



# **ESCUELA POLITÉCNICA NACIONAL**



## **FACULTAD DE INGENIERÍA MECÁNICA**

### **ESTUDIO DE PROPIEDADES DE FÍSICA TÉRMICA A PARTIR DE IMÁGENES DE UN UAV DE ALAS FIJAS USANDO REDES NEURONALES**

#### **TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO MECÁNICO**

**LÓPEZ SIGUENCIA JONATHAN LEONARDO**  
jonathan.lopez@epn.edu.ec

**MONTALVÁN SAMANIEGO JIMMY FRANCISCO**  
jimmy.montalvan@epn.edu.ec

**DIRECTOR: Ing. VALENCIA TORRES ESTEBAN ALEJANDRO, PhD.**  
esteban.valencia@epn.edu.ec

**DIRECTOR: Ing. HIDALGO DÍAZ VÍCTOR HUGO, DSc.**  
victor.hidalgo@epn.edu.ec

**SEPTIEMBRE, 2022**

## **CERTIFICACIÓN**

Certificamos que el presente trabajo fue desarrollado por JONATHAN LEONARDO LÓPEZ SIGUENCIA Y JIMMY FRANCISCO MONTALVÁN SAMANIEGO, bajo nuestra supervisión

\_\_\_\_\_  
Ing. Esteban Alejandro Valencia, PhD

DIRECTOR DE PROYECTO

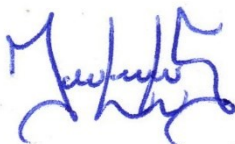
\_\_\_\_\_  
Ing. Víctor Hugo Hidalgo, DSc

CODIRECTOR DE PROYECTO

## DECLARACIÓN

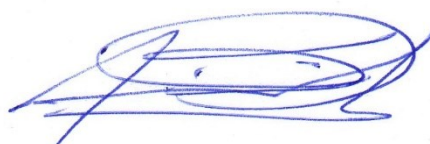
Nosotros, JONATHAN LEONARDO LÓPEZ SIGUENCIA Y JIMMY FRANCISCO MONTALVÁN SAMANIEGO, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.



---

Jonathan Leonardo López Siguencia



---

Jimmy Francisco Montalván Samaniego

## DEDICATORIA

Para mis tres pilares.

Mi amada Madre, por su apoyo incondicional, enseñanzas de vida que nunca olvidaré, su esfuerzo, sacrificio y dedicación hacia mí.

Mi querido Padre, por su perseverancia, constancia, buenos valores y deseos de superación hacia mí.

Mi inigualable hermano, por sus consejos de vida, todos los inolvidables momentos que hemos vivido y enseñarme a ser una mejor persona.

Mami, Papi, Ñaño esto es de ustedes y por ustedes, los amo.

Jonathan

A la mujer que nunca me soltó de la mano que estuvo día y noche pendiente de mí, haciéndome creer que todo esto es posible, gracias por todo tu amor y cariño mi madrecita hermosa.

Al mejor padre que un hijo puede tener, gracias por darme el ejemplo que con humildad y perseverancia se puede lograr todos los objetivos.

A mis dos hermanos por su amor y sabiduría, la vida no me pudo dar mejores hermanos que ustedes.

A Michelle Panchi gracias por caminar juntos de la mano en los momentos difíciles y por compartir los momentos hermosos de mi vida.

A Nancy Montalván gracias por enseñarme amar desde la distancia.

Jimmy

## **AGRADECIMIENTO**

A mi Mami, gracias por siempre apoyarme en todo.

A mi Papi, gracias por siempre estar en todo.

A mi Ñaño, gracias por siempre pensar en mí y no abandonarme.

A mis tutores, gracias por todo lo enseñado en esta tesis.

A todos mis amigos de la Poli, gracias por todo muchachos fueron los mejores momentos.

A María Fernanda Cassis Jefa de Manufactura de Tigre Ecuador, gracias por brindarme todas las facilidades para el desarrollo final de esta tesis.

A Julio Narváez amigo mío, gracias por confiar en mí, la Poli nos hizo amigos y el trabajo nos hizo hermanos.

A todos mis amigos de mi actual trabajo, gracias por todo muchachos son los mejores momentos.

A mi mejor amigo Jimmy, por lo vivido, sufrido, conversado y soñado, gracias por todo hermano.

Jonathan

A mi familia Montalván Samaniego sin ustedes esto no sería posible.

Al grupo ATA en especial al Dr. Esteban Valencia por su apoyo, tiempo, conocimiento y experiencia al dirigir la presente tesis, al Dr. Víctor Hidalgo y Ing. Víctor Alulema por su colaboración y guía.

A mis amigos de la poli, gracias por compartir toda esta etapa universitaria, creando momentos únicos e irrepetibles.

A mis amigos del colegio por siempre brindarme su apoyo y ganas de salir adelante.

A mi hermano Jonathan por iniciar y finalizar esta etapa universitaria tal y como un día lo trazamos.

Jimmy

## Contenido

CERTIFICACIÓN.....	2
DECLARACIÓN.....	3
DEDICATORIA.....	4
AGRADECIMIENTO.....	5
RESUMEN .....	10
INTRODUCCIÓN.....	11
OBJETIVO GENERAL.....	12
OBJETIVOS ESPECÍFICOS.....	12
1. MARCO TEÓRICO .....	13
1.1. Tecnificación de monitoreo aéreo en ecosistemas .....	13
1.1.1. Inspección de monitoreos con mecanismos UAV .....	13
1.2. Análisis del producto en el mercado (Benchmarking) .....	15
1.3. Mejoramiento de la resolución de imágenes.....	19
1.3.1. Métodos de interpolación.....	19
1.3.2. Métodos basados en redes neuronales artificiales.....	21
1.3.3. Métodos de medición de la calidad de imágenes.....	27
1.4. Cobertura vegetal mediante el procesamiento de imágenes .....	28
1.4.1. Modelo de colores .....	28
1.4.2. Contenido de agua en humedales altoandinos .....	30
2. METODOLOGÍA.....	31
2.1. Selección del mecanismo de sujeción.....	31
2.1.1. Casa de la Calidad .....	31
2.1.1.1. Voz del usuario .....	32
2.1.1.2. Voz del ingeniero .....	33
2.1.1.3. Resultados de la casa de la calidad .....	34
2.1.2. Análisis Funcional.....	35
2.1.2.1. Diagramas Funcionales.....	35
2.1.2.2. Análisis de diagramas funcionales .....	36
2.1.3. Subsistemas Funcionales .....	36
2.1.3.1. Subsistema 1: Base del mecanismo.....	37
2.1.3.2. Subsistema 2: Mecanismo de sujeción 1.....	37
2.1.3.3. Subsistema 2: Mecanismo de sujeción 2.....	38
2.1.4. Selección del diseño del mecanismo de sujeción .....	39
2.2. Selección de la red neuronal .....	42
2.3. Adaptación de la GAN.....	44
2.3.1. Adquisición y preparación de base de datos.....	45
2.3.2. Adaptación de la GAN para el entrenamiento .....	47

2.3.3. Adaptación de la GAN para el testeo .....	48
2.4. Aplicación de la metodología de estudio de propiedades físico - térmicas .....	49
3. RESULTADOS Y DISCUSIÓN .....	51
3.1. Mecanismo de sujeción de la cámara.....	51
3.2. Aumento de resolución de imágenes.....	52
3.3. Aplicación de la metodología de procesamiento de imágenes .....	54
3.4. Comparación de la metodología de procesamiento de imágenes con imágenes de la zona de estudio. ....	58
4. CONCLUSIONES .....	60
4.1. Conclusiones.....	60
4.2. Trabajos futuros .....	60
REFERENCIAS BIBLIOGRÁFICAS.....	62
ANEXOS .....	67
ANEXO I.....	67
ANEXO II.....	68
ANEXO III.....	74
ANEXO IV .....	76
ANEXO V .....	79
ANEXO VI .....	81

## Índice de Figuras

Figura 1.1: Ejes de rotación de gimbal.....	15
Figura 1.2: Gimbal de dos grados de libertad (Yaw y Pitch).....	16
Figura 1.3: Gimbal de dos grados de libertad (Roll y Pitch) .....	16
Figura 1.4: Gimbal de tres grados de libertad .....	17
Figura 1.5: Gimbal USG – 301 EO.....	17
Figura 1.6: Gimbal Sheen .....	18
Figura 1.7: Gimbal Epsilon 140.....	18
Figura 1.8: Comparación de los métodos de interpolación. (izq) Interpolación bilineal, (cen) Interpolación bicúbica, (der) El vecino más cercano .....	20
Figura 1.9: Esquemas de comportamiento de los métodos de interpolación. (a) El vecino más cercano, (b) Interpolación bilineal, (c) interpolación bicúbica .....	20
Figura 1.10: Resultado de la red neuronal más simple .....	21
Figura 1.11: Aplicaciones de la Red Neuronal CNN .....	22
Figura 1.12: Asignación de filtro deslizado en una imagen .....	22
Figura 1.13: Función de activación ReLu.....	23
Figura 1.14: Capa de agrupación o Pooling.....	23
Figura 1.15: Esquema de la red CNN .....	23
Figura 1.16: Esquema de una red neuronal antagonica.....	25

Figura 1.17: Diagrama del generador de una res neuronal antagónica.....	25
Figura 1.18: Diagrama del discriminador de una red antagónica .....	26
Figura 1.19: Matiz de colores.....	29
Figura 1.20: Saturación de colores .....	30
Figura 1.21: Valor de colores.....	30
Figura 1.22: Humedal del Antisana.....	31
Figura 2.1: Casa de la calidad .....	34
Figura 2.2: Diagrama funcional nivel 0 del mecanismo de sujeción .....	35
Figura 2.3: Diagrama funcional nivel 1 del mecanismo de sujeción .....	36
Figura 2.4: Subsistema 1: Base del mecanismo .....	37
Figura 2.5: Subsistema 2: Mecanismo de sujeción 1 .....	38
Figura 2.6: Subsistema 2: Mecanismo de sujeción 2 .....	39
Figura 2.7: Subsistema 2: Mecanismo final para la sujeción de la cámara .....	41
Figura 2.8: Comparación del método de interpolación bicubica. (izq) Imagen de baja resolución, (der) imagen interpolada.....	42
Figura 2.9: Comparación del método de CNN. (izq) Imagen de baja resolución, (der) imagen procesada .....	43
Figura 2.10: Comparación del método de GAN. (izq) Imagen de baja resolución, (der) imagen procesada .....	44
Figura 2.11: Características de la GPU proporcionada por Google Colab Pro. ....	45
Figura 2.12: Diagrama de flujo para la adquisición y preparación de la base de datos. ....	45
Figura 2.13: Muestra de la base de datos para el entrenamiento de la GAN .....	47
Figura 2.14: Diagrama de flujo para la adaptación de la GAN para el entrenamiento. ....	47
Figura 2.15: Diagrama de flujo para la adaptación de la GAN para el testeo .....	48
Figura 2.16: Diagrama de flujo para la aplicación de la metodología de estudio de propiedades físico – térmicas .....	49
Figura 2.17: Rango de tonalidad de la zona de interés para zonas forestales .....	50
Figura 3.1: Representación del mecanismo final de sujeción de la cámara .....	51
Figura 3.2: Comparación de las cinco GANs entrenadas.....	52
Figura 3.3: (Izq) Comparación visual de la cuarta GAN entrenada. (Der) Comparación visual de la quinta GAN .....	53
Figura 3.4: Comparación de los resultados de las métricas de la cuarta y quinta GAN entrenada .....	53
Figura 3.5: Imagen Antisana_Caso 1.....	55
Figura 3.6: Resultados de la aplicación de la metodología de estudio de propiedades físico-térmicas. Imagen Antisana_Caso 1 .....	56
Figura 3.7: Imagen Antisana_Caso 2 - Original .....	57
Figura 3.8: Imagen Antisana_Caso 2.....	57
Figura 3.9: Resultados de la aplicación de la metodología de estudio de propiedades físico-térmicas. Imagen Antisana_Caso 2. ....	58



## Índice de tablas

Tabla 1.1: Especificaciones técnicas del Event.....	14
Tabla 1.2: Especificaciones técnicas del Skywalker .....	14
Tabla 1.3: Especificaciones técnicas del Boeing X48 Híbrido .....	15
Tabla 1.4: Ficha técnica del gimbal USG - 301 EQ.....	17
Tabla 1.5: Ficha técnica del gimbal Sheen .....	18
Tabla 1.6: Ficha técnica del gimbal Epsilon 140 .....	19
Tabla 2.1: Resultados de la casa de la calidad .....	35
Tabla 2.2: Valores designados para los criterios.....	40
Tabla 2.3: Evaluación del peso específico de cada criterio .....	40
Tabla 2.4: Evaluación del peso específico del criterio de peso .....	40
Tabla 2.5: Evaluación del peso específico del criterio de dimensiones .....	40
Tabla 2.6: Evaluación del peso específico del criterio de ejes de rotación .....	41
Tabla 2.7: Ponderación final .....	41
Tabla 3.1: Comparación de los resultados de las cinco GANs entrenadas .....	52
Tabla 3.2: Rangos de segmentación .....	54
Tabla 3.3: Propiedades de la imagen para el caso 1 .....	55
Tabla 3.4: Propiedades de la imagen para el caso 2 .....	57
Tabla 3.5: Comparación de resultados del procesamiento de imágenes de los dos casos de estudio.....	59

## RESUMEN

Esta tesis presenta el estudio de propiedades de física térmica a partir de imágenes de un UAV de alas fijas usando redes neuronales. Esto se desarrolla en tres etapas: selección del mecanismo para la estabilización de una cámara de baja resolución, aumento de resolución de imágenes y finalmente la aplicación de la metodología de procesamiento de imágenes para estudio de propiedades de física-térmica, validada en el trabajo de Diana Puga.

Para la primera etapa se propone dos alternativas para un mecanismo que permita sujetar una cámara de baja resolución en un UAV de alas fijas. Se realiza un análisis de la casa de la calidad, a través de una encuesta para saber la voz del usuario y combinarla con la voz del ingeniero y de esta manera disminuir el costo para obtener imágenes aéreas y peso de la carga útil del UAV.

En la segunda etapa correspondiente al aumento de resolución de imágenes, se emplean redes neuronales GANs pre entrenadas en las cuales se modifica su código para adaptarlas y entrenarlas con una nueva base de datos compuesta por imágenes de entornos naturales. Los códigos son alterados para que puedan ser ejecutados en los entornos de Google Colab Pro y Google Drive Standard ya que ambos productos proporcionan las herramientas computacionales necesarias para la ejecución de los códigos. Cabe mencionar que los beneficios de Google Colab Pro y Google Drive Standard han sido suspendidos una vez finalizado el presente trabajo. Los códigos modificados quedan para el dominio público en una cuenta GitHub para futuros trabajos y todos los respaldos han sido entregados en conjunto con esta tesis al grupo de investigación ATA - FIM de la Escuela Politécnica Nacional.

Y como etapa final, se aplica la metodología del procesamiento de imágenes a color usando el modelo HVS validada en los trabajos de Diana Puga, María Victoria Granja y Ricardo Poveda, en donde se logra determinar la cobertura vegetal de zonas forestales en base al porcentaje de humedad del suelo.

Palabras clave: mecanismo de sujeción, UAV, redes neuronales, GAN, procesamiento de imágenes, propiedades de física-térmica, ATA – FIM.

# **ESTUDIO DE PROPIEDADES DE FÍSICA TÉRMICA A PARTIR DE IMÁGENES DE UN UAV DE ALAS FIJAS USANDO REDES NEURONALES**

## **INTRODUCCIÓN**

En los últimos años los UAS han tenido un gran potencial en distintas áreas como: la agricultura, geología, cartografía, fotografía, aplicaciones militares y medio ambientales, todo esto debido a su rápido desplazamiento sobre lugares irregulares superando cualquier tipo de dificultad a un menor costo comparado con sistemas aéreos tripulados, no obstante los UAS han presentado limitantes debido que al momento de realizar el monitoreo la cámara solo puede realizar tomas en una sola posición, generando dificultades que han conllevado a diseñar un sistema que permita realizar tomas en distintas direcciones cubriendo necesidades concretas en diversos sistemas aprovechando conjuntamente el uso de técnicas de la inteligencia artificial.

En consecuencia varios investigadores e ingenieros han desarrollado métodos que permiten el aumento de la resolución de imágenes, como por ejemplo el método del vecino más cercano [1], interpolación bilineal [2], [3], interpolación bicúbica, entre otros, los mismo que han presentado limitaciones, tanto visualmente como en las métricas de la calidad de imagen [4]. Actualmente el desarrollo de la inteligencia artificial mediante la utilización de redes neuronales se ha incrementado de manera exponencial en las últimas décadas, favoreciendo su aplicación en distintas áreas de información fotográfica digital como son: la reconstrucción de imágenes, el mejoramiento de la resolución, reconocimiento de objetos y generación tridimensional, presentando una ventaja abismal con respecto a métodos convencionales lineales y no lineales. [5][6][7][8][9]

Según diversos trabajos, la utilización de imágenes digitales para el estudio de propiedades físico-térmicas en zonas forestales se ha convertido en un método de menor costo en comparación con los métodos tradicionales de medición en campo. Además, los resultados obtenidos al usar imágenes digitales en investigaciones de propiedades físico-térmicas han presentado porcentajes de error pequeños al compararlos con los resultados obtenidos en campo. [10], [11]

Es por todo lo mencionado que se busca implementar un mecanismo a una UAV

de alas fijas que permita capturar imágenes de baja resolución en distintas posiciones para posteriormente procesarlas y obtener información de utilidad para diferentes estudios.

## **OBJETIVO GENERAL**

Estudiar las propiedades de física térmica a partir de imágenes de un UAV de alas fijas usando redes neuronales

## **OBJETIVOS ESPECÍFICOS**

- Recopilar el estado del arte entorno a sistemas de recolección de imágenes de UAV's de alas fijas.
- Seleccionar un mecanismo de sujeción de la cámara de baja resolución que permita posicionarla en distintos ángulos.
- Aplicar algoritmos GAN's que permitan el mejoramiento de la resolución de las imágenes capturadas.
- Aplicar la metodología de procesamiento de imagen para el estudio de propiedades de física-térmica.
- Comparar la metodología con imágenes de prueba correspondientes a la zona de estudio.

## **1. MARCO TEÓRICO**

### **1.1. Tecnificación de monitoreo aéreo en ecosistemas**

Los humedales altoandinos son ecosistemas de gran importancia para el desarrollo y sostenibilidad regional puesto que son la mayor fuente de abastecimiento de agua para varias ciudades Andinas. En particular, para la ciudad de Quito, estos ecosistemas representan el 85% del abastecimiento de agua de toda la ciudad.[12] Además, los humedales también contribuyen con el almacenamiento y regulación de caudales de las vertientes de los nevados, la generación de energía hidroeléctrica, la sostenibilidad de la biodiversidad y la fijación de carbono [13]. Por tal motivo en la actualidad numerosos científicos, ingenieros e investigadores han desarrollado distintos tipos de metodologías para el monitoreo y cuidado de ecosistemas sensibles debido a la carencia de datos históricos, llegando a tener una mejor apreciación en distintos tipos de ecosistemas que se detallan a continuación.

#### **1.1.1. Inspección de monitoreos con mecanismos UAV**

El desarrollo de monitoreo aéreo con UAV en los últimos años ha brindado nuevas oportunidades especialmente para el desarrollo de sistemas de aeronaves pilotadas a distancia.[14] Los UAVs se ven favorecidos debido a su capacidad para transportar una variedad de sensores espectrales y una accesibilidad conveniente a los sitios objetivo-locales.[15] Teniendo en cuenta el costo económico y la seguridad del equipo, la carga útil de los UAVS son comúnmente con cámaras espectrales, Lidars, GPS, y gimbals, en entornos extremos, como las regiones ventosas y de gran altitud. Se ha demostrado su eficiencia en la captura de identidad de especies.[16], [17]

Especialmente para ecosistemas forestales las imágenes UAVs pueden proporcionar productos de resolución espacial, abriendo oportunidades para monitorear pastos de baja estatura. Surgiendo nuevas oportunidades para obtener características de los ecosistemas a una escala más allá del paisaje y explorar sus relaciones con los factores ambientales. En este ámbito se han desarrollado

estudios de monitoreos en ecosistemas altoandinos con la utilización de UAVs como son: EVENT38, SKYWALKER X8 Y BOEING X48 HIBRIDO por el grupo de Aeronáutica y Termodinámica Aplicada (ATA) de la Escuela Politécnica Nacional encontrándose con limitantes con el peso de cada uno de los componentes.

## Event 38

Es uno de los drones de ala fija líderes en aplicaciones como son la agricultura, geología, minería y monitoreo de ecosistemas debido a su resistencia a distintos tipos de climas probado en junglas cálidas, húmedas y en altas montañas.

*Tabla 1.1: Especificaciones técnicas del Event*

UAV	Wing span [m]	Height [m]	Fuselage length [m]	Take off weight [kg]	Payload weight [kg]	Operational altitude [m]	Operating speed [Km/h]	Endurance [h]
EVENT 38	1,9	0,351	1,3	2,5	1	7000	46,8	1,5

(Fuente: [18])

## Skywalker

El Skywalker es un UAV pequeño de ala fija utilizado principalmente en aplicaciones como es la cartografía aérea debido a su facilidad de maniobra presentando winglets con distintos tipos de configuraciones en los extremos de las alas dando una mejor eficiencia, no obstante, los aumentos de peso en su carga útil llegan afectar su estabilidad y planeo es por tal motivo que son UAV de que su peso bruto es inferior con respecto a otros UAVs.

*Tabla 1.2: Especificaciones técnicas del Skywalker*

UAV	Wing span [m]	Height [m]	Fuselage length [m]	Take off weight [kg]	Payload weight [kg]	Operational altitude [m]	Operating speed [Km/h]	Endurance [h]
SKYWALKER	2,12	0,28	0,79	3	1,5	3000	32,5	1,5

(Fuente: [19])

## Boeing X48 híbrido

El modelo Boeing X48 tiene un fuselaje de cuerpo de ala combinada, dicho modelo

presenta distintos tipos de configuraciones como son de combustión interna, híbridos y totalmente eléctrico teniendo cada uno ventajas y desventajas uno con respecto al otro, cabe mencionar para este caso se está guiando la configuración híbrida propuesta por el grupo ATA en su artículo denominado “Modeling of a series hybrid propulsion UAV used for monitoring in the Galapagos Islands”, esta configuración tiene como propósito monitorear lugares de largo alcance donde se necesite una autonomía mayor de cinco horas de vuelo. [20]

Tabla 1.3: Especificaciones técnicas del Boeing X48 Híbrido

UAV	Wing span [m]	Height [m]	Fuselage length [m]	Take off weight [kg]	Payload weight [kg]	Operational altitude [m]	Operating speed [Km/h]	Endurance [h]
BOEING	6,22	-	-	160	33,27	3000	216	5

(Fuente: [20])

Estos tres tipos de modelos UAVs se desea utilizar para el monitoreo en la reserva hídrica del Antisana, anhelando implementar un mecanismo de sujeción de la cámara eliminando a los gimbals para alivianar el peso de carga útil, sacrificando la perdida de resolución de imágenes, pero dando como principal ventaja una mejor autonomía y la utilización de cámara de menor costo.

## 1.2. Análisis del producto en el mercado (Benchmarking)

Existen mecanismos que permiten la estabilización de cámaras en UAV's denominados Gimbal, estos son dispositivos que al estar formados por un soporte y dos o tres pares de pivotes permiten obtener cualquier orientación en el espacio tridimensional.[21]

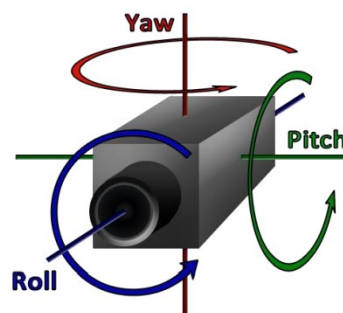


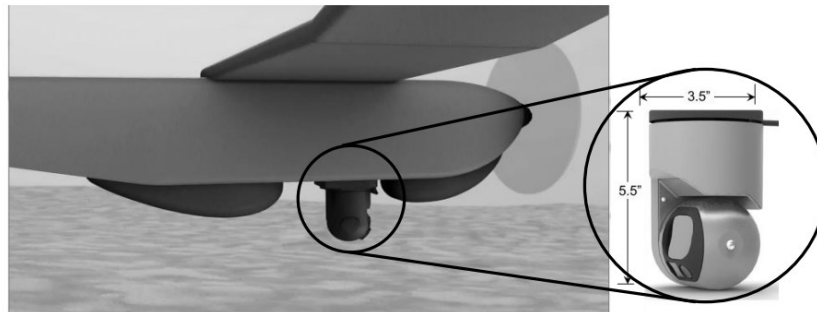
Figura 1.1: Ejes de rotación de gimbal (Fuente:[22] )

En la Figura 1.1 se muestra los tres ejes de rotación que posee un gimbal; cada eje tiene asignado un nombre en función del movimiento que gobierna y estos son: el eje longitudinal (Pitch), el eje transversal (Roll) y el eje vertical (Yaw).

Comúnmente los gimbal pueden ser de dos o tres grados de libertad y que por lo general los gimbal de dos grados de libertad son usados por UAV's de alas fijas, mientras que los UAV's de alas móviles utilizan gimbal de tres grados de libertad.[23]

### **Gimbal de dos grados de libertad**

Los gimbal de dos grados de libertad se pueden presentar en dos combinaciones de ejes de rotación, Yaw y Pitch o Roll y Pitch



*Figura 1.2: Gimbal de dos grados de libertad (Yaw y Pitch)  
(Fuente:[24] )*



*Figura 1.3: Gimbal de dos grados de libertad (Roll y Pitch)  
(Fuente:[25] )*

### **Gimbal de tres grados de libertad**

Los gimbal de tres grados de libertad presentan los tres ejes de estabilización, Pitch, Roll y Yaw



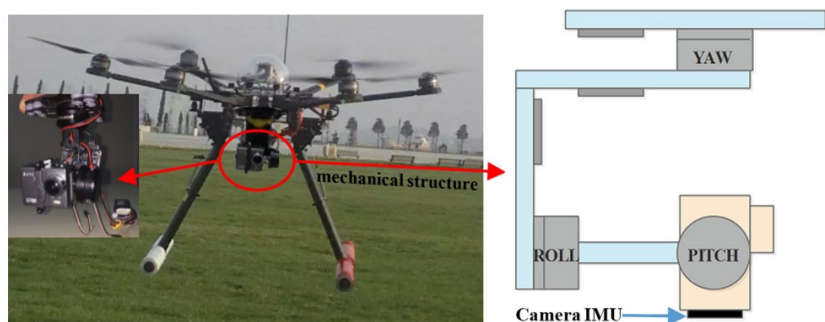


Figura 1.4: Gimbal de tres grados de libertad  
(Fuente:[26] )

En la Figura 1.4 se describe los ejes de un gimbal de tres grados de libertad y se logra apreciar su implementación en un UAV

La compañía UKRSPEC oferta un gimbal USG-301 EO de tres grados de libertad con un cardan giro estabilizado dando una mejor calidad de imagen por su sistema de antivibración de tal forma que las imágenes tengan un mejor enfoque, dicho gimbal este diseñado principalmente para aplicaciones industriales como son seguridad, búsqueda y rescate, cabe mencionar que su limitante de este gimbal tiende a ser su compatibilidad con el tipo de drones.



Figura 1.5: Gimbal USG – 301 EO  
(Fuente:[27])

Tabla 1.4: Ficha técnica del gimbal USG - 301 EQ

FICHA TÉCNICA	
EJES	3
PESO [kg]	2,2
LONGITUD [mm]	160x266
PROTECCIÓN	LLUVIA Y NIEVE
COMPATIBILIDAD CON DRONES	DJI S1000 Y MATRICE 600

(Fuente: [27] )

Sheen es una empresa dedicada totalmente al diseño y construcción de componentes de carga útil para drones, ofertando un gimbal de 3 ejes, siendo ligeramente liviano, teniendo una montura fija e incorporado una cámara de 12 MP con calidad en 4K siendo una opción precio vs calidad caro, cabe mencionar que este tipo de gimbal es compatible con cualquier tipo de dron.



Figura 1.6: Gimbal Sheen  
(Fuente:[28] )

Tabla 1.5: Ficha técnica del gimbal Sheen

FICHA TÉCNICA	
EJES	3
PESO [kg]	0,275
LONGITUD [mm]	96x79x120
RANGO MECÁNICO	Yam: $\pm 75^\circ$ ; Pitch: $310^\circ$ ; Roll: $+90^\circ$
COMPATIBILIDAD CON CÁMARAS	SONY

(Fuente: [28])

Octopus es una empresa pionera en el desarrollo de cámaras para distintos tipos de aplicaciones, conjuntamente el progreso de gimbals para drones teniendo como resultado productos como el Epsilon 140 utilizado comúnmente en drones de ala fija debido a su diseño y peso, no obstante, cabe mencionar que su precio es relativamente alto.



Figura 1.7: Gimbal Epsilon 140  
(Fuente: [29])

Tabla 1.6: Ficha técnica del gimbal Epsilon 140

FICHA TÉCNICA	
EJES	3
PESO [kg]	1,57
LONGITUD [mm]	140 x 189
RANGO MECÁNICO	Yaw: $\pm 360^\circ$ ; Pitch: $45^\circ$ ; Roll: $-90^\circ$
PROTECCIÓN	IP64

(Fuente: [29])

### 1.3. Mejoramiento de la resolución de imágenes

El mejoramiento o aumento de la resolución de imágenes digitales mediante algoritmos y procesamientos computacionales se ha convertido en un medio mucho más accesible y económico para obtener imágenes de buena calidad. El objetivo no solo es llegar a tener una imagen visualmente aceptable, sino que también, según el contexto de estudio, permita obtener datos lo más realistas posible y que una vez procesados brinden resultados confiables.

Las interpolaciones en años anteriores fueron los métodos más aplicados para poder realizar procesamientos de imágenes ya que se alcanzaba resultados aceptables, pero con el paso del tiempo y gracias al desarrollo tecnológico e investigaciones, nuevos métodos se han ido desarrollando los cuales logran resultados superiores a los métodos convencionales, llegando incluso a implementar temas como la super resolución de imágenes como es el caso de las redes neuronales artificiales.

#### 1.3.1. Métodos de interpolación

En términos generales el proceso de interpolar consiste en determinar el valor de cierto parámetro en las posiciones intermedias de las muestras originales mediante la aplicación y evaluación de una función. En este contexto y a lo largo del tiempo, los métodos de interpolación han llegado a desempeñar un papel importante en el campo del procesamiento de imágenes digitales con el objetivo de mejorar la resolución de estas. Existen tres métodos de interpolación: el vecino más cercano, interpolación bilineal e interpolación bicúbica. [3], [30]



Figura 1.8: Comparación de los métodos de interpolación. (izq) Interpolación bilineal, (cen) Interpolación bicúbica, (der) El vecino más cercano  
(Fuente [3])

## El vecino más cercano

Es el método de interpolación más simple ya que el valor asignado al punto interpolado corresponde al punto más cercano, según las coordenadas de dicho punto.

## Interpolación bilineal

La interpolación bilineal está basada en el método del vecino más cercano, pero considerando los cuatro puntos más cercanos al punto interpolado para determinar su valor.

## Interpolación bicúbica

Este método utiliza una función bicúbica para interpolar el valor de determinado punto, pero considerando los 16 puntos más próximos a dicho punto. Al igual que el método anterior, este también se basa en el método del vecino más cercano, básicamente lo que hace en primer lugar es cuatro interpolaciones bilineales horizontales y posteriormente cuatro interpolaciones bilineales verticales. [2]

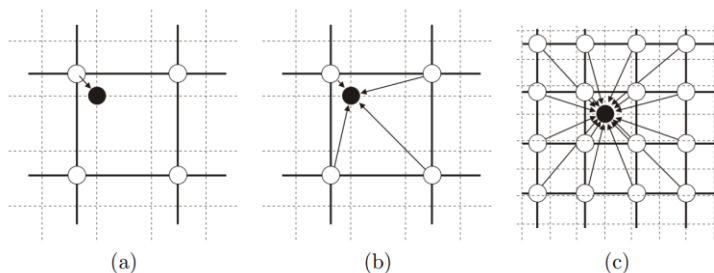
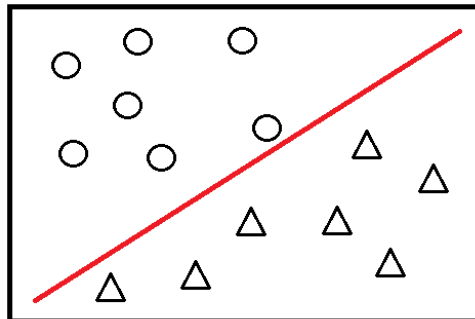


Figura 1.9: Esquemas de comportamiento de los métodos de interpolación. (a) El vecino más cercano, (b) Interpolación bilineal, (c) interpolación bicúbica  
(Fuente: [30] )

### 1.3.2. Métodos basados en redes neuronales artificiales

El método de las redes neuronales artificiales para el procesamiento de imágenes es un modelo basado en comportamiento biológico del cerebro, en donde las neuronas transmiten información entre ellas ya que se encuentran conectadas unas con otras. El acto de compartir datos, analizarlos y procesarlos, le permite al cerebro aprender y solucionar diferentes tipos de necesidades.

Si bien el concepto del método de las redes neuronales artificiales no es para nada nuevo, su aplicación práctica si, ya que siempre se veía comprometida por falta de capacidad de almacenamiento de datos, velocidad de procesamiento, entornos para su programación y ejecución, entre otros. Solo con el avanzado desarrollo tecnológico de los últimos años que ha permitido obtener recursos computacionales lo suficientemente aptos, se ha podido continuar con la investigación e implementación de las redes neuronales artificiales en distintos campos, uno de ellos el procesamiento de imágenes digitales. [31]



*Figura 1.10: Resultado de la red neuronal más simple  
(Fuente: Propia)*

### **CNN (Red Neuronal Convolutional)**

Las redes neuronales convolucionales (CNN) son algoritmos de inteligencia artificial basados en redes neuronales multicapa que aprenden características relevantes de las imágenes, siendo capaces de realizar varias tareas como clasificación, detección y segmentación de objetos.[8], [32]–[35]

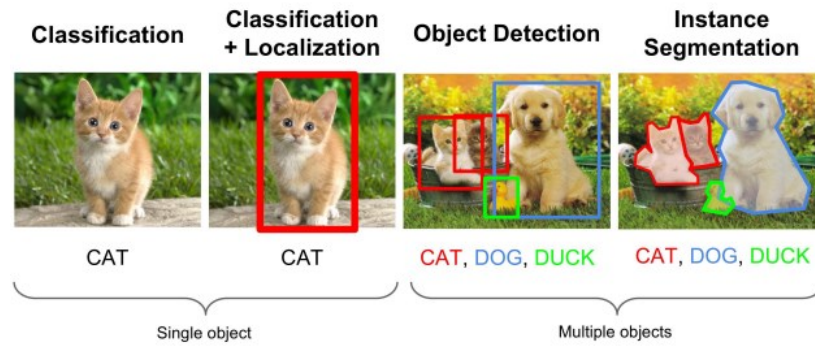


Figura 1.11: Aplicaciones de la Red Neuronal CNN  
(Fuente [36])

Las redes neuronales convolucionales se componen de cuatro tipos de capas que se detallan a continuación:

## Convolutacional

Una capa convolutacional está compuesta por un conjunto de filtros, también llamados Kernel, que se deslizan sobre los datos de entrada. Cada Kernel tiene pesos de ancho, alto y ancho  $\times$  alto que se utilizan para extraer características de los datos de entrada. Cada filtro en la capa convolutacional representa una entidad o asignación de número que van de 0 a 255. Cuando el filtro se desliza sobre la imagen y encuentra una coincidencia, la operación de convolución genera un gran número, activando el filtro a esa característica, caso contrario, el filtro no se activa.

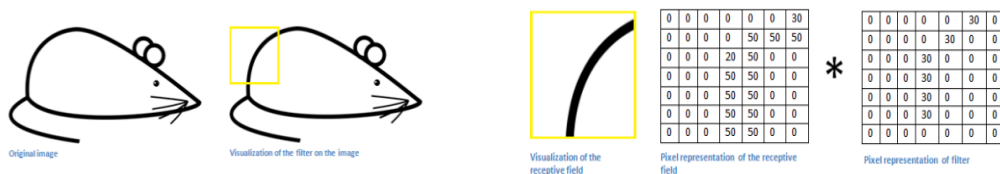


Figura 1.12: Asignación de filtro deslizado en una imagen  
(Fuente :[36])

## ReLU

Las capas ReLu, es una capa de activación unida después de una capa convolutacional para generar no linealidad en la red. El ReLu ayuda a la red a aprender funciones de decisión más difíciles.

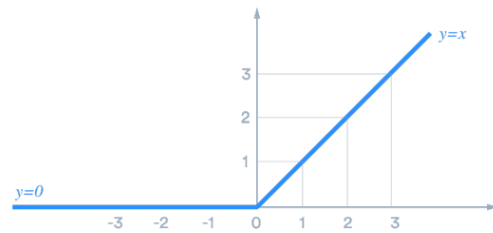


Figura 1.13: Función de activación ReLu  
(Fuente: Propia)

## Capa de agrupación

La capa de agrupación, o capa de muestreo descendente, se aplica para reducir la dimensionalidad de los mapas de características de manera que se guarde la información más relevante de los mapas de características.

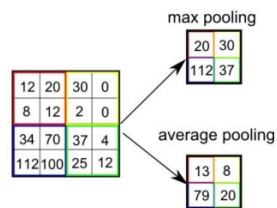


Figura 1.14: Capa de agrupación o Pooling  
(Fuente: [36] )

## Capa completamente conectada

Una CNN se divide en dos partes: El primero aprende las mejores funciones para extraer de las imágenes y el segundo aprende cómo clasificar las funciones en diferentes categorías.

La capa Totalmente Conectada es un Perceptrón Multicapa (MLP), compuesto por tres tipos de capas: capas de entrada, ocultas y de salida

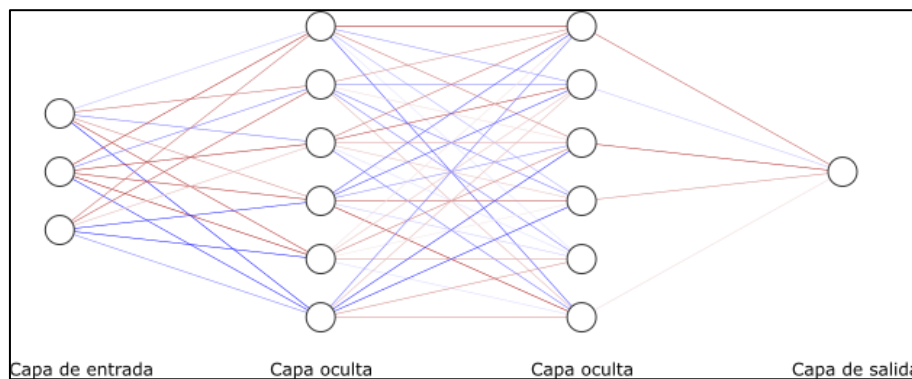


Figura 1.15: Esquema de la red CNN  
(Fuente: Propia)

## **Entrenamiento**

Para entrenar un modelo CNN, se utiliza un conjunto de datos de entrenamiento compuesto por un conjunto de imágenes y etiquetas (clases, cuadros delimitadores, máscaras). El algoritmo utilizado para entrenar una CNN se llama retro propagación y utiliza el valor de salida de la última capa para medir un valor de error. Este valor de error se usa para actualizar los pesos de cada neurona en esa capa. Los nuevos pesos se utilizan para medir un valor de error y actualizar los pesos del anterior. El algoritmo repite el proceso hasta llegar a la primera capa.

## **GAN (Red Neuronal Antagónica)**

Una GAN es una arquitectura de red neuronal profunda que está compuesta por dos redes, una red generadora y una red discriminadora, estas redes tienen múltiples períodos de generación y discriminación, entrenándose entre sí, al mismo tiempo que intentan ser más astutas una de la otra.

El Generador es una red generadora que utiliza datos reales para generar nuevos datos, siendo este el objetivo principal que el discriminador clasifique erróneamente su salida como real a partir de un vector de números generado aleatoriamente, denominado espacio latente.

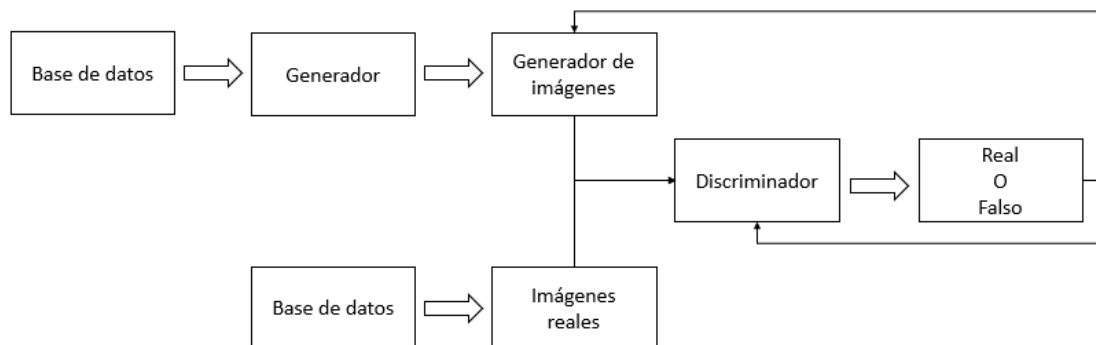
El Discriminador es una red discriminadora que está diseñada para diferenciar cuales son los datos reales y los datos generados por la red generadora. La red discriminadora intenta colocar los datos entrantes en categorías predefinidas, realizando una clasificación binaria de cero y uno, optimizando llegar una meta de 0.5, donde el discriminador no puede distinguir entre imágenes reales y generadas.

El problema Minimax (a veces llamado MinMax) es una teoría que se enfoca en maximizar una función con la mayor pérdida (o viceversa). En el caso de las GAN, esto está representado por los dos modelos que se entrenan de forma antagónica. El paso de entrenamiento se enfocará en minimizar el error en la pérdida de entrenamiento para el generador, mientras se acerca lo más posible a 0.5 en el discriminador donde el discriminador no puede diferenciar entre real y falso.



El discriminador necesita entrenar durante algunas épocas antes de comenzar el entrenamiento contradictorio ya que el discriminador necesitará poder clasificar imágenes. Hay una pieza final en esta estructura, llamada función de pérdida. La función de pérdida proporciona los criterios de parada para los procesos de entrenamiento del Generador y el Discriminador.

A continuación, se puede observar un diagrama como está constituida la red neuronal GAN

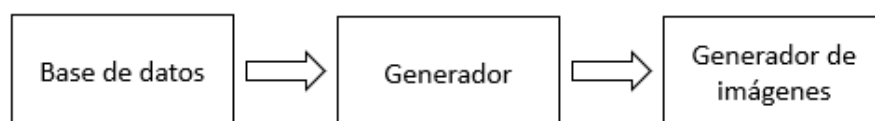


*Figura 1.16: Esquema de una red neuronal antagónica  
(Fuente: Propia)*

Esta arquitectura, se divide cada componente en: generador, discriminador y función de pérdida.

## El Generador

El siguiente diagrama representa las piezas importantes del generador:



*Figura 1.17: Diagrama del generador de una red neuronal antagónica  
(Fuente: Propia)*

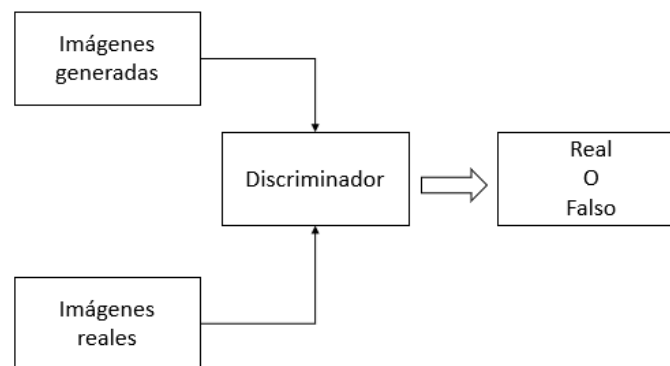
Cada uno de estos bloques de construcción es bastante único, pero podría decirse que el generador es el concepto más importante, por consiguiente, el generador producirá las imágenes o la salida que vemos después de que se complete todo

este proceso de aprendizaje. Como se mencionó anteriormente, el discriminador deberá entrenarse durante algunas épocas antes de comenzar el proceso de entrenamiento en la mayoría de las arquitecturas o nunca completaría el entrenamiento.

## El Discriminador

El discriminador se utiliza para determinar si la salida del generador son imágenes reales o falsas tomando todas las cosas buenas que tenemos con modelos discriminativos y actuar como una función de pérdida adaptativa para la GAN en su conjunto.

En última instancia, el discriminador evaluará la salida de la imagen real y la imagen generada para verificar su autenticidad. Las imágenes reales obtendrán una puntuación alta en la escala inicialmente, mientras que las imágenes generadas obtendrán una puntuación más baja. Eventualmente, el discriminador tendrá problemas para distinguir entre las imágenes generadas y las reales.



*Figura 1.18: Diagrama del discriminador de una red antagónica  
(Fuente: Propia)*

## Función de pérdida

El proceso de entrenamiento ajusta los pesos para optimizar la función de pérdida para el conjunto de problemas dado. La función de pérdida seleccionada para la red neuronal, por lo que, es esencial para garantizar que la red neuronal produzca buenos resultados y converja.

### 1.3.3. Métodos de medición de la calidad de imágenes

La calidad es un parámetro muy importante para determinar cuan óptimo es el tratamiento de una imagen ya que durante este proceso puede provocarse una importante pérdida de información o calidad. Para poder medir esta pérdida existen métodos objetivos y subjetivos, donde los métodos subjetivos están basados en el juicio humano y los resultados varían dependiendo de la percepción individual. Por otro lado, los métodos objetivos se fundamentan en comparaciones que utilizan criterios numéricos teniendo como referencia una imagen original. [37]

Por lo general, para la evaluación de la calidad de la imagen se aplican métricas objetivas como MSE (Mean Square Error) y PSNR (Peak Signal to Noise Ratio) y recientemente también se utiliza una métrica más completa denominada SSIM (Structured Similarity Index Method)

En este trabajo se aplican las métricas PSNR y SSIM para la evaluación de la calidad de imagen ya que la combinación de estas permite saber el error y la similitud estructural de una imagen procesada respecto a una imagen de referencia.

#### **PSNR (Peak Signal to Noise Ratio)**

Es una métrica muy utilizada ya que es simple de calcular, clara en significados físicos, y también conveniente de implementar matemáticamente en el contexto de optimización. [38]

PSNR mide la calidad de la reconstrucción de imágenes con pérdida, esto se logra calculando la relación entre la potencia de señal máxima posible y el poder del ruido distorsionado que afecta a la calidad de la imagen. La señal se considera como los datos originales y el ruido es el error producido por la distorsión en procesamiento. [39], [40]

PSNR se calcula generalmente en la escala de decibelios y se expresa como:

$$\text{PSNR} = \frac{10 \log_{10}(255^2)}{\text{MSE}}$$

Donde:

$$MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2$$

MSE es el error cuadrático medio entre la imagen de referencia  $f$  y la imagen de prueba  $g$ , ambas de tamaño  $M \times N$ .

### **SSIM (Structured Similarity Index Method)**

En este método se compara la similitud entre dos imágenes, una original con una procesada. La comparación se la realiza en base a la degradación de la imagen que se considera como el cambio de percepción en la información estructural, pero también influyen otros factores importantes como el cambio de luminosidad y contraste. [39]

El término información estructural enfatiza sobre los píxeles interdependientes o los píxeles cerrados espacialmente. Estos píxeles fuertemente interdependientes refieren alguna información más importante sobre los objetos visuales en el dominio de la imagen. La luminosidad es un término donde la parte de distorsión de una imagen es menos visible en los bordes de una imagen. Por otro lado, el cambio de contraste es un término donde las distorsiones también son menos visibles en la textura de una imagen. [41]

#### **1.4. Cobertura vegetal mediante el procesamiento de imágenes**

(D. Puga, 2019) en su trabajo “Desarrollo de un código de programación en lenguaje python para el estudio de zonas forestales”, desarrolla una metodología para determinar la cobertura vegetal mediante imágenes aéreas procesadas en base al modelo de color HSV y a la disponibilidad de agua en el suelo. Para un mejor entendimiento a continuación se mencionan los conceptos más relevantes.[42]

##### **1.4.1. Modelo de colores**

En base a estudios previos realizados en entornos naturales se ha demostrado que

el mejor modelo de color es el HSV, utilizado comúnmente en visión artificial e imágenes por segmentación en zonas de interés, además de que este modelo es capaz de enfatizar la percepción del ojo humano, para una mejor comprensión del modelo se describe a continuación el modelo HSV:

HSV o también conocido como HSB (Hue, Saturation, Value/Brightness), es un modelo de color constituido por coordenadas cilíndricas creado por Alvy Ray Smith en 1978. [42]–[45]

### Matiz:

La matiz está representado como un grado de ángulo de 0 a 360° donde se dividen los tres colores principales RGB, dando 120° por cada color, donde: 0=360° es rojo, 120° es verde y 240° es azul, dado esto se puede denotar que: 0 es rojo = RGB (1, 0, 0), 120 es verde = RGB (0, 1, 0), 240 es azul = RGB (0, 0, 1), para tener una mejor percepción se adjunta la siguiente imagen:

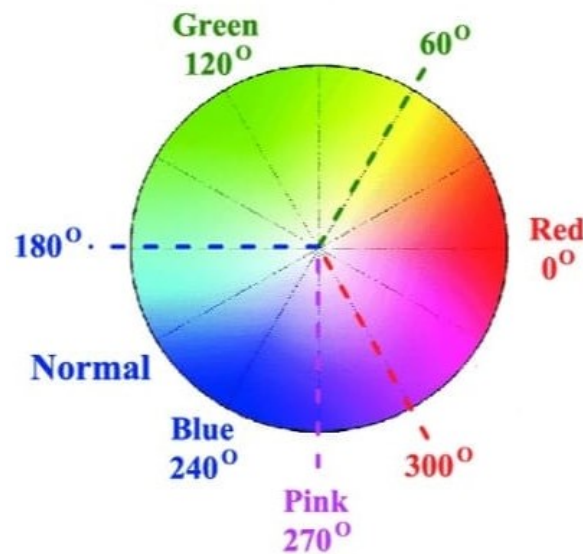


Figura 1.19: Matiz de colores  
(Fuente: [45])

### Saturación

La saturación es la distancia al eje del brillo de 0 al 100%, es decir, cuanto menor es la saturación de un color mayor tonalidad gris será y más decolorada será,

como se muestra en la siguiente figura:

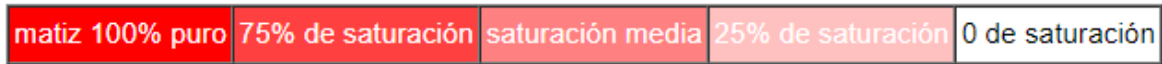


Figura 1.20: Saturación de colores  
(Fuente: [46])

### Valor

Es la distancia del eje blanco-negro del 0 al 100% el color del matiz, donde 0 siempre es negro, dependiendo del grado donde se encuentre el radio del matiz puede ser blanco o un color más o menos saturado.



Figura 1.21: Valor de colores  
(Fuente: [46])

#### 1.4.2. Contenido de agua en humedales altoandinos

Los humedales es un tipo de ecosistema muy común que forma parte de los páramos andinos y son zonas de acumulación de agua. En general los suelos pertenecientes a los páramos poseen una alta capacidad de retención de agua. [47]–[49]

Las condiciones climáticas de los páramos permiten que los humedales contengan grandes depósitos de agua por unidad de volumen, que pueden ir desde 80% o 90% por  $\text{cm}^3$  para el punto de saturación o capacidad de campo y disminuir a 40% o 50% por  $\text{cm}^3$  para el punto de marchitamiento. [47], [50]



*Figura 1.22: Humedal del Antisana  
(Fuente: GRUPO ATA)*

## **2. METODOLOGÍA**

Esta sección presenta la metodología a seguir para la selección del mecanismo más adecuado para la sujeción de una cámara de baja resolución en un UAV de alas fijas. Y también la aplicación de las herramientas para el procesamiento de las imágenes obtenidas; las cuales son, la red neuronal artificial que permite el aumento de la resolución y la metodología para determinar la cantidad de agua en el suelo.

### **2.1. Selección del mecanismo de sujeción**

El objetivo de esta sección es llegar a un equilibrio de requerimientos y necesidades del grupo ATA y los requerimientos propuestos en esta tesis, por consiguiente, se realizará la casa de calidad donde se observará la importancia de cada uno de los ítems calificando del menos importante al más importante, teniendo en cuenta la voz del usuario y la voz del ingeniero.

#### **2.1.1. Casa de la Calidad**

Para la casa de la calidad se realizó una serie de reuniones en las cuales se discutieron las necesidades y su importancia relacionada en los proyectos del grupo ATA, con respecto al diseño del mecanismo. Después de recopilar notas de las reuniones con el grupo se formularon 6 necesidades importantes del mecanismo las cuales se detallan a continuación.

#### **2.1.1.1. Voz del usuario**

- Diseño modular simple.
- Bajo costo de producción.
- Estructura liviana.
- De fácil mantenimiento.
- Múltiples modos de movimiento.
- Fácil compatibilidad con cámaras.

#### **Diseño modular simple**

Un mecanismo simple es una de las principales necesidades destacadas por el cliente, utilizando la menor cantidad posible de tornillos, pernos, placas, etc. Garantizando un bajo costo y que el mantenimiento del mecanismo no sea complejo.

#### **Estructura liviana**

El grupo ATA enfatizo que uno de los principales requerimientos es tener un mecanismo liviano, permitiendo utilizar todo ese peso en otros componentes de la carga útil de los drones o a su vez mejorando la duración del vuelo.

#### **Múltiples modos de movimiento**

Otra solicitud del grupo es que el mecanismo pueda moverse en múltiples modos, ya sea en uno o dos ejes, estos movimientos piden que sean controlados mecánicamente por el piloto o usuario, permitiendo programar el vuelo y saber con qué tipo de posición y ángulo la cámara va a tomar las fotografías.

#### **Compatibilidad con distintas cámaras**

El grupo busca la posibilidad que la unión de la cámara y el mecanismo sea universal anulando obstáculos de poder utilizar cualquier tipo de cámaras ya sean estas de alta o baja calidad. Buscando la necesidad de obtener mínimo dos acoples.



### **Fácil mantenimiento**

El grupo ATA, busca tener un mecanismo que las piezas sean de fácil mantenimiento debido al entorno en el cual debe de trabajar. Cuando sea necesario las piezas sean reemplazables y se puedan obtener fácilmente, por tal motivo busca que la mayoría de las piezas se pueda fabricar en sus impresoras 3D, obteniendo piezas de menor costo y fácil accesibilidad. El grupo lo catalogada como una necesidad de 4 sobre 5 debido a que el mecanismo se colocara en la parte superior y tiene una alta probabilidad que este se impacte con cualquier objeto.

### **Análisis de la competencia**

El análisis de la competencia se realizó lo que es el benchmarking en el apartado 1.2 recabando información sobre los marcas y productos que se ofrecen en el mercado, resaltando las mejores prácticas sobre el área de interés que es posicionadores de cámaras en los UAV. Obteniendo apreciaciones las cuales están tabuladas en las columnas B y C de la casa de la calidad.

#### **2.1.1.2. Voz del ingeniero**

- Peso
- Costo
- Dimensiones
- Ejes de rotación

#### **Peso**

Para que el mecanismo tenga un cambio significativo en el peso de la carga útil este no debe de sobrepasar los 0.5 kilogramos, dando como resultado un mejor tiempo de vuelo.

#### **Costo**

El costo del mecanismo total no debe de sobrepasar los 200 dólares, este requerimiento se debe de trabajar directamente con el tipo de material, elementos de fácil stock o a su vez que se puedan fabricar mediante una impresora 3D al menor costo.

## Dimensiones

El mecanismo debe tener un dimensionamiento que no interfiera en la aerodinámica, desempeño y espacio de los drones en donde se van a testear.

## Ejes de rotación

El mecanismo debe permitir que la cámara se mueva por lo menos en dos ejes de rotación para que esta pueda ser posicionada en un ángulo determinado.

### 2.1.1.3. Resultados de la casa de la calidad

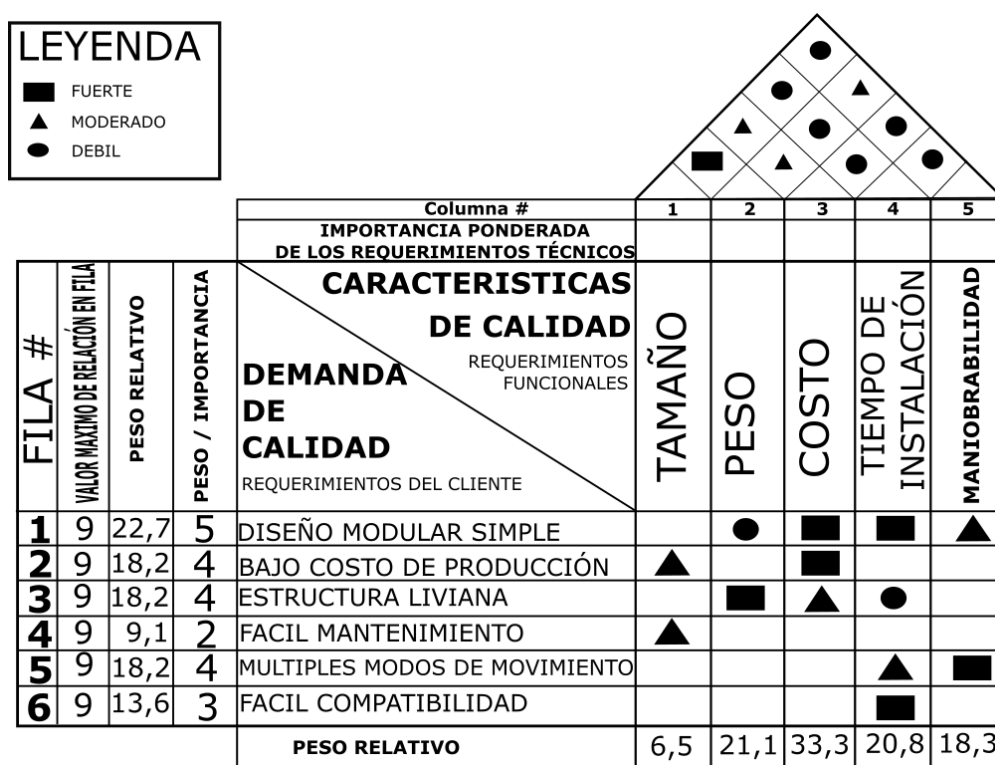


Figura 2.1: Casa de la calidad  
(Fuente: Propia)

Inicialmente se comparó las necesidades descritas por el grupo ATA con los requerimientos de ingeniería en la voz del usuario. En la Figura 2.1 al lado izquierdo se observa las necesidades y la importancia que cada una de ellas tiene dados en un valor de 1 a 5, siendo 1 nada importante y 5 lo más importante. Para las correlaciones de las necesidades y requerimientos se utilizan ponderaciones de: 1 débil, 3 moderada y 9 fuerte,

cabe mencionar que si la celda se queda en blanco no tiene relación alguna. Utilizando dichas ponderaciones se observa una importancia en donde el costo y peso son los requisitos técnicos de mayor requerimiento para el mecanismo, no obstante, las dimensiones y el movimiento de este también tienen un grado de protagonismo relevante.

En la siguiente tabla se resumen los resultados de la casa de la calidad con las especificaciones técnicas necesarias.

*Tabla 2.1: Resultados de la casa de la calidad*

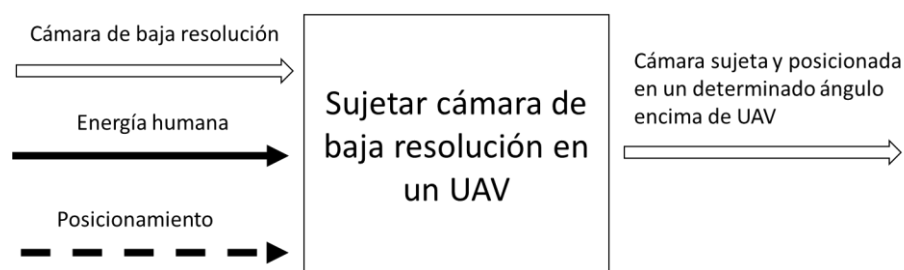
REQUERIMIENTO	UNIDAD	OBJETIVO	TOLERANCIA
Peso	kg	0,4	±0,1
Costo	\$	200	+50
Dimensiones	mm	70x8x70	N/A
Ejes de rotación	#	2	+1

*(Fuente: Propia)*

## 2.1.2. Análisis Funcional

En esta sección se determinará las funciones básicas del sistema de mecanismo de sujeción de la cámara con el drone para conseguir un mejor desarrollo de la estructura modular. Este análisis está orientado a identificar tanto la forma global como las subfunciones del mecanismo, representando la tarea integral que se debe realizar para obtener el producto a diseñar.

### 2.1.2.1. Diagramas Funcionales



*Figura 2.2: Diagrama funcional nivel 0 del mecanismo de sujeción  
(Fuente: Propia)*

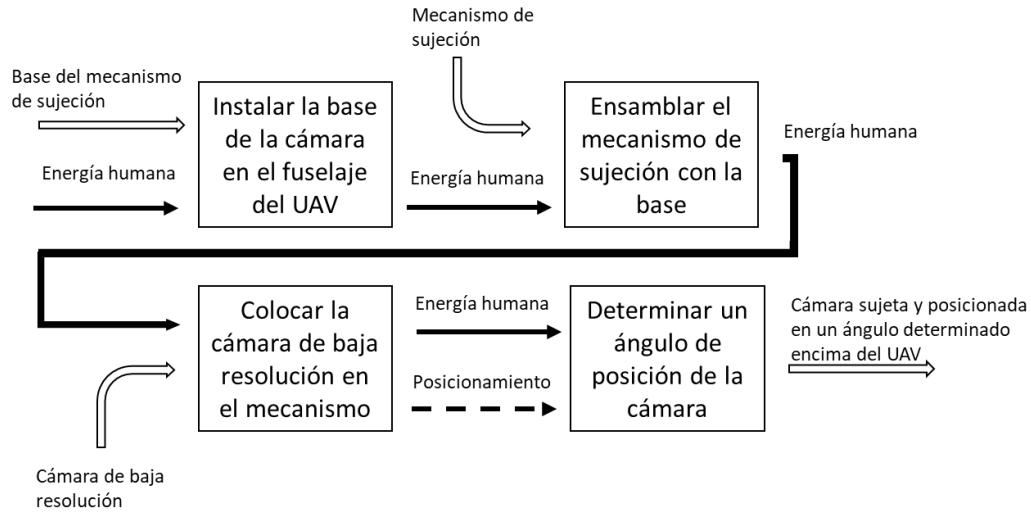


Figura 2.3: Diagrama funcional nivel 1 del mecanismo de sujeción  
(Fuente: Propia)

### 2.1.2.2. Análisis de diagramas funcionales

El mecanismo para implementar se desarrolla hasta un diagrama funcional nivel 1 debido a que este cumple una función específica, de esta manera se logra tener una visión más amplia para establecer la solución.

En el nivel 0 se representa la función global del mecanismo, que en el caso de este trabajo es la de posicionar una cámara de baja resolución encima de un UAV. Para lograr cumplir este objetivo es necesario el uso de energía emitida en su totalidad los usuarios.

En el caso del nivel 1, el proceso para posicionar la cámara en un determinado ángulo se expande un poco más, permitiendo observar un flujo más específico para el cumplimiento de la función global requerida por los usuarios. Destacando en este nivel, la primera acción de instalar la base de la cámara en el fuselaje del UAV permitirá realizar el acople entre el mecanismo de sujeción con el UAV.

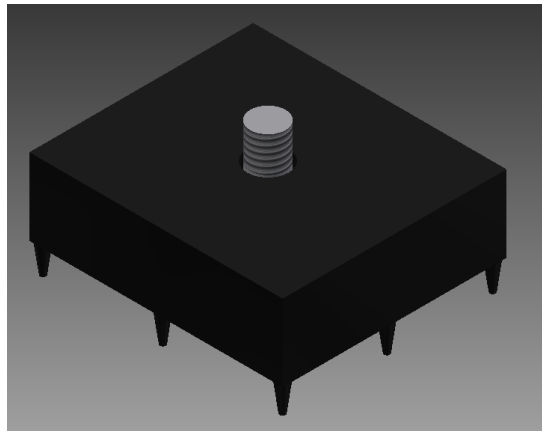
### 2.1.3. Subsistemas Funcionales

En base al análisis funcional definido en el punto anterior, se determina que el mecanismo de sujeción de la cámara está conformado por dos subsistemas funcionales. El primero

es la base del mecanismo que permitirá el acople con el UAV. Para el segundo subsistema se plantean dos opciones de mecanismos que permiten el posicionamiento de la cámara a través del movimiento en los tres ejes de rotación.

#### **2.1.3.1. Subsistema 1: Base del mecanismo**

La base del mecanismo es el primer subsistema y tiene la función importante de permitir el acople entre el mecanismo de sujeción de la cámara con la parte superior del UAV.

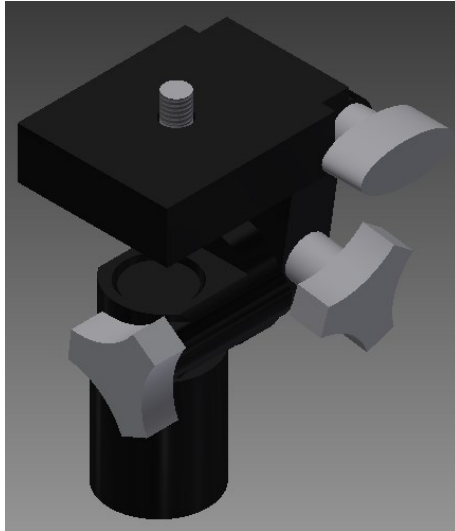


*Figura 2.4: Subsistema 1: Base del mecanismo  
(Fuente: Propia)*

La Figura 2.4 representa el diseño de la base del mecanismo, que consta de un tornillo en el cual irá acoplado el resto del mecanismo de sujeción.

#### **2.1.3.2. Subsistema 2: Mecanismo de sujeción 1**

La función del subsistema 2 es permitir posicionar la cámara en un ángulo determinado. A continuación, se presenta el diseño de la primera opción para cumplir con el objetivo de sujetar la cámara.



*Figura 2.5: Subsistema 2: Mecanismo de sujeción 1  
(Fuente: Propia)*

### **Ventajas**

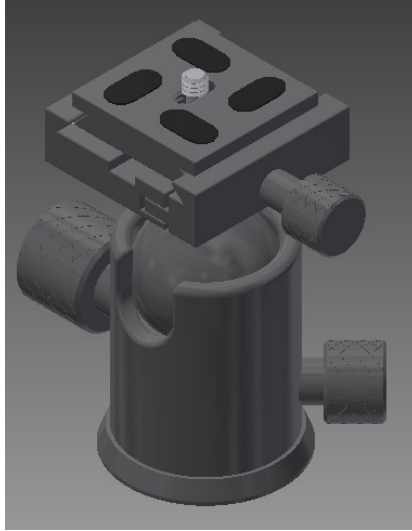
- Buen rango de giro en los tres ejes de rotación.
- Compatible con repuestos accesibles de trípodes básicos para cámaras

### **Desventajas**

- Posicionamiento relativamente demoroso debido a que cada eje debe ser bloqueado para mantenerse en cierto ángulo.
- Considerable número de piezas.

#### **2.1.3.3. Subsistema 2: Mecanismo de sujeción 2**

Como ya se había mencionado anteriormente, existe una segunda opción para cumplir la función del subsistema 2, su diseño se lo representa a continuación.



*Figura 2.6: Subsistema 2: Mecanismo de sujeción 2  
(Fuente: Propia)*

### **Ventajas**

- Rápido posicionamiento del ángulo deseado.
- Compacto para fácil almacenamiento y traslado.
- Facilidad de montaje

### **Desventajas**

- Rango de giro limitado en dos de sus ejes de rotación.

#### **2.1.4. Selección del diseño del mecanismo de sujeción**

Para la selección del mecanismo de sujeción de la cámara entre las propuestas planteadas anteriormente se emplea el método ordinal de criterios ponderados. Los criterios de valoración seleccionados son:

- Peso, para disminuir el payload del UAV
- Dimensiones, que no interfiera en el desempeño del UAV
- Ejes de rotación, que permita un rápido posicionamiento

Tabla 2.2: Valores designados para los criterios

Respuesta	Ponderación
Importancia de fila mayor que columna	1
Importancia de fila igual que columna	0.5
Importancia de fila menor que columna	0

(Fuente: Propia)

Tabla 2.3: Evaluación del peso específico de cada criterio

Peso > Dimensiones > Ejes de rotación					
Criterio	Peso	Dimensiones	Ejes de rotación	$\Sigma + 1$	Ponderación
Peso		0,5	1	2,5	0,417
Dimensiones	0,5		1	2,5	0,417
Ejes de rotación	0	0		1	0,167
Suma				6	1

(Fuente: Propia)

En base al método descrito anteriormente, se procede a realizar la evaluación de las propuestas para mecanismo de sujeción tomando como referencia los criterios más preponderantes obtenidos de la casa de la calidad.

A continuación, se evalúa las dos propuestas presentadas en base a cada criterio de referencia.

Tabla 2.4: Evaluación del peso específico del criterio de peso

Peso	Propuesta 1	Propuesta 2	$\Sigma + 1$	Ponderación
Propuesta 1		0	1	0,333
Propuesta 2	1		2	0,667
Suma			3	1

(Fuente: Propia)

Tabla 2.5: Evaluación del peso específico del criterio de dimensiones

Dimensiones	Propuesta 1	Propuesta 2	$\Sigma + 1$	Ponderación
Propuesta 1		0	1	0,333
Propuesta 2	1		2	0,667
Suma			3	1

(Fuente: Propia)



Tabla 2.6: Evaluación del peso específico del criterio de ejes de rotación

Ejes de rotación	Propuesta 1	Propuesta 2	$\Sigma + 1$	Ponderación
Propuesta 1		0,5	1,5	0,500
Propuesta 2	0,5		1,5	0,500
		Suma	3	1

(Fuente: Propia)

Después de evaluar cada característica con las propuestas, se realiza la ponderación final para saber el orden de prioridad, la cual se puede apreciar en la siguiente tabla.

Tabla 2.7: Ponderación final

Conclusiones	Peso	Dimensiones	Ejes de rotación	$\Sigma + 1$	Ponderación
Propuesta 1	0,333*0,417	0,333*0,417	0,5*0,167	1,361	0,454
Propuesta 2	0,667*0,417	0,667*0,417	0,5*0,167	1,639	0,546
			Suma	3	1

(Fuente: Propia)

La propuesta 2 es la mejor solución ya que cumple con los requerimientos propuestos de la casa de la calidad.

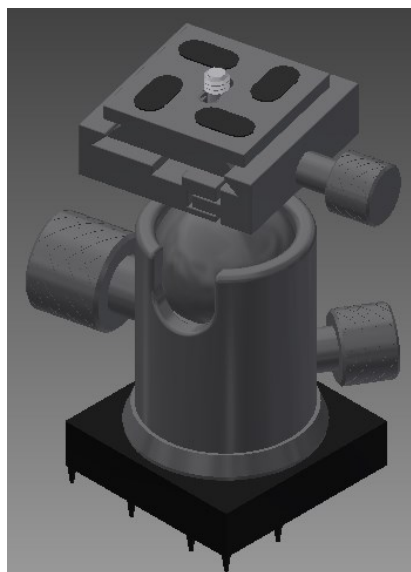


Figura 2.7: Subsistema 2: Mecanismo final para la sujeción de la cámara  
(Fuente: Propia)

## 2.2. Selección de la red neuronal

Para el mejoramiento de la calidad de imagen se adaptó códigos basados en el lenguaje Python, este IDLE es un lenguaje abierto que mediante la utilización de librerías como Pytorch, Matplotlib permiten construir, modificar y codificar redes neuronales, convirtiéndose en un lenguaje de alto nivel en la utilización de la inteligencia artificial utilizada en la actualidad en varias aplicaciones como son: reconstrucción de imágenes, generación de sonidos, libros, mantenimientos predictivos, reconocimientos de objetos, etc. Por este motivo se detallará a continuación las metodologías más utilizadas en el mejoramiento de la calidad de imagen tales como: Interpolación bicúbica, redes neuronales convolucionales y redes neuronales antagónicas.

### Interpolación Bicúbica

La interpolación bicúbica como se explicó en el apartado 1.2.1, este algoritmo se utiliza en aplicaciones de imágenes con calidad relativamente clara, Sin embargo, también se puede observar algunos inconvenientes significativos como al momento de realizar expansión de una imagen de baja resolución pierde información útil como se puede observar en la siguiente imagen:



*Figura 2.8: Comparación del método de interpolación bicubica. (izq) Imagen de baja resolución, (der) imagen interpolada  
(Fuente: Propia)*

Al momento de ejecutar el código con imágenes de baja resolución en el algoritmo tiende a utilizar demasiada memoria y un alto costo computacional.

### CNN

Las redes neuronales convolucionales son utilizadas en distintas aplicaciones como son:

la creación de textos, audios y principalmente clasificación y reconocimiento de objetos, no obstante se ha realizado varios estudios para el mejoramiento de resolución de imágenes obteniendo imágenes generadas mediante la red de entrenamiento pero no reales debido a que este tipo de red neuronal no tiene un discriminador que permita saber si los filtros y texturizaciones que se aplican en algoritmos siguen la misma tendencia de los píxeles perdidos, como se puede observar en la siguiente imagen existe una mejor apreciación y suavidad en la texturización de curvas pero a su vez pierde información necesaria en aplicaciones de monitoreo de humedales.



*Figura 2.9: Comparación del método de CNN. (izq) Imagen de baja resolución, (der) imagen procesada  
(Fuente: Propia)*

La imagen anterior es obtenida configurando la red con un batch size: 32 y un número de épocas: 300 así mismo utilizando una red pre entrenada con una base de datos de entornos naturales, obteniendo resultados en las siguientes métricas: PSNR: 20.33 y SSIM: 0.5, como se puede apreciar en estas dos métricas no se obtiene resultados cuantitativos esperados y esto es fácilmente apreciable a simple vista ya que la imagen de la derecha no tiene las mismas características de la original.

## **GAN**

Las redes neuronales antagónicas tienen una ventaja con respecto a otras por la adición de su discriminador con el objetivo de reconocer cuando una imagen es verdadera o falsa, en consecuencia, esta red es especialista en reconstrucción, generación y mejoramiento de la calidad de imágenes de modo que la imagen generada no pierde información útil.



*Figura 2.10: Comparación del método de GAN. (izq) Imagen de baja resolución, (der) imagen procesada  
(Fuente: Propia)*

Para el procesamiento de la imagen se realizó una configuración similar a la red CNN dando un PSNR: 24.98 y un SSIM: 0.693 que al comparar con los resultados de la red neuronal CNN se puede incrementar un incremento en las dos métricas, así mismo de tener una mejora visual, para una mejora apreciación de los resultados cualitativos se puede revisar en el Anexo I.

En base a los casos planteados anteriormente se selecciona la GAN como método para el procesamiento de la imagen en el aumento de la resolución.

### **2.3. Adaptación de la GAN**

Este trabajo se basa en las redes neuronales pre entrenadas desarrolladas en código abierto y escritas en Python de Left Thomas y de Xintao Wang donde se realizó cambios para que la base de datos estén relacionadas con entornos de ecosistemas.[51], [52] El código final para la etapa de entrenamiento se adjunta en el Anexo II, así como también el código implementado para el testeo de la nueva GAN entrenada en el Anexo III.

La adaptación de la GAN y de todos los códigos utilizados se los realiza en el entorno Google Colab Pro, que permite escribir y ejecutar códigos en Python, siendo óptimo para procesos de aprendizaje automático además de proporcionar recursos computacionales como GPU y RAM de gran capacidad que pueden ser usados de forma remota y menorando el tiempo de procesamiento. De forma similar los códigos están subidos al repositorio de

GitHub: <https://github.com/ATA-EPN/GANS-FOR-ECOSYSTEM> de forma de dominio público. La base de datos del entrenamiento de la red neuronal se encuentra bajo el dominio del grupo ATA - FIM.

```
Sun Oct 10 19:26:50 2021
```

NVIDIA-SMI 470.74				Driver Version: 460.32.03		CUDA Version: 11.2	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC	
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	MIG M.
0	Tesla P100-PCIE...	Off	00000000:00:04.0	Off		0	
N/A	42C	P0	27W / 250W	0MiB / 16280MiB	0%	Default	N/A

Figura 2.11: Características de la GPU proporcionada por Google Colab Pro.  
(Fuente: Propia)

### 2.3.1. Adquisición y preparación de base de datos

Para la adquisición de imágenes de entornos naturales y la preparación de la base de datos se sigue el flujo de trabajo descrito en figura que se muestra a continuación.

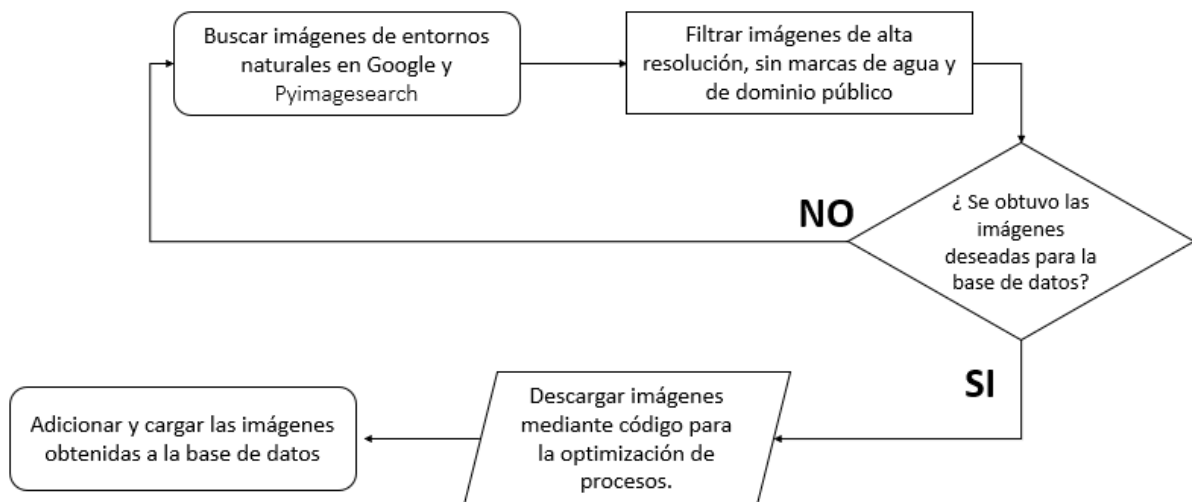


Figura 2.12: Diagrama de flujo para la adquisición y preparación de la base de datos.  
(Fuente: Propia)

Según el estudio realizado por Takano y Alaghband, las GANs presentan mejores resultados cuando se las utiliza para el procesamiento de imágenes en relación con las imágenes que han sido entrenada. Por ejemplo, si se va a utilizar la red para procesar imágenes de rostros de personas lo mejor es entrenar la red solo con imágenes de rostros de personas. Esto no quiere decir que la GAN no pueda procesar otro tipo de imágenes solo se menciona que los resultados pueden diferir cuando el entrenamiento se realiza con una base de datos con imágenes variadas.[53]

En este caso, lo mejor es entrenar la nueva GAN con imágenes de entornos naturales como bosques, áreas de extendida vegetación, páramos entre otros y que, en relación con lo mencionado anteriormente, los resultados del procesamiento de imágenes de entornos naturales serán los más aptos.

La base de datos está conformada mínimo por 900 imágenes, 800 para el entrenamiento y 100 para la validación y todas deben de poseer una resolución óptima ya que la GAN aprenderá a generar imágenes acordes a la calidad de imágenes con la que ha sido entrenada.

La recopilación de imágenes de entornos naturales se realizó mediante el motor de búsqueda de Google y para la descarga de éstas se aplicó los códigos del Anexo IV y Anexo V proporcionados por Pyimagesearch y GitHub.[54], [55]

Para la obtención de la base de datos se busca en Google imágenes relacionadas con entornos naturales, filtrando dos características primordiales, que sea de alta resolución e imágenes de dominio público, posterior a esto se aplica la segunda parte del código del Anexo V para proceder a descargar el compilado de imágenes.

Cabe mencionar que esta base de datos también se incluyen imágenes del Área de Conservación Hídrica Antisana proporcionadas por el grupo ATA, ya que uno de los propósitos de este trabajo es poder aplicar la metodología de aumento de resolución de imágenes para poder tecnificar el proceso de monitoreo ambiental de los humedales altoandinos.



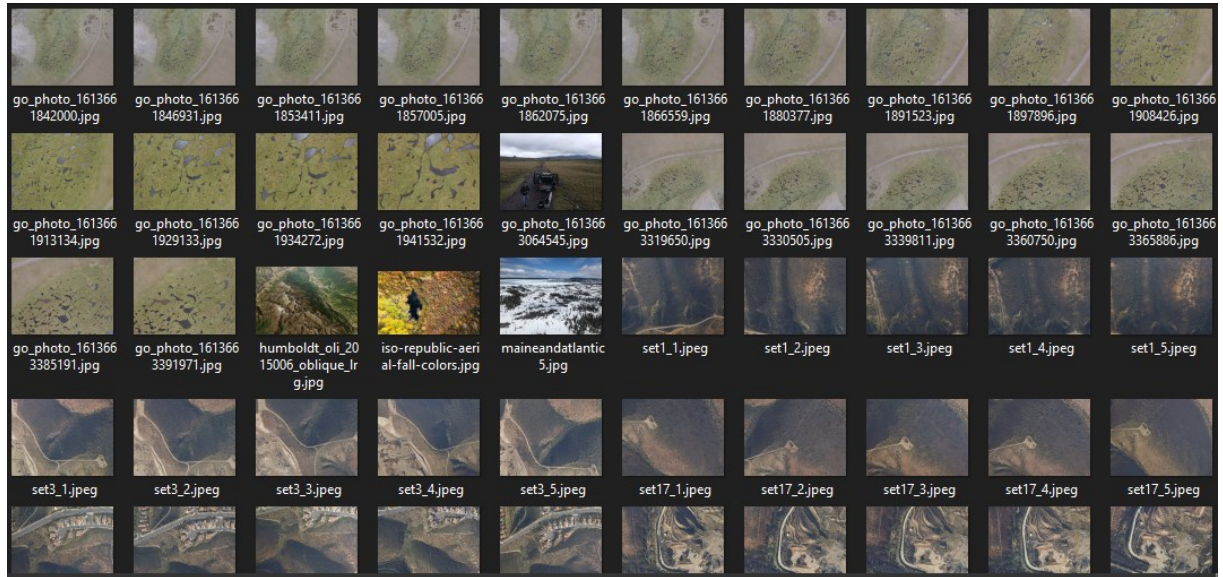


Figura 2.13: Muestra de la base de datos para el entrenamiento de la GAN  
(Fuente: Propia)

### 2.3.2. Adaptación de la GAN para el entrenamiento

Para la etapa de entrenamiento de la GAN se sigue el flujo de trabajo descrito en figura que se muestra a continuación.

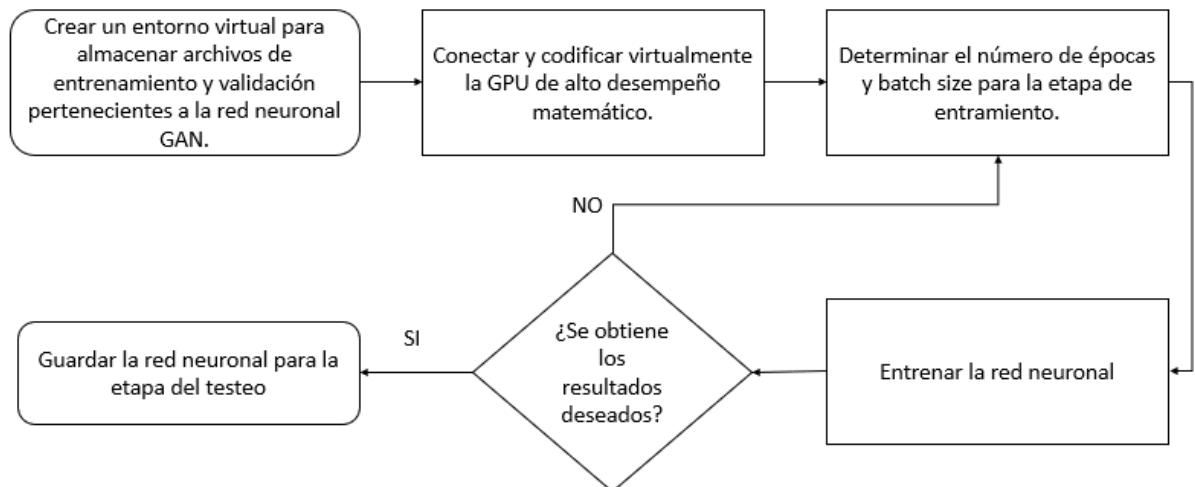


Figura 2.14: Diagrama de flujo para la adaptación de la GAN para el entrenamiento.  
(Fuente: Propia)

El código para el entrenamiento de la GAN basado en el trabajo de Left Thomas [51], es modificado para poder ejecutarse en el entorno de Google Colab Pro, para poder ejecutar

el entrenamiento se utiliza la base de datos preparada en la sección 2.3.1 y se determina los parámetros de mayor relevancia, que son el número de épocas y el número de batch size. Los mejores resultados tanto para el PSNR y el SSIM de este trabajo se obtuvieron utilizando 300 épocas para el entrenamiento, 64 para el número de batch size. El código completo del entrenamiento se encuentra en el Anexo II

En esta etapa la red neuronal aprende de cada una de las imágenes de la base de datos para generar nuevas imágenes las cuales serán clasificadas y comparadas con las imágenes destinadas para la validación hasta que no se puedan diferenciar o se asemejen a las imágenes reales.

### 2.3.3. Adaptación de la GAN para el testeo

Para comprobar que con la GAN entrenada en la sección anterior se obtienen los resultados esperados, se la pone a prueba con imágenes del dominio público y siguiendo el flujo de trabajo descrito en figura que se muestra a continuación.

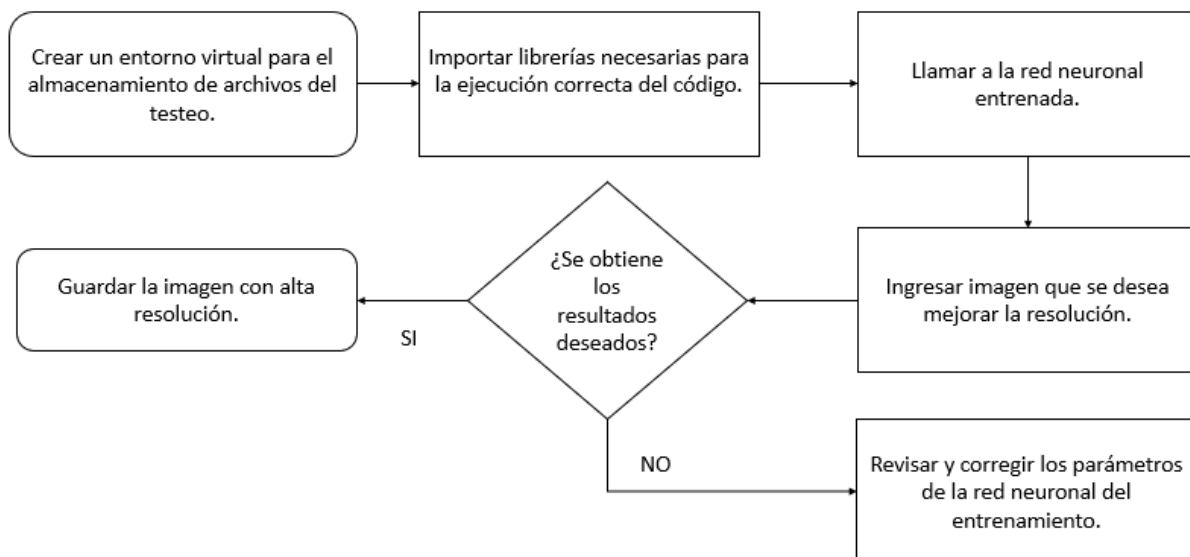


Figura 2.15: Diagrama de flujo para la adaptación de la GAN para el testeo  
(Fuente: Propia)



El código para el testeo de la GAN entrenada se basaba en el trabajo de Xintao Wang [52] y al igual que hizo para la etapa de entrenamiento, este código es modificado para poder ejecutarse en el entorno de Google Colab Pro. El código completo del testeo se encuentra en el Anexo III

Para el testeo se aplican imágenes de entornos naturales con baja resolución y se compara visualmente si existe una mejora de la imagen generada por la red con respecto a la imagen original.

## 2.4. Aplicación de la metodología de estudio de propiedades físico - térmicas

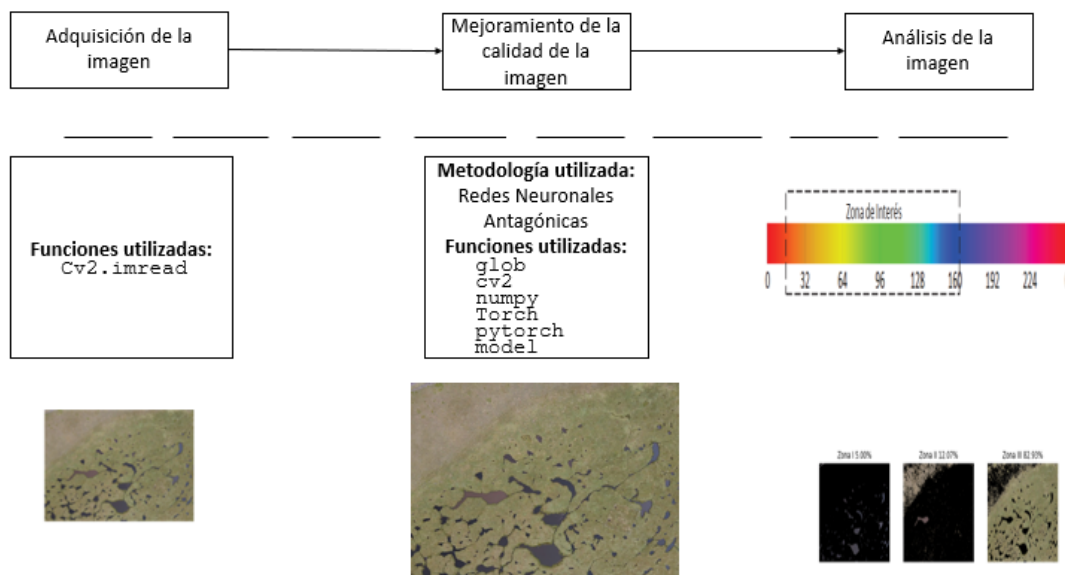
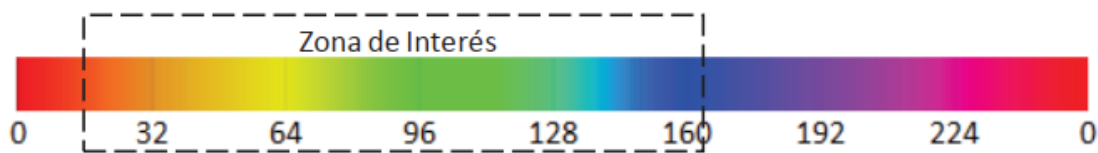


Figura 2.16: Diagrama de flujo para la aplicación de la metodología de estudio de propiedades físico – térmicas (Fuente: Propia)

(D. Puga, 2019) en su trabajo “Desarrollo de un código de programación en lenguaje Python para el estudio de zonas forestales” desarrolla y valida una metodología para el procesamiento de imágenes. En este estudio de propiedades físico-térmicas de zonas forestales en base a la humedad del suelo se menciona, al igual que en el apartado 1.4.2, que el modelo de color HSV presenta mejores resultados para la segmentación de imágenes en la zona de interés.



*Figura 2.17: Rango de tonalidad de la zona de interés para zonas forestales  
(Fuente: [42])*

En la Figura 2.17 se puede apreciar que el rango de colores dentro de la zona de interés abarca la gran mayoría de tonalidades que pueden estar presentes en los ambientes naturales.

En base a estos estudios, la presente tesis implementa la metodología para el procesamiento de imágenes a color en base al modelo HSV. Por esta razón existe la necesidad de adaptar el código de programación de esta metodología para que pueda responder de una mejor manera al procesamiento de imágenes de entornos naturales previamente sometidas a un aumento de resolución por parte de la GAN desarrollada anteriormente.

Como ya se ha venido desarrollando anteriormente con la etapa de entrenamiento y testeo, el código para el estudio de propiedades físico – térmicas también es modificado para poder ser ejecutado en el entorno de Google Colab Pro. El código modificado se encuentra en el Anexo VI

La segmentación de la zona de interés se realizó en tres zonas, zona de no vegetación, zona de vegetación húmeda y zona de vegetación muy húmeda. La segmentación es dada mediante máscaras que permiten identificar cada una de las zonas.

La modificación más relevante en este código es que en nuestro caso se añadió una sub-máscara a la zona de vegetación húmeda debido a que en un principio existía pérdida de información de esta debido a las tonalidades de color de los humedales.

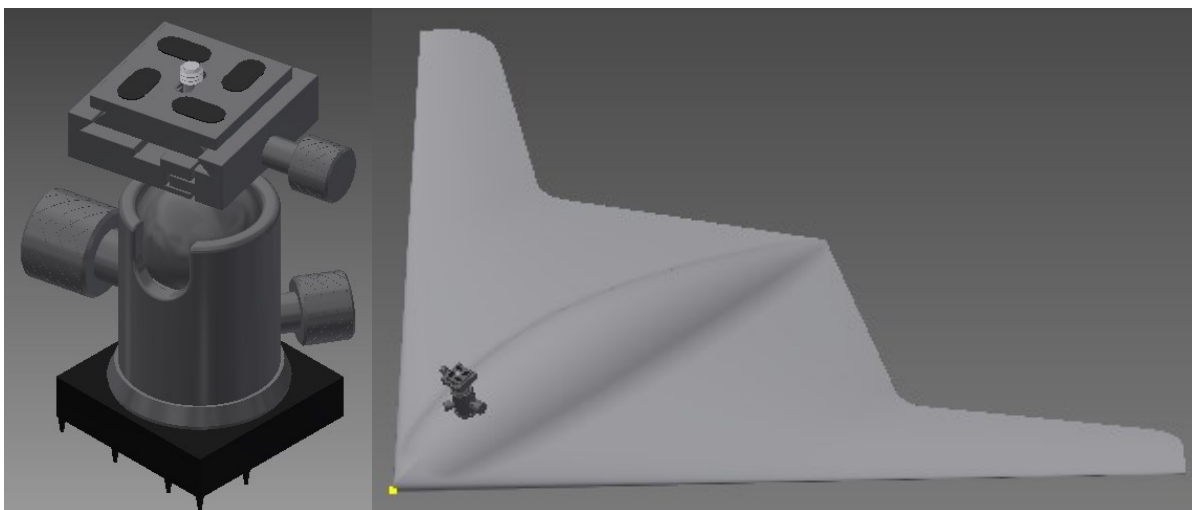
### 3. RESULTADOS Y DISCUSIÓN

#### 3.1. Mecanismo de sujeción de la cámara

En la Tabla 2.1 se presenta de manera resumida el resultado de los requerimientos y necesidades de la casa de calidad. Culminado dicho proceso, se observa que los requerimientos más importantes se basan directamente en el peso, costo y dimensiones del mecanismo de sujeción.

Posteriormente se realizó el análisis funcional obteniendo solo dos niveles: 0 y 1, representando la función global del mecanismo y el proceso de posición de la cámara en dron respectivamente, cabe mencionar que la energía utilizada en estos dos niveles es de energía mecánica (humana), obteniendo un ahorro económico, no obstante, el tiempo de posicionamiento es mayor con respecto a un mecanismo totalmente automatizado.

Finalmente, los subsistemas del mecanismo se divide en dos partes primordiales que son la base del sistema que conecta directamente con el dron y el cuerpo del mecanismo que conecta con la cámara de baja resolución, dando como ganador a la propuesta dos debido a que cumple con todos los requisitos propuestos en la casa de la calidad obteniendo una mayor ponderación debido a la facilidad de montaje, fácil almacenamiento y rápido posicionamiento.



*Figura 3.1: Representación del mecanismo final de sujeción de la cámara  
(Fuente: Propia)*

### 3.2. Aumento de resolución de imágenes

En este trabajo se realizaron cinco entrenamientos de la GAN variando los dos parámetros que más relevancia e influencia tienen en los resultados finales. Estos parámetros son el número de épocas de entrenamiento y el número de batch size.



Figura 3.2: Comparación de las cinco GANs entrenadas  
(Fuente: Propia)

En la Figura 3.2 se realiza una comparación de las cinco GANs entrenadas donde se varía parámetros relevantes como son: el número de épocas entre 100, 200 y 300; y el número de batch size entre 32, 64 y 128. Y como se puede apreciar los mejores resultados se obtienen a partir de las dos últimas GANs entrenadas.

Tabla 3.1: Comparación de los resultados de las cinco GANs entrenadas

GAN ENTRENADA	ÉPOCAS	BATCH SIZE	PSNR	SSIM
HR_e100_bz64	100	64	24.93	0.692
HR_e200_bz32	200	32	25.30	0.71
HR_e200_bz64	200	64	25.03	0.703
HR_e200_bz128	200	128	24.98	0.693
HR_e300_bz64	300	64	25.35	0.71

(Fuente: Propia)



Figura 3.3: (Izq) Comparación visual de la cuarta GAN entrenada. (Der) Comparación visual de la quinta GAN (Fuente: Propia)

Como se logra observar en la Figura 3.3 los mejores resultados del procesamiento de las imágenes se logran con un entrenamiento con 200 épocas, con un número de batch size de 128 o con un entrenamiento con 300 épocas, con un número de batch size de 64. Dando como resultado un PSNR de 24.98 y un SSIM de 0.693 en el primer caso y un PSNR de 25.35 y un SSIM de 0.71 en el segundo caso.

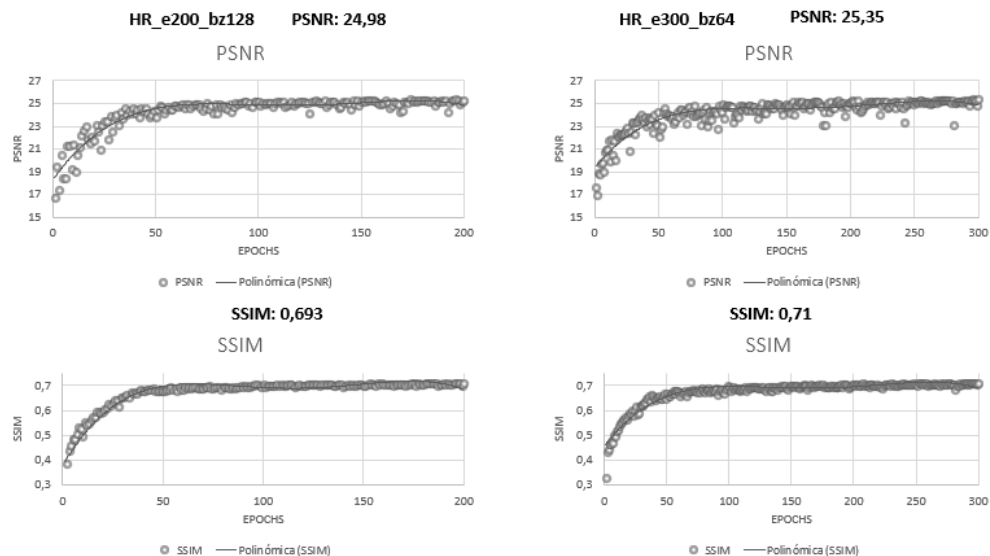


Figura 3.4: Comparación de los resultados de las métricas de la cuarta y quinta GAN entrenada (Fuente: Propia)

Ambos resultados son buenos no solo porque visualmente se observa una mejora en la resolución de las imágenes, sino que también las métricas para determinar la calidad de las imágenes tienen valores cercanos unos con otros según corresponda y por lo que se puede concluir que cualquiera de las dos últimas GANs entrenadas puede ser utilizada para el procesamiento de imágenes de entornos naturales.

En la Figura 3.4 se puede apreciar que los resultados de las métricas a partir de las 300 épocas de entrenamiento, posee un comportamiento muy cercano a la de una constante por lo que al entrenar una nueva GAN con más de 300 épocas se obtendría mejoras de muy poca relevancia.

### 3.3. Aplicación de la metodología de procesamiento de imágenes

En esta sección se aplica la metodología de procesamiento de imágenes a color para determinar las propiedades físico-térmicas expuesta en la sección 2.4 a las imágenes aéreas capturadas de los humedales del Área de Conservación Hídrica Antisana. En este tipo de ambientes la cantidad de agua en el suelo es alta incluso llegando a tener charcos llenos en su totalidad, considerando estos factores la segmentación de las tres zonas son:

Zona de no vegetación: se considera tanto el suelo seco sin ningún rastro de vida como también todas las zonas que contengan agua en su totalidad.

Zona de vegetación húmeda: en esta zona se involucra las partes con tonos más claros.

Zona de vegetación muy húmeda: para esta zona se considera los espacios más oscuros.

*Tabla 3.2: Rangos de segmentación*

ZONA	MÁSCARA	RANGO HVS DE SEGMENTACIÓN	
ZONA I: No vegetación	mask1	Límite Inferior	80, 0, 84
		Límite Superior	170, 255, 225
	mask1_1	Límite Inferior	20, 0, 50
		Límite Superior	162, 58, 92
ZONA II: Húmeda	mask2	Límite Inferior	20, 0, 50
		Límite Superior	175, 255, 255
	mask2_2	Límite Inferior	0, 0, 50
		Límite Superior	255, 255, 255
ZONA III: Muy húmeda	mask3	Límite Inferior	20, 0, 50
		Límite Superior	45, 255, 255

(Fuente: Propia)

Los rangos de color que se indican en la Tabla 3.2 para la segmentación de las imágenes, se aplican en dos casos de estudio en el área de humedales del Antisana.

En el primer caso se aplica la segmentación de color a una imagen de alta resolución captada a una baja altura.

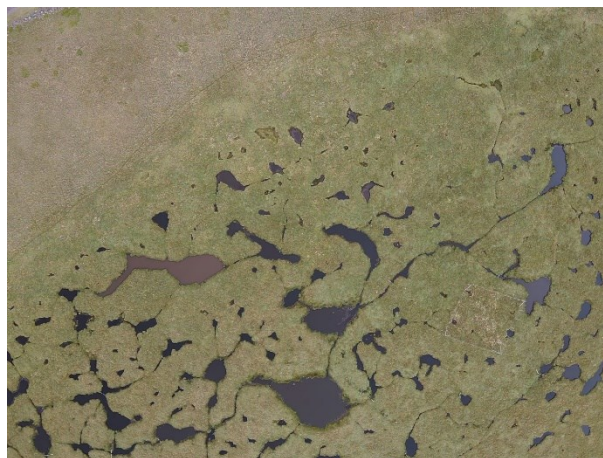
Para el segundo caso se utiliza una imagen capturada a una altura mayor y se recorta un área similar al espacio visual de la imagen del caso 1. A esta imagen se le aplica un aumento de resolución mediante la GAN definida en la sección 3.2. Posteriormente se aplica la segmentación de color al igual que en el primer caso y finalmente se comparan los resultados de ambas imágenes.

## CASO 1

*Tabla 3.3: Propiedades de la imagen para el caso 1*

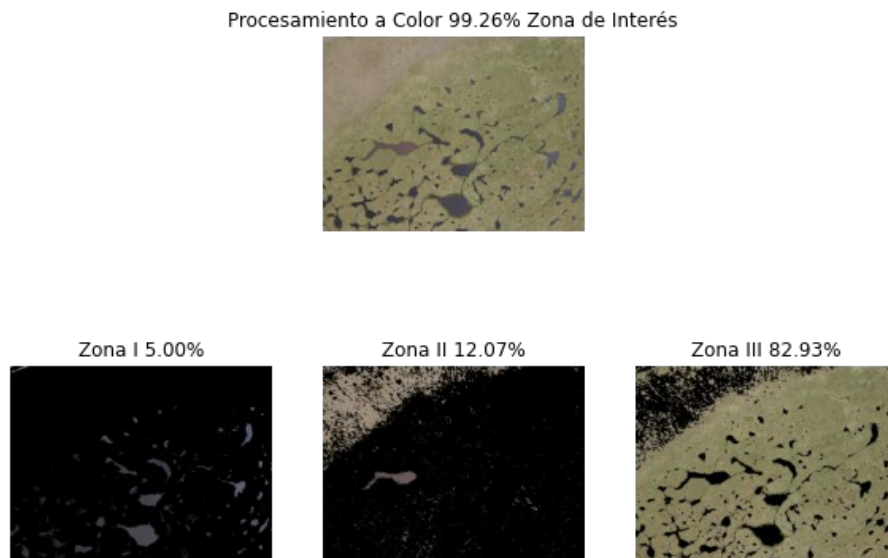
PROPIEDADES	VALOR
Ancho	1440 píxeles
Alto	1080 píxeles
Tamaño	1.29 MB
Tipo	jpg

*(Fuente: Propia)*



*Figura 3.5: Imagen Antisana\_Caso 1.  
(Fuente: Propia)*





*Figura 3.6: Resultados de la aplicación de la metodología de estudio de propiedades físico-térmicas. Imagen Antisana\_Caso 1 (Fuente: Propia)*

En la Figura 3.6 se puede apreciar a primera vista que los resultados de las tres zonas tienen concordancia en relación con las zonas de segmentación definidas anteriormente. En la Zona I se muestra las áreas en donde existe agua en su totalidad, esto es completamente válido ya que en estas partes no existe vegetación. En la Zona II se evidencia la mayor cantidad de datos en la parte izquierda superior esto debido a que en esta zona se encuentran las zonas húmedas. La segmentación de la Zona III, zona de alta humedad, tiene sentido ya que existen muchos charcos de agua en esta zona por lo que la humedad en este espacio de suelo será mayor que las demás zonas. Existe ciertos datos de alta humedad en la parte superior izquierda, alrededor de la Zona II, estos datos no son erróneos ya que se ha logrado observar en este tipo de ambiente que pequeños espacios de alta humedad pueden estar rodeados por zonas de baja humedad o de no vegetación.



CASO 2

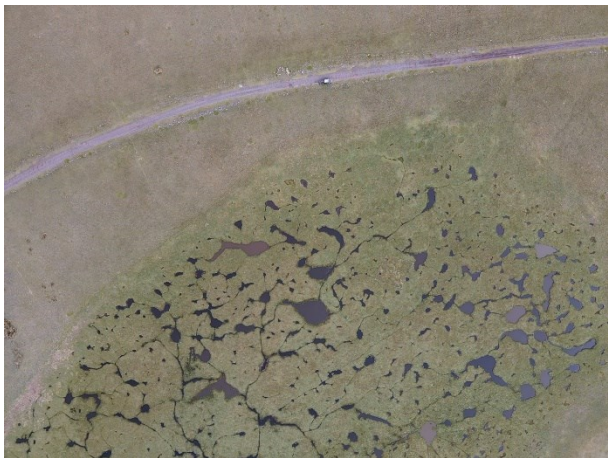


Figura 3.7: Imagen Antisana\_Caso 2 - Original  
(Fuente: Propia)

Tabla 3.4: Propiedades de la imagen para el caso 2

PROPIEDADES	VALOR
Ancho	729 píxeles
Alto	566 píxeles
Tamaño	180 KB
Tipo	jpg

(Fuente: Propia)

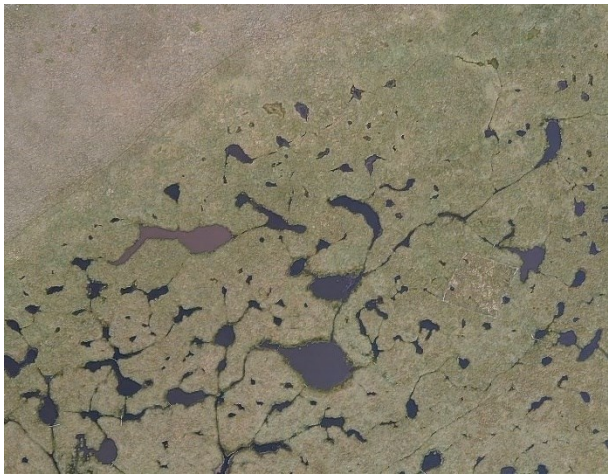
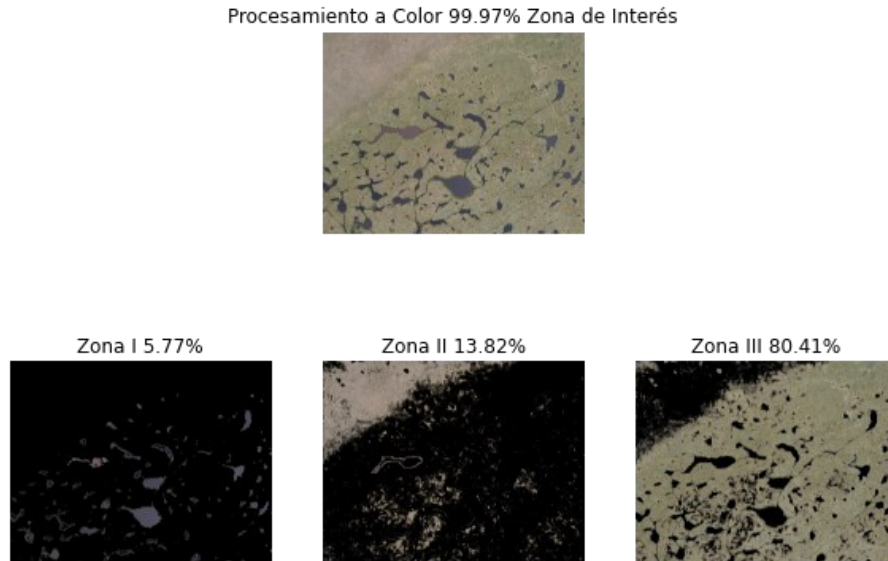


Figura 3.8: Imagen Antisana\_Caso 2  
(Fuente: Propia)

La Figura 3.8 representa el área correspondiente al caso 1 de estudio, extraída de la

Figura 3.7. Esta imagen ha sido procesada para el aumento de resolución mediante la GAN definida anteriormente.



*Figura 3.9: Resultados de la aplicación de la metodología de estudio de propiedades físico-térmicas. Imagen Antisana\_Caso 2. (Fuente: Propia)*

En la Figura 3.9 se puede observar que en la Zona I las áreas ocupadas por los charcos de agua. En la Zona II se evidencia ciertos datos dispersos esto debido a que es muy común en este tipo de ambientes que las zonas de baja humedad se mezclen con las zonas de alta humedad y debido a esto es posible que exista una pérdida de datos; pero con relación al resto de la imagen la zona más clara es identificada fácilmente y es en esta zona donde existe humedad. En la Zona III, se encuentran las áreas de alta humedad en gran porcentaje de la imagen, casi similar a la imagen del caso 1.

### **3.4. Comparación de la metodología de procesamiento de imágenes con imágenes de la zona de estudio.**

Para la comparación de la metodología de estudio de propiedades físico-térmicas en

imágenes se realiza dos casos de estudio, el primer caso se realiza con una imagen de alta resolución tomada a una baja altura, cabe recalcar que este caso de estudio ya ha sido validada en el trabajo de (D. Puga, 2019) para la aplicación en entornos naturales, mientras que para el segundo caso se realiza con imágenes de baja resolución tomadas una altura superior que abarca un mayor espacio a la cual se le aplica los algoritmos GAN para aumentar la resolución de la imagen y se emplea la metodología de estudio de propiedades físico-térmicas. Asimismo para calcular el error se aplica la ecuación del error relativo la misma que se indica más adelante, considerando como valor verdadero el porcentaje del primer caso y como aproximación el porcentaje del segundo caso.[42]

$$\varepsilon_t = \frac{\text{Valor verdadero} - \text{Aproximación}}{\text{Valor verdadero}} * 100\%$$

Tabla 3.5: Comparación de resultados del procesamiento de imágenes de los dos casos de estudio.

CASO	ZONA I	ZONA II	ZONA III
1	5%	12.07%	82.93%
2	5.77%	18.82%	75.41%
Error	15.40%	14.49%	3.04%

(Fuente: Propia)

La Tabla 3.5 muestra el error entre los resultados de cada una de las zonas para los dos casos de estudio. Hidalgo et al. en el artículo “Implicit large eddy simulation of unsteady cloud cavitation around a plane-convex hydrofoil”[56]. Menciona que el error de una predicción es aceptable hasta un 20% y ya que en este caso el error máximo es el 15.40% que corresponde a la Zona I, es aceptable la aplicación de la metodología en estudios similares al segundo caso.

La capacidad de campo es la cantidad de agua que se mantiene en el suelo una vez que este se haya humedecido por completo.[42] Esta es la característica más importante de los humedales la cual es representada por la Zona III de los resultados anteriormente obtenidos y que en base a la información recopilada de la sección 1.4.2, se considera como el 90% por cm<sup>3</sup>.

## **4. CONCLUSIONES**

### **4.1. Conclusiones**

Esta tesis propone un mecanismo de sujeción para una cámara de baja resolución que, a diferencia de los sistemas de sujeción convencionales, este permite alivianar el peso de la carga útil del UAV ya que una cámara de baja resolución es menos pesada que una cámara de alta calidad fotográfica, teniendo un ahorro económico considerable. Además, permite un rápido cambio de posición ya que al tener la forma de una rótula existe un mayor control para fijar el ángulo deseado para la cámara.

La adaptación de una GAN a diferencia de otras redes neuronales no pierde características e información de las imágenes debido a su discriminador, del mismo modo esta red neuronal con una base de datos de imágenes que gran parte son de entornos naturales y con un entrenamiento de 300 épocas permite obtener resultados favorables en el procesamiento de las imágenes aéreas de la zona de estudio, logrando obtener un aumento en la resolución de las imágenes que es apreciable visualmente.

La aplicación de la metodología de estudio de propiedades físico-térmicas desarrollada en estudios previos, modificando los rangos de segmentación y en conjunto con la GAN adaptada, presenta resultados deseables al momento de determinar las zonas húmedas y muy húmedas de cada uno de los casos de estudio.

La comparación de los resultados entre una imagen de alta resolución con una imagen de baja resolución procesada con la GAN, ambas sometidas a la metodología de estudio de propiedades físico-térmicas, permite determinar que es posible realizar estudios con imágenes aéreas con una mayor zona de interés ya que el error máximo alcanzado fue de 15.4% el cual es aceptable según investigaciones previas realizadas.

### **4.2. Trabajos futuros**

La fase de entrenamiento tiene mucha relevancia para obtener los mejores resultados

de una GAN por lo que implementar un base de datos más específica podría ser beneficioso al momento de procesar las imágenes. Por ejemplo, ya no solo utilizar imágenes de entornos naturales en general, sino que se puede efectuar una recopilación de imágenes solo de humedales o una recopilación de imágenes específicas de acuerdo con el caso de estudio que se vaya a realizar; imágenes de manglares, ambientes tropicales, áreas áridas, por nombrar algunos ejemplos. La base de datos debe estar compuesta como mínimo por 900 imágenes, 800 para el entrenamiento y 100 para el testeo. Pero mientras más robusta sea la base de datos es beneficioso para que la red aprenda más características.

Adicionalmente en trabajos futuros para especies en peligro de extinción se puede monitorear el comportamiento y estatus de población mediante la utilización de redes neuronales convolucionales (CNN) ya que permite la detección y selección de objetos.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] S. B. Melo, "TRANSFORMACIONES GEOMÉTRICAS SOBRE IMÁGENES DIGITALES," pp. 1–13.
- [2] T. Hermosilla, E. Bermejo, a. Balaguer, and L. a. Ruiz, "Detección de bordes con precisión subpíxel en imágenes : Interpolación lineal frente a esquemas de tipo no lineal," *VII Jornadas Matemática Apl. DMA–IMPA*, pp. 9–20, 2006, [Online]. Available: [http://cgat.webs.upv.es/bigfiles/Charla\\_Hermosilla\\_JMA\\_06.pdf](http://cgat.webs.upv.es/bigfiles/Charla_Hermosilla_JMA_06.pdf).
- [3] D. Povedano and D. Vetencourt, "Análisis Forense de Imágenes Digitales mediante interpolación cromática," pp. 1–8, 2014.
- [4] J. P. Guerschman, M. J. Hill, L. J. Renzullo, D. J. Barrett, A. S. Marks, and E. J. Botha, "Estimating fractional cover of photosynthetic vegetation, non-photosynthetic vegetation and bare soil in the Australian tropical savanna region upscaling the EO-1 Hyperion and MODIS sensors," *Remote Sens. Environ.*, vol. 113, no. 5, pp. 928–945, 2009, doi: 10.1016/j.rse.2009.01.006.
- [5] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," *BMVC 2012 - Electron. Proc. Br. Mach. Vis. Conf. 2012*, pp. 1–10, 2012, doi: 10.5244/C.26.135.
- [6] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "The 2018 PIRM challenge on perceptual image super-resolution," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11133 LNCS, pp. 334–355, 2019, doi: 10.1007/978-3-030-11021-5\_21.
- [7] R. Timofte, S. Gu, L. Van Gool, L. Zhang, and M. H. Yang, "NTIRE 2018 challenge on single image super-resolution: Methods and results," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2018-June, pp. 965–976, 2018, doi: 10.1109/CVPRW.2018.00130.
- [8] D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, "Learning a deep convolutional network for image super-resolution," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8692, no. September, pp. 184–199, 2014, [Online]. Available: <http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html>.
- [9] W. S. Lai, J. Bin Huang, N. Ahuja, and M. H. Yang, "Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks," *arXiv*, 2017.
- [10] V. M. Salas and F. Paz Pellat, "Estimación de la fracción de la cubierta forestal con el uso de fotografía digital en bosques de referencia del Estado de México," no. March, 2016.
- [11] Y. Xie, Z. Sha, and M. Yu, "Remote sensing imagery in vegetation mapping: a review," *J. Plant Ecol.*, vol. 1, no. 1, pp. 9–23, 2008, doi: 10.1093/jpe/rtm005.

- [12] W. Buytaert and B. De Bievre, "Water for cities: The impact of climate change and demographic growth in the tropical Andes," *Water Resour. Res.*, vol. 48, no. 8, pp. 1–13, 2012, doi: 10.1029/2011WR011755.
- [13] P. A. Keddy, *Wetland ecology: Principles and conservation*. 2010.
- [14] S. Jay *et al.*, "Exploiting the centimeter resolution of UAV multispectral imagery to improve remote-sensing estimates of canopy structure and biochemistry in sugar beet crops," *Remote Sens. Environ.*, vol. 231, no. April, pp. 0–1, 2019, doi: 10.1016/j.rse.2018.09.011.
- [15] Z. Tang *et al.*, "Spatial pattern of pika holes and their effects on vegetation coverage on the Tibetan Plateau: An analysis using unmanned aerial vehicle imagery," *Ecol. Indic.*, vol. 107, no. June, p. 105551, 2019, doi: 10.1016/j.ecolind.2019.105551.
- [16] F. López-Granados, J. Torres-Sánchez, F. M. Jiménez-Brenes, O. Arquero, M. Lovera, and A. I. De Castro, "An efficient RGB-UAV-based platform for field almond tree phenotyping: 3-D architecture and flowering traits," *Plant Methods*, vol. 15, no. 1, pp. 1–16, 2019, doi: 10.1186/s13007-019-0547-0.
- [17] F. Santini, S. C. Kefauver, V. Resco de Dios, J. L. Araus, and J. Voltas, "Using unmanned aerial vehicle-based multispectral, RGB and thermal imagery for phenotyping of forest genetic trials: A case study in *Pinus halepensis*," *Ann. Appl. Biol.*, vol. 174, no. 2, pp. 262–276, 2019, doi: 10.1111/aab.12484.
- [18] EVENT38, "EVENT38," 2018. <https://event38.com/fixed-wing/e384-mapping-drone/>.
- [19] Hobbyking, "Hobbyking," 2020. [https://hobbyking.com/fr\\_fr/skywalker-x-8-fpv-uav-flying-wing-2120mm-arf.html?\\_\\_store=fr\\_fr](https://hobbyking.com/fr_fr/skywalker-x-8-fpv-uav-flying-wing-2120mm-arf.html?__store=fr_fr).
- [20] E. Valencia *et al.*, "Modeling of a series hybrid propulsion uav used for monitoring in the galapagos islands," *AIAA Propuls. Energy 2020 Forum*, pp. 1–18, 2020, doi: 10.2514/6.2020-3960.
- [21] D. Sastre, "Diseño y adaptación de carga de pago para UAV ligero," 2017.
- [22] J. Ullauri, "Diseño y Construcción De Un Estabilizador De Imagen Tipo Gimbal," 2020.
- [23] A. Qadir, W. Semke, and J. Neubert, "Vision based neuro-fuzzy controller for a two axes gimbal system with small uav," *J. Intell. Robot. Syst. Theory Appl.*, vol. 74, no. 3–4, pp. 1029–1047, 2014, doi: 10.1007/s10846-013-9865-z.
- [24] R. Miller, G. Mooty, and J. M. Hilkert, "Gimbal system configurations and line-of-sight control techniques for small UAV applications," *Airborne Intell. Surveillance, Reconnaiss. Syst. Appl. X*, vol. 8713, p. 871308, 2013, doi: 10.1117/12.2015777.
- [25] P. Guerrero, *UNIVERSIDAD CENTRAL DEL ECUADOR FACULTAD DE*

*INGENIERÍA , CIENCIAS FÍSICAS Y MATEMÁTICAS INGENIERÍA EN DISEÑO INDUSTRIAL Diseño y desarrollo de gimbal para cámaras y láseres que funcionaran como escáneres 3d en drones . Trabajo de titulación de modalidad P. 2019.*

- [26] A. Altan and R. Hacıoğlu, "Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances," *Mech. Syst. Signal Process.*, vol. 138, 2020, doi: 10.1016/j.ymssp.2019.106548.
- [27] unmannedsystemssource, "USG-301 EO 3-Axis Gimbal." <https://www.unmannedsystemssource.com/shop/gimbals/usg-301-eo-3-axis-gimbal/>.
- [28] Viewsheen, "Gimbal 3 axis." <https://www.viewsheen.com/3-5x-4k-mini-3axis-drone-gimbal-camera-product/>.
- [29] OCTOPUS, "EPSILON 140." <https://octopus.uavfactory.com/uav-payloads-equipment/epsilon-140>.
- [30] J. V. Valenzuela, "Interpolación de formas en imágenes usando mordología matemática," *Oa.Upm.Es*, 2008, [Online]. Available: <http://oa.upm.es/id/eprint/1282/contents%5Cnhttp://oa.upm.es/1282>.
- [31] J. Gabriel and M. Cruz, "Diseño de una red generativa antagónica para el mejoramiento de la resolución de imágenes," 2019.
- [32] C. Dong, C. Change, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.*, vol. 6, no. 1, pp. 31–42, 2018, doi: 10.1080/21681163.2016.1154806.
- [33] C. Dong, C. Change, and T. Xiaoou, "Accelerating the Super-Resolution Convolutional Neural Network," *Eur. Conf. Comput. Vis.*, pp. 802–817, 2016, doi: 10.1007/978-3-319-46475-6.
- [34] G. Chen, Q. Chen, S. Long, and W. Zhu, "Quantum Convolutional Neural Network for Image Classification," *Proc. - 8th Int. Conf. Digit. Home, ICDH 2020*, pp. 116–120, 2020, doi: 10.1109/ICDH51081.2020.00028.
- [35] M. Sewak, R. Karim, and P. Pujari, *Practical Convolutional Neural Networks*. 2018.
- [36] B. Krinski, "Convolutional Neural Network.," pp. 1–68, 2019, [Online]. Available: <https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529>.
- [37] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," *Proc. - Int. Conf. Pattern Recognit.*, pp. 2366–2369, 2010, doi: 10.1109/ICPR.2010.579.
- [38] A. Tanchenko, "Visual-PSNR measure of image quality," *J. Vis. Commun. Image Represent.*, vol. 25, no. 5, pp. 874–878, 2014, doi: 10.1016/j.jvcir.2014.01.008.



- [39] U. Sara, M. Akter, and M. S. Uddin, "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study," *J. Comput. Commun.*, vol. 07, no. 03, pp. 8–18, 2019, doi: 10.4236/jcc.2019.73002.
- [40] P. Gupta, P. Srivastava, S. Bhardwaj, and V. Bhateja, "Assessment of Color Images," *2011 Int. Conf. Commun. Ind. Appl.*, pp. 1–4, 2011.
- [41] A. Rehman, M. Rostami, Z. Wang, D. Brunet, and E. R. Vrscay, "SSIM-inspired image restoration using sparse representation," *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, pp. 1–12, 2012, doi: 10.1186/1687-6180-2012-16.
- [42] D. Puga, "DESARROLLO DE UN CÓDIGO DE PROGRAMACIÓN EN LENGUAJE PYTHON PARA EL ESTUDIO DE ZONAS FORESTALES," Escuela Politécnica Nacional, 2019.
- [43] J. Pan, *Advances in Intelligent Information Hiding and Multimedia Signal Processing*. 2017.
- [44] M. Granja and R. Poveda, "DESARROLLO DE UN CÓDIGO DE PROGRAMACIÓN EN LENGUAJE PYTHON PARA EL ESTUDIO DE ZONAS FORESTALES," 2017.
- [45] N. a Ibraheem, M. M. Hasan, R. Z. Khan, and P. K. Mishra, "Understanding Color : A Review," *ARNP J. Sci. Technol.*, vol. 2, no. 3, pp. 265–275, 2012.
- [46] Wikipedia, "Modelo de color HSV." [https://es.wikipedia.org/wiki/Modelo\\_de\\_color\\_HSV](https://es.wikipedia.org/wiki/Modelo_de_color_HSV).
- [47] L. Llambí, A. Soto, P. Borja, B. Ochoa, R. Celleri, and B. Bievre, *Páramos Andinos: Ecología, hidrología y suelos de páramos*. 2012.
- [48] W. Buytaert, R. Céleri, B. De Bièvre, and F. Cisneros, "Hidrología del Páramo Andino: Propiedades, Importancia y Vulnerabilidad," *Soil Water*, pp. 1–26, 2003.
- [49] R. Hofstede, *Los Páramos del Mundo*. 2003.
- [50] EMAPS and FONAG, "Plan de manejo del área de conservación hídrica Antisana," 2018.
- [51] T. Left, "SRGAN." <https://github.com/leftthomas/SRGAN>.
- [52] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," *arXiv*, pp. 1–16, 2018.
- [53] N. Takano and G. Alaghband, "SRGAN: Training Dataset Matters," pp. 1–7, 2019, [Online]. Available: <http://arxiv.org/abs/1903.09922>.
- [54] Pyimagesearch, "How to create a deep learning dataset using Google Images." <https://pyimagesearch.com/2017/12/04/how-to-create-a-deep-learning-dataset-using-google-images/>.

- [55] D. Nowdon, "Imagen download for google." <https://gist.github.com/davesnowdon/2016d4e9f069ff1788ede4f2902bd198>.
- [56] V. Hidalgo, X. W. Luo, X. Escaler, B. Ji, and A. Aguinaga, "Implicit large eddy simulation of unsteady cloud cavitation around a plane-convex hydrofoil," *J. Hydrodyn.*, vol. 27, no. 6, pp. 815–823, 2015, doi: 10.1016/S1001-6058(15)60544-3.

## ANEXOS

### ANEXO I

#### Comparación visual del resultado de la GAN

Imagen de baja resolución



Imagen procesada



## ANEXO II

### Código de Entrenamiento

```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
    print('Select the Runtime > "Change runtime type" menu to enable a GPU accelerator, ')
    print('and then re-execute this cell.')
else:
    print(gpu_info)

from psutil import virtual_memory
ram_gb = virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM\n'.format(ram_gb))

if ram_gb < 20:
    print('To enable a high-RAM runtime, select the Runtime > "Change runtime type"')
    print('menu, and then select High-RAM in the Runtime shape dropdown. Then, ')
    print('re-execute this cell.')
else:
    print('You are using a high-RAM runtime!')

!pip uninstall torch
!pip install torch==1.4.0+cu92 torchvision==0.5.0+cu92 -
f https://download.pytorch.org/whl/torch_stable.html
!pip install import-ipynb

%cd '/content/drive/MyDrive/Códigos/ARI'

import import_ipynb
import argparse
import os
from math import log10

import pandas as pd
import torch.optim as optim
import torch.utils.data
import torchvision.utils as utils
from torch.autograd import Variable
```

```

from torch.utils.data import DataLoader
from tqdm import tqdm

import pytorch_ssim
from data_utils import TrainDatasetFromFolder, ValDatasetFromFolder, display_transform
from loss import GeneratorLoss
from model import Generator, Discriminator

parser = argparse.ArgumentParser(description='Train Super Resolution Models')
parser.add_argument('--crop_size', default=88, type=int, help='training images crop size')
parser.add_argument('--upscale_factor', default=4, type=int, choices=[2, 4, 8],
                    help='super resolution upscale factor')
parser.add_argument('--num_epochs', default=300, type=int, help='train epoch number')

parser.add_argument('-f')

if __name__ == '__main__':
    opt = parser.parse_args()

    CROP_SIZE = opt.crop_size
    UPSCALE_FACTOR = opt.upscale_factor
    NUM_EPOCHS = opt.num_epochs

    train_set = TrainDatasetFromFolder('data/train', crop_size=CROP_SIZE, upscale_factor=UPSCALE_FACTOR)
    val_set = ValDatasetFromFolder('data/valid', upscale_factor=UPSCALE_FACTOR)
    train_loader = DataLoader(dataset=train_set, num_workers=4, batch_size=64, shuffle=True) #64
    val_loader = DataLoader(dataset=val_set, num_workers=4, batch_size=1, shuffle=False)

    netG = Generator(UPSCALE_FACTOR)
    print('# generator parameters:', sum(param.numel() for param in netG.parameters()))
    netD = Discriminator()
    print('# discriminator parameters:', sum(param.numel() for param in netD.parameters()))

    generator_criterion = GeneratorLoss()

```

```

if torch.cuda.is_available():
    netG.cuda()
    netD.cuda()
    generator_criterion.cuda()

optimizerG = optim.Adam(netG.parameters())
optimizerD = optim.Adam(netD.parameters())

results = {'d_loss': [], 'g_loss': [], 'd_score': [], 'g_score': [], 'psnr': []
, 'ssim': []}

for epoch in range(1, NUM_EPOCHS + 1):
    train_bar = tqdm(train_loader)
    running_results = {'batch_sizes': 0, 'd_loss': 0, 'g_loss': 0, 'd_score': 0
, 'g_score': 0}

    netG.train()
    netD.train()
    for data, target in train_bar:
        g_update_first = True
        batch_size = data.size(0)
        running_results['batch_sizes'] += batch_size

        #####
        # (1) Update D network: maximize  $D(x) - 1 - D(G(z))$ 
        #####
        real_img = Variable(target)
        if torch.cuda.is_available():
            real_img = real_img.cuda()
        z = Variable(data)
        if torch.cuda.is_available():
            z = z.cuda()
        fake_img = netG(z)

        netD.zero_grad()
        real_out = netD(real_img).mean()
        fake_out = netD(fake_img).mean()
        d_loss = 1 - real_out + fake_out
        d_loss.backward(retain_graph=True)
        optimizerD.step()

        #####
        # (2) Update G network: minimize  $1 - D(G(z)) + \text{Perception Loss} + \text{Image Loss} + \text{TV Loss}$ 
        #####

```

```

netG.zero_grad()
g_loss = generator_criterion(fake_out, fake_img, real_img)
g_loss.backward()

fake_img = netG(z)
fake_out = netD(fake_img).mean()

optimizerG.step()

# loss for current batch before optimization
running_results['g_loss'] += g_loss.item() * batch_size
running_results['d_loss'] += d_loss.item() * batch_size
running_results['d_score'] += real_out.item() * batch_size
running_results['g_score'] += fake_out.item() * batch_size

train_bar.set_description(desc='[%d/%d] Loss_D: %.4f Loss_G: %.4f D(x):
%.4f D(G(z)): %.4f' % (
    epoch, NUM_EPOCHS, running_results['d_loss'] / running_results['bat
ch_sizes'],
    running_results['g_loss'] / running_results['batch_sizes'],
    running_results['d_score'] / running_results['batch_sizes'],
    running_results['g_score'] / running_results['batch_sizes']))

netG.eval()
out_path = 'training_results/' + 'us' + str(UPSCALE_FACTOR) + '_e' + str(NUM_EPOCHS) + '_bz64_dataeditada' + '/'
if not os.path.exists(out_path):
    os.makedirs(out_path)

with torch.no_grad():
    val_bar = tqdm(val_loader)
    valing_results = {'mse': 0, 'ssims': 0, 'psnr': 0, 'ssim': 0, 'batch_si
zes': 0}

    val_images = []
    for val_lr, val_hr_restore, val_hr in val_bar:
        batch_size = val_lr.size(0)
        valing_results['batch_sizes'] += batch_size
        lr = val_lr
        hr = val_hr
        if torch.cuda.is_available():
            lr = lr.cuda()
            hr = hr.cuda()
        sr = netG(lr)

```

```

        batch_mse = ((sr - hr) ** 2).data.mean()
        valing_results['mse'] += batch_mse * batch_size
        batch_ssim = pytorch_ssim.ssim(sr, hr).item()
        valing_results['ssims'] += batch_ssim * batch_size
        valing_results['psnr'] = 10 * log10((hr.max()**2) / (valing_results
['mse'] / valing_results['batch_sizes']))
        valing_results['ssim'] = valing_results['ssims'] / valing_results['
batch_sizes']

    val_bar.set_description(
        desc='[converting LR images to SR images] PSNR: %.4f dB SSIM: %
.4f' % (
            valing_results['psnr'], valing_results['ssim']))

    val_images.extend(
        [display_transform()(val_hr_restore.squeeze(0)), display_transf
orm()(hr.data.cpu()).squeeze(0)],
        display_transform()(sr.data.cpu()).squeeze(0)])
    val_images = torch.stack(val_images)
    val_images = torch.chunk(val_images, val_images.size(0) // 15)
    val_save_bar = tqdm(val_images, desc='[saving training results]')
    index = 1
    for image in val_save_bar:
        image = utils.make_grid(image, nrow=3, padding=5)
        utils.save_image(image, out_path + 'epoch_%d_index_%d.png' % (epoch
, index), padding=5)
        index += 1

    # save model parameters
    save_path = 'epochs/e' + str(NUM_EPOCHS) + '_us' + str(UPSCALE_FACTOR) + '_
bz64/' #+ '/'
    if not os.path.exists(save_path):
        os.makedirs(save_path)

    torch.save(netG.state_dict(), 'epochs/e' + str(NUM_EPOCHS) + '_us' + str(UP
SCALE_FACTOR) + '_bz64' + '/netG_epoch_%d_%d.pth' % (UPSCALE_FACTOR, epoch))
    torch.save(netD.state_dict(), 'epochs/e' + str(NUM_EPOCHS) + '_us' + str(UP
SCALE_FACTOR) + '_bz64' + '/netD_epoch_%d_%d.pth' % (UPSCALE_FACTOR, epoch))
    # save loss\scores\psnr\ssim
    results['d_loss'].append(running_results['d_loss'] / running_results['batch
_sizes'])
    results['g_loss'].append(running_results['g_loss'] / running_results['batch
_sizes'])
    results['d_score'].append(running_results['d_score'] / running_results['bat
ch_sizes'])

```



```

        results['g_score'].append(running_results['g_score'] / running_results['batch_sizes'])
        results['psnr'].append(valing_results['psnr'])
        results['ssim'].append(valing_results['ssim'])

    if epoch % 10 == 0 and epoch != 0:
        out_path = 'statistics/'
        data_frame = pd.DataFrame(
            data={'Loss_D': results['d_loss'], 'Loss_G': results['g_loss'], 'Score_D': results['d_score'],
                  'Score_G': results['g_score'], 'PSNR': results['psnr'], 'SSIM': results['ssim']},
            index=range(1, epoch + 1))
        data_frame.to_csv(out_path + 'srf_' + str(UPSCALE_FACTOR) + '_train_results_bz64.csv', index_label='Epoch')

```

## ANEXO III

### Código de Testeo

```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
    print('Select the Runtime > "Change runtime type" menu to enable a GPU accelerator, ')
    print('and then re-execute this cell.')
else:
    print(gpu_info)

!pip install import-ipynb
import import_ipynb

%cd '/content/drive/MyDrive/Códigos/ARI'

import os.path as osp
import glob
import cv2
import numpy as np
import torch
from model import Generator

upscale_factor = 4 # modificable según parámetros del entrenamiento
num_epochs_path = 300 # modificable según parámetros del entrenamiento
num_epochs = 300 # modificable según parámetros del entrenamiento
batch_size = 64 # modificable según parámetros del entrenamiento

model_path = 'epochs/e' + str(num_epochs_path) + '_us' + str(upscale_factor) + '_bz'
+ str(batch_size) + '/netG_epoch_' + str(upscale_factor) + '_' + str(num_epochs)
+ '.pth' # models/RRDB_ESRGAN_x4.pth OR models/RRDB_PSNR_x4.pth
device = torch.device('cuda') # if you want to run on CPU, change 'cuda' -> cpu

test_img_folder = 'Test_LR/*'

model = Generator(upscale_factor).eval()
if device:
    model.cuda()
    model.load_state_dict(torch.load(model_path))
else:
```

```

    model.load_state_dict(torch.load(model_path, map_location=lambda storage, loc:
storage))

print('Model path {:s}. \nTesting...'.format(model_path))

idx = 0
for path in glob.glob(test_img_folder):
    idx += 1
    base = osp.splitext(osp.basename(path))[0]
    print(idx, base)
    # read images
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    img = img * 1.0 / 255
    img = torch.from_numpy(np.transpose(img[:, :, [2, 1, 0]], (2, 0, 1))).float()
    img_LR = img.unsqueeze(0)
    img_LR = img_LR.to(device)

    with torch.no_grad():
        output = model(img_LR).data.squeeze().float().cpu().clamp_(0, 1).numpy()
        output = np.transpose(output[[2, 1, 0], :, :], (1, 2, 0))
        output = (output * 255.0).round()
        cv2.imwrite('Results_HR/' + 'r1t' + '_e' + str(num_epochs) + '_X' + str(upscale_
factor) + '_{:s}.png'.format(base), output)

```

## ANEXO IV

### Código para obtener las urls de las imágenes a descargar

/\*\*\*\*\*\* PARTE 1 \*\*\*\*\*/

```
function simulateRightClick( element ) {
    var event1 = new MouseEvent( 'mousedown', {
        bubbles: true,
        cancelable: false,
        view: window,
        button: 2,
        buttons: 2,
        clientX: element.getBoundingClientRect().x,
        clientY: element.getBoundingClientRect().y
    } );
    element.dispatchEvent( event1 );
    var event2 = new MouseEvent( 'mouseup', {
        bubbles: true,
        cancelable: false,
        view: window,
        button: 2,
        buttons: 0,
        clientX: element.getBoundingClientRect().x,
        clientY: element.getBoundingClientRect().y
    } );
    element.dispatchEvent( event2 );
    var event3 = new MouseEvent( 'contextmenu', {
        bubbles: true,
        cancelable: false,
        view: window,
        button: 2,
        buttons: 0,
        clientX: element.getBoundingClientRect().x,
        clientY: element.getBoundingClientRect().y
    } );
    element.dispatchEvent( event3 );
}

/**
 * grabs a URL Parameter from a query string because Google Images
 * stores the full image URL in a query parameter
 *
 * @param {string} queryString The Query String
 * @param {string} key The key to grab a value for
 *
 * @return {string} value
 */
function getURLParam( queryString, key ) {
    var vars = queryString.replace( /\^?/, "" ).split( '&' );
    for ( let i = 0; i < vars.length; i++ ) {
        let pair = vars[ i ].split( '=' );
        if ( pair[0] == key ) {
            return pair[1];
        }
    }
}
```

```

    }
    return false;
}

/**
 * Generate and automatically download a txt file from the URL contents
 *
 * @param {string} contents The contents to download
 *
 * @return {void}
 */
function createDownload( contents ) {
    var hiddenElement = document.createElement( 'a' );
    hiddenElement.href = 'data:attachment/text,' + encodeURIComponent( contents );
    hiddenElement.target = '_blank';
    hiddenElement.download = 'urls.txt';
    hiddenElement.click();
}

/**
 * grab all URLs va a Promise that resolves once all URLs have been
 * acquired
 *
 * @return {object} Promise object
 */
function grabUrls() {
    var urls = [];
    return new Promise( function( resolve, reject ) {
        var count = document.querySelectorAll(
            '.isv-r a:first-of-type' ).length,
            index = 0;
        Array.prototype.forEach.call( document.querySelectorAll(
            '.isv-r a:first-of-type' ), function( element ) {
            // using the right click menu Google will generate the
            // full-size URL; won't work in Internet Explorer
            // (http://pyimg.co/byukr)
            simulateRightClick( element.querySelector( ':scope img' ) );
            // Wait for it to appear on the <a> element
            var interval = setInterval( function() {
                if ( element.href.trim() !== '' ) {
                    clearInterval( interval );
                    // extract the full-size version of the image
                    let googleUrl = element.href.replace( /\.*(\?)/, '$1' );
                    fullImageUrl = decodeURIComponent(
                        getURLParam( googleUrl, 'imgurl' ) );
                    if ( fullImageUrl !== 'false' ) {
                        urls.push( fullImageUrl );
                    }
                    // sometimes the URL returns a "false" string and
                    // we still want to count those so our Promise
                    // resolves
                    index++;
                    if ( index == ( count - 1 ) ) {
                        resolve( urls );
                    }
                }
            }, 10 );
        } );
    } );
}

```

```
/****** PARTE 2 *****/
```

```
/**  
 * Call the main function to grab the URLs and initiate the download  
 */  
grabUrls().then( function( urls ) {  
    urls = urls.join( '\n' );  
    createDownload( urls );  
} );
```

## ANEXO V

### Código para descargar las imágenes para la base de datos

```
from imutils import paths
import argparse
import requests
import cv2
import os

%cd '/content/drive/MyDrive/Códigos/DDB'

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("--
urls", default="/content/drive/MyDrive/Códigos/DDB/urls.txt", type=str, help="path
to file containing image URLs")
#CAMBIAR LA RUTA DEL ARCHIVO TXT DE LAS URLS DE L
AS IMÁGANES A DESCARGAR
ap.add_argument("--
output", default="/content/drive/MyDrive/Códigos/DDB/Dow_Img", type=str, help="path
to output directory of images")
#CAMBIAR LA RUTA DE LA CARPETA DONDE SE DESCARGA
RÁ LAS IMÁGENES
ap.add_argument('-f')

# grab the list of URLs from the input file, then initialize the
# total number of images downloaded thus far
args = vars(ap.parse_args())
rows = open(args["urls"]).read().strip().split("\n")
total = 0
# loop the URLs
for url in rows:
    try:
        # try to download the image
        r = requests.get(url, timeout=60)
        # save the image to disk
        p = os.path.sep.join([args["output"], "{}.jpg".format(
            str(total).zfill(8))])
        f = open(p, "wb")
        f.write(r.content)
        f.close()
        # update the counter
        print("[INFO] downloaded: {}".format(p))
```

```

    total += 1
# handle if any exceptions are thrown during the download process
except:
    print("[INFO] error downloading {}...skipping".format(p))
# loop over the image paths we just downloaded
for imagePath in paths.list_images(args["output"]):
    # initialize if the image should be deleted or not
    delete = False
    # try to load the image
    try:
        image = cv2.imread(imagePath)
        # if the image is `None` then we could not properly load it
        # from disk, so delete it
        if image is None:
            delete = True
    # if OpenCV cannot load the image then the image is likely
    # corrupt so we should delete it
    except:
        print("Except")
        delete = True
# check to see if the image should be deleted
if delete:
    print("[INFO] deleting {}".format(imagePath))
    os.remove(imagePath)

```



## ANEXO VI

### Código de segmentación de imágenes para estudio de propiedades físico – térmicas

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

%cd '/content/drive/MyDrive/Códigos/PFT'

img = cv2.imread('Antisana_Caso2_rlt.png', cv2.IMREAD_UNCHANGED) #nombre de la imagen a procesar
img1 = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
hsv_img = cv2.cvtColor(img1, cv2.COLOR_RGB2HSV)
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

lower_zi = (20,0,0)
upper_zi = (170,255,255)
mask_zi = cv2.inRange(hsv_img, lower_zi, upper_zi)
result_zi = cv2.bitwise_and(img1, img1, mask=mask_zi)

#Máscara 1

lower_nv = (88,0,84)
upper_nv = (170,255,255)
mask1 = cv2.inRange(hsv_img, lower_nv, upper_nv)

lower_nv1 = (20,0,50)
upper_nv1 = (162,58,92)
mask1_1 = cv2.inRange(hsv_img, lower_nv1, upper_nv1)

masknv = mask1 + mask1_1
resultnv = cv2.bitwise_and(img1, img1, mask=masknv)

#Máscara 2

lower_lm = (20,0,50)
upper_lm = (175,255,255)
mask2 = cv2.inRange(hsv_img, lower_lm, upper_lm)

lower_lm2 = (0,0,50)
upper_lm2 = (255,255,255)
mask2_2 = cv2.inRange(hsv_img, lower_lm2, upper_lm2)
```

```

masklm2 = mask2 - mask2_2
result2 = cv2.bitwise_and(img1, img1, mask=masklm2)

#Máscara 3

lower_mh = (20,0,50)
upper_mh = (45,255,255)
mask3 = cv2.inRange(hsv_img, lower_mh, upper_mh)

result3 = cv2.bitwise_and(img1, img1, mask=mask3)

pxl_zona1 = cv2.countNonZero(masknv)
pxl_zona2 = cv2.countNonZero(masklm2)
pxl_zona3 = cv2.countNonZero(mask3)
pxl_zonas = pxl_zona1 + pxl_zona2 + pxl_zona3
pxl_img_gray = cv2.countNonZero(img_gray)
pxl_zonainteres = cv2.countNonZero(mask_zi)

p_zona1 = ("{0:.2f}".format(pxl_zona1*100/pxl_zonas))
p_zona2 = ("{0:.2f}".format(pxl_zona2*100/pxl_zonas))
p_zona3 = ("{0:.2f}".format(pxl_zona3*100/pxl_zonas))
p_all_cl = ("{0:.2f}".format(pxl_zonas*100/pxl_img_gray))

fig = plt.figure(figsize=(10,7))
fig.canvas.set_window_title('Zonas Forestales Procesamiento a Color')

plt.subplot(2,3,2)
plt.imshow(img1, cmap=plt.cm.gray)
plt.axis('off')
plt.title('Procesamiento a Color' + ' ' + p_all_cl + '% Zona de Interés')

plt.subplot(2,3,4)
plt.imshow(resultnv, cmap=plt.cm.gray)
plt.axis('off')
plt.title('Zona I' + ' ' + p_zona1 + '%')

plt.subplot(2,3,5)
plt.imshow(result2, cmap=plt.cm.gray)
plt.axis('off')
plt.title('Zona II' + ' ' + p_zona2 + '%')

plt.subplot(2,3,6)

```

```
plt.imshow(result3,cmap=plt.cm.gray)
plt.axis('off')
plt.title('Zona III' + ' ' + p_zona3 + '%')

plt.show()
```