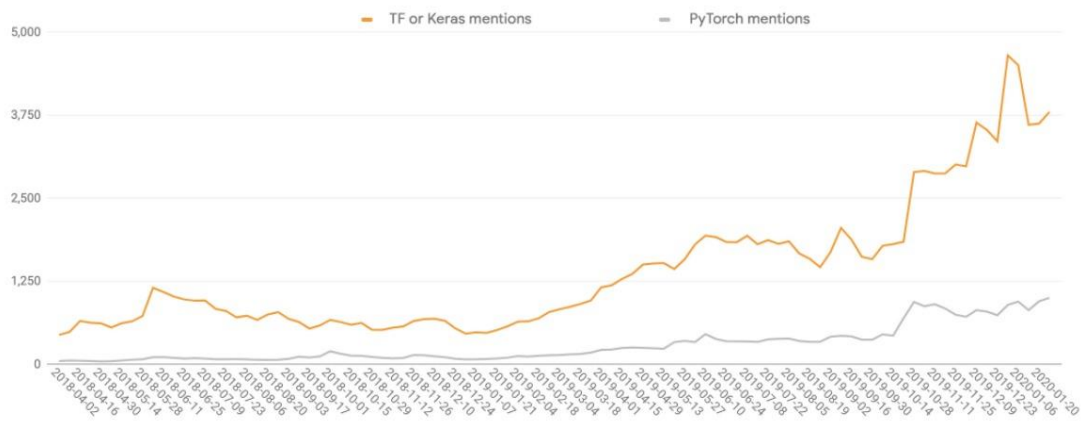
**Figura 17** Marco primarios y auxiliares. **Fuente:** [https://keras.io/why\\_keras/](https://keras.io/why_keras/)

Empresas como Netflix, Uber, Yelp, Instacart, Zocdoc, entre otros, están interactuando con las funciones con las que cuenta Keras, es bastante popular entre las nuevas empresas que implementa el aprendizaje profundo en sus productos. Al nivel de investigación Keras y TensorFlow también son favoritos y se encuentran en el puesto número uno (1) de referencias en artículos científicos según Google Scholar. Grandes organizaciones científicas también han adoptado a Keras para sus investigaciones, como es el caso de CRN y la NASA. [20]

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.



**Figura 18** PyPI descargas por día. Fuente: <http://keras.io>



**Figura 19** Artículos referenciados o mencionados de aprendizaje profundo indexado en Google Scholar por semana. Fuente: <http://keras.io>

Keras facilita la conversión de módulos en productos, estos módulos se pueden implementar de una manera fácil en un gran número de plataformas que cualquier otra API de aprendizaje profundo:

- En el servidor a través de un tiempo de ejecución de Python o de Node.js.
- En el servidor a través de TFX/TF Serving.
- En el navegador a través de TF.js.
- En Android o iOS a través de TF Lite o CoreML de Apple.
- En Raspberry Pi, Edge TPU u otro sistema integrado.

Al utilizar la API de Keras con TensorFlow es fundamental: primero, preparar los datos antes de entrenar convirtiéndolos en matrices, en nuestro caso, convirtiéndolos en objetos de Numpy como son Arrays; luego, se lleva a cabo el preprocesamiento de los datos, en este paso, se realiza la normalización de

características; en tercer lugar, se crea un modelo el cual convierta los datos en predicciones útiles implementando la API funcional de Keras, este modelo es entrenado con el módulo fit () que está integrado en Keras, teniendo en cuenta los factores como los puntos de control, el monitoreo de métricas y la tolerancia a fallas; después de entrenar el modelo, se evalúa dicho modelo con datos de prueba. [20]

**1.5.4. OpenCV (Open source computer visión library).** Es una librería de visión artificial y aprendizaje automático de código abierto. Fue creada para suministrar una infraestructura determinada, para aplicaciones de visión por computadora y para acelerar el tiempo utilizado por la máquina en los productos comerciales. Cuenta con licencia BSD, lo que conlleva que las empresas utilicen y modifiquen el código. Además, posee interfaces C++, Python, Java y MATLAB, es compatible con Windows, Linux, Android y Mac OS.

OpenCV cuenta con más de 2500 algoritmos optimizados, dentro de estos, se encuentran un conjunto íntegro de algoritmos de aprendizaje automático de visión por computadora de última generación. Dentro de las aplicaciones OpenCV puede utilizarse para la detección y reconocimiento de rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, buscar imágenes similares de una base de datos, entre otros.

Empresas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, utilizan esta biblioteca de una manera amplia. Los proyectos llevados a cabo con OpenCV van desde imágenes de streetview, detectar intrusiones en videos de vigilancia en Israel, monitorear equipos de minas en China, ayudar a los robots a navegar, detectar ahogamiento en piscinas en Europa, ejecutar arte interactivo en España y Nueva York, inspeccionar las etiquetas de productos en fábricas de todo el mundo, hasta la detección rápida de rostros. [21]

En el campo del procesamiento de imágenes OpenCV proporciona una infraestructura asequible y muy sencilla, cuenta con una estructura de cinco componentes principales:

- El componente de CV contiene los algoritmos de procesamiento de imágenes y de visión por ordenador en nivel básico y superior.
- ML es la biblioteca de aprendizaje de máquina, que incluye muchos clasificadores estadísticos y herramientas de agrupación.
- HighGUI contiene rutinas y funciones de I/O (entrada/salida) para el almacenamiento y carga de video e imágenes.
- CXCore contiene las estructuras básicas y algoritmos, apoyo XML, y funciones gráficas. Recibe información CV, MIL y HighGUI.

- CvAux, que contiene algunos algoritmos experimentales. [21]

**1.5.4.1. Segmentación por umbralización.** Se utiliza con el fin de establecer límites que permita decidir dentro de un rango de escala de grises, el conjunto de tonos de una región u objeto separando y extrayendo la región que nos interesa estudiar o analizar. Se pueden seleccionar uno o más umbrales, en el primer caso se obtiene una imagen binarizada y en el otro caso, una imagen clasificada. Las ventajas fundamentales de esta técnica de segmentación son su rapidez, el bajo costo computacional y que se puede desarrollar en tiempo real. Por lo contrario, las desventajas de esta técnica es que considera la intensidad de los pixeles, sin tener en cuenta la vecindad de los mismos.

La imagen segmentada  $g(x, y)$  por umbralización viene dada por:

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) > T \\ 0, & \text{if } f(x, y) \leq T \end{cases}$$

En OpenCV aplicamos la técnica de umbralización por medio de la función `CV::threshold`, esta función dispone de varios tipos de umbralización, para aplicarla, la imagen debe estar en escala de grises. [21]

**1.5.4.2. Procesos morfológicos.** La segmentación no brinda un resultado exacto para limitar los objetos de una región de interés. Dentro de esta clasificación por segmentación aparecen pixeles que no se clasifican adecuadamente, además, bordes imprecisos de los objetos o regiones que se están clasificando, por esta razón, antes de extraer características de medio nivel se requiere una etapa después del procesamiento; en dicha etapa se debe emplear el tratamiento morfológico. Esta es una técnica de procesamiento no lineal de la señal que se caracteriza en desarrollar la geometría y forma de los objetos, esta técnica se basa en fundamentos matemáticos, específicamente en la teoría de conjuntos. Estas herramientas se pueden utilizar en el procesado como en las etapas de segmentación y posprocesado o en etapas de mayor nivel de información visual. Algunas aplicaciones donde se pueden encontrar este tipo de herramientas son: en la restauración de imágenes, detección de bordes, análisis de textura, aumento de contraste y hasta en la compresión de imágenes. [22]

**1.5.4.2.1. Erosión binaria.** La transformación de la erosión es equivalente al resultado de verificar si el elemento estructurante  $B$  está completamente incluido dentro del conjunto  $X$  dado que no ocurra, el efecto de la erosión es el conjunto vacío presente en la siguiente fórmula:

$$\mathcal{E}_B (X)=X\ominus B= \{X| B_x \subseteq X\}$$

Si los objetos o pixeles en el caso de las imágenes sean menores que el elemento estructurante, este desaparece. Otra explicación de la erosión presume tomar valores de a imagen en el entorno por vecindad definido por el elemento estructurante. La utilizada de la erosión establecer una geometría determinada al elemento estructurante, para pasarla sobre la imagen.

Los objetos que se encuentran por debajo del elemento estructurante no se visualizan en la imagen resultante, por el contrario, los elementos que sean mayores al elemento estructurante se degradan. Por ende, la erosión supone una degradación de la imagen, lo que significa que la erosión es una transformación antiextensiva: [22]



**Figura 20** Imagen original, imagen binarizada e imagen erosionada de radio 2.  
Fuente: [22]

**1.5.4.2.2. Dilatación binaria.** La dilatación es una transformación contraria a la erosión. El resultado de la dilatación es el conjunto de elementos tal, que al menos un elemento del conjunto estructurante B está en el conjunto X, cuando B se mueve sobre el conjunto X.

$$\delta_B (X)=X\oplus B= \{x| X\cap B_x \neq \emptyset\}$$

Esta operación representa un crecimiento del conjunto X. Como elemento estructurante del conjunto, no se modificará. Sin embargo, en el límite del conjunto X al desplazar B, el conjunto resultante se expandirá.

La dilatación es una transformación extensiva:  $X \subseteq \delta_B (X)$



La dilatación también se interpreta como el valor máximo del entorno en las proximidades de vecindario definido por el elemento estructurante. [22]



Figura 21 Imagen de entrada, imagen binarizada e imagen dilatada. Fuente: [22]

**1.5.4.2.3. Gradiente morfológico.** El gradiente morfológico es la diferencia entre la dilatación y la erosión o también es la diferencia entre la imagen original y su erosión o entre la imagen original y la dilatación de una imagen.

- $g^- = X \setminus X \ominus B$  Gradiente por erosión
- $g^+ = X \oplus B \setminus X$  Gradiente por dilatación
- $g = (X \oplus B) \setminus (X \ominus B)$  Gradiente simétrico

Su utilidad está en localización de bordes o perímetros de los objetos, la selección del tipo de gradiente morfológico depende de la geometría de los objetos y además es útil para encontrar el contorno de un objeto como se refleja en la siguiente imagen. [22]



Figura 22 Imagen original e imagen con gradiente. Fuente: <https://sites.google.com/site/cg05procesamientodeimagenes/home/2-image-morphology-ii-morfologia-de-imagenes-ii>

**1.5.4.3. FindContours.** Los contornos expresan simplemente una curva que une todos los puntos continuos que poseen el mismo color e intensidad. Son una herramienta útil para analizar formas y para la detección y reconocimiento de objetos.

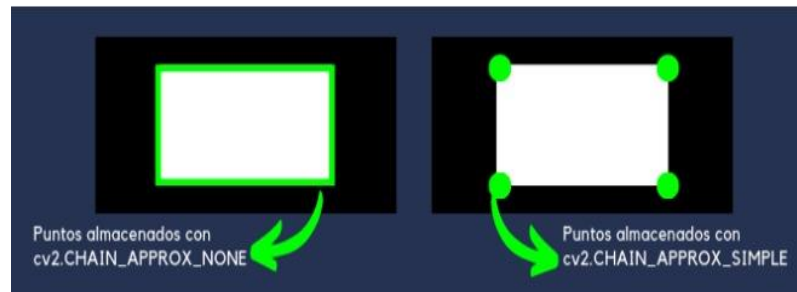
Para obtener una mayor precisión la imagen debe encontrarse binarizada. De la versión OpenCV 3.2 la función `findContours` ya no modifica la imagen de origen, por el contrario, devuelve una imagen modificada con el primero de los tres parámetros de retorno; en conclusión, encontrar contornos con OpenCV es como encontrar un objeto blanco sobre un fondo negro.

La jerarquía de contorno manifiesta la relación que tienen los contornos entre sí, podemos tener contornos dentro de otros contornos mayores y éstos dentro de otros contornos. En este caso el contorno externo se le atribuye como mayor, por ende, lo nombramos contorno padre, mientras que el contorno interno lo llamamos hijo. Existen varios métodos para recuperar los contornos presentes en una imagen:

- **cv2:RETR\_EXTERBAL:** Este argumento recupera los contornos externos, es decir, los mayores de la familia y el resto de contorno menores los omite.
- **cv2:RETR\_LIST:** Este argumento recupera todos los contornos sin establecer una jerarquía de padre e hijo.
- **cv2:RETR\_CCOMP:** Este argumento organiza los contornos en jerarquía de dos niveles padre e hijo, es decir, que, si tenemos en nuestra imagen un contorno externo, este tendrá jerarquía 1, mientras que, un contorno interno tendrá jerarquía 2; pero si este contorno a su vez tiene otro contorno, este tendrá una jerarquía 1.
- **cv2:RETR\_TREE:** Este argumento recupera todos los contornos con su respectiva jerarquía

También existen métodos de aproximación de contornos, éstos almacenan las coordenadas x e y, que corresponden a los contornos encontrados. Estos métodos los diferencia la cantidad de punto que almacena, estos métodos son:

- **cv2.CHAIN\_APPROX\_NONE:** Este método almacena todos los puntos del contorno.
  - **cv2.CHAIN\_APPROX\_SIMPLE:** Este método comprime los segmentos diagonales, horizontales y verticales, dejando solamente los puntos finales permitiendo así ahorrar memoria al no almacenar todos los puntos redundantes.
- [21]



**Figura 23** Métodos de aproximación de contornos. **Fuente:** <https://omes-va.com/contornos/>

**1.5.4.4. ¿Cómo dibujar los contornos?** Para dibujar los contornos hallados anteriormente con la función `findContours`, utilizamos la función `cv2.drawContours` lo que nos permite dibujar cualquier forma siempre y cuando tengamos los puntos límites. Esta función en su primer argumento le asignamos la imagen origen, en su segundo argumento, le proporcionamos la lista de contornos hallados anteriormente con la función `findContours`, en el tercer argumento le asignamos el índice de contornos, esto es útil cuando se dibuja un contorno de manera individual. Para dibujar todos los contornos le asignamos `-1` y los argumentos restantes son el color y grosor.

**1.5.4.5. cv2.cvtColor.** Esta función permite cambiar el espacio de color de una imagen. Ésta requiere tres parámetros, el primero la imagen original, el segundo, matriz destino y el último es el código de conversión. Este modelo de color es una codificación que permite representar los colores de una forma numérica. Hay más de 150 métodos de conversión de espacio de color disponibles en OpenCV, para realizar la conversión usamos la función `cv2.cvtColor(imput_image,flag)`.

Para la conversión de BGR a gray, utilizamos las banderas `cv2.COLOR_BGR2GRAY`, de igual forma se realiza la conversión de BGR a HSV y se utiliza la bandera `cv2.COLOR_BGR2HSV`, además para convertir de BGR a RGB se utiliza la bandera `cv2.COLOR_BGR2RGB`, entre otras conversiones. Para HSV el rango de tono es  $\{0,179\}$ , el rango de saturación es de  $\{0,255\}$ , y el rango de valor es de  $\{0,255\}$ .

**1.5.4.6. Desenfoque gaussiano.** Consiste en mezclar ligeramente los niveles de grises entre los píxeles adyacentes, lo que genera como resultado una imagen con los bordes más suaves en la que se pierde pequeños detalles similar al proceso que ocurre en las fotografías desenfocadas. Dicha transformación se realiza mediante la función `cv2.Gaussianblur` de OpenCV. [23]



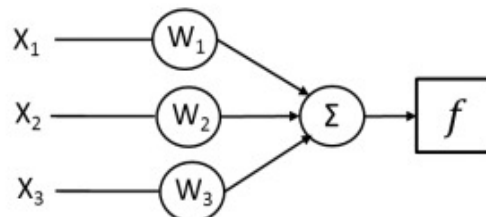
**Figura 24** imagen original en escala de grises e imagen con el filtro gaussiano. **Fuente:** <https://es.mathworks.com/help/images/ref/imgaussfilt.html>

### 1.6. INTELIGENCIA ARTIFICIAL.

La inteligencia artificial es una rama de las ciencias computacionales, encargadas del diseño y construcción de sistemas capaces de desarrollar tareas parecidas a la inteligencia humana. Las aplicaciones en la cual se ve reflejada la inteligencia artificial, es en el reconocimiento de imágenes o videos de objetos y personas, traductores automáticos de textos y hasta toma de decisiones. La inteligencia artificial se ha venido desarrollando aceleradamente gracias a la mayor disponibilidad de los datos, los recursos tecnológicos y financieros, así como en el avance de las técnicas de aprendizaje computacionales que es una de las principales áreas. [24]

#### 1.6.1. REDES NEURONALES.

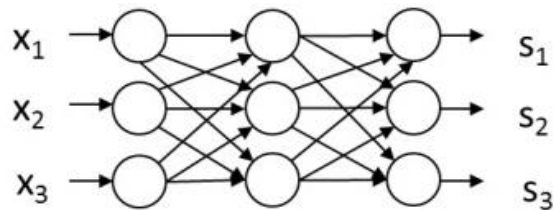
Las redes neuronales son paradigmas de aprendizaje automático, basado en el funcionamiento del cerebro biológico, estas redes están compuestas por neuronas que se interconectan entre sí, para generar una salida de datos con base a las entradas de la red. Cada neurona artificial, es una unidad de procesamiento que recibe como entrada señales y que luego se multiplica por un peso sináptico, esta neurona calcula la suma del producto de cada entrada por su peso sináptico al cual se le adhiere un factor de correlación llamado bias y al resultado de esta operación se le aplica una función de activación que produce un valor de salida dependiendo si la suma de las señales y pesos superan el umbral determinado. [23]



**Figura 25** Esquema de una red neuronal artificial con tres capas. **Fuente:** [23]

La organización de las redes neuronales se basa en capas donde cada capa procesa la información recibida de la capa anterior, la cantidad de capas y el tipo de función de activación de las neuronas determinan la complejidad de los problemas que puede solucionar la red, desde detectar patrones sencillos a relaciones complejas no lineales.

Las redes que poseen una o más capas intermedias entre la capa de entrada y la de salida se denominan, redes profundas y son la base de deep learning o como es conocido en español, aprendizaje profundo, la siguiente imagen representa una red neuronal profunda, ya que posee una capa oculta: [23]

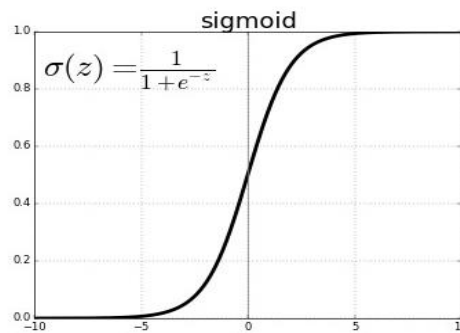


**Figura 26** Representación de una red profunda. **Fuente:** [23]

### 1.6.2. APRENDIZAJE PROFUNDO.

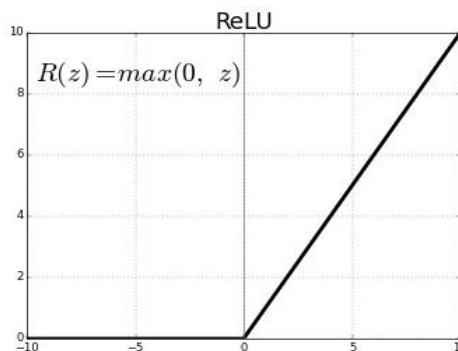
Es un área del aprendizaje automático que se caracteriza por el esfuerzo de crear módulos o capas de aprendizaje con varios niveles en donde los niveles superiores son alimentados por los niveles anteriores a los mismos. [25]. Conceptos del aprendizaje profundo:

- ✓ **Perceptrón:** Es una neurona artificial que toma varias entradas y realiza una suma ponderada para producir una salida. los pesos se calculan y ajustan durante el proceso de entrenamiento. [26]
- ✓ **Función de activación:** Se encarga de devolver una salida con respecto a una entrada, las funciones de activación la mayoría son funciones continuas y diferenciales excepto la unidad rectificadora en 0. Las funciones de activación son de gran importancia para el ajuste de los gradientes que nos permiten reducir el error en los sistemas de aprendizaje. Existen diferentes funciones de activación, las más utilizadas son las siguientes:
  - **Sigmoideo:** Es una función escalonada suavizada y por ende diferencial, es útil para convertir cualquier valor en probabilidades y se utiliza para clasificar datos binarios. La función sigmoidea mapea la entrada a un valor entre 0 a 1. [26] Las funciones sigmoideas por muchas décadas fueron la base de la mayoría de redes neuronales, aunque en los últimos años han perdido popularidad, ya que las redes neuronales de muchas capas se vuelven muy complejas de entrenar dado el problema de desaparición de gradientes. Actualmente la mayoría de redes neuronales usan otro tipo de función de activación llamado unidad lineal rectificadora (ReLU). [26]



**Figura 27** Sigmoideo. **Fuente:** [https://ml4a.github.io/ml4a/es/neural\\_networks/](https://ml4a.github.io/ml4a/es/neural_networks/)

- **Tangente hiperbólica:** Es parecida a la función sigmoideo ya que es escalonada y suavizada, la diferencia es que muestrea la entrada a valores entre -1 a 1. En esta función los gradientes son más estables que en la función sigmoidea y por ende tiene menos gradiente de fuga. [26]
- **La unidad lineal rectificadora (ReLU):** La función ReLU transforma los valores negativos en cero y las entradas positivas se emiten sin ningún cambio, deja pasar grandes números lo que produce que algunas neuronas se vuelvan obsoletas y no se activen. [26]



**Figura 28** ReLU. **Fuente:** [https://ml4a.github.io/ml4a/es/neural\\_networks/](https://ml4a.github.io/ml4a/es/neural_networks/)

### 1.6.3. APRENDIZAJE AUTOMÁTICO O MACHINE LEARNING.

El aprendizaje automático es un área de investigación en las ciencias de computación que se ocupa de los diferentes métodos que nos permiten identificar e implementar sistemas y algoritmos por medio de los cuales las computadoras pueden aprender a reconocer de manera automática patrones complejos y tomar decisiones lo más inteligentes posibles. Todo el proceso de aprendizaje está compuesto por un conjunto de datos de entrenamiento y un conjunto de datos de prueba. [25]

- El conjunto de entrenamiento: Es la base de datos que se utiliza para entrenar el algoritmo de aprendizaje automático. Durante esta fase los parámetros del modelo se pueden ajustar según el rendimiento obtenido.
- El conjunto de prueba: Esta base se utiliza para evaluar el rendimiento del modelo.

**Etiquetas.** Es el valor que se va a predecir. Las etiquetas pueden ser el precio futuro de la gasolina, en términos de clasificación de imágenes, por ejemplo: el tipo de enfermedad que se encuentra en una imagen, entre otras cosas.

**Atributos.** Un atributo es una variable de entrada que se expresan por lo valores de  $X$ . En proyectos de aprendizaje automático se podría usar un solo atributo o múltiples atributos dependiendo de su complejidad.

$$X_1, x_2, \dots X_N$$

**Modelos.** Un modelo define la relación entre el atributo y la etiqueta. Podemos destacar dos fases en el ciclo de un modelo:

- **Entrenamiento**, es donde se crea o aprende el modelo, en este punto le mostramos ejemplos etiquetados al modelo y este logra que aprenda gradualmente la relación que tiene sus atributos y etiquetas.
- **Inferencia**, en este caso le aplicamos al modelo entrenado ejemplos sin etiquetar, es decir, usamos el modelo entrenado para predicciones.
- **Regresión**, un modelo de regresión predice valores continuos. Los modelos de regresión hacen predicciones y responden a preguntas tales como: ¿Cuál es el valor de una casa en Pamplona?, ¿Cuál es la probabilidad que un usuario de clic un determinado anuncio?
- **Clasificación**, un modelo de clasificación predice valores discretos. Los modelos de clasificación responden a preguntas tales como: ¿Esta imagen es un perro o un gato?, ¿Un mensaje de correo electrónico, es spam o no lo es? [27]

Dentro del aprendizaje automático existen tres paradigmas de aprendizaje:

**1.6.3.1. Aprendizaje supervisado:** Es el trabajo de aprendizaje automático más sencillo y conocido. Se basa en una serie de elementos preclasificados, en los que se conoce a priori la categoría a la que pertenece cada elemento.

El conjunto consta de datos etiquetados, en donde tenemos los objetos y las clases asociadas. Los datos etiquetados conforman el conjunto de entrenamiento. La mayoría de los algoritmos de aprendizaje supervisado tiene una característica la cual se basa en que el entrenamiento se realiza mediante una minimización de la función de pérdida con el fin de disminuir el error de salida proporcionado por el sistema con respecto al resultado deseado. [25]

**1.6.3.2. Aprendizaje no supervisado:** Al igual que el aprendizaje supervisado se le proporciona los datos de entrada al sistema durante el periodo de entrenamiento, pero la diferencia es que en el aprendizaje no supervisado no se etiquetan los datos con clases.

**1.6.3.3. Aprendizaje reforzado:** Este tipo de aprendizaje se enfoca en la interacción con el entorno. El sistema adapta sus parámetros en función de la retroalimentación recibida del entorno, que luego proporciona una retroalimentación sobre las decisiones tomadas. [25]

**1.6.3.4. Aprendizaje reforzado:** Este tipo de aprendizaje se enfoca en la interacción con el entorno. El sistema adapta sus parámetros en función de la retroalimentación recibida del entorno, que luego proporciona una retroalimentación sobre las decisiones tomadas. [25]

Para resolver problemas de aprendizaje automático vamos a incluir una descripción general de alto nivel de trabajo que se basa en una serie de pasos para llevar a cabo el proyecto:

**Paso 1° recopilación de datos:** es uno de los pasos más importantes para resolver cualquier problema de aprendizaje automático supervisado. Estos son algunos aspectos importantes al momento de recopilar datos:

- Asegurar que la cantidad de muestra por clase no esté demasiado desequilibrada lo que quiere decir que, se debe tener un número comparable de muestra en cada clase.
- Asegurar que las muestras cubran de manera adecuada los espacios posibles de entradas.
- En el caso de utilizar API pública, tener en cuenta todas las limitaciones establecidas por esta antes de su utilización.

**Paso 2° explore sus datos:** entender las características de los datos a utilizar con anticipación nos permite elaborar un mejor modelo, generando así una mayor precisión en la clasificación, a parte, que podemos optimizar el entrenamiento de la red, además de consumir menos recursos computacionales al requerir menos datos.

**Paso 2.5 elige un modelo:** hasta este punto debemos tener el conjunto de datos optimizados con sus características claves, luego en función de las métricas que recopilamos en el paso anterior debemos seleccionar el modelo de clasificación, por ende nos preguntamos ¿cómo presentamos los datos a un algoritmo que recibe entradas numéricas?, esto sería el preprocesamiento y vectorización de los datos, además preguntarnos ¿qué tipo de modelo debemos usar y qué parámetros debemos usar para configurar nuestro modelo?.

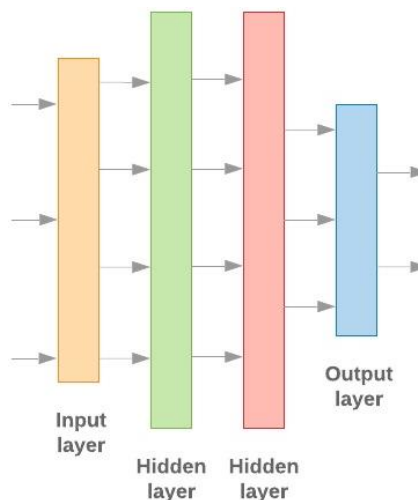
Gracias a numerosas investigaciones desarrolladas durante los años, poseemos una gran variedad de opciones de configuraciones de modelos y procesamiento de datos, lo que conlleva a incrementar la complejidad de elección de los diferentes métodos y modelos. Una de las formas de utilizar estos métodos es implementando todas las opciones y de esa manera por medio de la intuición utilizar el más adecuado.



**Paso 3° prepare sus datos:** es fundamental transformar los datos a un formato que el modelo pueda entender antes de enviarlo directamente al modelo, lo que conlleva a realizar primero una combinación de los datos antes de realizar cualquier agrupación de estos; en el caso de tener los datos divididos en conjuntos de validación y entrenamiento es necesario asegurarse de que los datos de validación se transformen de igual forma que los datos de entrenamiento. Es habitual utilizar un 80% de los datos para entrenamiento y un 20% de los datos para validación.

Hay que tener en cuenta, que los algoritmos de aprendizaje automático en sus entradas reciben números, por lo tanto, es necesario convertir cualquier tipo de datos diferentes a estos, con el fin de vectorizarlos de una forma numérica.

**Paso 4° construya, entrene y evalúe sus modelos:** La construcción de modelos de aprendizaje automático con Keras se trata de ensamblar las capas de neuronas de una forma semejante al ensamble de bloques LEGO, esto nos permite definir la secuencia de transformación de lo que queremos realizar de nuestras entradas. [27]



**Figura 29** Capas de una red neuronal. **Fuente:** <https://developers.google.com/machine-learning/guides/text-classification/step-3?hl=es-419>

Los datos en el caso de este proyecto, son imágenes que, son cargadas y luego preprocesadas con el fin de normalizar las imágenes en un rango de 0 a 1; donde también se ajustan las dimensiones de la matriz que conforman a esta.

Cabe resaltar, que los bloques básicos que conforman una red neuronal son las capas o layers, estas capas extraen representaciones del set de datos con el que se les alimentan. La mayoría de las capas que están conformadas por módulos de alto nivel de TensorFlow y Keras, tienen parámetros fundamentales durante el proceso de entrenamiento, un ejemplo de esto es: **tf.keras.layers.flatten** que transforman el formato de un arreglo

bidimensional a un arreglo unidimensional (aplanado) con el fin de proporcionárselo a las entradas ya que de una manera bidimensional no es posible.

Luego de establecer las capas necesarias con sus respectivos parámetros, procede el proceso de compilación del modelo, antes de esto es fundamental conocer algunas configuraciones durante este proceso:

- **Loss function:** que permite medir que tan exacto es el modelo durante el proceso de entrenamiento.
- **Optimizer:** nos permite que el modelo se actualice basado en el set de datos que adquiere y la función de pérdida.
- **Metrics:** que se utiliza para monitorear los pasos de entrenamiento y prueba.

Luego de tener estas configuraciones que cuentan como ejemplo, entrenamos el modelo. Para entrenar el modelo se requieren los siguientes pasos: Primero, entregamos los datos de entrenamiento al modelo; segundo, el modelo por sí mismo aprende a asociar imágenes con etiquetas; tercero, luego de asociar las etiquetas con sus respectivas entradas, verificamos que sean iguales y luego llamamos al método **model.fit**; para finalizar, se evalúa la exactitud del modelo donde se utiliza el método **model.evaluate** con sus respectivas configuraciones. [19]

**Paso 5° ajustes de los hiperparámetros:** se selecciona una serie de hiperparámetros para definir y entrenar el modelo, nos basamos en la intuición y recomendación de las mejores prácticas; tal vez, las elecciones de estos valores no generen el mejor resultado, pero nos proporciona un punto de partida para el entrenamiento. Dependiendo del problema a solucionar, ajustamos estos hiperparámetros que nos ayuda a depurar nuestro modelo para mejorar de manera particular el problema a tratar. Los siguientes son hiperparámetros que usamos y ajustamos dependiendo del caso en cuestión:

- **Primero. Número de capas en el modelo:** Este es un caso que nos indica la complejidad ya que debemos tener cuidado al elegir este valor. Demasiadas capas generan que el modelo aprenda demasiado sobre los datos de entrenamiento lo que producirá un ajuste excesivo.
- **Segundo. Número de unidades por capa:** Las unidades de una capa deben incluir la información para la transformación que realiza una capa. En el caso de la primera capa depende del número de entidades, por ejemplo: en el caso de una imagen que pasa por un proceso de aplanamiento sería el número de píxeles que tiene la imagen. En el caso de otras capas el número de unidades depende, de la expansión o reducción de las capas anteriores, por ejemplo: siguiendo con el ejemplo de matriz reducimos el número píxeles proporcionados a la entrada a la mitad o una reducción mayor.

- **Tercero. Tasa de abonado:** Se utiliza en el modelo para la regularización. Define la fracción de entrada que se eliminará por precaución por sobre ajuste. Intervalos recomendados por Google van de 0.2 de 0.5.
- **Cuarto. Tasa de aprendizaje:** Esta tasa a parte de números de iteraciones de la red neuronal va cambiando los pesos de sus neuronas. Una tasa de aprendizaje alta, puede generar cambios bruscos en los pesos y puede generar que nunca encontremos valores óptimos; a diferencia de una tasa de aprendizaje baja que es adecuada, ya que, solo se necesita un poco más de iteraciones para su convergencia. Si el entrenamiento es muy lento aumentamos la tasa de aprendizaje, en el caso en que el modelo no está aprendiendo se intenta disminuir la tasa de aprendizaje.

**Paso 6° implemente su modelo:** Hay que tener en cuenta, los siguientes ítems claves al implementar el modelo: primero, asegurarse que los datos generados sigan la misma distribución que los datos de entrenamiento y de evaluación; segundo, volver a evaluar el modelo de manera periódica mediante la adquisición de más datos de entrenamiento; por último, si la distribución de datos cambia hay que volver a capacitar el modelo. [27]

### 1.7. VISIÓN ARTIFICIAL.

Se define como un campo de la inteligencia artificial que a través de técnicas específicas permite la obtención, procesamiento y análisis de información derivado de imágenes digitales. La visión artificial abarca diferentes ramas del conocimiento como son la informática, la óptica, la mecánica y la automatización industrial. [28]

La visión por computadora es la ciencia que se encarga de comprender y manipular imágenes o videos. Tiene muchas aplicaciones en las cuales se encuentra la conducción automática, la inspección industrial, la identificación de enfermedades en plantas como en seres humanos y la realidad aumentada. [26]

### 1.8. REDES NEURONALES CONVOLUCIONALES.

La CNN, es un tipo de red neuronal que realiza un proceso con sus capas, parecida al cortex visual del cerebro humano con el fin de identificar características en sus entradas, este tipo de red posee varias capas ocultas especializadas y con una jerarquía determinada, lo que significa que las primeras capas identifican formas básicas y a medida que las capas son más profundas reconocen formas o patrones más complejos.

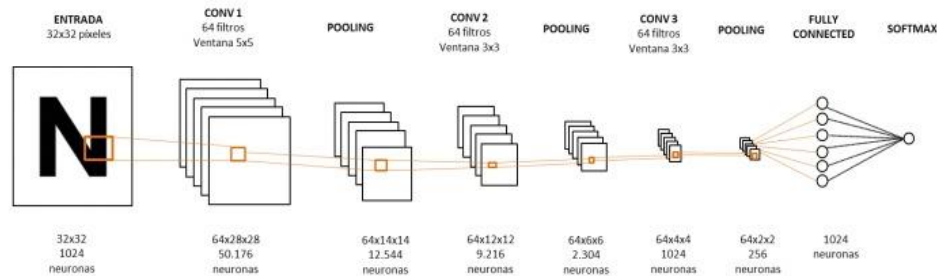
La red neuronal por su propia cuenta, debe reconocer una gran cantidad de imágenes para identificar características propias de cada objeto y poder así, generalizarlo. Las redes convolucionales como primera medida toman los pixeles de cada una de las

# DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

imágenes, constituidas por tres canales distintos: red, green y blue; generando así una imagen con los siguientes parámetros (width x height x 3).

Antes de alimentar a la red, se normalizan los valores en un rango de 0 a 1, esto se logra dividiendo cada pixel entre 255; luego de estos pasos, se realiza la operación de convolución, que consiste en tomar un grupo de pixeles del tamaño del kernel o filtro seleccionado, ese kernel recorre todas las neuronas de entrada, de izquierda a derecha y de arriba hacia abajo, generando una nueva matriz de salida conocida como matriz de activación a las imágenes no se le aplica un solo filtro, sino múltiples filtros para hallar características.

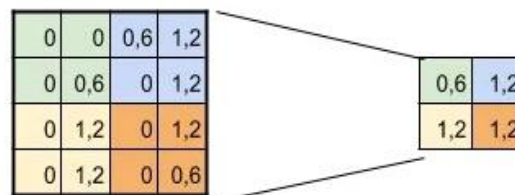
A medida que el filtro se desplaza sobre la imagen, obtenemos una imagen nueva a la cual se le aplica una función de activación, esta función devuelve una salida a partir de los valores de entrada. En conclusión, se busca funciones tales que sus derivadas sean simples para reducir el costo computacional. [29]



**Figura 30** Capas de las redes neuronales convolucionales. **Fuente:**

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/52222/7/fnunezsTFM0616memoria.pdf>

**1.8.1. Maxpooling.** El Maxpooling busca el valor máximo entre una ventana de muestra que sería el caso del filtro y este valor máximo es pasado como resumen de características del área analizada, el tamaño de los datos analizados se reduce a un factor equivalente al tamaño del filtro. Por ejemplo, un conjunto de 32 imágenes de un tamaño de 50x50 pixeles al cual se le realiza un maxpooling con un filtro 2x2, da como resultado una imagen que se reduce a la mitad, con un tamaño de 25x25 pixeles. Esto permite reducir el número de neuronas que necesitará la siguiente capa convolucional, en este ejemplo de 80000 neuronas se pasa a tener 20000 neuronas. [29]



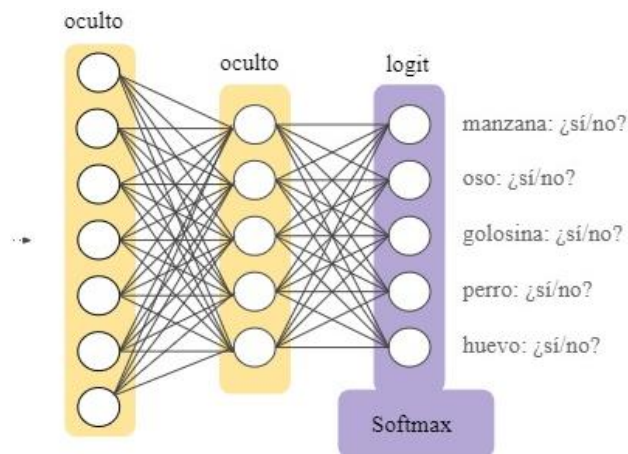
**Figura 31** Reducción matricial con Maxpooling. **Fuente:** [20]

**1.8.2. Softmax.** La función softmax asigna probabilidades decimales a cada clase en un caso de múltiples clases. La probabilidad decimal de esta clase debe sumar 1.0, eso logra que haya restricciones adicionales, logrando que el entrenamiento converja rápido. Softmax se implementa antes de la capa de resultado y debe tener la misma cantidad de nodos o neuronas que la capa de resultado.

$$P(Y = j|x) = \frac{e^{(w_j^T x + b_j)}}{\sum_{k \in K} e^{(w_k^T x + b_k)}}$$

El muestreo de candidatos, significa que Softmax realiza un cálculo de probabilidad para todas las etiquetas positivas mientras que para las muestras negativas toma un número aleatorio de etiquetas.

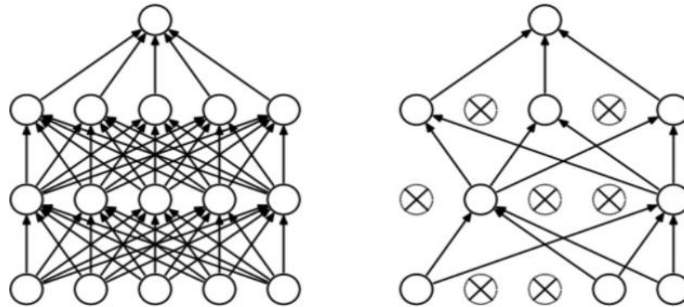
La función Softmax en conclusión es relativamente económica cuando la cantidad de clases es reducida, en cambio, cuando la cantidad de clase incrementa se vuelve demasiado costoso. [29]



**Figura 32** Capa de Softmax en una red neuronal. **Fuente:** <https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax?hl=es>

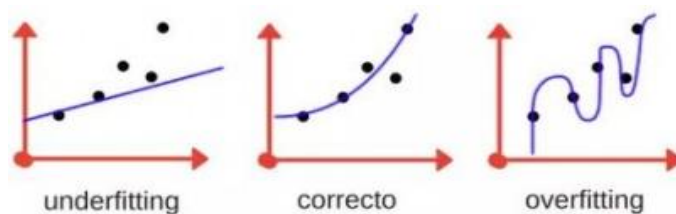
**1.8.3. Dropout.** Este método permite desactivar un número determinado de neuronas en nuestra red neuronal de forma aleatoria, estas neuronas desactivadas no se tienen en cuenta para la propagación en ambas direcciones, lo que obliga a las neuronas romper con la dependencia que tienen con estas neuronas desactivadas, este proceso ayuda al reducir el overfitting lo que obliga a las neuronas a trabajar mejor de forma autónoma para no depender tanto de las relaciones con las neuronas más cercanas.

Dropout cuenta con un parámetro que indica la probabilidad de que las neuronas permanezcan o no activas tomando valores de 0 a 1, entre más cercano a 0 estos valores, el Dropout desactivará menos neuronas que en el caso de que esté más cercanos a 1. Este método de Dropout solo se utiliza durante la fase de entrenamiento. [29]



**Figura 33** Red neuronal estándar vs red neuronal con dropout. **Fuente:** <https://towardsdatascience.com/machine-learning-part-20-dropout-keras-layers-explained-8c9f6dc4c9ab>.

**1.8.4. Overfitting y Underfitting.** Las causas fundamentales de obtener malos resultados en el machine learning se deben a estos dos parámetros, que hacen referencia al fallo en el modelo cuando se generaliza el conocimiento que se quiere que adquiera. Si los datos de entrenamiento son escasos, la máquina de machine learning será incapaz de generalizar el conocimiento y estará incurriendo en el Underfitting. En el caso contrario, si la máquina es entrenada con valores o imágenes demasiado específicas sobre lo que se quiere interpretar, esta será incapaz de comprender si se trata de una imagen u otro tipo de datos. Estas dos situaciones son negativas ya que la máquina no tiene la capacidad de generalizar el conocimiento y por ende las predicciones van a ser erradas. [29]



**Figura 34** Overfitting y Underfitting. **Fuente:** [20]

### 1.9. ESTADO DEL ARTE.

Los señores P. Constante, A. Gordon, O. Chang, E. Pruna, F. Acuna, y I. Escobar, creadores del artículo Técnicas de visión artificial para la clasificación industrial de la fresa, utilizaron redes Neuronales Artificiales entrenadas mediante aprendizaje compensado por ruido, que consiste en proporcionar durante ciertas fases del aprendizaje, ruido blanco, el cual les permitió realizar procesos de clasificación de acuerdo a parámetros visuales complejos y subjetivos, como lo son la forma de las células o la sinuosidad de las fresas, que es independiente de la posición o tamaño de la fruta bajo inspección.

Para la realización del proyecto utilizaron una serie de equipos fundamentales para realizar las capturas y por ende el procesamiento digital de señales. [30]

Primero se implementó un banco de prueba en cual se compuso de:

un dispositivo de iluminación el cual fue una lámpara fluorescente, una cámara web HD para la construcción de una base de datos, seguido de una computadora donde se realizó la base de datos de las imágenes y por último la ejecución del algoritmo de clasificación de fresa y posteriormente la visualización del resultado.

En la adquisición de imágenes utilizaron una base de datos de 50 imágenes con resolución de 250x150 pixeles en formato JPG, el cual utilizaron 40 para el entrenamiento y los 10 restantes para contraejemplo a la red.

La arquitectura neuronal profunda utilizada está motivada por la arquitectura profunda que se encuentra en el cerebro de los insectos donde existen dos componentes los cuales son el lóbulo Antenal y el cuerpo de hongo. En el lóbulo Antenal de la red, son pixeles de una región de interés que es de 100x100 pixeles. Esta región de interés contiene imágenes segmentadas con detección de bordes de Canny procedentes de la base de datos.

La red neuronal respectiva contiene 10000 entradas, 37 neuronas en la capa oculta y 22 neuronas de salida. Todos estos datos fueron seleccionados a ensayo y error, lo que quiere decir que solo se utilizó solo tres capas.

Las imágenes de pixeles binarios (0.1 y 0.9) son las que alimenta a la RNA con neuronas sigmoidales que son pre entrenadas con back propagation.

Por último, en el resultado se presentan tres tipos de clasificación dependiendo de la calidad de la fresa.

En la primera prueba realizada, el caso de las fresas de máxima calidad en donde se obtuvo un acierto para 37 imágenes de 40 imágenes de fresas con diferentes contornos y 10 fresas con algún tipo de deformación.

Para la segunda prueba, el caso de las fresas de consumo obtuvo un acierto de 36 imágenes. [30]

Y por último las fresas para materia prima industrial en donde definieron una nueva base de datos, añadiendo imágenes de fresa con contorno poco agradable al consumidor. En esta prueba se obtuvo un acierto de 36 imágenes.

Los señores Bhangé y Hingoliwala promotores del proyecto Smart Farming: Pomegranate Disease Detection Using Image Processing, proponen un sistema basado en una página web en donde los agricultores que cultivan la fruta granada realizan las capturas fotográficas a las plantas afectadas de sus cultivos y luego suben estas imágenes a la página web, esta se encarga de analizar y comprobar a través del modelo de inteligencia artificial si hay o no alguna enfermedad.

Primero elaboraron dos bases de datos de imágenes donde utilizaron una para entrenamiento y la otra para realizar pruebas. Las imágenes proporcionadas por los agricultores se les realizó un preprocesamiento para extraer características, el preprocesamiento constó de un redimensionamiento de las imágenes de un tamaño de 300 por 300px que posteriormente se les realizó la extracción de característica como son el color, morfología y CCV (vector de coherencia de color), luego por medio de k-means se agrupó y por último se realizó la clasificación con máquina de soporte vectorial dando como resultado una precisión del 85% con una cámara de 10 megapíxeles.[31]

En el siguiente artículo formulado por D. I. Patricio y R. Rieder, el objetivo fundamental de éste, es identificar la aplicabilidad o uso de la visión artificial en la agricultura de precisión para la producción de los cinco granos más importantes que se produce en el mundo (maíz, arroz, trigo, soja y cebada). Se presentaron 25 trabajos seleccionados en los últimos 5 años con diferentes enfoques con el fin de detectar en estas plantas las diferentes enfermedades, la calidad que presentan los granos y el fenotipo. Durante la metodología establecieron un protocolo para la identificación, evaluación e interpretación de los resultados que corresponde al proceso de investigación utilizando métodos explícitos y sistemáticos de análisis de los documentos extraídos de 5 bases de datos. En la investigación se utilizó un intervalo que se definió del 2013 al 2017 es decir 5 años.

De los trabajos seleccionados, la documentación se agrupó en tres grupos los cuales son: primero enfermedades y plaga, la siguiente es la calidad del grano y por último la fenología y fenotipado.

Comenzando con los trabajos realizados entre 2013 y 2017 el primero se basa en las enfermedades de los granos e infestaciones de insectos, debido a que los granos son susceptibles a una serie de patógenos, insectos y plagas. Estos factores tienen efectos negativos tanto en el desarrollo del cultivo y la reducción de la calidad del grano y, en consecuencia, el precio en el mercado. Para garantizar una agricultura productiva y sostenible, es importante la detección e identificación oportuna de las enfermedades en los granos. Debido a que los procesos de identificación se realizan visualmente el proceso se torna laborioso y subjetivo y por lo tanto el trabajo lo realizaron utilizando métodos de visión por computadora con inteligencia artificial para la respectiva automatización de los procesos de detección de enfermedades en estas plantas, para detectar si presenta o no determinada enfermedad. Aunque la detección de enfermedades en las plantas es un desafío ya que las imágenes obtenidas en el campo presentan fondos complejos y por lo



tanto presentan interferencia en las imágenes como las plantas de otras especies, el suelo, el reflejo entre otros.

Para la detección de BAKANAE en cultivos de arroz propusieron un método para clasificar plántulas de tres semanas de edad. Una de las claves de esta enfermedad es que afecta a las hojas de tal manera que presenta cambios de color, además produce alargamiento del tallo, retraso del crecimiento o un ángulo mayor entre hojas y el tallo.

El método que presentaron utilizó imágenes de plantas obtenidas con escáner.

En el preprocesamiento eliminaron el fondo mediante una operación de umbral y se obtuvieron atributos morfológicos y de coloración. Para el proceso de clasificación utilizaron máquinas de soporte vectorial en donde se utilizaron dos tipos de clasificación dispuestos en cascadas para detectar y determinar el estado de salud de la planta. El primer clasificador distingue entre plantas sanas y contaminadas y el otro clasificador determina el nivel de contaminación.

Otro artículo propone métodos de visión por computadora APTITUD para detectar enfermedades de las hojas en soya, en donde utilizaron métodos de visión por computadora para detectar y clasificar las enfermedades foliares de las plantas de soya. En este estudio analizaron diferentes factores dentro de los cuales están el WDH que es el histograma de color descompuesto de wavelet, otro es el BIC, que es un método que caracteriza cada pixel de la imagen como un borde, otro de estos métodos están el CCV que permite realizar comparaciones usando su densidad de color, entre otros factores fundamentales. Para la clasificación se analizaron tres métodos, el primero fue máquinas de soporte vectorial, el siguiente fue el k-Nearest neural network (KNN) y por último el PNN (probabilistic Neural Network). Como resultado pudieron destacar que el BIC, histograma y WDH presentaron mejores resultados en la detección de enfermedades de la hoja de soja mientras que otros métodos como el CDH Y LAP fueron más robustos para imágenes desenfocadas. Por último, los métodos de clasificación KNN y SVM obtuvieron mejor resultado.

En la detección de pulgones en TRIGO. Para la detección de estos insectos propusieron el método SMH (Support Vector Machine, Maximly Stable Extremely Regions e Histograms of Orientd Gradient Method), además el método combina el uso de máquina de soporte vectoriales, el algoritmo MSER (regiones extremas máximamente estables) y HOG (histograma de gradientes). Las imágenes que utilizaron se obtuvieron de una cámara convencional con un ángulo de posición lateral con respecto a la planta. En la parte de preprocesamiento se utilizaron filtros para reducir el ruido y utilizaron el algoritmo MSER para reducir el fondo de las imágenes. Al extraer los atributos de la imagen utilizaron HOG para ser utilizado en los parámetros para la clasificación usando máquinas de soporte vectorial para presencia de pulgones. El método SMH genero un rendimiento superior para la detección de este tipo de plaga en comparación a otros métodos utilizados.

En los descriptores locales para la detección de enfermedades de soja, propusieron un método que se basa en descriptores locales y el método BOV (bolsa de valores). El lugar

donde se obtuvieron las hojas fue Brasil, después de esta recolección las hojas fueron escaneadas para generar imágenes. El método de descriptores locales no requiere de segmentación para las imágenes porque demostró ser robusto. El método evaluó 5 descriptores diferentes dentro de los cuales se encuentra SURF (característica robusta acelerada), HOG, DSIFT (transformación de características invariantes a escala) e histogramas de palabras visuales (PHOW). Para la clasificación, definieron dos clases, enfermas y saludables, en el clasificador de la máquina de soporte vectorial.

En conclusión, el uso de técnicas de aprendizaje automático para clasificar imágenes se ha destacado en los últimos años con el fin de desarrollar sistemas de clasificación más precisos e inteligentes. Además, los sistemas de visión son una alternativa para todos esos agricultores que realizan la inspección de sus cultivos de manera manual. Así mismo los clasificadores automáticos pueden evitar errores humanos en el proceso ya sea de clasificación como de identificación de alguna enfermedad por lo tanto estos métodos son una alternativa a la inspección manual. [32]

El Trabajo de grado desarrollado por Daniel conde Ortiz, Inteligencia Artificial con TensorFlow para predicción de comportamientos, propone el desarrollo de una página web que cuenta con front-end y back-end logrando simular un servicio de streaming de música y en la que intentó recomendar al usuario los grupos que deberían gustarle mediante las acciones tomadas en página web.

Durante el desarrollo del proyecto seleccionó el tipo de red prealimentada con es el perceptrón multicapa, por sus características no lineales gracias a sus capas ocultas. En donde crea una capa de entrada, capas intermedias y capa de salida conectadas entre ellas.

Utilizó un modelo llamado codificador-decodificador, que permite primero transformar las entradas en un set de datos en el que se extrae la información más relevante y transformarlo de vuelta deduciendo esa información.

Para la base de datos tomo como referencia la API de Last.fm que posee una base de datos abierta al público, en el cual creó 10 grupos más escuchados. Luego, generó las solicitudes con 300000 usuarios, de los cuales se extrajo los 10 grupos que más escuchan, obteniendo así un fichero de 276083 usuarios x93579 grupos.

Para la creación de las etiquetas seleccionó 1 para el primer grupo y para el resto de grupo fue disminuyendo cada 0.06 hasta llegar al último grupo equivalente a 0.46.

Desarrolló una red neuronal con 27000 entradas con 10 grupos cada una, para la capa la primera capa oculta agregó 235 neuronas y para la segunda capa agregó 100 con función de activación sigmoide que es más relevante.

El entrenamiento duró aproximadamente 4 días con 150 épocas, en dónde posteriormente realizó el modelo predictivo codificador-decodificador con los valores de entrenamiento arrojados por red.[16]

Con respecto a la página web, esta contenía cuatro cajas en donde la primera

recomendaba 3 grupos por la red, la segunda un buscador de grupos y la tercera una selección de grupos aleatorios que genera con cada nueva página y un botón para refrescar.

El trabajo desarrollado por los autores Samir Kanaan Izquierdo Y Carles Ventura Royo llamado Diseño de un sistema de reconocimiento autónomo de matrículas de vehículos mediante una red neuronal convolucional. Proponen una red que adquiere imágenes de matrículas de vehículos con dimensiones de 230 x50 pixeles que pasan por un proceso de segmentación, el cual consistió en la detección de caracteres presentes en la placa, esta detección se basa en encontrar contornos utilizando la librería de opencv. Luego de esta etapa realizaron un proceso de reconocimiento de caracteres mediante una red neuronal convolucional, generando como salida una cadena de texto compuesta por los caracteres equivalentes a la placa.

Para el entrenamiento utilizaron una datasets que cuenta con 28.012 imágenes, para validación 3113 imágenes en formato PNG. Estas imágenes pasan por un proceso de conversión a escala de grises, desenfoque gaussiano, técnica de umbralización con la función thresholding y por último la detección de contornos con función findContours y boundingRect de opencv para el recorte de la imagen.

Para la primera capa convolucional utilizaron un numero de neuronas de 32x32 equivalentes a 1024 neuronas para la primera capa, luego generan 3 capas convolucionales con una capa de pooling. Finalmente, en las capas de la red clásica implementan 1024 neuronas utilizando la función softmax para calcular la probabilidad de que la imagen pertenezca a cada uno de los 36 caracteres posibles.

Resultados según tres colecciones de placas de matrículas nacionales, matriculas estadounidenses y matriculas reales. En donde obtuvieron 100 imágenes acertadas de 100 imágenes totales en la primera colección, en la segunda colección obtuvieron 60 imágenes acertadas de 100 y, por último, en la última colección obtuvieron 10 imágenes acertadas de 10.

Obteniendo así un porcentaje de validación del 97.83% y para entrenamiento del 99% con todo el set de validación.[23]

## **CAPÍTULO II.**

2. VARIABLES FAVORABLES PARA UN AMBIENTE CONTROLADO QUE DEFINA EL PROTOCOLO PARA LA ADQUISICIÓN DE IMÁGENES.

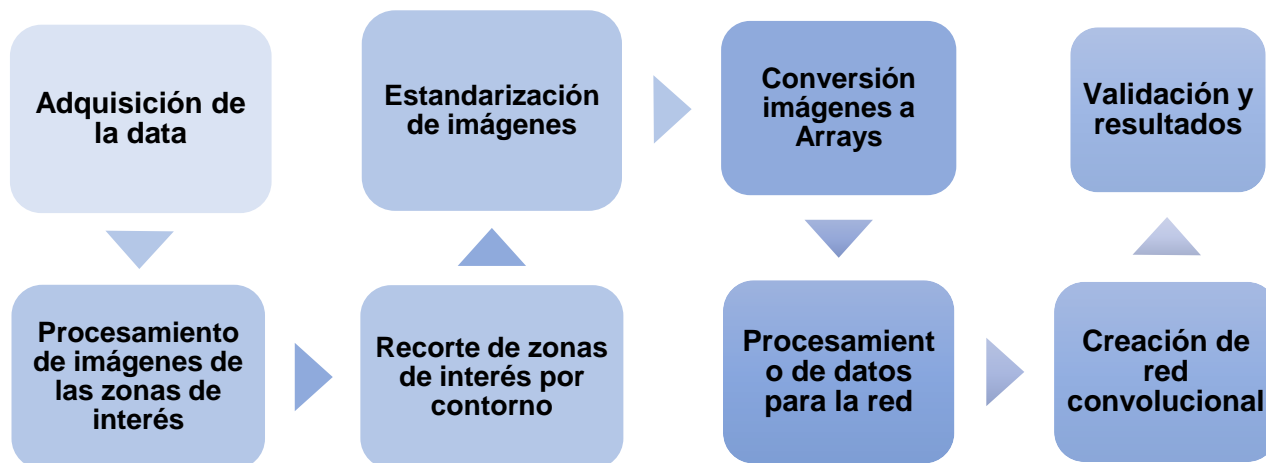


Figura 35 Etapas del proyecto. Fuente: autor.

Previo a la ejecución del proyecto se realizó la consulta de los diferentes drones que actualmente se encuentran activos en el mercado y que presentaban las características apropiadas para dicho proyecto, dentro de las posibilidades se encontraron los siguientes:

CARACTERÍSTICAS	DJI MAVIC 2 PRO	DJI MATRICE 100	PARROT BEBOP 2
<b>Resolución cámara</b>	20 megapíxeles	12 megapíxeles	14 megapíxeles
<b>Formato</b>	1080	1080p	1080p
<b>Autonomía</b>	Batería de 3800mAh (31 min de vuelo)	Batería 4500mAh (13 min con TB47D y 16 min con TB48D)	Batería de 2700mAh (25 min de vuelo)
<b>Almacenamiento interno</b>	8 GB de memoria	No	8 GB de memoria
<b>Antenas WiFi</b>	2,4 GHz o 5.8 GHz	2,4 GHz o 5 GHz	2,4 GHz o 5 GHz
<b>Potencia de carga máxima</b>	80W	180 W	42,18 W
<b>Alcance de la señal</b>	13km	5km	1.5km
<b>Velocidad de ascenso máximo.</b>	18km/h	5m/s	6m/s
<b>Velocidad máxima horizontal</b>	72km/h	22m/s	18m/s
<b>Precio</b>	1731 dólares	3300 dólares	655 dólares
<b>peso</b>	907g	2355g o 2431g	525g

Tabla 3 Comparación entre drones: Mavic 2 pro, Matrice 100 y Parrot Bebop 2. Fuente: [12][14]

Una vez vistas las características de cada drone, se optó por la opción del Mavic 2 pro, ya que este cuenta con una buena resolución para realizar las capturas de las imágenes permitiendo la redimensión de las imágenes sin perder características importantes de las mismas, en comparación con el Matrice 100 y el Parrot Bebop 2 que poseen cámaras con menor resolución, a parte es un dispositivo que cuenta con una baja latencia de 120ms y un alcance apto para el plan de vuelo estipulado, además con respecto al tiempo de duración de la batería el Mavic 2 pro posee mayor capacidad permitiendo que el drone se mantenga durante más tiempo en el aire. La practicidad que tiene el Mavic 2 Pro gracias a su tamaño y ligereza, le permiten funcionar a una menor potencia generando menos turbulencias o corrientes de aire que afecten al cultivo al estar en contacto con éste a una mínima distancia para la adquisición de la data, contrario al Matrice 100, que es un drone mucho más grande y por su tamaño y peso maltrataría los cultivos, por la turbulencia que genera y por la mínima distancia establecida en el protocolo de adquisición, además, se vería afectada su estabilidad por su tamaño y peso.

Por otro lado, en la selección de las cámaras se contó con 4 opciones: la HASSELBLAD L1D-20C, la Zenmuse Z3, la cámara multispectral RedEdge.MX y la cámara del Parrot Bebop 2, en donde la cámara del Mavic 2 pro, la HASSELBLAD L1D-20C calificó como la mejor opción ya que permitió capturar imágenes con una buena resolución en comparación con las cámaras RGB del Matrice 100 y del Parrot Bebop 2. Con respecto a la cámara multispectral con la que cuenta el Matrice 100 tiene excelentes características para la aplicación, pero los datos adquiridos por ésta son demasiados pesados para el entrenamiento de la red neuronal, que se alimenta de un gran número de imágenes, generando así con el formato de las imágenes multispectrales un mayor trabajo para el procesador del ordenador, que no cuenta con características avanzadas. Por esta razón se descartó la selección de la multispectral, dejando como opciones las tres cámaras RGB mencionadas anteriormente. Por las características que sobresalen de la HASSELBLAD

Al momento de tomar las capturas, el drone fue posicionado o ubicado de manera paralela a los surcos con el fin de que no presentara una inclinación considerable, que luego, pudiera afectar el recorte de la primera zona de interés, dicha zona de interés es el plástico que envuelve los surcos del cultivo de fresa. El drone además del posicionamiento de forma paralela a los surcos, se encontraba a una altura de entre 1.8 a 2.5 metros, logrando así realizar la captura por surco y una óptima visualización de los arbustos, que permitiera una clara obtención de las plantas sanas y enfermas, ya que al generar imágenes a una altura mayor, se pierden características importantes de antracnosis debido a que, las manchas generadas por la misma no posee áreas que se puedan visualizar después de cierta distancia, por esta razón se seleccionó la distancia límite del drone, cumpliendo con las condiciones establecidas en el preprocesamiento para los parámetros del código de recorte.

Con respecto a la luminosidad las imágenes fueron capturadas en horarios diurnos con cielo despejado con el fin de tener un rango de intensidad manejable. Estas capturas se realizaron dentro de un rango de temperatura de 13.02 y 21.85 grado centígrados, estas temperaturas fueron las que se presentaron durante el mes de septiembre en horarios de 9 am a 12 pm.

Las capturas se realizaron en horas de la mañana entre 9 am y 12 pm ya que además de la importancia que tiene la intensidad solar también se presentó un factor importante en las capturas de las imágenes, que fue la sombra generada por el dron entre las 12 pm y 3 pm, ya que entre estas horas los rayos del sol se encuentran perpendicular a la superficie y por ende se genera una sombra que podría afectar el procesamiento de las imágenes, esta sombra se posiciona por debajo del dron. Por esta razón se optó por realizar estas capturas en horas de la mañana ya que la sombra se genera con un ángulo mayor o menor de 90 grados.

También se encuentra un factor fundamental como es la velocidad del viento, esta velocidad limitaba entre 0.87 m/s velocidad del viento mínima y 2.89 m/s velocidad del viento máxima que se presentaron en el mes de septiembre, en estas fechas fue cuando se adquirió la data. Estas velocidades son despreciables en comparación a la velocidad que soporta el Mavic 2 pro el cual resiste velocidades entre 104.4 m/s y 136.8 m/s, por ende, el viento no fue impedimento para realizar la adquisición.

Este proceso de adquisición se vio interrumpido y por ende aplazado por las precipitaciones presentadas durante este mes, que fueron de 4.64 mm por día en promedio y que además el cielo presentaba una cantidad de nubes en horarios de la mañana de un 70% a nivel general en el mes, los pocos días soleados se aprovecharon para la adquisición durante cuatro días.

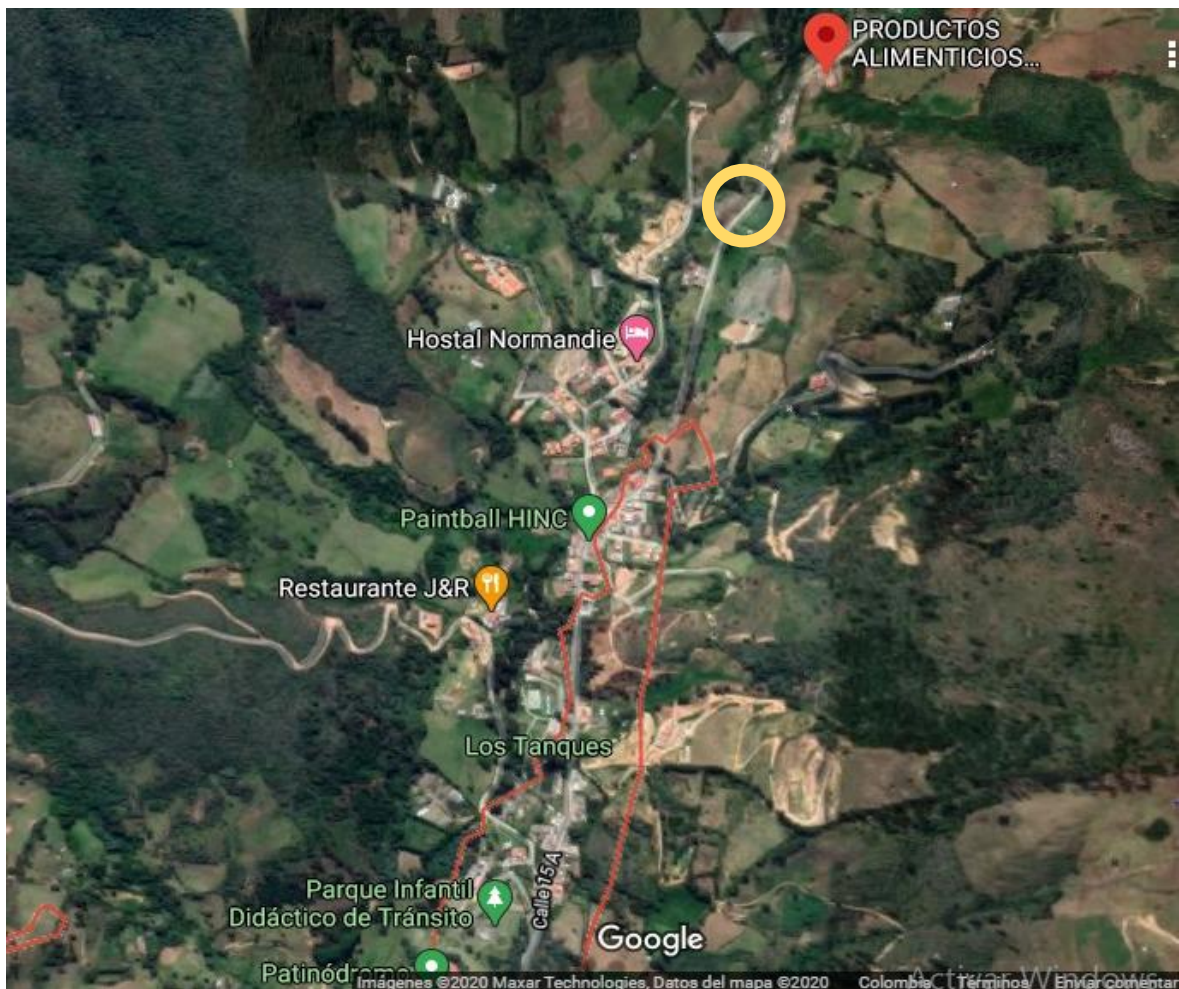
## **CAPÍTULO III.**

.



### 3. ADQUISICIÓN DE LAS IMÁGENES DE PLANTAS SANAS Y PLANTAS AFECTADAS POR ANTRACNOSIS PARA CONSTRUCCIÓN DE UNA BASE DE DATOS.

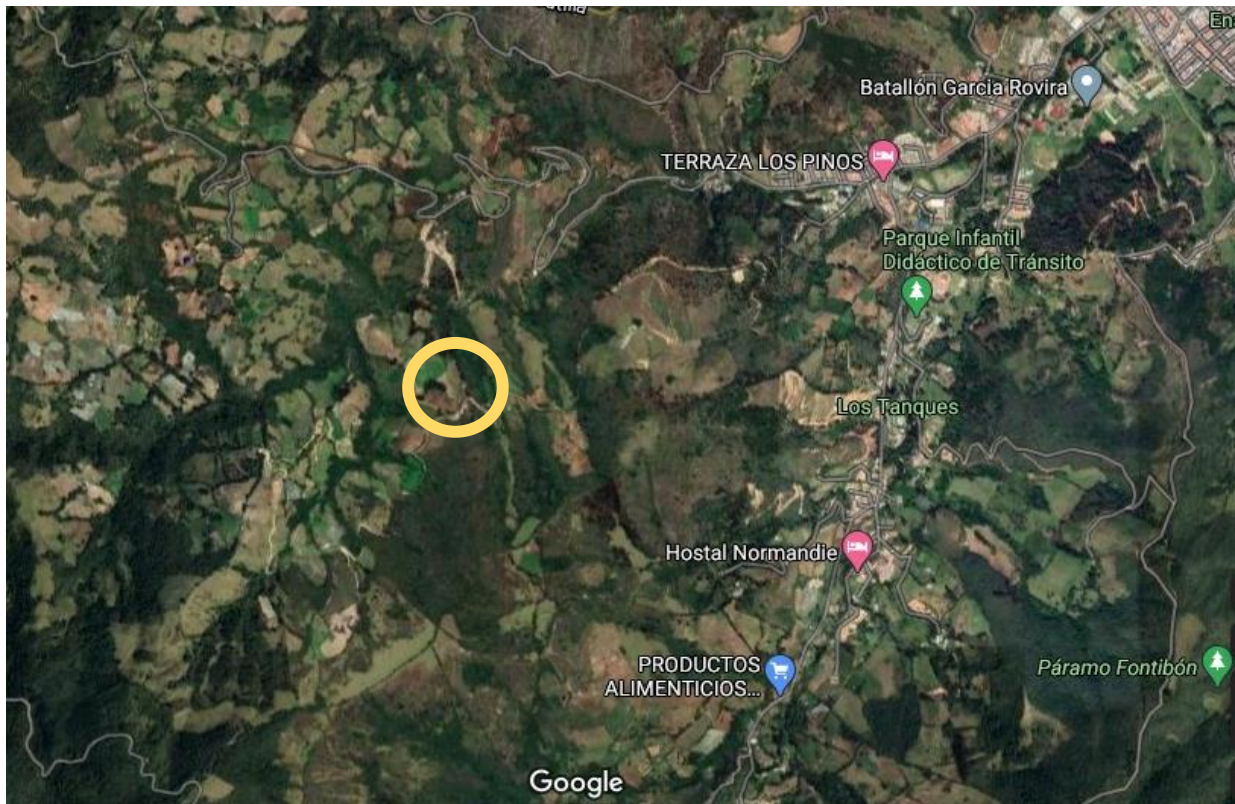
La data fue adquirida en dos zonas diferentes, ambas se encuentran en el municipio de Pamplona, la primera, es conocida como el sector de Monte Adentro, dicha zona se caracteriza por ser agrícola, en este lugar adquirimos la data de plantas sanas ya que cuentan con buenos protocolos en sus plantaciones; por otro lado, la segunda zona es una vereda llamada el Totumo en donde se adquirió la data de plantas enfermas con Antracnosis y otras enfermedades, ya que en dichas plantaciones no se implementan protocolos de sanidad adecuados.



**Figura 36** El círculo amarillo encierra la zona Monte Adentro donde se adquirió una parte de la data. **Fuente:** Google maps

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

Esta zona de Monte Adentro tiene facilidad de acceso por carretera, lo que permitió el traslado a dicha zona por medio de transporte terrestre, en este lugar se encontraron cultivos menores de una hectárea, con arbustos en etapa de cosecha y postcosecha. Se observó que los arbustos que estaban en etapa de postcosecha, específicamente después de la primera cosecha, presentaban síntomas de antracnosis de manera leve, mientras que, los arbustos que aún no habían generado su primera cosecha, no presentaban síntomas de antracnosis. Además, estos cultivos frecuentemente fueron fumigados con insecticidas y plaguicidas según las descripciones dadas por el propietario.



**Figura 37** El círculo amarillo encierra la vereda el Totumo donde se adquirió la otra parte de la data.  
**Fuente:** Google maps.

En esta otra zona, la vereda el Totumo, el acceso fue un poco complicado dado que, para llegar a ella, se debía acceder por un camino destapado y la única forma de hacerlo era caminando, el recorrido duró aproximadamente 2 horas desde el barrio los Tanques hasta llegar a la vereda donde se adquirió la data de las plantas enfermas con antracnosis. Al llegar se identificó que el cultivo estaba comprendido por un área menor de una hectárea, además, las plantas en su totalidad se encontraban en postcosecha y presentaban síntomas de antracnosis a un grado elevado principalmente en el follaje, en donde se notaba a simple vista la afectación de estos cultivos por dicha enfermedad.

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

Durante la estancia en el lugar, se conoció que a los cultivos no se les realizaba el control pertinente en cuanto a la aplicación de insecticidas y plaguicidas para evitar presencia de diversas enfermedades.

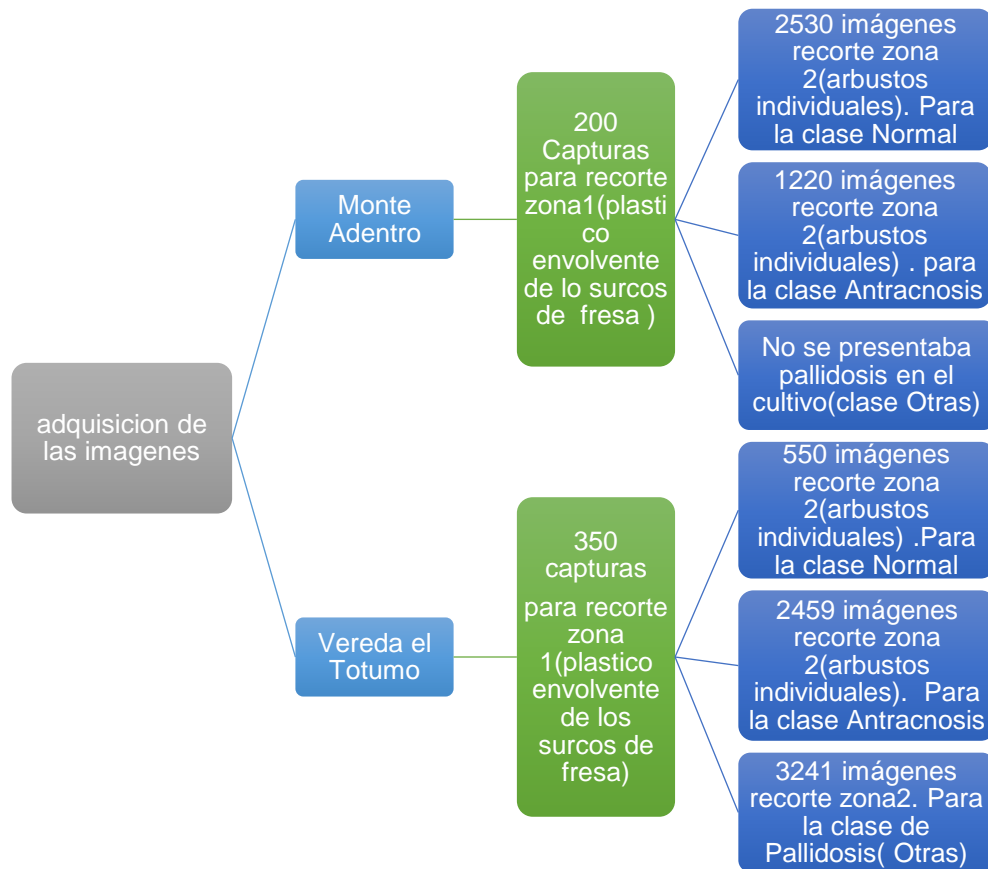


**Figura 38** Trayectoria de vuelo en forma de cuadrícula. **Fuente:** Autor

A cada uno de estos cultivos se le realizó un mismo plan de vuelo que consistió en realizar primero el despegue del dron en una superficie plana que permitió el libre ascenso del dispositivo, luego se llevó a cabo una trayectoria de vuelo de forma de polígono con un solape frontal del 80% y un solape lateral de un 75% permitiendo así, capturar mayor número de fotografías cubriendo por completo la zona de trabajo y con una posición de la cámara de manera ortogonal apuntado al suelo, además con altura en un rango de 1.8 a 2.5 metros obteniendo un tamaño del fotograma y de la zona mapeada adecuados. La velocidad de vuelo fue de 5m/s con una recurrencia de captura en un tiempo 0.4 segundo por fotograma con una distancia entre puntos de capturas de 2 metros, en la vereda el totumo que cuenta con un cultivo de aproximadamente media hectárea calculado así un número de capturas equivalente a 25 por pasada longitudinal adquiriendo una data aproximada de 350 imágenes de todos los surcos. En el caso de la vereda de monte adentro el cultivo tenía una menor dimensión de aproximadamente 3834 metros cuadrados, con la misma velocidad de 5m/s se adquirió las capturas a una distancia entre puntos de aproximadamente 2 metros con un tiempo de recurrencia de captura de 0.4, adquiriendo 14 fotograma por pasada longitudinal que generó una

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

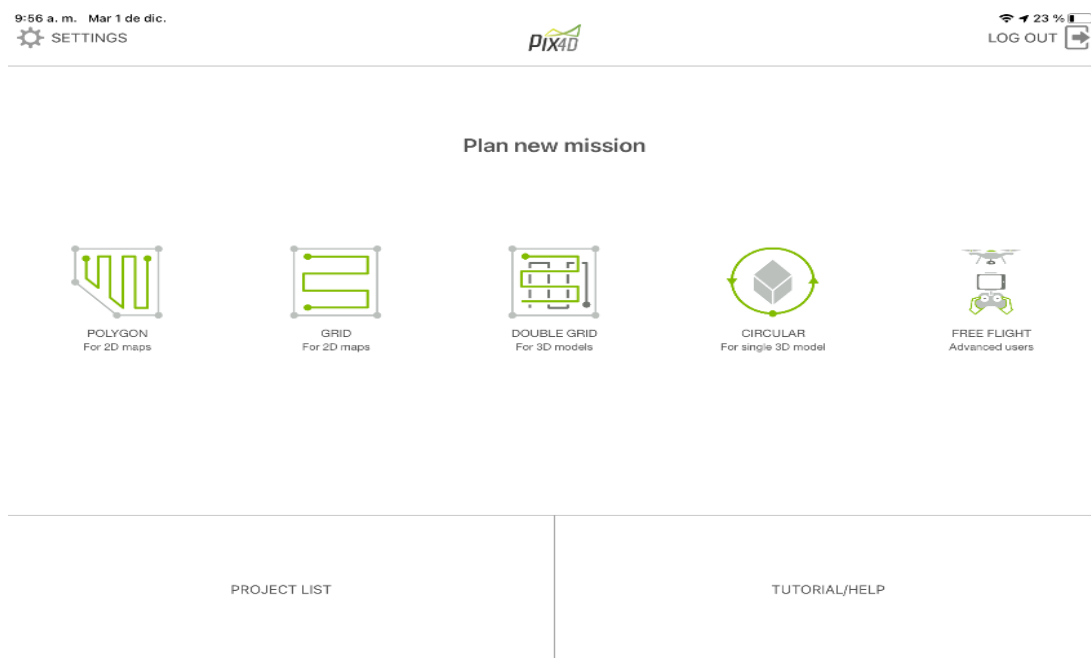
totalidad de 200 capturas de toda el área. Esta data fue adquirida en 4 días distintos en donde el primer día se recolectaron 110 capturas de los surcos de fresa en el sector de monte adentro, de estas imágenes se extrajo 1518 imágenes para la clase normal y 732 imágenes para la clase antracnosis, en el segundo día se recolectó el resto de la data, en este día se recolectaron 90 imágenes de los surcos y de estas imágenes se extrajo 1012 imágenes para la clase normal y 488 para la clase antracnosis. Para el tercer día la data se recolectó en la vereda del totumo en donde se adquirieron 140 imágenes de los surcos y de estas imágenes se extrajo 220 imágenes para la clase normal, 1476 imágenes para la clase antracnosis y para la clase “otras” equivalente al factor de confusión (pallidosis) se obtuvo 1296 imágenes. Por último, en el cuarto día se recolectaron 210 imágenes de los surcos y de estas imágenes se adquirieron 330 imágenes para la clase normal, 983 imágenes para la clase antracnosis y por último para la clase otras (pallidosis) se adquirieron 1945 imágenes.



De la anterior gráfica se puede observar que para la clase NORMAL se adquirieron 3080 imágenes en total, para la clase ANTRACNOSIS se adquirieron 3679 imágenes en total y para la clase PALLIDOSIS (otras) se recolectaron 3241 imágenes en total.

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

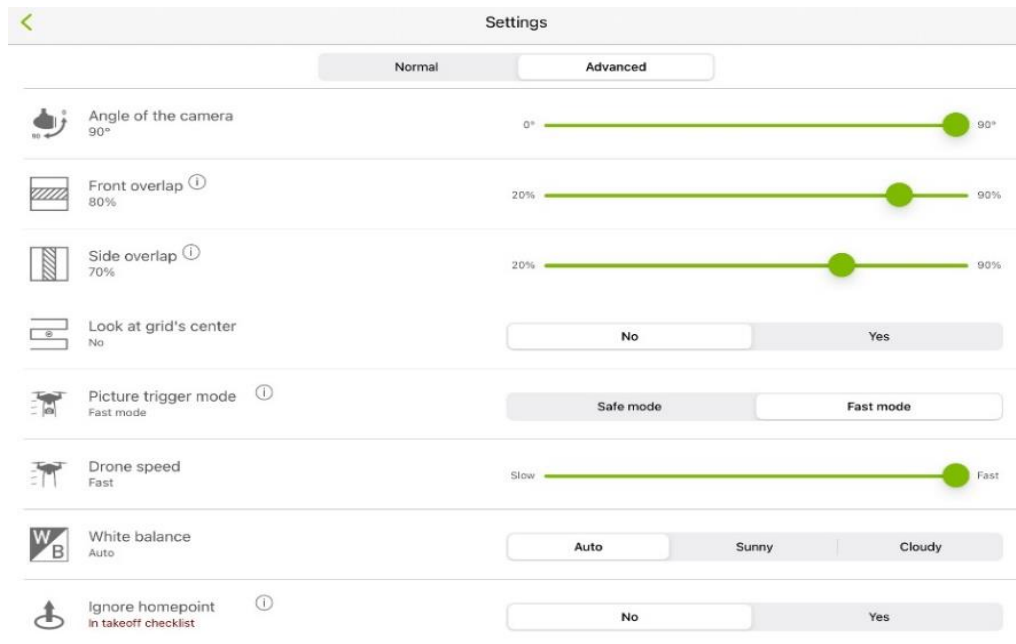
La aplicación que se utilizó para realizar el plan de vuelo fue pix4Capture, esta aplicación admite drones de DJI y Parrot, que son drones adaptados para la agricultura extensiva, el cual posee diversas misiones específicamente la primera es en forma de cuadrícula el cual realiza misiones rectangulares, es adecuado para la mayoría de los entornos, para este tipo de misión se recomienda áreas grandes y superficies planas, la segunda misión es en forma de rejilla doble, realiza dos misiones consecutivas en el nadir del rectángulo, esta misión es recomendada para volar más cerca de los objetos con el fin de adquirir con más detalles las imágenes, además por realizar doble misión se recomienda áreas pequeñas ya que se duplica el tiempo de vuelo y para zonas donde hayan fluctuaciones de altura, la siguiente misión es en forma de polígono, esta misión es adecuada para la mayoría de los entornos que requieren límites flexibles, esta misión se recomienda para zonas complejas y que posee restricciones en los límites de vuelo y por último la misión circular que garantiza que se tome las capturas desde todos los ángulos alrededor de un punto específico, es recomendado para objetos aislados y áreas pequeñas. Conociendo las condiciones de las zonas de adquisición que no son simétricas y además poseen diferentes alturas se optó para este plan de vuelo la misión POLYGON for 2D maps, ósea el plan de vuelo en forma de polígono.



**Figura 39** misiones disponibles de la aplicación pix4Capture. **Fuente:** Autor

Luego de este proceso de la selección de misión, se procedió a realizar el mapeo del área, en este punto se estableció los parámetros de la superposición de imágenes, el ángulo de la cámara y la altura de vuelo se estableció anteriormente.

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.



**Figura 40:** Características de vuelo en el sector de Monte Adentro. **Fuente:** Autor

Finalmente se inició la misión seleccionada y se supervisó en vivo la captura de imágenes, desde la aplicación móvil como se visualiza en la siguiente figura relacionada con la zona de monte adentro.



**Figura 41.** Vision del área desde la aplicación móvil. **Fuente:** Autor




Para la detección de antracnosis en los cultivos de fresa, se tomó como referencia el follaje de los cultivos, ya que, es la zona con mayor área visible. Además, en el momento

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

de la adquisición se observó características específicas de esta enfermedad con una mayor frecuencia en el follaje.

De igual forma para la otra clase llamada palidosis en donde el área afectada es el follaje presentando cambios de color en las hojas del arbusto de fresa llegando a tomar colores amarillos y tonalidades rojizas.

En la siguiente tabla se presenta las tres clases Normal, Antracnosis y Pallidosis (Otras).

Clase Normal	Clase Antracnosis	Clase pallidosis
		
<b>Característica</b>	<b>Características</b>	<b>Características</b>
Presenta hojas totalmente sanas sin presencia de ninguna enfermedad, que se caracteriza por el follaje totalmente verde si ningun tipo de distorción.	Presenta lobulos de color negro, café en el follaje presentado el mismo patron en los diferentes arbusto cercanos al mismo en una fase intermedia-final.	Presenta en el follaje coloración rojiza o morada, especialmente las hojas mas antigua y ademas se quebranta los bordes de las hojas.

**Tabla 4.** Clase Normal, Clase Antracnosis y Clase Pallidosis (Otras). **Fuente:** Autor

## **CAPÍTULO IV.**



4. IMPLEMENTACIÓN DE UN ALGORITMO DE PREPROCESAMIENTO A LA BASE DE DATOS QUE PERMITA DEPURAR LA INFORMACIÓN ADQUIRIDA.



Figura 42 Preprocesamiento de imágenes para zona de interés 1. Fuente: autor.

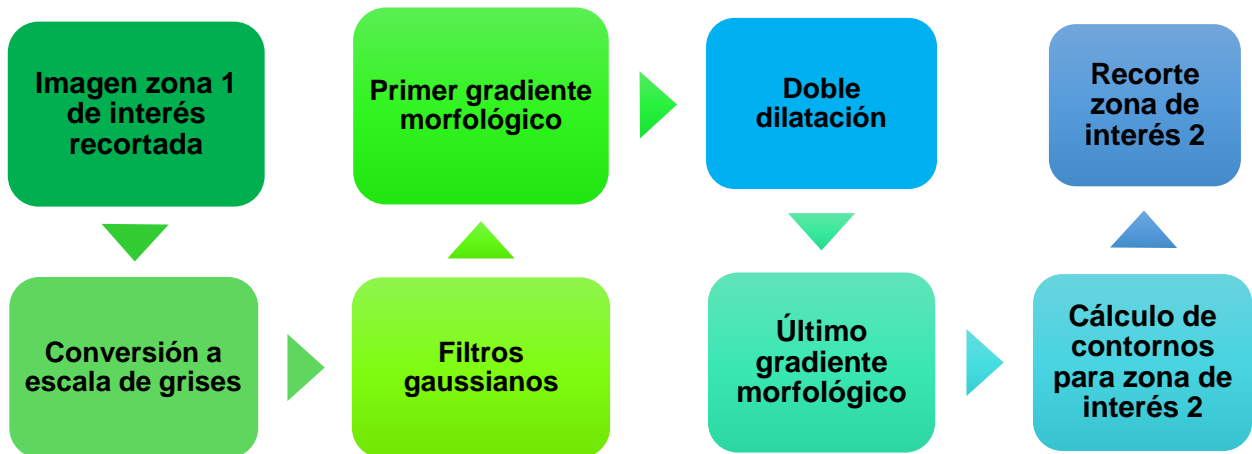


Figura 43 Preprocesamiento de imágenes para zona de interés 2. Fuente: autor.

Antes de realizar el preprocesamiento de las imágenes se acondicionó el entorno en el cual se desarrolló el proyecto, este acondicionamiento está constituido por la instalación de librerías y paquetes necesarios.

Primero se instaló la interfaz gráfica de usuario anaconda navigator que cuenta con un IDE (entorno de desarrollo integrado) multiplataforma de código abierto llamado spyder

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

---

en donde se ejecutó los diferentes códigos. Para lograr este objetivo se comenzó instalando la versión de Python 3.7.9 que es compatible con TensorFlow ya que al instalar versiones superiores presentó problemas al ejecutarlo. Luego dentro de la interfaz gráfica de usuario anaconda navigator se creó un entorno personal llamado tesis al cual se le instaló el paquete de TensorFlow 2, por medio del comando `conda install TensorFlow`, de igual forma se instalaron los paquetes de `opencv`, `numpy`, `matplotlib`, `tqdm`, `imutils`. Luego de estas instalaciones ya se contó con un entorno apto para el desarrollo de proyecto teniendo en cuenta las versiones.

Luego de las instalaciones de los paquetes se crea un cuaderno en spyder en donde se exporta cada librería con la función `import`, para ser luego ser utilizada por sus respectivas funciones y métodos.

las imágenes adquiridas del drone se almacenaron directamente en una micro SD que posee el drone para su respectivo almacenamiento, luego, esta memoria fue extraída e insertada en un adaptador USB para crear la base de datos. Para la creación dicha base, primero, fue necesario realizar los recortes de la primera y segunda zona de interés, en este caso la primera zona correspondió a los surcos donde se encontraban todos los arbustos, mientras la segunda zona se conformó por los arbustos de fresas de manera individual.



**Figura 44** Imagen original de la data. **Fuente:** autor.

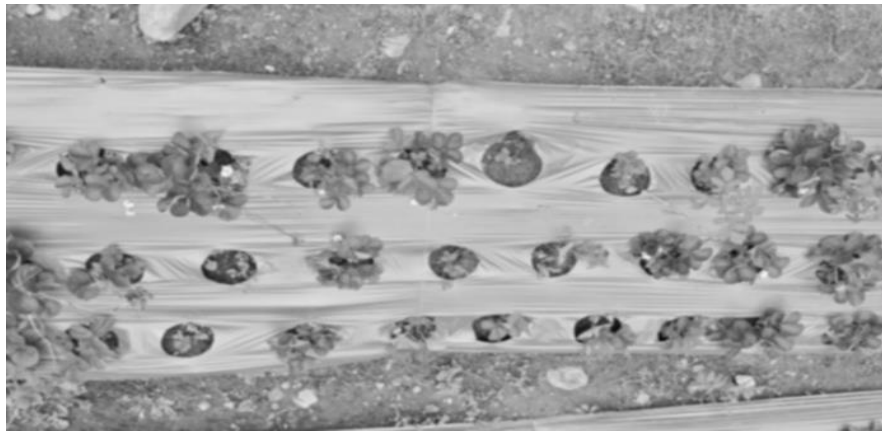
Para lograr este objetivo se realizaron los siguientes pasos:

- Esta data que se adquirió con el drone se cargó por medio de la función `recortes_zonas_interes ()`, esta función recibe como parámetro la carpeta donde se encuentran las imágenes. Esta carpeta es recorrida por la función `listdir` de la librería `os`, identificando cada imagen de la carpeta, estas imágenes se cargaron con la función `cv2.imread` que luego pasaron por un ajuste de dimensión de 700p con la función `imutils`, que recibe las imágenes adquirida de la función `cv2.imread`, estas

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

---

imágenes vienen con una resolución de 1080 pixeles de gran tamaño y por esta razón se estandarizó todas las imágenes a la misma proporción. Las imágenes estandarizadas se convirtieron a escala de grises con la función `cv2.Color` que recibe como parámetros la imagen y el atributo `cv2.COLOR_BGR2GRAY`, generando así, un solo canal de los tres canales RGB, facilitando la identificación de un rango para la binarización. Luego se suavizó las imágenes con 3 filtro gaussiano de dimensiones (5x5), que pasaron por un proceso de dilatación por medio de la función `cv2.dilate` que recibe como parámetros las imágenes pasadas por los anteriores filtros y además un filtro con morfología rectangular de tamaño 7x7.



**Figura 45** Imagen suavizada y filtrada (filtro gaussiano). **Fuente:** autor.

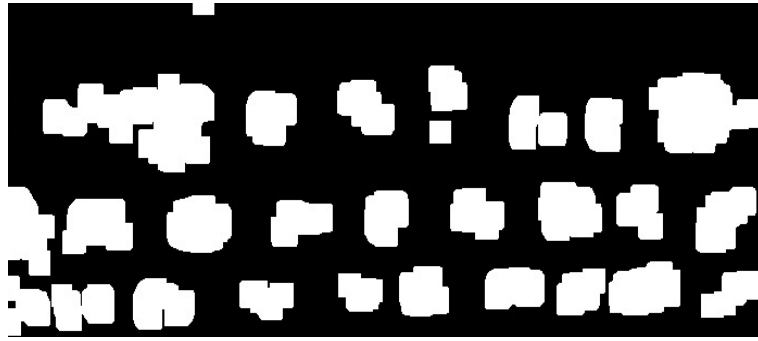
Luego de todos estos procesos se visualizó por medio de histogramas un rango que logre una binarización que clasifique la primera zona de interés del resto de la imagen, esto con el fin de encontrar el contorno externo más grande. El rango se estableció con la función de OpenCV `cv2.inRange` con valores de 175 a 250, los pixeles que se encontraban en este rango se establecieron en 1 mientras los que se encontraban fuera de este, se establecieron en 0. El rango fue atribuido a la función `cv2.findContours` que además recibe otros parámetros, el primero el atributo `cv2.RETR_EXTERNAL` que calcula los contornos externos y el segundo parámetro es `cv2.CHAIN_APPROX_SIMPLE` almacena todo lo finales ahorrando almacenamiento. Ahora por medio de un ciclo se recorrió el Arrays donde se encuentran almacenados los contornos hallados en el paso anterior, luego se estableció un área de comparación con el fin de extraer el contorno de interés, que luego son almacenados en un Arrays, este Arrays se le calculó el máximo y mínimo de los valores del contorno externo con la función `Max` y `min`, esto con el fin de establecer los límites del recuadro para recortar la primera zona de interés en la imagen original.

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

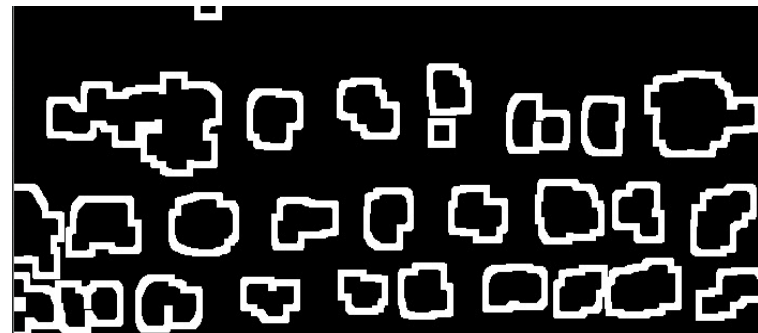


**Figura 46** El contorno verde encierra la primera zona de interés. **Fuente:** autor.

- Las imágenes que se recortaron en el paso anterior se convirtieron una por una a escala de grises y estas fueron pasadas por tres filtros gaussianos con el fin de suavizar las imágenes para calcular el rango en escala de grises que permita la binarización, después se somete por medio de un gradiente morfológico con la función `cv2.Morphology` que recibe como atributo el rango donde se encontraba la segunda zona de interés, este rango se encontraba entre 100 y 255, además recibe como parámetros la función `cv2.MORPH_GRADIENT` y además el kernel de morfología rectangular de dimensión  $7 \times 7$ , dicho gradiente se le proporciona a la función `cv2.dilate` con un filtro rectangular de  $7 \times 7$ , esta dilatación se realiza dos veces con el fin de expandir la zona donde se encuentra cada arbusto. A esta imagen dilatada se le aplicó nuevamente el gradiente morfológico generando así una imagen binarizada con contornos más visibles que luego, por medio de la función `cv2.findContours` proporcionándole la imagen con las alteraciones anteriores y el método `cv2.RETR_LIST` se logró encontrar la lista de todos los contornos, de igual forma que con la primera zona de interés se estableció un área de comparación permitiendo extraer los contornos de interés, estos contornos se recorrieron uno por uno calculando sus máximos y mínimos, por medio de las funciones `Max` y `Min` se obtuvo los puntos específicos que permitieron establecer los recortes individuales de cada arbusto y estos recortes fueron almacenados en una carpeta por medio de la función `cv2.imwrite` a la que se le proporcionó la función `path.join` de la librería `os` para proporcionar una carpeta específica al almacenamiento de las imágenes, además se le proporcionó la imagen y el nombre de la imagen con su formato.



*Figura 47 Doble dilatación. Fuente: autor.*



*Figura 48 Gradiente morfológico. Fuente: autor.*



*Figura 49. Dibujo de contornos. Fuente: Autor*

- Las imágenes almacenadas en la carpeta se clasifican de manera manual, esto se realizó con la ayuda del doctor en ciencias agrarias Leónides Castellanos Gonzáles, dicha clasificación se almacenó en tres carpetas de etiquetas las cuales son: las enfermas por antracnosis, las normales o sanas y las plantas con otro tipo de enfermedad, este otro tipo de enfermedad es llamada palidosis que se encontraba en los cultivos cada una de estas carpetas de etiquetas se encuentran almacenadas en dos directorios, el primero, el de entrenamiento, en donde cada una de éstas tienen

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

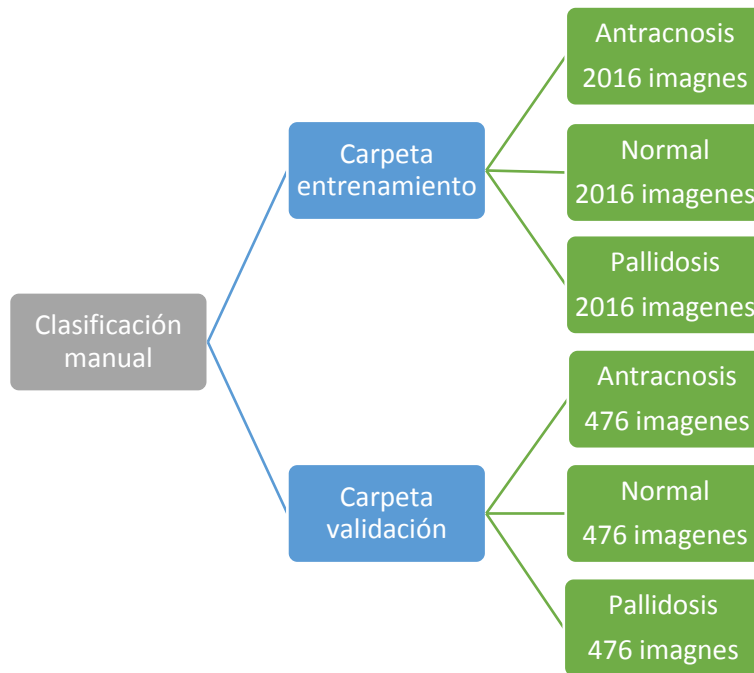
2016 imágenes; y el segundo, el de validación, aquí cada carpeta contiene 476 imágenes.



**Figura 50** Carpetas de entrenamiento y de validación. **Fuente:** autor.



**Figura 51** Carpetas de las tres etiquetas. **Fuente:** autor.



## **CAPÍTULO V.**

## 5. IMPLEMENTACIÓN DE DIFERENTES MÉTODOS DE CLASIFICACIÓN INTELIGENTE PARA LA IDENTIFICACIÓN DE ANTRACNOSIS EN LOS CULTIVOS DE FRESA.

Antes de proporcionarle los datos a los métodos de inteligencia artificial se tuvo que redimensionar las imágenes a un mismo tamaño, por medio de una función desarrollada por el autor llamada `get_data`, esta función se le proporcionó la dirección general en donde se encontraba la data de entrenamiento y la data de validación, esta dirección se obtuvo por medio de las funciones de la librería `OS`. Luego de proporcionarle la dirección de entrenamiento y validación, la función `os.listdir` generó una lista de los elementos que se encontraban dentro de dichas carpetas, estos elementos fueron recorridos por medio de un ciclo en el que se realizó condiciones, si se encontraba dentro de las subcarpetas: `NORMAL`, `ANTRACNOSIS` y `OTRAS(Pallidosis)`, se realizaba la lectura de las imágenes de cada subcarpeta para continuar con un redimensionamiento de las imágenes de `400x400` y que luego fueron guardadas con la función `cv2.imwrite` al cual se proporcionó la dirección de almacenamiento por medio de la función `path.join` de la librería `os`.

Se creó una función llamada `get_data1()` que recibe la dirección de las carpetas de la data de validación y la data de entrenamiento, estas carpetas son recorridas por medio de un ciclo y por medio de condicionales se identifican las subcarpetas, `ANTRACNOSIS`, `NORMAL` y `OTRAS(Pallidosis)`. Al identificar la carpeta normal, se le asignó una etiqueta `label=0`, si se identificaba antracnosis, se etiquetaba `label=1` y si se identificó otras, se le asignaba la etiqueta `label=2`. Luego, se recorre las imágenes por carpetas de etiquetas cargando las imágenes que después son convertidas en Arrays y que se añaden a una lista, de igual forma, sucede con las etiquetas en donde la lista `X` representa las imágenes convertidas en Arrays y `Y` representa las etiquetas de cada una de las imágenes de cada subcarpeta, estos valores son retornados por la función. Después, se estableció las variables de `X` entrenamiento, `Y` entrenamiento, `X` validación y `Y` validación, el cual se les asignó la función `get_data1()` que lleva como atributo la data de validación y la data de entrenamiento. Seguido se guardó estos Arrays para proporcionárselos al preprocesamiento de los datos de la red.

Se estableció el primer algoritmo de inteligencia a utilizar, este algoritmo está conformado por redes neuronales convolucionales basados en capas, primero se utiliza el método de las librerías de Keras llamadas `keras.Sequential`, dicho método permite crear modelos con topologías no lineales con múltiple entradas o salidas. Este método es asignado a una variable `model`, que es la que se replicó en las diferentes capas de la red, en la primera capa se añadió el método de convolución de dos dimensiones al cual se le asignó como atributos:



- Primero, la cantidad de filtros de características, este permite hacer un barrido por todas las imágenes y encontrar características similares en todo el set de datos, cada filtro se encarga de una característica en particular.
- Segundo, el tamaño del filtro y la función de activación rectificante ReLu.
- Por último, el tamaño de la imagen con los tres canales.

Después de esta capa, se añade el método de filtrado maxpool2D con un tamaño de 2x2, este filtro realiza un barrido por toda el área de las imágenes hallando el máximo valor de activación en cada posición del filtro, esto nos permite reducir los tamaños de los filtros establecidos.

En la segunda capa se establece el mismo método de convolución 2D, con los mismos parámetros, excluyendo el tamaño de las imágenes con los canales, seguido de esta capa se añade de filtrado maxpool2D. De igual forma se genera la tercera capa seguida de un maxpool2D, estas capas permiten recolectar características importantes de las imágenes.

Luego de generar estas capas, se crea una capa de aplanamiento utilizando el método Flatten, que convierte la matriz en un Arrays de una sola columna con el fin de proporcionárselo a la primera capa densa de nuestra red neuronal, esta primera capa densa (Dense) tiene como atributos: primero, la cantidad de neuronas y segundo, la función de activación rectificante (ReLU). Por último, se genera una capa de salida utilizando el método Dense con tres neuronas y la función de activación softmax. Con esto finaliza las capas de la red.

Seguido, se utilizó la función summary () para generar una tabla con todas las capas que constituye la red, esto con el fin de visualizar las transformaciones que tuvieron los diferentes parámetros de la red.

Seguido de la visualización de los componentes de la red con la función summary, se compila la red por medio de la función compile, este tiene como parámetros la función de pérdida, que refleja durante el proceso de entrenamiento de la red, lo lejos que se encuentra en un momento dado el resultado de la salida con respecto al resultado que se considera correcto o deseado, en este caso se seleccionó sparse\_categorical\_crossentropy, optimizador adam y una métrica accuracy que se utiliza para evaluar el rendimiento de la red.

Después se utilizó el método de Keras callbacks.TensorBoard, a este método se le proporcionaron unos parámetros, en donde el primero, es la librería OS que generó una carpeta llamada data y el segundo la librería datetime para establecer fecha y hora en la que se generó la carpeta, en esta carpeta generada se almacenan los archivos que me redirigen a la plataforma de TensorBoard. Dicho método se utiliza con el fin de generar gráficos de entrenamiento y validación en la plataforma de TensorBoard en tiempo real, esto quiere decir que, a medida que va entrenando y validando se van visualizando los

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

cambios en la plataforma de TensorBoard; esto es de gran ayuda ya que, podemos evitar el sobre ajuste de nuestra red neuronal, pues gracias a los gráficos generados sabemos si la red está aprendiendo o no.

Luego, se llama al método fit al cual se le proporcionan los sets de entrenamiento y validación, además, de la cantidad de épocas y la función callback generada en el paso anterior.

```
Model: "sequential_4"
Layer (type)                Output Shape                Param #
-----
conv2d_16 (Conv2D)          (None, 400, 400, 80)       1040
max_pooling2d_16 (MaxPooling (None, 200, 200, 80)       0
conv2d_17 (Conv2D)          (None, 200, 200, 80)       25680
max_pooling2d_17 (MaxPooling (None, 100, 100, 80)       0
conv2d_18 (Conv2D)          (None, 100, 100, 80)       25680
max_pooling2d_18 (MaxPooling (None, 50, 50, 80)         0
conv2d_19 (Conv2D)          (None, 50, 50, 80)         25680
max_pooling2d_19 (MaxPooling (None, 25, 25, 80)         0
flatten_4 (Flatten)         (None, 50000)              0
dense_8 (Dense)             (None, 80)                  4000080
dense_9 (Dense)             (None, 3)                   243
-----
Total params: 4,078,403
Trainable params: 4,078,403
Non-trainable params: 0
```

**Figura 52** Capas de la red convolucional. **Fuente:** autor.

La segunda red que se implementó fue una red mediante aprendizaje por transferencia en donde se cuenta con una red a gran escala proporcionada por Google que fue previamente entrenada con un gran conjunto de datos, este modelo se puede utilizar preentrenado o se puede utilizar para personalizar el modelo para una tarea en particular.

Primero se cargaron los datos con las librerías de sistema operativo OS, luego se crea el modelo base, llamando a la función `keras.applications` y asignándole el modelo pre entrenado llamado MobileNet2V, al cual se le proporciona los parámetros de ancho y alto de la imagen, los pesos por medio del atributo `imagenet` y además indicándole por medio del parámetro `include_top= False`, que no se va a utilizar el output de la red original. Luego se visualizó el modelo con la función `summary`, seguido se decide si se congela el modelo o no con la función `trainable` estableciendo "True" o "False", esto con el fin de no modificar los pesos establecidos por MobileNet2V de Google, a continuación, se creó una capa de maxpooling global con la función `GlobalAveragePooling2D` con el fin de reducir al máximo la data de entrada, realizando un promedio del resultado de pooling, además proporcionándole el output de la red anterior, siendo el input de la nueva red. Se genera

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

una capa de salida con una neurona con el fin de adaptarla a la red anterior, con función de activación sigmoide y además pasándole como parámetro la función GlobalAveragePooling2D, con este último paso se estableció la cabeza para el modelo.

Ahora para definir el modelo se llama al método Keras.model.Model al cual se le proporcionó como parámetros los inputs que es igual al modelo base y además indicarle los outputs que son equivalente a la capa de salida anterior. Luego se compila el modelo utilizando el método compile, que tiene como optimizador RMSprop (0.0001) que para este tipo de aplicación es mejor que adam y además se estableció como parámetro la función de pérdida Binary\_crossentropy y por último una métrica de precisión accuracy. A continuación, se genera la normalización de las imágenes con la función ImageDataGenerator estableciéndole un reescalado de 1/255 para entrenamiento como validación, luego se establece por medio de la función Flow\_from\_directory la dirección donde se encuentra la data de entrenamiento, además se estableció el tamaño de la imagen de 224x224 y un class\_mode binary. De igual forma se realiza este proceso con el set de validación. Luego de todo este proceso se entrena el modelo con la función fit asignándole el set de entrenamiento, el numero de épocas=5 y por ultimo el set de validación, por último, se evalúa el modelo con el método evaluate que tiene como parámetros la data de validación.

El resultado de la red se presenta en la siguiente tabla.

N° de Épocas	Set entrenamiento	Set validación	Porcentaje de entrenamiento	Porcentaje de validación
1	2016	476	33.40%	33.33%
2			33.33%	33.33%
3			33.33%	33.33%
4			33.37%	33.33%
5			33.50%	33.33%

**Tabla 5.** Entrenamiento y validación de la red de aprendizaje por transferencia. **Fuente:** Autor

Al comparar las dos redes, se seleccionó la red convolucional ya que desempeñó una mejor adaptación con los datos proporcionados, además, en comparación al método de aprendizaje por transferencia la red convolucional obtuvo mejores resultados ya que esta red esta exclusivamente diseñada para la identificación de imágenes gracias a sus capas convolucionales que extraen la máxima información de los pixeles de las imágenes.

## **CAPÍTULO VI.**

## 6. VALIDACIÓN DEL MÉTODO DE INTELIGENCIA ARTIFICIAL SELECCIONADO INCLUYENDO FACTORES DE CONFUSIÓN Y RESULTADOS.

Durante este proceso se establecieron diferentes parámetros a la red neuronal convolucional para lograr un buen porcentaje de validación. El primer parámetro que se le alteró a la red fue el número de capas convolucionales dejando estable el número de capas densas de la red.

La configuración inicial que presentó la red fue una con 4 capas convolucionales, 80 filtros por cada capa convolucional con sus respectivos Maxpooling. Esta configuración no generó los porcentajes esperados. Se realizó un cambio en la cantidad de data, de 1000 a 2016 para el set de entrenamiento y de 200 a 476 el set de validación imágenes e incrementado a 5 el número de capas convolucionales esto permitió obtener mejores porcentajes de validación.

se tomó el set de validación el cual contiene las imágenes de validación en formato Arrays con sus respectivas etiquetas correspondientes a normal, Antracnosis y otra enfermedad (Pallidosis). Estos datos se le proporcionaron al método evaluate quien se encarga de validar. En esta parte se imprime el porcentaje de error y el porcentaje de evaluación que generó el método mencionado, con los datos de validación y sus etiquetas.

Nº de Épocas	Set entrenamiento	Set validación	Porcentaje de entrenamiento	Porcentaje de validación	Numero de capas CNN	Loss train y Loss val
1	2016	476	74.28	95.59	5	1.7291-0.1676
2			85.69	97.29		0.3524-0.1422
3			87.15	96.11		0.3252-0.1472
4			87.90	95.80		0.3018-0.1354
5			88.24	96.85		0.2877-0.1082
6			89.38	98.11		0.2705-0.1000
7			91.05	96.95		0.2293-0.0953
8			95.48	95.48		0.1923-0.1202

**Tabla 6:** Entrenamiento y validación de la red convolucional sin factor de confusión. **Fuente:** Autor

## DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL REMOTO PARA LA DETECCIÓN DE ANTRACNOSIS O MANCHA FOLIAR EN LOS CULTIVOS DE FRESA.

Nº de Épocas	Set entrenamiento	Set validación	Porcentaje de entrenamiento	Porcentaje de validación	Numero de capas CNN	Loss train y Loss val
1	2016	476	84.74	73.25	5	0.3622-0.6238
2			86.19	75.77		0.3316-0.5882
3			89.19	78.08		0.2636-0.5375
4			91.05	74.37		0.2272-0.6904
5			93.83	76.26		0.1562-0.8014
6			94.91	74.16		0.1337-0.8619
7			95.25	71.57		0.1266-1.0393

**Tabla 7.**entrenamiento y validación con patrón de confusión. **Fuente:** Autor

Para la validación del método se implementó el algoritmo con y sin patrón de confusión con el fin de observar la importancia que tiene este parámetro en la red.

La red sin patrón de confusión presenta un porcentaje de validación alto gracias al número de datos con el que se entrenó la red. Se utilizaron 64 filtros convolucionales con 5 capas de extracción de características convolucionales, cada una con su respectivo Maxpooling.

Se optó por seleccionar un numero de iteraciones de épocas equivalente a 6, dado que en este punto la red obtiene su máximo valor de 98.11% en el porcentaje de validación y un 89.38% en el porcentaje de entrenamiento.

Estos valores se presentan dado que la Data de validación fue seleccionada de manera minuciosa y además por el número de datos adquiridos.

La diferencia de valores de entrenamiento y validación se debe a que la red presenta datos más complejos en la fase de entrenamiento en comparación a la de validación que presenta datos más relevantes y específicos y además por el número de datos. Además, se debe a que la métrica de entrenamiento informa el promedio de una época, mientras que las métricas de validación se evalúan después de la época, por esta razón la métrica de validación ven un modelo que se ha entrenado un poco más.

El patrón de confusión en este punto es otra enfermedad (Pallidosis) hallada en los arbustos de fresa en la vereda el Totumo que se solapa con antracnosis, pero que

presentan diferentes características y, por ende, pueden ser separables, en este caso, la red presenta un buen comportamiento, no excelentes, pero son porcentajes validos en las condiciones que se encuentra estos parámetros con el factor de confusión.

La red con factor de confusión tuvo un reto más grande, cuenta con los mismos parámetros de la red anterior, pero en este caso se añade la clase llamada “otras” equivalente a la enfermedad pallidosis. Esta clase cuenta con 2016 imágenes en el set de entrenamiento y 476 imágenes para el set de validación.

Esta red es entrenada con las tres clases, cada una de estas clases contaba con 2016 imágenes para entrenamiento y 476 para validación, durante este proceso la red se adaptó a la nueva condición dada por la enfermedad pallidosis en donde esta enfermedad se solapa con antracnosis en varias imágenes de la data, confundiendo la red, pero sin importar se logró un buen porcentaje de validación.

Este porcentaje fue mejorando desde la primera época donde se obtuvo un 73,25% en la validación y en entrenamiento un porcentaje de 84.74% aumentando estos valores en la segunda época con porcentajes de 75.77% para la validación y 86.19 % para entrenamiento y logrando para la tercera época un mejor porcentaje de 78.08% para validación y un porcentaje del 89.18% para entrenamiento.

Se analizó los resultados arrojados por la red con y sin factor de confusión con una data de 2016 imágenes para entrenar y 476 para validar para cada una de las clases. En donde se observó la eficiencia del método sin patrón de confusión el cual arrojó porcentaje excelente de validación y la importancia que tiene este parámetro en el porcentaje de validación como el de entrenamiento.

La validación de la eficiencia de la red con el factor de confusión, tuvo diferentes procesos, al establecer una red con 80 filtros y una data de 1000 imágenes y 200 de validación por cada clase, además se estableció 5 épocas que son equivalente a los ciclos de propagación en ambos sentidos para reducir las pérdidas y además, las red contaba con 4 capas convolucionales, obteniendo un 66.80% del porcentaje de validación, este porcentaje se dio por el sobre ajuste de la red manteniéndose dentro de un canal alrededor de un valor, lo que significa que la red no aprendió ya que comenzó en la primera época con un 66.12% seguido de un 58.42% en la segunda época, manteniendo estos valores sin importar el número de épocas que se le asignaran . Dado estos errores de la red se optó por primero aumentar la cantidad de capas de la red a 5 capas generando un resultado por debajo del anterior equivalente a un 33.33%, optando por otra solución, se incrementó el número de datos por clase a un numero de 2016 imágenes para el set de entrenamiento y 476 imágenes para el set de prueba o validación, además se redujo el número de filtros a 64 con 4 capas convolucionales. Con estas condiciones se obtuvo un 70.066% en el test de validación en la segunda época generando una mejora en el comportamiento de aprendizaje de la red, estas condiciones todavía eran regulares, lo cual se tomó como primera solución agregar una quinta capa

convolucional que permitió extraer mayores características de las imágenes logrando mejores resultados equivalentes a un 73.25% en la primera época, logrando un reflejo del aprendizaje de la red en la segunda época con un porcentaje del 75.77% y ratificando dicho aprendizaje con un 78.08%. en la tercera época. Desde este punto la red ingresa en rango siendo innecesario agregar más épocas y además el porcentaje de perdidas descendió hasta la época 3, después de esta época el error incrementó.

Durante en proceso de entrenamiento de la red se obtuvieron cuatro resultados en cada época, el primero el porcentaje de perdida tanto para validación como entrenamiento, el segundo el porcentaje de aprendizajes con los datos de entrenamiento de la red y por último la validación con los datos nuevos. Estos datos se ven reflejados en la siguiente figura.

```
6048/6048 [=====] - 2204s 364ms/sample - loss: 0.3622 -  
accuracy: 0.8474 - val_loss: 0.6238 - val_accuracy: 0.7325  
Epoch 2/7  
6048/6048 [=====] - 2109s 349ms/sample - loss: 0.3316 -  
accuracy: 0.8619 - val_loss: 0.5882 - val_accuracy: 0.7577  
Epoch 3/7  
6048/6048 [=====] - 2106s 348ms/sample - loss: 0.2636 -  
accuracy: 0.8919 - val_loss: 0.5375 - val_accuracy: 0.7808  
Epoch 4/7  
6048/6048 [=====] - 2206s 365ms/sample - loss: 0.2272 -  
accuracy: 0.9105 - val_loss: 0.6904 - val_accuracy: 0.7437  
Epoch 5/7  
6048/6048 [=====] - 2192s 362ms/sample - loss: 0.1562 -  
accuracy: 0.9383 - val_loss: 0.8014 - val_accuracy: 0.7626  
Epoch 6/7  
6048/6048 [=====] - 2141s 354ms/sample - loss: 0.1337 -  
accuracy: 0.9491 - val_loss: 0.8619 - val_accuracy: 0.7416  
Epoch 7/7  
6048/6048 [=====] - 2171s 359ms/sample - loss: 0.1266 -  
accuracy: 0.9525 - val_loss: 1.0393 - val_accuracy: 0.7157  
1428/1428 [=====] - 116s 82ms/sample - loss: 1.0393 -  
accuracy: 0.7157  
test accuracy:0.7156862616539001
```

**Figura 53:** Resultados de entrenamiento y validación de la mejor configuración con factor de confusión. **Fuente:** Autor

Se aprecia en la figura anterior la relación de crecimiento de los porcentajes de entrenamiento y validación de la red hasta la época 3, en donde el testeo de validación después de esta época comienza a bajar hasta un valor 71.57% en la época 7, lo que refleja que la red presentó su límite de aprendiza hasta la época 3 en donde brinda un porcentaje aprobado del testeo. Además del test de validación se puede visualizar el test de perdida con los datos de validación, en donde se presenta una reducción de este valor que comienza con un porcentaje del 36.22% en la primera época seguido de un 33.16% en la segunda época hasta la quinta época donde presenta una reducción del 15.62%.

## 7. CONCLUSIONES.

Uno de los parámetros fundamentales en el entrenamiento y validación para obtener un buen porcentaje de testeo, fue la calidad de la clasificación manual y la cantidad de imágenes para la constitución de la base de datos, este proceso fue muy detallado en el



que se seleccionó cuidadosamente y con conocimiento de las características de la enfermedad para la distribución de cada clase. Esto evitó en gran medida el sobre ajuste que presentaba la red por la ausencia de datos, gracias a que se duplicó la data de validación y entrenamiento, logrando el aprendizaje de la red y permitiendo así un buen porcentaje de validación.

Además de generarse un overfitting por ausencia de datos también se generó en las primeras etapas del proyecto un Underfitting, que es un sub ajuste de la red porque la data que se utilizó para set entrenamiento era similar al utilizado en el set validación, lo que generaba altos índices de aprendizaje sin errores notables de clasificación. Esto puso en duda la eficiencia de la red en donde se realizó una inspección minuciosa de los datos evitando que se encontraran data similar entre los sets de entrenamiento y validación, solucionando este problema.

Al añadir el factor de confusión a la data, la red presentó un proceso de entrenamiento más complejo, generando una mala interpretación la red en algunas imágenes, que presentaban las dos enfermedades (Antracnosis y Pallidosis) y por ende en estos instantes la validación no fue excelentes.

Durante en preprocesamiento de los datos se generó grandes errores por las versiones y la compatibilidad de los paquetes de TensorFlow y opencv añadido a esto las versiones de Python que entraba en conflicto con TensorFlow desde la versión 3.8 en adelante, además por las actualizaciones de opencv de la versión 3 a la 4, cambian algunos parámetros que generaban complicaciones el proceso, por ende, hay que conocer los cambios en sus actualizaciones.

Las condiciones de adquisición son específicas, por tal razón, el código se diseñó con restricciones que limitan la distancia de adquisición del drone y su orientación, permitiendo que el código de preprocesamiento funcione de una manera eficiente dentro de estos rangos, en el caso de proporcionarle una data que se encuentre fuera de estos rangos, el código de preprocesamiento de datos comienza a fallar dando como resultado recortes de zonas de interés no deseados.

En cuanto a la comparación de las redes, se obtuvo resultados favorables en la implementación de la red neuronal convolucional, aunque se prestaron diversos problemas por las características del ordenador donde se ejecutó el código, se pudo lograr que la red respondiera de manera positiva, mientras que, en la red perceptrón multicapa se presentaron más dificultades respecto al entrenamiento y validación ya que mi ordenador no contaba con las mejores especificaciones computacionales para este tipo de prácticas.

El factor de confusión resultó de gran importancia debido a la adaptación que tuvo que tener el método de convolución para la detección de éste, no fue excelente, pero se logró

un porcentaje aprobación valido permitiendo de manera aceptable la detección de antracnosis y pallidosis en los cultivos de fresa de manera oportuna.

### 8. TRABAJOS FUTUROS.

Diseñar una plataforma web con métodos de inteligencia artificial que logre detectar la presencia de diferentes enfermedades de manera cualitativa en los cultivos de fresa. Para lograr este proyecto es fundamental la implementación de cloud computing en donde se realiza por medio de plataformas como Google el alquiler de máquinas virtuales que poseen grandes capacidades de memoria y GPU, dado que los ordenadores regulares se ven forzados para realizar este tipo de proyectos, como es el caso de mi ordenador en el cual desarrollé el proyecto. Este ordenador presentó varios problemas por las capacidades limitadas que posee. De este modo el proceso se agiliza obteniendo en un tiempo menor mejores resultados, logrando así entrenar con una data grande el modelo de inteligencia artificial de manera muy rápida, permitiendo la configuración de parámetros e implementación de diferentes algoritmos por la velocidad que presentan estas plataformas.

### REFERENCIAS BIBLIOGRÁFICAS.

- [1] J. B. M. Gaitán, R. L. Ferrucho, and J. G. Á. Herrera, “Efecto de dos cepas de *Trichoderma* en el control de *Botrytis cinerea* y la calidad del fruto en fresa (*Fragaria* sp.),” *Rev. Colomb. Ciencias Hortícolas*, vol. 8, no. 1, pp. 44–56, 2014, doi: 10.17584/rcch.2014v8i1.2799.
- [2] M. A. Cano, “Estrategias biológicas para el manejo de enfermedades en el cultivo de fresa (*Fragaria* spp.),” *Rev. Colomb. Ciencias Hortícolas*, vol. 7, no. 2, pp. 263–276, 2014, doi: 10.17584/rcch.2013v7i2.2240.
- [3] A. Baldovino, “Alternativas orgánicas para el logro de producciones más limpias de la fresa en Pamplona Norte de Santander,” 2018.
- [4] D. J. L. P. Coto, “MANEJO DE PLAGAS Y ENFERMEDADES EN PRODUCCIÓN ECOLÓGICA.”
- [5] R. Carmona Angulo, “Fresa,” 2009.
- [6] Cámara de comercio de Bogotá, “Manual de fresa,” pp. 1–54, 2015.
- [7] G. Giménez, J. Paullier, and D. Maeso, *Identificación y manejo de las principales enfermedades y plagas en el cultivo de frutilla*, no. 82. 2003.
- [8] Ministerio de Agricultura, “Subsector productivo de la Fresa.”
- [9] K. Rane, “Antracnosis De La Rosa,” no. 14, 2004, [Online]. Available: [https://ag.purdue.edu/btny/ppdl/Pages/POTW\\_old/6-14-04.html](https://ag.purdue.edu/btny/ppdl/Pages/POTW_old/6-14-04.html).

- [10] L. Urdaneta, M. E. Sanabria, D. Rodríguez, and M. Pérez de Camacaro, "Antracnosis causada por *Colletotrichum acutatum* Simmonds en frutos de fresa en los estados Lara y Trujillo, Venezuela," *Rev. la Fac. Agron.*, vol. 30, no. 4, pp. 504–528, 2013.
- [11] C. Central, C. Central, C. Extensi, and S. Agr, "Manual de producción de fresa," 2015.
- [12] L. García, "Drones, el cielo está al alcance de todos," p. 65, 2017, [Online]. Available: [https://www.edubcn.cat/rcs\\_gene/treballs\\_recerca/2015-2016-03-1-TR.pdf](https://www.edubcn.cat/rcs_gene/treballs_recerca/2015-2016-03-1-TR.pdf).
- [13] DJI, "DJI Matrice 100 Technical Sheet," vol. 1, p. 78, 2016, [Online]. Available: [https://dl.djicdn.com/downloads/dev/Matrice/en/M100\\_In\\_the\\_Box\\_en\\_160304.pdf](https://dl.djicdn.com/downloads/dev/Matrice/en/M100_In_the_Box_en_160304.pdf).
- [14] DJI, "Manual Cámara Zenmuse Z3," 2016.
- [15] DJI, "Manual de Mavic 2 Pro/Zoom," pp. 1–68, 2019.
- [16] D. Conde Ortiz, "Inteligencia artificial con TensorFlow para predicción de comportamientos," 2018, [Online]. Available: [https://idus.us.es/handle/11441/80122#.X0neHv\\_vQyY.mendeley](https://idus.us.es/handle/11441/80122#.X0neHv_vQyY.mendeley).
- [17] G. van R. and F. L. Drake, "Python language," *Python Ref. Manual*, vol. PythonLabs, 2001.
- [18] J. A. Perez Prieto, "Curso de Python Astronomía," p. 128, 2018, [Online]. Available: [http://www.iac.es/sieinvens/python-course/docs/curso\\_python\\_astronomia\\_2018-02.pdf](http://www.iac.es/sieinvens/python-course/docs/curso_python_astronomia_2018-02.pdf).
- [19] Google, "TensorFlow." <https://www.tensorflow.org/?hl=es-419>. (accessed Nov. 28, 2020).
- [20] Keras, "Keras." <https://keras.io/about/> (accessed Nov. 28, 2020).
- [21] 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).
- [22] C. Platero, "Procesamiento Morfológico," *Madrid Univ.*, pp. 171–198, 2012, [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Procesamiento+morfol?gico#4>.
- [23] F. J. Nuñez, "Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional," *Univ. Oberta Catalunya*, p. 53, 2016, [Online]. Available:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/52222/6/fnunezsTFM0616memoria.pdf>.

- [24] INCyTU, “Inteligencia Artificial Distribuida y Sistemas Multiagentes,” *Intel. Artif.*, vol. 2, no. 6, pp. 1–6, 2018, doi: 10.4114/ia.v2i6.614.
- [25] O. Datatau and O. H. News, “Getting Started with Deep Learning and Python Getting Started with Deep Learning and Python,” pp. 1–21, 2015.
- [26] P. Ashton, “Getting started: Getting started,” *Quick Hits New Fac. Success. Strateg. by Award. Teach.*, pp. 4–6, 2004.
- [27] Google, “Developers.” <https://developers.google.com/machine-learning/guides/text-classification?hl=es-419>. (accessed Nov. 28, 2020).
- [28] G. De Catalunya, “PROYECTO: Aplicación práctica de la visión artificial en el control de procesos industriales.” 2011.
- [29] A. Moreno, “Clasificación de imágenes usando redes neuronales convolucionales en Python,” p. 80, 2019.
- [30] P. Constante, A. Gordon, O. Chang, E. Pruna, F. Acuna, and I. Escobar, “Artificial Vision Techniques to Optimize Strawberrys Industrial Classification,” *IEEE Lat. Am. Trans.*, vol. 14, no. 6, pp. 2576–2581, 2016, doi: 10.1109/TLA.2016.7555221.
- [31] M. Bhangé and H. A. Hingoliwala, “Smart Farming: Pomegranate Disease Detection Using Image Processing,” *Procedia Comput. Sci.*, vol. 58, pp. 280–288, 2015, doi: 10.1016/j.procs.2015.08.022.
- [32] D. I. Patrício and R. Rieder, “Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review,” *Comput. Electron. Agric.*, vol. 153, no. August, pp. 69–81, 2018, doi: 10.1016/j.compag.2018.08.001.