

CONSTRUCCIÓN DE UN MODELO DE DETECCIÓN DE GANADO BOVINO EN EL
ORIENTE ANTIOQUEÑO, MEDIANTE EL SISTEMA DE VISIÓN DE UN DRON USANDO
ALGORITMOS MODERNOS DE DETECCIÓN DE OBJETOS

Juan José Giraldo Jimenéz
Anderson Julián Jimenez Gómez
José Miguel Henao Cardona

UNIVERSIDAD CATÓLICA DE ORIENTE
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
RIONEGRO

2021

CONSTRUCCIÓN DE UN MODELO DE DETECCIÓN DE GANADO BOVINO EN
ORIENTE ANTIOQUEÑO, MEDIANTE EL SISTEMA DE DETECCIÓN DE UN DRON
USANDO ALGORITMOS MODERNOS DE DETECCIÓN DE OBJETOS

Juan José Giraldo Jimenéz
Anderson Julián Jimenez Gómez
José Miguel Henao Cardona

Trabajo para obtener el título de ingeniero de Sistemas

Asesores

Reinel Castrillón
José Daniel Ramirez Soto
María Victoria Silva Domínguez

UNIVERSIDAD CATÓLICA DE ORIENTE
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
RIONEGRO

2021

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Dedicatoria

Dedicamos esta tesis a nuestras familias las cuales nos apoyaron monetaria y moralmente durante nuestros estudios, las cuales representaron motivación más que suficiente para esforzarnos en gran medida por la conclusión de esta tesis y nuestra carrera.

Agradecimientos

Agradecemos a nuestra Universidad Católica de Oriente por la formación durante todos estos años, haciéndonos crecer a nivel profesional y personal. Por crear un entorno de personas y profesionales competentes y apasionados por su trabajo, que fueron capaces de transmitirnos la misma pasión con que ejercen su profesión.

Agradecemos también a la facultad de Ingeniería de la Universidad y especialmente al programa de Ingeniería de sistemas, por ir más allá del deber y de los horarios establecidos en pro de sus estudiantes.

Agradecemos a nuestro mentor y director del semillero de Machine learning José Daniel Ramírez Soto, por su compromiso, por su capacidad de inspirar y su desinterés por compartir su gran conocimiento y experiencia, encaminándonos en nuestra carrera hacia uno de los campos más prometedores de este siglo.

Agradecemos a nuestro asesor directo de tesis Luis Reinel Castrillón, porque desde su experiencia y conocimiento del campo de estudio de esta tesis, pudo ayudarnos y enfocarnos durante cada una de las etapas de desarrollo de la misma. Además nos motivó por el camino de la investigación durante sus clases magistrales en la universidad.

Agradecemos al grupo de Machine learning I del semestre académico 2019-II, por el apoyo brindado durante las etapas de recolección y etiquetado de imágenes, las cuales posteriormente constituyeron el dataset empleado para el desarrollo de esta tesis.

Finalmente, agradecemos a nuestras familias por representar para nosotros ese apoyo incondicional en todo momento, y más ahora, próximos a concluir nuestros estudios universitarios.

Contenido

ANTECEDENTES	7
PLANTEAMIENTO DEL PROBLEMA	9
JUSTIFICACIÓN	10
OBJETIVOS	12
General	12
Específicos	12
MARCO TEÓRICO	13
DISEÑO METODOLÓGICO	20
Recolección de datos	20
Etiquetado	21
Limpieza	21
Aumento de datos	21
Exploración de diferentes modelos	21
Entrenar los modelos seleccionados	22
Evaluación de resultados	22
RESULTADOS	22
Objetivo General	22
Objetivo Específicos	22
CONCLUSIONES	30
REFERENCIAS BIBLIOGRÁFICAS	32
APÉNDICES	34

1. ANTECEDENTES

La visión por computador y más particularmente, la detección de objetos por medio de algoritmos de inteligencia artificial, son procesos que han venido evolucionando rápidamente. Los avances más recientes y cuyos resultados han demostrado grandes cambios en estos procesos, han sido los relacionados con el uso de Redes Neuronales Convolucionales (CNN por sus siglas en inglés). Entre los modelos más reconocidos en este campo están Yolo, Mask RCNN y sus variantes como Faster RCNN y Fast-RCNN, y la más reciente de todas Detectron 2.

Todos estos modelos han venido mostrando muy buenos resultados en la detección general de objetos, esto es, objetos muy cotidianos y en posiciones muy específicas. Sin embargo, ninguno de estos modelos es capaz de realizar un buen desempeño cuando se trata de tareas muy específicas, como la detección en imágenes a diferentes alturas y distancias, a diferentes tonalidades y brillos y con objetos de diversos tamaños y colores.

Mohammad Danish (2018) Trabajó con Mask-RCNN, obteniendo inicialmente las métricas con el pre-entrenamiento de los datos pertenecientes a MS-COCO y más tarde, el algoritmo fue entrenado con los datos tomados para el experimento. Para ajustar la precisión del modelo durante la etapa de entrenamiento, se "obtuvieron" 4 modelos diferentes mediante la variación de los hiper parámetros de la arquitectura como el número de épocas y el número de aumentos de las imágenes.

Wen Shao y otros (2019) trabajaron con una arquitectura de CNN, aplicaron una ingeniosa técnica para mejorar el rendimiento de la detección. Esta técnica consistió en aprovechar el hecho de que es posible obtener un tamaño aproximado del objeto detectado suponiendo que la altura del UAV desde el suelo es constante. Cambiaron el tamaño de las imágenes introducidas en la CNN a una resolución óptima obtenida con el cálculo mencionado anteriormente. Además, para evitar la sobreposición de las imágenes, se utilizó un modelo tridimensional generado a partir del vuelo del UAV. Esta investigación concluyó que el rendimiento del modelo mejora cuando las imágenes se acercan a la resolución óptima calculada mediante la obtención de un F-RATE de 0,952.

Jayne García Arnal Barbedo y otros (2019) utilizaron más de 15 arquitecturas basadas en la CNN construyendo un total de 900 modelos, lo que les permitió realizar un análisis profundo de los aspectos que afectan directa e indirectamente a la detección del ganado a través de imágenes aéreas. Los resultados llevaron a la conclusión de que muchas arquitecturas basadas en la CNN son lo suficientemente potentes como para detectar con un porcentaje de confianza considerable las imágenes del objeto de estudio incluso en circunstancias que distan mucho de ser ideales, mostrando así un gran potencial para la aplicación de los sistemas de vigilancia del ganado mediante la implantación de un sistema con vehículos aéreos no tripulados

Beibei Xu et al (2020), postulan que los avances en inteligencia artificial están mostrando resultados prometedores en la tarea de contar ganado en áreas vastas donde las condiciones climáticas y topográficas son extremas como Nueva Zelanda o Australia a partir de imágenes aéreas.

Uno de los trabajos realizados con respecto al conteo de ganado bovino dentro de la universidad fue realizado por Diana Isabel Gómez (2019), quien se centró en el conteo de vacas con el fin de detectar posibles robos dentro de una parcela. Ella obtuvo una precisión media (mAP) de 71.51% con una sensibilidad de 75% después de entrenar durante 8000 iteraciones con 9000 imágenes.

Por medio de la construcción de un conjunto de datos más variado se puede entrenar un algoritmo de inteligencia artificial de modo que se pueda identificar ganado con una mayor nivel de confianza y en condiciones mucho más variadas, como distancias, ubicaciones de los objetos de estudio, intensidad lumínica o condiciones ambientales en general, etc.

2. PLANTEAMIENTO DEL PROBLEMA

La detección y monitoreo de ganado bovino actualmente en el Oriente Antioqueño es una labor que se realiza “a ojo”, por ello se genera una gran gestión operativa en terrenos muy extensos, además, en dichos predios se hace necesario realizar un desplazamiento hasta el lugar en donde se encuentra el ganado, lo cual representa un gran esfuerzo para los campesinos en términos de tiempo, trabajo, posibles costos de desplazamiento y exposición a las condiciones climáticas.

A pesar de que se han establecido algunos estudios con respecto a la detección de ganado bovino, aún no se ha logrado tener un algoritmo capaz de detectar y contar ganado a través de drones y menos aún en el contexto específico del Oriente Antioqueño en donde se deben tener en cuenta variables como lejanía de las tomas en el dron, intensidad lumínica terrenos montañosos y variedad de los objetos de estudio, como en los tamaños del ganado o la pigmentación de la piel.

Es por esto que se ve la necesidad de probar un algoritmo de aprendizaje haciendo uso de técnicas que permitan aumentar la cantidad de datos de forma que se obtenga un conjunto de datos más variados en cuanto a distancias, intensidad lumínica, y variedad de los objetos de estudio.

3. JUSTIFICACIÓN

La utilización de un modelo para la detección de ganado bovino representa un impacto directo en la vida del campesino que lo emplea, pues, evita posibles desplazamientos, reduce los tiempos que le toma realizar esta labor y el tiempo que el campesino debe estar expuesto a las condiciones climáticas, con lo cual, también se reduce la probabilidad de contraer alguna enfermedad cutánea. Todo lo que el campesino ahorra haciendo uso de este modelo, puede ser invertido en compartir con su familia, formación integral, estudiar para cambiar sus condiciones de vida, en resumen, este modelo, puede mejorar sustancialmente la calidad de vida de los campesinos.

A la hora de trabajar con deep learning se necesitan principalmente dos cosas: Gran cantidad de datos y poder de procesamiento, el poder de procesamiento puede ser “fácilmente”¹ solventado, con la configuración en la nube de una máquina con las características que se necesitaran para la ejecución del modelo, sin embargo, la cantidad de datos es un problema que no es tan fácil de solventar, puesto que la construcción de un dataset implica desde la recolección de los datos hasta su posterior clasificación y limpieza, lo cual representa una gran cantidad de tiempo invertido en el desarrollo de esta tarea. Esto hace que la construcción de un dataset de imágenes sea aún más complejo, puesto que, en estos, los datos deberían ser recolectados, variando parámetros cómo iluminación, brillo, lejanía de las tomas, etc.

Desde un punto de vista más técnico la dificultad de la construcción de un dataset, hace que muchos proyectos optan por ser ejecutados con datos públicos y contruidos por terceros, de tal forma que se logre ahorrar una vasta cantidad de tiempo y trabajo. Sin embargo, el caso de uso de imágenes de ganado bovino tomadas por un dron no es un recurso que pueda ser encontrado fácilmente en internet. De manera que, este proyecto no sólo contribuirá a la construcción de un modelo, sino que además permitirá que diferentes investigadores y futuros modelos de

¹ En lenguaje natural se suele resaltar entre comillas algunas palabras para indicar que el adjetivo o adverbio no están calificando de una manera literal, sino que califican de una manera relativa a otro componente del mismo discurso. En este caso el procesamiento es relativamente fácil de solventar si se compara con lo tedioso que puede llegar a ser la manipulación de grandes volúmenes de datos.

inteligencia artificial puedan hacer uso de estos datos para acelerar sus actividades y disminuir costos.

Dado que los trabajos anteriormente mencionados destacan que la variedad de los datos debe ser mejorada de modo que las imágenes propendan a tener la menor correlación posible, se ve la necesidad de realizar un trabajo enfocado en la variedad y la adquisición de los datos. Este proyecto es una continuación del artículo publicado previamente por Diana Isabel Gómez (2019), en el que se plantea como una posible mejora la expansión del dataset original, buscando evitar que los datos de entrenamiento y prueba estén correlacionados. Esto permite que, además de hacer un mejor acercamiento a un modelo desplegable en entornos reales de producción, se pueda contar con un modelo menos sesgado.

Es por esto que se hace necesario un trabajo más profundo en la construcción del dataset, de modo que se generen imágenes con alturas y lejanías variadas, que contengan vacas en diferentes posiciones, además de, condiciones climáticas y de iluminación variables haciendo que el dataset contenga datos más variados y cercanos a la realidad. Algunas de estas variaciones de los datos se puede hacer mediante el uso de técnicas de modificación de imágenes comúnmente conocido en el mundo de los modelos de inteligencia artificial como aumento de datos. Además de la construcción y entrenamiento del modelo de detección basado en modelos de detección de objetos de inteligencia artificial.

4. OBJETIVOS

4.1. General

Construir un modelo de detección de ganado bovino en el Oriente Antioqueño, mediante el sistema de visión de un dron usando algoritmos modernos de detección de objetos.

4.2. Específicos

- 4.2.1.** Generar un conjunto de datos de ganado bovino con los cuales se puedan entrenar el modelo.
- 4.2.2.** Explorar diferentes modelos o algoritmos de detección de objetos de forma que se pueda encontrar potenciales aplicaciones para la detección de ganado bovino.
- 4.2.3.** Entrenar el modelo elegido usando los datos generados, con el fin de que este aprenda de los datos en el contexto objeto de estudio.

5. MARCO TEÓRICO

Cuando nos referimos a **Ganado Bovino** hacemos referencia al conjunto de animales compuesto por las vacas, toros o bueyes domesticados por el humano para su aprovechamiento principalmente en la producción de lácteos y cárnicos. En algunas ocasiones se habla de ganado bovino, ganado o simplemente vacas para referirse a este concepto. Este término es ampliamente usado en los **entornos rurales** los cuales son el espacio geográfico que se caracteriza por tener grandes extensiones de tierra. Se diferencia principalmente de los **entornos urbanos** en la densidad poblacional y la identidad productiva, siendo los entornos rurales poco poblados y destinados principalmente para la producción agrícola e industrial.

Los **drones**² son un tipo de aeronave pequeña no tripulada, las cuales han sido usadas por entidades gubernamentales y otro tipo de entidades para realizar tareas de exploración y vigilancia. Con los avances en tecnología este tipo de aeronaves ha disminuido su costo convirtiéndose en un artículo de fácil acceso para las personas o los grupos pequeños de investigación. Hoy en día es incluso usado para fines competitivos o de ocio. El tipo de dron más ampliamente usado actualmente por las personas son los drones propulsados por hélices.

Para muchas personas la revolución generada por la digitalización e internet en el siglo pasado está siendo sucedida por la revolución generada por las inteligencias artificiales. Para comprender el concepto de **Inteligencia Artificial** es bueno comprender algunos conceptos previos. Un **agente** tiene la capacidad de actuar u obrar pero un **Agente Inteligente** es “un sistema que actúa inteligentemente: Lo que hace es apropiado para sus circunstancias y su objetivo, es flexible a entornos cambiantes y objetivos cambiantes, aprende de la experiencia y toma decisiones apropiadas dadas las limitaciones perceptivas” (Pool, Mackworth y Gobel, 1998, p.1). En la actualidad, aún no se ha logrado crear una inteligencia artificial capaz de ser muy buena en varias disciplinas al mismo tiempo. Por lo general se habla de inteligencias artificiales muy buenas en detección de objetos, o de inteligencias artificiales muy buenas en el reconocimiento de lenguaje hablado o escrito, pero aún no se ha logrado una inteligencia artificial

² Según la RAE un dron es simplemente una *aeronave no tripulada* no se especifican tipos de propulsión ni tamaños. ver <https://dle.rae.es/dron>

multidisciplinaria. Es por esto que estas características mencionadas por el autor enmarcan los límites de las inteligencias artificiales.

Cuando en el contexto de inteligencia artificial se habla de un **modelo** se hace referencia a un programa o algoritmo que busca predecir un resultado con datos que nunca ha visto. Esto es posible gracias a que el modelo durante la etapa de entrenamiento encuentra patrones que le permiten realizar la inferencia.

Las **Redes Neuronales Artificiales** o simplemente Redes Neuronales, están motivadas en la forma en que los sistemas nerviosos biológicos son capaces de procesar la información (Izaurieta, F., & Saavedra, C., 2000, p.1). El sistema nervioso humano es un sistema altamente complejo y a su vez altamente eficiente. Izaurieta (2000), asegura que muchas de las tareas que son sumamente simples para el hombre, pueden llegar a ser altamente complejas para las máquinas como por ejemplo en la detección de patrones a través de la vista, en donde los humanos fácilmente son capaces de detectar los cambios de una persona a través en el tiempo o simplemente un cambio de aspecto en el tiempo de algún objeto.

Las redes neuronales artificiales están construidas de una forma semejante a las redes neuronales biológicas. Estas nacieron con la intención de imitar su comportamiento en el proceso de aprendizaje como se mencionó anteriormente. Las redes neuronales artificiales están compuestas por capas que a su vez se componen de neuronas. Las capas pueden tener un número variado de neuronas y a su vez las redes neuronales tienen diferente número de capas. De la forma en que se construyen estos modelos de redes neuronales y la forma como se conectan las capas entre sí es que resultan los tipos de redes neuronales. Se hablará de esto más adelante.

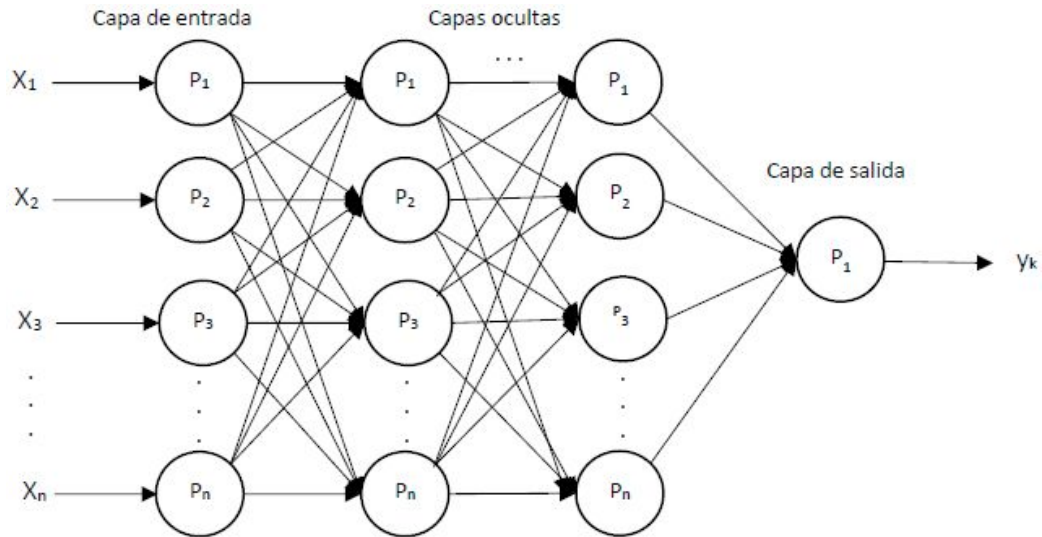


Figura 1. Estructura general de una red neuronal

Fuente: Elaboración Propia. enero de 2020

Las neuronas son las unidades básicas de procesamiento dentro de una red neuronal, estas se encargan de recibir señales o entradas de otras neuronas predecesoras y generan una salida o señal que puede ser tratada mediante una función de activación.

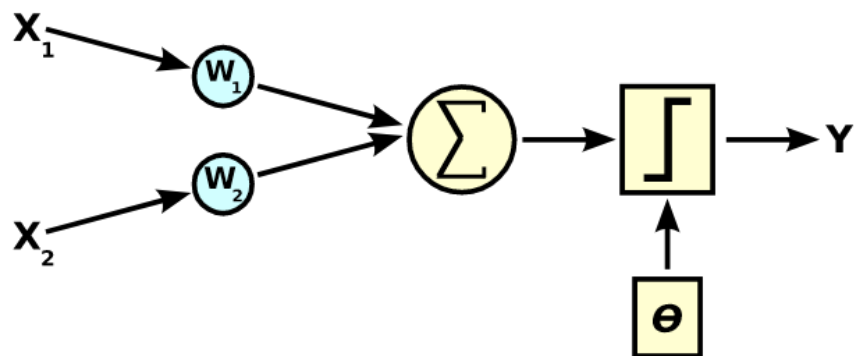


Figura 2. Artificial neuron (Computer science). Estructura general de una neurona artificial.

Fuente: Elaboración Propia. enero de 2020

Las funciones de activación en las redes neuronales artificiales son una operación o función matemática que genera un valor de salida. Su objetivo es mantener estas salidas en un

rango de valores específico. Las funciones de activación más comunes son **ReLU** que consiste en asignar la salida a cero si el valor arrojado por la neurona es negativo, de lo contrario asigna la salida con el valor arrojado por la neurona. La función **Sigmoid** asegura que el valor arrojado por la neurona esté en el rango de $[0, 1]$. La función **Tangente Hiperbólica** asegura que la salida sea un valor en el rango $[-1, 1]$. La función **Valor absoluto** asegura que la salida sea un valor positivo transformando la salida de la neurona mediante la función matemática valor absoluto. Finalmente la función **Binaria** asegura que la salida que se produzca sea 0 si el valor arrojado por la neurona es menor a cero o 1 si el valor arrojado por la neurona es mayor a cero.

Finalmente las estructuras conformadas por las neuronas de un mismo nivel de profundidad en la red neuronal son consideradas **Capas**. Por lo regular las neuronas de una misma capa se compone de neuronas con funciones de activación semejantes. Las cantidad de capas de una red neuronal pueden ser variada, dentro de la terminología que se emplea para nombrarlas podemos hablar en tres categorías. La capa de entrada, que se refiere a la primera capa de la red neuronal, es la capa encargada de recibir los valores tal cual vienen desde las observaciones (datos de entrenamiento). La capa de salida es la capa que está al final de la red neuronal y es la que entrega los resultados finales de la red neuronal. Finalmente las capas intermedias son las capas que se encuentran después de la capa de entrada y la capa de salida; el número de estas capas suele ser un valor grande.

La **visión por computador** es una de las ramas de investigación de la inteligencia artificial y consiste en analizar imágenes digitales de modo que se pueda entender lo que representa la imagen o simplemente reconocer los objetos que la componen (**object detection**). Este tipo de investigaciones se están soportando hoy día con la implementación de nuevas arquitecturas de redes neuronales, en donde se ha encontrado que de acuerdo a las arquitecturas o modelos que se construyan se pueden reconocer ciertos tipo de objetos con mayor precisión.

Las **Redes Neuronales Convolucionales**, CNN por sus siglas en inglés, fueron introducidas inicialmente por Kunihiko Fukushima en 1980. Estas son tipos de redes neuronales artificiales que usan un tipo de filtros de una o más dimensiones y están inspiradas en el sistema nervioso visual de los vertebrados (Fukushima, K., & Miyake, S., 1982, p.1). Este tipo de redes

son especialmente usadas en el procesamiento de imágenes. Las CNN fueron mejoradas por LeCun & Bengio en 1998 cuando introdujeron un modelo con propagación hacia atrás, más conocido como **Redes Neuronales Recurrentes**, RNN. La propagación hacia atrás consiste en un ajuste de los pesos en una red neuronal donde se da retroalimentación a las neuronas partiendo de la capa de salida hacia la capa de entrada retroalimentando solo una parte del valor final, de modo que sea equivalente al aporte en la predicción de cada una de las neuronas. En otras palabras, las RNN, a diferencia de una red neuronal tradicional, son redes donde las neuronas de capas posteriores (más cercanas a las capas de salida) contienen conexiones con neuronas en capas previas (más cercanas a las capas de entrada).

Al estar inspiradas en las redes biológicas del sistema nervioso visual, las CNN están especialmente pensadas para tareas de reconocimiento de patrones en imágenes o más generalmente, se encasillan dentro de tareas de Visión por Computador. Muchos de los modelos de detección basados en CNN requieren de un modelo de proposición de regiones, estos modelos tratan de extraer de antemano algunas posibles regiones dentro de una imagen. Estas RPN (Region Proposal Networks, por sus siglas en inglés), han sido propuestas por Ross Girshick (2015) en su trabajo sobre un modelo de predicción de objetos en tiempo real, llamado Faster RCNN, que es la base del modelo Mask R-CNN usado en este trabajo.

Una vez entendido a modo grosso el concepto de los agentes artificiales inteligentes podemos ahora centrarnos en entender la forma en que se pueden construir estos agentes. Es aquí donde empezamos a hablar de las técnicas de modelado de inteligencias artificiales. Actualmente los conceptos de Deep Learning, Machine Learning o Inteligencia Artificial son conceptos que se escuchan por diferentes lados y muchas veces de forma sensacionalista, son como los temas del momento, sin embargo son temáticas que se llevan hablando desde mediados del siglo pasado cuando Alan Turing (1954) diseñó el actualmente denominado “Test de Turing” que consiste en poner a una máquina y a una persona en un interrogatorio donde el interrogador no puede saber quien es la máquina y quién es la persona. Si el interrogador no logra deducir cuál de los interlocutores es la máquina y cuál es la persona, entonces se dice que la máquina ha superado la prueba.

Cuando hablamos de **framework** nos referimos a un marco o entorno de trabajo el cual está compuesto de ciertos métodos y funciones los cuales tienen como objetivo facilitarnos el trabajo reduciendo tiempos de desarrollo.

A la hora de trabajar con modelos de inteligencia artificial hay un pipeline o conjunto de pasos sumamente aceptado para el desarrollo de algoritmos de este índole, este pipeline se divide en las siguientes etapas:

- **Construcción del dataset.** El cual está compuesto de los datos de estudio necesarios para las etapas posteriores.
- **Limpieza del dataset.** Una vez construido el dataset debemos garantizar la calidad del mismo, pues el modelo será tan bueno o tan malo como los datos usados durante su entrenamiento.
- **Data Augmentation (Opcional).** Esta etapa es sumamente útil para casos de estudio en los que se cuenta con poca cantidad de datos, pues como su nombre lo indica consiste en aumentar los datos de los que se dispone aplicando diferentes técnicas de aumento de datos.
- **Mezclar los datos.** Esto se hace con el fin de que el modelo no aprenda cosas innecesarias o erróneas debido a la forma en que está estructurado el dataset, con esto también conseguimos darle el mismo peso o importancia a todos los datos dentro del dataset.
- **Separar datos de test y entrenamiento.** Esta etapa es importante pues permite garantizar que los datos usados durante la etapa de entrenamiento sean datos que el modelo nunca ha visto, con esto podemos determinar si nuestro modelo tiene **overfitting**, el cual es un término usado para referirnos cuando un modelo se adapta demasiado bien a los datos de entrenamiento pero su desempeño se reduce sustancialmente con datos nuevos.
- **Entrenamiento.** Durante esta etapa, el modelo busca encontrar patrones en los datos de entrenamiento variando los **pesos**, los cuales, son parámetros que determinan la importancia de una característica en la **inferencia** o predicción final realizada por el modelo.

- **Test y obtención de métricas.** La finalidad de esta etapa es probar el modelo con datos nuevos para él. Se realiza una obtención de **métricas**, las cuales nos indican la calidad de las predicciones arrojadas por el modelo.

Nota: Es común, que cuando las métricas calculadas durante el entrenamiento difieren demasiado a las obtenidas en la etapa de test, se opte por volver a entrenar vareando los **hiperparametros**, los cuales representan parámetros externos al modelo y que ayudan a mejorar el rendimiento del mismo, tales como lo son las **epochs** o **épocas** (número de veces que se recorre el dataset completo) y los **step per epoch** o **pasos por epoca** (número de grupos en los que se parte el dataset para ser ingresados al modelo).

Dependiendo del modelo con el que se está trabajando puede ser necesario aplicar un algoritmo para eliminar **bounding boxes** (zonas rectangulares detectadas, que encierran el objeto del caso de estudio en una forma rectangular) en el caso de modelos de detección y localización o **máscaras** (zonas que encierran el objeto caso de estudio por su silueta) para el caso de algoritmos de segmentación. Las cuales en algunos casos pueden sobreponerse, detectando varias veces el mismo objeto dicho algoritmo consiste en descartar las bounding boxes que sobrepasen un hiper parámetro conocido como **intersection over union (IoU)**, el cual nos permite evaluar qué tan similar es una bounding box predicha a la bounding box real. (Oleksii Sheremet, 2020)

Al momento de obtener las métricas del modelo, es necesario tener en cuenta ciertas medidas de desempeño necesarios para el cálculo de las mismas, dichos medidas son:

- Falsos positivos (FP, false positive)
- Falsos negativos (FN, false negative)
- Verdaderos positivos (TP, true positive)
- Verdaderos negativos (TN, true negative)

Debemos entender los verdaderos como los casos en los que el modelo detecta un objeto y lo clasifica como una **clase**, la cual representa el valor de un conjunto de valores de segmentación enumerados para una etiqueta (para el caso del presente estudio, la etiqueta es “vaca”). (Google developers, s.f.). Los falsos representan los casos en los que el modelo no

clasifica un objeto dentro de nuestra etiqueta (vaca). Por último, los positivos y negativos representan si el modelo acertó o no en su predicción. Para ejemplificar, un **falso positivo** (FP) es un ejemplo en el que el modelo predijo de manera incorrecta la clase positiva. Por ejemplo, el modelo infirió que un mensaje de correo electrónico en particular era spam (la clase positiva), pero ese mensaje de correo electrónico en realidad no era spam. (Google developers, s.f.).

Teniendo claro las medidas con las cuales se calculan las métricas podemos definir como se calculan las métricas (véase la Figura 3):

$$\begin{array}{l}
 \text{Precisión} = \frac{TP}{TP+FP} \qquad \text{Exactitud} = \frac{\text{Predicciones correctas}}{\text{Total de muestras}} \\
 \text{Sensibilidad} = \frac{TP}{TP+FN}
 \end{array}$$

Figura 3. Cálculo de las métricas presentadas en el capítulo de resultados. Elaboración propia.

Cada una de las métricas representadas en la anterior figura nos indican un rendimiento diferente de nuestro modelo, la **precisión** identifica la frecuencia con la que un modelo predijo correctamente la clase positiva (Google developers, s.f.). La **exactitud o accuracy** es la fracción de predicciones que se realizaron correctamente en un modelo de clasificación (Google developers, s.f.). Y la **sensibilidad o recall** nos indica, de todas las etiquetas positivas posibles, cuántas identificó correctamente el modelo (Google developers, s.f.).

6. DISEÑO METODOLÓGICO

6.1. Recolección de datos

- Se grabarán múltiples videos desde el sistema visual de un dron variando factores tales como elevación, inclinación, distancia y velocidad; de manera que se capture la mayor variabilidad posible.
- Una vez se tenga el video, se procederá a extraer muestras de un frame cada segundo de manera que se tenga la suficiente variabilidad entre imágenes.

6.2. Etiquetado

- Se distribuirán las imágenes previamente extraídas entre el grupo de trabajo y se definirá un estándar de etiquetado y nombramiento para su posterior etiquetado.
- Se etiquetarán las imágenes haciendo uso de una herramienta llamada “VGG Image Annotator (VIA)”, donde se generará una máscara para cada vaca y se diferenciarán las clases por tipo de vaca, es decir, que si hay 10 vacas diferentes se generarán 10 etiquetas diferentes.
- Basado en el anterior proceso de etiquetado, se procederá a generar un nuevo formato de etiquetas, en el cual se eliminará la clasificación multiclase, es decir, que se conservará la máscara y su etiqueta solo será “vaca”.

6.3. Limpieza

- Se procederá a hacer un análisis de calidad del etiquetado.
- Se aplicarán diferentes procedimientos que permitan identificar y corregir los errores del proceso de etiquetado.
- Se dividirán los datos en 3 subgrupos: *entrenamiento* para el ajuste de los pesos de los modelos época tras época; *validación* para identificar el rendimiento del modelo a lo largo de cada época y *test* para medir el rendimiento final del modelo sin sesgos.

6.4. Aumento de datos

- Se explorarán múltiples técnicas en busca de las transformaciones que den mayor variabilidad y menor correlación a los datos, tales como: iluminación, rotación, transformación espejo, efectos de lluvia, contracción o extensión de las imágenes y combinaciones de las mismas.
- Aumentar la cantidad de datos recolectados, seleccionando tentativamente al menos 10 técnicas de transformación que permitan multiplicar por 10 el número de imágenes, para una posible mejora en las métricas de los algoritmos.

6.5. Exploración de diferentes modelos

- Se explorarán diferentes algoritmos en el estado del arte para la detección de objetos.
- Se seleccionarán 1 modelo, que pueda potencialmente resolver la tarea en cuestión.

6.6. Entrenar los modelos seleccionados

- Se procederá a entrenar los modelos con los datos originales (es decir, sin considerar el aumento de datos) bajo la técnica de transferencia de aprendizaje, usando pesos pre-entrenados que proveen las diferentes arquitecturas de detección seleccionadas.
- Basado en los resultados del paso anterior en términos de rendimiento y velocidad computacional, se determinará si nuevamente se entrenará el modelo usando transferencia de aprendizaje, en este caso haciendo uso de los datos originales y los datos aumentados, en busca de un mejor rendimiento del modelo.
- Se ajustarán los hiper-parámetros en busca de mejorar el rendimiento del algoritmo, tales como intersección sobre unión (IoU), learning rate, número de épocas, pasos por época y variaciones en los subgrupos de los datos de entrenamiento, test y validación según sea el caso.

6.7. Evaluación de resultados

- Se calcularán las métricas del proceso de transferencia de datos, tales como especificidad y precisión.
- Se compararán las métricas de los procesos propuestos en las etapas anteriores.
- Se mostrarán las ventajas y desventajas de los frameworks, herramientas utilizadas y hacia dónde deberán ir enfocados los próximos proyectos para la detección de ganado bovino.

7. RESULTADOS

7.1. Objetivo General

Se logró construir un modelo de detección y segmentación de objetos para la identificación de ganado bovino, con imágenes tomadas a través del sistema de visión de un dron en el Oriente Antioqueño, que permitieron evidenciar el potencial de estas nuevas tecnologías en la implementación de actividades del mundo real, tales como cuidado y búsqueda de ganado, inventario de ganado y eficiencia de monitoreo.

7.2. Objetivo Específicos

7.2.1. Se generó un conjunto de datos con un total de 1.311 imágenes originales y 24.017 imágenes aumentadas haciendo uso de 13 diferentes técnicas de aumento de datos, equivalente a 2 meses de trabajo continuo dedicadas a esta tarea específica, (véase la **Figura 4**) que cuentan además con sus correspondientes etiquetas multiclases y binarias debidamente documentadas, seguido de un estricto proceso de calidad y depuración.

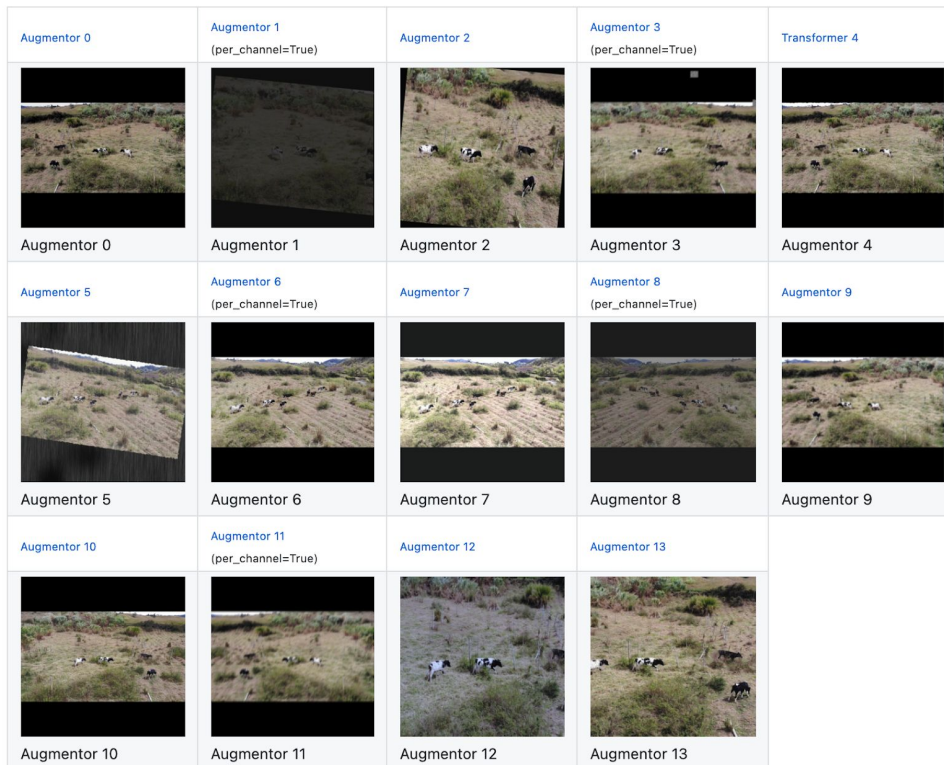


Figura 4. Muestras de las imágenes bajo las técnicas de data augmentation.
Fuente: Elaboración propia - Agosto 2020

Ayudando de esta manera a futuros proyectos, estudiantes e investigadores a que ahorren el dispendioso tiempo de recolección, construcción y depuración de un conjunto de datos que puede ser aplicado a detección, localización, segmentación de imágenes e incluso identificación de identidad de una vaca, puesto que las imágenes cuentan con una etiqueta única por vaca, véase **Figura 5**.



Figura 5. Guia de nombramiento de las imágenes y etiquetas para las vacas.

Fuente: Elaboración propia - Noviembre 2019

Adicional a ello, se realizó una exploración de 3 de algunos de los datos públicos más populares: COCO dataset, Sun dataset y Pascal dataset en busca de remarcar la importancia y la escasez del dataset construido en este proyecto.

Partiendo en primer lugar de COCO dataset, donde se pueden encontrar 330 mil imágenes, 1.5 millones de instancias de objetos y 80 categorías entre ellas vacas. Particularmente este dataset contiene 2.055 imágenes que involucran vacas que realmente no contiene información similar y que no cumple directamente con el objetivo del presente artículo, dado que cuentan con las siguiente particularidades:

- Vacas en playas que no es muy común en Colombia
- Vacas en corrales
- Vacas que no son reales, es decir son un logotipo de alguna tela, o una publicidad de un restaurante.
- Imágenes muy cercanas de las vacas.
- Todas ellas comparten el hecho de que fueron tomadas desde el suelo y no por un dron como es el caso del presente proyecto.

Sun dataset es otro conjunto de datos muy popular que contiene 131.067 imágenes con 4.479 categorías, sin embargo, este sólo contiene 36 imágenes de vacas y un total 172 instancias.

Finalmente, se encontró otro dataset relacionado con animales llamado Pascal el cual contiene 11.530 imágenes y 20 categorías donde ninguna de ellas son vacas.

7.2.2. Se abordaron dos algoritmos en el estado del arte de detección, localización y segmentación de objetos en una imagen, los cuales fueron analizados en diferentes escenarios y bajo diferentes métricas de desempeño.

Mask RCNN

En términos de Mask RCNN se midió y analizó el rendimiento del modelo con los pesos pre-entrenados de COCO dataset haciendo inferencia a todos los datos recolectados y etiquetados (sin aumento de datos).

Una vez se obtuvieron los resultados, se notó una tendencia de algunas predicciones erróneas en un tipo de imágenes en particular, con lo cual se planteó la siguiente hipótesis: *“El modelo mask RCNN no funciona bien para imágenes que no sean tomadas desde el suelo”*. Con esta hipótesis en mente, se procedió a identificar el tipo de imágenes con las que fue entrenada esta arquitectura (COCO dataset) en busca de respuesta a la hipótesis planteada y para el caso concreto de las imágenes se obtuvieron resultados que no apuntan particularmente al objetivo del presente proyecto, los cuales están descritos en la sección 7.2.1.

Partiendo de los anteriores factores, se procedió a probar la hipótesis planteada, separando dos muestras de las imágenes recolectadas (véase la **Tabla 1**). La primera muestra corresponde a imágenes con una posición similar a como si hubiesen sido tomadas desde el suelo o muy cercano al mismo; y en el segundo caso se seleccionaron imágenes aéreas y lejanas.

Perspectiva	Description	Inference set	image pixels
Cercana al suelo	Imágenes con una perspectiva cercana al suelo, y con leves ángulos de inclinación.	136	1920 × 1080
Aéreas y lejanas	Imágenes con una perspectiva lejana y con un ángulo muy vertical con respecto a la posición de las vacas y el dron.	303	1920 × 1080

Tabla 1. Muestras de las imágenes aéreas y cercanas al suelo

Para cada uno de los dos datos obtenidos en la **Tabla 1**, se realizaron dos pruebas con las cuales se tenía por objetivo dar información detallada del rendimiento del modelo midiendo dos factores.

1. Detección de objetos de interés.
2. Detección los objetos de interés y adicionalmente si el modelo lo categorizó correctamente, es decir, se detectó como “vaca”.

Tipo de imágenes	Filtro	TP	FP	FN	Precisión	Recall	Accurac y
Cercana al suelo	Detección	884	113	180	0.90	0.83	0.92
	Detección y clasificación	730	334	267	0.74	0.69	0.81
Lejanas y aéreas	Detección	737	373	1731	0.57	0.30	0.39
	Detección y Clasificación	136	974	2332	0.10	0.06	0.32

Tabla 2. Rendimiento de Mask RCNN midiendo detección de objetos y clasificación para imágenes cercanas al suelo y lejanas o aéreas.

Como se puede observar en la **Tabla 2**, el rendimiento del modelo detectando vacas con imágenes lejanas disminuyó considerablemente en comparación a los resultados obtenidos con imágenes tomadas del modelo. Demostrando exitosamente que la hipótesis anteriormente mencionada es correcta y afirmando así que el modelo Mask RCNN con los pesos de coco dataset no tiene un buen rendimiento en la detección de imágenes aéreas y lejanas.

Nombre	Número de vacas	Vacas no detectadas	Proporción de error
Colorada	124	31	0,25
ColoradaOscura	110	29	0,26
ManchasDefinidas	124	19	0,15
ManchasGrandes	124	17	0,14
ManchasNoDefinidas	136	13	0,10
Negra	120	25	0,21
NegraManchasBlancas	146	19	0,13
PatasBlancas	134	9	0,07
RodillasBlancas	31	7	0,23
Ternero	15	11	0,73

Tabla 3. Métricas del subconjunto de imágenes cercanas al suelo en el filtro de detección

Adicionalmente, se seleccionaron las métricas del tipo de imágenes cercanas al suelo en el filtro de detección (véase la **Tabla 3**), haciendo un análisis más detallado de las vacas con las que el modelo tiene los mejores y peores resultados, los cuales quedaron estipulados en la **Tabla 3**. Obteniendo que la etiqueta “*PatasBlancas*” es la vaca que mejor detecta el modelo con un porcentaje de error del 7%, mientras que “*Ternero*” es la vaca con el porcentaje de error en detección más alto equivalente a un 73%.

Detectron2

Ya teniendo los resultados de Mask RCNN de la anterior exploración, se procedió a identificar los algoritmos más avanzados de la familia RCNN, pues como se especificó en los antecedentes del presente documento, éste algoritmo ha evolucionado desde RCNN, fast RCNN, faster RCNN, Mask RCNN y más recientemente Detectron2 como la última mejora a la fecha, el cual se ha

demostrado ser más rápido que Mask RCNN tanto en entrenamiento como en inferencia. (Detectron2, 2020).

Detectron2 es un framework creado por Facebook para respaldar la rápida implementación y evaluación de nuevos investigadores sobre la visión por computador. Siendo este la última implementación de la familia RCNN, se ha formado una comunidad muy activa la cual ha creado una implementación con el modelo construido y una API para transfer learning completamente extensible usando la famosa herramienta Google Colab la cual ofrece servicios que no requieren configuración adicional para usarse, provee gratuitamente CPUs y TPUs³, y cuenta con la misma facilidad para ser compartida que un documento de Google.

7.2.3. Para trabajar sobre la construcción de este modelo se tomaron dos enfoques, el primero consistió en hacer transferencia de aprendizaje usando el primer conjunto de datos original (sin aumento de datos). El segundo enfoque consistió en realizar la transferencia de aprendizaje con el conjunto datos aumentado, alrededor de

³ Se habla de TPU cuando se refiere a unidades de procesamiento que constan de arreglos de GPUs, están especialmente diseñadas para tareas relacionadas con visión por computador, pues requieren de millones de operaciones en su procesamiento



Figura 6. Predicciones del modelo entrenado mediante el uso de Detectron2.

Fuente: Elaboración propia Enero de 2020

Los resultados obtenidos en los dos enfoques tomados son los siguientes:

Enfoque 1: En este caso se reporta un Presición Média de 0,979 y 0,972 para la detección de la región de interés (BOX) y la máscara (SEGMENT) cuando el porcentaje de traslapamiento es del 50% entre la predicción y la región etiquetada; Para el caso de al menos un 75% de traslapamiento entre las inferencias y las etiquetas se reporta una Precisión Média 0,781 y 0,624 para las inferencias de región de interés y máscara respectivamente. Véase **Tabla 4**.

Enfoque 2: Se reporta una Precisión Media de 0,937 y 0,910 para las regiones de interés y máscaras respectivamente en el caso de al menos un 50% de traslapamiento

entre las predicciones y la etiquetas. Una precisión media de 0,645 y 0,436 para regiones de interés y máscaras respectivamente con al menos un 75% de traslapamiento en las inferencias.

A pesar de que en las métricas reportadas más altas los porcentajes de traslapamiento tienden a caer, hay que entender que se trata de resultados muy buenos ya que el modelo logra detectar la ubicación de los objetos de interés dentro de la imagen. Además de esto se logra observar que en regiones muy pequeñas (las cuales se tratan de etiquetas a objetivos lejanos) el modelo tiene mayores dificultades para detectar el objeto de interés cuando se trata de regiones medianas o grandes. Esto es completamente lógico ya que al estar un objeto de interés más alejado del dron que captura la imagen hay menos detalles que puedan ser tenidos en cuenta por el modelo.

	APPROACH 1		APPROACH 2	
	BOX	SEGMENTS	BOX	SEGMENTS
Average Precision (AP) @[IoU=0.50:0.95 area= all maxDets=100]	0,673	0,565	0,579	0,464
Average Precision (AP) @[IoU=0.50 area= all maxDets=100]	0,979	0,972	0,937	0,910
Average Precision (AP) @[IoU=0.75 area= all maxDets=100]	0,781	0,624	0,645	0,436
Average Precision (AP) @[IoU=0.50:0.95 area= small maxDets=100]	0,575	0,468	0,458	0,327
Average Precision (AP) @[IoU=0.50:0.95 area=medium maxDets=100]	0,734	0,617	0,676	0,556
Average Precision (AP) @[IoU=0.50:0.95 area= large maxDets=100]	0,705	0,577	0,720	0,639
Average Recall (AR) @[IoU=0.50:0.95 area= all maxDets= 1]	0,099	0,084	0,095	0,078
Average Recall (AR) @[IoU=0.50:0.95 area= all maxDets= 10]	0,710	0,609	0,614	0,499
Average Recall (AR) @[IoU=0.50:0.95 area= all maxDets=100]	0,725	0,622	0,634	0,513
Average Recall (AR) @[IoU=0.50:0.95 area= small maxDets=100]	0,623	0,53	0,510	0,393
Average Recall (AR) @[IoU=0.50:0.95 area=medium maxDets=100]	0,776	0,675	0,732	0,608
Average Recall (AR) @[IoU=0.50:0.95 area= large maxDets=100]	0,781	0,629	0,800	0,691

Tabla 4. Métricas de predicción de los modelos entrenados usando el framework Detectron2. Este framework entrega métricas mucho más detalladas sobre diferentes configuraciones del modelo. El porcentaje de traslapamiento de las regiones detectadas contra las regiones etiquetadas o el comportamiento en objetivos más lejanos o cercanos

8. CONCLUSIONES

El desarrollo de este proyecto presenta ciertas implicaciones a nivel de facultad y universidad, pues, permite que la universidad sea más visible en la aplicación y enseñanza de estas tecnologías emergentes. Por otro lado, para la Facultad de ingeniería y más específicamente para el programa de Ingeniería de sistemas, presenta una ampliación para el entendimiento de la aplicación de esta ingeniería, pues, dentro del programa gran parte de los trabajos de grado propenden en su gran mayoría al desarrollo de software y en mucha menor proporción a la inteligencia artificial y ciencia de datos.

Uno de los principales retos durante el desarrollo de este trabajo de grado fue la potencia computacional necesaria para el entrenamiento de modelos tan robustos como Mask-RCNN y Detectron. Como sugerencia a la Facultad de Ingeniería nos gustaría proponer la inversión en equipos con los requerimientos necesarios para trabajar con este tipo de arquitecturas, pues esta inversión facilita y motivaría la aplicación de Inteligencia Artificial en trabajos de grado futuros.

El desarrollo de este proyecto representa retos logísticos que deben ser considerados y estudiados antes de abordar la construcción de otro conjuntos de datos similares, dado que se cometieron muchos errores en el proceso de recolección y etiquetado de imágenes que repercutieron en diferentes aspectos tales como un largo y arduo proceso en la etapa de depuración, un menor número de imágenes recolectadas a las esperadas e incluso daños físicos causados por el desconocimiento del comportamiento de las vacas en el escenario en el que fueron tomados los datos.

Con los resultados obtenidos, podemos concluir que sistemas basados en arquitecturas como Mask-RCNN o Detectron2 presentan un gran potencial para su implementación en entornos productivos para el monitoreo de ganado bovino en el Oriente Antioqueño. Como una posible ampliación para este proyecto planteamos el hecho de señalar si estas arquitecturas cuentan con el potencial necesario para determinar características fenotípicas en el ganado detectado.

Durante el desarrollo de esta tesis tuvimos un aprendizaje consecuente a “un error” cometido en el proceso de recolección de datos y entrenamiento de la arquitectura. Dicho aprendizaje queda consignado como una conclusión a continuación:

- Inicialmente, al tener las imágenes previamente etiquetadas, dimos inicio a la etapa de entrenamiento, pero, al probar el modelo con los pesos resultantes, nos dimos cuenta que su precisión en la detección y clasificación no era muy buena, tras esto se realizó una revisión de que podría estar pasando, cuando llegamos al punto de evaluar el dataset, llegamos a la conclusión de que nos habíamos saltado una etapa muy importante del pipeline o metodología seguida a la hora de desarrollar modelos de Inteligencia artificial, habíamos obviado la etapa de limpieza y preparación de datos. Con este error nos quedó como aprendizaje que **la calidad de un modelo está dada por la calidad de los datos usados durante su entrenamiento.**
- Adicionalmente, se entendió que para el caso de proyectos de ciencias de datos es muy importante el seguir una línea base y una documentación continua de los experimentos realizados, esto permite que futuros proyectos se desarrollen mucho más rápido y que el trabajo realizado se vea reflejado.

Recomendamos a futuros profesionales con interés en el área del presente proyecto a escoger arquitecturas que cuenten con toda la implementación de paralelización para inferencia, cálculo de métricas y ajuste automatizado de hiper parámetros. O por otro lado usar arquitecturas más simples, ya que de lo contrario el proceso puede requerir amplios conocimientos de librerías y conceptos muy técnicos que podrían resultar frustrantes para el estudiante.

9. REFERENCIAS BIBLIOGRÁFICAS

- Redmon, Joseph and Farhadi, Ali. Yolo, tomado de <https://pjreddie.com/darknet/yolo/>
- Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. Mask R-CNN, tomado de: <https://arxiv.org/abs/1703.06870>
- Ross Girshick (2015). Fast R-CNN, tomado de: <https://arxiv.org/abs/1504.08083>
- Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN, tomado de <https://arxiv.org/abs/1506.01497>
- Ross Girshick, Yuxin Wu, Alexander Kirillov, Francisco Massa and Wan-Yen Lo. Detectron2, tomado de <https://ai.facebook.com/blog/-detectron2-a-pytorch-based-modular-object-detection-library-/>
- Danish, M. (2018). Beef Cattle Instance Segmentation Using Mask R-Convolutional Neural Network. <https://arrow.tudublin.ie/scschcomdis/128>
- Poole, D., Mackworth, A., & Goebel, R. (1998). Computational Intelligence: a logical approach. 1998, New York: Oxford University Press
- Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267-285). Springer, Berlin, Heidelberg.
- LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.
- Wen Shao, Rei Kawakami, Ryota Yoshihashi, Shaodi You, Hidemichi Kawase & Takeshi Naemura (2020). Cattle detection and counting in UAV images based on convolutional neural networks, tomado de <https://doi.org/10.1080/01431161.2019.1624858>
- Turing, A. M. (1956). Can a machine think. *The world of mathematics*, 4, 2099-2123.
- Izaurieta, F., & Saavedra, C. (2000). Redes neuronales artificiales. *Departamento de Física, Universidad de Concepción Chile*.
- Galindo, Eiber A., Perdomo, Jairo A., & Figueroa-García, Juan C.. (2020). Estudio comparativo entre máquinas de soporte vectorial multiclase, redes neuronales artificiales y sistema de inferencia neuro-difuso auto organizado para problemas de clasificación.

Información tecnológica, 31(1), 273-286 [Gráfico], Tomado de

<https://dx.doi.org/10.4067/S0718-07642020000100273>

- Yearofthedragon (2006). Artificial neuron (Computer science), Wikimedia Commons [Gráfico], tomado de <https://commons.wikimedia.org/wiki/File:Computer.Science.AI.Neuron.svg>
- Jayme Garcia Arnal Barbedo, Luciano Vieira Koenigkan, Thiago Teixeira Santos, Patrícia Menezes Santos. <https://www.mdpi.com/1424-8220/19/24/5436>
- Diana Isabel Gómez Bedoya y Reinel Castrillón. (2019). Reconocimiento automático de ganado bovino a partir de imágenes aéreas tomadas con drones: Un enfoque exploratorio, tomado de <https://revistas.utp.ac.pa/index.php/memoutp/article/view/2287/3177>
- Beibei Xua, Wensheng Wanga, Greg Falzonb, Paul Kwand, Leifeng Guoa, Guipeng Chene, Amy Taitf, Derek Schneider. Automated cattle counting using Mask R-CNN in quadcopter vision system, tomado de: <https://www.sciencedirect.com/science/article/pii/S0168169919320149>
- Google developers. (s.f.). Clase. En glosario de aprendizaje automático. Recuperado el 25 de enero de 2021, de: <https://developers.google.com/machine-learning/glossary?hl=es-419#clase-class>
- Google developers. (s.f.). Clase. En glosario de aprendizaje automático. Recuperado el 25 de enero de 2021, de: <https://developers.google.com/machine-learning/glossary?hl=es-419#falso-positivo-fp,-false-positive>
- Google developers. (s.f.). Clase. En glosario de aprendizaje automático. Recuperado el 25 de enero de 2021, de: <https://developers.google.com/machine-learning/glossary?hl=es-419#precisi%C3%B3n-precision>
- Google developers. (s.f.). Clase. En glosario de aprendizaje automático. Recuperado el 25 de enero de 2021, de: <https://developers.google.com/machine-learning/glossary?hl=es-419#exactitud-accuracy>
- Google developers. (s.f.). Clase. En glosario de aprendizaje automático. Recuperado el 25 de enero de 2021, de:

<https://developers.google.com/machine-learning/glossary?hl=es-419#recuperaci%C3%B3n-recall>

- Detectron2 (2019 - 2020). *detectron2.readthedocs.io*, recuperado de <https://detectron2.readthedocs.io/en/latest/notes/benchmarks.html>

10. APÉNDICES

10.1. **Apéndice 1:** Notebooks interactivo en Google Colab en donde se evidencia el código y los resultados arrojados en la transferencia de aprendizaje de Mask RCNN.

- <https://colab.research.google.com/drive/1xsfldKjhUWtGHnmyzwwgal3CQrmCyTWe7?usp=sharing>
- <https://colab.research.google.com/drive/1z-2fY2YFawRXY5IpRrvlSL8rI1HWMlJS?usp=sharing>

10.2. **Apéndice 2:** Notebook interactivo en Google Colab en donde se evidencia el código y los resultados arrojados en la transferencia de aprendizaje de Detectron2 enfoque 1

- https://colab.research.google.com/drive/12H9GjkJhzBhHjR6Hs_eVYBGAsuJdTNSo?usp=sharing

10.3. **Apéndice 3:** Notebook interactivo en Google Colab en donde se evidencia el código y los resultados arrojados en la transferencia de aprendizaje de Detectron2 Enfoque 2

- https://colab.research.google.com/drive/1o0t8I-Z1BYSqG1Pfk4a3_MHrAITsQDet?usp=sharing

10.4. **Apéndice 4:** Repositorio de GitHub donde se encuentra el dataset. (el dataset aún no se encuentra publico pero, será liberado en los próximos meses).

- <https://github.com/cows-mask-rcnn/dataset>