

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

**APLICACIÓN DE MACHINE LEARNING PARA ESTIMAR LA
PRODUCCIÓN DE EMPRESAS AGRÍCOLAS EXPORTADORAS Y
ANÁLISIS DE PRODUCTOS EN EMPRESAS IMPORTADORAS**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
CIENCIAS DE LA COMPUTACIÓN**

KEVIN MAURICIO PALLO AMAGUAYA

kevin.pallo@epn.edu.ec

DIRECTOR: MSc. HENRY PATRICIO PAZ ARIAS

henry.paz@epn.edu.ec

DMQ, febrero 2024

CERTIFICACIONES

Yo, Kevin Mauricio Pallo Amaguaya declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

KEVIN PALLO

Certifico que el presente trabajo de integración curricular fue desarrollado por KEVIN MAURICIO PALLO AMAGUAYA, bajo mi supervisión.

MSc. HENRY PAZ
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Kevin Pallo

MSc. Henry Paz

Kevin Guachi

Jorge Caicedo

DEDICATORIA

El resultado de este trabajo va dedicado a mi familia, padres, hermanos, tíos, primos, abuelitos y amigos.

A mi padre Vicente por enseñarme a ser una persona fuerte, dedicada y responsable, quien día a día estuvo pendiente de la continuidad de mi trabajo, dándome ánimos para que no me rinda en este proceso, a quien se lo agradezco de todo corazón por su preocupación.

A mi madre Gladys quien fue la persona que tuvo que soportar todas las consecuencias de mi trabajo, mi mal carácter, mi cansancio, mi constante estrés producto de la realización del mismo, quien además me apoyo incondicionalmente con alimentación y supo guiarme día a día por el camino del bien y llegar a este punto de mi vida, ya casi un profesional.

A mi hermano Johnny por demostrarme que se puede lograr todo lo que uno se proponga en la vida, por ser un guerrero y salir adelante a pesar de las adversidades.

A mis hermanas Gabriela y Carolina por apoyarme día a día y por enseñarme que las cosas con amor y paciencia salen mejor.

A mis tíos, primos, amigos y abuelitos quienes estuvieron conmigo en momentos difíciles, aconsejándome, regañándome y sobre todo apoyándome.

Finalmente, a mis seres queridos que ya no están conmigo en este momento para disfrutar de este momento muy especial, mi gran amigo Paul, mi tío Miguel, mis abuelitos Enrique e Inés, quienes estoy seguro de que me brindaron su apoyo, bendiciones y me motivaron a esforzarme para lograr lo que uno quiere.

AGRADECIMIENTO

Quiero agradecer primeramente a Dios por regalarme la vida y la salud para salir adelante día a día, también por las oportunidades brindadas que no las he desperdiciado.

A mis abuelitos, Francisco y Teresa, quienes son la base, los cimientos de este gran trabajo, por enseñarme a ser humilde, trabajador, responsable y guiarme hasta el día de hoy por el camino del bien.

A mis padres Vicente y Gladys, quienes se esforzaron por ayudarme a conseguir este sueño tan anhelado, quienes soportaron mis malas actitudes, mi estrés constante, sin embargo, nunca dejaron de apoyarme y guiarme por el camino correcto.

A mis hermanos Johnny, Gabriela, Carolina, quienes fueron el bastón junto a mis padres para lograr este sueño, con quienes compartí momentos familiares llenándome de amor y cariño para inspirarme a ser mejor día a día.

A mis tíos y primos, por aconsejarme y darme palabras de aliento para lograr este objetivo, por estar siempre conmigo y junto a mi familia para apoyarnos en todo momento y no dejarnos solos.

A mis amigos universitarios Johan, Sergio, Michael, Henry, Adrián Boris, Kevin, Toa, Joss y Jordy con quienes compartí grandiosos momentos en esta travesía. En especial a Sergio, Henry y Adrián quienes fueron mis compañeros de trabajos, proyectos, tareas y quienes siempre estuvieron conmigo y me apoyaron para no rendirme en mis momentos de frustración y desesperación.

A mis amigos del barrio Johao, Joel, David, Bryan y Edison con quienes compartí momentos divertidos los fines de semana con el fin de desestresarme y disfrutar cada momento de vida.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos.....	2
1.3 Alcance	2
1.4 Marco teórico.....	4
1.4.1 Origen de las empresas florícolas en el Ecuador.....	4
1.4.2 Exportación de flores en Ecuador	4
1.4.3 Exportación total por año	5
1.4.4 Exportaciones por tipo de flor.....	5
1.4.5 Exportaciones de flores por país de destino	6
1.4.6 Variedad de rosas ecuatorianas	6
1.4.7 Estados fonológicos de una rosa.....	8
1.4.8 Inteligencia artificial en la producción de rosas.....	10
1.4.9 Inteligencia Artificial.....	10
1.4.10 Machine Learning.....	10
1.4.11 Aprendizaje profundo	11
1.4.12 Redes neuronales	11
1.4.13 Redes neuronales convolucionales	12
1.4.14 Visión Artificial.....	14
1.4.15 YOLO.....	14
1.4.16 DeepSort.....	19
1.4.17 ByteTrack.....	19
2 METODOLOGÍA.....	20
2.1 Entendimiento del negocio.	20
2.2 Entendimiento de datos.....	21
2.3 Preparación de los datos.....	24
2.3.1 Data Image Generator.....	24
2.3.2 Make Sense	25

2.4	Modelado	27
2.4.1	Modelo utilizando redes neuronales convolucionales.....	27
2.4.2	Modelo utilizando YOLO.....	29
2.5	Evaluación	30
2.6	Despliegue.....	33
2.6.1	DeepSort.....	33
2.6.2	ByteTrack.....	34
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	36
3.1	Resultados	36
3.1.1	Resultado del modelo evaluado en imágenes	36
3.1.2	Resultado del modelo evaluado en un video.....	38
3.2	Conclusiones.....	39
3.3	Recomendaciones	39

RESUMEN

El sector exportador agrícola, es uno de los sectores donde no se ha podido implementar tecnología o pocas personas tienen la motivación suficiente para introducirse en este campo, por tal motivo existen problemas que necesitan de una solución tecnológica y científica. Uno de los problemas es que al sector exportador agrícola le es fundamental conocer la producción que tiene en sus terrenos. En la actualidad este control se realiza operativamente por medio de trabajadores de la industria, lo cual tiene un margen de error alto ya que muchas de las rosas llegan a medir alrededor de 3 metros lo cual es un problema para los humanos detectar toda la producción.

En el documento se presenta una posible solución para corregir problemas de conteo y mejorar tiempos que toma realizar esta tarea utilizando técnicas de Machine Learning y Deep Learning para crear un modelo de IA que realice esta tarea. Para el desarrollo del modelo de IA se utilizaron modelos de redes neuronales pre entrenadas como Yolov5, Yolov6, Yolov8, entre otros. Estos modelos ya se encuentran pre entrenados con grandes datasets de datos que brindan un conocimiento base para clasificar, detectar y segmentar objetos de nuestro interés, en nuestro caso, rosas. Sin embargo, no todo está solucionado ya que se necesita conocer el número de rosas dependiendo su estado fonológico y para esto se utilizó una técnica llamada ByteTrack que nos proporciona herramientas que facilitan el conteo y seguimiento de una rosa durante su movimiento en el video proporcionado.

PALABRAS CLAVE: Visión computacional, ByteTracker, Detección de objetos, Yolo, Rosas, Inteligencia artificial, Python, Redes Neuronales Artificiales, CRISP-DM, DeepSort.

ABSTRACT

The agricultural export sector is one of the sectors where technology has not been implemented or few people have sufficient motivation to enter this field, for this reason there are problems that require a technological and scientific solution. One of the problems is that it is essential for the agricultural export sector to know the production it has on its land. Currently this control is carried out operationally by industry workers, which has a high margin of error since many of the roses measure around 3 meters, which is a problem for humans to detect the entire production.

The document presents a possible solution to correct counting problems and improve the times it takes to perform this task using Machine Learning and Deep Learning techniques to create an AI model that performs this task. To develop the AI model, pre-trained neural network models such as Yolov5, Yolov6, Yolov8, among others, were used. These models are already pre-trained with large data datasets that provide a base knowledge to classify, detect and segment objects of our interest, in our case, roses. However, not everything is solved since it is necessary to know the number of roses depending on their phonological state and for this a technique called ByteTrack was used that provides us with tools that facilitate the counting and tracking of a rose during its movement in the video provided.

KEYWORDS: Computer vision, ByteTracker, Object detection, Yolo, Rosas, Artificial Intelligence, Python, Artificial Neural Networks, CRISP-DM, DeepSort.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El objetivo de estudio es identificar los 2 primeros estados fonológicos de un tipo de rosa (garbanzo, raya color) y conocer la cantidad de rosas que hay por estado en una cama en específica.

Primero, se identificó tres tipos de rosas para realizar el estudio, los parámetros de selección se basaron en el tiempo que transcurre entre un corte anterior del producto al corte actual, más conocidos como periodo corto, medio y largo; otro parámetro es el color de los pétalos de la rosa rojo, rosado y fucsia y por último es el tamaño que crece una rosa, de igual manera se seleccionó el bajo, medio y alto.

En la primera visita a la florícola se colocó una etiqueta a la rosa que se encuentra en el primer estadio "botón" con la fecha de la visita, este proceso se hizo con 100 rosas. A la siguiente semana se cambió la fecha de la etiqueta y se contabilizó cuantas etiquetas se mantienen, en este caso se perdieron alrededor de 4 etiquetas por especie de rosa debido a que crecían más rosas o porque al estar en contacto con trabajadores se perdían o se desataban de la rosa. A la tercera semana se obtuvo una pérdida de alrededor del 50% de rosas debido a que varias rosas ya tenían que ser cortadas para salir a producción. Finalmente, a la cuarta semana se perdió alrededor de un 90% de rosas por motivos de que ya tenían que salir a producción.

Para la captura de imágenes, se utilizó las cámaras de los celulares de los integrantes del grupo, para esto se ajustó los parámetros de las cámaras para tener imágenes de un solo tamaño, se decidió sacar una foto por rosa con una dirección perpendicular al botón, luego se decidió recortar las imágenes para enfocar sólo el área de interés, en este caso el botón de la rosa, esta fue una mala decisión ya que al momento de entrenar el modelo obteníamos excelentes predicciones con imágenes pero al momento de pasar el modelo por un video en donde los botones de las rosas se encontraban más alejadas el modelo no tenía buenas predicciones, por esta razón se decidió obtener frames de videos capturados durante las 4 visitas técnicas y entrenar el modelo con imágenes muy similares a como se verían desde un video.

En el entrenamiento del modelo de IA, en primera instancia se decidió crear una red neuronal para entrenar el modelo con las imágenes clasificadas en seis clases 'explorer f1', 'explorer f2', 'komodo f1', 'komodo f2', 'pink f1' y 'pink f2' en donde f1 corresponde a garbanzo y f2 corresponde a raya color. En la evaluación del modelo se obtuvo buenas métricas de evaluación debido a la poca cantidad de datos ya se contaba con alrededor de 500 imágenes lo cual no era suficiente para entrenar un modelo, se decidió generar nuevas

imágenes con procesamiento de imágenes específicamente con data generator, sin embargo, el modelo no mejoró considerablemente.

En este punto de entrenamiento del modelo, se decidió utilizar técnicas de transfer learning para ajustar los parámetros de las redes neuronales convolucionales de un modelo ya pre entrenado, se experimentó con varios modelos pre entrenados para seleccionar el que mejor se acomode al caso de estudio. Los modelos seleccionados para experimentar y seleccionar el mejor fueron 3, YoloV5, YoloV7, YoloV8 específicamente para la detección de objetos, cabe mencionar que estos modelos prestan servicios para clasificar, segmentar y detectar objetos, en este caso se seleccionó el modelo para detectar ya que es el objetivo del proyecto.

Una vez creado el modelo se procede a desplegarlo utilizando técnicas de seguimiento para contar el número de rosas por clases detecta en un video en vivo en un archivo de video. Para realizar el seguimiento de los objetos se utilizó dos técnicas 'DeepSort' y 'ByteTrack' los cuales son compatibles con la versión del modelo pre entrenado, estas técnicas utilizan las coordenadas del objeto detectado y predice el lugar donde podría estar el objeto en el siguiente frame y así seguirle el rastro durante todos los frames.

1.1 Objetivo general

Desarrollar un modelo de Machine Learning para identificar los 2 primeros estados fonológicos de rosas.

1.2 Objetivos específicos

1. Estudiar y comprender el estado del arte de modelos de Machine Learning aplicados a la floricultura.
2. Utilizar técnicas de preprocesamiento de datos para el tratamiento de imágenes.
3. Evaluar varios modelos y seleccionar el mejor modelo diseñado
4. Diseñar un modelo de Machine Learning para identificar los 2 primeros estados fonológicos a partir de las imágenes preprocesadas.
5. Evaluar el modelo diseñado.
6. Diseñar un microservicio para el modelo de Machine Learning desarrollado.

1.3 Alcance

El componente será desarrollado en base a las etapas de CRISP – DM. [1]

1. Entendimiento del negocio

El desarrollo del proyecto comienza con el entendimiento del negocio, mismo que incluye la lectura de proyectos relacionados para conocer y comprender las técnicas de Machine Learning que son utilizadas en el sector. Además, la realización de una visita técnica a la empresa con la finalidad de comprender la línea de producción de rosas.

2. Entendimiento de los datos

Completada la anterior etapa, se realiza el entendimiento de los datos. Esta etapa involucra la proposición de medios para la captura de datos y la misma captura de las imágenes.

3. Preparación de los datos

En la siguiente etapa, se realiza la preparación de los datos. En esta etapa se incluyen actividades relacionadas con la manipulación del conjunto de imágenes capturadas tales como: redimensionamiento, rotación, interpolación y aplicación de filtros.

4. Modelado

En la etapa posterior, se construye un modelo de Machine Learning para identificar los 2 últimos estados fenológicos de las rosas. Esto incluye la selección de un modelo adecuado para el conjunto de imágenes preprocesadas, así como el establecimiento de la estrategia de verificación de calidad del modelo.

5. Evaluación del modelo

En la etapa final se realiza la evaluación del modelo planteado. Se verifica grado de acercamiento del modelo al objetivo planteado. De ser necesario se podrían repetir las fases anteriores a partir de lo establecido en la metodología que se utiliza.

6. Despliegue

Finalmente, se diseña un microservicio que, a través del modelo de Machine Learning desarrollado sea capaz de cumplir el objetivo general planteado a través de nuevos datos.

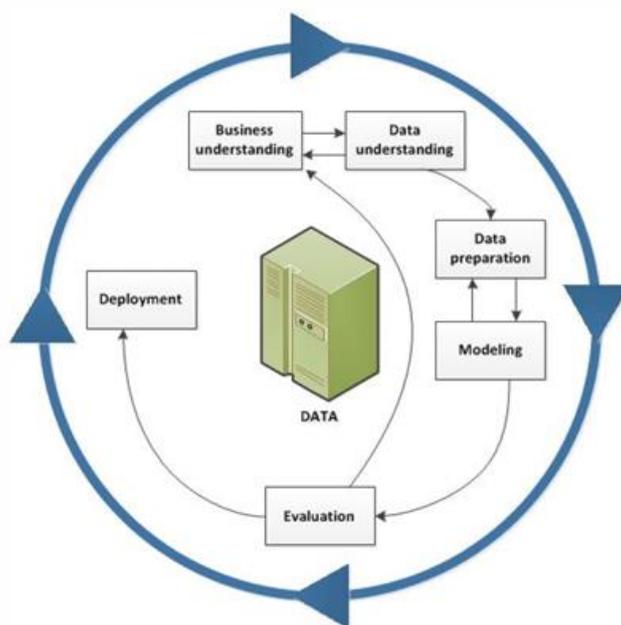


Figura 1.1: Metodología CRISP-DM

1.4 Marco teórico

1.4.1 Origen de las empresas florícolas en el Ecuador

La producción florícola en Ecuador surgió a finales de los años 70, pero el auge de la exportación de rosas inició en los años 80 en la provincia de Pichincha en las zonas de Cayambe y Tabacundo, zonas pioneras donde se ubicaron las primeras fincas de rosas más grandes. [2]

La primera empresa ecuatoriana en plantar claveles y crisantemos fue Jardines del Ecuador. El éxito obtenido por esta empresa motivo a otras empresas como la compañía de Rosas del Ecuador S.A. empezó a exportar rosas, por otro lado, empresarios éxitos empiezan a invertir dinero en el negocio de flores, generando fuentes de trabajo con la creación de fincas para la plantación de rosas, clavelines y claveles en Cayambe y Tabacundo. En el año de 1985 ya existían alrededor de 25 hectáreas cultivadas, especialmente de rosas. [3]

Las rosas son la especie más importante de la producción de flores en el País. Las rosas tienen alrededor de 500 variedades diferentes vigentes y se encuentra en la búsqueda o generación de nuevas variedades.

1.4.2 Exportación de flores en Ecuador

Datos publicados por Expoflores basándose en los datos recopilados por el Banco central del Ecuador indica que en el año 2019 el sector florícola generó alrededor de 879.8 millones

de dólares en exportaciones. Estados Unidos es el primer país más consumidor de flores ecuatorianas con 397.1 millones, el cual representa el 45% de la demanda. Rusia es el segundo país más consumidor con el 14.5%, en tercer lugar, se encuentra Países Bajos con el 8.6%. Italia, Ucrania, Canadá y España son países que también adquieren flores ecuatorianas, pero en bajo porcentaje en comparación a los países ya mencionados. [2]

1.4.3 Exportación total por año

En el periodo enero-mayo del año 2023, las exportaciones de flores ecuatorianas llegaron a superar los 440 millones de dólares, esta medida indica que el país superó en 2% a la cantidad de exportaciones que se alcanzó en el año 2022, en el mismo periodo. [4]

En la **Tabla 1.1**, comparando el año 2022 con el año 2023 se observar el incremento de dinero en exportaciones obtenidos en los periodos Enero-Octubre, las flores Naturales presentan un incremento de 30,1 millones de dólares, determinando que las flores naturales son unos de los productos de exportación no petrolera no tradicional en generar ingresos significativos para el País.

Tabla 1.1: Productos no petroleros exportados en los años 2022 y 2023.

EXPORTACIONES	Ene-Dic 2021	Ene-Dic 2022	% Var. Anual 2022	Var. Anual Abs. 2022 USD millones	Ene-Oct 2022	Ene-Oct 2023	% Var. Anual 2023	Var. Anua Abs. 2023 USD millones
	USD Millones	USD Millones			USD Millones	USD Millones		
PRODUCTOS DE EXPORTACIÓN NO PETROLERA NO TRADICIONALES	7.900,5	9.005,3	14,0%	1.338,1	7.477,3	7.990,0	6,9%	512,7
 Oro	572,6	827,6	44,5%	254,9	659,6	1.178,9	78,7%	519,2
 Concentrado De Plomo Y Cobre	920,6	1.288,2	39,9%	367,6	1.102,3	1.167,5	5,9%	65,1
 Enlatados De Pescado	1.262,5	1.366,6	8,2%	104,1	1.156,1	1.098,1	-5,0%	-58,0
 Flores Naturales	927,3	950,5	2,5%	23,2	787,5	817,6	3,8%	30,1
 Otros Productos Mineros	597,8	658,7	8,1%	294,1	548,0	591,0	7,8%	43,0
 Otros	3.619,7	3.913,8	8,1%	294,1	3.223,7	3.137,0	-2,7%	-86,7

1.4.4 Exportaciones por tipo de flor

Hasta mayo del 2023, el 76% de las exportaciones de flores las concentra las rosas como muestra la **Tabla 1.2**, este indicador muestra que Ecuador es el segundo País en exportar rosas a nivel mundial muy parejos con nuestro país vecino Colombia. [4]

Tabla 1.2: Especies de flores más exportadas de Ecuador

	Valor FOB Millones de dólares	Crecimiento Valor	Toneladas	Crecimiento Volumen	Participación en Valor
 Rosas	335	7% ▲	62.856	15% ▲	76%
 Flores de verano	50	-13% ▼	7.700	-10% ▼	11%
 Gypsophila	31	1% ▲	4.475	0,03% ▲	7%
 Claveles	9	-5% ▼	1.396	-19% ▼	2%
 Flores preservadas	8	-36% ▼	192	-31% ▼	2%
 Alstromeria	3	-5% ▼	703	5% ▲	1%
 Crisantemos	2	-22% ▼	403	-8% ▼	0,4%
 Aster	1	82% ▲	194	80% ▲	0,3%

1.4.5 Exportaciones de flores por país de destino

Como ya se mencionó anteriormente en el año 2019 los países más consumidores de flores ecuatorianas fueron Estados Unidos, Rusia, y Países Bajos. [2]

Para el año 2023, en el periodo de enero a mayo Estados Unidos se mantiene como el principal consumidor de rosas ecuatorianas seguido de Kazajistán, en tercer lugar, tenemos a Holanda, cuarto tenemos a Canadá e Italia y por último a Países de la Unión Europea como se observa en la **Tabla 1.3**.

Tabla 1.3: Principales destinos de exportación

	Valor FOB Millones de dólares	Crecimiento Valor	Toneladas	Crecimiento Volumen	Participación en Valor
 EE.UU.	143	-15% ▼	21.279	-14% ▼	32%
 Kazajistán	60	+100% ▲	11.671	94% ▲	14%
 Canadá	17	3% ▲	2.688	8% ▲	4%
 Chile	9	+100% ▲	1.613	65% ▲	2%
 UE	115	6% ▲	21.449	14% ▲	26%
 • Holanda	47	2% ▲	8.429	3% ▲	11%
 • Italia	17	-5% ▼	3.083	-1% ▼	4%
 • España	16	11% ▲	2.944	16% ▲	4%
 • Alemania	5	-6% ▼	969	1% ▲	1%

1.4.6 Variedad de rosas ecuatorianas

Las rosas es el producto más solicitado en diferentes partes del país y en el exterior debido a su hermosura, aroma, color, textura y por su gran variedad. La rosa es considerada como la reina de las flores ya que se las puede utilizar en varias ocasiones como ramos, arreglos florales y tiene una gran acogida en fechas festivas como por ejemplo día de la madre, día de los difuntos, San Valentín, es decir, es un producto muy comercial.

Debido a las extensas variedades de rosas, no se pudo estudiar a todas, pero en este caso se seleccionaron tres tipos de rosas, las cuales son: Explorer, Komodo y Pink.

Explorer: La variedad de rosas explorer es la más demandada en la actualidad debido a su impresionante color rojo, su tamaño y durabilidad, es una de las variedades más queridas e impresionantes ya que el color rojo tiene una representación significativa alrededor del mundo como es el amor, por lo tanto, la hace muy requerida por las personas. [5]

Características:

- Color: Rojo
- Tamaño del botón: 6.5 – 7 cm
- Largo del tallo: 50 – 90 cm
- Número de pétalos: 40
- Tiempo estimado de vida en un florero: 15 días.



Figura 1.2: Variedad de rosa Explorer.

Pink: La variedad de rosas Pink corresponde a un tipo de rosa que se caracteriza por el color verde en el contorno de los pétalos que se encuentran en el exterior del botón y con un color fucsia en el centro del botón, es muy popular por su color, textura y durabilidad. Es un tipo de rosa que no tiene un significado especial, por lo tanto, se lo puede utilizar en cualquier ocasión. [6]

Características:

- Color: Fucsia

- Tamaño del botón: 5.5 – 6 cm
- Largo del tallo: 40 - 70 cm
- Número de pétalos: 42
- Tiempo estimado de vida en un florero: 12 - 14 días.



Figura 1.3: Variedad de rosa Pink

Komodo: La rosa Komodo es una especie propia de la florícola donde se realizó el estudio, es una rosa que tiende a tener un color de Rosa lavanda, con un aroma muy especial y que crece muy rápido ya que es de ciclo corto.

1.4.7 Estados fonológicos de una rosa

A continuación, en la **Tabla 1.4** se muestra el cómo la rosa se desarrolla o crece en cada uno de sus estados fonológicos.

Tabla 1.4: Estados fonológicos de una variedad de rosas.

ESTADO	IMAGEN	TIEMPO	CARACTERISTICA
Arroz		24 días	Formación del botón.

Arveja		28 días	El botón aumenta su tamaño. Los sépalos empiezan a separarse
Garbanzo		33 días	El botón aumenta de tamaño, los sépalos se separan más y se empieza a observar el color de los pétalos.
Raya color		38 días	El botón aumenta de tamaño, los sépalos se separan más y se empieza a observar ya más segmentos de color.

Los estados fonológicos que se estudiaron en este caso son las 2 primeras fases del estadio raya color, en este caso el periodo de cambio entre estados era muy corto por lo que se optó identificar a los estados como garbanzo que sería su etapa final y empezando el estadio raya color, pero para que sea más visible y entendible el estado se muestra en la **Figura 1.4** donde solamente se le aprecia una sola línea de color en otros casos hasta dos líneas, pero muy delgadas. En este caso como la fase se encuentra en el estado final de garbanzo y raya color se optó por llamarle “botón”.



Figura 1.4: Estado fonológico determinado botón de la variedad Explorer

La siguiente fase que sería ya solamente raya color por esta razón se la llamó por su propio nombre, en la **Figura 1.5** se observa que los pétalos ya se encuentran más separados y se puede observar claramente el color de la rosa, sin embargo, sus pétalos aún siguen estando juntos.



Figura 1.5: Estado fonológico determinado raya color de la variedad Komodo

1.4.8 Inteligencia artificial en la producción de rosas

Reconocer y contar el número de total de productos por cada variedad que existe en la florícola es importante, ya que es un producto muy comercial y se desea aprovechar en fechas festivas como el Día de las Madres, Día de los difuntos y San Valentín, el dato sobre la cantidad de rosas es muy importante para los floricultores. [7]

Esta tarea es complicada para los trabajadores de la industria debido a las grandes alturas que puede llegar a medir una planta, por esta razón, les puede tomar horas en realizar el conteo de las rosas y clasificarlas. Para solucionar este problema, con el uso de algoritmos de Machine learning se puede generar un modelo de inteligencia artificial para que realice esta actividad en un menor tiempo posible, pero ¿Que es la inteligencia Artificial?

1.4.9 Inteligencia Artificial

La inteligencia artificial es un software o programa que tiene la capacidad de aprender a partir de un conjunto de datos recopilados a través del tiempo. Luego utiliza ese conocimiento para actuar ante una nueva situación, simulando a la toma de decisiones de los humanos. Científicos de datos buscan la manera de crear nuevos modelos que tengan la capacidad de leer textos, ver imágenes, crear videos y audios. [8]

1.4.10 Machine Learning

Es la ciencia que busca desarrollar modelos o algoritmos estadísticos que se utilizan en las computadoras para llevar a cabo tareas sin instrucciones basándose en patrones aprendidos de un conjunto de datos. Los modelos se crean a partir de patrones identificados en grandes conjuntos de datos almacenados históricamente utilizando algoritmos de machine learning, la creación de modelos requiere grandes capacidades de procesamiento y almacenamiento de datos. [9]

Aprendizaje supervisado

Es el tipo de aprendizaje que utiliza un conjunto de datos etiquetados para entrenar algoritmos que tengan la capacidad de clasificar un objeto de forma precisa. Mientras el algoritmo recibe más datos etiquetados el modelo va ajustando sus ponderaciones para tener una gran precisión al momento de clasificar un objeto. [10]

Aprendizaje no supervisado

El aprendizaje no supervisado utiliza un conjunto de datos no etiquetados para su entrenamiento. El algoritmo estudia el comportamiento de los datos para establecer conexiones entre entradas y salidas con el fin de detectar patrones y categorizar los datos. [9]

1.4.11 Aprendizaje profundo

Es una técnica de aprendizaje de machine learning que trata de simular el comportamiento de un cerebro humano. Este tipo de aprendizaje utiliza algoritmos para analizar los datos con una estructura muy similar a la que utiliza el cerebro humano. Las redes neuronales son sistemas inteligentes que utiliza el aprendizaje profundo para procesar información por capas. El flujo por el que atraviesan los datos empieza desde la capa de entrada conformado por varias redes neuronales profundas ocultas previo a la capa de salida. Las capas ocultas permiten un aprendizaje más eficaz en comparación a los modelos tradicionales del machine learning. [9]

1.4.12 Redes neuronales

Es un modelo de la inteligencia artificial que trata de simular el comportamiento de un cerebro humano para procesar datos en sistemas computacionales. Es un tipo de aprendizaje de machine learning conocido como Deep learning, este tipo de proceso utiliza los nodos que están conectados formando una estructura de capas muy parecido al cerebro humano. Genera un sistema en donde las computadoras aprenden de sus errores y mejoran continuamente. De esta manera, las redes neuronales tratan de resolver

problemas complejos como el reconocimiento de objetos, clasificación de objetos, redactar libros con mayor precisión. [11]

Una neurona básicamente funciona de la siguiente manera. La red neuronal recibe señales de entrada (*variables independientes*), las cuales se le multiplica a cada una con sus respectivos pesos obteniendo como resultado la función de ponderación la cual es una combinación lineal de las entradas y los pesos.

$$X * W^t = (x_1, x_2, \dots, x_n) * \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \sum_{i=1}^k x_i * w_i$$

Ecuación 1.1: Combinación lineal entre entradas de la red neuronal y los pesos

Luego se le aplica la función de activación.

$$\phi\left(\sum_{i=1}^k x_i * w_i\right)$$

Ecuación 1.2: Función de activación

Como se mencionó anteriormente, dicho valor puede ser la entrada de la siguiente neurona, así formando una red neuronal, el valor también puede ser el valor de respuesta. A continuación, en la **Figura 1.6** mediante un gráfico se puede comprender mejor el comportamiento de una red neuronal.

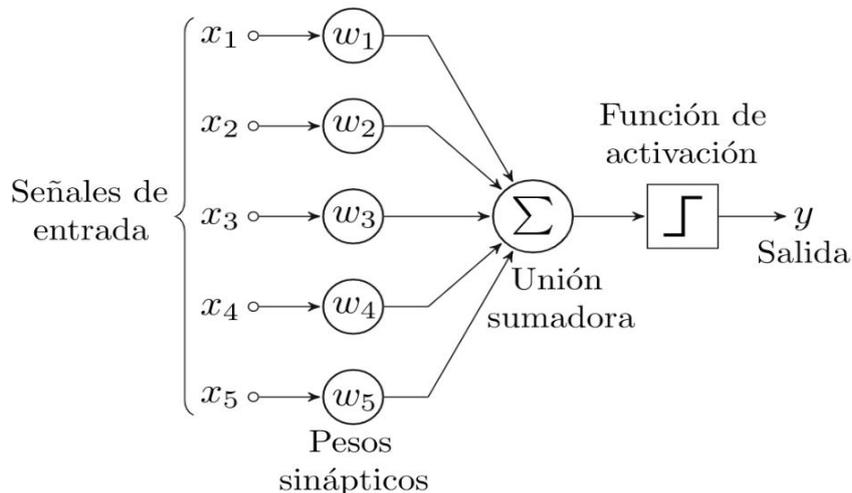


Figura 1.6: Esquema de una neurona artificial.

1.4.13 Redes neuronales convolucionales

Las redes neuronales convolucionales son diseñadas especialmente para trabajar con datos no estructurados como imágenes, textos, señales de audio, etc. Las redes neuronales son una versión mejorada de las redes neuronales inspiradas en como el

cerebro humano procesa la información, estas redes han demostrado ser extremadamente efectivas al momento de realizar tareas como reconocimiento de patrones. [12]

Una red neuronal convolucional se estructura con 3 capas de la siguiente manera:

Capas convolucionales.

Estas capas son el corazón de una red neuronal convolucional. El objetivo de esta capa es aplicar filtros para escanear la entrada y extraer características para ir aprendiendo. Cada filtro aplicado detecta patrones como bordes, vértices, colores o texturas.

Debido a su operación matemática convolución estas capas son conocidas como tal, esta operación se utiliza para diversas aplicaciones como en la estadística, probabilidad, visión por computados, procesamiento de texto (procesamiento natural de lenguaje).

El funcionamiento de una convolución consiste en tomar dos funciones y generar una nueva, la cuál expresa la modificación de una función sobre la otra, en este caso la primera función es la que experimenta el cambio que realiza la segunda, usualmente a estas funciones se las conoce como canal y kernel respectivamente.

$$d_{ij} = \frac{1}{c} \sum_{r=1}^{2N+1} a_{i-N+r, i-N+r-1}^{CT, S}$$

Ecuación 1.3: Expresión matemática de una convolución.

En la **Figura 1.7** se aprecia gráficamente el proceso de convolución utilizando una matriz de 3x3 como kernel.

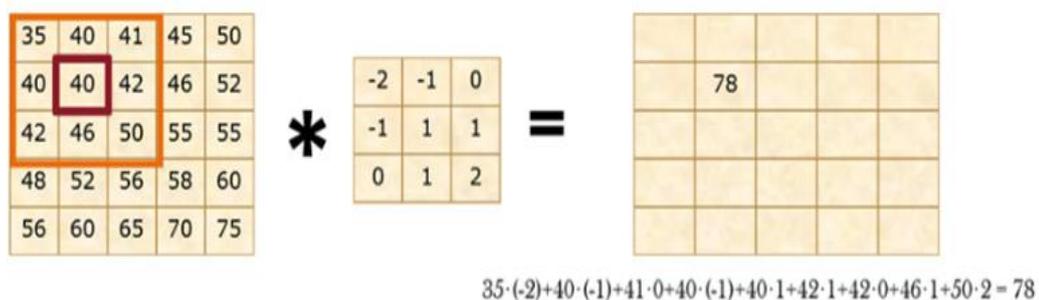


Figura 1.7: Ejemplo gráfico de una convolución.

La convolución sirve para extraer características de imágenes digitales, este proceso es conocido como filtrado de imágenes. La capa como tal describe la profundidad de la arquitectura. La convolución se divide en dos partes principales.

Capas de agrupación.

Conocidas también como capas pooling, estas capas reducen la dimensión de la representación y disminuyen la cantidad de parámetros en la red. Comúnmente, se utilizan operaciones de máximo o promedio para conservar características muy relevantes. [12]

Capas totalmente conectadas

Al final de la red, se encuentran capas que conectan todas las neuronas de la capa anterior, permitiendo aprender relaciones más complejas y realizar clasificaciones finales. [12]

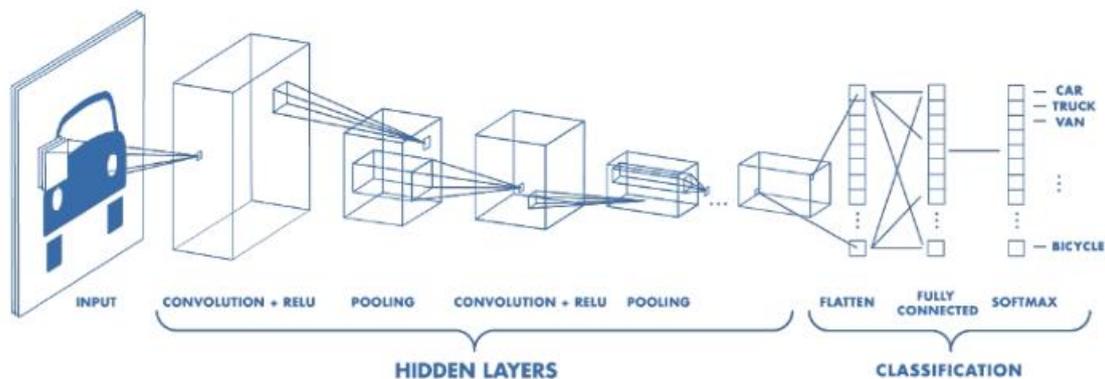


Figura 1.8: Arquitectura de una red neuronal convolucional.

1.4.14 Visión Artificial

La visión artificial es un modelo de aprendizaje profundo que brinda a las computadoras la capacidad de ver, observar y actuar de la misma manera que algoritmos de machine learning les permite pensar. [9]

Analizar, entender, extraer características de imágenes o videos son tareas que la visión artificial desea solventar utilizando algoritmos y técnicas de aprendizaje automático. La utilización de estos algoritmos sirve para generar modelos que tengan la capacidad de ver, detectar, clasificar, segmentar objetos dentro de una imagen o video, siendo más específicos sirven para tareas complejas como detectar rostros o detectar movimientos. [13]

1.4.15 YOLO

YOLO, “You Only Look Once” en español, “Sólo se mira una vez”, surgió con una idea muy llamativa para la detección de objetos. Anteriormente, la detección de objetos era una tarea muy compleja que se realizaba en dos pasos: seleccionar regiones y luego clasificarlas. YOLO simplificó el problema al transformarlo en un único problema de regresión. YOLO propone que el modelo procese toda la imagen una sola vez, lo cuál agiliza el proceso sin comprometer el valor de la precisión. [14]

Hace algunos años, el proceso que se empleaba para la detección de objetos era utilizar una red neuronal para clasificar objetos, después dividir la imagen en pequeños recuadros formando una malla, sobre la cual se iba pasando un rectángulo de dimensión definida por cada elemento de la malla e ir clasificando objetos.

Cada vez que el rectángulo se iba desplazando por cada casilla se iba detectando el objeto por partes hasta que finalmente encontramos las coordenadas del objeto completo como se observa en la **Figura 1.9**.

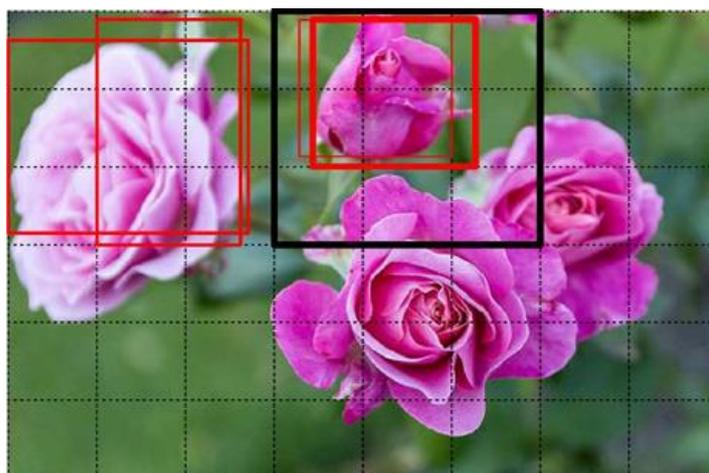


Figura 1.9: División de imágenes.

Esta es una técnica no muy eficaz, implica la detección del mismo objeto en varias partes de la imagen para determinar el área del objeto se necesita de un proceso posterior para determinar correctamente la posición del recuadro y realizar toda esta tarea consume muchos recursos y tiempo de ejecución.

Funcionamiento

Se trata de una arquitectura muy rápida que funciona muy bien para detectar objetos en videos streaming.

YOLO está formado por una red neuronal convolucional que utiliza una imagen completa para trazar cuadros llamados bounding boxes en las que se extrae características. Mediante este proceso se consigue que la red neuronal evalúe a todos los objetos que se encuentran en la imagen. A la imagen se la divide en pequeños recuadros que forman una matriz de tamaño $S \times S$ tal que, si un objeto se encuentra en dicha cuadrícula, esa celda es la encargada de detectar el objeto como se observa en la **Figura 2.10**, en este caso coincide con la casilla central.

Cada casilla de la imagen predice B cuadros delimitadores con sus respectivos grados de confianza. Las puntuaciones de confianza nos indica que tan confiable es el modelo para predecir si esa celda tiene un objeto y que tan preciso es al predecirlo. Si el valor de confianza es 0 quiere decir que en esa casilla no hay un objeto, de lo contrario, si hay un objeto en la casilla, la confianza se calcula con la siguiente formula: la intersección sobre la unión (IoU) un algoritmo que se explicará más adelante.

Cada casilla está compuesto por 5 elementos, x,y,w,h y la confianza. Las coordenadas x,y representa el centro de cada casilla. En la **Figura 1.10** el punto se encuentra marcado de color rojo, pero como se tiene que calcular en referencia a la celda que se está estudiando entonces se tiene que normalizar y así tener un balón entre 0 y 1 como se observa en los cálculos de la gráfica. Las coordenadas de altura y ancho no se normalizan en relación a la casilla sino se calcula en relación a la imagen completa. Por lo tanto, se toma el ancho y alto de la casilla y se normaliza con respecto al tamaño de la imagen.

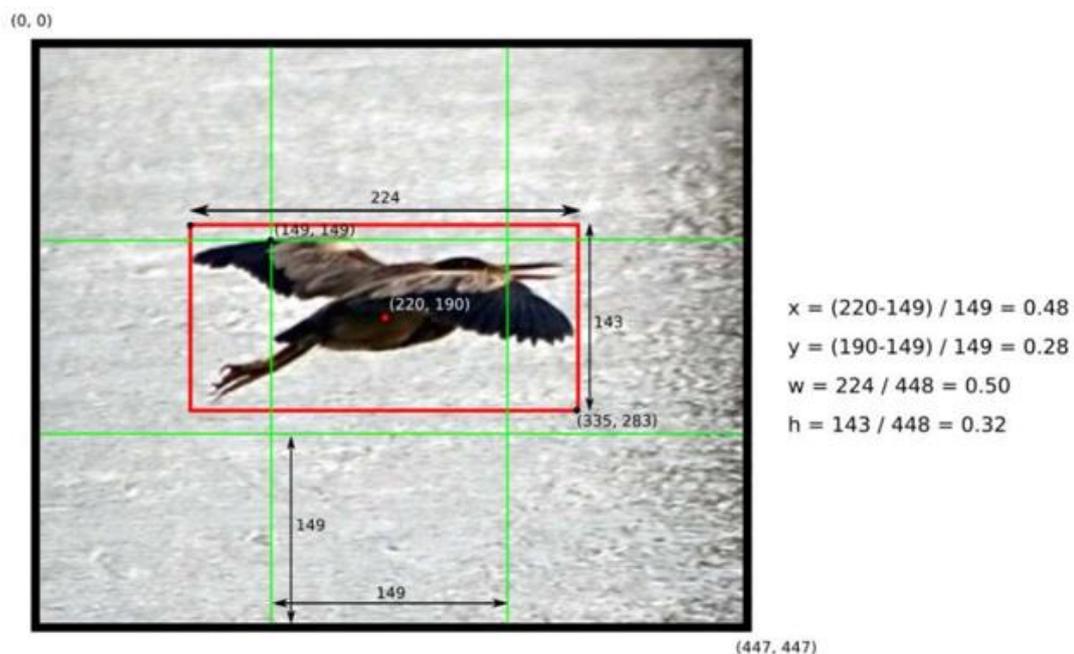


Figura 1.10: División de imágenes.

Finalmente, para calcular el valor de confianza de la detección se necesita varios parámetros. En primer lugar, se calcula el valor de la intersección sobre la unión del cuadro delimitador sobre la anotación que se realizó sobre la imagen como se observa en la **Figura 1.11** este cálculo nos da de resultado la intersección.

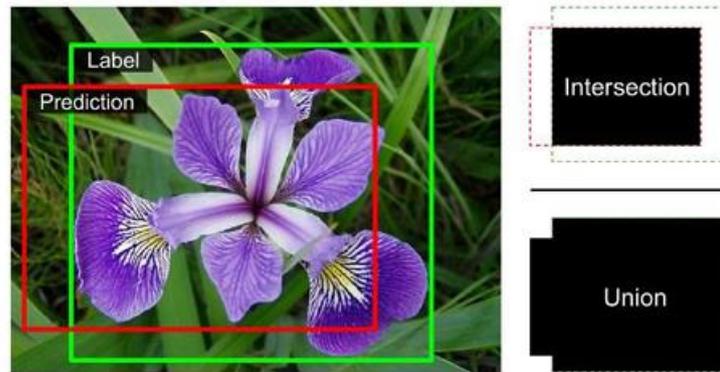


Figura 1.11: Intersección sobre la Unión de una imagen

Después de calcular la intersección, se determina la confianza mediante un conjunto de probabilidades. Cada casilla predice N probabilidades para clases condicionales, las cuales están atadas a la casilla que contiene el objeto y predice a un solo objeto sin importar el número cuadros delimitadores. En la detección, se multiplica la probabilidad de un cuadro individual por la probabilidad de un cuadro condicional generando así puntuaciones específicas para cada uno de los cuadros delimitadores. Este valor indica que tan probable es que un objeto se encuentre en esa casilla y que tan preciso es el cuadro para encerrar el objeto. [14]

Arquitectura de la red

El modelo de la red neuronal se crea a partir de una red neuronal convolucional, esto implica que sólo tiene capas convolucionales. A este tipo de red se la conoce como FCN o fully convolucional network, este tipo de red extrae características de la imagen en las capas iniciales y utiliza las capas completamente conectadas para predecir las probabilidades de que sea una de las clases y las coordenadas que encierran el objeto. [14]

La red está compuesta por un total de 24 capas convolucionales y 2 capas completamente conectadas. Para reducir la cantidad de capas, los creadores utilizan una convolución 1x1, lo que permite disminuir la profundidad de los mapas de características. A esta capa le sigue una capa de convolución 3x3. Estas capas 1x1 y 3x3 se van alterando como podemos apreciar en la **Figura 1.12**. Además, la última capa de convolución, genera un tensor de forma (7,7,1024). Posteriormente, el tensor se reduce mediante la aplicación de 2 capas completamente conectadas, resultando en un tensor de dimensiones 7x7x30 como salida final. [14]

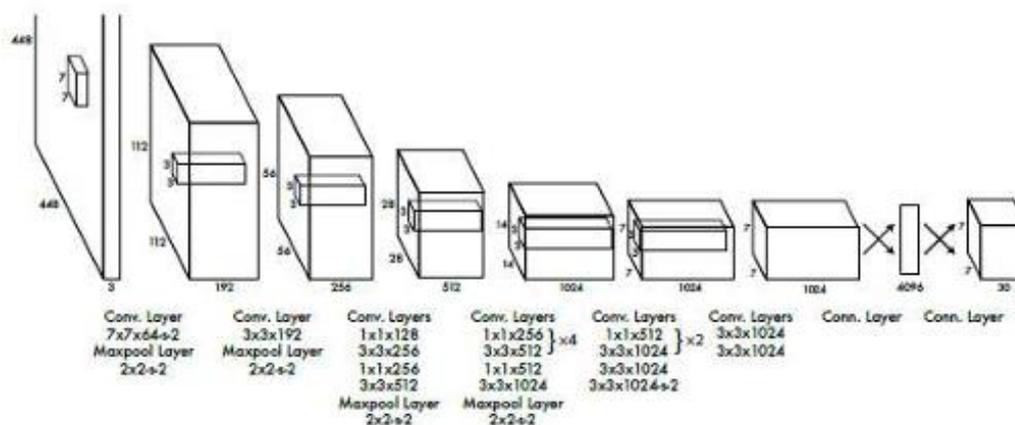


Figura 1.12. Esquema de la arquitectura de la red neuronal

En esta red la función de activación empleada es LeakyRelu, excepto en la capa lineal, que utiliza una capa de activación lineal. LeakyRelu modifica los valores de entrada multiplicando los negativos por un coeficiente para rectificarlos, mientras que los coeficientes positivos permanecen sin cambios. En esencia, esta función de activación permite una mayor flexibilidad al mantener la información en los valores positivos y corregir los valores negativos de manera controlada. [14]

En la función de pérdida de YOLO, se realizan predicciones de varios cuadros delimitadores (Bounding Boxes) para cada celda. Al calcular la pérdida para un verdadero positivo, se considera únicamente uno de los cuadros delimitadores posibles. Este cuadro se elige en base al cálculo de la intersección sobre la unión, seleccionando aquel que proporcione el resultado más alto con respecto al cuadro de anotación. Para calcular la pérdida, YOLO utiliza el resultado más alto con respecto al cuadro de anotación. Para calcular la pérdida, YOLO utiliza el error de suma cuadrada (*SSEC, Sum Squared Error*) entre los cuadros delimitadores de las detecciones y los cuadros delimitadores de las anotaciones. Este enfoque busca minimizar la discrepancia entre las predicciones y las anotaciones a través de la suma de los errores cuadráticos. [14]

Actualmente existen varias versiones de YOLO pero no todas pertenecen al mismo autor de la versión original, varias universidades y empresas sacaron una versión enfocados a un objetos de su interés como son autos autónomos, posición de humanos, etc. En este estudio se decidió utilizar las versiones del autor de la primera versión, es decir, las versiones de ultralytics especialmente en la versión 5 y versión 8 ya que son las versiones que cuentan con más documentación.

YOLOv8 es la última iteración de la serie YOLO para la detección de objetos en videos e en tiempo real, esta versión ofrece un rendimiento alto en términos de precisión y velocidad. Utilizando características de las versiones anteriores de YOLO, YOLOv8, esta vez introduce nuevas funciones y optimizaciones que lo convierten en la opción ideal para realizar diversas tareas de detección de objetos en una amplia gama de aplicaciones. En la **Figura 1.13** se observa la comparación entre las distintas versiones de YOLO, en donde se observa que la versión 8 de YOLO ofrece mayor rendimiento en términos de precisión y velocidad.

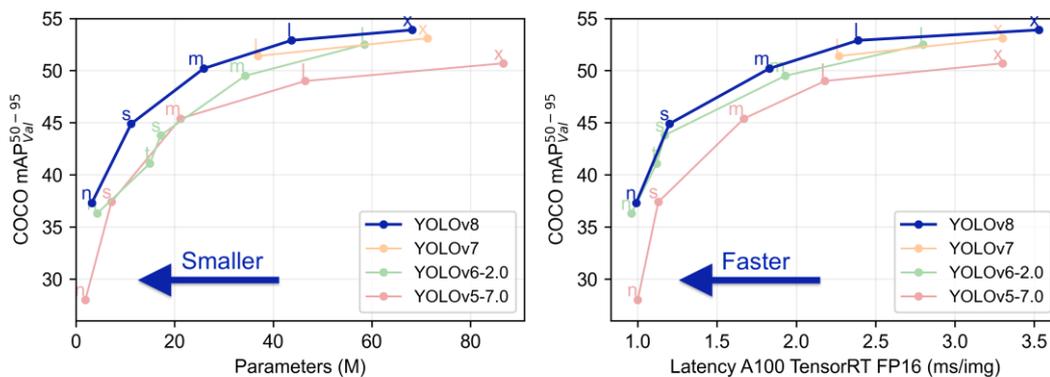


Figura 1.13: Diferentes modelos de Yolo pre entrenados

1.4.16 DeepSort

DeepSort es un algoritmo usado para el seguimiento de múltiples objetos en videos streaming. Es una extensión del algoritmo SORT (Simple Online and Realtime Tracking) el cual usa el filtro de Kalman para el seguimiento de objetos. Deep SORT incorpora una métrica de asociación profunda basada en características de apariencia aprendidas por una red neuronal convolucional profunda. Esto permite que el algoritmo maneje situaciones en las que los objetos pueden desaparecer temporalmente o quedar ocluidos en la transmisión de video. También incorpora la asignación de ID para rastrear objetos individuales en múltiples fotogramas. [15]

1.4.17 ByteTrack

ByteTrack se erige como una fuerza pionera en el ámbito de la visión por computadora, diseñada específicamente para la compleja tarea de seguimiento de objetos múltiples (MOT). Este innovador algoritmo de IA no es solo una herramienta, sino un revolucionario, diseñado para asignar identificadores únicos a objetos dentro de un video, permitiendo así el seguimiento consistente y preciso de cada objeto a lo largo del tiempo. [16]

2 METODOLOGÍA.

El componente será desarrollado en base a las etapas de CRISP – DM. [1]

2.1 Entendimiento del negocio.

La agricultura moderna evoluciona en varias direcciones al mismo tiempo. Sin embargo, su principal objetivo es hacer uso de las nuevas tecnologías para el sector agrícola con el fin de aumentar el rendimiento de los cultivos mediante una mejor planificación y una gestión más inteligente. [17]

En el sector florícola es importante contabilizar las cabezas florales debido a la fuerte competencia internacional, dicho conteo permite a los productores tener una gran visión sobre estrategias para obtener un cultivo más provechoso con la finalidad de tener menos gastos energéticos, químicos y gastos innecesarios.

La cuantificación de las flores permite tener conocimiento de la productividad obtenida además de ayudar a generar un prospecto económico de las ganancias al momento de vender la cosecha. Tanto para grandes como pequeños productores, es requerido realizar la cuantificación de sus cabezas florales, siendo esto un control tangible que se sugiere realizar en diferentes estados del botón floral debido a la alta demanda en cantidad y calidad que se tiene en el mercado floral.

En la actualidad no existe una herramienta gratuita que brinde la ayuda para realizar este tipo de trabajo, sin embargo, mediante este estudio se busca la manera de implementar un modelo de Inteligencia Artificial que realice esta tarea de detectar, clasificar y contabilizar cuantas cabezas florales existen por cada estado fonológico en varias variedades de rosas.

El conteo de activos biológicos es indispensable como parámetro a partir del cual se establecen las acciones de mantenimiento, monitoreo o se realiza la toma de algunas decisiones en la cadena de producción.

Esta información es importante conocerla para tener una idea de cómo el cultivo se está desarrollando, esta información es valiosa ya que se puede utilizar para estimar la productividad de la florícola, es decir, saber cuántas rosas podrían estar listas para una cosecha inmediata, o de igual forma ayuda al plantear futuras decisiones sobre el resto cabezas florales dependiendo sus estados fonológicos.

Entonces el objetivo es crear un modelo de inteligencia artificial que realice esta tarea, el proceso que se lleva a cabo es: con la ayuda un dron capturar un video en tiempo real paralelo a la cama donde se encuentran las plantas, extraer fotogramas del video y

evaluarlos en el modelo creado con inteligencia artificial, este modelo detecta una cabeza florar y lo clasifica según su estado fonológico, a continuación se realiza el seguimiento a la rosa detectada para que cuando llegue a la zona central del fotograma se proceda a contabilizar la detección según el estado identificado. Finalmente, en las esquinas superiores izquierda y derecha se muestra el conteo en tiempo real de cada estado fonológico. Cabe mencionar que las detecciones in y out dependen del sentido del video, si el video se mueve de derecha a izquierda se lo toma como **out**, en cambio si el video de izquierda a derecha se lo toma como **in**.

2.2 Entendimiento de datos.

Después de las indicaciones proporcionadas por el personal encargado del cuidado de las plantas se procedió a la recolección de datos, en este caso, la empresa no contaba con un almacén de datos propio por lo tanto tuvimos que recolectar los datos personalmente.

Antes de iniciar con el proceso de recolección se menciona algunas de las herramientas y materiales que se utilizó para dicho proceso.

- Celular Samsung Galaxy A30 con una cámara dual trasera de 16 Mpx + 5 Mpx con la que se realizó la captura de datos.
- Celular Xiami Redmi 8 con cámara trasera de 12 MPx + 2 MPx.
- Celular Xiaomi Redmi Note 11 con cámara Cuádruple, 50 MP + 8 MP + 2 MP + 2 MP.
- Soportes para teléfonos celulares.
- Libretas.
- Etiquetas con la fecha de la visita.

Para la recolección de datos necesitamos realizar un seguimiento al desarrollo o crecimiento de la rosa, para esto se etiquetó a un conjunto de rosas en específico, se creó 100 etiquetas de papel, las cuales se colocaron dentro de una funda plástica para que el papel no pierda o no sea visible la información colocada debido a que la etiqueta estaba expuesta a fertilizantes químicos que utiliza el personal de la empresa para el cuidado y crecimiento de las rosas. Al realizar estas tareas de cuidado las etiquetas estaban expuestas a extraviarse debido al contacto físico que había entre la planta y el personal de cuidado.

La recolección de datos se la realizó en 6 visitas técnicas a la florícola donde se realizó el estudio. En la primera visita se seleccionó un tipo de rosa de ciclo corto, esta variedad de rosas se la conoce como KOMODO, una especie propia de la florícola, de esta variedad se seleccionó 100 rosas que se encontraban en el inicio del estado fonológico “raya color”, después, se precedió a colocar una etiqueta en el tallo de la rosa para llevar un control en la próximas visitas y un tener un registro de cuantas rosas fueron etiquetadas por cada cama del invernadero y ver como se desarrolló en el transcurso del tiempo.



Figura 2.1: Etiqueta estado fonológico raya color variedad KOMODO.

La segunda visita técnica se la realizó una semana después. En esta visita, se trató de encontrar todas las etiquetas colocadas en la visita previa, sin embargo, como se lo tenía pensado, debido a la interacción que hay entre las plantas y los trabajadores se obtuvo una pérdida de 4 etiquetas. Por otro lado, se seleccionó la siguiente variedad de rosas, en este caso se seleccionó una era el turno de una especie de ciclo medio, para lo cual se seleccionó la variedad PINK, de la misma manera que en la visita previa se procedió a seleccionar 100 rosas que se encontraban en el inicio del estado fonológico “raya color”, después, se precedió a colocar una etiqueta en el tallo de la rosa para llevar un control en la próximas visitas y un tener un registro de cuantas rosas fueron etiquetadas. En esta visita es importante mencionar que, al ser una especie de ciclo medio, la altura de esta variedad llega a medir alrededor de 1.80m por lo que se dificulta la recolección de datos ya que no se podía enfocar el botón de la rosa y la imagen no era la adecuada, entonces se optó por sacar un video de la rosa para después con el procesamiento de imágenes obtener frames que nos sirve para el entrenamiento



Figura 2.2: Etiqueta estado fonológico raya color variedad PINK.

La tercera visita se realizó la siguiente semana, al igual que en la anterior visita se procedió a contabilizar el número de etiquetas en las variedades de rosas anteriormente etiquetadas en las cuales se obtuvo que en la variedad KOMODO se perdió alrededor de 6 etiquetas mientras que en la variedad PINK se perdió 13 etiquetas por motivos anteriormente mencionados, después se procedió a capturar imágenes de las rosas etiquetadas y capturar un video de las camas donde fueron etiquetadas las rosas. En la misma visita se procedió a seleccionar la última variedad de rosas, en este caso fue la variedad EXPLORER de ciclo largo, siguiendo el mismo procedimiento, se seleccionó un conjunto de 100 rosas, se las etiquetó y se contabilizó cuantas rosas fueron etiquetadas por cama. En este caso, la planta de la variedad EXPLORER medía alrededor de 2.10m lo cual causaba dificultades la identificación y captura de imágenes de esta variedad, pero al tener el mismo problema en la visita anterior se logró conseguir sancos para que un integrante del grupo pudiera elevar su estatura y se logre obtener datos de mejor calidad.



Figura 2.3: Etiqueta estado fonológico raya color variedad EXPLORER.

La cuarta visita ya no se seleccionó una variedad de rosas ya que nuestro estudio se enfocaba solamente en 3 variedades. En esta visita se contabilizó cuantas etiquetas fueron extraviadas, pero en este caso la pérdida ya sería significativa en la variedad KOMODO

debido a que ya es tiempo de que las rosas salgan a producción, en esta visita se perdieron 86 etiquetas, en la variedad PINK se perdieron 41 etiquetas, mientras que en la variedad EXPLORER se perdieron 12 etiquetas. De igual manera se procedió a capturar imágenes de las rosas etiquetas y un video de las camas donde se encontraban las rosas etiquetadas.

En la quinta visita, ya no se encontró etiquetas en la variedad KOMODO debido a que todas las rosas tuvieron que salir a producción al igual que la variedad PINK, las únicas etiquetas que se encontraron fue en la variedad EXPLORER donde se perdieron 10 etiquetas. De igual manera se procedió a capturar imágenes de las rosas etiquetas y un video de las camas donde se encontraban las rosas etiquetadas.

Finalmente, en la última visita se contabilizó el número de etiquetas de la especie EXPLORER donde perdió 36 etiquetas. Al igual que en las visitas previas se procedió a capturar las imágenes y videos de las rosas.

Todas las imágenes y videos capturados se colocaron en una carpeta compartida en la plataforma OneDrive, las imágenes y videos fueron colocadas dentro de una carpeta con el nombre de la variedad, a su vez una subcarpeta con el número de la visita y otra subcarpeta donde se encuentra el número de la cama para realizar un mejor seguimiento de la captura de datos.



Figura 2.4: Repositorio de datos

2.3 Preparación de los datos.

En esta etapa ya con los datos recolectados y organizados según su clase en la nube. Los datos fueron tratados para diferentes tipos de entrenamiento.

2.3.1 Data Image Generator

En primera instancia se pensó en entrenar una red neuronal convolucional desde cero y clasificar las imágenes de rosas enviados al modelo. Para ello el procedimiento fue el siguiente. Se descargó el conjunto de imágenes colocados en el OneDrive, se procedió a recortar las imágenes originales y extraer solamente la zona de estudio que en este caso sería el botón de la rosa como tal, a su vez con la ayuda de librerías de Python se procedió

a incrementar el conjunto de datos variando el contraste de las imágenes, cambiando el ángulo, haciendo un zoom a la imagen o alejándose, redimensionando las imágenes, etc.

La plataforma que se utilizó para el incremento de datos fue TensorFlow, el cual permite realizar varias operaciones con extensos conjuntos de datos y sirve para agilizar la creación de modelos de aprendizaje automático, una de las librerías de Tensorflow que se utilizó fue DataImageGenerator el cual es usado para realizar operaciones a un conjunto de datos y devolverlos con las nuevas transformaciones como se observa en la **Figura 2.5**.



Figura 2.5: Imagen generada con DataImageGenerator.

Las imágenes generadas y las imágenes originales se las clasificó en 6 clases Pink-f1, Pink-f2, explorer-f1, explorer-f2, komodo-f1, komodo-f2 y se las dividió en dos subcarpetas train y val en donde para train se almacenó el 90% de imágenes y en val el 10%. El repositorio para almacenar las imágenes fue GoogleDrive ya que brindaba la facilidad de enlazarse con GoogleColab.

Sin embargo, la cantidad de imágenes no fueron suficientes para tener un modelo que cumpla con los objetivos planteados y se decidió optar por otra técnica de entrenamiento, en este caso se hizo uso de modelos ya pre entrenados, pero en esta ocasión ya no para clasificación de rosas sino para detección de rosas.

2.3.2 Make Sense

Para la detección de objetos el etiquetado se lo realizó con la ayuda de Make Sense IA que es una plataforma web que permite realizar varias operaciones con datos, entre estas operaciones se encuentra el etiquetado para la detección de objetos conocido como “bounding boxes”, esta herramienta retorna archivos en distintos formatos como YOLO, COCO, VOC según la necesidad de estudio. En este caso se optó por el formato YOLO el

cual nos retorna un archivo de texto que contiene el identificador de la clase y las coordenadas del bounding box del objeto.

La vista de la herramienta Make Sense IA se observa en la **Figura 2.6**.

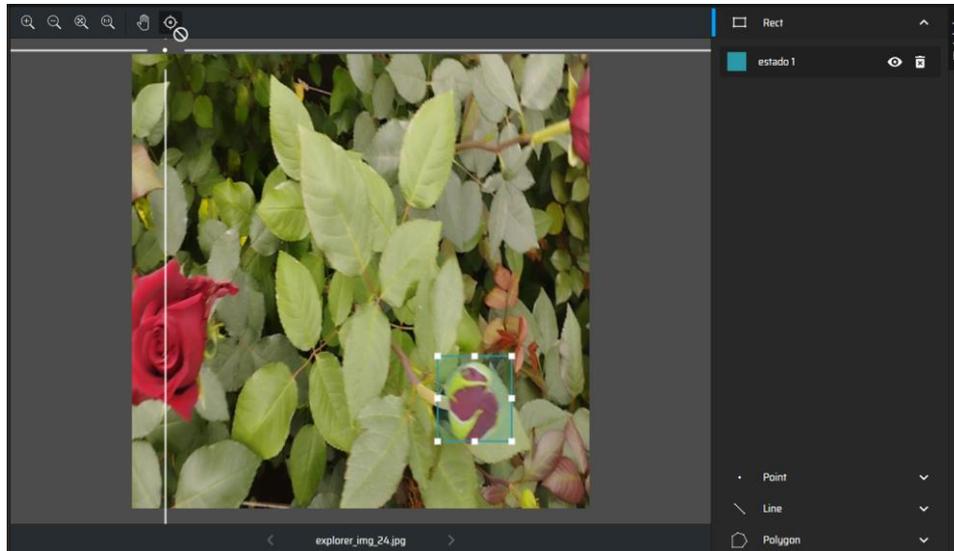


Figura 2.6: Herramienta Make Sense IA para el etiquetado de datos.

El resultado de la etiqueta se muestra en la **Figura 2.7** en donde el primer valor corresponde a la clase etiquetada y los cuatro valores de la derecha corresponden a las coordenadas Xmax, Xmin, Ymax, Ymin del objeto, cabe mencionar que en una imagen pueden haber varios objetos de diferentes clases por lo tanto se registran todas las etiquetas en el mismo archivo .txt. El mismo procedimiento se llevó a cabo con todo el conjunto de datos, ya sean las imágenes originales y las imágenes generadas con DataImageGenerator.

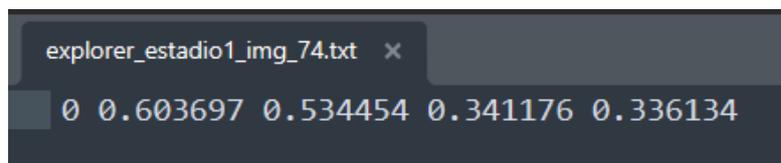


Figura 2.7: Etiqueta de datos en formato YOLO.

Una vez terminado el etiquetado de datos, se crea un archivo de texto con extensión yaml. Los ficheros YAML se los utiliza para declarar datos de manera que los usuarios tengan la capacidad de escritura. Este formato se encarga de almacenar archivos de configuración que se los puede utilizar en conjunto a lenguajes de programación [18]

Previo a la escritura en el fichero yaml se creó 2 carpetas “Imágenes” y “Labels”. La carpeta imágenes contiene 2 subcarpetas llamadas train y val en donde se almacenan imágenes para validación y el entrenamiento, de la misma manera en la carpeta Labes se creó la

carpeta train y val los archivos de texto respectivamente. Por otro lado, en el contenido del archivo de configuración se colocó el nombre de las clases, el número de clases y la dirección de las carpetas donde están ubicados las imágenes y las etiquetas de datos como se muestra en la **Figura 2.8**.

```

custom.yaml
names:
  - 'boton'
  - 'raya-color'
nc: 2
train: /content/drive/MyDrive/tesis/dataset/rosas-todo-tipo/images/train
val: /content/drive/MyDrive/tesis/dataset/rosas-todo-tipo/images/val

```

Figura 2.8: Configuración del fichero yaml

En este caso de igual manera se utilizó como repositorio de datos a Google Drive pero en este caso la distribución fue diferente; como cada archivo txt tenía el registrada la clase a la cual pertenecía la imagen no hacía falta dividir las sino unir todas las imágenes y dividir un porcentaje para validación y otro porcentaje para entrenamiento.

2.4 Modelado

El modelo se lo realizó utilizando las herramientas de Google ya que son gratuitas y fáciles de utilizar. Para el desarrollo del script se utilizó Google Colab mientras que Google Drive se utilizó como repositorio para almacenar las imágenes que fueron utilizadas para entrenar y evaluar el modelo, videos y archivos de configuración.

2.4.1 Modelo utilizando redes neuronales convolucionales.

Después de analizar la necesidad del negocio se optó por crear un modelo de Inteligencia Artificial utilizando una Red Neuronal Convolucional con la siguiente arquitectura como se muestra en la **Tabla 2.1**.

Tabla 2.1: Arquitectura red neuronal convolucional

Capa	Número de filtros / Neuronas	Tamaño kernel	Función de activación	Dropout Rate	Dimensión de salida
Conv2D	32	(3,3)	ReLU	N/A	(200,200,32)
MaxPooling2D	N/A	(2,2)	N/A	N/A	(100,100,64)
Conv2D	64	(3,3)	ReLU	N/A	(100,100,64)
MaxPooling2D	N/A	(2,2)	N/A	N/A	(50,50,64)
Conv2D	128	(3,3)	ReLU	N/A	(50,50,128)
MaxPooling2D	N/A	(2,2)	N/A	N/A	(25,25,128)
Conv2D	128	(3,3)	ReLU	N/A	(25,25,128)

MaxPooling2D	N/A	(2,2)	N/A	N/A	(12,12,128)
Dropout	N/A	N/A	N/A	0.5	N/A
Flatten	N/A	N/A	N/A	N/A	(18432,)
Dense	256	N/A	ReLU	N/A	(256,)
Dense	6	N/A	Softmax	N/A	(6,)

- Capa Conv2D (32 filtros): La red comenzó con una capa Conv2D que aplicó 32 filtros de convolución con un tamaño de kernel de (3,3) y una función de activación ReLU. Esta capa procesó datos de entrada con una forma de (200,200,3).
- Capa MaxPooling2D: A esta capa le siguió una capa de MaxPooling2D con un tamaño de pool de (2,2), reduciendo la dimensionalidad espacial de la salida de la capa anterior.
- Capa Conv2D (64 filtros): Posteriormente, se incluyó otra capa Conv2D con 64 filtros y una función de activación ReLU, seguida por otra capa de MaxPooling2D.
- Capa Conv2D (128 filtros): Se repitió el proceso con una tercera capa Conv2D que aplicó 128 filtros, seguida de una capa de MaxPooling2D.
- Capa Conv2D (128 filtros): Luego, se añadió una cuarta capa Conv2D con 128 filtros y una función de activación ReLU, seguida de otra capa de MaxPooling2D.
- Capa Dropout (0.5): Para mitigar el sobreajuste, se incorporó una capa Dropout que desactivó aleatoriamente el 50% de las conexiones entre las capas.
- Capa Flatten: Después de las capas convolucionales y de pooling, se utilizó una capa Flatten para transformar la salida tridimensional en un vector unidimensional.
- Capa Dense (256 neuronas): Se agregó una capa totalmente conectada (Dense) con 256 neuronas y una función de activación ReLU.
- Capa Dense (6 neuronas): La red concluyó con una capa Dense de salida con 6 neuronas y una función de activación softmax, indicando la clasificación multiclase.

El modelo fue compilado utilizando el optimizador 'adam', la función de pérdida 'sparse_categorical_crossentropy' y midiendo la precisión como métrica.

Es importante mencionar que en la capa final se utilizó 6 neuronas debido a que inicialmente se buscaba clasificar la clase y el estado fonológico de la rosa basándose en el color y la forma de los sépalos y pétalos de la rosa como se explicó en la Fase 2 preparación de los datos, sin embargo, no fue buena idea ya que no habían los suficientes

datos para que el modelo pueda aprender bien todas las características, por tal motivo se decidió modificar las clases y solamente enfocarse en la forma de los sépalos y pétalos de la rosa sin importar el color y así reducir el número de clases y detectar solamente la clase botón y raya color.

2.4.2 Modelo utilizando YOLO.

Como la versión 8 de YOLO es la más actual e implementó el tracking que simplemente es el seguimiento de variables, es la versión adecuada para lograr los objetivos planteados inicialmente.

Para el entrenamiento del modelo se seleccionó un modelo de Yolov8 ya que en esta versión hay variaciones dependiendo los parámetros como se muestra en la.

Tabla 2.2: Métricas de rendimiento de los diferentes modelos de YOLO

Modelo	tamaño (píxeles)	mAPval 50-95	Velocidad CPU ONNX (ms)	Velocidad A100 TensorRT (ms)	parámetros (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

El entrenamiento del modelo fue muy sencillo ya que Yolo presenta una guía muy amigable para realizar el entrenamiento con nuestro propio conjunto de datos.

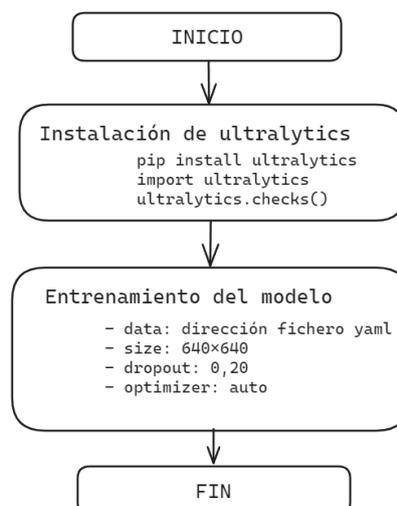


Figura 2.9: Pasos para entrenar un modelo con YOLO.

El primer paso fue instalar las dependencias de ultralytics y comprobar que la ejecución del notebook de Google se lo realice utilizando una GPU ya que caso contrario no se podía continuar con el entrenamiento.

El entrenamiento del modelo se realizó con el comando yolo y se especificó la configuración del entrenamiento mediante argumentos. Además, se observa la arquitectura de la red neuronal en la **Figura 2.10** utilizada para el entrenamiento del modelo y al final un resumen de todo el modelo.

```

Overriding model.yaml nc=80 with nc=2

      from  n  params module                    arguments
0         -1 1    2320 ultralytics.nn.modules.conv.Conv      [3, 80, 3, 2]
1         -1 1   115520 ultralytics.nn.modules.conv.Conv      [80, 160, 3, 2]
2         -1 3   436800 ultralytics.nn.modules.block.C2f      [160, 160, 3, True]
3         -1 1   461440 ultralytics.nn.modules.conv.Conv      [160, 320, 3, 2]
4         -1 6   3281920 ultralytics.nn.modules.block.C2f      [320, 320, 6, True]
5         -1 1   1844480 ultralytics.nn.modules.conv.Conv      [320, 640, 3, 2]
6         -1 6  13117440 ultralytics.nn.modules.block.C2f      [640, 640, 6, True]
7         -1 1   3687680 ultralytics.nn.modules.conv.Conv      [640, 640, 3, 2]
8         -1 3   6969600 ultralytics.nn.modules.block.C2f      [640, 640, 3, True]
9         -1 1   1025920 ultralytics.nn.modules.block.SPPF     [640, 640, 5]
10        -1 1          0 torch.nn.modules.upsampling.Upsample   [None, 2, 'nearest']
11        [-1, 6] 1          0 ultralytics.nn.modules.conv.Concat     [1]
12        -1 3   7379200 ultralytics.nn.modules.block.C2f      [1280, 640, 3]
13        -1 1          0 torch.nn.modules.upsampling.Upsample   [None, 2, 'nearest']
14        [-1, 4] 1          0 ultralytics.nn.modules.conv.Concat     [1]
15        -1 3   1948800 ultralytics.nn.modules.block.C2f      [960, 320, 3]
16        -1 1   922240 ultralytics.nn.modules.conv.Conv      [320, 320, 3, 2]
17        [-1, 12] 1          0 ultralytics.nn.modules.conv.Concat     [1]
18        -1 3   7174400 ultralytics.nn.modules.block.C2f      [960, 640, 3]
19        -1 1   3687680 ultralytics.nn.modules.conv.Conv      [640, 640, 3, 2]
20        [-1, 9] 1          0 ultralytics.nn.modules.conv.Concat     [1]
21        -1 3   7379200 ultralytics.nn.modules.block.C2f      [1280, 640, 3]
22        [15, 18, 21] 1 8719894 ultralytics.nn.modules.head.Detect     [2, [320, 640, 640]]

Model summary: 365 layers, 68154534 parameters, 68154518 gradients, 258.1 GFLOPs

```

Figura 2.10: Arquitectura utilizada para el entrenamiento del modelo

Para entrenar el modelo se seleccionó Yolov8x, los datos para el entrenamiento y validación se los envió a través del fichero yaml, las dimensiones de las imágenes de 640px x 640px que es requisito del modelo seleccionado, las épocas fueron variando entre 60, 80 y 100 para visualizar cual modelo obtiene mejor las métricas, de igual manera, los optimizadores utilizados fueron 'Adam', 'SGD' y 'auto' y el dropout varió de 0.20 a 0.50 para finalmente seleccionar el mejor modelo según sus métricas de evaluación.

2.5 Evaluación

Para evaluar el entrenamiento del modelo se utilizaron varias métricas que nos proporciona Yolov8, entre ellas está la matriz de confusión, la cual permite analizar los resultados de cómo funciona un algoritmo de aprendizaje supervisado. La matriz es una tabla que presenta en las columnas el número de predicciones de cada clase, mientras que cada fila muestra el número real de instancias por cada clase como se muestra en la **Figura 2.11**. Sin embargo, en la matriz de confusión que nos presenta Yolov8 se agrega una clase

llamada “background”, la cual representa los falsos positivos, es decir detecta un objeto donde no lo hay.

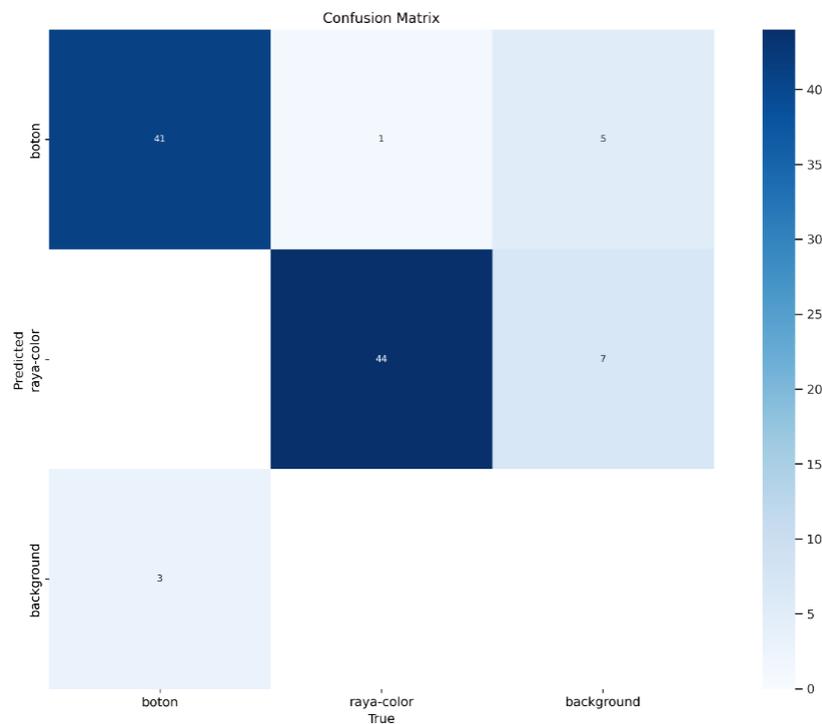


Figura 2.11: Matriz de confusión del modelo creado con Yolov8x.

Curva Precisión-Recuperación: Esta curva, que es una visualización integral de cualquier problema de clasificación, muestra las compensaciones entre precisión y recuperación con distintos umbrales. Es especialmente importante cuando se trata de clases desequilibradas.

Curva de precisión: En la **Figura 2.12** se observa como la precisión va aumentando a medida que la confianza al detectar una clase aumenta el cual es un buen indicador, otra observación importante es que al tener una confianza de 0.4 el modelo supera la precisión de 80%, el cual quiere decir el modelo tiene un error mínimo al momento de detectar un objeto donde no lo hay, sin embargo, por la cantidad de datos que hubo es poco probable que el modelo aprenda de la mejor manera y esto puede causar problemas en negocios que necesitan saber el valor de verdad pero es un buen indicador a pesar de los inconvenientes.

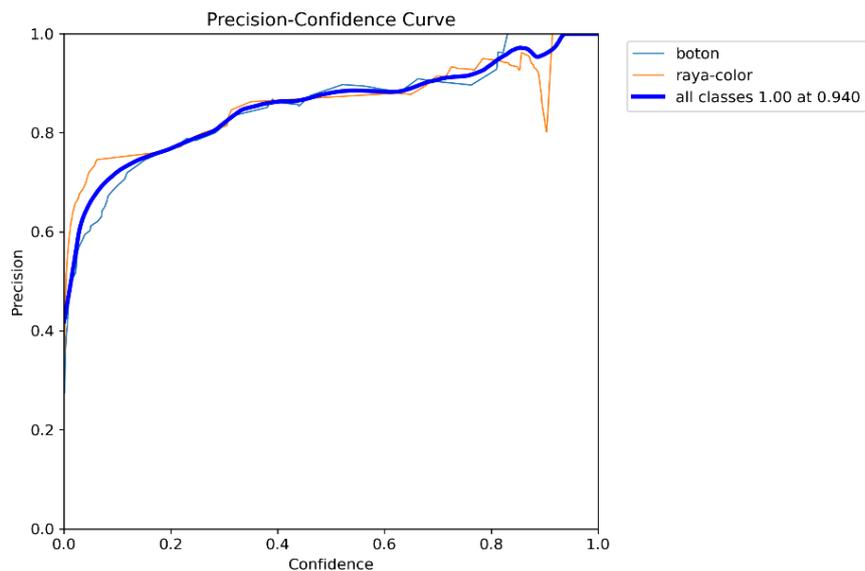


Figura 2.12: Curva precision_confidence creado con YOLOv8x.

Curva de recuperación: En la **Figura 2.13** se observa como la recuperación del modelo disminuye al momento de aumentar la confianza, esto quiere decir el modelo va a pasar por alto algunas cabezas florales, es decir, no va a detectar las rosas cuando si la hay.

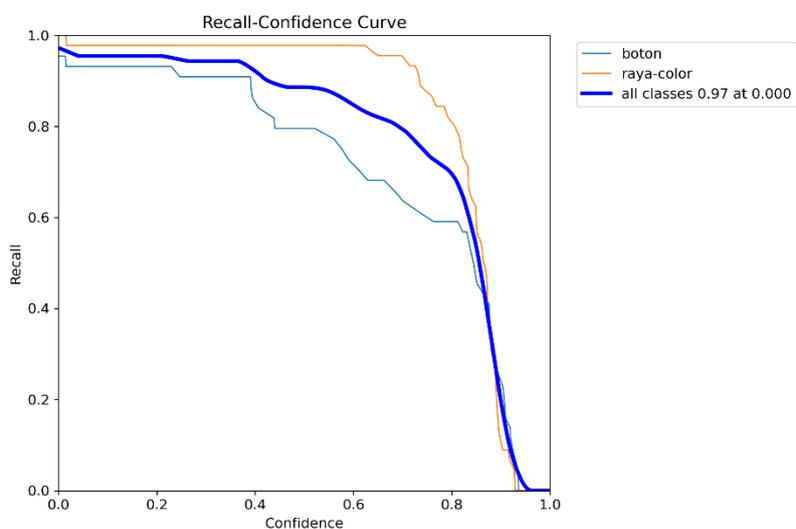


Figura 2.13: Curva recall_confidence creado con YOLOv8x.

Finalmente, YOLO nos proporciona otras gráficas en las que se puede visualizar como varían las métricas de evaluación durante la ejecución del entrenamiento como se observa en la **Figura 2.14**.

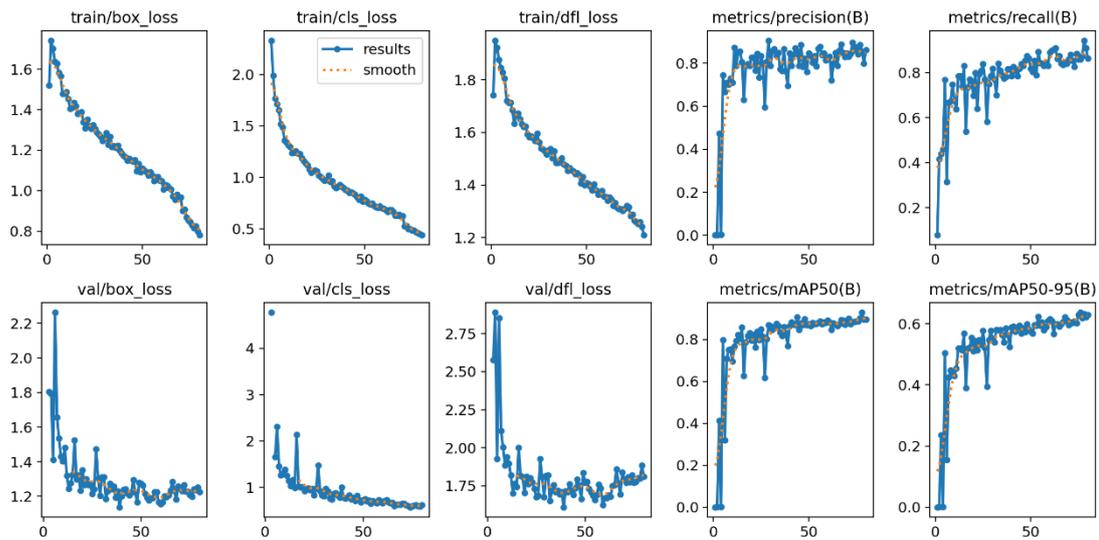


Figura 2.14: Métricas del creado con YOLOv8x durante el entrenamiento.

Después de evaluar todas las métricas que nos retorna la herramienta de Yolo se puede decir que el modelo es capaz de realizar las tareas planteadas al inicio del estudio, el modelo tiene sus errores debido a las causas explicadas en fases anteriores, sin embargo, el modelo es capaz de cumplir con los objetivos.

2.6 Despliegue

Finalmente, con el modelo evaluado y revisado cada una de las métricas el modelo seleccionado se lo utilizó para hacer el despliegue creando un script en Python, el cual consiste en consumir el modelo entrenado, evaluar en un conjunto de imágenes o con un video, en el caso de un video para la detección y seguimiento se utilizó tecnologías como DeepSort o ByteTracker.

2.6.1 DeepSort

El algoritmo realiza un enfoque de dos etapas: Primero genera detecciones de objetos, segundo asocia esas detecciones a pistas existentes.

DeepSort está hecho con 4 componentes claves los cuales son:

Detección y extracción de características: DeepSort inicia con la detección de objetos utilizando YOLO, cada detección es asociada a un descriptor de apariencia extraída por otra red neuronal. Estos descriptores codifican la apariencia de los objetos detectados y se utilizan para realizar coincidencias.

Filtro de Kalman en predicción de estado: El estado de un objeto típicamente incluye su posición, velocidad y aceleración. El filtro de Kalman predice el estado de un objeto en el frame actual basándose en el estado del frame anterior.

Asociación de datos con métricas de apariencia profunda: La asociación de datos en Deep SORT es donde el objeto actual se desvía significativamente de su predecesor. En lugar de depender únicamente de IoU, emplea una combinación de información de movimiento y funciones de apariencia. El algoritmo húngaro realiza la comparación basándose en una matriz de costos que considera tanto la distancia de Mahalanobis para la coherencia del movimiento como la distancia del coseno para la similitud de apariencia. El uso de una métrica de asociación profunda permite a Deep SORT continuar rastreando durante períodos cortos de oclusión. [15]

Gestión de pistas: DeepSort gestiona los ciclos de vida de las pistas utilizando varias heurísticas. Introduce conceptos como la confirmación de seguimiento, donde un seguimiento solo se confirma después de ser detectado en varios fotogramas consecutivos, reduciendo así los falsos positivos. También hay un parámetro de antigüedad para eliminar pistas antiguas que no se han detectado recientemente, lo que garantiza que el sistema solo mantenga pistas activas.

Las ventajas de utilizar DeepSort es que fácilmente puede identificar un objeto de frames en los siguientes frames, puede distinguir elementos que tienen apariencias similares, sin embargo, en nuestro caso de estudio se debe colocar el nombre de las clases predichas y con DeepSort no se logró, otra característica importante por lo que no se utilizó DeepSort es que, al identificar un elemento, el identificador no empezaba desde el número 1, por esos motivos se optó por utilizar otra técnica.

2.6.2 ByteTrack

Es un algoritmo de seguimiento que se utiliza para la detección y seguimiento de objetos, este algoritmo fue creado por ultralytics, los mismos creadores de YOLO por lo que es compatible con modelos de ultralytics. Al igual que el algoritmo DeepSort, ByteTracker se basa en el filtro de Kalman para la estimación del estado.

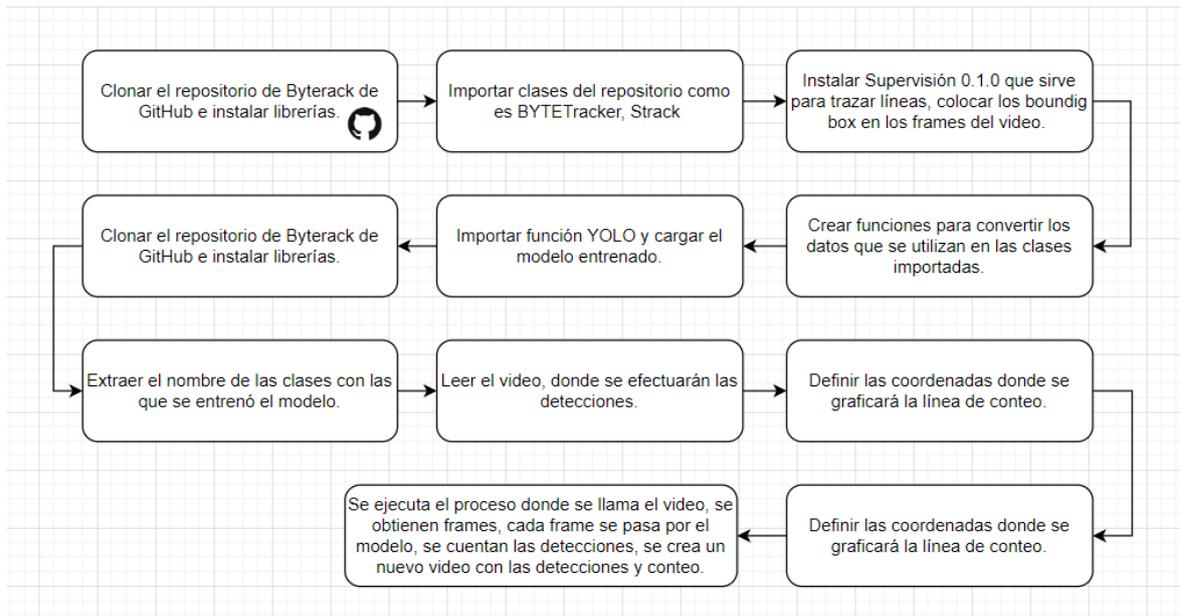


Figura 2.15: Pasos para realizar el despliegue del modelo usando ByteTrack.

El primer paso que se realizó fue clonar el repositorio de ByteTracker depositado en GitHub e instalar las diferentes librerías que se utilizaron en el desarrollo del despliegue como se muestra en la **Figura 2.15**. Un dato importante es que la versión de yolox tiene que ser la 0.1.0 ya que si es otra puede ocasionar problemas en el despliegue.

Luego de verificar que la versión de yolox sea la correcta, se procedió a instalar paquetes e importar clases de ByteTrack como BYTETracker, Strack que fueron utilizados para el seguimiento de los objetos.

A continuación, se procedió a instalar supervisión y sus dependencias para graficar los boxes en los frames, graficar la línea que será el lugar de conteo de las clases. Una consideración importante en este punto es que la línea de conteo puede contabilizar tanto de derecha a izquierda y se lo denomina como IN y OUT, por tal motivo es importante determinar el sentido del video para saber dónde se mostrarán los resultados del conteo.

Luego, se agregaron clases que realizan tareas específicas para utilizarlos luego en otras funciones importadas de ByteTrack. Una consideración importante es que las clases agregadas utilizan paquetes de ByteTrack los cuales no han sido actualizados por lo que suelen surgir errores por estos motivos.

Ya con el entorno de ByteTrack listo, se cargó el modelo entrenado con Yolo.

Se extraen las clases del modelo entrenado, las cuales serán utilizadas para clasificar el objeto detectado en el video que muestra las detecciones.

Por motivos de movilidad, accesos a la florícola donde se realizó el estudio, la falta de un dron para capturar el video en tiempo no se pudo completar el despliegue como se tenía pensado en un inicio, sin embargo, se optó por utilizar un video que fue capturado en una de las visitas técnicas para la recolección de datos.

Se establece las coordenadas de donde se coloca la línea de conteo dependiendo las especificaciones del video que se muestra en el paso previo y se establece la dirección y nombre del video donde se obtendrán los resultados.

Finalmente, se hace uso de ByteTrack con las configuraciones establecidas en los pasos previos para realizar las detecciones en el video de salida y que finalmente los resultados se muestren en video y en consola como se muestra en la **Figura 2.16**. El código completo se encuentra en la sección de Anexos.

```
ELEMENTOS CONTADOS OUT:  
boton: 0  
raya-color: 0  
ELEMENTOS CONTADOS In:  
boton: 3  
raya-color: 16
```

Figura 2.16: Resultados del conteo total por consola.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

Terminadas las 6 fases de las metodologías CRIPS-DM se obtuvieron los siguientes resultados.

3.1.1 Resultado del modelo evaluado en imágenes

Para la evaluación del modelo en imágenes Yolo nos proporciona varias salidas de conjuntos de frames en cuales se puede visualizar como el modelo está funcionando al detectar y clasificar las cabezas florales presentes en los frames. Para mostrar los resultados se toma dos muestras de imágenes.

Muestra 1: El modelo detecta correctamente las cabezas florales y el coloca correctamente la clase a que pertenece con una confianza superior al 80%.



Figura 3.1: Muestra de imágenes número 1.

Muestra 2: El modelo detecta correctamente las cabezas florales, sin embargo, se observa que en una cabeza floral se detecta las dos clases por motivos de que la rosa tiene características de una clase como la de la otra pero muestra el grado de confianza en el que uno supera el 50% y el otro está por debajo, esta consideración no se la puede controlar ya que ByteTrack hace uso de los grados de confianza en su algoritmo IoU pero al momento de contar los objetos por clase en el video si se puede controlar colocando una condición, por otro lado, en otras imágenes el recuadro de detección falla y se marca dos veces lo cual genera un grado de desconfianza.

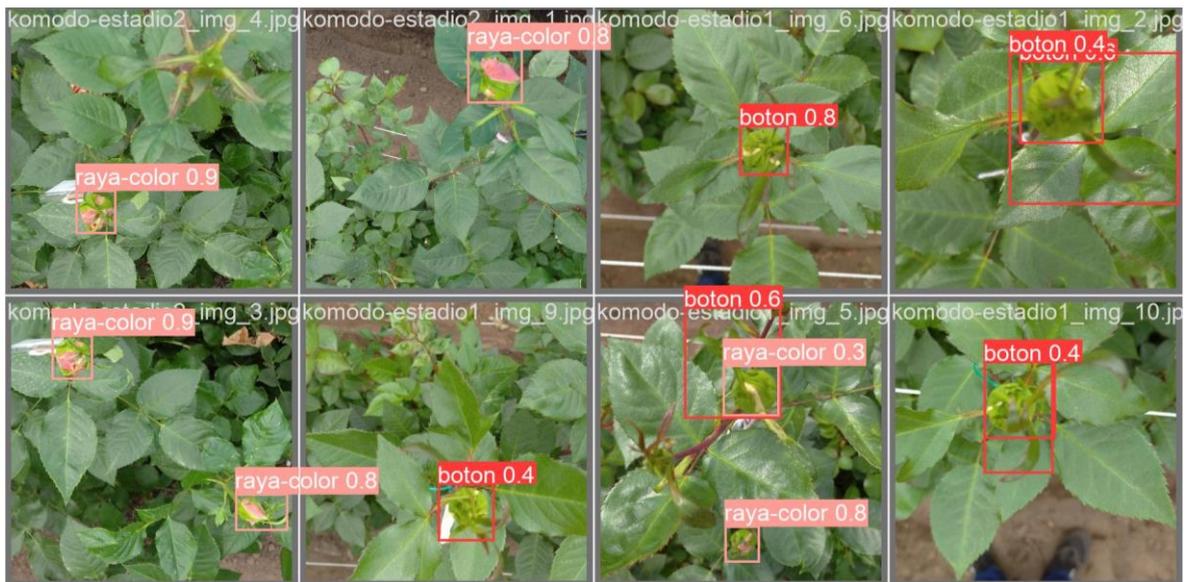


Figura 3.2: Muestra de imágenes número 2.

Después de analizar los dos casos, se puede concluir que el modelo tiene más aciertos que errores, además el modelo puede mejorar con la captura de más datos y utilizando recursos computacionales más avanzados.

3.1.2 Resultado del modelo evaluado en un video.

Para determinar si el modelo es capaz de detectar y contar todas las cabezas florales que hay en una cama se evalúa en un video capturado en la recolección de datos comparando con el conteo manual que se muestra en la **Tabla 3.1**. Para este ejemplo se toma el video de la 2da semana de la clase Komodo.

Tabla 3.1: Registro de los datos extraídos en la clase Komodo.

Komodo								
Toma	Cama4	Cama5	Cama6	Cama7	Cama8	Cama9	Cama10	Cama11
1era	6	15	12	5	11	9	8	8
2da	6	15	12	5	11	9	8	8
3ra	6	15	11	5	11	9	8	8
4ta	0	3	0	0	1	1	2	3

En la **Figura 3.3** en la parte superior izquierda se muestra el resultado total obtenido por el modelo y el seguimiento realizado con ByteTrack.

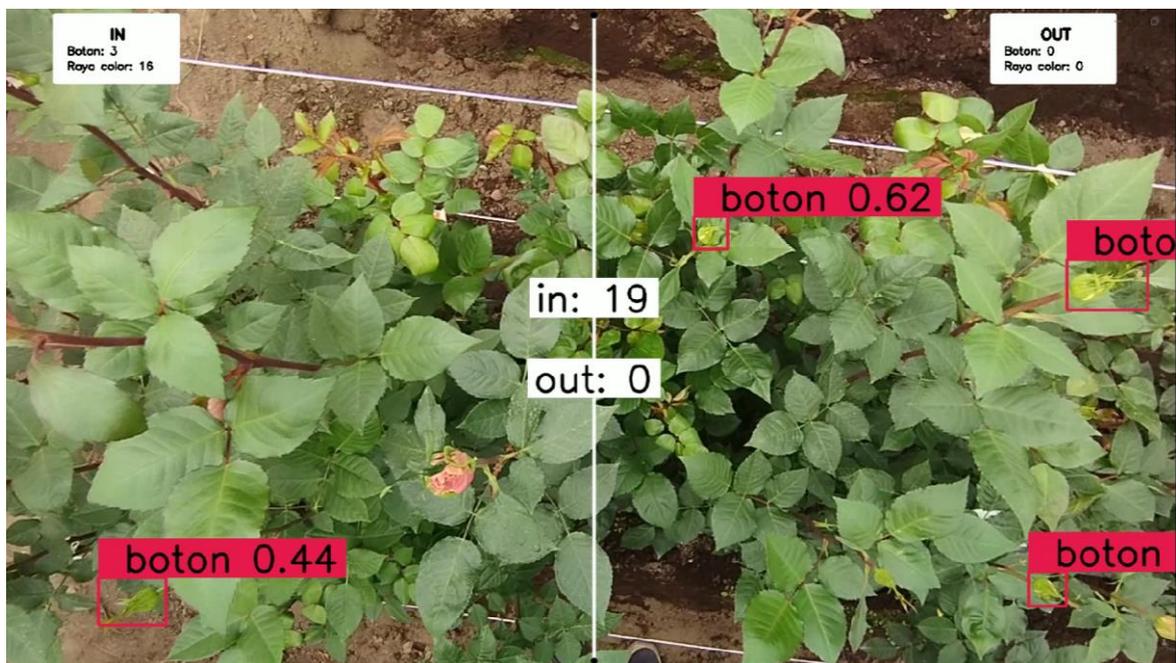


Figura 3.3: Resultado obtenido al evaluar el modelo en un video.

Se compara los resultados obtenidos en la **Tabla 3.1** y en la **Figura 3.3** y se tiene un error de un 50%. En este caso, el error se debe a que en un inicio fueron seleccionadas 9 rosas

para hacer el seguimiento y al transcurrir el tiempo aparecieron desarrollaron más rosas y efectivamente al momento de visualizar completamente el video hay más de 16 cabezas florales en estado raya color, sin embargo, no todas fueron detectadas ya que al evaluar el modelo en un video que fue capturado por un humano, los cambios de ángulos entre frames hace que el algoritmo de ByteTracke le pierda el rastro a la rosa y lo detecte nuevamente.

3.2 Conclusiones

- Utilizar visión artificial para este tipo de negocios puede llegar a ser muy complejo y no muy seguro debido a que las cabezas florales de algunas variedades de rosas no llegan a tener el mismo tamaño de estatura por lo que detectar y clasificar el estado de una rosa que se encuentran considerablemente distantes a las que sobresalen pueden causar malos cálculos y como consecuencia tomar malas decisiones.
- Planificar los días de las visitas técnicas es importante debido a que las condiciones climáticas del invernadero o del ambiente hacen que las rosas desarrollen más rápido o a su vez sea más lento y esto produce que el modelo tenga dificultades de aprender las características del estado de una rosa.
- Utilizar modelos pre entrenados con grandes conjuntos de datos y diferentes clases ayuda a cumplir los objetivos ya que tratar de entrenar una red neuronal desde cero con pocos datos es muy complicado ya que el modelo red neuronal no puede aprender muy bien y tiene malas métricas.
- ByteTracker funciona muy bien y presenta los bounding boxes en los frames de manera muy preciosa, sin embargo, no es actualizado frecuentemente por lo que está expuesto a cometer errores en algún momento.
- Los resultados obtenidos al evaluar el modelo en los videos no son precisos debido a que existen cambios bruscos del ángulo con el que es capturado el frame por tal motivo el modelo no puede detectar y llevar el conteo correctamente.
- Para entrenar una red neuronal convolucional desde cero se tiene que tener un conjunto de datos muy grande y conocimiento sobre como estructurar una red neuronal para que el modelo tenga métricas muy buenas.

3.3 Recomendaciones

- Debido a los problemas que se presentaron en la captura de los datos se recomienda realizar una visita previa para seleccionar las variedades de rosas que se va a estudiar y extraer las características de la planta como altura, volumen de tallos para tener los recursos necesarios y así que la captura de los datos no sea compleja para que los datos sean de mejor calidad.
- Para la captura de datos se recomienda utilizar un solo dispositivo móvil con una configuración específica para que todos los datos tengan la misma calidad.
- Se recomienda ampliar el conjunto de datos utilizando técnicas de preprocesamiento de imágenes para que el banco de datos sea más extenso y el modelo tenga un mejor aprendizaje.
- Se recomienda hacer el despliegue del modelo utilizando un video de buena calidad o utilizando un dron en velocidad constante para que las detecciones y conteo sean más precisos.
- Los recursos computacionales juegan un papel importante para poder hacer más pruebas variando algunos parámetros como las épocas, optimizadores, el valor del dropout y los diferentes modelos de YOLO. Por eso se recomienda utilizar una tarjeta gráfica propia y no utilizar plataformas que ofrecen el servicio ya que el tiempo de uso de estos recursos es limitado.
- Reducir los tiempos entre visitas técnicas ayuda a que la captura de los datos sea más precisa ya que hay un mejor control de las etiquetas y se evita que estas se pierdan.
- Seleccionar más variedades de rosas será de gran ayuda para aumentar el tamaño del conjunto de datos así tener más características sobre un estado fonológico el cual se requiera identificar y finalmente obtener un mejor modelo de aprendizaje.

REFERENCIAS BIBLIOGRÁFICAS

- [1] IBM, «IBM,» 17 Agosto 2021. [En línea]. Available: <https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview>. [Último acceso: 19 Febrero 2024].
- [2] «Fincas de rosas,» 12 Noviembre 2021. [En línea]. Available: <https://fincasderosas.com/variedades-de-rosas-ecuador/>. [Último acceso: 5 Febrero 2024].

- [3] M. T. B. Ramirez, «Análisis y propuesta de mejora de los procesos financieros de las PYME del sector florícola de Tabacundo,» Creative Commons, Quito, 2020.
- [4] Expoflores, «Reporte estadístico mensual quimestral,» 2023.
- [5] Fincas de rosas, «Fincas de rosas explorer,» [En línea]. Available: <https://fincasderosas.com/rosas-frescas/rosa-explorer/>. [Último acceso: 12 02 2024].
- [6] Fincas de rosas, «Fincas de rosas,» [En línea]. Available: <https://fincasderosas.com/rosadas/rosa-pink-mondial/>. [Último acceso: 12 Febrero 2023].
- [7] D. Arias, «Enter.co,» 7 Noviembre 2018. [En línea]. Available: <https://www.enter.co/especiales/colombia-conectada/inteligencia-artificial-produccion-flores-cana-azucar/>. [Último acceso: 5 Febrero 2024].
- [8] D. Gershgorn, «QUARTZ,» 10 September 2017. [En línea]. Available: <https://qz.com/1046350/the-quartz-guide-to-artificial-intelligence-what-is-it-why-is-it-important-and-should-we-be-afraid/>. [Último acceso: 31 Enero 2024].
- [9] «AWS,» [En línea]. Available: <https://aws.amazon.com/es/what-is/machine-learning/>. [Último acceso: 5 Febrero 2024].
- [10] IBM, «IBM - SUPERVISED,» [En línea]. Available: <https://www.ibm.com/mx-es/topics/supervised-learning#:~:text=El%20aprendizaje%20supervisado%20utiliza%20un,modelo%20aprenda%20con%20el%20tiempo..> [Último acceso: 5 Febrero 2024].
- [11] AWS, «AWS - ¿Que es una red neuronal?,» [En línea]. Available: <https://aws.amazon.com/es/what-is/neural-network/>. [Último acceso: 2024 Febrero 12].
- [12] K. Team, «KeepCoding,» 31 Octubre 2023. [En línea]. Available: <https://keepcoding.io/blog/que-es-y-para-que-sirve-el-fichero-yaml/>. [Último acceso: 12 Febrero 2024].
- [13] P. Lucena, «Universidad Cesuma,» [En línea]. Available: <https://www.cesuma.mx/blog/que-es-la-vision-artificial.html>. [Último acceso: 3 Febrero 2024].
- [14] S. R. Raneros, «Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning,» Valladolid, 2021.
- [15] Koudiri, Allan, «ikomia - Mastering Deep Sort: The Future of Object Tracking Explained,» 11 Octubre 2023. [En línea]. Available: <https://www.ikomia.ai/blog/deep-sort-object-tracking-guide#what-is-deep-sort>. [Último acceso: 13 Febrero 2023].
- [16] ikomia, «ikomia - ByteTrack: The Future of Multi-Object Tracking in Computer Vision,» 21 Noviembre 2023. [En línea]. Available: <https://www.ikomia.ai/blog/bytetrack-multi-object-tracking#what-is-bytetrack>. [Último acceso: 20 Febrero 2024].

- [17] K. Sergieieva, «EOS DATA ANALYTICS,» 14 Julio 2023. [En línea]. Available: <https://eos.com/es/blog/tecnologias-en-la-agricultura/>. [Último acceso: 09 Febrero 2024].
- [18] Redaccion KeepCoding, «KeepCoding,» 12 Diciembre 2023. [En línea]. Available: https://keepcoding.io/blog/redes-neuronales-convolucionales/#Que_son_las_redes_neuronales_convolucionales. [Último acceso: 5 Febrero 2024].
- [19] L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.
- [20] I. E. Naqa, «What Is Machine Learning?,» de *Machine Learning in Radiation Oncology*, Montreal, Springer International Publishing Switzerland, 2015, p. 3.
- [21] D. Gershgorn, «QUARTZ,» 10 Septiembre 2017. [En línea]. Available: <https://qz.com/1046350/the-quartz-guide-to-artificial-intelligence-what-is-it-why-is-it-important-and-should-we-be-afraid>. [Último acceso: 5 Febrero 2024].
- [22] «IBM,» [En línea]. Available: <https://www.ibm.com/es-es/topics/data-labeling>. [Último acceso: 5 Febrero 2024].
- [23] M. A. A. ROJAS, «DISEÑO DE UNA HERRAMIENTA COMPUTARIZADA PARA LA ESTIMACIÓN AUTOMÁTICA DE LA PRODUCCIÓN DE ROSAS FREEDOM EN EL CULTIVO DE FLORES DE LOS ANDES MEDIANTE PROCESAMIENTO DE IMÁGENES (BOGOTÁ D.C.),» Bogota, 2018.
- [24] C. Lozano, «Bosonit,» 19 Diciembre 2023. [En línea]. Available: <https://bosonit.com/deteccion-de-objetos-con-la-integracion-yolov8-de-kerascv/>. [Último acceso: 12 Febrero 2024].
- [25] Expoflores, «Reporte estadístico Mensual trimestral,» Quito, 2023.

ANEXOS

ANEXO I. BANCO DE DATOS

ANEXO II. CONTEO DE LOS DATOS CAPTURADOS EN LAS VISITAS TÉCNICAS

ANEXO III. SCRIPT DE DESARROLLO

ANEXO IV. MODELO

ANEXO V. VIDEO SIN DETECCIONES

ANEXO VI. VIDEO CON LAS DETECCIONES Y CONTEO

ANEXO I

Enlace al conjunto de datos:

https://drive.google.com/drive/folders/11O-WMpM7VVwdmx1tCcz5IDJtp2be0OyN?usp=drive_link

ANEXO II

Enlace al registro del conteo de datos:

[Datos.xlsx](#)

ANEXO III

Enlace al Script donde se desarrolló el proyecto:

https://drive.google.com/file/d/1Bt_eztUUEObFbDVXya8Kt2JKe7jfhGdv/view?usp=drive_link

ANEXO IV

Enlace al modelo entrenado:

https://drive.google.com/file/d/1ocMwDHS498WI8QLQzV8JWqdDSdQghAA-/view?usp=drive_link

ANEXO V

Enlace al video original sin detecciones:

https://drive.google.com/file/d/18l141CcUUjOnAgq1u3vuLxJaPizxk1lx/view?usp=drive_link

ANEXO VI

Enlace al video con las detecciones y conteo:

https://drive.google.com/file/d/1cCjufALnm4yaQF4WsIIIDaYXszhjbq2ds/view?usp=drive_link