



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**Comparación de algoritmos de segmentación
de imágenes digitales de plantas de arroz en
ambientes no controlados**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
DE SISTEMAS**

Autor:

Bach. Sandoval Odar William

ORCID: <https://orcid.org/0000-0003-0665-8981>

Asesor:

Mg. Bravo Ruiz Jaime Arturo

ORCID: <https://orcid.org/0000-0003-1929-3969>

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú

2023

**COMPARACIÓN DE ALGORITMOS DE SEGMENTACIÓN DE IMÁGENES
DIGITALES DE PLANTAS DE ARROZ EN AMBIENTES NO CONTROLADOS.**

Aprobación Del Jurado

MG. BANCES SAAVEDRA DAVID ENRIQUE

Presidente Del Jurado De Tesis

MG. ARCILA DIAZ JUAN CARLOS

Secretario Del Jurado De Tesis

MG. BRAVO RUIZ JAIME ARTURO

Vocal Del Jurado De Tesis



DECLARACIÓN JURADA DE ORIGINALIDAD

Quien(es) suscribe(n) la **DECLARACIÓN JURADA**, soy(somos) **egresado (s)** del Programa de Estudios de **Ingeniería de Sistemas** de la Universidad Señor de Sipán S.A.C, declaro(amos) bajo juramento que soy (somos) autor(es) del trabajo titulado:

COMPARACIÓN DE ALGORITMOS DE SEGMENTACIÓN DE IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN AMBIENTES NO CONTROLADOS

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación a las citas y referencias bibliográficas, respetando al derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firman:

Sandoval Odar William	DNI: 43516962	 
-----------------------	---------------	--

Pimentel, 03 de noviembre de 2023

Dedicatoria

*A Dios, mi padre celestial quien me
acompaña en el trayecto de mi vida
y quien me levanta cuando tropiezo
en el transcurrir de mi camino.*

*A mi padre Marcelino y hermana Flor,
seres queridos a quienes más amo por
lograr formarme como la persona que
soy actualmente por el apoyo
incondicional y sus sabios consejos que
me brindan.*

Agradecimiento

*A Dios Padre Celestial por permitirme
estar al lado de mi familia y disfrutar
de la vida junto a ellos a quienes más
quiero.*

*A mis Padres Marcelino y Olga, y también
a mi Hermana Flor por apoyarme en cada
decisión tomada en mi trayectoria
personal y profesional.*

*A mis Maestros y Compañeros de
estudios por haberme compartido
sus conocimientos durante la
formación de mi carrera profesional.*

ÍNDICE

Dedicatoria.....	iv
Agradecimiento	v
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xiii
Resumen	xiv
Abstract	xv
I. INTRODUCCIÓN.....	16
1.1. Realidad Problemática.....	16
1.1.1 Antecedentes de Estudio	17
1.2. Formulación del problema.....	48
1.3. Hipótesis.....	49
1.4. Objetivos	49
1.4.1. Objetivo General.....	49
1.4.2. Objetivos Específicos	49
1.5. Teorías relacionadas al tema.....	50
1.5.1. Procesamiento digital de imágenes.....	50
1.5.2. Relaciones básicas entre pixeles	50
1.5.3. Vecinos de un pixel	50
1.5.3.1 La vecindad 4-vecinos.....	50
1.5.3.2 La vecindad 8-vecinos.....	51
1.5.4 Conectividad	51
1.5.4.1 Conectividad- 4.....	51
1.5.4.2 Conectividad- 8.....	52
1.5.5 Las medidas de distancia	52
1.5.6 Segmentación	53
1.5.7 Segmentación de imágenes	53
1.5.8 Imágenes binarias.....	53
1.5.9 Textura.....	54
1.5.10 Segmentación por Bordes.....	54
1.5.11 Color.....	54
1.5.12 Conversión de RGB a HSV	54
1.5.13 El Modelo de CIE L*a*b*	55
1.5.14 Separación de las componentes RGB en imágenes de grises.....	55
1.5.15 Imágenes en escala de grises	55
1.5.16 Filtrado de la mediana	55

1.5.17	¿Qué es un histograma?	56
1.5.18	Curva ROC (Característica operativa del receptor)	56
1.5.19	Matriz de confusión	56
1.5.20	Métricas derivadas de la matriz de confusión	57
1.5.21	Intersección sobre la unión (IoU)	59
1.5.22	Complejidad de un Algoritmo	59
1.5.23	Las Neuronas	59
1.5.24	Dimensiones de una red Neuronal	60
1.5.25	Funcionamiento y entrenamiento de una neurona	61
1.5.26	Algoritmo	61
1.5.26.1	Algoritmo PSO	61
1.5.26.2	Agrupamiento jerárquico	61
1.5.26.3	Algoritmo K-means	61
1.5.26.4	Metodo OTSU	66
1.5.26.5	Redes neuronales convolucionales	66
1.5.26.6	Algoritmo SLIC	67
1.5.26.7	Algoritmo MS	67
1.5.26.8	Algoritmo DTSM	67
1.5.26.9	Algoritmo DBSCAN	67
1.5.27	Clasificadores	70
1.5.27.1	Máquina de vectores de soporte SVM	70
1.5.27.2	Random Forest RB	71
1.5.28	Sistemas	72
1.5.29	Sistemas de Información	72
1.5.30	Lenguaje de Programación	72
1.5.30.1	C++	72
1.5.30.2	Qué es java	72
1.5.30.3	Python	73
II.	MATERIAL Y MÉTODO	73
2.1.	Tipo y Diseño de Investigación	73
2.1.1.	Tipo de investigación	73
2.1.2.	Diseño de investigación	73
2.2.	Variables, Operacionalización	74
2.2.1.	Operacionalización	74
2.2.2.	Variables	76
2.3.	Población de estudio, muestra, muestreo y criterios de selección	76

2.4.	Técnicas e instrumentos de recolección de datos, validez y confiabilidad.....	76
2.4.1.	Técnica.....	76
2.4.2.	Instrumentos	77
2.5.	Procedimientos de análisis de datos.....	77
2.5.1.	Variable Independiente.....	79
2.5.2.	Variable dependiente	81
2.6.	Criterios éticos	85
2.6.1.	Confidencialidad	85
2.6.2.	Criterios de rigor Científico	85
III.	RESULTADOS	86
3.1.	Resultados.....	86
3.2.	Discusión	96
3.3.	Aporte de la investigación.....	99
3.3.1.	Aporte Practico.....	99
IV.	CONCLUSIONES Y RECOMENDACIONES	156
4.1.	Conclusiones	156
4.2.	Recomendaciones	158
	REFERENCIAS.....	159
	ANEXOS.....	166

ÍNDICE DE FIGURAS

Figura 1: Vecindad 4. Fuente: [20]	51
Figura 2: Vecindad 8. Fuente: [20]	51
Figura 3: Conectividad - 4. Fuente: [20]	52
Figura 4: Conectividad - 8. Fuente: [20]	52
Figura 5. Distancia euclidiana. Fuente: [20]	53
Figura 6: Matriz de confusión binaria. Fuente: [27].....	57
Figura 7. El error de la clasificación. Fuente: [27].....	57
Figura 8. Exactitud. Fuente: [27]	58
Figura 9. El recall y la sensibilidad. Fuente: [27]	58
Figura 10. Especificidad. Fuente: [27].....	58
Figura 11. Intersección sobre la unión. Fuente: [28].....	59
Figura 12. Ejemplo de visualización de resultados de K-MEANS. Fuente: [27]	62
Figura 13. Hiperplanos lineales de la máquina de vectores de soporte. Fuente: [32]	71
Figura 14. Árbol de decisión. Fuente: [32].....	71
Figura 15: Rendimiento de cómputo en 25 imágenes de plantas de arroz hacia adelante en día nublado. Fuente: Elaboración Propia.	87
Figura 16: Rendimiento de computo en 25 imágenes de plantas de arroz hacia atrás en día nublado. Fuente: Elaboración Propia.	87
Figura 17: Rendimiento de cómputo en 25 imágenes de plantas de arroz en posición diagonal hacia la derecha en día soleado. Fuente: Elaboración Propia.....	88
Figura 18: Rendimiento de cómputo en 25 imágenes de plantas de arroz en posición diagonal hacia la izquierda en día soleado. Fuente: Elaboración Propia.	88
Figura 19: Rendimiento de cómputo total de 100 imágenes digitales de plantas de arroz utilizando los algoritmos K-MEANS y DBSCAN. Fuente: Elaboración Propia.....	89
Figura 20: Segmentación de 25 imágenes digitales de plantas de arroz hacia adelante en día nublado. Fuente: Elaboración Propia.	91

Figura 21: Segmentación de 25 imágenes digitales de plantas de arroz hacia atrás en día nublado. Fuente: Elaboración Propia.	92
Figura 22: Segmentación de 25 imágenes digitales de plantas de arroz en posición diagonal hacia la derecha en día soleado. Fuente: Elaboración Propia.	93
Figura 23: Segmentación de 25 imágenes digitales de plantas de arroz en posición diagonal hacia la izquierda en día soleado. Fuente: Elaboración Propia.	94
Figura 24: Promedio total de 100 imágenes digitales de plantas de arroz segmentadas utilizando los algoritmos K-MEANS y DBSCAN. Fuente: Elaboración Propia.	95
Figura 25: Adquisición de imágenes de plantas de arroz. Fuente: Elaboración propia.	101
Figura 26: Formas de adquisición y cantidades de imágenes de plantas de arroz. Fuente: Elaboración Propia.	103
Figura 27: Dispositivo de adquisición de imágenes digitales. Fuente: Elaboración Propia.	104
Figura 28: Base de datos de imágenes de plantas de arroz. Fuente: Elaboración Propia.	106
Figura 29: Esquema de segmentación de plantas de arroz en ambiente no controlado utilizando algoritmo K-MEANS. Fuente: Elaboración propia.	117
Figura 30: Librerías utilizadas en Python versión 3.7. Fuente: Elaboración propia.	120
Figura 31: Imagen original de plantas de arroz en ambiente no controlado. Fuente: Elaboración propia.	121
Figura 32: Codificación del ingreso de imágenes de plantas de arroz. Fuente: Elaboración propia.	121
Figura 33: Redimensión de las imágenes de plantas de arroz. Fuente: Elaboración propia.	122
Figura 34: Formula de Espacio de color RGB. Fuente: [45]	122
Figura 35: Blancos de referencia en la luminancia. Fuente: [45]	123
Figura 36: Luminosidad en coordenada L. Fuente: [45]	124
Figura 37: Diferencias de color en coordenadas a y b. Fuente: [45].	124
Figura 38: Función de prevención de pendiente infinita. Fuente: [45].	124
Figura 39: Función delta. Fuente: [45].	125
Figura 40: Valores convertidos al tipo de dato destino. Fuente: [45]	125

Figura 41: Formula de espacio de color CIELab. Fuente: [45].....	125
Figura 42: Convirtió las imágenes RGB a Espacio de color CIELab. Fuente: Elaboración propia.....	126
Figura 43: Resultado de imagen original de plantas de arroz. Fuente: Elaboración propia.	127
Figura 44: Resultado de imagen de plantas de arroz en su dimensión L. Fuente: Elaboración propia.....	127
Figura 45: Resultado de imagen de plantas de arroz en su dimensión a. Fuente: Elaboración propia.....	128
Figura 46: Resultado de imagen de plantas de arroz en su dimensión b. Fuente: Elaboración propia.....	128
Figura 47: Algoritmo Kmeans K=2. Fuente: Elaboración propia.	129
Figura 48: Criterios utilizados para realizar la segmentación de imágenes de plantas de arroz. Fuente: Elaboración propia.	131
Figura 49: Función del algoritmo K-MEANS. Fuente: Elaboración propia.....	132
Figura 50: Aplicación de etiquetas a los centroides para segmentación de las imágenes de plantas de arroz. Fuente: Elaboración propia.....	132
Figura 51: Imagen de ingreso de plantas de arroz binarizada. Fuente: Elaboración propia.....	138
Figura 52: Imagen de salida de plantas de arroz binarizada. Fuente: Elaboración propia.	138
Figura 53: Esquema de segmentación de plantas de arroz en ambiente no controlado utilizando algoritmo DBSCAN. Fuente: Elaboración propia.	139
Figura 54: Redimensión de las imágenes de plantas de arroz. Fuente: Elaboración propia.....	140
Figura 55: Conversión de espacio de color RGB a las dimensiones V y S del espacio de color HSV. Fuente: [36]	141
Figura 56: Dimensión H del espacio de color HSV. Fuente: [36]	142
Figura 57: Convierte una imagen de espacio de color RGB a HSV. Fuente: Elaboración propia.....	143
Figura 58: Imagen de plantas de arroz RGB. Fuente: Elaboración propia.....	144

Figura 59: Imagen de plantas de arroz en su dimensión H. Fuente: Elaboración propia. ..	144
Figura 60: Imagen de plantas de arroz en su dimensión S. Fuente: Elaboración propia. ..	145
Figura 61: Imagen de plantas de arroz en su dimensión V. Fuente: Elaboración propia. ..	145
Figura 62: Algoritmo DBSCAN. Eps = 10 Fuente: Elaboración Propia.	146
Figura 63: Parámetros utilizados para segmentar con el algoritmo DBSCAN. Fuente: Elaboración propia.	148
Figura 64: Imagen de ingreso de plantas de arroz binarizada. Fuente: Elaboración propia.....	154
Figura 65: Imagen de salida de plantas de arroz binarizada. Fuente: Elaboracion propia..	155

ÍNDICE DE TABLAS

Tabla 1.	Producción nacional en el Perú. Sub sector agrícola: junio 2023.....	16
Tabla 2.	Operacionalización de variables	74
Tabla 3.	Descripción de las variables de la fórmula Tiempo de Respuesta	79
Tabla 4.	Descripción de las variables de la formula Consumo de memoria RAM	80
Tabla 5.	Descripción de las variables de la fórmula del Consumo del CPU.....	81
Tabla 6.	Descripción de las variables de la formula Exactitud	82
Tabla 7.	Descripción de las variables de la Formula Sensibilidad.....	83
Tabla 8.	Descripción de las variables de la formula Especificidad	84
Tabla 9.	Configuración del dispositivo de adquisición de imágenes	101
Tabla 10.	Características del dispositivo de adquisición de imágenes digitales.	104
Tabla 11:	Comparación de algoritmos de segmentación.	107
Tabla 12.	Resultados de imágenes digitales de plantas de arroz que fueron segmentadas hacia adelante con el algoritmo K-MEANS en día nublado	133
Tabla 13.	Resultados de imágenes digitales de plantas de arroz que fueron segmentadas hacia atrás con el algoritmo K-MEANS en día nublado.	135
Tabla 14.	Resultados de imágenes digitales de plantas de arroz que fueron segmentadas en posición diagonal a la derecha con el algoritmo K-MEANS en día soleado.	136
Tabla 15.	Resultados de imágenes digitales de plantas de arroz que fueron segmentadas en posición diagonal a la izquierda con el algoritmo K-MEANS en día soleado.	137
Tabla 16.	Resultado de imágenes digitales de plantas de arroz que fueron segmentadas hacia adelante con el algoritmo DBSCAN en día nublado.	149
Tabla 17.	Resultado de imágenes digitales de plantas de arroz que fueron segmentadas hacia atrás con el algoritmo DBSCAN en día nublado.	150
Tabla 18.	Resultado de imágenes digitales de plantas de arroz que fueron segmentadas en posición diagonal hacia la derecha con el algoritmo DBSCAN en día soleado.	152
Tabla 19.	Resultado de imágenes digitales de plantas de arroz que fueron segmentadas en posición diagonal a la izquierda con el algoritmo DBSCAN en día soleado.	153

Resumen

A partir del año 2023 el departamento de Lambayeque disminuyó en producción de arroz el 22,50 %, sin embargo existieron desafíos como plagas de Mosquilla y Sogata que limitaron obtener cosechas adecuadas en los sembríos de arroz, debido a la escasez de agua y climas variables que se dieron en los últimos años, por esta razón el tratamiento y identificación de plantas de arroz a tiempo es importante porque permite a los agricultores buscar una solución adecuada, sin embargo realizarlo de manera manual es complicado, existió una necesidad de obtener técnicas avanzadas que automaticen la identificación de plantas de arroz sanas.

Es por eso que en esta investigación se necesitó segmentación de imágenes digitales de plantas de arroz, que consistió en separar las plantas de arroz sanas de manera automática de otros elementos que no eran plantas de arroz en la imagen. Siendo de utilidad para diferenciar plantas de arroz donde pueden ser utilizados para el estudio de un desarrollo tecnológico en detección de plagas, para poder realizarlo se utilizó un dispositivo de adquisición de imágenes CANON, adquiriendo 50 imágenes en día soleado y 50 imágenes en día nublado, luego se pre procesó convirtiendo a espacio de color CIELab y el otro espacio de color fue HSV, se pre procesó porque las imágenes fueron adquiridas en un ambiente no controlado, luego en CIELab se tomó la dimensión b y en HSV se tomó la dimensión S, porque tuvieron un mejor performance, la dimensión b fue el ingreso para la segmentación con el algoritmo K-MEANS y la dimensión S fue el ingreso para segmentar con el algoritmo DBSCAN, En la segmentación con el algoritmo K-MEANS se obtuvo resultados en Exactitud de 92.25% y una Especificidad de 100%. El algoritmo K-MEANS obtuvo mejores resultados en detección de plantas de arroz porque a pesar que las imágenes tuvieron desenfoques, sombras, barro, agua, logró obtener una mejor Exactitud en píxeles que fueron plantas, confundiendo muy poco con los píxeles que no eran plantas.

Palabras Clave: Algoritmo K-MEANS, Algoritmo DBSCAN, Plantas pixel, Plantas no pixel.

Abstract

As of the year 2023 the department of Lambayeque decreased rice production by 22.50%, however, there were challenges such as Mosquilla and Sogata pests that limited obtaining adequate harvests in rice fields, due to water shortages and variable climates that occurred in recent years, for this reason the treatment and identification of rice plants in time is important because it allows farmers to seek an adequate solution, however doing it manually is complicated, there was a need to obtain advanced techniques that automate the identification of healthy rice plants.

That is why this research required segmentation of digital images of rice plants, which consisted of separating healthy rice plants automatically from other elements that were not rice plants in the image. Being useful to differentiate rice plants where they can be used for the study of a technological development in pest detection, to be able to do this we used a CANON image acquisition device, acquiring 50 images in sunny day and 50 images in cloudy day, then preprocessed by converting to CIELab color space and the other color space was HSV, was preprocessed because the images were acquired in an uncontrolled environment, then in CIELab the b dimension was taken and in HSV the S dimension was taken, because they had a better performance, the b dimension was the input for the segmentation with the K-MEANS algorithm and the S dimension was the input to segment with the DBSCAN algorithm, In the segmentation with the K-MEANS algorithm was obtained results in Accuracy of 92.25% and a Specificity of 100%. The K-MEANS algorithm obtained better results in detection of rice plants because although the images had blurs, shadows, mud, water, managed to obtain a better accuracy in pixels that were plants, confusing very little with the pixels that were not plants.

Keywords: K-MEANS algorithm, DBSCAN Algorithm, Pixel plants, Non-pixel plants.

I. INTRODUCCIÓN

1.1. Realidad Problemática

[1] Frente al instituto nacional de estadística e informática, realizaron estudios donde se mostró decrecimiento de producción de arroz en cascara en el Perú, disminuyendo en 22,50%, debido a las áreas de cultivo de arroz que fueron afectadas por la llegada del fenómeno del niño costero. Siendo Lambayeque uno de los departamentos afectados porque solo logró aportar el 31,3% en producción agrícola.

Debido a las temperaturas climáticas existentes en el valle promovieron en los sembríos de arroz diversidad de plagas como mosquilla[2] y sogata[3], prevaleciendo problemas en la producción de arroz por los cambios climáticos, altas temperaturas, lluvias retrasadas, falta de agua para el regadillo, existiendo así una necesidad en la etapa vegetativa y reproductiva del cultivo de arroz[4] donde se necesita mejorar para poder obtener excelentes resultados en la campaña agrícola.

TABLA 1.

PRODUCCIÓN NACIONAL EN EL PERÚ. SUB SECTOR AGRÍCOLA: JUNIO 2023
(AÑO BASE 2007)

Producto	Ponderación	Variación porcentual	
		2023/2022	
		Junio	Enero - Junio
Arroz Cáscara	8,68	-22,50	-3,07
Aceituna	0,49	-8,52	-13,03
Cacao	0,69	-9,04	-4,84

Café	4,73	-1,87	0,02
Orégano	0,16	-55,75	-2,49
Caña de Azúcar	2,11	-6,84	-2,12
Palma Aceitera	0,32	33,48	11,82
Frijol Caupi Grano Seco	0,14	304,17	233,62
Palta	0,62	9,86	10,29
Papa	6,67	8,29	-14,52

Fuente: [1]

1.1.1 Antecedentes de Estudio

Según los autores [5], para detectar plantas en condiciones normales y plantas en condiciones de estrés se ejecutaron diversos estudios como la que realizó, en el laboratorio de ingeniería forestal situado en Universidad Forestal de Nanjing, en Nanjing, provincia de Jiangsu China donde realizaron "Segmentación automática de hojas de plántulas de álamo superpuestas combinando Mask R-CNN y DBSCAN". Realizado el 24 de agosto del 2020, lograron segmentar hojas de plántulas de álamo en crecimiento bajo condiciones normales y plántulas bajo estrés por metales pesados, se cultivaron varias especies de plántulas de álamo de un crecimiento rápido en macetas, siendo Nanlin895, Nanlin95 y Nanlin3412 cuando crecieron las plántulas y lograron obtener 5 hojas se realizó el experimentos de estrés por metales pesados, de las tres variedades de plántulas de álamo la mitad de ellas fueron cultivadas como grupo de control y la otra mitad fueron estresadas con metales pesados, donde se añadió plomo(Pb) a las muestras de grupo de estrés hasta que lograron un elemento contaminante de la polución urbana.

En total plantaron 60 plántulas de álamo para poder realizar el experimento, la cantidad que se añadió a las plantas estresadas fue de 400 Mg. L-1, las imágenes fueron tomadas el 31 de agosto y 1 de octubre con Cámara Kinect V2, siendo una cámara RGB-D, para el experimento se tomaron 7988 imágenes a color donde estas imágenes obtuvieron profundidades de las hojas de álamo los cuales incluyeron hojas maduras y nuevas hojas, Las imágenes fueron tomadas de diferentes ángulos (horizontal, vertical y oblicuo) fueron capturadas durante el día en horarios de 9:00 - 11:00 am y de 4:00 - 6:00Pm, se incluyeron estados de hojas separadas, adyacentes, superpuestas y sombreadas por otras hojas, las imágenes de referencia se anotaron manualmente utilizando el software VIA(VGG Image Annotator). Demostraron la eficacia del algoritmo de segmentación para detectar hojas individuales en distintos estados, 2000 imágenes RGB-D se adquirieron al azar de todas las imágenes de hojas, se incluyeron 946 pares en condiciones normales y 1054 pares bajo estrés por metales pesados, la cantidad de imágenes del conjunto de entrenamiento validación y prueba es de 1200,400,400, respectivamente.

Las imágenes tuvieron una resolución de 1920 x 1080 pixeles. cada par de imágenes obtuvo una media de 15,4 hojas. El software de recopilación de datos lo desarrollaron en C++ para recoger y guardar la información automáticamente, el programa obtuvo una imagen en color y en profundidad correspondiente para cada captura y guardar cada par de imágenes RGB_D al mismo tiempo, consideraron la consistencia del tamaño de imagen de entrada, donde lograron unificar en 1920 x 1080 pixeles en el conjunto de datos. Este estudio lo ejecutaron en un ordenador con una CPU de 12 núcleos que funcionó a 3,70 Hz con 32 Gb de Memoria Ram, y tuvo un GPU Nvidia GTX1080Ti con un sistema operativo Ubuntu versión 18.04, los

algoritmos de detección de aprendizaje profundo estuvieron basados en Tensorflow y Keras.

Se utilizó el método de calibración de la cámara donde lograron mejorar la eficacia de la utilización de la información de profundidad, Las imágenes de profundidad fueron pre procesadas donde utilizaron el método de codificación de mapa de color jet antes del entrenamiento. posteriormente utilizaron R-CNN para entrenar los datos RGB_D donde lograron fusionar sus características en la estructura FPN para mejorar la fusión de la información RGB_D, además lograron reflejar con precisión la distancia entre dos puntos de la hoja durante el proceso de segmentación, utilizaron densidad de aplicaciones con ruido DBSCAN, estuvo basado en la distancia múltiple para segmentar hojas individuales de las hojas superpuestas en los resultados de detección de Mask R-CNN. Se utilizó Mask R-CNN para detectar y segmentar hojas basados en datos RGB-D los cuales se obtuvieron con la Cámara Kinect V2, luego DBSCAN basado en distancia múltiple lograron agrupar las hojas individuales de las hojas detectadas que se solapaban. Se capturaron las imágenes con una cámara RGB-D Kinect V2 porque las imágenes RGB-D obtuvieron mejores colores y profundidades que las imágenes RGB, para pre procesar la profundidad emplearon el mapa de color Jet, se fusionaron las características de datos RGB-D en FPN mediante el enfoque de mapeo de características en la columna vertebral de Mask RCNN, se entrenaron por separado los datos de color y profundidad, para combinar información de datos de color y profundidad en entrada y profundidad en la columna vertebral de Mask RCNN se fusionaron la profundidad de pirámide de características en la pirámide de características de los datos RGB. Donde cada capa de pirámide de características de profundidad se fusionó con la capa de características de color correspondiente. para realizar una detección precisa determinaron extraer características profundas

donde la red debe ser profunda sin embargo cuando una red neuronal es muy profunda es difícil de entrenar y la precisión se redujo rápidamente. Propusieron un marco de aprendizaje residual que lograron facilitar el entrenamiento de las redes profundas que pudo extraer características bastante profundas de CNN y aumentó en gran medida la eficiencia del aprendizaje de la red. Lograron tener en cuenta la precisión y velocidad de convergencia, también el recurso informático es por ello que se eligió ResNet 50 para el entrenamiento realizado.

Cuando utilizaron el aprendizaje profundo en la detección de hojas, las áreas de estas fueron inexactas al ser detectadas porque existieron hojas vecinas, de fondo o superpuestas, por esta razón utilizaron el algoritmo DBSCAN que logró segmentar las hojas que se superponían unas entre otras, ellos consideraron el algoritmo DBSCAN porque no necesitaba establecer un número de clústers por adelantado y es resistente a los píxeles ruidosos, con el algoritmo DBSCAN lograron separar las hojas superpuestas porque utilizaron la distancia múltiple, donde el algoritmo que se propuso eligió cada punto como punto central y consideró a sus vecinos por un determinado umbral de distancia múltiple, si un conjunto de vecinos obtuvieron al menos puntos MinPts, se considera que pertenece al mismo clúster. También determinaron que un umbral demasiado pequeño condujera fácilmente a una segmentación excesiva. Utilizaron el algoritmo Floyd para construir la matriz de adyacencia.

Para el análisis del rendimiento del algoritmo Mask RCNN realizaron gran número de experimentos con plántulas de álamo de crecimiento normal y plántulas de álamo con estrés por metales pesados donde en las figuras de 8 - 10 mostraron los resultados de las plántulas de "Nanlin895", "Nanlin 95", "Nanlin 3412", La figura 8a le crecieron una gran cantidad de hojas incluyendo las superpuestas, adyacentes y

de ángulos diferentes, en la figura 8b comprobaron que la mayoría de hojas fueron detectadas por Mask R-CNN incluyeron las hojas adyacentes o las que estaban ligeramente cubiertas por otras sin embargo se obtuvo algunas hojas muy ocluidas y algunas demasiado pequeñas los cuales fueron difíciles de detectar, las hojas muy ocluidas su forma fue diferente de las hojas completas, al etiquetar solo las hojas maduras, las hojas nuevas fueron diferentes en forma y color en comparación con las hojas maduras, en la figura 8c y 8d mostraron resultados de las plántulas de álamo en condiciones normales y plántulas de álamo bajo estrés por metales pesados en la figura 8c cambió ya que algunas hojas se volvieron caídas y áreas cercanas a las venas de las hojas se volvieron amarillas en el cual aumentó la dificultad al detectar las hojas en comparación de la figura 8c y 8a mostraron superficies foliares más planas hubo cambio en la superficie de la hoja estresada ya que la mayor parte de la superficie se volvió amarilla, mostraron un estado de caída y rizado, Mask R-CNN logró detectar con precisión las hojas individuales de las hojas estresadas de las plántulas de álamo, en la figura 8d la hoja que estuvo detrás de la hoja marcada en azul no se logró detectar ya que estuvo muy oscurecida y separada en partes, la hoja marcada en azul lo detectaron con precisión sin causar una alteración errónea de la hoja de atrás. a pesar que se determinaron diferentes ángulos y un ligero solapamiento la figura 9 obtuvo resultados precisos tanto en hojas normales como también en hojas estresadas.

Además de las hojas separadas también se detectaron las hojas superpuestas y las hojas cubiertas por ramas en la figura 9a, mientras que en la hoja que fue marcada en verde en la figura 9b apareció un límite nítido porque hubo una hoja detrás de ella, en la figura 9c parte de las hojas tuvieron superficies más retorcidas que en la figura 9a. Sin embargo no lograron detectar una hoja situada a la izquierda en la figura 9c porque fue muy diferente a las demás hojas y pudieron

visualizar una distorsión de hoja más grave además la superficie de la hoja fue casi paralela a la dirección de disparo de la cámara lo que dieron lugar a una pequeña zona visible, la oclusión de las ramas de adelante también interfirieron para que no se detecten las hojas, también señalaron que la hoja de la derecha de la figura 9c presentó un error de detección debido a la interferencia de la hoja superpuesta anterior.

Las hojas de la figura 10 fueron más grandes y las ramas fueron más largas, donde lograron diferenciarse de las figuras 8 y 9 por lo tanto tuvieron una menor probabilidad de superposición en las imágenes sin embargo debido a las hojas más grandes y las ramas más largas, las hojas tendieron a inclinarse más severamente además varias hojas mostraron una grave distorsión ya que la mayoría de las hojas no estuvo orientada directamente hacia la cámara. En la figura 10c incluso una hoja estuvo cubierta por ramas, en tales circunstancias todas las hojas que se mostraron en la figura 10a y 10c fueron detectadas por el algoritmo Mask R-CNN. para la evaluación de hojas superpuestas se seleccionaron 80 imágenes de hojas en el conjunto de prueba, se utilizó la tasa de cobertura para definir el grado de solapamiento que fue definida como la relación de píxeles observados y píxeles reales, la tasa de detección correcta de las hojas con una cobertura inferior a 30% fue de 96,9%, la tasa de detección correcta de las hojas con tasa de cobertura entre el 30% y 50% fue del 91,7% y con tasa de cobertura superior al 50% fue de 70,4%. Para lograr obtener una superficie foliar más completa de las hojas con un índice de cobertura superior al 80% no se etiquetaron durante el proceso de etiquetado.

En la figura 11 se mostraron resultados de detección de hojas con diferentes grados de solapamiento la primera y segunda fila se presentaron hojas normales, y la tercera y cuarta fila presentaron hojas estresadas. Los diferentes grados de solapamiento estuvieron marcados por diferentes colores, el color azul indico grado de cobertura inferior al 30%, el color amarillo tuvo hojas con un grado de cobertura del 30% y el 50%, y el color rojo indicó un grado de cobertura superior al 50%. En la figura 11 lograron detectar la mayoría de las hojas con un índice de cobertura pequeño. Con Mask R-CNN lograron detectar las hojas con un bajo índice de cobertura sin embargo varias hojas estuvieron muy ocluidas por lo que el algoritmo no logro detectarlas debido a la compleja distribución que tuvieron las hojas. En la figura 13 mostraron segmentaciones donde la figura 13a determinó una nube de puntos que fueron segmentados con Mask R-CNN, en la figura 13b-j los puntos verdes representaron la nube de puntos de una hoja que se agrupó con el método propuesto. Donde establecieron un umbral de distancia en 2,5 cm que mantuvieron la sensibilidad de segmentación de la unión de hojas y no pudieron realizar una sobre segmentación, en la figura 13 e-g cuando establecieron el umbral de distancia en 2,2 cm la nube de puntos fue fácil de tener una sobre segmentación en el área de superposición, en la figura 13h-j cuando el umbral de distancia se fijó en 2,8 cm el resultado de la segmentación obtuvo un cambio ligeramente y el algoritmo de segmentación no fue sensible a las partes conectadas.

Con la evaluación de diferentes nubes de puntos segmentados por Mask R-CNN se seleccionaron el umbral de distancia en 2,5 cm y Min Puntos en 180 respectivamente. Para agrupar las hojas superpuestas utilizaron el algoritmo DBSCAN basándose en la distancia múltiple como una métrica para agrupar hojas superpuestas, adoptaron dicha métrica ya que ayudó a lograr obtener características más robustas además se pudo reflejar mejor las similitudes entre dos puntos de las

hojas que estuvieron superpuestas especialmente hojas rizadas debido al estrés por metales pesados. Para determinar su rendimiento se compararon con la métrica de distancia euclidiana y el método clásico de agrupación euclidiana donde obtuvieron resultados en hojas normales utilizando DBSCAN con distancia múltiple con una precisión de 0.985, así también DBSCAN con distancia euclidiana con una precisión de 0.978 y agrupación euclidiana obtuvo una precisión de 0.932, mientras que en las hojas estresadas por metales pesados DBSCAN con distancia múltiple obtuvo una precisión de 0.982, DBSCAN con distancia euclidiana obtuvo una precisión de 0.941 y en cuanto a la agrupación euclidiana obtuvieron una precisión de 0.894, según los resultados obtenidos lograron diagnosticar que la distancia múltiple basado en DBSCAN obtuvo mejores resultados en precisión. para evaluar los resultados de detección con mayor precisión utilizaron P-IoU donde obtuvieron 0,885 en hojas normales y 0,874 en hojas estresadas, con valores de precisión media de 0,897 y 0,888.

Presentaron una metodología novedosa para lograr identificar hojas individuales, utilizaron la combinación de Mask R-CNN y DBSCAN ,el uso de distancia múltiple que lograron medir similitudes entre dos puntos hizo que DBSCAN fuera más preciso al momento de segmentar las hojas superpuestas, uno de los experimentos determinó que la precisión y recuperación de la Mask R-CNN para lograr detectar las hojas en diferentes umbrales, a pesar que obtuvieron hojas bajo estrés por metales pesados y diferentes ángulos de disparo en el conjunto de datos obtuvieron una precisión media de 0.912 y un umbral de 0.7 donde los diferentes estudios demostraron que el uso de esta red y el método de preprocesamiento obtuvo una mayor precisión en el entrenamiento de los datos RGB-D. Por último, se utilizó métrica de evaluación de píxeles para demostrar el rendimiento de evaluación de hojas con mayor claridad para plantas en condiciones normales y plantas estresadas

por metales pesados, obtuvieron valores de P-IoU de 0.885 y 0.874. Donde la mayoría de los falsos píxeles fueron eliminados por el método propuesto. Los resultados demostraron que el método propuesto superó básicamente el problema de detección de hojas superpuestas considerándose una mejora con respecto a métodos anteriores. El estudio que se realizó podría aportar en el fenotipo tridimensional de la hoja y conllevaría a mejorar la precisión del deshojado automático. La principal limitación que se dio en esta metodología fue el lento cálculo en la matriz de distancia múltiple donde es necesario optimizar el tiempo de cálculo, además el entorno para la recogida de los datos RGB-D es el interior, donde las condiciones de escena fueron menos variables que en el campo. El efecto del método realizado se estudiará en entorno de campo en un trabajo futuro. Además, planean trabajar la detección de hojas con aprendizaje profundo basado en nubes de puntos 3D para así lograr detectar directamente las hojas individuales de las nubes de puntos con una velocidad más rápida.

Los autores[6], en la investigación “Detección de plantas y frutos”, como la que realizaron, en la Facultad de Ingeniería de la Universidad Agrícola de China obtuvieron en los últimos meses la “Detección de fruta basada en MASK-RCNN de robots recolectores de fresas en un entorno no estructurado” de varios cultivos en el distrito de Shunyi, Beijing. En junio del 2019, adoptaron el modo de plantación en crestas y se agregaron las variedades Hongyan y Zhangji a otras variedades de maduración temprana. Las imágenes de las fresas se tomaron con una cámara de mano. La imagen se tomó perpendicularmente a la cresta y a 30 cm de la fresa en la cresta, otras variedades de maduración temprana incluyen Hongyan y Zhangji. En la cima, bajo diferente intensidad de luz (soleado y nublado) y diferentes grados de interferencia (como superposición, oclusión y oscilación), se recolectaron 2000 imágenes de fresa en diferentes momentos de la mañana y la tarde, y las imágenes

fueron almacenadas con extensión .jpg La resolución que tuvieron fue de 2352 X 1568 píxeles. Con el fin de reducir el tiempo de cálculo y ejecución del entrenamiento y las pruebas del modelo, las imágenes se ajustaron a una proporción de 640 x 480 píxeles, se utilizaron un total de 1900 imágenes y se manejaron para el entrenamiento del modelo y la optimización de parámetros. MASK R-CNN utiliza un 80% de entrenamiento y un 20% de conjunto de validación. Una vez finalizado el ejercicio, las 100 imágenes restantes se utilizaron para evaluar, probar y verificar el modelo entrenado. Además, utilizaron la herramienta LABELME para realizar anotaciones de máscara y poder comprobar los resultados de la segmentación, se utilizaron para la predicción y se marcan las regiones de frutos de fresa maduros e inmaduros, por lo que las regiones restantes son el fondo de la imagen.

Mask R-CNN constó de tres etapas. La primera fue la red troncal que pudieron extraer mapas de características de la imagen de entrada. La segunda es enviarlos a la Red de propuesta de región (RPN) para lograr la región de interés (Rois), y la tercera, la propuesta de región de mapeo. El ROI generado por la red (RPN) para extraer las características de destino correspondientes al mapa de características compartidas, y luego fueron enviadas a la capa completamente conectada (FC) y al mismo tiempo a la red completamente convolucional (FCN). Este proceso pudo generar puntajes de clasificación, Caja delimitadora y máscara de segmentación.

Los modelos actuales de redes neuronales profundas son alexnet, zf, vgg, googlenet y resnet. Estos tipos de redes se utilizaron porque mostraron muy buena precisión. Las investigaciones utilizaron resnet50 como columna vertebral de la extracción y extracción de características. Las características se basaron en la capa convolucional compartida. La red de bajo nivel pudo extraer particularidades de bajo

nivel (siendo los bordes y ángulos) y particularidades de alto nivel (como la función de red piramidal (FPN)). Estas características describieron las categorías de destino extraídas en niveles superiores. Represento fruta. La Funcion Pyramid Network (FPN) fue una extensión de la red troncal, que es particularmente eficaz para detectar objetos pequeños. Con el número total de píxeles de frutas individuales en la imagen se diseñaron 3 anclajes siendo áreas de 32 x 32, 64 x 64 y 128 x 128. Se usaron los 3 anclajes de relación longitud – ancho (1:1, 1:2 y 2:1) para los anclajes que se generaran. La red propuesta de región (RPN) tenía dos tareas:

1. Utilizaron la capa softmax-loss y entrenaron y clasificaron los anclajes generados.
2. Se utilizaron la capa smoothl1 y se modificaron las coordenadas de los anclajes.

En resumen, la precisión global de frutos maduros e inmaduros es de 95,78% y la segmentación alcanzada fue de 89.85%.

Los autores [7], “En la segmentación de imágenes de hojas de tomate basada en el número de clustering adaptativo del algoritmo K-MEANS”, realizado en la facultad de la conservación del agua e ingeniería civil, universidad agrícola del sur Guangdong, en octubre del 2019, en china, adquirieron hojas de tomate con enfermedades para la realización del estudio propuesto, las muestras fueron tomadas del invernadero de la facultad de conservación del agua e ingeniería civil ubicado en la universidad agrícola del sur de china. Fueron plántulas de tomate las inoculadas, adquiriendo el virus de rizada de la hoja que fue de color amarilla y virus del mosaico que también fueron infectados en los tomates, siendo estudiados en el instituto de investigación de protección de plantas de la academia de ciencias agrícolas de guangdong y luego fueron trasplantadas al invernadero, las plantas de tomate fueron criados después de siete días de aclimatación, durante los días de

crecimiento se roció manualmente la solución nutritiva Hoagland donde se aplicó 150 ml cada 2 días, para rosecarlo utilizaron una lata para la aspersión, así se aseguraron que las plantas sobrevivieran con los suficientes nutrientes. Las muestras de imágenes de tomates durante el crecimiento fueron capturadas con una cámara digital SONY DSC-HX400, siendo configurada para ajustar automáticamente la distancia focal y la apertura, el balance de blancos fue automático, el apagado del flash también fue automático. Las imágenes tuvieron una resolución de 5184 x 3888 pixeles. Donde fueron capturadas independientemente de los factores meteorológicos.

Las imágenes de fondo de papel blanco se utilizaron como datos experimentales para los algoritmos y las imágenes de fondo natural fueron empleados para verificar la efectividad del algoritmo expuesto.

Presentaron el índice de Davies- Bouldin(DB) que fue una función en relación de la suma de la dispersión dentro del clúster a la separación entre clúster, el objetivo de DB fue minimizar esta medida ya que se quería reducir la dispersión dentro del clúster y maximizar la separación entre clústeres. Luego mencionaron también el índice Calinski-Harabasz (CH) lo utilizaron como medida de interés interno del clúster, que calificaron a los mismos, calificando al clúster creados por cada individuo. y por último el coeficiente de silueta fue una medida de similitud de un objeto con su mismo clúster (Cohesión) en comparación con otros clústeres (Separación) la silueta obtuvo una oscilación entre - 1 y +1 cuando fue expresado un valor alto significó que el objeto estuvo bien cohesionado con su propio clúster y mal cohesionado con los clústeres vecinos, si la mayoría de los objetos tuvieron un valor alto significó que la configuración del clúster es adecuada. Propusieron un algoritmo

adaptativo denominado K-MEANS para segmentaciones de imágenes de hojas en enfermedades de tomates. En el algoritmo mantuvo en cuenta datos de entrada para la agrupación, el número de clases K y por último los centroides iniciales. En los datos de entrada para la segmentación la imagen RGB lo transformaron a un espacio de color HSI y en el espacio de color Lab donde los componentes (Slab) se extrajeron como parámetros de características es decir datos cuatridimensionales(S,I,a,b) los cuales fueron sometidos al experimento de segmentación. los resultados de la agrupación de las imágenes se mostraron en la figura 2b donde la imagen de la hoja que solo se necesitó para segmentar fue sobre segmentada siendo adecuada para la investigación porque tuvo la necesidad extraer las venas de hojas, para superar la sobre segmentación se seleccionaron los parámetros bidimensionales (SI) para las pruebas. Los resultados se mostraron en la figura 2c donde el efecto de la segmentación de la hoja fue casi impecable, los datos de entrada coincidentes debieron ser segmentados en un objeto variable, a través del experimento se seleccionaron los componentes (Slab) de la imagen que tuvo un fondo natural como los datos de entrada de clustering.

En las imágenes que tenían el fondo de papel blanco, los datos de entrada fueron los componentes SI. descubrieron que la redundancia del tiempo era demasiado grave cuando se realizó la ejecución de la imagen original de 3024 x 4032 pixeles es por eso que se redujo a 303 x 404 pixeles donde lograron mejorar la eficiencia del algoritmo sin afectar los resultados de agrupación. Para obtener mejores segmentaciones de diferentes imágenes debieron utilizar diferentes números de clúster para agrupar, en la figura 4a cuando el número de agrupación se estableció en 2, en el resultado de la segmentación lograron demostrar que el fondo del papel blanco fue segmentado en toda la imagen pero cuando el número del clustering se estableció en 3 la sombra del borde de la hoja se agruparon en una

sola categoría por lo tanto el número de clustering para esta imagen debió ser 3 sin embargo para la figura 4 b fue evidente que el papel en blanco se dividió en dos regiones cuando el número de clustering se fijó en 3, esto no tuvo sentido por lo que el número óptimo de clusters para esta imagen debió ser 2. en el experimento demostraron que los diferentes números de clustering obtuvieron distintos resultados de segmentación.

Se estimó manualmente el número óptimo de clúster de 200 imágenes de fondo de papel blanco, establecieron (S,I) como dato de entrada y limitaron el número de clusters al rango de búsqueda(2,3) calcularon el índice DB, el índice CH, y el coeficiente de silueta de 200 imágenes respectivamente y obtuvieron el número de clustering calculado por cada índice, se compararon el número de clúster calculado con el número de clúster óptimo para calcular la precisión. El experimento lo llevaron a cabo utilizando una CPU Intel(R) Core (TM) i7-3770, con 8 GB de memoria RAM y la plataforma de desarrollo MATLAB R2018b el cual fue ejecutado en el sistema operativo Windows 10.

En los resultados cualitativos el método propuesto lo compararon con otros tres algoritmos de segmentación siendo DBSCAN, Mean shift y ExG -ExR, en la segmentación de plántulas de tomate maduras e imágenes que tuvieron objetos de color similar, en la figura N° 7 se mostraron los resultados donde el algoritmo propuesto pudo obtener la mejor segmentación, mientras que el algoritmo DBSCAN y Mean Shift fueron más fáciles en la sub segmentación o sobre segmentación en las hojas porque sus parámetros de entrada fueron más sensibles, pudieron observar que en el algoritmo DBSCAN se excluyeron los tayos verdes en la figura 7a pero incluyeron el fondo en la figura 7b y con Mean Shift lograron poder segmentar parte

del fondo de la imagen como la cubeta blanca que fue mostrado en la figura 7a y 7c, también pudieron observar que no se realizó la eliminación de la cinta roja que había en la figura 7a, en la figura 7b la segmentación con ExG-ExR no determinaron realizarlo como es debido, pero en la figura 7a y 7c el resultado fue mucho mejor.

La diferencia entre las dos imágenes fue el color de las hojas. Aunque en el ExG-ExR obtuvieron un indicador efectivo y sencillo para la segmentación de plantas verdes porque estuvo perfectamente restringido al color verde, y no es sensible a los objetos no verdes, incluso se obtuvo segmentaciones de tallos que eran verdes en las plantas de tomate como se pudo apreciar en la figura 7a, los resultados obtenidos en este trabajo reflejaron que el algoritmo mejorado K-MEANS obtuvo un efecto de segmentación perfecto al lograrse establecer el número óptimo de agrupación con la ayuda del cálculo de índice de validez de cada imagen. Mientras que en los resultados cuantitativos se seleccionaron 20 imágenes al azar entre ellos 800 imágenes de fondo natural, se tuvieron en cuenta para evaluar estos cuatro métodos, los resultados se mostraron en la Tabla 2 donde el algoritmo propuesto realizado obtuvo el procesamiento de la imagen y los cálculos de índice de validez, el tiempo de ejecución fue ligeramente más lento que el método ExG-ExR pero fue mucho más rápido que los otros dos algoritmos de agrupación (DBSCAN, Mean Shift).

En los resultados del algoritmo propuesto determinaron que los valores promedio de F1 fue de 0,982, y la Entropía fue de 0,118, siendo estos los mejores resultados en comparación con los tres métodos de segmentación. Para la comparación con el algoritmo tradicional de K-MEANS y el algoritmo propuesto en la tabla 2 se probaron con 12 imágenes las cuales fueron N° 3,4,5,6,7,8,10,12,13,14,16,19 donde obtuvieron indicadores de evaluación similares

porque el número óptimo de clusters para estas imágenes fue 2. Demostraron que el cálculo de índice de validez era consistente con el número de clustering manual. También debemos mencionar que algoritmo propuesto tuvo incorporado el procesamiento morfológico por eso cuando el número de clusters era 2 tuvo una mejor precisión en la segmentación de imágenes.

Para otras imágenes que también se realizaron pruebas siendo N° 1,2,9,11,15,17,18,20, donde la puntuación(F1) del algoritmo propuesto fue muy superior a la del algoritmo tradicional K-MEANS ya que la razón principal fue que la agrupación de estas imágenes de prueba fue 3. El algoritmo tradicional de K-MEANS asignó artificialmente el número de clúster de todas las imágenes a 2, tuvieron en cuenta que las imágenes que tenían ruido de fondo complejo como sombras, suelos, cubos blancos, etc. la agrupación solo distinguió entre las dos partes cuyas características fueron categorías drásticamente diferentes, se tuvo el riesgo de las hojas y el fondo de la imagen se pudieran clasificar erróneamente como un solo clúster.

Cuando existió una gran diferencia de color de imagen, el algoritmo tradicional K-MEANS sobre segmento o subsegmento la imagen de fondo de la naturaleza, para realizar la verificación en el método que propusieron también realizaron la evaluación con imágenes públicas de hojas con enfermedades de tomate, obtuvieron dichas imágenes de la base de datos Bugwood. Recogieron 30 imágenes de fondo de la enfermedad del tomate. Solo hubo una hoja principal en cada una de estas imágenes y segmentaron la hoja manualmente como groundtruth, donde obtuvieron el promedio F1 0,765 y la entropía de 0,643 respectivamente. Al observar los resultados dedujeron que el obstáculo fue la mancha enferma porque

las manchas enfermas se volvieron marrones aún más cuando las hojas de tomate maduraron, dado que el fondo fue también plantas verdes, en ese caso las manchas marrones fueron más fáciles de agrupar en un clúster esto conlleva a que la hoja y el fondo se agruparon.

Las pruebas de las imágenes demostraron que: En las imágenes de fondo natural, los componentes a,b y S,I, de los espacios de color Lab y HIS fueron seleccionados como datos de entrada para el clustering; luego las comparaciones que se realizaron tanto en el índice CH y el índice Silhouette, el índice DB tuvo la mayor precisión y el menor tiempo de cálculo. El algoritmo propuesto obtuvo una mejora en precisión de segmentación en comparación con el algoritmo tradicional K-MEANS. El método propuesto requirió de más computación para calcular el índice de validez y así demostrar la viabilidad del método, se realizaron pruebas utilizando una base de datos pequeña, se requirió más trabajo para desplegar completamente el algoritmo propuesto teniendo en cuenta realizarlo en un futuro en aplicaciones con procesamiento de imágenes a gran escala refiriéndose principalmente a imágenes que contienen objetos de colores similares.

Los autores [8] , en "Estimación de alojamiento del arroz usando índices obtenidos de drones visibles e imágenes infrarrojas térmicas", obtuvieron imágenes infrarrojas visibles o térmicas en China el 15 de abril de 2018 (RGB y /o TIF) usaron vehículo aéreo no tripulado (UAV) y adquirieron las imágenes mixtas recolectadas en el campo, y fue necesario usar la técnica enjambres de partículas PSO y máquinas de vectores de soporte SVM. El proceso desarrollado en cultivos de arroz, y capacidad para utilizar diferentes fertilizantes nitrogenados para la alimentación y el procesamiento, se pudo distinguir que la temperatura del arroz es cambiante

durante el día, por lo que la temperatura del arroz de embarque y del arroz en invernadero es más alta. En esta investigación desarrollada los resultados demostraron que las características de color y textura de las plantaciones de arroz son diferentes. En el modelo propuesto lograron obtener falsos positivos y falsos negativos menores al 10% y la tasa de alojamiento estimada es del 0,9% los resultados muestran que la combinación de imágenes infrarrojas visibles e imágenes térmicas ha logrado una precisión significativa en la identificación de alojamiento de arroz.

Los autores [9], señalaron que el 15 de septiembre de 2018 en China, donde realizaron "Observación automática de la etapa de partida del arroz mediante el método de detección de espigas del arroz basado en cascada de múltiples clasificadores", descubrieron que existen problemas para recolectar semillas de arroz en el campo. En términos de cultivo, son independientes la cámara Olympus se utilizó como dispositivo de observación para recolectar imágenes, y se adquirieron imágenes de 3648 x 2736 píxeles. Estas imágenes se transmitieron a una computadora remota mediante transmisión de datos 3g. Por lo tanto, utilizaron clasificador SVM con función de color, más histograma de gradiente para eliminar manchas amarillas en las hojas, también emplearon una red neuronal convolucional para reducir los falsos positivos y produjeron resultados similares a las técnicas de observación manuales o convencionales, por lo que, dado que se espera que se implemente el clasificador SVM, esta técnica se pudo utilizar para reemplazar las observaciones manuales. Se detectaron 93,2% de los picos de arroz.

[10] En tal sentido, llevaron a cabo la “Segmentación de plántulas de arroz utilizando el espacio de color YCrCb y el método OTSU mejorado” en la Escuela de Ingeniería de la Universidad Agrícola de Anhui el 19 de noviembre de 2018. Según el estudio de subdivisión, cada variedad tuvo dos variedades, a saber, Fengliangyou 4, Huiliang You 898, tipo de arroz híbrido y Zhendao 18, tipo de arroz japónica convencional Shangnongjing 2, el espacio entre filas fue de aproximadamente 20-30 cm, la densidad de plantas fue de 27×10^4 - 30×10^4 hectáreas / ha por hoyo, cada hoyo 4-5 plántulas, el área total de siembra de arroz fue de 0,15 hectáreas. Para obtener las imágenes utilizaron una cámara Hikvision mv-ca050-20um / c (Hangzhou Hikvision Digital Technology, Hangzhou, China), tenía un puerto usb, la distancia focal del lente fue de 25 mm y el dispositivo estuvo instalado a 1,8 metros sobre el suelo. 50 imágenes se obtuvieron en días soleados y nublados, se adquirieron a la computadora mediante un cable usb, y la extensión de la imagen fue .jpg, siendo una imagen en color de 24 bits con dimensiones de color rgb, la resolución de la representación original fue de 2560 X 1920 píxeles. Esto fue necesario para reducir el tamaño a 320 X 237 píxeles. Porque fueron computacionalmente costosos.

El formato RGB es utilizado en imágenes tomadas por la mayoría de cámaras digitales, pero debido a su alta correlación, no es adecuado para realizar la segmentación de imágenes, por lo que se debe convertir al espacio de color YCrCb. De esta forma fueron resueltos el brillo y el croma Junto, la introducción del componente Cg represento la diferencia de color y el índice de color es 2CgCrCb. Determinaron la mejor precisión, probaron utilizando 2CgCrCb, NVE (Neighborhood Valley Focus) y técnica OTSU mejorado, donde el 2CgCrCb basado en el espacio de color logró el mejor rendimiento de 0.003 segundos., En comparación con otras técnicas, debido a su robustez a los cambios de luz, la técnica NVE (Neighborhood Valley Focus) perdió muchos píxeles de arroz en el borde de las hojas de las plantas

y logró un rendimiento de 0,027 segundos. OTSU segmentó muchos píxeles de fondo en píxeles de plántulas de arroz y logró un rendimiento de 0,005 segundos.

Los autores [11], en la “Segmentación e identificación de hojas enfermas mediante la fusión de superpíxeles, K-MEANS y PHOG”. Realizado el 22 de noviembre de 2017 en China, utilizaron la técnica SLIC (clustering iterativo lineal simple) y K-MEANS para realizar un estudio de segmentación para detectar enfermedades en hojas de pepino y hojas de manzana, para comparar, utilizaron otras técnicas de segmentación, como KMSNN, SIFT (Transformación de características invariantes de escala) e IRT (Artículo de teoría de respuesta). Utilizo la técnica de segmentación K-MEANS y detectaron la parte enferma de la hoja, dejando a un lado la parte no afectada y el área de fondo (como el suelo), y resaltaron esta última en color negro. La agrupación K-MEANS determinaron k centroides optimizando la adición de las distancias de los píxeles de datos de los centroides correspondientes.

Sin embargo, fue difícil determinar el número de centroides y clústeres iniciales. Por lo tanto, el uso de esta técnica SLIC pudo ayudar a la segmentación de la imagen en color porque ha sido utilizado en varios algoritmos de segmentación. Utilizaron dos bases de datos, se obtuvieron 150 imágenes de hojas de manzana enfermas en cada base de datos, de manera que se pudo subdividir las áreas afectadas, como mancha foliar, mosaico, moho, y el resultado promedio es 75,87% para KMSNN, SIFT con el 80,63%, TRI con el 80,82%, la técnica que utilizaron logró un promedio del 85,64% de las tres enfermedades diagnosticadas. De igual manera, en las hojas de pepino se subdividieron las áreas enfermas de plagas y enfermedades, tales como: antracnosis, mancha angular, moho polvoriento, el efecto

promedio de KMSNN es 79.65%, el resultado promedio de SIFT es 82.98% y el resultado promedio de IRT con un 84,08%, la técnica que utilizamos rindió un promedio de 87,55%.

La enfermedad se identificó mediante el clasificador C-SVM, utilizaron técnica PHOG (pirámide de histograma de gradiente) para extraer características. Las hojas de las plantas de pepino y manzana se adquirieron en la ciudad de Yangling, provincia de Shanxi, China, y la tecnología fue implementada en Matlab 7.6 en una computadora personal IntelXeon X3430.

Los autores [12], en el año 2018 en la Fundación Nacional de Ciencias Naturales de China llevó a cabo un "Suavizado de imagen de destino de barco multimodal basado en el cambio medio adaptativo", obtuvo un conjunto de imágenes proporcionadas por investigadores y lograron obtener 310 imágenes reales imágenes de objetivos de barcos por infrarrojos (IR) y 500 imágenes visuales reales con diferentes objetivos y fondos de barcos, las imágenes infrarrojas fueron captadas por cámaras fijas en barcos con diferentes fondos, estas imágenes se utilizaron permanentemente para la vigilancia en el campo marino, algunas de estas imágenes se adquirieron de sitios web de acceso público (<http://www.cnss.com>). El tamaño de la imagen fue de 320 x 256 píxeles y 720 x 576 píxeles. Para imágenes infrarrojas de barcos, también tuvieron una dimensión de 320 x 256 píxeles y 1025 x 768 píxeles. La técnica sugerida lo realizaron usando parámetros fijos.

Este trabajo fue implementado utilizando tecnología Matlab R2012b en una computadora con características de hardware de procesador Intel (R) Core (TM) I5-2400, unidad central de proceso(CPU) de 3.10 ghz y memoria ram de 4 gb. pudo filtrar cambios promedio y parámetros de ancho de banda de rango Hr y el

parámetro de ancho de banda espacial H_s fue muy importante para el rendimiento (especialmente H_r). Para suavizar eficazmente el fondo y preservar el objetivo, fue importante lograr distinguir el objetivo del fondo y elegir un ancho de banda diferente. En la investigación, desarrollaron un método de filtrado de desplazamiento medio adaptativo para suavizar la imagen de los objetivos de las naves espaciales infrarrojas. Este método suavizó eficazmente el fondo y mantuvo la estructura detallada de la nave espacial bajo infrarrojos. En este estudio, utilizaron una técnica para suavizar la imagen multimodal. Primero se interesaron por las imágenes infrarrojas y visuales. Las imágenes objetivo de barcos con el fondo del mar y el cielo se consideraron principalmente dos razones de esta expansión.

Uno de ellos fue considerado en aplicaciones prácticas, por ejemplo, para identificar objetivos de barcos. Ambas técnicas se utilizaron comúnmente y siempre es conveniente eliminar la imagen para determinar la realización de la calidad de la misma. El manejo de los dos métodos es el mismo, lo cual fue beneficioso porque redujo el tedioso trabajo de diseño técnico y configuración de hardware. Otra razón es que la técnica propuesta no limitó la modalidad de imagen. Se observaron que, aunque diferentes imágenes tienen diferentes fondos y objetivos, todavía tienen algo en común. Generalmente, hay tres áreas: cielo, nave y fondo marino, y estas tres áreas tuvieron diferentes características de color. Al observar las imágenes infrarrojas y visibles del objetivo de la nave espacial, se encontraron que tuvieron las siguientes tres características en común: Característica 1: El objetivo de la nave es un área conectada con mayor prominencia visual y mayor contraste local que el fondo circundante. Característica 2: El área objetivo de la nave espacial logró alcanzar una desviación estándar relativamente grande y un gradiente cerca del límite. Característica 3: Para una gran cantidad de imágenes, el área del lecho marino obtuvo una amplitud de gradiente relativamente grande, pero la desviación

estándar fue pequeña, porque la mayor parte del volumen de onda lo distribuyeron de alguna manera.

Se observaron que este método es aplicable tanto a imágenes infrarrojas como a imágenes visuales con diferentes objetivos y fondos, y se compararon con el mapa de bordes de la imagen principal y el mapa de bordes de la imagen que se logró obtener. Las Figuras 6 (E) y 7 (E), la imagen se visualizó bien, la mayor parte del fondo de la imagen se pudo eliminar y el área objetivo, es decir, el objetivo del área de la nave, se logró conservar bien. En cuanto a la comparación visual, para tener la efectividad de la técnica propuesta, se comparó con otras cuatro técnicas, a saber, filtrado de difusión anisotrópica, filtrado bilateral, filtrado de propagación y filtrado de desplazamiento medio con un rango fijo de ancho de banda. Se eligieron estas técnicas porque eliminaron diferentes tipos de ruido en los bordes de la imagen capturada en aplicaciones prácticas. En aplicaciones prácticas, el ruido es más complejo y es difícil establecer un modelo de ruido preciso. Fue necesario ajustar los parámetros relevantes para poder comparar los razonables.

Se iteraron el número de iteraciones y el umbral del módulo de gradiente, y utilizaron un filtro de difusión anisotrópico, el tamaño medio de la ventana del filtro y la desviación estándar; utilizaron el radio y la desviación de la ventana del filtro bilateral para ampliar el estándar del filtro y el rango medio del ancho de banda del filtro. En la Figura 8 (E-1), se observaron que, en un rango fijo de ancho de banda, el desplazamiento promedio en la técnica de filtrado ha logrado eliminar de manera efectiva las perturbaciones oceánicas graves, pero también se redujo el límite del objetivo de la nave. En un rango fijo de ancho de banda, El área objetivo del barco debió estar bien protegida, pero no se pudo eliminar bien en el fondo. Por lo tanto,

fue difícil elegir un ancho de banda adecuado para la tecnología de filtrado de desplazamiento medio de ancho de banda fijo. En nuestro experimento, determinaron tratar de elegir un rango de ancho de banda que pueda equilibrar el suavizado de fondo y la preservación de la región. La Figura 8 (F-1) llegaron a mostrar que, a través de la técnica propuesta, podemos eliminar de manera efectiva los objetos oceánicos complejos que existen alrededor del área objetivo de la nave espacial, y también retener el área mencionada anteriormente. La técnica que propusieron en esta investigación es mejor el resultado. Los resultados de la comparación mostraron que la técnica propuesta tiene un rendimiento superior, con valores promedio más altos de LCP-SI, EBMC y SMPI, mientras que los valores promedio de ME y RAE son más bajos. Es muy útil para mejorar el rendimiento de varias aplicaciones, como la separación de imágenes de objetivos de barcos y la localización de áreas salientes.

Los autores [13], en la "Red UNET mejorada para la segmentación de nódulos pulmonares ", realizado el 23 de agosto del 2018 en la Escuela de Ciencias e Ingeniería de la Información de Shenyang, Universidad del Noreste, 110819, China. La segmentación de imágenes de nódulos pulmonares en imágenes de entrada de tipo TC se realizaron mediante tomografía computarizada (Las imágenes de TC) son muy pequeñas y se confunden fácilmente con otros tejidos, y el ojo humano no las reconoce, por lo que tienen formato dicom o mhd (tomografía computarizada) para la investigación. UNET mejorada, incluido el conjunto de datos de reconstrucción, la capa convolucional, la capa de agrupación y el muestreo de capa, esta técnica se basa en el aprendizaje profundo, primero procesó previamente la imagen TC y luego usa métodos morfológicos para segmentar imágenes de parénquima pulmonar, dividieron el parénquima pulmonar segmentado en cubos de 64×64 píxeles, y finalmente, utilizaron la técnica UNET mejorada para el

entrenamiento de imágenes, y así lograron obtener la segmentación de los nódulos pulmonares.

Para resolver las muestras desequilibradas en el proceso de entrenamiento de segmentación de nódulos pulmonares, reconstruyeron un conjunto de datos de segmentación de nódulos pulmonares equilibrados a partir de las muestras basándose en las imágenes de corte etiquetadas. En el nuevo conjunto de datos lograron entrenar con éxito un modelo de red más preciso. Luego, agregaron la red residual a la red no establecida. La red residual fue similar al método general y pudo mejorar nuestra predicción final, Al formar el módulo residual, generaron múltiples salidas de capa convolucional y entradas iniciales para extraer las características de la imagen y reducir los parámetros de entrenamiento. Lo último fue lograr agregar una capa de normalización por lotes a la red UNET.

Al usar sgd, La función de activación es minibatch. Para este grupo de pruebas, se seleccionaron 975 nódulos pulmonares como grupo de entrenamiento, 130 nódulos pulmonares se usaron como grupo de verificación y 140 nódulos pulmonares se usaron como grupo de prueba, y se compararon con el nivel de grupo y cortes de gráfico respectivamente. El conjunto de datos utilizado fue luna 2016, en imágenes tomográficas de los pulmones en formato mhi. Tomaron una muestra de 1,000 pacientes y se marcaron un total de 1,245 nódulos pulmonares, todos los cuales fueron asignados al azar durante el entrenamiento. Los nódulos midieron de 5 a 10 mm y también nódulos de 10 a 30 mm, lo que represento el 50% respectivamente. Cada imagen de nódulo pulmonar se le realizo múltiples cortes de 512 x 512 píxeles. Una imagen TC con un grosor de 1,25 mm suele contener más de 200 cortes y cada imagen tenía un tamaño de 512 x 512 píxeles.

Los resultados fueron probados en una computadora teniendo características de hardware como: procesador Intel Core I5-3570 cpu @ 3.4 hz , tarjeta grafica gtx660 / pci e / sse 2, 8 gb de memoria ram, disco duro de 500 gb. Los resultados obtenidos en la segmentación mostraron que el rendimiento obtenido por UNET fue de 0,713 seg. , el rendimiento del conjunto horizontal fue de 0,601seg. , el rendimiento del corte de gráfico fue de 0,695 seg. y el rendimiento del método propuesto fue de 0,736 seg. [14]En efecto en la aplicación "EasyPCC: Conjunto de datos de referencia y herramientas para la medición de alto rendimiento de la relación de la cobertura del dosel de la planta bajo condiciones de campo" se desarrolló en el Instituto de Servicios de Ecosistemas Agrícolas de la Universidad de Tokio el 4 de abril de 2017. Obtuvieron segmentación de la cubierta vegetal de PCCR en condiciones ambientales o de luz natural. Desarrollaron un paquete R llamado EasyPCC en Matlab basado en el árbol de decisión DTSM para extraer áreas de vegetación, se sembraron semillas de la variedad de arroz japonés hamamelis el 26 de abril del 2013 y trasplantaron las plántulas el 31 de mayo de 2013 , utilizaron el sistema de monitoreo in situ para obtener imágenes, instalaron una cámara Canon EOS Kiss X5 con lente EF-S18-55mm en un campo de arroz a una altura de 2 metros, la imagen fue tomada durante 1 hora y transferida a un servidor web gratuito (www.flickr.com) usaron una red 3G. EasyPCC se distribuyó junto con los datos de entrenamiento, como imágenes de trigo, arroz, sorgo, algodón, soja y caña de azúcar, para que posteriormente solo se pueda segmentar la vegetación o áreas verdes. Los datos de entrenamiento incluyeron nueve características de color (R, G, B, H, S, V, L *, A *, B *) definidas en tres espacios de color (RGB, HSV, CIELAB) para realizar el parche de datos de entrenamiento.

El resultado que obtuvieron de la imagen "2013/08/15/15: 02 (pm)" y otra imagen capturada 1 hora antes "2013/08/15/14:02 (pm)", observando en dos imágenes las sombras adquiridas que condujeron a una grave subestimación de la cobertura vegetal. La imagen "2013/08/15/ 15: 02 (pm)" perdió muchos píxeles de vegetación, debido a que esta área tenía un área oscura más grande y el error en esta área es aceptable porque el entrenamiento inicial se realizó en la imagen con píxeles de vegetación sombreados, de modo que la sombra y / o las partes oscuras se clasificaron como parte del fondo de la imagen., Para superar esta debilidad, se agregaron 12.000 píxeles seleccionados del área oscura de la vegetación de la imagen y el modelo se entrenó con nuevos datos de entrenamiento. Con el nuevo modelo de entrenamiento, se mejoraron los resultados de segmentación de imágenes. La imagen "2013/08/15/15:02(pm)" logro aumentar el valor estimado del 58% al 79%, mientras que el resultado de la imagen "2013/08/15/14:02(pm)" casi sin área de sombra obtuvo similares resultados de precisión logrando solo tener del 72% al 76%, el valor R2 entre el valor derivado de DTSM y el valor real alcanzó 0,99.

Las pruebas realizadas en EasyPCC incluyeron el análisis de su rendimiento y la eficacia de los datos de entrenamiento para la precisión de PCCr. El resultado de la imagen segmentada lo lograron publicar en formato "*.csv" para que pueda ser leído por otro software (como Excel o R). Esta herramienta fue muy poderosa en imágenes tomadas en exteriores, ya que pudo tardar unos 30 segundos en imágenes afectadas por el viento y la vegetación aplastada por daños desconocidos. En términos de tiempo de procesamiento de imágenes de 6 megabits, tardo unos 30 segundos. Utilizando una CPU Intel I7 con 16 GB de memoria ram, por lo que existieron muchos tipos de sistemas de monitoreo in situ para el procesamiento simultáneo de imágenes, como servidores in situ para detectar una gran cantidad de plantas, vehículos móviles y drones.

Los autores [15], utilizaron "Segmentación Jerárquica de imagen no supervisada a través de la maximización difusa de la entropía ", realizado el 6 de marzo de 2017, en China, se enfrentó al problema del método de entropía de partición difusa c existente, que normalmente tenía dos limitaciones, a saber, el número de particiones C fue ajustado manualmente para diferentes entradas y métodos de modo que solo pudo procesar imágenes en escala de grises, y utilizaron un algoritmo de segmentación multinivel no supervisado (algoritmo jerárquico). El paso principal de nuestro algoritmo fue un operador de segmentación de dos niveles, que utilizo el corte binario de gráficos para maximizar el valor máximo difuso y la suavidad de segmentación de las dos particiones., El algoritmo primero selecciono el canal de color que mejor dividió la imagen en dos etiquetas, y luego seleccionaron iterativamente el canal que dividió cada etiqueta en dos canales hasta que los puntos se encuentran.

A continuación, demostraron que el esquema de segmentación jerárquica propuesto puede segmentar efectivamente imágenes en escala de grises e imágenes en color. Los resultados que fueron mostrados por la base de datos de Berkeley son comparables a los métodos de segmentación multiescala más avanzados, pero tuvieron las ventajas de no estar supervisados, fueron eficientes y fáciles de implementar. Concluyeron que el método propuesto logro ser robusto y logro segmentar automáticamente diferentes tipos de imágenes. El tiempo de procesamiento tuvo un rendimiento para imágenes grises de 5,3 segundos y el tiempo de procesamiento para imágenes en color fue de 10,8 segundos. La limitación del enfoque propuesto en la segmentación de imágenes es que utilizaron solo información de color, conduciendo así a resultados subóptimos cuando los objetos y

el fondo tienen colores similares. Tuvieron señales de alto nivel, como la textura y bordes, mantuvieron un enfoque no supervisado, no es tarea fácil. Es por ello los autores recomendaron realizar más investigación futura en este ámbito.

Los autores [16], estuvieron “Contando manzanas y naranjas con aprendizaje profundo: un enfoque basado en datos”, recontaron frutas en ambientes no estructurados, siendo un gran reto debido a variaciones de apariencia, cambios de iluminación, oclusiones de follaje, y frutas vecinas existentes en los campos. Se realizó un enfoque basado en aprendizaje profundo para diagnosticar imágenes de entrada hasta obtener un recuento total de frutas en las plantaciones, en la tubería utilizaron plataforma personalizada de crowd-sourcing para realizar el etiquetado robusto. Mediante un detector de manchas basada en una red convolucional lograron una IoU media de 0.813 en las naranjas y 0.838 en las manzanas, y obtuvo regiones similares en las imágenes, usando el algoritmo de conteo, esta es otra red convolucional, y finalmente en el modelo de regresión lineal mostraron el conteo de frutas hasta el cálculo final.

Se utilizaron dos conjuntos de datos diferentes para analizar la tubería: la precisión del recuento de naranjas se adquirió por la noche siendo de 0,968 y la precisión del recuento de manzanas verdes fue de 0,913. Incluso en frutos ocultos que son un desafío para los etiquetadores humanos, en la técnica implementada lograron obtener un funcionamiento perfectamente. Los investigadores afirmaron que las técnicas que utilizaron no solo se pueden aplicar a la agricultura de precisión, si no también pueden ser aplicados a la identificación y el recuento de plantas, la detección de enfermedades e incluso el recuento de células microbianas.

El autor [17], en su tesis “Procesamiento de imágenes digitales utilizando descriptores de color para la identificación y clasificación de diversos tipos de semilla de Quinoa” se utilizó la librería opencv, realizo el uso del entorno de desarrollo integrado QT y Netbeans, donde realizaron adquisiciones de imágenes bajo medidas controladas, este es un entorno en el que se recomendó encarecidamente utilizar imágenes capturadas para evitar la distorsión de la misma y mantener las sombras. Capturaron imágenes iniciales en condiciones ideales, e imágenes que tengan defectos al usar el clasificador para el entrenamiento. aplicaron filtros para corregir los colores en la imagen, para que las características de cada semilla de quinua puedan ser diferentes, entre ellas: MedianBlur, guisarBlur, desenfoque o promedio simple, contraste, corrosión y expansión, eligiendo los mejores resultados. Utilizaron el algoritmo de segmentación OTSU, Binarizacion, EM, porque el propósito fue mejorar el efecto del entrenamiento. Del mismo modo, utilizaron los descriptores de color Haar y Svm para la clasificación, histogramas de imágenes para la clasificación y Perceptron para el análisis de píxeles. Llegaron a la conclusión que, al utilizar los algoritmos seleccionados para el análisis, les permitió distinguir imágenes por sus colores, por lo que también se pueden aprender los descriptores utilizados para obtener una mejor clasificación.

Los autores [18], realizaron “Detección de algodón en el campo a través de la segmentación de imagen semántica basada en las regiones”, el 8 de julio de 2016 en China, utilizaron la cámara digital Olympus E-450 para capturar imágenes, tenía una distancia focal de 16 mm, se colocaron en el campo de cultivo para ser estudiados y disparar Sobre el soporte de la imagen, una cámara de 8 metros cuadrados se ubica en la parte superior de la planta a una altura de 5 m, y la otra cámara se ubicó en la parte inferior a una altura de 0,3 m. Ambas brindaron imágenes de plantas de algodón. La técnica que se utilizo fue CIE $L^* a^* b^*$ y SLIC

para espacio de color y también la técnica DBSCAN para ruido y movimiento de las imágenes adquiridas, descubrieron formas o grupos que las plantas pueden doblar, líneas ovaladas, por lo que se pudo utilizar la métrica wasserstein basada en el estándar euclidiano para obtener áreas no supervisadas y supervisadas, mediante métodos de segmentación semántica pudieron detectar pepinos o frutos cuando se encuentran en proceso de maduración.

Es decir, lograron recolectar y se han probado otros algoritmos basados en el espacio de color. Por ejemplo, HSV-T no funcionó bien en la segmentación, YCBCR-FV fue aún más inseguro bajo una fuerte radiación solar, LCM y OHTA-SVM fueron altamente efectivos en cuando solo se segmenta una parte del algodón, el valor medio del método utilizado fue mejor que el de otros métodos utilizados en investigaciones de alta calidad, y la desviación estándar fue baja y estable. Nuestro método se asemejo con Groundtruth. se clasificaron imágenes de color de 3648 x 2736 Pixeles, utilizaron clasificador random forest, dentro de las limitaciones los autores recomendaron para futuras investigaciones realizar captura de imágenes con vehículo aéreo no tripulado para tener un mejor control en el campo de estudio.

El autor [19], en su tesis “Segmentación automática de texturas en imágenes agrícolas”, afirmo que hay problemas que afectaron el crecimiento en los cultivos, siendo las malezas que crecieron en los campos de sembríos de maíz y cereales, por lo que utilizaron drones no tripulados o equipos robóticos para solucionar este problema. Estas imágenes sufrieron un proceso de umbralización o segmentación, por lo que el promedio estadístico y la técnica OTSU se pudieron usar para separar el maíz y los granos de las hierbas malas que crecieron y para separar el suelo en la imagen. Este último no requirió supervisión. También tuvo que mencionar de la

investigación sobre la técnica de umbral automático, como la técnica K-MEANS, el método de entropía, el algoritmo de error mínimo, el umbral difuso, el algoritmo de isodatos y de retención de par. La extracción de textura espacial se realizó a través del índice de cromaticidad, esta estrategia tendió a combinar enverdecimiento e información, o extraer textura espacial donde utilizaron índice de vegetación y textura de extracción espacial para la transformación de ondículas. Se logro dividir en coeficientes de detalle horizontales, las líneas verticales y diagonales logran mejores resultados en la segmentación de imágenes. De esta manera la técnica OTSU es la principal estrategia para realizar el procesamiento de umbral para que esta técnica no presente ningún problema. La técnica OTSU supero a otras estrategias en aproximadamente un 4.5%. El autor recomendó que en futuras investigaciones se examinen otros cultivos cuyas características espectrales de otros cultivos (como el arroz o el frijol) puedan haber cambiado.

1.2. Formulación del problema

¿Cuál será el mejor algoritmo de segmentación de imágenes digitales para detectar plantas de arroz en ambientes no controlados?

– Justificación e importancia del estudio

Se evaluó diversos algoritmos de segmentación en imágenes digitales, tales como: SLIC, DBSCAN, K-MEANS, MASK RCNN, MS, PSO_SVM, PSO_MM, FCN, RED UNET MEJORADA, FUZZY – C PARTITION ENTROPY, DTSM, SLFDA, OTSU mejorado, sin embargo, dichos algoritmos pueden ser mejorados en la implementación de cultivos es por eso que se pretendió realizar un análisis comparativo en la segmentación de plantas de arroz en ambientes no controlados,

al finalizar la investigación se mostró la técnica que obtuvo mayor precisión en segmentación de imágenes de plantas de arroz, fue importante porque se desea minimizar un problema eminente en agricultura, además puede servir de base para futuras investigaciones, propuestas de mejoras o continuidad en investigaciones de detección de plagas.

1.3. Hipótesis

El algoritmo de segmentación en imágenes digitales que más precisión obtuvo fue K-MEANS para detectar plantas de arroz en ambientes no controlados.

1.4. Objetivos

1.4.1. Objetivo General

Realizar análisis comparativo de los algoritmos de segmentación de imágenes digitales para determinar cuál es mejor en detección de plantas de arroz en ambientes no controlados.

1.4.2. Objetivos Específicos

- a) Desarrollar el protocolo de adquisición de imágenes de plantas de arroz.
- b) Construir una base de datos de imágenes de plantas de arroz.
- c) Seleccionar las técnicas de segmentación de imágenes de plantas de arroz.
- d) Implementar los algoritmos para la segmentación de imágenes de plantas de arroz.

- e) Realizar las pruebas de segmentación y analizar los resultados de las segmentaciones obtenidas de las imágenes de plantas de arroz.

1.5. Teorías relacionadas al tema

1.5.1. Procesamiento digital de imágenes

Según los autores [20], son los diversos estudios que se realizaron computacionalmente para un posterior desarrollo, para adquirir estas imágenes se requirió realizar un proceso que involucre captura, muestreo, cuantificación y codificación.

1.5.2. Relaciones básicas entre píxeles

Son puntos o nodos importantes que existieron en una imagen para determinar relaciones existentes entre ellos. [20]

1.5.3. Vecinos de un píxel

[20] Es la convivencia que pudo existir entre píxeles, se denominarán vecinos cercanos porque mantuvieron una distancia, existen dos tipos:

1.5.3.1 La vecindad 4-vecinos

[20] Fueron cuatro píxeles constituidos por (V1, V2, V3, V4) que tuvieron cuatro ubicaciones que son: Arriba, abajo, derecha e izquierda del píxel en cuestión de P.

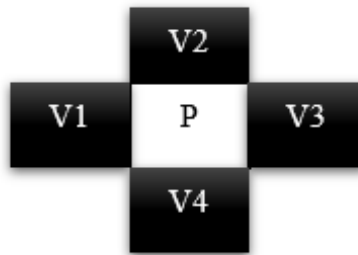


FIGURA 1: VECINDAD 4. FUENTE: [20]

1.5.3.2 La vecindad 8-vecinos

[20] Fueron ocho píxeles constituidos por (V1, V2, V3, V4, V5, V6, V7 y V8) siendo cuatro vecinos, más los cuatro vecinos que se encontraron en los diagonales de estos en cuestión de P.

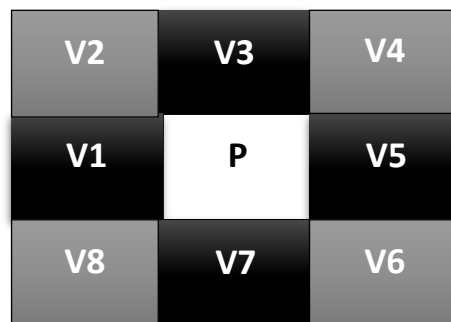


FIGURA 2: VECINDAD 8. FUENTE: [20]

1.5.4 Conectividad

[20] Es la unión de los píxeles, que se utilizaron para detectar regiones en una imagen, existen dos tipos de conectividad:

1.5.4.1 Conectividad- 4

La conectividad de píxeles se dio en forma horizontal y vertical. [20]

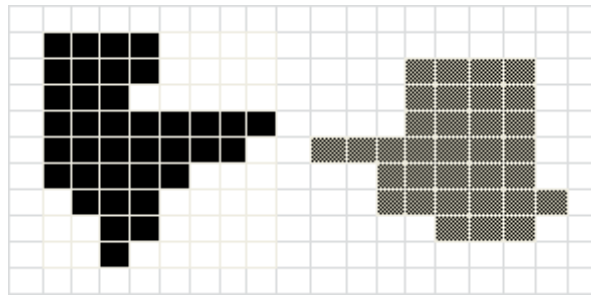


FIGURA 3: CONECTIVIDAD - 4. FUENTE: [20]

1.5.4.2 Conectividad- 8

La conectividad de pixeles se dieron en forma horizontal, vertical y diagonal. [20]

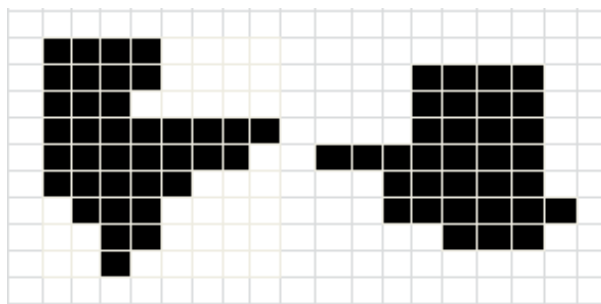


FIGURA 4: CONECTIVIDAD - 8. FUENTE: [20]

1.5.5 Las medidas de distancia

Es la longitud entre dos pixeles, dentro de ellos existieron la distancia euclidiana. [20]

– Distancia euclidiana

Es la longitud que existió entre dos píxeles con el fin de lograr obtener elementos con características similares. [20]

$$DE(P1, P2) = \sqrt{((X2 - X1)^2 + (Y2 - Y1)^2)}$$

**FIGURA 5. DISTANCIA EUCLIDIANA. FUENTE:
[20]**

1.5.6 Segmentación

[21] Realizo el proceso de dividir una imagen digital para obtener características que lograron ser de brillo y color con el objetivo de reconocimiento o de localización de una cara, persona, planta, que estuvieron dentro de una imagen.

1.5.7 Segmentación de imágenes

Es la descomposición de una imagen de tal forma que se subdividió o se separó con el fin de obtener zonas de interés que se presentarón en ella.[22]

1.5.8 Imágenes binarias

Una imagen al ser introducida en un ordenador donde debió ser reconocido en lenguaje máquina como ceros y unos.[20]

1.5.9 Textura

[23] Es una superficie a través de los materiales que pudieron ser táctil, cuando la textura estuvo siendo tratada en el relieve. Se detectaron a través de la vista, oído y tacto en lo que respecta a características de superficie pueden ser áspero, suave, marchito.

1.5.10 Segmentación por Bordes

Se denominó también detección de contornos el cual permitió encontrar las zonas de interés en una imagen a segmentar, recurriendo a la búsqueda de discontinuidades locales como detección de puntos, detección de líneas, y bordes teniendo la intensidad de una imagen en nivel de grises.[22]

1.5.11 Color

Derivó de sustancias reflejantes o de luz siendo el color blanco el que contuvo todos los colores del arcoíris y en los centros cerebrales de la visión humana crearon emociones o sensaciones. [23]

1.5.12 Conversión de RGB a HSV

Se empezó de los valores de los planos de color RGB donde propiamente se encontró dentro de un rango entre 0 y 255.[20]

1.5.13 El Modelo de CIE L*a*b*

Es un espacio de color que estuvo conformado por tres variables siendo L* la que representa la luminosidad, y los componentes de tonalidad que fueron representados por a* donde a* define la separación de longitud del eje rojo-verde y b* represento la separación de longitud del eje azul-amarillo. [20]

1.5.14 Separación de las componentes RGB en imágenes de grises

Dada una imagen a color se pudo extraer de esta los colores RGB, colores primarios rojo verde y azul, una vez extraídos estos tres canales o colores, se logró realizar la conversión en una figura en escala de grises.[24]

1.5.15 Imágenes en escala de grises

Es una matriz cuyos valores representaron un número de intervalos.
[20]

1.5.16 Filtrado de la mediana

Su principal función de un filtrado de mediana fue realizar los nodos con intensidades diferentes y fueron semejantes o muy idénticos a sus píxeles vecinos, borrando picos intensos que mostraron aislamiento en el área de la máscara de filtro.[22]

Por ejemplo se determinó un entorno de vecindad de 3 x 3 píxeles donde la mediana fue el quinto valor más grande, y tomando otro vecindario donde obtuvo un entorno de 5 x 5 píxeles donde la mediana tuvo que ser

el décimo tercer valor más grande y así sucesivamente para otras medidas de vecindarios, cuando se obtuvo varios pixeles en la vecindad y tienen el mismo valor se agruparon como se menciona a continuación: si tenemos una vecindad de 3 x 3 pixeles y tiene valores(20, 30, 10, 50, 50, 60, 50, 50, 30).Estos valores se ordenaron de la siguiente manera:(10, 20, 30,30, 50, 50, 50, 50, 60),se obtuvo el elemento 50 de la mediana.[25]

1.5.17 ¿Qué es un histograma?

Es la representación gráfica de frecuencias e intervalos, mostraron información de intensidad o contraste, iluminación y dinámica, asimismo propiedades importantes del comportamiento de una imagen. [20]

1.5.18 Curva ROC (Característica operativa del receptor)

[26] Constituida por dos puntos X e Y en la que determino que X es la proporción de falsos positivos (1- especificidad) e Y viene a ser la proporción de verdaderos positivos (sensibilidad) se dijo que cuando la curva está más próxima al ángulo superior izquierdo más asertividad tiene la prueba, mientras que cuando la curva está más próxima a la línea interrumpida menor es su utilidad en el estudio.

1.5.19 Matriz de confusión

[27] Es una herramienta que permitió la visualización del desempeño de un algoritmo utilizado, es empleado en aprendizaje supervisado, también

se le conoció como matriz de contingencia o matriz de errores, permitió mostrar errores cometidos por un modelo de clasificación.

		Clase predicha	
		P	N
Clase verdadera	P	TP	FN
	N	FP	TN

FIGURA 6: MATRIZ DE CONFUSIÓN BINARIA. FUENTE: [27]

1.5.20 Métricas derivadas de la matriz de confusión

A raíz de la matriz de confusión que se mencionó anteriormente se definió variedades de métricas que nos permitieron medir cuantitativamente la calidad de un esquema de clasificación. [27]

a. El error de la clasificación (ERR) y la exactitud (ACC)

Nos mostró un indicador general sobre instancias de clasificaciones incorrectas, **EI ERROR** es la suma de predicciones donde se logró tener un resultado falso sobre el número total de clasificaciones.[27]

$$ERROR = \frac{FP + FN}{FP + FN + TP + TN}$$

FIGURA 7. EL ERROR DE LA CLASIFICACIÓN. FUENTE:

Mientras que en LA PRECISIÓN se especificó que es el número de respuestas correctos sobre el total de clasificaciones.

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR$$

FIGURA 8. EXACTITUD. FUENTE: [27]

b. El Recall (REC) y la Sensibilidad (SEN)

Son los que correspondieron con la probabilidad de verdaderos positivos. [27]

$$REC = SEN = TPR = \frac{TP}{FN + TP}$$

FIGURA 9. EL RECALL Y LA SENSIBILIDAD. FUENTE: [27]

c. Especificidad (SPE)

Son los que correspondieron a la probabilidad que fueron verdaderamente clasificados como negativos respecto a todas las probabilidades que fueron negativos. [27]

$$SPE = \frac{TN}{TN + FP} = 1 - FPR$$

FIGURA 10. ESPECIFICIDAD. FUENTE: [27]

1.5.21 Intersección sobre la unión (IoU)

[28] Es conocido también como similitud de jaccard es una métrica de calidad para la segmentación semántica, el cual realizó el cálculo de similitud entre la intersección y la unión entre dos conjuntos, en el caso de segmentación de imágenes se trata de la imagen groundtruth y la segmentación efectuada por el algoritmo propuesto. Ayuda a determinar la sobre segmentación y la sub segmentación.

$$IoU = \frac{TP}{TP + FP + FN}$$

FIGURA 11. INTERSECCIÓN SOBRE LA UNIÓN. FUENTE:
[28]

1.5.22 Complejidad de un Algoritmo

Es la deficiencia que se obtuvo a través de datos matemáticos las características que obtendrá un sistema de información. [29]

1.5.23 Las Neuronas

Cada neurona de la red tuvo que ser aplicada a una función dedicada a sus datos de sus ingresos provenientes de las vinculaciones con otras neuronas así obtuvo un dato reciente que se convirtió en la salida de la neurona. [27]

a. Función de entrada o combinación

Su objetivo de dicha función fue combinar los diferentes ingresos con su peso y colocar los datos que se obtuvieron de todas las conexiones que ingresaron para lograr tener un único dato.[27]

b. Función de activación o transferencia

Adquirieron el dato obtenido por la función de combinación y lo modificaron antes de ser obtenido por la salida. [27]

1.5.24 Dimensiones de una red Neuronal

La capacidad de la capa de ingreso se fijó al comienzo con la cantidad de atributos que se consideraron de la totalidad de datos. La capa de partida de una red neuronal se definió a partir de la cantidad de categorías y de la codificación utilizada para su representación. Por lo general, en una dificultad de clasificación, se utilizó una neurona en la capa de salida para cada clase a representar, de modo que solo una neurona debería de «activarse» cuando se le solicite.

La topología y el tamaño de la capa oculta no fue un problema común, y no existió una respuesta única y excelente para esta dificultad. Existió literatura precisa dedicada al logro de la arquitectura eficaz para una red neuronal artificial. [27]

1.5.25 Funcionamiento y entrenamiento de una neurona

El movimiento común de una neurona fue definido por la función: $y = \sigma(wx - \alpha)$. [27]

1.5.26 Algoritmo

Tiene una secuencia de pasos a seguir para logra conseguir un propósito u objetivo, es como la realización de tareas o actividades diarias de una persona. [30]

1.5.26.1 Algoritmo PSO

Se basó en tener una cantidad de habitantes fijos fue conocida también como inteligencia de cúmulo, el habitante viene a ser la población conocida como partícula, de los cuales sus movimientos se encuentran dispuesto a un espacio de búsqueda. mantuvo un comportamiento social de vuelo de grupos de aves que fueron conformados por un líder y es a quien siguen. [31]

1.5.26.2 Agrupamiento jerárquico

Es jerárquico porque busco tener una jerarquía de pixeles en una determinada agrupación de datos, estuvo construido en el origen de árboles o llamados también dendogramas. [27]

1.5.26.3 Algoritmo K-means

Conocido como algoritmo de agrupamiento buscó una distribución de los datos, de tal manera que cada pixel estuvo asignado a un grupo, el

núcleo fue manifestado como centroide. Necesariamente no tuvieron que ser pixeles de datos, debiendo asignar el número K de clusters deseado ya que no es capaz de determinarlo por sí solo. [32]

Tuvo consideraciones importantes respecto a K-MEANS:

Número de grupos o de clúster fue expresado como K, debió ser definido antes de llevarse a cabo la ejecución del algoritmo.

Los clústeres o también conocidos como grupos estuvieron definidos por un nodo, fue reconocido como el centro y tendió a denominarse centroide del clúster. [27]

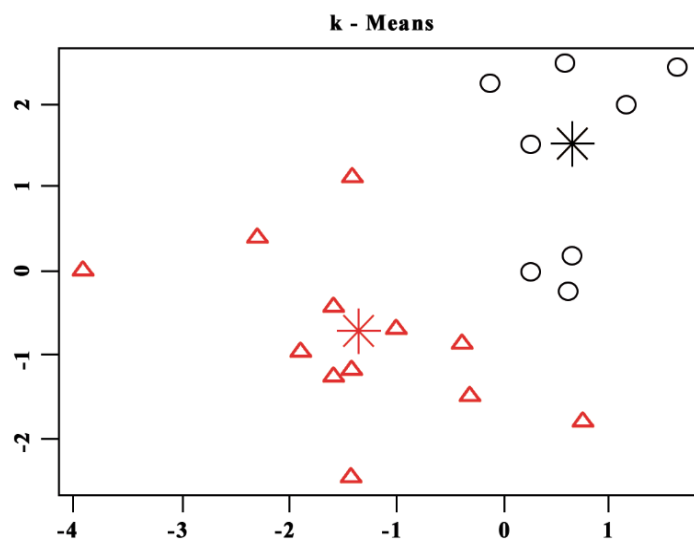


FIGURA 12. EJEMPLO DE VISUALIZACIÓN DE RESULTADOS DE K-MEANS. FUENTE: [27]

[27] El algoritmo logró funcionar en dos aspectos principales como se puede observar en el Pseudocódigo que se muestra a continuación:

Algoritmo: Pseudocódigo del algoritmo K-MEANS Entrada:

D(conjunto de datos) y K (número de clústeres)

Inicializar los clústeres vacíos $P = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_k\}$

Seleccionar los centroides iniciales $\zeta_1, \zeta_2, \dots, \zeta_k$

Mientras(Condición de parada no satisfecha) **hacer**

para todo $d_i \in D$ **hacer**

Asignar la instancia d_i al clúster \mathcal{P}_j | $d(d_i, \zeta_j)$ es mínima

fin para

para todo $\mathcal{P}_i \in p$ **hacer**

Recalcular ζ_i según los valores $d_i \in \mathcal{P}_i$

fin para

fin mientras

devolver El conjunto de clúster P

a. Inicialización de los centroides

Inicializó sus centroides con k puntos aleatorios de un grupo de datos. Se pudo apreciar que, $\zeta_i = \text{rand}(d_j) \mid d_j \in D, i = \{1, \dots, k\}$. las esenciales causas que se derivaron del inicio de centroides mencionado es que llegaron a implicar la evolución de clustering y fue no determinista. Esto quiso decir, en las distintas pruebas del algoritmo

podieron diagnosticar a diferentes modelos y por ende a distintas proporciones de calidad de resultado. [27]

El algoritmo K-MEANS determinó que al iniciar sus centroides fue aleatoria, existieron en él también otras continuidades que fueron más complejas y se pudieron utilizar de técnicas derivadas de K-MEANS como: K-MEDIANS, K-MEDOIDS, FUZZY C-MEANS.

Donde se pudo obtener centroides iniciales de la repartición de posibilidades de petición de D, y así poder exponer todo el conjunto de valores de los datos en esencial a todas las zonas con gran cantidad de concentración de instancias. [27]

b. Cálculo de la distancia

En el algoritmo K-MEANS por lo general tendió a utilizar la métrica de distancia euclídeana. Pero también puede adoptar a la utilización de otro tipo de métrica que tenga igual similitud. [27]

c. Recalculo de los centroides

El centroide obtuvo un valor donde es calculado como la media (Mean), el nombre de K-MEANS deriva de la aglomeración de puntos que existieron en este segmento, además el algoritmo propuesto solo pudo aplicarse a atributos continuos, de lo contrario de tener condiciones no consecuentes en el grupo de datos se debió emplear una rectificación preliminar. [27]

d. Criterio de parada

Según la convergencia que obtuvo el algoritmo fue su condición de parada natural y última que se obtuvo, esto conlleva a alcanzarse siempre y cuando no hayan repercutido una alteración en el recálculo de los centroides mientras la repetición completa prevalece en que no existió modificaciones en la disposición de las instancias en las distintas distribuciones o clústeres. Esto manifestó que logro llegar en un momento que hay consistencia en la asignación de las instancias, dicha condición pudo ser garantizada luego que se haya dado un número limitado de iteraciones, esto dependió de la métrica que se empleó al calcular la distancia y en el recálculo que se realizó en los centroides. [27]

e. Criterio para seleccionar un valor K

La principal característica que tuvo K-MEANS fue lograr expresar el número de particiones o clusters a establecer. Esta simplicidad adoptada le permitió estar con una alta eficiencia, aunque también pudo tener inconvenientes de gran importancia. Lograron identificar el número adecuado de particiones (k) no fue nada fácil, el rango de valores en k que fueron considerados muchas veces no logro ser muy grande con lo que se puede demostrar distintos elementos o valores y aquel que muestre los óptimos resultados. Uno de los criterios es reducir la suma del resto de los cuadrados (residual sum of squares, RSS), esto quiso decir que se debió reducir la adquisición de la longitud de cualquier vector o instancia al centroide que esté más cerca, lograron así el criterio buscar la creación de segmentos lo más consistente posible. [27]

Otro de los criterios que se consideró para tomar en cuenta es que se maximizó la suma de distancias entre segmentos pudiendo ser sus centros que estos tienen, y priorizando así lograr obtener segmentos más alejados posibles entre ellos, con lo mencionado se obtuvieron segmentos diferenciados entre sí, en los criterios anteriormente mencionados (reducción de distancias intragrupo o ampliación de distancias intergrupo) pudieron utilizarse y lograron establecer un valor exacto y justo para el parámetro de las particiones. [27]

1.5.26.4 Metodo OTSU

[33]Estuvo basado en analisis de discriminante, realizaron la maximizacion de algunas medidas de separabilidad entre las clases, la operación de binarización fue visualizada como la selección de un umbral que particiono pixeles en dos clases.

$$C_0 = \{0, 1, 2, \dots, t\} \text{ y } C_1 = \{t+1, t+2, \dots, L-1\},$$

Donde las clases pudieron corresponder al fondo y también lograron corresponder al objeto de interés.

1.5.26.5 Redes neuronales convolucionales

Las neuronas se asimilaron a neuronas que fueron situadas en la cubierta visual primaria dentro de un ingenio biológico. [34]

1.5.26.6 Algoritmo SLIC

Es un algoritmo de agrupación lineal simple, estuvo basado en la obtención de superpíxeles, tuvieron dos distinciones importantes uno de ellos es el número de cálculos de distancias donde la optimización se disminuyó rigurosamente al limitar el espacio de búsqueda a una región proporcional al tamaño del superpíxel y por otra parte la dimensión de distancia ponderada combinó el color y la continuidad espacial mientras que simultáneamente tiene el control sobre el tamaño y la capacidad de los superpíxeles. [35]

1.5.26.7 Algoritmo MS

Es un algoritmo de cambio medio que fue utilizado principalmente en la sustracción de características de interés y la segmentación de imágenes, fue capaz de clasificar imágenes verdes y no verdes. [36]

1.5.26.8 Algoritmo DTSM

Es un modelo de segmentación que se basó en un árbol de decisión creado por el algoritmo CART así mismo adoptaron los ítems de reducción de ruido utilizando información de color RGB de cada píxel de las imágenes teniendo de esta manera varias características de color. [36]

1.5.26.9 Algoritmo DBSCAN

Es un algoritmo de agrupación espacial que estuvo basado en la densidad, utilizó la densidad del conjunto de puntos de la muestra como base para lograr determinar la categoría del resultado de agrupación. La

muestra de la misma categoría debió estar estrechamente conectada, al llegar a alcanzar el umbral de densidad preestablecido, el algoritmo tendió a dividir el bloque de puntos compuesto por este conjunto de puntos en múltiples regiones, la gran fortaleza de este algoritmo es que todos los nodos o puntos del grupo de datos fueron tratados todos por igual y no se obtuvo resultados de clustering que fueron afectados por sensibilidad a los valores atípicos que se puedan obtener[37].

En el algoritmo DBSCAN, la vecindad Eps viene a ser el radio de agrupación en la ejecución del algoritmo. El punto (objeto) central tendió a referirse a los objetos de punto en el conjunto de datos espaciales que se encontraron con los puntos MinPts en su vecindad Eps. Los objetos de punto fueron denominados vecinos Eps del punto central, que fueron expresados como $N_{eps}(p)$, la densidad alcanzable directa se refirió a dos puntos cualesquiera en la vecindad del conjunto de datos espaciales Eps, de los cuales sólo uno fue el objeto central. Si el objeto q y el objeto p llegaron a ser directamente alcanzables por la densidad, la condición debió quedar satisfecha. La densidad alcanzable fue que, si dos puntos adyacentes en el conjunto de puntos, que viene a ser un conjunto ordenado de puntos en el conjunto de datos espaciales, son directamente alcanzables por la densidad, el punto inicial y el punto final son alcanzables por la densidad. La conexión de densidad consistió en que, en el conjunto de datos espaciales, si se ha determinado el punto 0 y los otros dos puntos del conjunto fueron alcanzables por la densidad, los dos puntos estuvieron conectados por la densidad. Se necesitaron dos parámetros para implementar el algoritmo: vecindad Eps y vecindad mínima MinPts [37].

A continuación, se describe el algoritmo tradicional DBSCAN:

Algoritmo: Pseudocódigo del algoritmo DBSCAN

Entrada: $N_D, \epsilon, \text{MinPts}$

Salida: Colección de agrupaciones basadas en la densidad

Marcar todos los objetos de N_D como no visitados;

Hacer;

 Seleccionar aleatoriamente P_i no visitados en N_D ;

 Marcar P_i como visitado;

 Establezca N como el número de puntos en la vecindad ϵ de P_i ;

Si($N \geq \text{MinPts}$)

 Crear un nuevo clúster C , $P_i \rightarrow C$;

 Establece N_P como el conjunto de puntos en la vecindad ϵ de P_i ;

 Para cada punto no visitado P_x en N_P ;

Si (P_x no es visitado)

 Marcar P_x como visitado;

 Establezca n como el número de puntos en la vecindad ϵ de P_x ;

Si($n \geq \text{MinPts}$)

Si($P_x \notin C$)

$P_x \rightarrow C$;

Fin si

Fin si

Fin si

Fin para;

Salida C;

Si no

Marca P_i como punto de ruido;

Fin si

Hasta que se hayan visitado todos los puntos de N_D ;

1.5.27 Clasificadores

Consiste en detectar el reconocimiento de un patrón o algoritmo a través de sus cualidades que lo caracterizan.

1.5.27.1 Máquina de vectores de soporte SVM

[32] Es el que realiza una descripción de hiperplano siendo los píxeles más cercanos al él. A continuación, se mostraron en la siguiente figura:

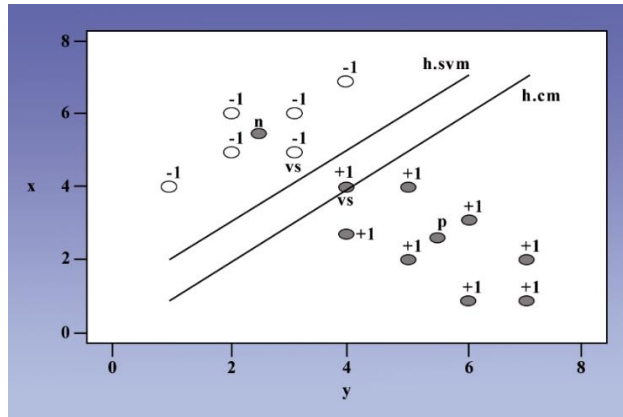


FIGURA 13. HIPERPLANOS LINEALES DE LA MÁQUINA DE VECTORES DE SOPORTE. FUENTE: [32]

1.5.27.2 Random Forest RB

Es de mucha utilidad para representar reglas de clasificación, en cada rama de un árbol de decisión representa una regla que se determina entre un conjunto de valores de un atributo básico. [32]

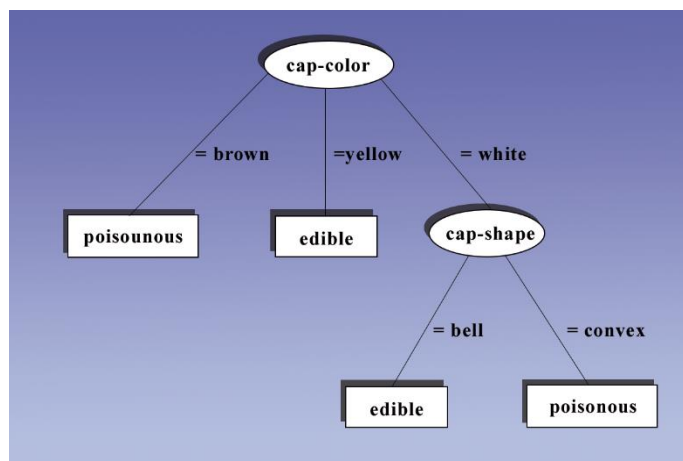


FIGURA 14. ÁRBOL DE DECISIÓN. FUENTE: [32]

1.5.28 Sistemas

Es la interconexión de elementos interrelacionados para obtener un fin u objetivo en un determinado ámbito. [38]

1.5.29 Sistemas de Información

Es la reunión de dos elementos o más para convertir los datos en información precisa y confiable y obtener una respuesta de manera rápida y eficiente. [38]

1.5.30 Lenguaje de Programación

Sirve para escribir un programa, es desarrollado a través de algoritmos que son interpretados por el lenguaje maquina (ceros y unos). [39]

1.5.30.1 C++

[39] Es un lenguaje que fue utilizado en programación estructurada que provino del lenguaje C, donde se realizaron las programaciones de letras, números, símbolos, estos fueron interpretados por el lenguaje máquina para automatizar procesos que se plantearon y poder realizar la adquisición de datos.

1.5.30.2 Qué es java

Tecnología informática que fue basada en la programación, fácil de usar y entender, porque sus instrucciones fueron interpretados por su propia máquina

virtual java(SVM), se realizaron programas pequeños y amplios que fueron ejecutados en cualquier plataforma.[40]

1.5.30.3 Python

Es un lenguaje que fue utilizado mucho en programaciones de alto nivel para usuarios principiantes y usuarios avanzados, es más fácil el acceso a librerías o bibliotecas de datos muy amplios.[41]

II. MATERIAL Y MÉTODO

2.1. Tipo y Diseño de Investigación

2.1.1. Tipo de investigación

El tipo de investigación es cuantitativa porque se evaluaron los algoritmos de segmentación que mejores resultados obtuvieron en exactitud y/o precisión en investigaciones de segmentación de imágenes digitales, para poder ser utilizado posteriormente en desarrollos tecnológicos para determinar plagas en las plantas de arroz.

2.1.2. Diseño de investigación

El diseño de la investigación es cuasi – experimental porque se evaluaron imágenes de plantas de arroz con variables dependientes en función de variables independientes que determinaron los mejores resultados obtenidos en ambientes no controlados y así poder observar sus efectos alcanzables.

2.2. Variables, Operacionalización

2.2.1. Operacionalización

TABLA 2.

OPERACIONALIZACIÓN DE VARIABLES

Variable de estudio	Definición conceptual	Definición operacional	Dimensiones	Indicador	Ítem	Instrumento	Valores finales	Tipos de variables
Variable independiente: Algoritmos de segmentación de imágenes digitales.	Medir los Algoritmos de segmentación de imágenes digitales.	Tiempo de demora	Recursos	Tiempo de respuesta	$TR = TF - TI$	Ficha de registro electrónico	Tiempo	Horas
		Consumo de recursos		Consumo de memoria RAM	$MR = \frac{TM - UM}{TM}$		Gigabyte	Megabyte
				Consumo de CPU	$CPU = \frac{TCPU}{NCPU}$		Gigabyte	Megabyte
Variable dependiente: Plantas de arroz en	Caso de estudio son las plantas de arroz en	Testeo		Exactitud	$ACC = \frac{TP+TN}{FP+FN+TP+TN}$	Ficha de registro electrónico	Cantidad	Numero

ambientes no controlados. ambientes no controlados.

Test de clasificación

Sensibilidad

$$\mathbf{SEN} = \frac{TP}{FN+TP}$$

Cantidad

Numero

Especificidad

$$\mathbf{SPE} = \frac{TN}{TN + FP}$$

Cantidad

Numero

Fuente: Elaboración propia

2.2.2. Variables

a) Variable Independiente

Algoritmos de segmentación de imágenes digitales.

b) Variable Dependiente

Plantas de arroz en ambientes no controlados.

2.3. Población de estudio, muestra, muestreo y criterios de selección

En esta presente investigación se logró determinar que la población estuvo compuesta por 15 algoritmos de segmentación de imágenes digitales, siendo de Hojas, plantas, frutas y imágenes médicas, los cuales fueron utilizados en diferentes investigaciones relacionadas al tema, los trabajos estudiados fueron corroborados en el anexo N° 2, y la muestra que se obtuvo para esta investigación fue por dos algoritmos de segmentación de acuerdo al análisis realizado en la evaluación de algoritmos de segmentación de la tabla N°11, la muestra que se logró fueron con dos algoritmos siendo DBSCAN y K-MEANS porque obtuvieron mejores precisiones en segmentación de imágenes digitales.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

2.4.1. Técnica

a) Observación

Es una técnica que fue de gran consideración en el análisis de la presente investigación ya que se observó resultados obtenidos de los algoritmos de segmentación en imágenes digitales que lograron tener

eficientes precisiones para poder aplicarlos en análisis de escenarios de ambientes no controlados.

b) Análisis documental

Luego de haber logrado tener una revisión en el estudio de los resultados obtenidos en los diferentes artículos de segmentación de imágenes digitales, se obtuvieron las técnicas más precisas para ser puestas a disposición y poder probarlos en ambientes no controlados y aplicarlos en nuestro sistema de información comparativo.

2.4.2. Instrumentos

En el presente proyecto desarrollado el investigador logro obtener las imágenes con una cámara digital Canon PowerShot SX150 IS, luego fueron llevados a un sistema de información, proceso que fue realizado por Algoritmos de separar internamente las plantas de arroz de otros elementos que existieron en la imagen como: barro, agua, hierbas. Así mismo se eliminaron los brillos y ruidos que tenían las imágenes, como el aire, sombras y reflejos de luz cambiante según sea el horario del día en que se capturaron las imágenes digitales de plantas de arroz, Se tomó en cuenta el color y textura de las plantas para poder determinar los resultados de cuales fueron plantas pixel y plantas no pixel.

2.5. Procedimientos de análisis de datos

Para la adquisición de las imágenes digitales se tomó en cuenta la colaboración del agricultor que tiene cultivo de arroz, se consideraron los pasos que detallamos a continuación:

- a. Se realizó la captura de imágenes digitales en campos de arroz haciendo uso de una cámara digital Canon PowerShot SX150 IS.
- b. Se realizó el proceso de segmentación con las imágenes digitales de plantas de arroz haciendo uso de los algoritmos K-MEANS y DBSCAN.
- c. Se obtuvieron las formas de las plantas de arroz en ambientes no controlados haciendo uso de los algoritmos de segmentación seleccionados.
- d. Se realizó el proceso digital de las imágenes de plantas de arroz utilizando lenguaje de programación Python versión 3.7 y haciendo uso de laptop Lenovo Ideapad S145, Core I7, Memoria Ram 8 gb, Tarjeta de video Nvidia Geforce 2gb.
- e. Se desplego los resultados haciendo uso de las métricas de calidad para medir las precisiones en las segmentaciones de imágenes digitales y así poder permitirnos mostrar información de interés y ser evaluados posteriormente.

Los resultados fueron evaluados con estadísticas como recursos de hardware del computador utilizado y métricas de clasificación de imágenes digitales que nos permitieron determinar la optimización de los algoritmos de segmentación utilizados en el proyecto de investigación desarrollado, detallándose a continuación:

2.5.1. Variable Independiente

– Tiempo de respuesta

Es el tiempo de reacción que se utilizó desde que iniciaron el proceso de ejecución hasta la finalización del proceso de ejecución.

$$\text{Formula: } TR = TF - TI$$

Donde:

TABLA 3.

DESCRIPCIÓN DE LAS VARIABLES DE LA FÓRMULA TIEMPO DE RESPUESTA

Variable	Descripción
TF	Tiempo Final.
TI	Tiempo Inicial.

Fuente: Elaboración propia

El Tiempo de respuesta fue adquirido del tiempo final menos el tiempo inicial.

– Consumo de memoria RAM

Fue el consumo promedio de la memoria que se utilizó en la ejecución de un proceso.

$$\text{Formula: } MR = \frac{TM - UM}{TM}$$

Donde:

TABLA 4.

DESCRIPCIÓN DE LAS VARIABLES DE LA FORMULA CONSUMO DE MEMORIA RAM.

Variable	Descripción
TM	Total, de Memoria RAM.
UM	Memoria RAM utilizada.

Fuente: Elaboración propia.

El total de memoria RAM fue restado con la memoria RAM utilizada y luego el resultado fue dividido con el total de memoria RAM, lográndose obtener el consumo de memoria RAM que fue utilizado en el proceso de segmentación de imágenes digitales.

– **Consumo de CPU**

Es la unidad central de proceso principal de un computador, fue el encargado de realizar internamente el procesamiento de los datos y llevar la realización de los cálculos matemáticos.

$$\text{Formula: CPU} = \frac{\text{TCPU}}{\text{NCPU}}$$

Donde:

Tabla 5.

Descripción de las variables de la fórmula del Consumo del CPU.

Variable	Descripción
TCPU	Uso total de unidad central del proceso.
NCPU	Uso no utilizado en la unidad central de proceso.

Fuente: Elaboración propia

El consumo del uso total de unidad central del proceso fue dividido entre el uso no utilizado en la unidad central de proceso donde obtuvieron como resultado el consumo de unidad central de proceso.

2.5.2. Variable dependiente

– **Exactitud**

Indica lo más próximo o cercano que se encuentra el resultado de una medición del valor verdadero.

$$\text{Formula: ACC} = \frac{TP+TN}{FP+FN+TP+TN}$$

Donde:

TABLA 6.

DESCRIPCIÓN DE LAS VARIABLES DE LA FORMULA EXACTITUD.

Variable	Descripción
TP	Verdaderos Positivos.
TN	Verdaderos Negativos.
FP	Falsos Positivos.
FN	Falsos Negativos.

Fuente: Elaboración Propia.

La Exactitud (ACC) se obtuvo de la suma de todos los Verdaderos Positivos y Verdaderos negativos, estos fueron divididos entre la suma de todos los Falsos Positivos, Falsos negativos, Verdaderos positivos, Verdaderos negativos.

– **Sensibilidad**

Represento los casos correctamente identificados que dieron como verdaderos positivos.

$$\text{Formula: SEN} = \frac{TP}{FN+TP}$$

Donde:

Tabla 7.

**DESCRIPCIÓN DE LAS VARIABLES DE LA FORMULA
SENSIBILIDAD.**

Variable	Descripción
TP	Verdaderos Positivos.
FN	Falsos Negativos.

Fuente: Elaboración Propia

La sensibilidad fue obtenida por los verdaderos positivos entre la suma de falsos negativos y verdaderos positivos.

– **Especificidad**

Represento los casos correctamente identificados que dieron como verdaderos negativos.

$$\text{Formula: SPE} = \frac{\text{TN}}{\text{TN+FP}}$$

Donde:

TABLA 8.

DESCRIPCIÓN DE LAS VARIABLES DE LA FORMULA ESPECIFICIDAD.

Variable	Descripción
TN	Verdaderos negativos.
FP	Falsos positivos.

Fuente: Elaboración Propia

La especificidad fue obtenida por los verdaderos negativos entre la suma de los verdaderos negativos y falsos positivos.

Nota: Se debe tener en cuenta lo siguiente:

VP: Verdadero Positivo, existieron plantas de arroz en la imagen y detectaron la presencia de plantas de arroz.

FP: Falso Positivo, No existieron plantas de arroz en la imagen y detectaron la presencia de plantas de arroz.

VN: Verdadero Negativo, No existieron plantas de arroz en la imagen y no detectaron la presencia de plantas de arroz.

FN: Falso Negativo, existieron plantas de arroz en la imagen y no se detectaron la presencia de plantas de arroz.

2.6. Criterios éticos

2.6.1. Confidencialidad

El material que se utilizó para el procedimiento del desarrollo de la tesis estuvo descrito y mencionados con sus correspondientes autores como colaboradores del estudio.

2.6.2. Criterios de rigor Científico

– **Fiabilidad**

El proyecto cumplió con las posibilidades de desarrollo expuestos en su contenido y su elaboración con el estándar y políticas para su desarrollo.

– **Validez**

Información que fue obtenida por el estudio o proyectos que fueron diagnosticados y estudiados por los ingenieros especialistas en la materia para establecer su autenticidad.

– **Consistencia**

El estudio investigado represento información sólida y certificada por la comunidad científica.

III. RESULTADOS

3.1. Resultados

En esta investigación fueron evaluados el algoritmo K-MEANS y el algoritmo DBSCAN, ambos algoritmos se utilizaron para la comparación en segmentación de imágenes digitales de plantas de arroz en ambientes no controlados, se evaluó un total de 100 imágenes digitales de plantas de arroz, fueron 50 imágenes en día nublado, donde se dividieron en 25 imágenes hacia adelante y 25 imágenes hacia atrás, las otras 50 imágenes fueron tomadas en día soleado, siendo 25 imágenes en posición de diagonal a la izquierda y las otras 25 imágenes fueron tomadas en posición de diagonal a la derecha.

Con respecto a los rendimientos de hardware en los algoritmos evaluados se observó en las figuras N° 15, 16, 17, 18, el algoritmo DBSCAN tuvo un alto rendimiento en procesamiento de las imágenes con respecto al algoritmo K-MEANS, estos resultados se debieron a que se tuvo un distinto pre procesamiento de imágenes en ambos porque para el algoritmo K-MEANS se preproceso con el espacio de color CIELab y se tomó la dimensión b, mientras que en el algoritmo DBSCAN se pre proceso con el espacio de color HSV donde se tomó la dimensión S.

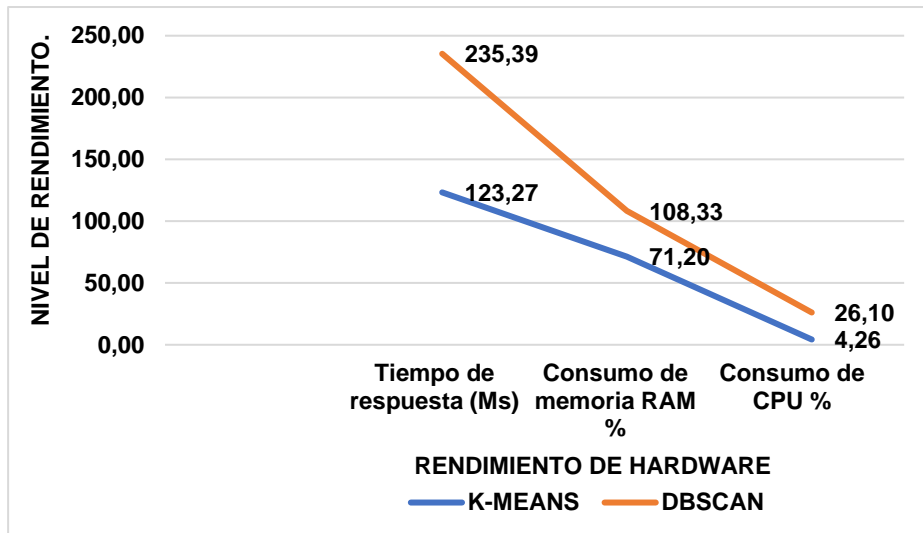


FIGURA 15: RENDIMIENTO DE CÓMPUTO EN 25 IMÁGENES DE PLANTAS DE ARROZ HACIA ADELANTE EN DÍA NUBLADO. FUENTE: ELABORACIÓN PROPIA.

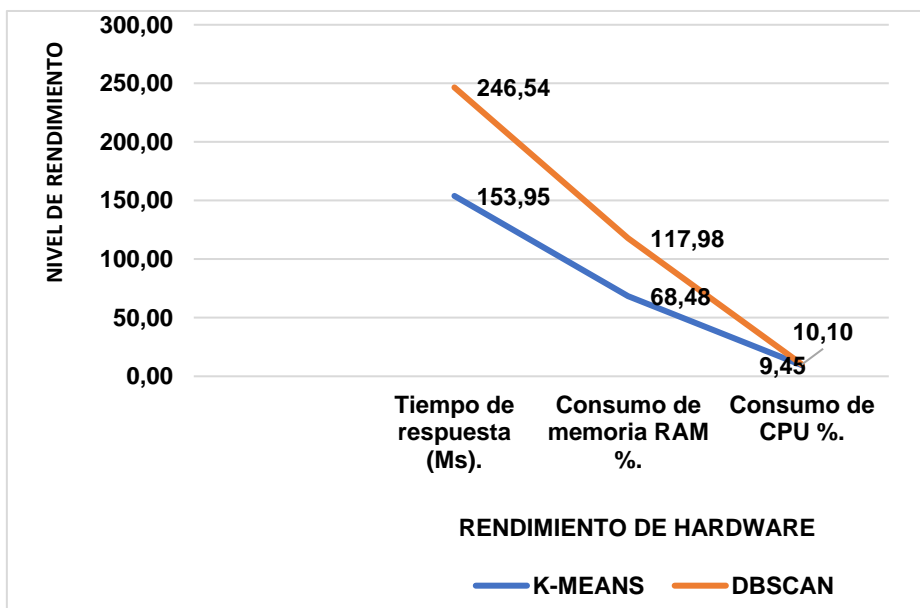


FIGURA 16: RENDIMIENTO DE COMPUTO EN 25 IMÁGENES DE PLANTAS DE ARROZ HACIA ATRÁS EN DÍA NUBLADO. FUENTE: ELABORACIÓN PROPIA.

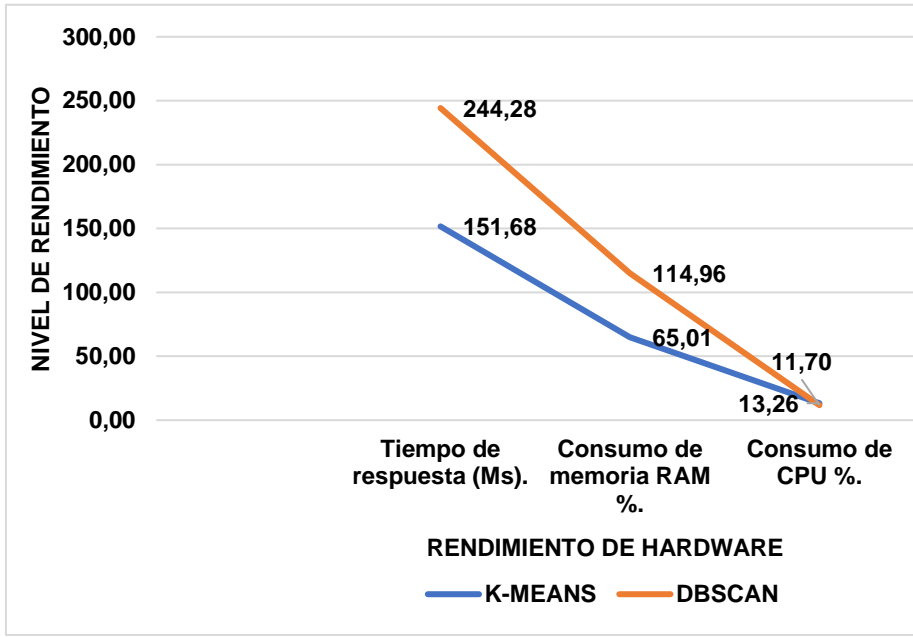


FIGURA 17: RENDIMIENTO DE CÓMPUTO EN 25 IMÁGENES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA DERECHA EN DÍA SOLEADO. FUENTE: ELABORACIÓN PROPIA

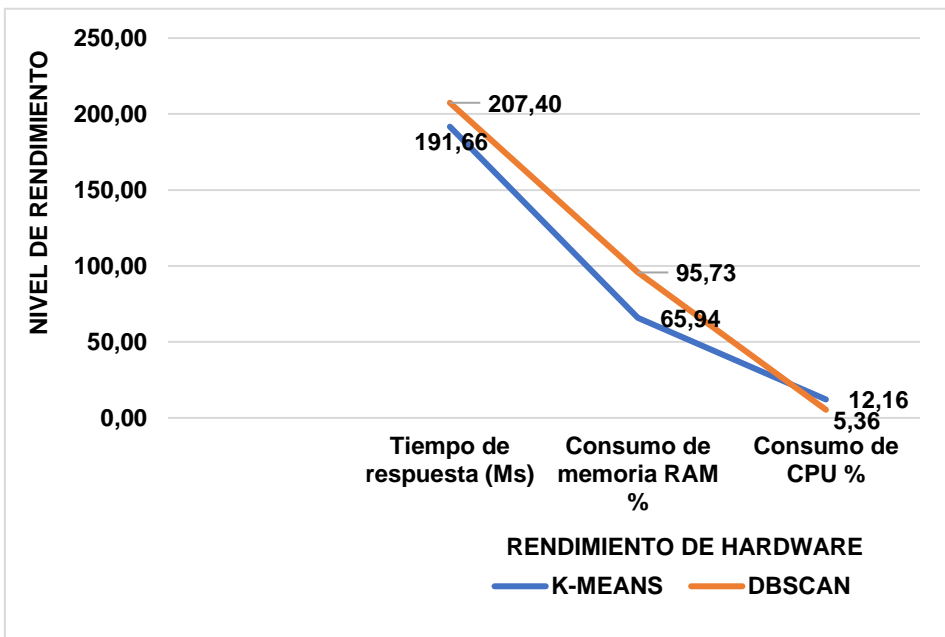


FIGURA 18: RENDIMIENTO DE CÓMPUTO EN 25 IMÁGENES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA IZQUIERDA EN DIA SOLEADO. FUENTE: ELABORACIÓN PROPIA.

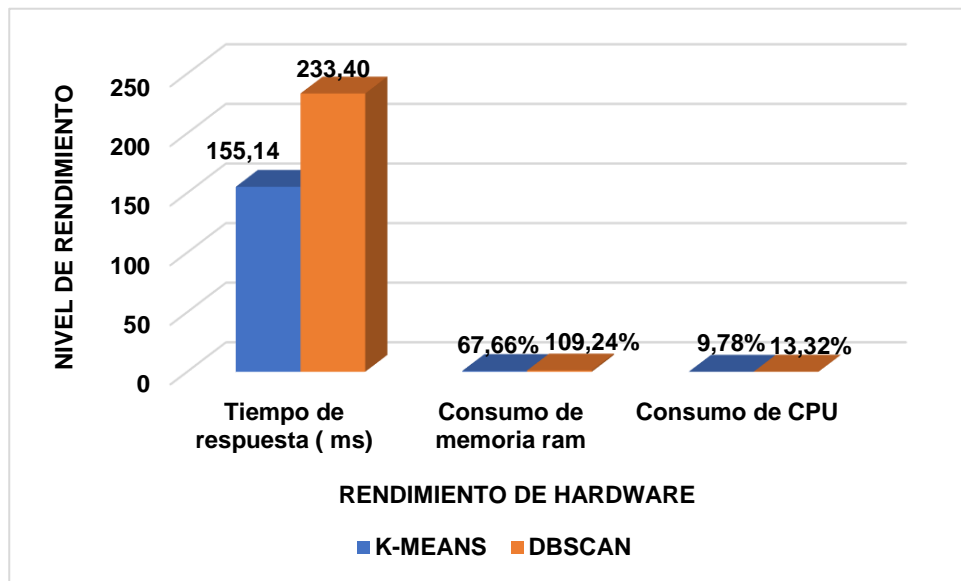


FIGURA 19: RENDIMIENTO DE CÓMPUTO TOTAL DE 100 IMÁGENES DIGITALES DE PLANTAS DE ARROZ UTILIZANDO LOS ALGORITMOS K-MEANS Y DBSCAN. FUENTE: ELABORACIÓN PROPIA.

En la figura N° 19 se observó el rendimiento total de las 100 imágenes digitales de plantas de arroz en ambientes no controlados, donde el algoritmo K-MEANS el tiempo de respuesta fue de 155.14 ms, consumo de memoria RAM fue de 67.66%, consumo de CPU fue de 9.78%, mientras que con el algoritmo DBSCAN tuvo un tiempo de respuesta de 233.40 ms, consumo de memoria RAM de 109.24%, consumo de CPU fue de 13.32%. además de tener en cuenta que los espacios de color fueron distintos en tener diferentes cromaticidades en sus coordenadas donde influenciaron mucho en el rendimiento de hardware para el desempeño de la segmentación con los algoritmos en mención, se debe considerar también los requerimientos de hardware del computador que fueron puntuados en el procesamiento de análisis de datos de esta investigación donde se observó requerimientos de hardware mínimos para el desempeño del procesamiento de los algoritmos de segmentación.

En la figura N°20 se observó, con respecto a la comparación en la segmentación de las 25 imágenes digitales de plantas de arroz hacia adelante en día nublado, el que menor Exactitud y Sensibilidad tuvo fue el algoritmo DBSCAN debido a que habían imágenes en las cuales fueron adquiridas sus partes sobresalientes, como las orillas de donde fueron sembradas las plantas, así mismo también se pudo apreciar que las imágenes tuvieron en la parte baja del agua hiervas de color verde, con respecto a esta complejidad logro obtener segmentaciones sub óptimas debido a que confundió mucho las plantas de arroz con los que no eran plantas.

Por otro lado, el algoritmo K-MEANS obtuvo mejores resultados debido a que logró detectar plantas de arroz de los que no eran arroz, tuvo algunas dificultades en cuanto a segmentación porque algunos pixeles plantas fueron detectados como pixeles del suelo (Agua, Barro, Hiervas).

Para ambos algoritmos también interfirió el desenfoque que tuvieron las imágenes originales de plantas de arroz adquiridas en el ambiente no controlado porque se observó que las partes donde se tuvo dicho inconveniente no se logró segmentar bien esa área desenfocada.

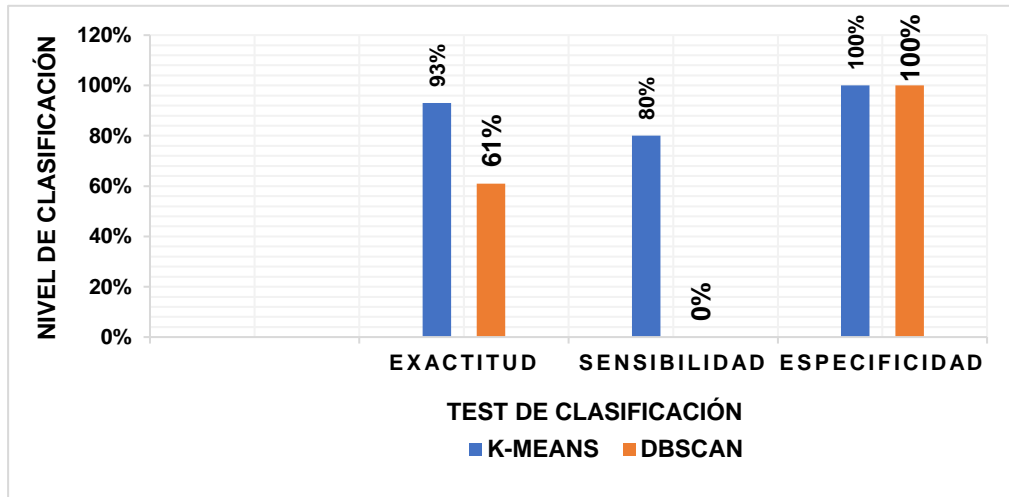


FIGURA 20: SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ADELANTE EN DÍA NUBLADO. FUENTE: ELABORACIÓN PROPIA.

En la Figura N° 21, se observó que, en la segmentación de las 25 imágenes de plantas de arroz hacia atrás en día nublado, el algoritmo DBSCAN tuvo resultados subóptimos debido a que hubo en el agua hierbas de color verde, así también tuvieron áreas sobresalientes en las imágenes como los bordos con especies de hierbas y por último hubo imágenes en las cuales las plantas de arroz estuvieron muy aglomerados producto del aire, razón por lo que no logró identificar bien cuales fueron plantas de la que no fueron plantas.

Por otro lado, el algoritmo K-MEANS identificó mejor las plantas de arroz de las que no eran plantas de arroz, tuvo pocos pixeles plantas que se confundieron con el agua, barro, hierbas que existieron en las imágenes digitales.

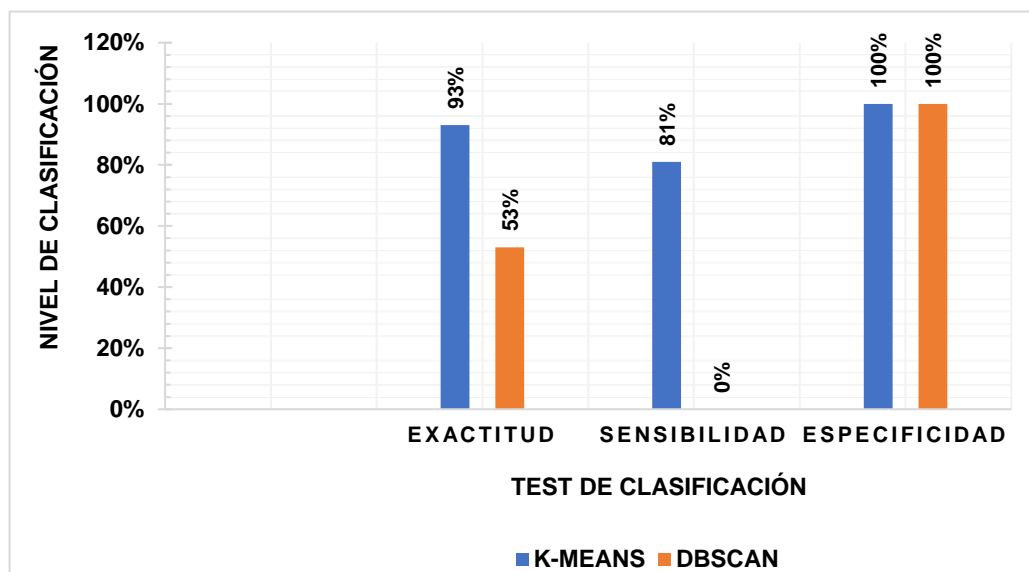


FIGURA 21: SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ATRÁS EN DÍA NUBLADO. FUENTE: ELABORACIÓN PROPIA.

En la Figura N° 22 se observó que el algoritmo DBSCAN tuvo resultados sub óptimos en la segmentación de las imágenes digitales de plantas de arroz en posición en diagonal hacia la derecha en día soleado, porque en las partes de las imágenes que no eran arroz se apreció como una especie de coágulos de color blanco en el agua lo que dio lugar a la confusión entre las plantas de arroz y las partes que no eran plantas de arroz como el agua, barro, hierbas.

El algoritmo K-MEANS logró superar esa sub segmentación que tuvo el algoritmo DBSCAN sin embargo solo algunos pixeles plantas se confundieron con los pixeles que no fueron plantas como el agua, barro, hierbas que existieron en las imágenes digitales.

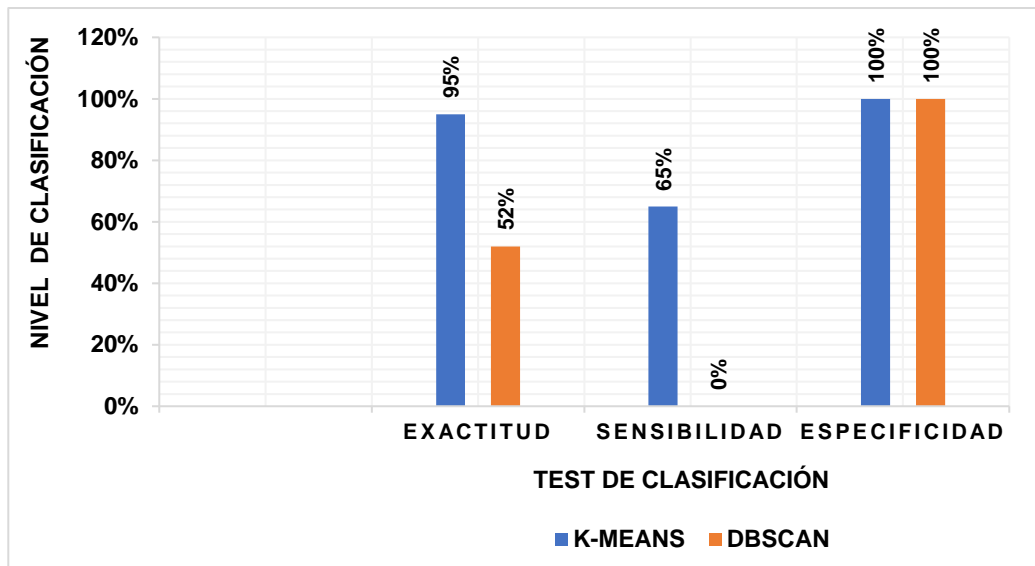


FIGURA 22: SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA DERECHA EN DÍA SOLEADO. FUENTE: ELABORACIÓN PROPIA.

En la Figura N° 23 se observó que el algoritmo DBSCAN tuvo resultados sub óptimos con respecto al algoritmo K-MEANS en la segmentación de imágenes digitales de plantas de arroz en posición diagonal hacia la izquierda en día soleado, porque hubo partes en las imágenes con sombras de algunos arbustos que había cerca al arrozal, esto dio lugar a que no logró procesar bien la segmentación en las áreas afectadas porque las plantas fueron confundidas como parte del suelo.

Por otra parte, el algoritmo K-MEANS tuvo un similar inconveniente con las imágenes de plantas de arroz sombreadas, solo que el error de segmentación fue mucho menor.

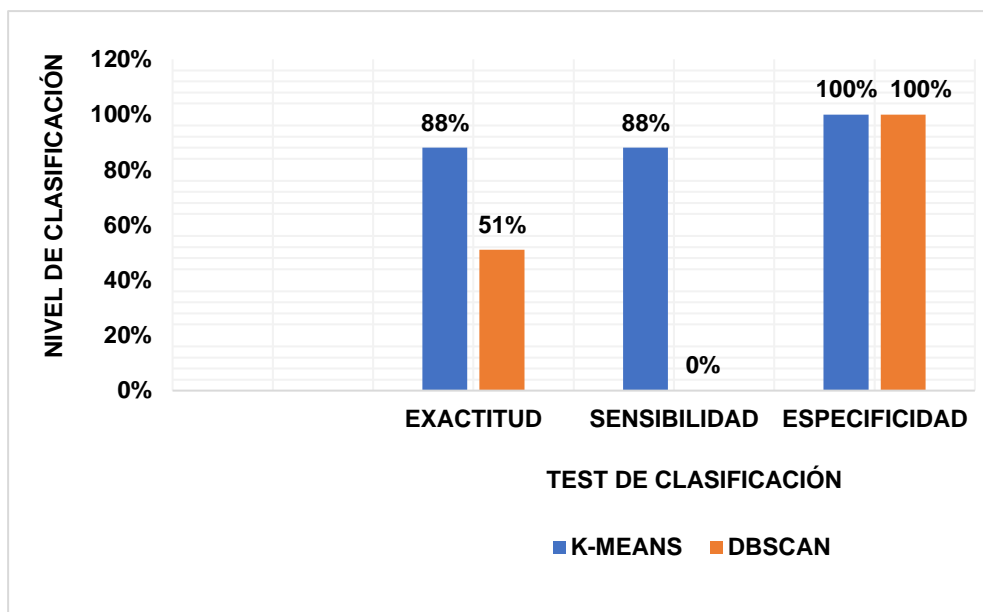


FIGURA 23: SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA IZQUIERDA EN DÍA SOLEADO. FUENTE: ELABORACIÓN PROPIA.

En la figura N° 24 Se promedió el total de todas las 100 imágenes digitales de plantas de arroz que fueron segmentadas hacia adelante, hacia atrás, en posición en diagonal hacia la izquierda, en posición en diagonal hacia la derecha. Donde el algoritmo K-MEANS obtuvo una Exactitud de 92.25%, una Sensibilidad de 78.50%, y Especificidad de 100%, mientras que el algoritmo DBSCAN obtuvo una Exactitud de 54.25%, una Sensibilidad de 0% y Especificidad de 100%.

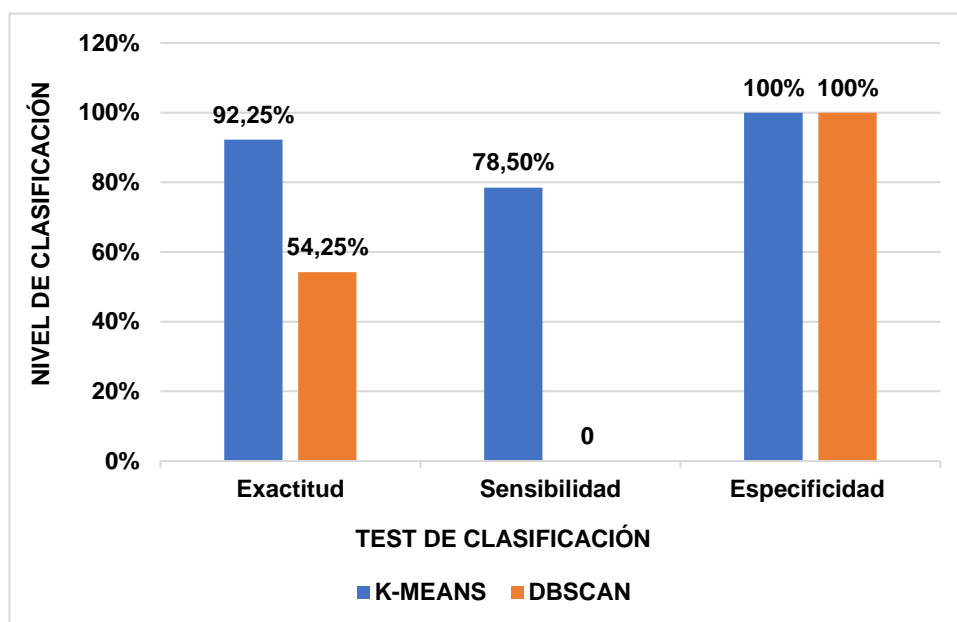


FIGURA 24: PROMEDIO TOTAL DE 100 IMÁGENES DIGITALES DE PLANTAS DE ARROZ SEGMENTADAS UTILIZANDO LOS ALGORITMOS K-MEANS Y DBSCAN. FUENTE: ELABORACIÓN PROPIA.

Se demostró que el algoritmo DBSCAN obtuvo resultados poco favorables con respecto al algoritmo K-MEANS en la segmentación de imágenes digitales de plantas de arroz porque no logró detectar en la mayoría de los píxeles que fueron plantas de los píxeles que no fueron plantas de arroz. Para detectar las áreas que fueron plantas, el algoritmo DBSCAN se basó en píxeles que tuvieron regiones demasiado densas en decir áreas donde los píxeles lograron estar demasiado cerca para poder lograr la identificación, teniendo dificultades al momento de segmentar imágenes digitales que tuvieron píxeles densos o superpuestos confundiendo demasiado plantas píxeles con otros elementos que existieron en las imágenes que no fueron plantas píxeles.

3.2. Discusión

Cuando se aplicó la segmentación de imágenes de hojas de tomate, se basó en el número de clustering adaptativo, donde se realizó con el algoritmo mejorado K-MEANS. Se logró la segmentación con 20 imágenes de hojas de tomate con enfermedad de mosaico y hojas de tomate con enfermedad de rizada, las imágenes tuvieron dimensiones de 303 x 404 píxeles, para medir la calidad de segmentación, se aplicó el índice de Davies Bouldin(DB), porque fue una métrica de calidad interna donde los valores pequeños fueron vistos como los clusters compactos y los centros estuvieron bien separados unos de los otros donde el número de clusters que minimizó el índice DB fue tomado como el óptimo, en el resultado obtuvieron un Accuracy de 90% y un tiempo computacional de respuesta de 0.04s. [7]

La presente investigación se realizó con el algoritmo K-MEANS tradicional, donde la segmentación fue de 100 imágenes digitales de plantas de arroz, para determinar los resultados de segmentación se utilizó métricas de calidad externas que derivaron de la matriz de confusión.

Se obtuvo una exactitud del 92.25% y un rendimiento computacional en tiempo de respuesta de 155.14 ms, donde la Exactitud se superó a pesar que fueron más imágenes las procesadas y tuvieron vistas de toma de imágenes, hacia adelante, hacia atrás, en posición de diagonal izquierda y en posición de diagonal derecha, con respecto al tiempo de respuesta retardó más en el procesamiento debido a que las imágenes adquiridas tuvieron demasiado desenfoques, además fueron imágenes tomadas en un ambiente no controlado en día nublado y día

soleado, las imágenes tuvieron dimensiones más robustas, porque fueron de 800 x 600 píxeles.

Por otra parte en la segmentación automática de hojas de plántulas de álamo superpuestas combinando a Mask R-CNN y DBSCAN donde el algoritmo DBSCAN, basado en la toma de imágenes con una cámara RGB-D fue de gran apoyo para segmentar hojas superpuestas, así se logró segmentar hojas de plántulas de álamo superpuestas en condiciones normales y en condiciones estresadas utilizando la distancia múltiple, porque ayudó a lograr características más robustas, y se pudo tener una mejor similitud entre dos puntos de las hojas superpuestas, fue mejor que la distancia euclidiana para segmentar hojas superpuestas, también nos dio a conocer que utilizaron los parámetros de la cámara para convertir información de profundidad en nube de puntos.

La información de nube de puntos tendió a consumir mucho tiempo en procesamiento razón por el cual se combinó con el aprendizaje profundo Mask R-CNN para reducir en gran medida el tiempo de procesamiento. En un trabajo futuro se planeó realizarlo en un entorno de campo porque se obtuvo la recogida de datos RGB-D en condiciones de escena que fueron menos variables debido a que tendió a ser en el interior de un ambiente.

Para demostrar el rendimiento en detección de hojas con mayor eficiencia se utilizó una métrica de evaluación por píxeles de intersección sobre la unión (p-IoU) donde obtuvo resultados en hojas normales de 0.885 y hojas estresadas de

0.874, así también se utilizó mean Accuracy para obtener la exactitud media que fue en hojas normales de 0.897 y en hojas estresadas de 0.888. [5]

En la presente investigación el algoritmo DBSCAN logro segmentar 100 imágenes de plantas de arroz. Las imágenes fueron en posición de diagonal izquierda y en posición de diagonal a la derecha en día soleado, también se adquirieron imágenes hacia adelante e imágenes hacia atrás en día nublado, las imágenes fueron tomadas de un ambiente no controlado.

Se logró obtener una Exactitud total de 54.25%, el cual demostró que no obtuvo una buena exactitud en la segmentación debido a que los pixeles que fueron plantas se confundieron demasiado con los pixeles que no lo eran, así también en las imágenes que tuvieron desenfoques no se logró apreciar bien la imagen de interés segmentada, por esta razón se confundió al momento de realizar el procesamiento de segmentación.

Para lograr obtener con claridad resultados en las plantas pixeles se realizó el cálculo con la métrica de sensibilidad en cual obtuvo un resultado subóptimo de 0%. Con respecto al rendimiento de hardware fue demasiado lento en el procesamiento de la segmentación de imágenes de plantas de arroz porque obtuvo resultados demasiados elevados siendo un tiempo de respuesta de 233.40 Ms, consumo de memoria RAM de 109.24%, consumo de CPU fue de 13.32%.

3.3. Aporte de la investigación

3.3.1. Aporte Practico

– Adquisición de imágenes de plantas de arroz

En la presente investigación las imágenes se adquirieron en Lambayeque en el Fundo Yencala Boggiano, se consideró 50 imágenes en día soleado, obtenidas el 27 de marzo del 2019 a la 1:00 pm, porque fue un horario en donde se tuvo mayor iluminación y existieron cambios en las plantas de arroz como los distintos reflejos de luz solar.

También se obtuvieron 50 imágenes en día nublado el 23 de febrero del 2019 a las 3:00 pm, se estableció dicho horario porque el cielo tendió a cubrirse por el atardecer, existiendo también cambios en las plantas de arroz por factores como el aire y sombras. Se consideró esas fechas porque es la fase inicial del cultivo de arroz y por ser meses donde prevalecieron las lluvias por ser temporada de verano. Se adquirieron un total de 100 imágenes porque con dicha cantidad se logró superar el número de estadísticas para que el resultado sea significativo.

Las imágenes se adquirieron con una dimensión de 4320 X 3240 Pixeles, porque cuando más pixeles tenga en la imagen se obtendrá una mejor información y será más precisa. Tuvieron una resolución horizontal de 180 puntos por pulgada y una resolución vertical de 180 puntos por pulgada, respecto a la resolución tendió a referirse al número de pixeles que fueron formados en líneas para ser mostrados en pantalla, cuando es la resolución horizontal las líneas fueron mostradas verticalmente y en cuanto la resolución sea vertical las líneas fueron visualizadas horizontalmente, con

ello se tuvo en cuenta que cuanto mayor sean las líneas horizontales y verticales se obtuvo una mejor calidad en las imágenes digitales.

También las imágenes adquirieron una profundidad de 24 bits, donde la profundidad hizo referencia a las imágenes RGB porque cada canal fue equivalente a 8 bits, así también tuvo una representación de color sRGB, con esta representación las imágenes fueron más saturadas y claras es por ello que se tomó en consideración porque se mostraron de manera más consistente, la unidad de resolución fue de dos pixeles y bits comprimido fue también de dos pixeles, al comprimirse tuvo el beneficio de reducirse el número de bytes y lograron ocupar un menor espacio, el Punto F fue de $f/3.4$, donde represento a la abertura máxima o luminosidad con las que se pudieron capturar las imágenes digitales con el resultado se visualizó que tuvo más luz, y menos enfoque o profundidad en el campo porque las imágenes fueron más desenfocadas, también logro tener un tiempo de exposición de $1/80$ s y/o de $1/100$ s, siendo el tiempo en que la luz logró alcanzar hacia el sensor digital para poderse realizar la fotografía, el tiempo de exposición estuvo expuesto en segundos lo que quiere decir que la condición de luz fue muy mala, mantuvieron una velocidad ISO-80.

Fue el encargado de controlar la sensibilidad de la película o del sensor que absorbió la luz, cuanto mayor sea el ISO se tendrá más luz y las imágenes serán más ruidosas por el contrario si el ISO fue menor se tendrá menos luz y menor será el ruido en imágenes, en nuestro caso se tomó el ISO mínimo que tuvo el dispositivo de adquisición de imágenes digitales, se

obtuvo también una distancia focal de 5 milímetros y una longitud focal de 35 milímetros, fueron la representación del sensor del dispositivo de adquisición y el centro óptico del lente fotográfico, en nuestras imágenes nuestra distancia focal fue corta y por ende capturo más escena debido a que la altura de cámara fue colocada a 1.10 mts de distancia como se puede observar en la figura N° 23, Las imágenes adquirieron una apertura máxima de 3.53125, la apertura nos ayudó a determinar el brillo y oscuridad del sensor del objetivo mediante el diafragma de la imagen, las imágenes fueron capturadas sin flash porque en los horarios de adquisición tenía distintas iluminaciones de radiación solar.

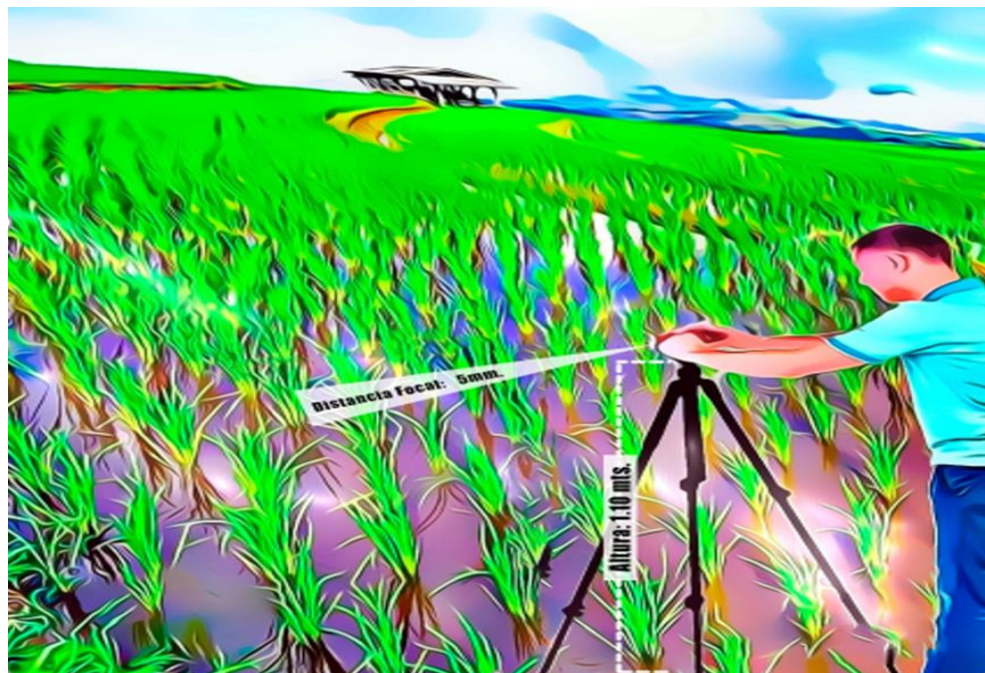


FIGURA 25: ADQUISICIÓN DE IMÁGENES DE PLANTAS DE ARROZ. FUENTE: ELABORACIÓN PROPIA.

TABLA 9.**CONFIGURACIÓN DEL DISPOSITIVO DE ADQUISICIÓN DE IMÁGENES.**

Característica	Detalle
Dimensión de imagen	4320 X 3240 px.
Resolución horizontal	180 ppp.
Resolución vertical	180 ppp.
Profundidad de imagen	24 bits.
Representación de color	sRGB.
Unidad de resolución	2.
Bits comprimidos	2 px.
Punto F	F/3.4.
Tiempo de exposición	1/80 – 1/100 s.
Velocidad ISO	ISO – 80.
Distancia focal	5 mm.
Longitud Focal	35 mm.
Apertura Máxima	3.53125.
Modo de Flash	Sin flash.

Fuente: Elaboración Propia.

Las formas y cantidades en que se adquirieron las imágenes digitales fueron como se muestra en la figura N° 24 siendo: a. 25 imágenes en posición de diagonal izquierda, b. 25 imágenes en posición de diagonal derecha, c. 25 imágenes hacia atrás y d. 25 imágenes hacia adelante, se

realizaron capturas de imágenes en distintas perspectivas de enfoque hacia el arrozal, con el fin de lograr determinar en distintas posiciones las formas de las plantas de arroz. Se debe tener en consideración que las imágenes adquiridas hacia adelante y las imágenes adquiridas hacia atrás fueron tomadas en día nublado, mientras que las imágenes adquiridas en posición diagonal izquierda y las imágenes adquiridas en posición diagonal derecha fueron tomadas en día soleado.

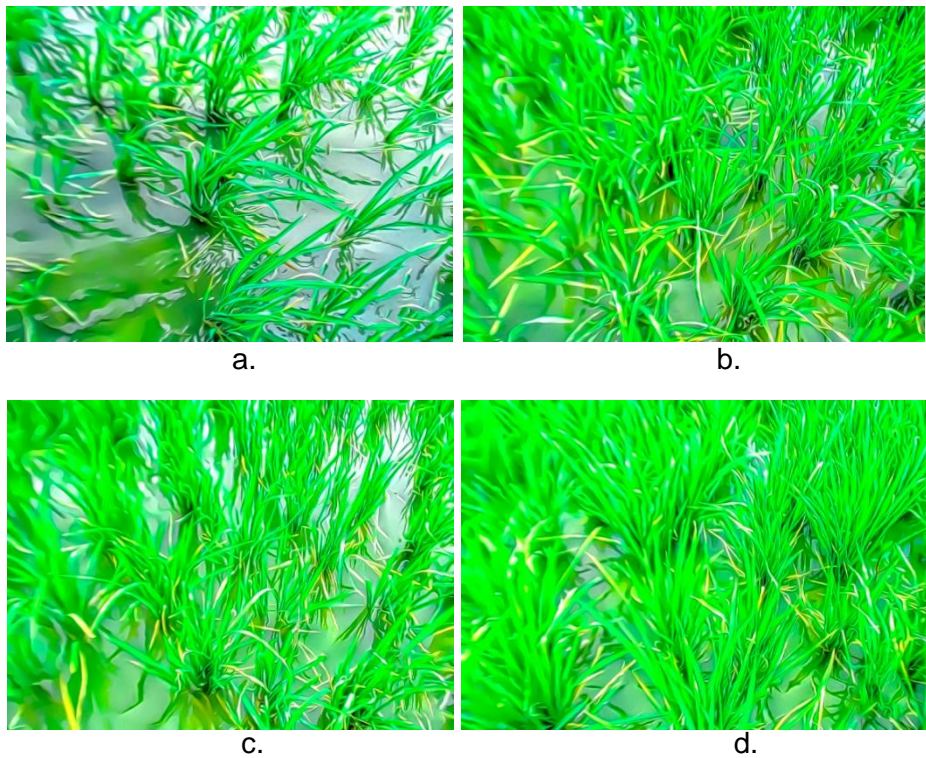


FIGURA 26: FORMAS DE ADQUISICIÓN Y CANTIDADES DE IMÁGENES DE PLANTAS DE ARROZ. FUENTE: ELABORACIÓN PROPIA.

- a. 25 imágenes en diagonal izquierda.
- b. 25 imágenes en diagonal derecha.
- c. 25 imágenes hacia atrás.
- d. 25 imágenes hacia adelante.



FIGURA 27: DISPOSITIVO DE ADQUISICIÓN DE IMÁGENES DIGITALES. FUENTE: ELABORACIÓN PROPIA.

TABLA 10.

CARACTERÍSTICAS DEL DISPOSITIVO DE ADQUISICIÓN DE IMÁGENES DIGITALES.

Característica	Detalle
Canon	PowerShot SX150 IS.
Tamaño de pantalla	3 pulgadas.
Altura	113.3 mm.
Ancho	73.2 mm
Espesor	45.8 mm.
Megapíxeles	14.1 mpx.
Resolución	553 X 415 px.
Densidad de píxeles	230 ppi.
Zoom óptico	12x.

Flash	Incorporado.
Longitud focal mínima	28 mm.
Longitud focal máxima	336 mm
Pixel	3 subpíxeles.
Cantidad total de luz permitida en el sensor de imagen	15 sg.
Máxima sensibilidad a la luz	1600 ISO.

FUENTE: ELABORACIÓN PROPIA.

– **Imágenes de plantas de arroz**

La base de datos es una carpeta donde se alojó un total de 100 imágenes con extensión .jpg, la extensión en mención nos permitió ahorrar gran cantidad de almacenamiento, razón por la cual las imágenes adquiridas tuvieron capacidades aproximadas de 1.10 MB a 2.75 MB, fueron tomadas en día soleado y día nublado, de las cuales se utilizaron 25 imágenes de plantas de arroz capturadas en posición de diagonal hacia el lado izquierdo, 25 imágenes de plantas de arroz en posición de diagonal del lado derecho, 25 imágenes de plantas de arroz en posición hacia adelante y 25 imágenes de plantas de arroz en posición hacia atrás como se logró observar las formas de adquisición en la figura N° 24, las imágenes que fueron mencionadas son utilizadas para la segmentación y obtención de los resultados según al horario en que fueron adquiridas.

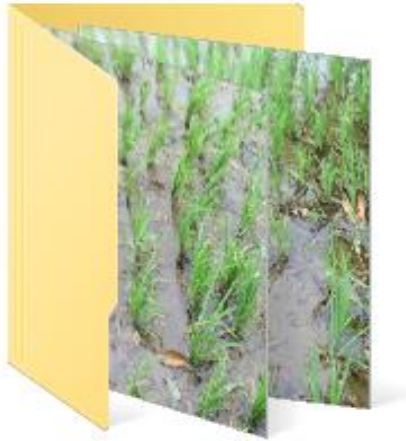


FIGURA 28: BASE DE DATOS DE IMÁGENES DE PLANTAS DE ARROZ. FUENTE: ELABORACIÓN PROPIA.

– **Técnicas de segmentación en imágenes digitales**

Para la selección de los algoritmos de segmentación de imágenes de plantas de arroz se consideró ocho trabajos previos, porque en los últimos cinco años se publicaron investigaciones relacionados directamente a segmentación en escenarios de agricultura de precisión que coincidieron en el área de interés de esta investigación, donde muestran las precisiones y rendimientos de los algoritmos que fueron utilizados, a continuación, se presentaron en la siguiente tabla:

TABLA 11: COMPARACIÓN DE ALGORITMOS DE SEGMENTACIÓN.

Autores	Titulo	Algoritmos de segmentación	Criterios		Evaluación de rendimiento	Escenario	Algoritmos para segmentación de plantas de arroz
			Rendimiento	Precisión			
Xuan Liu, Chunhua Hu, Pingping Li.	Segmentación automática de hojas de plántulas de álamo superpuestas que combinan Mask R-CNN y DBSCAN.	Mask RCNN	-	Obtuvo el 89.7% en hojas normales que no se superponían con otras.	Top 2	Realizado en el Laboratorio de ingeniería forestal situado en Universidad Forestal de Nanjing china.	No
		DBSCAN	-	Obtuvo el 98.5% en hojas normales que se superponían	Top 1		Si

				unas hojas con otras.			
Kai Tian, Jiu hao Li, Jiefeng Zeng, Asenso Evans, Lina Zhang.	Segmentación de imágenes de hojas de tomate basada en el número de clustering adaptativo del algoritmo K-means.	K-MEANS	En imágenes de 303 x 404 píxeles retardo 0.04 Seg.	Obtuvo el 98.2% en hojas de tomate con enfermedad de hoja amarilla y 90% en hojas de tomate con enfermedad de mancha marrón.	Top1	Realizado en la facultad de la conservación del agua e ingeniería civil, universidad agrícola del sur Guangdong.	SI
Yang Yu, Kailiang Zhang, Li Yang, Dongxing Zhang.	Detección de fruta para un robot recolector de fresas en un entorno no estructural basado en Mask-RCNN.	Mask RCNN	-	Obtuvo el 89.85% en segmentar fresas donde utilizaron un	Top 2	Se realizo en la Facultad de Ingeniería de la Universidad Agrícola de China.	No

				robot recolector de frutas.			
Shanwen Zhanga, Haoxiang Wang, Wenzhun Huang, Zhuhong Youa.	Segmentación y reconocimiento de hojas enfermas de plantas mediante fusión de superpixel, K-means y PHOG.	K-MEANS	-	Obtuvo el 90.43% en las hojas de manzana con enfermedad de mosaico y 92.15% en las hojas de pepino con enfermedad de antracnosis.	Top 1	Se adquirieron en los campos y se llevaron al laboratorio en la ciudad de Yangling, provincia de Shanxi, China	Si
Juan Liao, Yao Wang, Junnan Yin, Lu Liu, Shun Zhang, Dequan Zhu.	Segmentación de las plántulas de arroz utilizando el Espacio de color YCrCb y un método Otsu mejorado.	OTSU	En imágenes de 320 x 237 pixeles retardo 0.003 Seg.	-	Top 3	El experimento se llevó en un campo de arroz sembrado en el sur de china.	No

Guofeng Tong, Yong Li, Huairong Chen, Qingchun Zhang, Huiying Jiang.	Red U-NET mejorada para la segmentación de nódulos pulmonares	RED UNET MEJORADA.	-	Obtuvo el 73% en segmentar imágenes de nódulos pulmonares tipo Tomografía computarizada donde eran muy pequeños y podían confundirse con otros tejidos.	Top 3	realizado en escuela de Ciencias e Ingeniería de la Información de Shenyang, Universidad del Noreste China.	No
---	--	-----------------------	---	--	-------	---	----

Wei Guo, Bangyou Zheng, Tao Duan, Tokihiro Fukatsu, Scott Chapman, Seishi Ninomiya.	EasyPCC: conjunto de datos y herramientas de referencia para medición de alto rendimiento del dosel de la planta ratios de cobertura en condiciones de campo.	DTSM	Tuvo un retardo de 30 seg, en imágenes afectadas por el viento y vegetación aplastada por daños desconocidos.	Obtuvo el 76% cuando la vegetación no tuvo muchas áreas oscuras y el 79% cuando la vegetación en la planta tuvo un área oscura más grande.	Top 3	Realizaron segmentación del dosel vegetal a plántulas de arroz en condiciones de campo o de iluminación natural.	No
--	--	------	--	---	-------	---	----

Yanan Li, Zhiguo Cao, Hao Lu, Yang Xiao, Yanjun	Detección de algodón en campo mediante segmentación semántica de	SLIC	-	Obtuvo una sensibilidad de 77.3%, una especificidad de 99.3% y una exactitud de 97.0% para	Top 2	Cultivo de algodón bajo iluminación natural obtenidas en la Región Autónoma	No
---	--	------	---	--	-------	---	----

Zhua, Armin B. Cremers.	imágenes basada en regiones.	segmentar solo el algodón de otros elementos que existieron en la imagen como las ramas, suelo, hojas, plásticos y el cielo.	Uygur de Xinjiang en China.
----------------------------	---------------------------------	--	-----------------------------------

Fuente: Elaboración Propia.

En la tabla N.º 11 se realizó la descripción y análisis de los algoritmos de segmentación de imágenes digitales que se utilizaron en distintas investigaciones científicas.

En la tabla en mención se mostró a los autores del artículo, el título, los algoritmos utilizados, los criterios utilizados tanto en rendimiento y precisión, evaluación del rendimiento donde se consideró si fue Top 1, Top 2 o Top 3, así también el escenario es decir donde fue desarrollado la investigación y por último la mención de si o no de los algoritmos con que se optó para realizar la segmentación de imágenes digitales de plantas de arroz.

A continuación, se detalla las escalas de rendimiento definidas por el autor del proyecto de investigación:

Top1: Algoritmo de segmentación que demostró ser la mejor opción en precisión y rendimiento.

Top2: Algoritmo de segmentación que demostró ser en segundo plano el mejor en precisión y rendimiento.

Top3: Algoritmo de segmentación que demostró tener el rendimiento y precisión más bajo.

Luego de haber puntuado a los algoritmos de segmentación de cada investigación según su rendimiento y segmentación se consideraron de la siguiente manera:

– **Algoritmos considerados como top 1:**

Se consideró y se seleccionó a los algoritmos DBSCAN Y KMEANS basados en clusterización o conocidos como algoritmos no supervisados porque se requirió de características de color y textura para poder obtener áreas de interés dentro de las imágenes digitales de plantas de arroz.

Por otra parte, DBSCAN obtuvo una buena precisión en segmentar hojas de plántulas de álamo especialmente aquellas que se superponían unas a otras es por ello que fue considerado dentro de esta investigación debido a que las plantas de arroz son muy flexibles a los factores externos como el aire y también tendieron a superponerse la diversidad de hojas que existieron en la planta.

Así también se optó por el algoritmo k-means porque obtuvo buenas precisiones de segmentación en hojas enfermas de tomate y para detectar enfermedades en hojas de manzana y de pepino, por lo tanto, fue utilizado para segmentar imágenes digitales que tuvieron expuestas a un ambiente no controlado es decir imágenes adquiridas en campo o al aire libre donde se tiene diferentes condiciones de iluminación.

Estos algoritmos de Machine Learning por ser no supervisados no necesitan de una fase de entrenamiento para poder realizar la segmentación de las imágenes digitales de plantas de arroz es por esta razón que se eligieron en esta investigación, pero si debe considerarse mínimo 100 imágenes digitales de plantas de arroz para que sus resultados estadísticos de segmentación tengan un valor significativo.

Además, fueron considerados los algoritmos en mención debido a que en los últimos años no se ha realizado investigaciones en segmentaciones de plantas de arroz como se demostró en el anexo N° 3.

– **Algoritmos considerados como top 2:**

El algoritmo Mask Rcnm no es considerado en esta investigación porque obtiene segmentaciones obtenidas por debajo del top 1 como las investigaciones que realizó en segmentaciones de hojas de álamo normales que no se superponían unas de otras y segmentaciones en fresas por robot recolector de frutos, además para realizarlo se necesitó de una fase de entrenamiento por ser un algoritmo supervisado por esta razón se debe tener diversidad de imágenes digitales para realizarlo y también requerimiento de hardware robusto en una computadora personal, no cumpliendo en esta investigación las expectativas para ponerlo en funcionamiento.

El algoritmo SLIC también no fue considerado debido a que estaban sus precisiones por debajo del Top 1, otra razón fue que se hibrida mucho con otros algoritmos para poder obtener resultados significativos en la segmentación de imágenes digitales.

– **Algoritmos considerados como top 3:**

El algoritmo OTSU basado en umbrales no fue considerado porque existió segmentaciones de filas de plántulas de arroz si bien es cierto obtuvo imágenes adquiridas en un ambiente no controlado es decir en campo al aire libre pero no tenía la cantidad de imágenes para obtener un

valor significativo en los resultados, además solo se midió el rendimiento o tiempo de respuesta de imagen por imagen.

La RED UNET MEJORADA es un algoritmo basado en la segmentación semántica conocido también como algoritmo supervisado, fue utilizado para segmentar nódulos pulmonares tipo TC, tampoco fue considerado en esta investigación debido a que obtuvo resultado sub óptimo y además requirió de diversidad de imágenes digitales para realizar una fase de entrenamiento por lo tanto no cumplió con las expectativas en el desarrollo de esta investigación.

El algoritmo DTSM es un algoritmo basado en árbol de decisión y se utilizó para segmentar dosel vegetal de plantas de arroz obtuvo un rendimiento sub óptimo debido a las diferentes iluminaciones que se reflejaban en las hojas además tampoco es considerado porque ya existió investigación en segmentar plantas de arroz como se mostró en el anexo N° 3 y también por ser un algoritmo supervisado donde requirió de diversidad de imágenes digitales para poder realizar su fase de entrenamiento.

Algoritmos de segmentación de imágenes de plantas de arroz.

En el siguiente esquema de segmentación se muestra el proceso para lograr segmentar plantas de arroz en ambientes no controlados utilizando algoritmo K-MEANS.

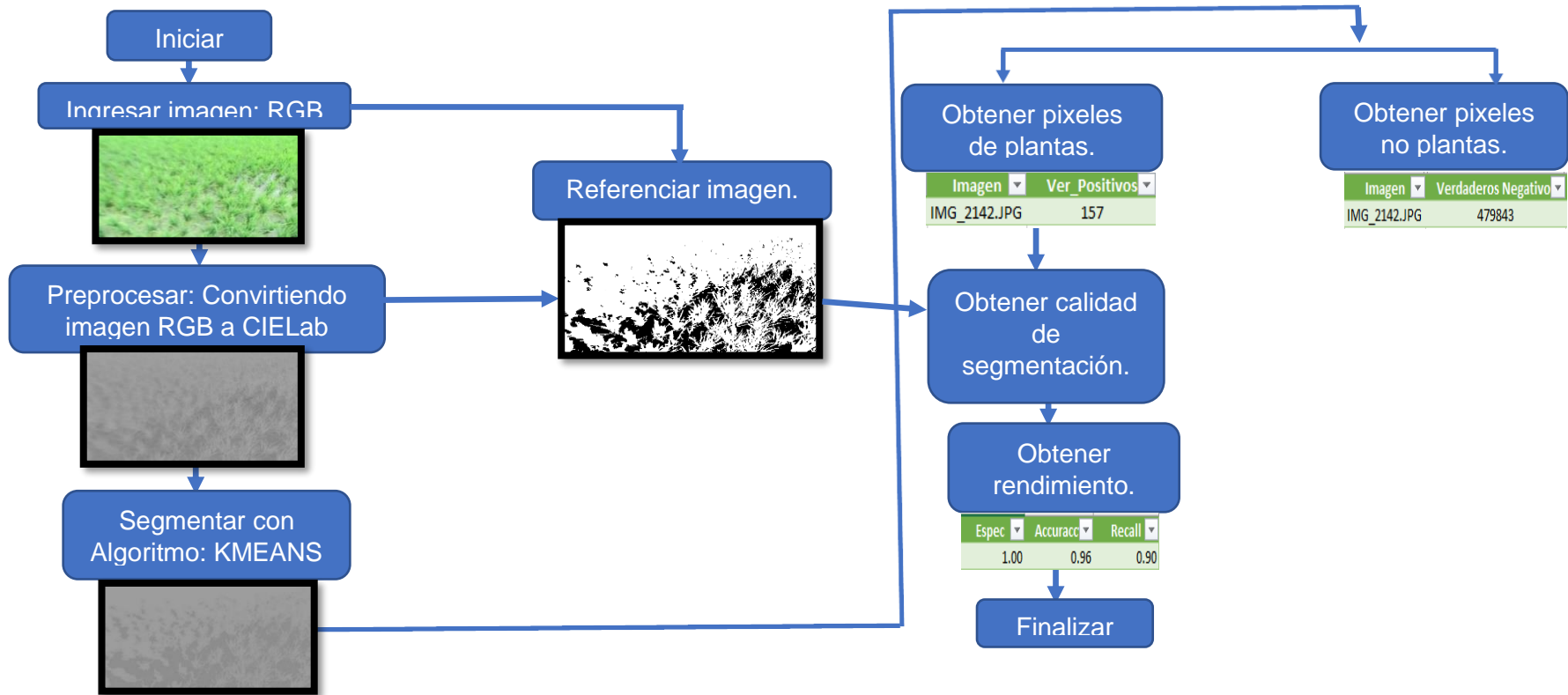


FIGURA 29: ESQUEMA DE SEGMENTACIÓN DE PLANTAS DE ARROZ EN AMBIENTE NO CONTROLADO UTILIZANDO ALGORITMO K-MEANS.

FUENTE: ELABORACIÓN PROPIA.

Se utilizó lenguaje de programación Python versión 3.7. para el desarrollo del proceso de segmentación de imágenes digitales de plantas de arroz.

Se implementó las siguientes librerías para el desarrollo de los algoritmos de segmentación.

- **OS:** Fue utilizado para la manipulación de nombres en las rutas y poder acceder a archivos del sistema.

- **Times:** Fue utilizado para manifestar los tiempos y tendió a realizar algunas conversiones.

- **PIL:** Denominado python imaging library nos permitió realizar la edición de imágenes.

- **Image:** Fue un objeto de PIL que nos permitió la manipulación de imágenes.

- **Tkinter:** Fue utilizado para la realización de la interfaz de usuario donde utilizo kit de herramientas GUI TK.

- **Types:** Fue un aporte en la creación de realizar dinámica de tipos de datos nuevos.

- **LambdaType:** Fueron denominados formas lambda y se usaron para crear funciones anónimas.

- **matplotlib:** Fue utilizado para la creación de los gráficos en dos dimensiones.
- **FileDialog:** Proporciono clases y funciones para crear ventanas de selección en los archivos y directorios a utilizar.
- **Cv2:** Se utilizó para el procesamiento de imágenes y segmentación de las imágenes digitales.
- **Pandas:** Es una extensión de Numpy, que permitió realizar la manipulación de análisis de datos.
- **umetrics:** Fue utilizada para realizar las estadísticas de segmentación de imágenes digitales tales como Exactitud, sensibilidad, especificidad. La librería umetric tuvo la matriz de confusión para poder realizar dichas métricas de calidad.
- **Imutils:** Conjunto de funciones que permitió realizar tareas básicas de procesamiento de imágenes de manera más fácil y sencilla.
- **Sklearn:** Fue utilizado para la implementación del algoritmo DBSCAN basado en clusterización.
- **StandardScaler:** Estandarizó las características dando por eliminar la media y realizar el escalado a una varianza unitaria.
- **Numpy:** Fue utilizado para realizar cálculos numéricos matemáticos y también se utilizó en el análisis de datos.

- **Index:** Fue utilizado para encontrar la aparición de un valor que fue especificado en una lista.
- **Psutil:** Fue un módulo que brindó una interfaz para obtener datos de un determinado proceso y su conexión con el sistema.
- **Skimage:** Fue utilizado para operaciones realizadas de entrada y salida de las imágenes.
- **Webbrowser:** Permitió mostrar un documento que fue ejecutado localmente en la web.

```

from os import times
from PIL import Image
import tkinter as tk
from types import LambdaType
import matplotlib.pyplot as plt
from tkinter import filedialog
from tkinter.constants import MULTIPLE
from tkinter import *
import cv2, pandas, umetrics, imutils, os, time
from sklearn.preprocessing import StandardScaler
from numpy.core.defchararray import index
import psutil as ps
from numpy.core.shape_base import block
from sklearn.cluster import DBSCAN
import numpy as np
from skimage.transform import resize
import webbrowser

```

FIGURA 30: LIBRERÍAS UTILIZADAS EN PYTHON VERSIÓN 3.7. FUENTE: ELABORACIÓN PROPIA.

Se inició con el ingreso de una imagen RGB de plantas de arroz, donde la imagen obtenida tuvo una dimensión de 4320 x 3240 pixeles y fue redimensionada a 800 x 600 pixeles por la carga computacional que las imágenes adquiridas tuvieron.



FIGURA 31: IMAGEN ORIGINAL DE PLANTAS DE ARROZ EN AMBIENTE NO CONTROLADO. FUENTE: ELABORACIÓN PROPIA.

El ingreso de las imágenes de plantas de arroz a color que fueron tomadas en un ambiente no controlado y tuvieron dimensiones de 4320 x 3240 pixeles fueron alojadas en una ruta específica, a continuación, se mostró en la siguiente codificación.

```
self.lista=filedialog.askopenfilename(title='Seleccionar  
Imágenes De Plantas De Arroz', initialdir='./',  
filetypes=['Imágenes *.jpg'], multiple=True)
```

FIGURA 32: CODIFICACIÓN DEL INGRESO DE IMÁGENES DE PLANTAS DE ARROZ. FUENTE: ELABORACIÓN PROPIA.

Se redimensionó las imágenes de plantas de arroz originales a dimensiones de 800 x 600 píxeles por la carga computacional que éstas tuvieron por ser de tamaño muy robusto.

```
image=self.imagenes_b[i]
dsize=(800,600)
imagen_4=cv2.resize(image,dsize)
```

FIGURA 33: REDIMENSIÓN DE LAS IMÁGENES DE PLANTAS DE ARROZ. FUENTE: ELABORACIÓN PROPIA.

Después de haberse redimensionado las imágenes digitales de plantas de arroz se realizó la conversión de las imágenes RGB a un espacio de color CIE Lab:

Se observó que X, Y y Z fueron las coordenadas que se utilizó en el espacio de color CIE XYZ y se posicionaron linealmente en el espacio de color RGB porque X, Y y Z tuvieron una mejor percepción en color para la visualización de todos los colores percibidos por el ojo humano. Donde Y hizo referencia a la luminancia con respecto a G. Luego X y Z hicieron referencia a la crominancia con respecto a R y B.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.355580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

FIGURA 34: FORMULA DE ESPACIO DE COLOR RGB. FUENTE:

X y Z fueron las coordenadas que tomaron los valores del blanco de referencia que se tuvo en la luminancia X_n , Z_n ; donde X_n tuvo que ser igual a 0.950456 y Z_n también tuvo el valor de 1.088754. para poder obtener el iluminador acromático blanco especificado el cual fue almacenado en X y Z.

$$X \leftarrow \frac{X}{X_n}, \text{ where } X_n = 0.950456$$

$$Z \leftarrow \frac{Z}{Z_n}, \text{ where } Z_n = 1.088754$$

FIGURA 35: BLANCOS DE REFERENCIA EN LA LUMINANCIA. FUENTE:

CIE Lab obtuvo el iluminante estándar D65, es de uso común que fue definida por la comisión internacional de iluminación (CIE) la serie D represento condiciones de iluminación estándar al aire libre en distintas partes del mundo es por esa razón que es muy requerido en estudios de imágenes digitales de ambientes no controlados estas imágenes tendieron a tener diferentes cambios de iluminación en el exterior, es por eso que se realizó la conversión a un espacio de color donde permitió cuantificar la diferencia visual en una imagen de entrada RGB.

Para encontrar la luz percibida se tomó la luminancia en "Y" donde si $Y > 0.008856$ será reemplazada Y en $116 * Y^{\frac{1}{3}} - 16$ por otro lado si $Y \leq 0.008856$ entonces Y será reemplazada en $903.3 * Y$, así se logró obtener la luminosidad en L.

$$L \leftarrow \begin{cases} 116 * Y \frac{1}{3} - 16 & \text{for } Y > 0.008856 \\ 903.3 * Y & \text{for } Y \leq 0.008856 \end{cases}$$

FIGURA 36: LUMINOSIDAD EN COORDENADA L. FUENTE:

Por lo tanto, en las dimensiones a y b tuvieron referencia a la suma de delta, ayudándonos a determinar la diferencia de color tanto en la coordenada a como también en la coordenada b.

$$a \leftarrow 500(f(X) - f(Y)) + delta$$

$$b \leftarrow 200(f(Y) - f(Z)) + delta$$

FIGURA 37: DIFERENCIAS DE COLOR EN COORDENADAS A Y B. FUENTE: [45]

La función f(t) mostró dos dominios, fueron realizados para prevenir una pendiente infinita. Donde en el primer dominio $t \frac{1}{3}$ se realizó si $t > 0.008856$ por el contrario en el segundo dominio $7.787t + \frac{16}{116}$ se realizó si $t \leq 0.008856$.

$$f(t) = \begin{cases} t \frac{1}{3} & \text{for } t > 0.008856 \\ 7.787t + \frac{16}{116} & \text{for } t \leq 0.008856 \end{cases}$$

FIGURA 38: FUNCIÓN DE PREVENCIÓN DE PENDIENTE INFINITA. FUENTE: [45]

y delta tomó el valor de 128 si la imagen es de 8 bits por el contrario tomó el valor de 0 si la imagen tiene puntos flotantes.

$$\text{delta} = \begin{cases} 128 & \text{for } 8\text{-bit images} \\ 0 & \text{for floating-point images} \end{cases}$$

FIGURA 39: FUNCIÓN DELTA. FUENTE: [45]

los valores se convirtieron al tipo de datos de destino tanto para la coordenada L y así mismo también para las coordenadas a y b.

$$0 \leq L \leq 100, -127 \leq a \leq 127, -127 \leq b \leq 127$$

FIGURA 40: VALORES CONVERTIDOS AL TIPO DE DATO DESTINO. FUENTE: [45]

CIELab tuvo 3 dimensiones con los cuales permitió realizar la diferencia, entre los parámetros que fueron la luminosidad, donde sus valores asignados oscilaron entre 0 y 100 donde 0 identificó el color oscuro o negro y 100 identificó color brillante o blancos, sus posiciones entre los colores fueron las coordenadas a* y b* donde: a* identificó rojo y verde, siendo los valores positivos +a refirió al color rojo, mientras que en los valores negativos de -a refirió al color verde, luego la posición de b* identificó los colores amarillo y azul donde: la coordenada +b identificó al color amarillo, -b identificó al color azul. Las imágenes de 8 bits convertidas a CIELab quedaron de la siguiente manera:

$$L \leftarrow L * \frac{255}{100}, \quad a \leftarrow a + 128, \quad b \leftarrow b + 128$$

FIGURA 41: FORMULA DE ESPACIO DE COLOR CIELAB. FUENTE: [45]

En la figura N° 42, la función `cv2.cvtColor_BGR2Lab` nos permitió convertir las imágenes de ingreso RGB a espacio de color CIELab, Se realizó la conversión porque las imágenes de plantas de arroz fueron adquiridas en un ambiente no controlado y tenían ruidos y distintas iluminaciones en diversas partes de su entorno.

```
labimg=cv2.cvtColor(img,cv2.COLOR_BGR2Lab)
```

FIGURA 42: CONVIRTIÓ LAS IMÁGENES RGB A ESPACIO DE COLOR CIELAB. FUENTE: ELABORACIÓN PROPIA.

Se visualizó en la figura N° 43 la ejecución de imagen original RGB, Dicha imagen fue convertida a espacio de color CIELab y fue descompuesta en sus dimensiones L,a,b donde: L fue mostrado en la figura N°44 , a se mostró en la figura N° 45 y por ultimo b se mostró en la figura N°46, se descompuso en sus tres dimensiones para determinar cuál de los 3 canales tuvo mejor comportamiento visual de plantas y no plantas de arroz, Donde se observó que la dimensión que tuvo un mejor comportamiento visual fue la dimensión b porque se pudo diferenciar las plantas de arroz de los demás factores que existieron en la imagen como agua y barro.

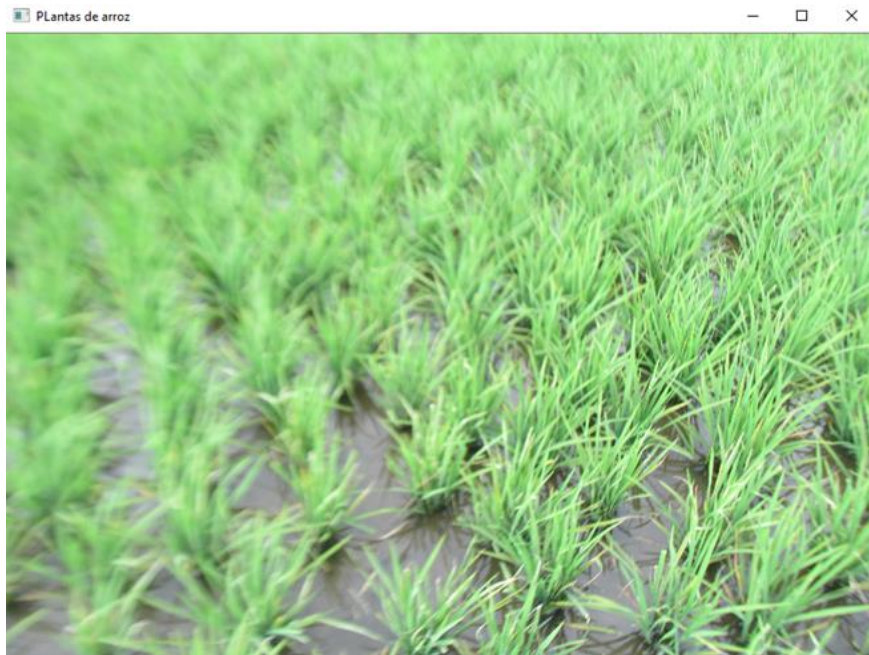


FIGURA 43: RESULTADO DE IMAGEN ORIGINAL DE PLANTAS DE ARROZ. FUENTE: ELABORACIÓN PROPIA.

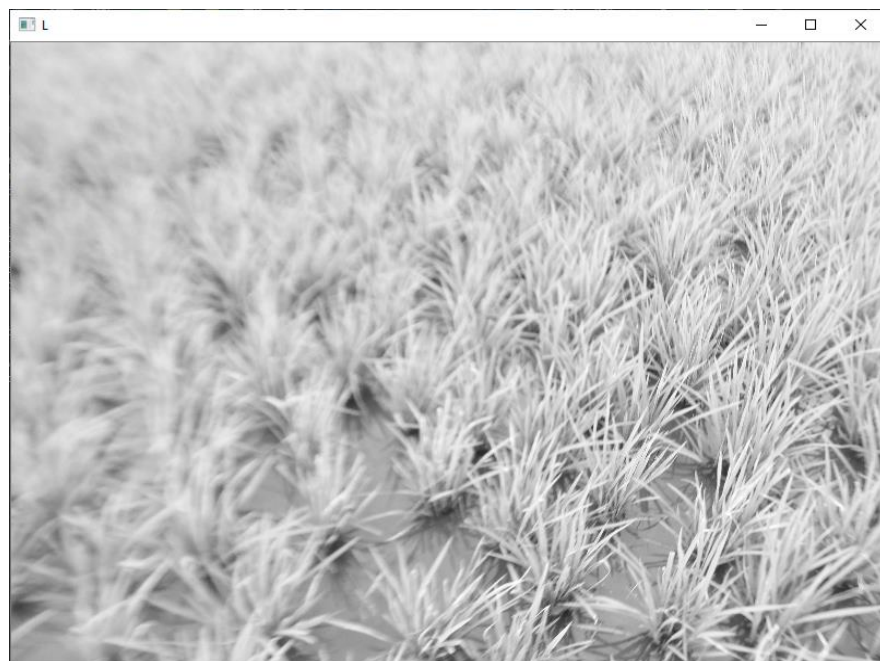


FIGURA 44: RESULTADO DE IMAGEN DE PLANTAS DE ARROZ EN SU DIMENSIÓN L. FUENTE: ELABORACIÓN PROPIA.

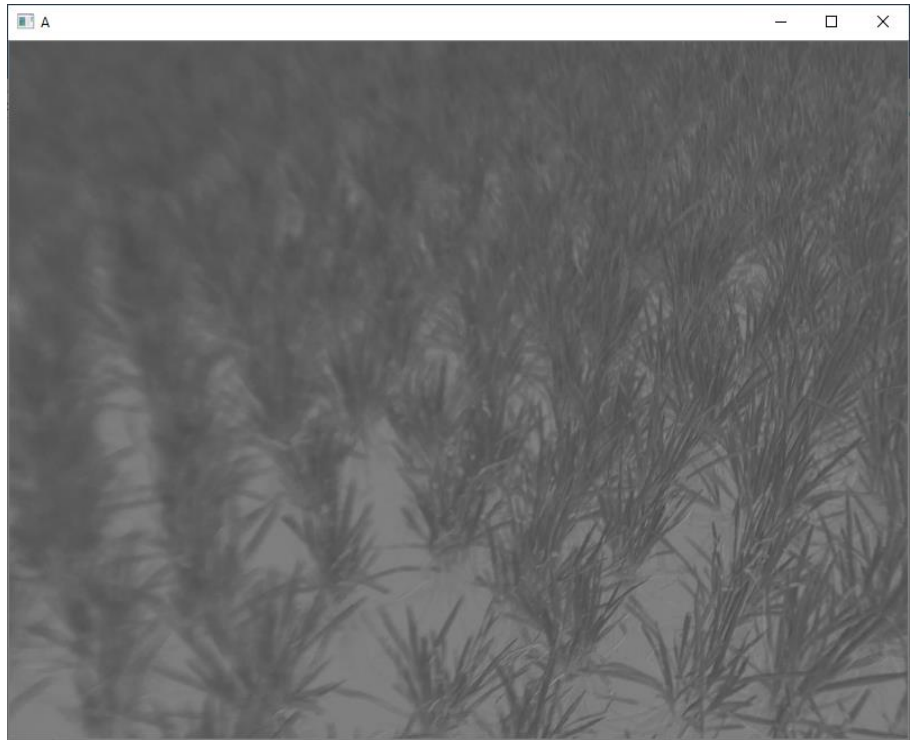


FIGURA 45: RESULTADO DE IMAGEN DE PLANTAS DE ARROZ EN SU DIMENSIÓN A. FUENTE: ELABORACIÓN PROPIA.

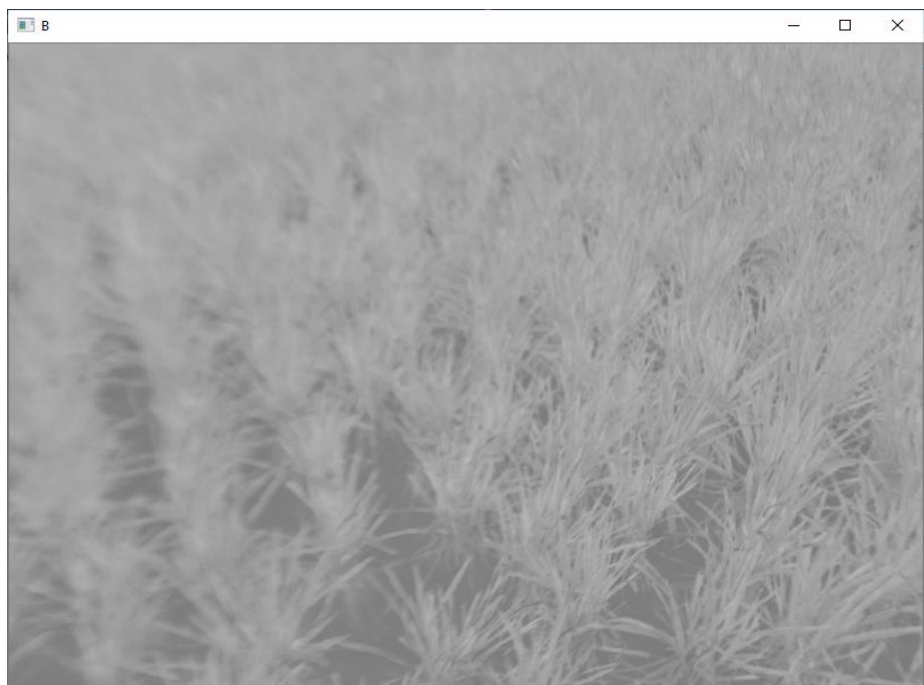


FIGURA 46: RESULTADO DE IMAGEN DE PLANTAS DE ARROZ EN SU DIMENSIÓN B. FUENTE: ELABORACIÓN PROPIA.

La dimensión b se mostró en la figura N°46, y fue elegida porque obtuvo una mejor performance, distinguiéndose las plantas de arroz, de otros factores que no fueron arroz, los elementos que no sé detectaron como arroz en la imagen estuvieron diferenciados de un color más oscuro.

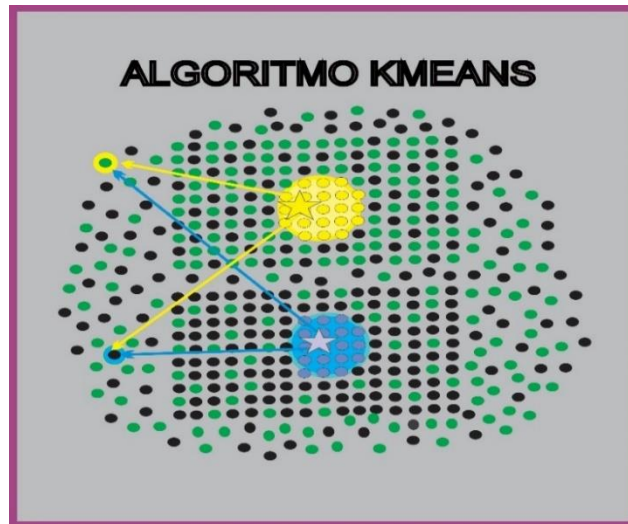


FIGURA 47: ALGORITMO KMEANS K=2. FUENTE: ELABORACIÓN PROPIA.

En la figura N°47 se observó que para realizar la segmentación de imágenes de plantas de arroz se consideró 2 áreas de interés donde el área de los puntos verdes fueron pixeles de plantas y las áreas de puntos negros fueron pixeles no plantas, en esta investigación se consideró que los centroides fueron aleatorios, siendo $K=2$, donde dichos centroides tomaron datos de pixeles que tuvieron similitudes en sus características, así también para definir los pixeles a que centroides podían pertenecer se consideró la distancia euclidiana lo que permitió saber que tan cerca podía encontrarse un pixel del otro y así determinar a qué centroide podía asignarse. El análisis se realizó hasta asignar los mismos pixeles a cada grupo o centroide en rondas consecutivas es decir

cuando empezamos a obtener resultados repetidos en análisis consecutivos el algoritmo kmeans convergió y mostró los mejores resultados.

A continuación en la figura N°48 se utilizó la función reshape para redimensionar la imagen a un vector de 3 puntos porque se representó los valores de los pixeles de la imagen en el espacio de color, así también se utilizó la dimensión establecida en -1 con ello pudo determinarse automáticamente al inferior los valores especificados en otras dimensiones siendo útil cuando se tiene una imagen que estableció una matriz demasiado grande. Luego se convirtió la matriz de la imagen redimensionada de plantas de arroz a números flotantes de 32 bits para después determinar el criterio o condición de parada que realizara el algoritmo, donde se estableció dos cláusulas una de ellas fue el número de iteraciones TERM_CRITERIA_MAX_ITER, fue el que realizó efecto de no continuar si logra alcanzar como máximo 10 iteraciones, y el épsilon definido como TERM_CRITERIA_EPS que fue establecido en 1.0 haciendo efecto de no continuar si el número de distancia hubiese sido muy pequeña. por último, number_of_attempts especificó el número de intentos lo que quiere decir que openCV va a ejecutarse 10 veces en el algoritmo K-MEANS con 10 inicializaciones diferentes de los centroides donde al final nos reportó el mejor resultado de estas 10 ejecuciones.

```
pixel_values = image_copy.reshape((-1, 3))
pixel_values = np.float32(pixel_values)

stop_criteria = (cv2.TERM_CRITERIA_EPS +
cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)

number_of_attempts = 10
```

FIGURA 48: CRITERIOS UTILIZADOS PARA REALIZAR LA SEGMENTACIÓN DE IMÁGENES DE PLANTAS DE ARROZ. FUENTE: ELABORACIÓN PROPIA.

Una vez que nos mostró el mejor resultado, en la figura N°49 se utilizó la constante `KMEANS_RANDOM_CENTERS`, dicha constante nos inicializó los centroides de manera aleatoria, después se ejecutó K-MEANS con los parámetros: `pixel_values` fue el que realizó el arreglo de pixeles de donde saldrán los grupos, luego se estableció 2, que fueron el número de clústers, que nos permitió buscar en las imágenes áreas de interés como plantas de arroz que los diferenció de otros factores como agua o barro, se estableció `None` porque no pasaremos un arreglo opcional de las mejores etiquetas, `stop_criteria` fue la condición de parada, `number_of_attempts` determinó el número de veces que se va a ejecutar el algoritmo K-MEANS, `centroid_initialization_strategy` fue la estrategia con que se inicializaron los centroides.

```
centroid_initialization_strategy =  
cv2.KMEANS_RANDOM_CENTERS  
  
_, labels, centers = cv2.kmeans(pixel_values, 2, None,  
stop_criteria,  
  
number_of_attempts, centroid_initialization_strategy)
```

FIGURA 49: FUNCIÓN DEL ALGORITMO K-MEANS.

FUENTE: ELABORACIÓN PROPIA.

Por último, en la figura N°50, en centers se aplicó las etiquetas a los centroides para segmentar los píxeles en su grupo correspondiente luego se convirtió los centroides en enteros sin signos de 8 bits, para después tomar la imagen segmentada y se volvió a redimensionar para que pueda adquirirse la forma de la imagen original.

```
centers = np.uint8(centers)  
segmented_data = centers[labels.flatten()]  
segmented_image =  
segmented_data.reshape(image_copy.shape)
```

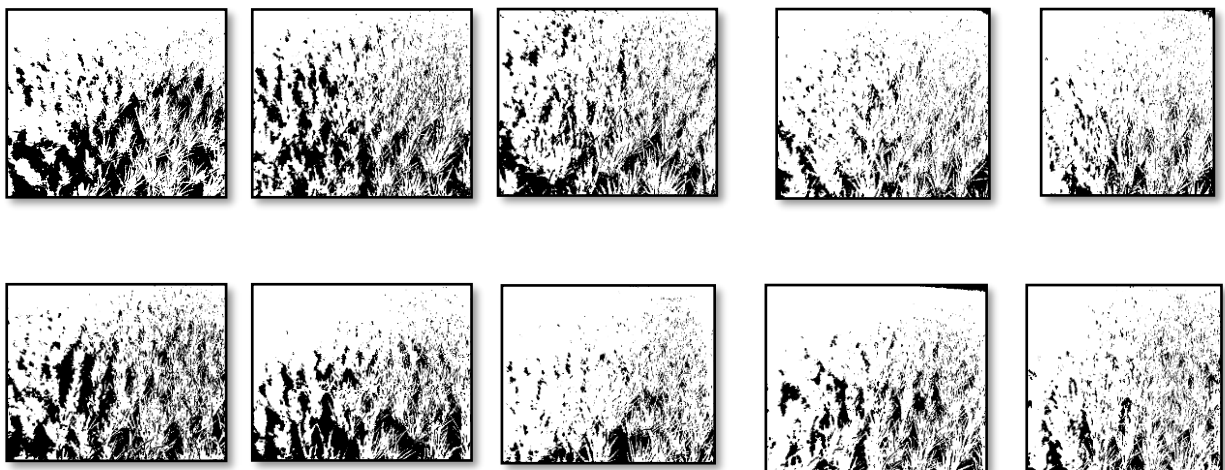
FIGURA 50: APLICACIÓN DE ETIQUETAS A LOS CENTROIDES PARA SEGMENTACIÓN DE LAS IMÁGENES DE PLANTAS DE ARROZ. FUENTE: ELABORACIÓN PROPIA.

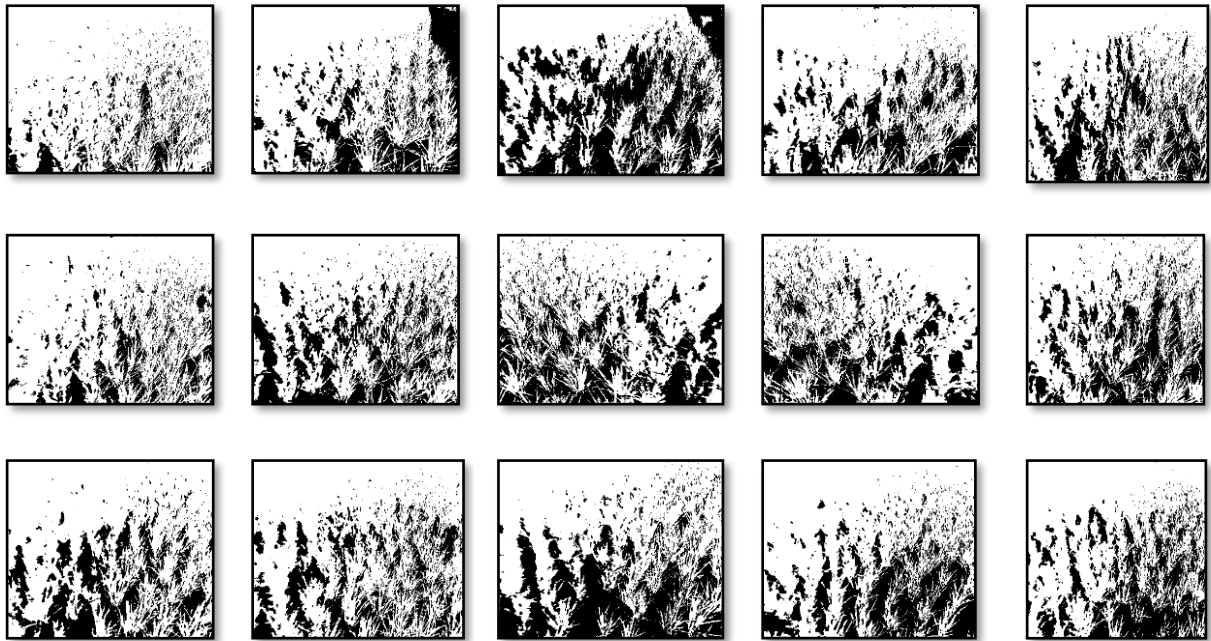
En la Tabla N° 12, se observó que las imágenes digitales de plantas de arroz hacia adelante, segmentadas con el algoritmo K-MEANS, obtuvo una mejor detección en las plantas porque se confundieron muy pocos los pixeles que no eran plantas con otros elementos que existieron en las imágenes como agua, barro. También se pudo observar que las imágenes digitales de plantas de arroz tuvieron desenfoques, el algoritmo K-MEANS logró obtener su segmentación debidamente tal cual como fuese una imagen normal, pero tendió a no visualizarse como es debido la apariencia de las plantas de arroz.

TABLA 12.

RESULTADOS DE IMÁGENES DIGITALES DE PLANTAS DE ARROZ QUE FUERON SEGMENTADAS HACIA ADELANTE CON EL ALGORITMO K-MEANS EN DÍA NUBLADO

25 imágenes De Plantas De Arroz Segmentadas Hacia Adelante En Día Nublado.





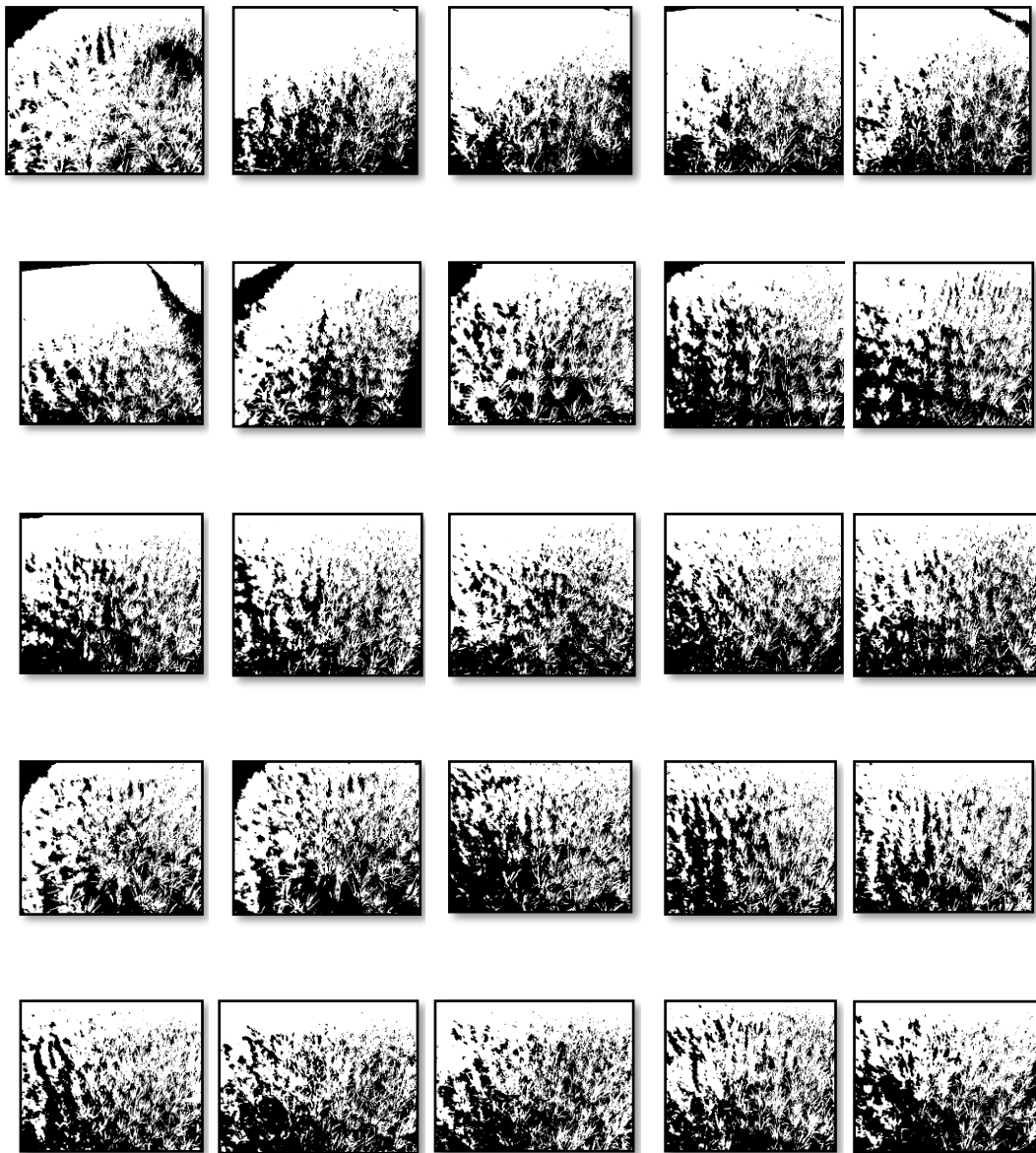
Fuente: Elaboración Propia

En la Tabla N°13, la segmentación de imágenes digitales de plantas de arroz hacia atrás con el algoritmo K-MEANS, se observó que las plantas de arroz fueron segmentadas muy bien en la mayoría de píxeles que fueron plantas incluso segmentó hasta las plantas que tuvieron desenfoques pero con la deficiencia de no apreciarse bien como es debido en la planta de arroz, solo algunos píxeles fueron confundidos con partes que no fueron plantas de arroz como agua, barro que existieron en las imágenes digitales.

TABLA 13.

RESULTADOS DE IMÁGENES DIGITALES DE PLANTAS DE ARROZ QUE FUERON SEGMENTADAS HACIA ATRÁS CON EL ALGORITMO K-MEANS EN DÍA NUBLADO.

25 imágenes De Plantas De Arroz Segmentadas Hacia Atrás En Día Nublado.



Fuente: Elaboración Propia

En la Tabla N°14, la segmentación de imágenes digitales de plantas de arroz en posición en diagonal hacia la derecha con el algoritmo K-MEANS logró segmentar bien pixeles que fueron plantas de arroz incluso aquellas que tuvieron desenfoques confundiendo solo algunos pixeles con partes que no eran plantas en las imágenes digitales.

TABLA 14.

RESULTADOS DE IMÁGENES DIGITALES DE PLANTAS DE ARROZ QUE FUERON SEGMENTADAS EN POSICIÓN DIAGONAL A LA DERECHA CON EL ALGORITMO K-MEANS EN DÍA SOLEADO.

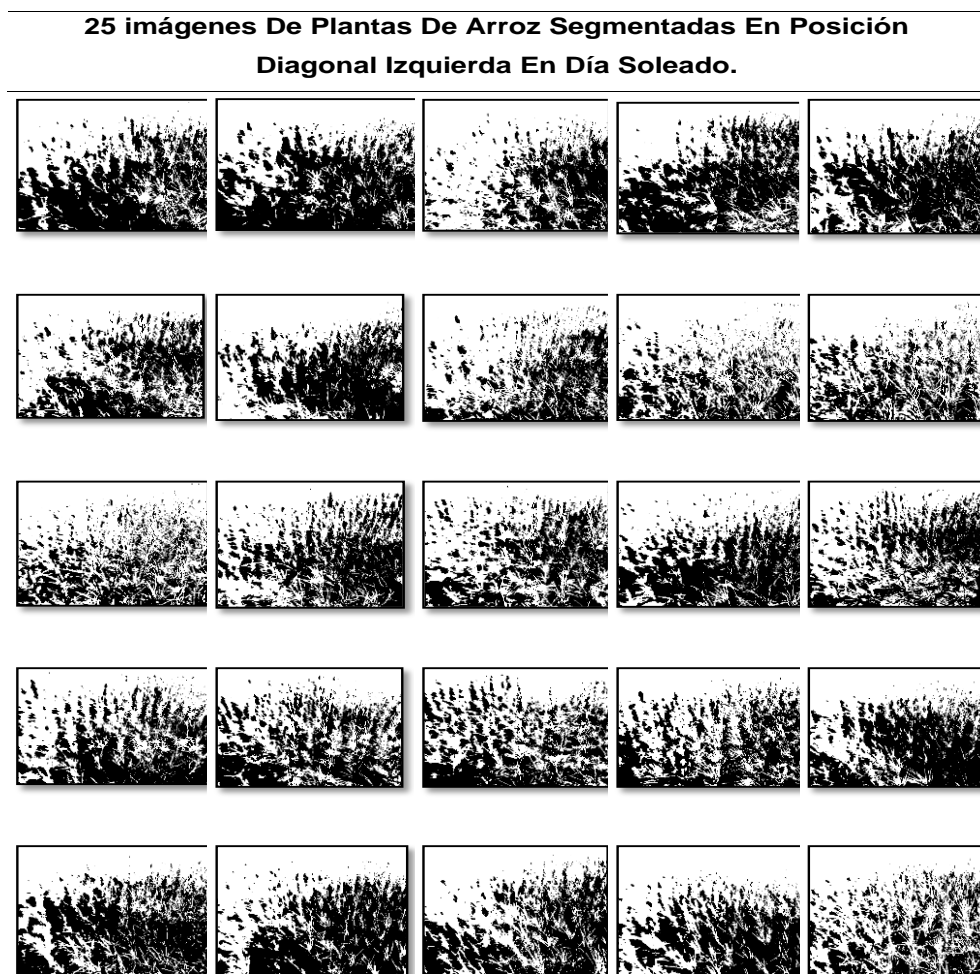


Fuente: Elaboración propia

En la Tabla N° 15, la segmentación de imágenes digitales de plantas de arroz en posición en diagonal hacia la izquierda con el algoritmo K-MEANS, no se logró segmentar bien las plantas de arroz sombreadas por arbustos que había cerca al arrozal, porque se confundió mucho las plantas sombreadas con partes que no fueron plantas de arroz en las imágenes digitales.

TABLA 15.

RESULTADOS DE IMÁGENES DIGITALES DE PLANTAS DE ARROZ QUE FUERON SEGMENTADAS EN POSICIÓN DIAGONAL A LA IZQUIERDA CON EL ALGORITMO K-MEANS EN DÍA SOLEADO.



Fuente: Elaboración propia.

Para la determinación de las métricas de calidad en el algoritmo K-MEANS, en la figura N° 51 y la figura N° 52, se observó la imagen de ingreso y la imagen de salida binarizadas que sirvieron como imágenes de referencia para la obtención de las métricas de calidad utilizando la librería Umetric.



FIGURA 51: IMAGEN DE INGRESO DE PLANTAS DE ARROZ BINARIZADA. FUENTE: ELABORACIÓN PROPIA.

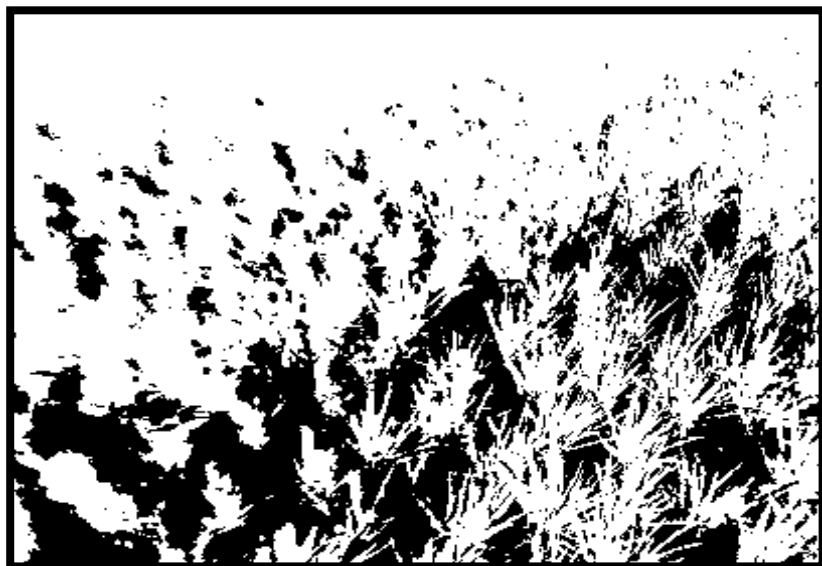


FIGURA 52: IMAGEN DE SALIDA DE PLANTAS DE ARROZ BINARIZADA. FUENTE: ELABORACIÓN PROPIA.

En el siguiente esquema de segmentación se muestra el proceso para lograr segmentar plantas de arroz en ambientes no controlados utilizando algoritmo DBSCAN.

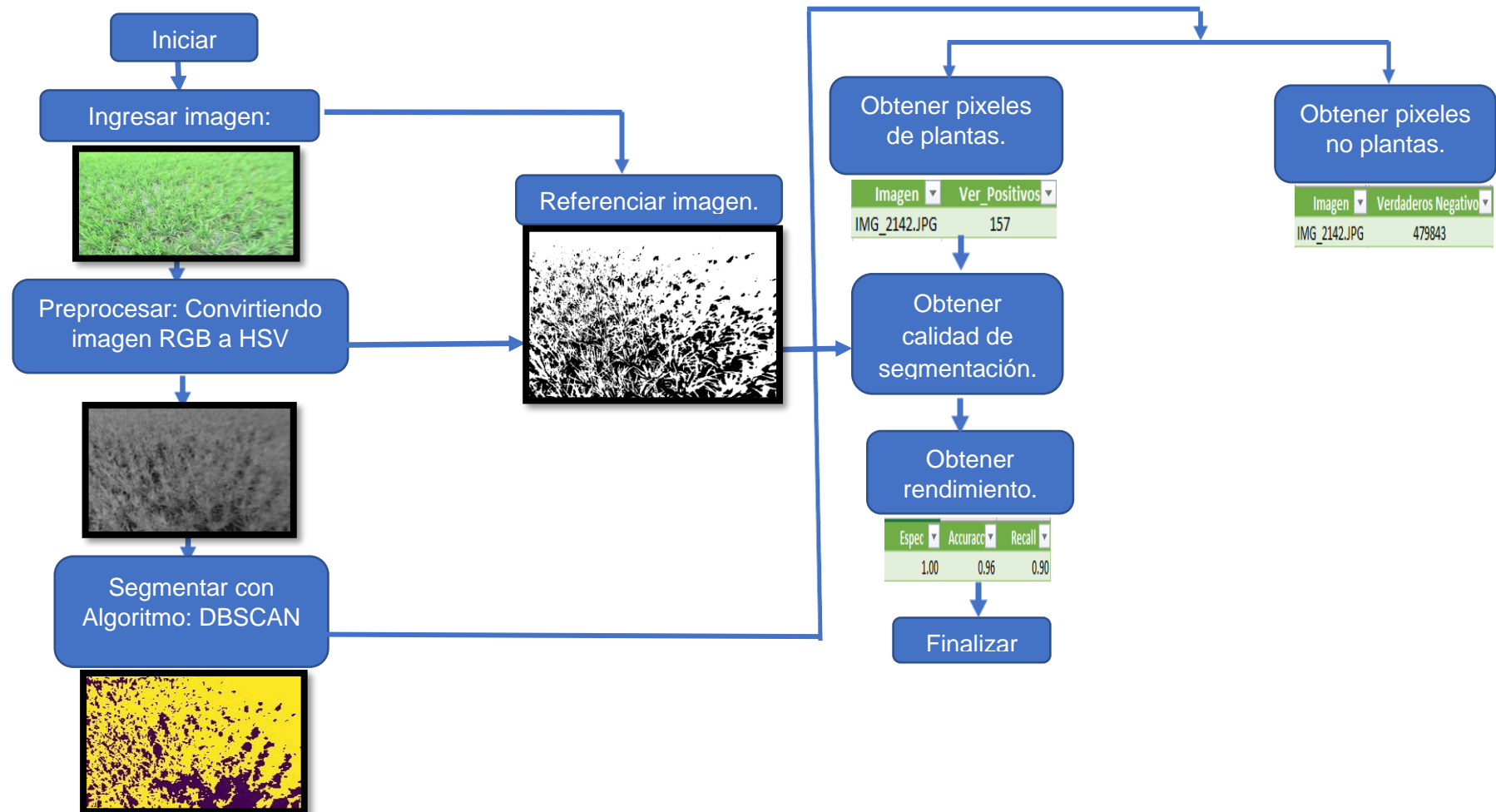


FIGURA 53: ESQUEMA DE SEGMENTACIÓN DE PLANTAS DE ARROZ EN AMBIENTE NO CONTROLADO UTILIZANDO ALGORITMO DBSCAN.

FUENTE: ELABORACIÓN PROPIA.

Las imágenes de plantas de arroz fueron adquiridas en dimensiones de 4320 X 3240 pixeles, y requirieron ser redimensionadas a 800 X 600 pixeles por la carga computacional que estas tuvieron, fueron muy grandes y consumían muchos recursos de hardware al momento que se realizó el procesamiento.

La función que se utilizó en Python 3.7 para cambiar de tamaño a las imágenes originales fue `cv2.resize` donde tomó la imagen original de ingreso al algoritmo DBSCAN que fue la dimensión S y se redimensionó a 800 x 600 pixeles.

```
dsiz=(800,600)  
self.imagen.append(cv2.resize(S, dsiz))
```

FIGURA 54: REDIMENSIÓN DE LAS IMÁGENES DE PLANTAS DE ARROZ.

FUENTE: ELABORACIÓN PROPIA.

Las imágenes de plantas de arroz RGB fueron convertidos a espacio de color HSV porque las adquisiciones de las imágenes fueron tomadas en un ambiente no controlado, por esta razón se procedió a realizar el pre procesamiento para evitar tener distintos ruidos que se tuvieron en las imágenes, siendo mayormente las distintas iluminaciones, factores naturales como el aire y sombras.

En la figura N° 55, el espacio de color R, G, B, donde tomaron los valores de R= rojo, G=verde y B=azul sus atribuciones máximas fueron sus colores primarios porque se establecieron en 255 y estos colores fueron almacenados en la dimensión V que es el valor del

espacio de color HSV, se almacenó en la dimensión V porque dicha dimensión permitió controlar la claridad del color donde cero(0) fue su valor mínimo y dio lugar a ser el color negro y 100 que fue el valor máximo que produjo colores más claros como el color blanco, todo ello dependiendo de la saturación (S) porque S representó la pureza en la distancia del eje del brillo entre el color negro y blanco.

La dimensión S, tomó al valor (V) que fue la que almacenó los valores máximos del espacio de color RGB y luego los restó con los valores mínimos del espacio de color RGB, los valores mínimos fueron ceros (0), que combinados valores máximos y valores mínimos se pudo obtener otros colores complementarios luego se dividieron entre el valor (V). Si $V \neq 0$ entonces procedió a realizar lo establecido en la Matiz de color(H) que se mostró en la figura N° 56.

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \end{cases}$$

FIGURA 55: CONVERSIÓN DE ESPACIO DE COLOR RGB A LAS DIMENSIONES V Y S DEL ESPACIO DE COLOR HSV. FUENTE:

En la figura N° 56, se pudo observar el matiz(H) que fue una de las dimensiones del espacio de color HSV, donde el matiz fue el color referido desde 0 a 360 grados, entonces: el color rojo fue expresado entre 0 y 60 grados, el color amarillo fue expresado entre 61 y 120 grados, el color verde fue expresado entre 121 y 180 grados, el color cian fue expresado entre 181 y 240 grados.

El color azul fue expresado entre 241 y 300 grados y el color magenta fue expresado entre 301 y 360 grados. Al llegar a 360 grados volvió a su estado inicial que fue el color rojo.

En el matiz(H), realizó lo establecido en cada fila del Switch según la condición que reflejo el *if* con referencia a la dimensión del valor(V) que comparó igualando a las dimensiones R, G, B del espacio de color RGB.

$$H \leftarrow \begin{cases} \frac{60(G-B)}{V-\min(R,G,B)} & \text{if } V = R \\ 120 + \frac{60(B-R)}{V-\min(R,G,B)} & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \\ 0 & \text{if } R = G = B \end{cases}$$

FIGURA 56: DIMENSIÓN H DEL ESPACIO DE COLOR HSV. FUENTE: [36]

Si $H < 0$ entonces $H \leftarrow H + 360$. En la salida $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$.

Después los valores fueron convertidos al tipo de dato destino donde:

En las circunstancias de que las imágenes sean de 8 y 16 bits, R, G y B se tuvo que convertir al formato flotante y se escaló para lograr ajustarse al rango de 0 a 1.

Las imágenes de 8 bits: $V \leftarrow 255V$, $S \leftarrow 255S$, $H \leftarrow H/2$ (para poderlo ajusta desde 0 a 255)

Las imágenes de 16 bits:(En la actualidad no son admitidas)
V< -65535V, S< -65535 S, H< -H

Las imágenes de 32 bits: H, S y V se dejaron como estaban.

En la figura N° 57, se utilizó la función `cvtColor` para realizar la conversión de las imágenes de plantas de arroz que estuvieron establecidos en espacio de color RGB y fueron transformadas a Espacio de color HSV.

Donde cada imagen estuvo almacenada en la variable `img` que a través del método `cv2.COLOR_BGR2HSV` fue convertida a espacio de color HSV, ya convertida a espacio de color HSV se almacenó en la variable `labimg`. Ya almacenada la imagen en la variable `labimg` se procedió a la descomposición de la imagen HSV, utilizando el método `cv2.split`, el método mencionado realizó la descomposición de la imagen en sus tres dimensiones H, S, V.

```
labimg=cv2.cvtColor(img, cv2.COLOR_BGR2HSV)  
H , S, V= cv2.split(labimg)
```

FIGURA 57: CONVIERTE UNA IMAGEN DE ESPACIO DE COLOR RGB A HSV. FUENTE: ELABORACIÓN PROPIA.

La imagen original de ingreso se mostró en la figura N°58 y luego en las figuras N°59, 60 y 61 se mostraron sus dimensiones H, S, V.

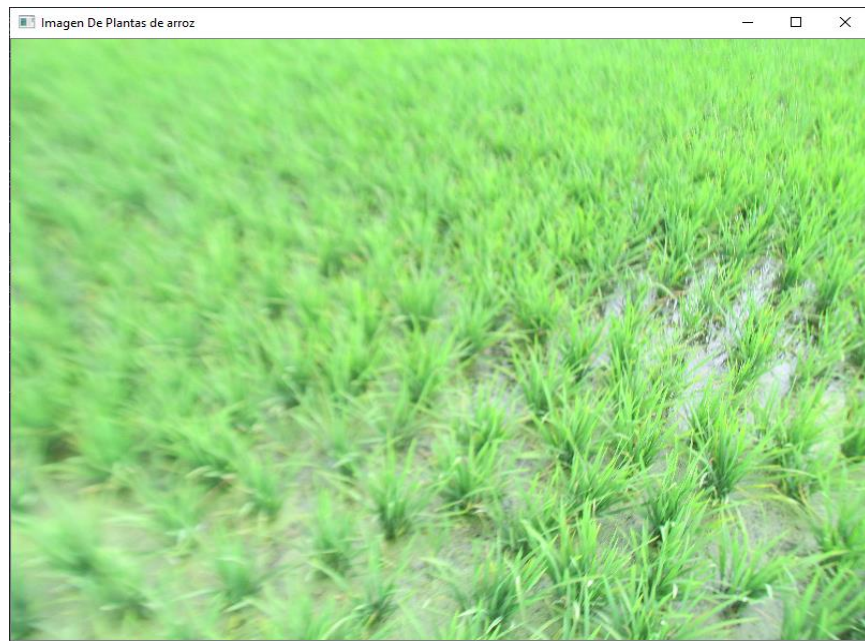


FIGURA 58: IMAGEN DE PLANTAS DE ARROZ RGB. FUENTE: ELABORACIÓN PROPIA.

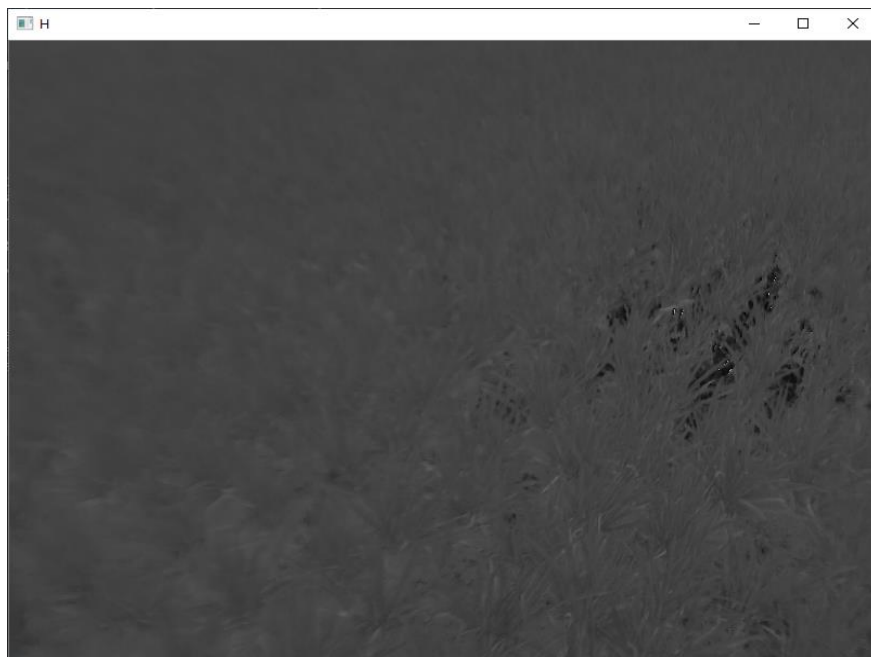


FIGURA 59: IMAGEN DE PLANTAS DE ARROZ EN SU DIMENSIÓN H. FUENTE: ELABORACIÓN PROPIA.

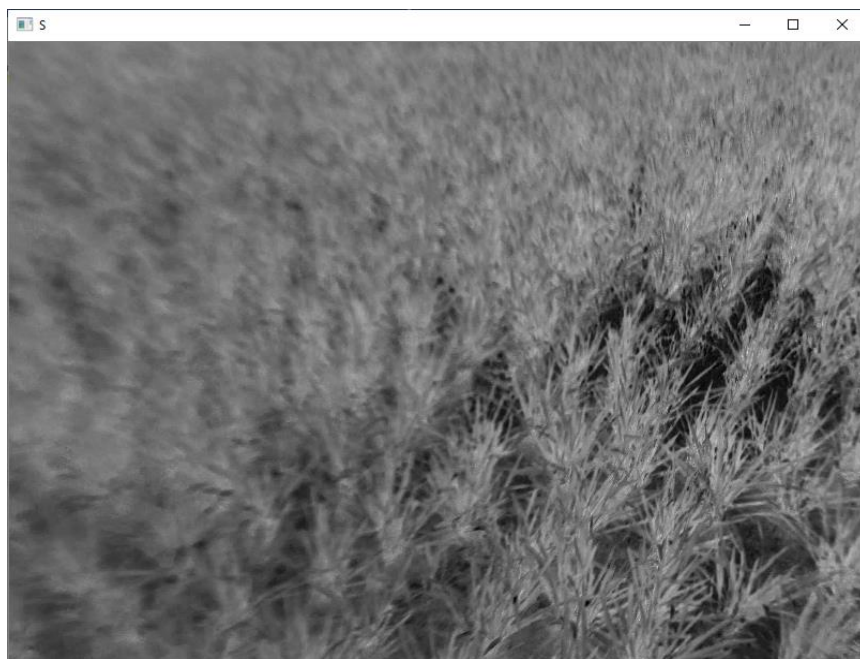


FIGURA 60: IMAGEN DE PLANTAS DE ARROZ EN SU DIMENSIÓN S.
FUENTE: ELABORACIÓN PROPIA.

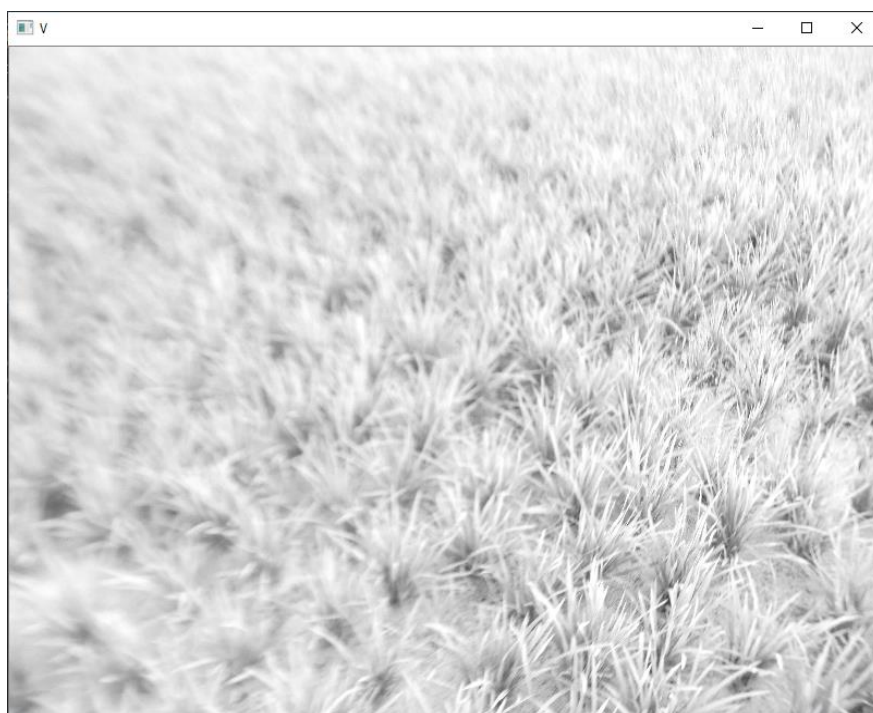


FIGURA 61: IMAGEN DE PLANTAS DE ARROZ EN SU DIMENSIÓN V.
FUENTE: ELABORACIÓN PROPIA.

Luego de haberse dividido la imagen en 3 dimensiones H, S, V, cada una de ellas, se observó que las imágenes que mejor performance dispusieron, fueron las de dimensión S porque tuvieron una mejor distinción entre plantas de arroz y los otros factores que existieron en la imagen como agua, barro. Las 100 imágenes digitales plantas de arroz que fueron tomadas en el ambiente no controlado en sus distintas posiciones, como imágenes hacia adelante, imágenes hacia atrás, imágenes en posición en diagonal hacia la izquierda e imágenes en posición en diagonal hacia la derecha, se tomó la dimensión S (Figura N°60) de cada una de ellas y fue el que ingresó al algoritmo DBSCAN para poder realizar la segmentación.

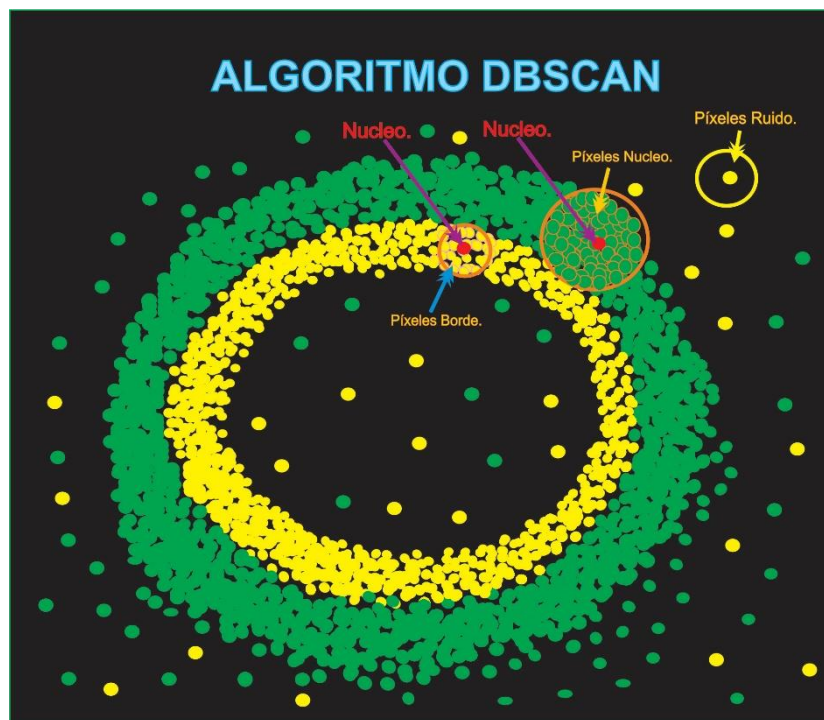


FIGURA 62: ALGORITMO DBSCAN. EPS = 10 FUENTE: ELABORACIÓN PROPIA.

En la figura N° 62 se observó que el algoritmo DBSCAN en su radio eps debe tener un número mínimo de píxeles que fueron las muestras para poder tener las consideraciones en generar los núcleos.

Para poder genera los núcleos en las imágenes de plantas de arroz se debe tener las siguientes consideraciones:

Dato Núcleo: Fue considerado núcleo si en el entorno del radio eps (épsilon) existieron el número mínimo de píxeles en min_samples.

Dato Borde: Fue considerado borde si en el radio eps (épsilon) no tuvo el mínimo número de píxeles en min_samples , pero existió el otro dato que es el núcleo.

Dato Ruido: Fue considerado ruido si no existió un núcleo ni un dato borde.

En la figura N° 63, el algoritmo de agrupamiento espacial basado en densidad de aplicaciones con ruido DBSCAN tubo parámetros muy importantes de los que dependió para poder realizar la segmentación en las imágenes digitales de plantas de arroz.

Uno de los parámetros fue el radio del entorno eps donde se denominó épsilon, en El se estableció 10 porque existieron diversidad de píxeles plantas y píxeles no plantas por ende el radio tuvo que ser muy amplio porque el conjunto de datos fue muy robusto, para medir que distancia tuvo un pixel de otro se utilizó la métrica de la distancia euclidiana así se determinó que tan cerca pudo estar un

pixel de otro y poder determinar si podía pertenecer a un grupo en el radio.

Por otro lado en el parámetro `min_samples` se estableció 1900, donde `eps` se convierte en núcleo si el número de vecinos en `min_samples` es mayor o igual a 1900 si es así el grupo es válido y se convierte en una región densa, y por último se estableció que el algoritmo realice segmentación de manera automática.

```
db= DBSCAN(eps=10, min_samples=1900, metric= 'euclidean',  
algorithm='auto')
```

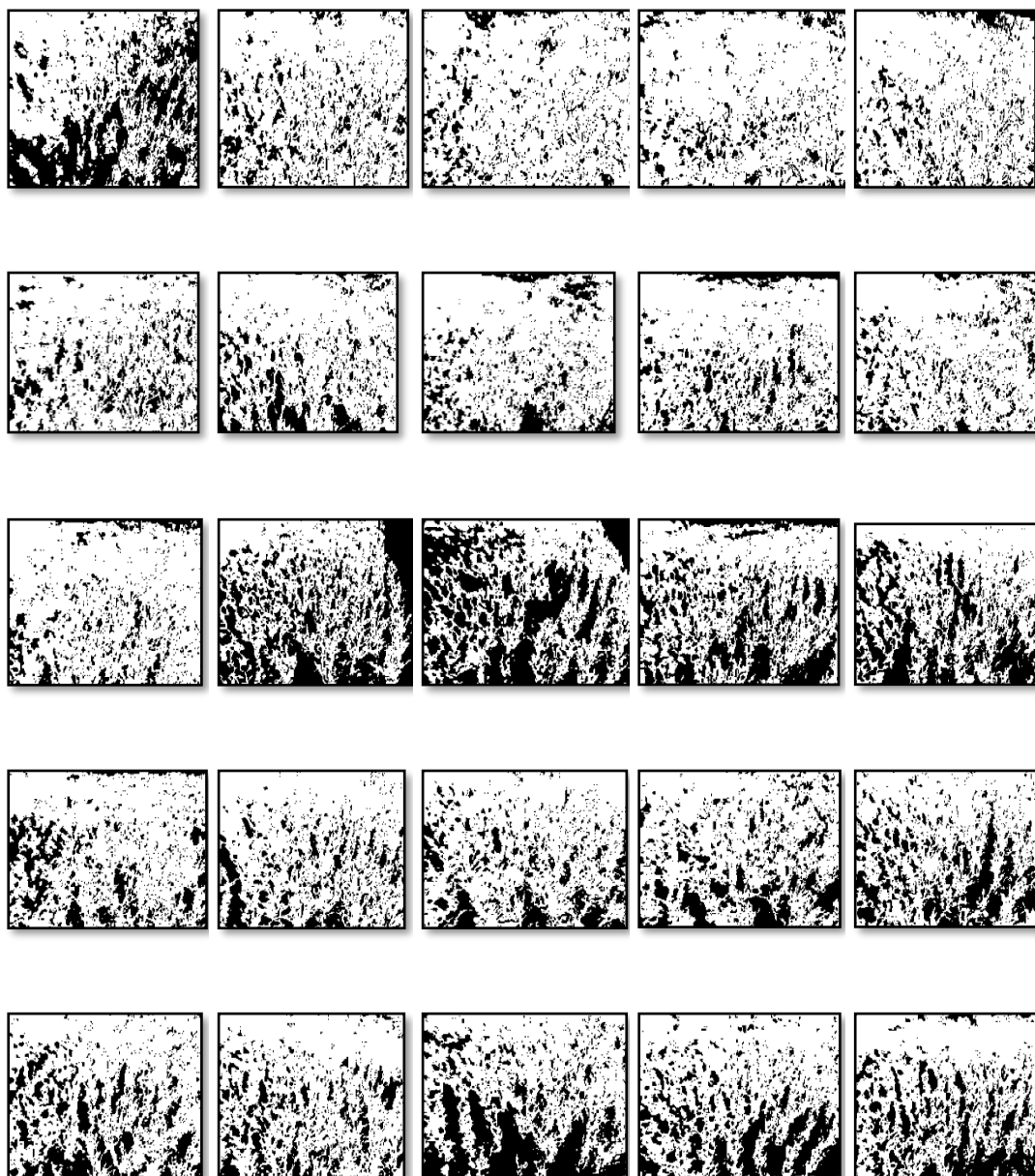
FIGURA 63: PARÁMETROS UTILIZADOS PARA SEGMENTAR CON EL ALGORITMO DBSCAN. FUENTE: ELABORACIÓN PROPIA.

En la tabla N° 16 se mostraron imágenes de plantas de arroz segmentadas hacia adelante en día nublado con el algoritmo DBSCAN y se observó que no se logró distinguir con exactitud las áreas de interés que fueron las plantas de arroz porque las imágenes se mostraron demasiado ruidosas.

TABLA 16.

RESULTADO DE IMÁGENES DIGITALES DE PLANTAS DE ARROZ QUE FUERON SEGMENTADAS HACIA ADELANTE CON EL ALGORITMO DBSCAN EN DÍA NUBLADO.

25 imágenes De Plantas De Arroz Segmentadas Hacia Adelante En Día Nublado.



Fuente: Elaboración Propia

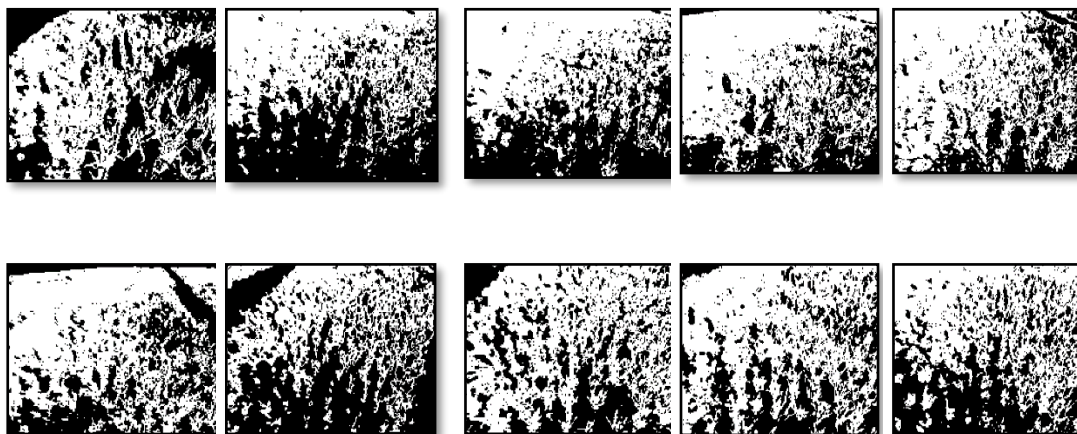
En la Tabla N° 17, se mostró imágenes de plantas de arroz segmentadas hacia atrás en día nublado con el algoritmo DBSCAN y se observó que no se logró segmentar bien las plantas de arroz, porque las imágenes no mostraron píxeles que fueron plantas de arroz y se confundieron mucho con los píxeles que fueron parte del suelo como agua, barro.

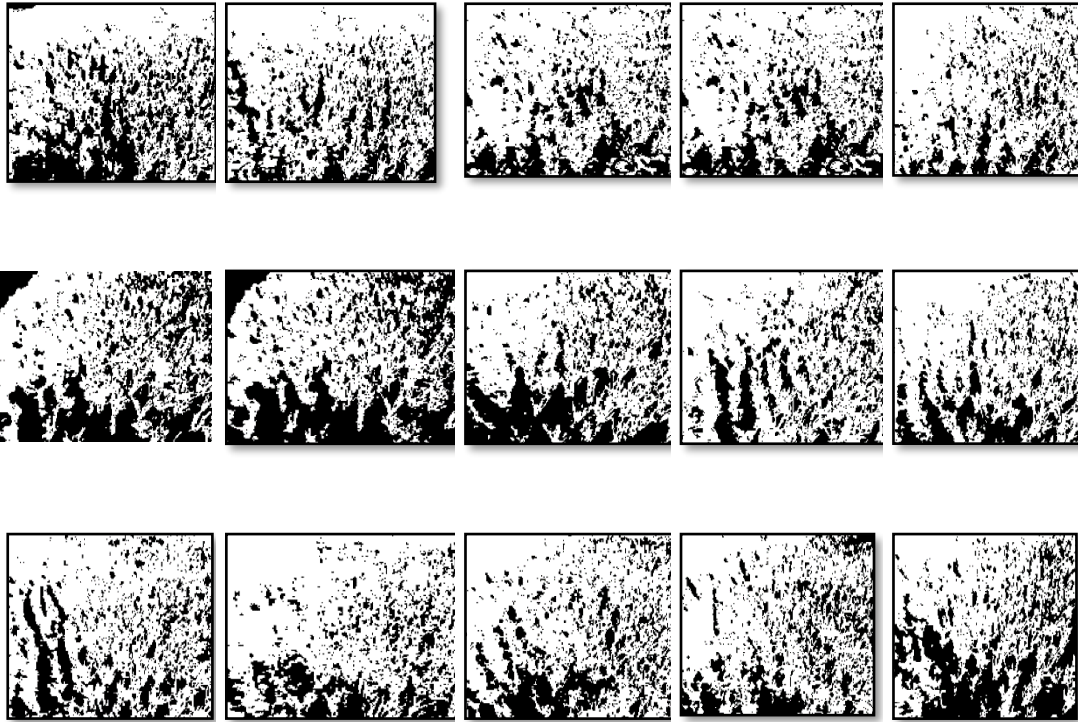
También las partes que tuvieron plantas demasiado aglomeradas y desenfocadas no logro procesar bien la segmentación.

TABLA 17.

RESULTADO DE IMÁGENES DIGITALES DE PLANTAS DE ARROZ QUE FUERON SEGMENTADAS HACIA ATRÁS CON EL ALGORITMO DBSCAN EN DÍA NUBLADO.

25 imágenes De Plantas De Arroz Segmentadas Hacia Atrás En Día Nublado.





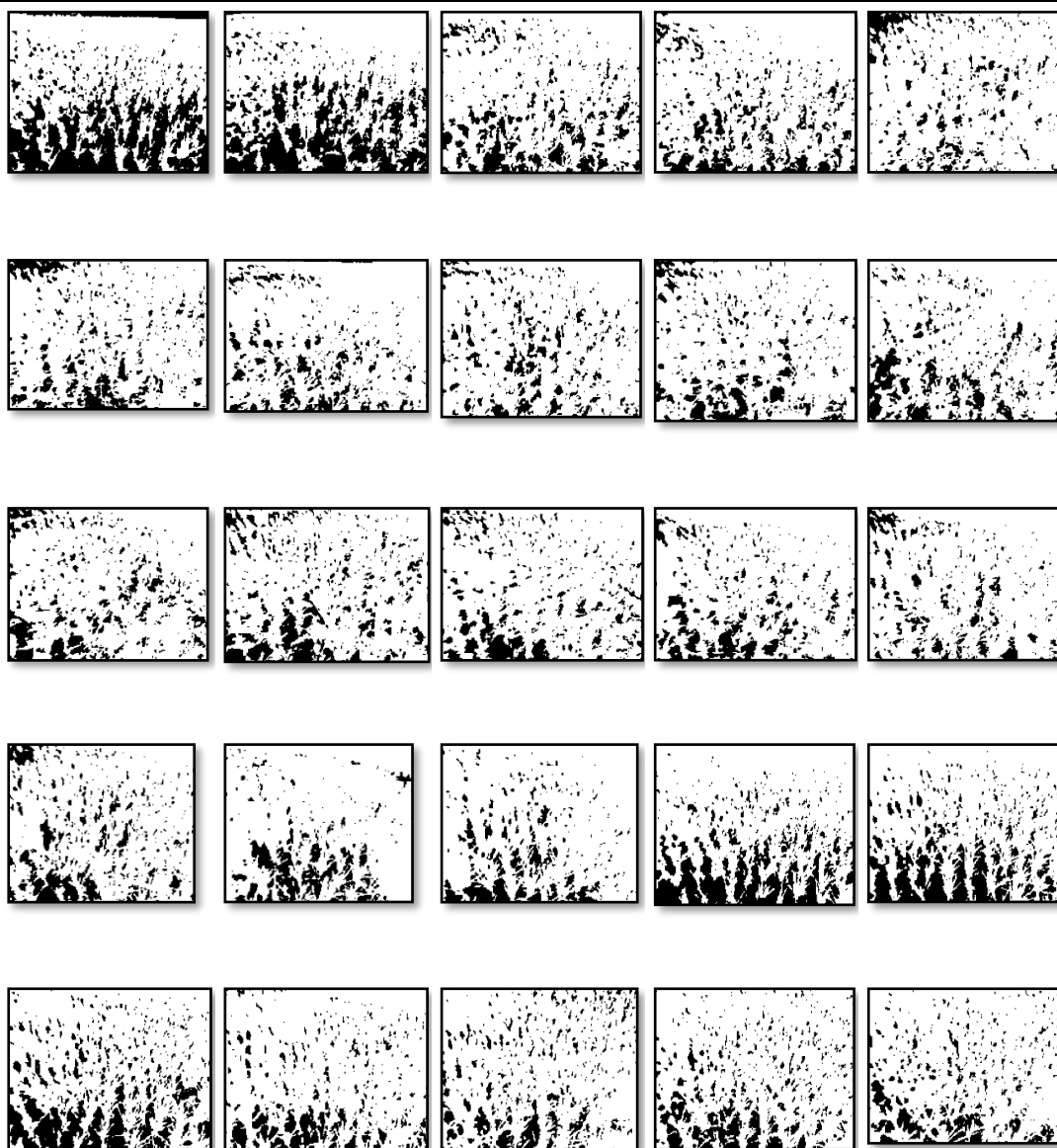
Fuente: Elaboración Propia.

En la Tabla N° 18, se mostró imágenes de plantas de arroz segmentadas en posición diagonal hacia la derecha en día soleado con el algoritmo DBSCAN y se observó que no se logró segmentar bien las plantas de arroz, porque en las imágenes confundió mucho los pixeles de suelo con los pixeles que son plantas.

TABLA 18.

RESULTADO DE IMÁGENES DIGITALES DE PLANTAS DE ARROZ QUE FUERON SEGMENTADAS EN POSICIÓN DIAGONAL HACIA LA DERECHA CON EL ALGORITMO DBSCAN EN DÍA SOLEADO.

25 imágenes De Plantas De Arroz Segmentadas En Posición Diagonal Derecha En Dia Soleado.



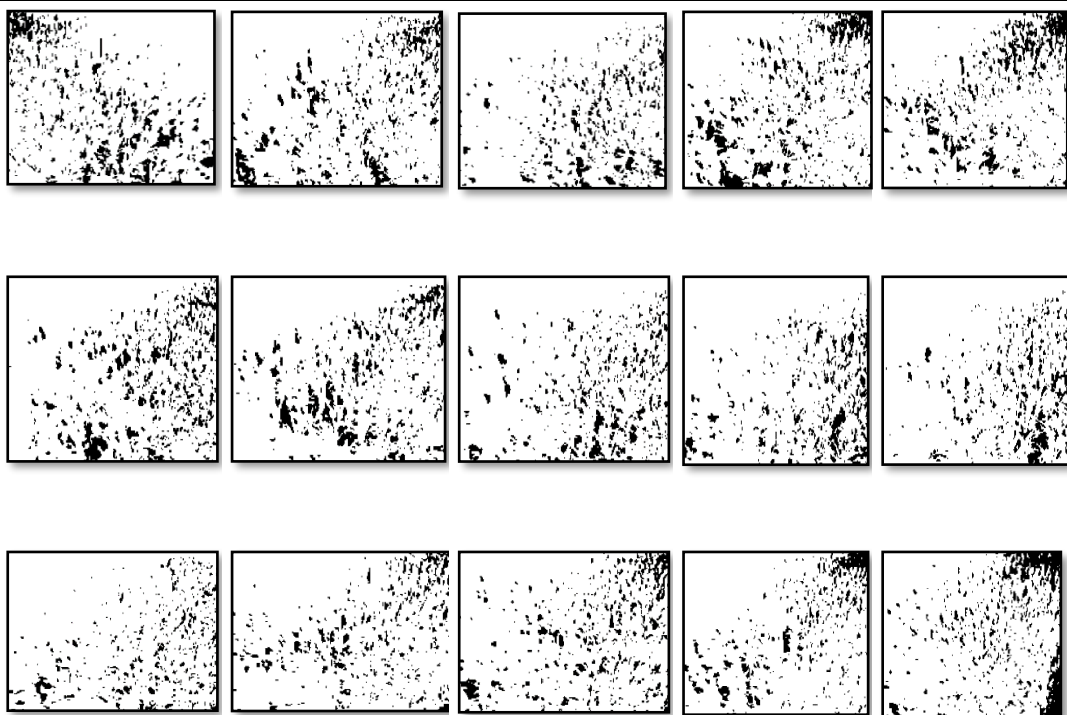
Fuente: Elaboración Propia

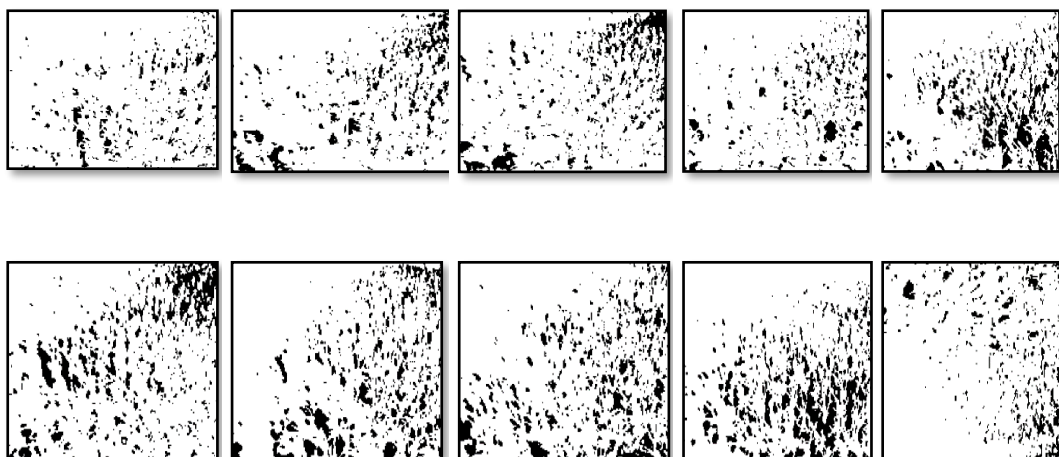
En la Tabla N° 19, se mostró imágenes de plantas de arroz segmentadas en posición en diagonal a la izquierda en día soleado con el algoritmo DBSCAN y se observó que no se logró segmentar bien las plantas de arroz, porque había demasiada sombra en el arrozal.

TABLA 19.

RESULTADO DE IMÁGENES DIGITALES DE PLANTAS DE ARROZ QUE FUERON SEGMENTADAS EN POSICIÓN DIAGONAL A LA IZQUIERDA CON EL ALGORITMO DBSCAN EN DÍA SOLEADO.

25 imágenes De Plantas De Arroz Segmentadas En Posición Diagonal Izquierda En Día Soleado.





Fuente: Elaboración Propia

Para la determinación de las métricas de calidad en el algoritmo DBSCAN, se utilizó la imagen de ingreso y la imagen de salida binarizadas pudiéndose apreciar en la Figura N°64 y Figura N°65 que sirvieron como imágenes de referencia para la obtención de las métricas de calidad, para ello se utilizó la librería Umetric.

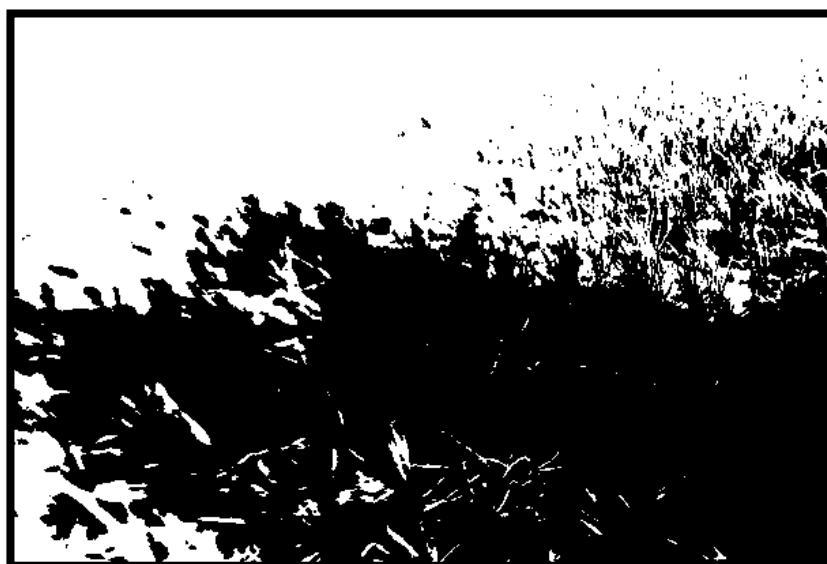


FIGURA 64: IMAGEN DE INGRESO DE PLANTAS DE ARROZ BINARIZADA. FUENTE: ELABORACIÓN PROPIA.

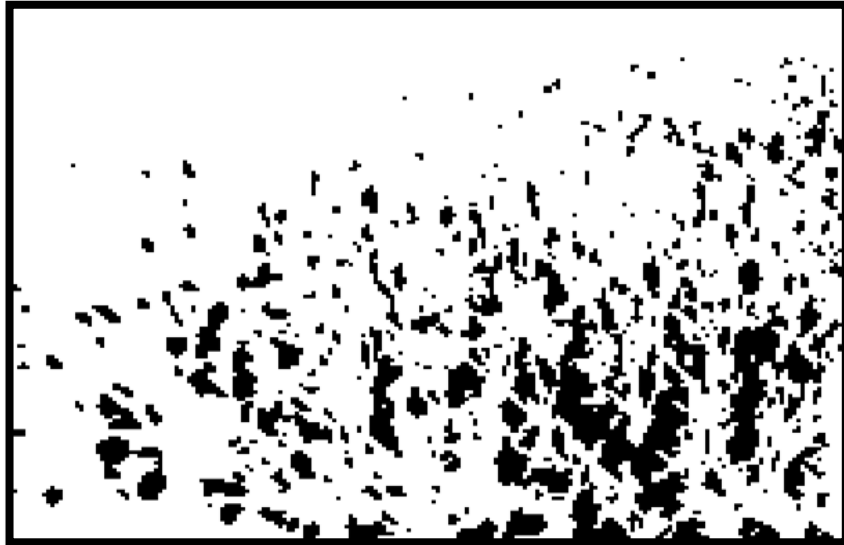


FIGURA 65: *IMAGEN DE SALIDA DE PLANTAS DE ARROZ BINARIZADA.*
FUENTE: ELABORACION PROPIA.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Las 100 imágenes digitales de plantas de arroz, tuvieron dimensiones originales de 4320 x 3240 pixeles, imágenes en diferentes perspectivas de sus ubicaciones (hacia adelante, hacia atrás, en posición en diagonal a la izquierda, posición en diagonal a la derecha) pero para tener un mejor pre procesamiento y procesamiento de segmentación concluí reducir las imágenes digitales a 800 x 600 pixeles debido a que realizaba mucho consumo de hardware y por ende existía demasiado retardo en segmentar imagen por imagen.

Se llegó a la conclusión de considerar en la base de datos 100 imágenes digitales de plantas de arroz para la segmentación con los algoritmos K-MEANS y DBSCAN, porque es la cantidad mínima para obtener resultados estadísticos significativos.

Se concluyó que utilizando el espacio de color CIELab y el espacio de color HSV en el pre procesamiento es adecuado debido a que cumplen con el estándar de iluminación D65 para la captura de imágenes digitales al aire libre o al exterior o muy conocido como un ambiente no controlado donde existen diferentes tipos de iluminaciones y sombras.

Logré concluir que las pruebas de rendimiento total para el algoritmo K-MEANS, obtuvo un tiempo de respuesta de 155.14 Ms, consumo de memoria RAM de 67.66%, consumo de CPU de 9.78%, mientras que con el algoritmo DBSCAN tuvo un tiempo de respuesta de 233.40 Ms, consumo de memoria RAM de 109.24%, consumo de CPU fue de 13.32%, obtuvo

resultados diferentes debido a que el pre procesamiento fue con espacios de color distintos, donde el espacio de color HSV tardó más de lo debido al momento de realizar el pre procesamiento con la dimensión S por lo que fue un espacio de color que tuvo más diversidad de combinación de colores en sus coordenadas influenciando así en la cromaticidad distinta que tuvieron los espacios de color, también se debe considerar el requerimiento de hardware mínimo que fue utilizado para el desarrollo de los algoritmos porque de ello dependió el desempeño en el procesamiento .

Las pruebas de segmentación realizadas obtuvieron resultados parciales tanto en imágenes de día nublado como en imágenes de día soleado, sin embargo concluimos que el total de las 100 imágenes digitales de plantas de arroz segmentadas, donde el algoritmo K-MEANS obtuvo una Exactitud de 92.25%, una Sensibilidad de 78.50%, y la Especificidad fue del 100%, mientras tanto en el algoritmo DBSCAN tuvo una exactitud de 54.25%, una Sensibilidad de 0% y una Especificidad de 100%, el algoritmo K-MEANS tuvo mejores resultados en rendimiento y en procesamiento porque a pesar que las imágenes tenían desenfoques y sombras en el arrozal producto de arbustos que había muy cerca , logro obtener una mejor Exactitud en pixeles que fueron plantas, confundiendo muy poco con los pixeles que no eran plantas como el barro, agua que existieron en las imágenes digitales.

4.2. Recomendaciones

El algoritmo DBSCAN en los resultados de las imágenes digitales de plantas de arroz segmentadas obtuvo una Exactitud de 54.25%, donde demostró un resultado subóptimo en la detección de plantas de arroz porque las imágenes adquiridas tuvieron algunos desenfoques en ellas, Así también no logró tener una buena sensibilidad porque no pudo detectar con precisión los pixeles que fueron plantas de arroz, para realizarlo se debe adquirir imágenes con cámara digital profesional.

Para un trabajo futuro se recomendó realizar la adquisición de imágenes con drones debido a que la calidad es mucho más eficaz porque se debe tener en cuenta la posición y altura del sol y así evitar sobre exposiciones y/o sombras.

REFERENCIAS

- [1] Ministerio de Energia y Minas, "Junio 2023," *Principales Indicadores Del Sect. Eléctrico a Niv. Nac.*, vol. 2023, no. Cifras preliminares del mes de mayo 2023, p. 2023, 2023, [Online]. Available:
<https://www.gob.pe/institucion/minem/informes-publicaciones/tipos/2-boletin>
- [2] R. Meneses, "Manejo Integrado de los Principales Insectos y Ácaros Plagas del Arroz CUBA," *liarroz*, p. 130, 2008, [Online]. Available:
http://cagricola.uclv.edu.cu/descargas/libros/LIBRO_Manejo_Integrado_de_los_principales_insectos_y_acaros_plagas_del_arroz.pdf
- [3] J. Sounders, D. Coto, and A. King, "Plagas invertebradas de cultivos anuales." p. 234, 1998.
- [4] V. Degiovanni Beltramo, J. Gómez, and J. Sierra, "Análisis de crecimiento y etapas de desarrollo de tres variedades de arroz (*Oryza sativa* L.) en Montería, Córdoba," *Temas Agrar.*, vol. 9, no. 1, pp. 21–29, 2004, doi: 10.21897/rta.v9i1.620.
- [5] X. Liu, C. Hu, and P. Li, "Automatic segmentation of overlapped poplar seedling leaves combining Mask R-CNN and DBSCAN," *Comput. Electron. Agric.*, vol. 178, no. May, p. 105753, 2020, doi: 10.1016/j.compag.2020.105753.
- [6] Y. Yu, K. Zhang, L. Yang, and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN," *Comput. Electron. Agric.*, vol. 163, no. April, p. 104846, 2019, doi: 10.1016/j.compag.2019.06.001.
- [7] K. Tian, J. Li, J. Zeng, A. Evans, and L. Zhang, "Segmentation of tomato leaf images based on adaptive clustering number of K-means algorithm," *Comput. Electron. Agric.*, vol. 165, no. August, p. 104962, 2019, doi:

- 10.1016/j.compag.2019.104962.
- [8] T. Liu *et al.*, “Estimates of rice lodging using indices derived from UAV visible and thermal infrared images,” *Agric. For. Meteorol.*, vol. 252, no. December 2017, pp. 144–154, 2018, doi: 10.1016/j.agrformet.2018.01.021.
- [9] X. Bai *et al.*, “Rice heading stage automatic observation by multi-classifier cascade based rice spike detection method,” *Agric. For. Meteorol.*, vol. 259, no. 1037, pp. 260–270, 2018, doi: 10.1016/j.agrformet.2018.05.001.
- [10] J. Liao, Y. Wang, J. Yin, L. Liu, S. Zhang, and D. Zhu, “Segmentation of Rice Seedlings Using the YCrCb Color Space and an Improved Otsu Method,” pp. 1–16, 2018, doi: <https://doi.org/10.3390/agronomy8110269>.
- [11] S. Zhang, H. Wang, W. Huang, and Z. You, “Optik Plant diseased leaf segmentation and recognition by fusion of,” *Opt. - Int. J. Light Electron Opt.*, vol. 157, pp. 866–872, 2018, doi: 10.1016/j.ijleo.2017.11.190.
- [12] Z. Liu, X. Bai, C. Sun, F. Zhou, and Y. Li, “Multi-Modal Ship Target Image Smoothing Based on Adaptive Mean Shift,” *IEEE Access*, vol. 6, no. c, pp. 12573–12586, 2018, doi: 10.1109/ACCESS.2018.2794141.
- [13] G. Tong, Y. Li, H. Chen, Q. Zhang, and H. Jiang, “Optik Improved U-NET network for pulmonary nodules segmentation,” *Opt. - Int. J. Light Electron Opt.*, vol. 174, no. July, pp. 460–469, 2018, doi: 10.1016/j.ijleo.2018.08.086.
- [14] W. Guo, B. Zheng, T. Duan, T. Fukatsu, S. Chapman, and S. Ninomiya, “EasyPCC: Benchmark datasets and tools for high-throughput measurement of the plant canopy coverage ratio under field conditions,” *Sensors (Switzerland)*, vol. 17, no. 4, Apr. 2017, doi: 10.3390/s17040798.
- [15] S. Yin, Y. Qian, and M. Gong, “Unsupervised hierarchical image segmentation

- through fuzzy entropy maximization,” *Pattern Recognit.*, vol. 68, pp. 245–259, 2017, doi: 10.1016/j.patcog.2017.03.012.
- [16] S. W. Chen *et al.*, “Counting Apples and Oranges With Deep Learning: A Data-Driven Approach,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 781–788, 2017, doi: 10.1109/LRA.2017.2651944.
- [17] J. C. J. Montalvo Samamé, “Procesamiento de imágenes digitales utilizando descriptores de color para la identificación y clasificación de diversos tipos de semilla de Quinoa,” *Repos. Inst. - USS*, 2017, Accessed: Sep. 16, 2018. [Online]. Available: <http://renati.sunedu.gob.pe/handle/sunedu/185059>
- [18] Y. Li, Z. Cao, H. Lu, Y. Xiao, Y. Zhu, and A. B. Cremers, “In-field cotton detection via region-based semantic image segmentation,” *Comput. Electron. Agric.*, vol. 127, pp. 475–486, 2016, doi: 10.1016/j.compag.2016.07.006.
- [19] M. Riomoros, “Tesis doctoral,” 2016.
- [20] E. Cuevas, D. Zaldivar, and M. Pérez-Cisneros, “A novel multi-threshold segmentation approach based on differential evolution optimization,” *Expert Syst. Appl.*, vol. 37, no. 7, pp. 5265–5271, 2010, doi: 10.1016/j.eswa.2010.01.013.
- [21] Vélez Serrano José F., Moreno Díaz Ana Belén, Sánchez Calle Ángel, and Sánchez-Marín José L., *Vision por computador*. Madrid, 2003. Accessed: Sep. 08, 2018. [Online]. Available: <https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=3158323&query=vision+por+computador>
- [22] NICOLÁS AGUIRRE DOBERNACK, “Procesamiento de imágenes,” *Imágenes Digit.*, vol. 2, no. 786, pp. 42–72, 2003, [Online]. Available:

<https://docplayer.es/37174768-3-procesamiento-de-imagenes.html>

- [23] Lan Maria Eggers and Damián Dilon Armando, *Artes visuales producción y análisis de la imagen*. Buenos Aires, 2014. Accessed: Sep. 20, 2018. [Online]. Available:
<https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=3221550&query=color+y+textura+de+una+imagen#>
- [24] P. J. Quirós Soto, “Convolución Matricial Aplicado al Procesamiento de Imágenes,” pp. 1–3, 2021, [Online]. Available:
https://www.tec.ac.cr/sites/default/files/media/uploads/presentacion_pablosoto.pdf
- [25] “Tema 3 : Filtros,” pp. 1–37, [Online]. Available:
<https://alojamientos.us.es/gtocom/pid/tema3-1.pdf>
- [26] M. Millan, “Estudio de los criterios electrocardiográficos de voltaje para la detección de hipertrofia ventricular izquierda en el deportista español de alta competición.,” Universidad complutense Facultad de medicina, 2018. Accessed: Sep. 27, 2018. [Online]. Available:
<https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=3160605&query=curva+roc#>
- [27] Jordi Gironés Roig, Jordi Casas Roma Jordi, Juliá Minguillón Alfonso, and Ramon Caihuelas Quiles, *Minería de datos Modelos y algoritmos*. Ramba del poblenou, 2017. Accessed: Sep. 19, 2018. [Online]. Available:
<https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=5045398&query=algoritmo+de+agrupamiento+jerarquico>
- [28] J. Gonzáles, “Clustering de Zonas en Imágenes Una aplicación a la detección de peatones.,” 2019.


- [29] C. Reynoso, *Complejidad y caos una exploración antropológica*, SB. 2016.
Accessed: Sep. 20, 2018. [Online]. Available:
[https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=4676023
&query=complejidad+de+un+algoritmo#](https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=4676023&query=complejidad+de+un+algoritmo#)
- [30] E. Moreno and H. Ramírez, *Grafos : fundamentos y algoritmos*. Accessed: Sep. 19, 2018. [Online]. Available:
[https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=3196572
&query=definicion+de+algoritmo#](https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=3196572&query=definicion+de+algoritmo#)
- [31] J. López, "Optimización Multi-objetivo Aplicaciones a problemas del mundo real," 2013. doi: 10.1007/s13398-014-0173-7.2.
- [32] Gerar Escudero Raul Benitez and David Masip Rodo Samir Kanaan, *Inteligencia Artificial Avanzada*. barcelona, 2013. Accessed: Sep. 17, 2018. [Online]. Available:
[https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=3217957
&query=Máquina+de+vectores+de+soporte+SVM%3A#](https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=3217957&query=Máquina+de+vectores+de+soporte+SVM%3A#)
- [33] H. Górriz Sáez, A. J. Álvarez Martínez, and R. M. Oliva Molina, *Procesamiento digital de imagen para la caracterización morfométrica de la especie Frankliniella occidentalis*. Editorial Universidad de Almería, 2018. Accessed: Oct. 01, 2018. [Online]. Available:
https://books.google.com.pe/books?id=GghZDwAAQBAJ&pg=PA4&lpg=PA4&q=rmse+segmentacion+de+imagenes&source=bl&ots=kVrOTer5ST&sig=OQGaKFmtnCs05_YHyaZCMwvhc0&hl=es-419&sa=X&ved=2ahUKEwiF7Zrn5ObdAhUQrFMKHc1TBhIQ6AEwEnoECAIQAQ#v=onepage&q=rmse&f=false
- [34] R. Marcos and L. Baena, "Técnicas de estadística computacional para visión

- por computador,” 2018.
- [35] A. DEGUCHI, “Reconsideration of Comparison as Anthropological Method : An Introduction,” *Japanese J. Ethnol.*, vol. 68, no. 2, pp. 214–225, 2012, doi: 10.14890/minkennewseries.68.2_214.
- [36] Y. Li, Z. Huang, Z. Cao, H. Lu, H. Wang, and S. Zhang, “Performance evaluation of crop segmentation algorithms,” *IEEE Access*, vol. 8, pp. 36210–36225, 2020, doi: 10.1109/ACCESS.2020.2969451.
- [37] K. Cao, G. Meng, Z. Wang, Y. Liu, H. Liu, and L. Sun, “An adaptive pulmonary nodule detection algorithm,” *J. Xray. Sci. Technol.*, vol. 28, no. 3, pp. 427–447, 2020, doi: 10.3233/XST-200656.
- [38] Marco Simó Josep Maria, Marco Galindo Maria Jesús, Macau Nadal Rafael, Pastor Collado Joan Antoni, Rodríguez Bermúdez José Ramón, and Guitart Hormigo Isabel, *Sistemas de información(en las organizaciones)*. Barcelona, 2013. Accessed: Sep. 20, 2018. [Online]. Available: <https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=3219604&query=sistemas+de+informacion#>
- [39] Joyanes Aguilar Luis, *Programación en c++ Algoritmos estructura de datos y objetos.*, MC Graw Hi. Madrid, 2006. Accessed: Sep. 20, 2018. [Online]. Available: <https://ebookcentral.proquest.com/lib/bibsipansp/reader.action?docID=3194877&query=lenguaje+de+programacion+dev+c%2B%2B#>
- [40] J. Cruz V., “Programación en Java Francisco Javier Cruz Vílchez ISBN-13: Pendiente,” 2012.
- [41] A. Downey, J. Elkner, and C. Meyers, *Aprenda a Pensar Como un*

- Programador*, vol. 8, no. 2. 2002. [Online]. Available:
<http://www.unav.es/SI/manuales/Java/indice.html>[http://www.iaa.es/python/c
urso-python-para-principiantes.pdf](http://www.iaa.es/python/curso-python-para-principiantes.pdf)
- [42] Z. Wang, Q. Ruan, and G. An, "Author ' s Accepted Manuscript Facial expression recognition using sparse local Fisher discriminant analysis Reference : To appear in : Neurocomputing," *Neurocomputing*, 2015, doi: 10.1016/j.neucom.2015.09.083.
- [43] X. Bai *et al.*, "Vegetation segmentation robust to illumination variations based on clustering and morphology modelling," *Biosyst. Eng.*, vol. 125, pp. 80–97, 2014, doi: 10.1016/j.biosystemseng.2014.06.015.
- [44] W. Guo, U. K. Rage, and S. Ninomiya, "Illumination invariant segmentation of vegetation for time series wheat images based on decision tree model," *Comput. Electron. Agric.*, vol. 96, pp. 58–66, 2013, doi: 10.1016/j.compag.2013.04.010.
- [45] O. CV, "Color conversions." [Online]. Available:
[https://docs.opencv.org/3.4.15/de/d25/imgproc_color_conversions.html#color_c
onvert_rgb_lab](https://docs.opencv.org/3.4.15/de/d25/imgproc_color_conversions.html#color_convert_rgb_lab)

ANEXOS

ANEXO N° 1 – RESOLUCIÓN DE AMPLIACIÓN DEL PROYECTO DE INVESTIGACIÓN

 Universidad Señor de Sipán

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°0745-2023/FIAU-USS
Pimentel, 30 de septiembre de 2023

ANEXO

APROBAR LA AMPLIACIÓN DE VIGENCIA DE TEMA DE TESIS

	APellidos y Nombres	TESIS	RESOLUCION APROBACIÓN/AMPLIACIÓN	AMPLIACION DE VIGENCIA DE TEMA TESIS
1	CAMPOS BARRERA SANDRO PAUL PASTOR OLIVA CESAR AUGUSTO	Implementación de un método de clasificación para detectar la deserción de estudiantes de la carrera de Ingeniería de Industrias Alimentarias de una universidad nacional peruana basado en aprendizaje de maquina	0445-2021/FIAU-USS	Se amplía hasta el 31 de diciembre del 2023
2	TOCTO LOPEZ CHRISTIAN ALEXIS	Evaluación del nivel de seguridad que ofrecen algoritmos criptográficos en una red privada virtual	2320-2020/FIAU-USS	Se amplía hasta el 31 de diciembre del 2023
3	SANDOVAL ODAR WILLIAM	Comparación de algoritmos de segmentación de imágenes digitales de plantas de arroz en ambientes no controlados	2322-2020/FIAU-USS	Se amplía hasta el 31 de diciembre del 2023
4	APARI VALENZUELA OSCAR	Evaluación de herramientas informáticas para el tratamiento de riesgos en la norma NTP-ISO/IEC 27001 2014 en una facultad de ingeniería del sistema universitario peruano	0434-2021/FIAU-USS	Se amplía hasta el 31 de diciembre del 2023

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE


 Dr. Victor Alexi Tuesta Manriquez
Decano (E) / Facultad de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.


 Dr. Halyn Alvarez Vásquez
Secretario Académico Facultad de
Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

**ANEXO Nº 2 - EVALUACIÓN DE ALGORITMOS DE SEGMENTACIÓN UTILIZADOS
EN DISTINTOS ESTUDIOS**

Ítem	Algoritmo	Precisión	Rendimiento	Autor	Escenario
1	MASK RCNN Y DBSCAN	Normal 0.897 - 0.985 Estrés 0.888-0.982	-	[5]	Segmentaron hojas de plántulas de álamo en condiciones normales y bajo estrés por metales pesados.
2	K - MEANS	0.982-0.90	0.04	[7]	Segmentaron hojas de tomate con enfermedades de rizada (Hoja amarilla) y Mosaico(Hoja marrón).
3	MASK RCNN	89.85%	-	[6]	Segmentaron frutas para un robot cosechador de fresas en un entorno no estructural.
4	Red UNET mejorada	73%	-	[13]	Segmentaron nódulos pulmonares bajo imágenes de entrada tipo TC.
5	Mean shift adaptive	69% -56%	3.71 s. – 2.08 s.	[12]	Realizaron borrado de ruido de imágenes IR y imágenes visuales, siendo naves marinas el objeto de interés.
6	KMEANS	90.43% - 92.15%	-	[11]	Segmentaron hojas enfermas de manzanas y hojas enfermas de pepino.

7	OTSU MEJORADO	-	0.003 s.	[10]	Segmentaron filas plántulas de arroz.
8	PSO-SVM	FPR y FNR inferior al 10%	-	[8]	Realizaron comparaciones de reconocimiento en alojamiento de arroz en imágenes RGB y imágenes TIF.
9	FCN	97% - 91%	-	[16]	Contaron manzanas y naranjas en un ambiente estructural.
10	FUZZY C-PARTITION ENTROPY	77%	5.3 s y 10.8s	[15]	Segmentaron imágenes en escala de grises y imágenes a color.
11	DTSM	76% - 79 %	-	[14]	Realizaron segmentación del dosel vegetal a plántulas de arroz en condiciones de iluminación natural.
12	SLIC	77,3%, 99,3% y 97,0%	-	[18]	Segmentaron plantas de algodón en campo.
13	SLFDA	77.2%	-	[42]	Lograron reconocer expresiones faciales.
14	PSO-MM	88.1% - 91.7%	-	[43]	Segmentar plantas de arroz y de algodón en condiciones de iluminación variable.
15	DTSM	80%	-	[44]	Segmentación de Imágenes de trigo bajo condiciones diferentes de iluminación.

ANEXO Nº 3 - EL RENDIMIENTO (%) DE DIFERENTES ALGORITMOS EN IMÁGENES DE ARROZ.

Algoritmo	IoU		Se		Sp		Ac		Autor
	μ	σ	μ	σ	μ	σ	μ	σ	
EXG	21.83	3.36	35.95	6.95	96.87	4.12	76.92	4.12	
EASA	60.41	13.50	74.62	17.59	92.90	10.01	86.40	5.12	
NDI	62.55	10.52	64.58	10.82	93.12	10.91	88.69	5.64	
CIVE	20.45	3.39	72.89	6.98	94.46	3.47	94.56	3.45	
VEG	76.83	12.56	91.52	9.53	92.99	10.59	91.80	6.11	
MS	12.82	8.07	21.37	18.10	93.07	11.57	62.15	11.52	
MS_FLD	25.11	4.33	91.85	8.95	93.99	3.69	94.11	3.52	
RT	68.94	18.75	79.55	18.99	92.65	9.13	87.76	15.47	
LCM	23.03	4.88	69.87	6.87	78.84	8.30	80.05	7.70	
AP_HI	12.83	4.77	41.98	18.19	93.15	10.61	68.84	15.17	
DTSM	53.98	6.98	61.24	2.65	93.84	9.78	91.56	6.13	[36]
ES	33.74	4.54	70.45	4.77	93.65	3.54	93.87	3.37	
Labseg	23.07	4.65	40.79	12.67	93.71	3.54	93.87	3.37	
MS_CIVE	81.20	9.16	85.45	11.20	93.09	9.21	94.59	2.01	
MS_EXG	71.98	8.19	84.36	10.26	96.80	0.85	96.82	0.84	
MS_VVI	76.48	9.43	73.85	9.84	90.24	10.60	93.68	3.47	
DA	42.85	4.78	75.25	6.54	94.20	3.64	94.34	3.45	
DWT	74.54	14.61	84.30	15.96	93.04	10.71	97.03	0.95	
PFMRF	63.48	3.69	85.66	6.44	90.26	4.99	88.97	3.69	
JOINT_CT	66.08	3.98	73.77	4.98	94.11	3.09	92.55	4.98	
LUV_SVM	54.88	6.19	68.16	3.89	91.47	6.49	90.72	4.35	

ANEXO Nº 4 - RENDIMIENTO DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ADELANTE EN DÍA NUBLADO UTILIZANDO EL ALGORITMO K-MEANS.

Ítem	Imágenes	Tiempo De Respuesta (Ms)	Consumo De Memoria RAM (%)	Consumo De CPU (%)
1	IMG_2142.JPG	4.88	71.40	3.70
2	IMG_2143.JPG	16.75	71.30	4.40
3	IMG_2144.JPG	26.11	71.50	3.20
4	IMG_2145.JPG	35.14	71.40	4.50
5	IMG_2146.JPG	45.02	71.40	4.90
6	IMG_2148.JPG	53.75	71.40	4.00
7	IMG_2149.JPG	63.63	71.90	4.00
8	IMG_2150.JPG	73.89	71.40	3.10
9	IMG_2151.JPG	84.13	71.40	3.20
10	IMG_2152.JPG	94.12	71.10	5.40
11	IMG_2153.JPG	102.88	71.40	3.70
12	IMG_2157.JPG	112.11	71.20	4.00
13	IMG_2158.JPG	122.44	71.30	5.20

14	IMG_2159.JPG	133.24	71.20	4.60
15	IMG_2160.JPG	142.30	71.00	5.70
16	IMG_2161.JPG	151.09	70.90	3.80
17	IMG_2162.JPG	160.87	70.90	3.90
18	IMG_2163.JPG	170.88	71.00	4.00
19	IMG_2164.JPG	180.94	70.90	3.90
20	IMG_2165.JPG	192.28	71.00	3.60
21	IMG_2166.JPG	202.65	71.00	4.10
22	IMG_2167.JPG	213.28	71.00	4.50
23	IMG_2168.JPG	223.19	71.30	5.70
24	IMG_2169.JPG	232.51	70.90	4.80
25	IMG_2170.JPG	243.56	70.90	4.70
	Promedio	123.27	71.20	4.26

Fuente: Elaboración propia

ANEXO Nº 5 - RENDIMIENTO DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ATRÁS EN DÍA NUBLADO UTILIZANDO EL ALGORITMO K-MEANS.

Ítem	Imágenes	Tiempo De Respuesta (Ms)	Consumo De Memoria RAM (%)	Consumo De CPU (%)
1	IMG_2116.JPG	6.63	68.20	10.60
2	IMG_2118.JPG	18.62	68.10	10.80
3	IMG_2119.JPG	31.09	68.20	12.90
4	IMG_2120.JPG	43.50	68.30	10.50
5	IMG_2121.JPG	54.87	68.20	10.60
6	IMG_2122.JPG	67.92	68.20	19.80
7	IMG_2123.JPG	80.27	68.50	18.40
8	IMG_2124.JPG	92.39	68.30	12.10
9	IMG_2125.JPG	104.70	68.20	12.20
10	IMG_2126.JPG	118.44	68.20	17.50
11	IMG_2127.JPG	130.81	68.00	10.00
12	IMG_2128.JPG	142.24	68.10	15.20
13	IMG_2129.JPG	154.63	68.10	10.40

14	IMG_2130.JPG	167.85	68.00	15.40
15	IMG_2131.JPG	182.77	69.20	5.60
16	IMG_2132.JPG	195.98	69.20	5.00
17	IMG_2133.JPG	207.02	69.20	6.00
18	IMG_2134.JPG	218.32	68.90	4.10
19	IMG_2135.JPG	229.33	68.60	3.70
20	IMG_2136.JPG	239.42	68.70	7.20
21	IMG_2137.JPG	250.45	68.70	3.30
22	IMG_2138.JPG	260.68	68.60	3.30
23	IMG_2139.JPG	272.84	68.90	3.00
24	IMG_2140.JPG	283.46	68.60	4.50
25	IMG_2141.JPG	294.49	68.70	4.20
	Promedio	153.95	68.48	9.45

Fuente: Elaboración propia

ANEXO Nº 6 - RENDIMIENTO DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA IZQUIERDA EN DÍA SOLEADO UTILIZANDO EL ALGORITMO K-MEANS.

Ítem	Imágenes	Tiempo De Respuesta (Ms)	Consumo De Memoria RAM (%)	Consumo De CPU (%)
1	IMG_1938.JPG	11.44	65.90	9.50
2	IMG_1939.JPG	30.06	65.90	13.50
3	IMG_1940.JPG	49.55	66.00	15.50
4	IMG_1941.JPG	66.00	66.10	16.80
5	IMG_1942.JPG	81.82	66.10	15.60
6	IMG_1944.JPG	94.38	66.10	12.70
7	IMG_1945.JPG	110.92	66.10	14.30
8	IMG_1950.JPG	124.83	66.10	9.80
9	IMG_1951.JPG	140.23	66.00	10.50
10	IMG_1953.JPG	151.69	66.00	10.30
11	IMG_1955.JPG	163.92	65.90	9.50
12	IMG_1960.JPG	178.01	66.00	11.60
13	IMG_1961.JPG	190.55	65.90	10.60
14	IMG_1962.JPG	204.04	65.90	9.50

15	IMG_1963.JPG	216.61	65.90	12.10
16	IMG_1964.JPG	229.99	65.80	8.40
17	IMG_1965.JPG	242.25	66.00	15.20
18	IMG_1966.JPG	255.15	66.00	16.50
19	IMG_1968.JPG	268.79	66.10	9.40
20	IMG_1972.JPG	288.13	66.00	22.80
21	IMG_1973.JPG	300.33	66.20	7.00
22	IMG_1974.JPG	326.27	65.20	13.80
23	IMG_1975.JPG	342.28	65.60	8.10
24	IMG_1976.JPG	355.82	66.10	11.40
25	IMG_1980.JPG	368.43	65.50	9.60
	Promedio	191.66	65.94	12.16

Fuente: Elaboración propia

ANEXO Nº 7 - RENDIMIENTO DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA DERECHA EN DÍA SOLEADO UTILIZANDO EL ALGORITMO K-MEANS.

Ítem	Imágenes	Tiempo De Respuesta (Ms)	Consumo De Memoria RAM (%)	Consumo De CPU (%)
1	IMG_2000.JPG	8.66	66.20	15.50
2	IMG_2002.JPG	20.24	66.00	13.80
3	IMG_2003.JPG	32.48	66.10	12.10
4	IMG_2004.JPG	43.66	66.10	21.70
5	IMG_2012.JPG	54.05	65.10	17.00
6	IMG_2013.JPG	66.53	65.10	11.60
7	IMG_2014.JPG	78.11	65.20	11.80
8	IMG_2015.JPG	89.50	65.60	11.70
9	IMG_2016.JPG	100.75	65.20	13.60
10	IMG_2017.JPG	111.45	65.10	13.70
11	IMG_2018.JPG	122.76	65.00	9.20
12	IMG_2019.JPG	134.73	64.80	18.70
13	IMG_2020.JPG	147.87	64.80	10.00
14	IMG_2021.JPG	160.01	64.80	11.20

15	IMG_2022.JPG	171.54	64.80	10.00
16	IMG_2023.JPG	183.65	64.90	15.90
17	IMG_2025.JPG	195.83	64.80	11.70
18	IMG_2026.JPG	208.02	64.80	13.70
19	IMG_2030.JPG	219.96	64.60	10.40
20	IMG_2031.JPG	231.48	64.70	17.00
21	IMG_2032.JPG	244.87	64.60	13.40
22	IMG_2033.JPG	258.21	64.80	13.10
23	IMG_2034.JPG	289.20	63.80	12.00
24	IMG_2035.JPG	303.30	64.20	10.40
25	IMG_2036.JPG	315.04	64.20	12.20
	Promedio	151.68	65.01	13.26

Fuente: Elaboración propia

ANEXO Nº 8 - RENDIMIENTO DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ADELANTE EN DÍA NUBLADO UTILIZANDO EL ALGORITMO DBSCAN.

Ítem	Imágenes	Tiempo De Respuesta (Ms)	Consumo De Memoria RAM (%)	Consumo De CPU (%)
1	IMG_2142.jpg	5.02	109.30	21.10
2	IMG_2143.jpg	20.80	109.10	13.10
3	IMG_2144.jpg	36.34	109.20	13.10
4	IMG_2145.jpg	53.74	109.10	16.30
5	IMG_2146.jpg	72.49	109.30	23.70
6	IMG_2148.jpg	91.83	109.30	42.80
7	IMG_2149.jpg	110.19	110.40	44.00
8	IMG_2150.jpg	128.20	110.40	38.20
9	IMG_2151.jpg	146.61	110.40	57.10
10	IMG_2152.jpg	166.61	111.10	40.50
11	IMG_2153.jpg	184.45	106.90	27.70
12	IMG_2157.jpg	202.77	106.80	16.70
13	IMG_2158.jpg	229.25	105.90	27.50

14	IMG_2159.jpg	255.35	107.60	14.50
15	IMG_2160.jpg	279.07	107.00	42.20
16	IMG_2161.jpg	300.98	110.90	15.10
17	IMG_2162.jpg	322.69	107.00	24.10
18	IMG_2163.jpg	343.51	107.10	28.70
19	IMG_2164.jpg	365.59	108.30	19.70
20	IMG_2165.jpg	383.48	106.60	17.00
21	IMG_2166.jpg	400.62	106.70	31.20
22	IMG_2167.jpg	418.32	108.50	29.90
23	IMG_2168.jpg	437.05	106.70	17.30
24	IMG_2169.jpg	455.01	108.20	7.20
25	IMG_2170.jpg	474.73	106.50	23.90
	Promedio	235.39	108.33	26.10

Fuente: Elaboración propia

ANEXO Nº 9 - RENDIMIENTO DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ATRÁS EN DÍA NUBLADO UTILIZANDO EL ALGORITMO DBSCAN.

Ítem	Imágenes	Tiempo De Respuesta (Ms)	Consumo De Memoria RAM (%)	Consumo De CPU (%)
1	IMG_2116.jpg	12.44	118.10	5.70
2	IMG_2118.jpg	37.56	118.40	30.80
3	IMG_2119.jpg	56.30	118.30	3.10
4	IMG_2120.jpg	75.28	118.90	32.10
5	IMG_2121.jpg	93.84	118.70	4.80
6	IMG_2122.jpg	114.43	118.60	8.40
7	IMG_2123.jpg	137.59	119.10	3.90
8	IMG_2124.jpg	154.94	119.00	19.70
9	IMG_2125.jpg	174.55	117.20	8.10
10	IMG_2126.jpg	192.99	117.20	9.00
11	IMG_2127.jpg	211.17	118.60	8.90
12	IMG_2128.jpg	229.41	117.30	8.50
13	IMG_2129.jpg	247.88	118.40	5.70
14	IMG_2130.jpg	264.62	117.20	6.50

15	IMG_2131.jpg	282.13	117.20	12.50
16	IMG_2132.jpg	300.97	117.30	5.80
17	IMG_2133.jpg	323.99	118.30	22.20
18	IMG_2134.jpg	345.62	118.50	7.00
19	IMG_2135.jpg	364.37	117.30	10.30
20	IMG_2136.jpg	381.85	117.10	4.20
21	IMG_2137.jpg	397.46	118.00	9.30
22	IMG_2138.jpg	413.94	117.30	3.70
23	IMG_2139.jpg	431.93	117.40	9.90
24	IMG_2140.jpg	450.55	117.30	4.60
25	IMG_2141.jpg	467.54	118.80	7.90
	Promedio	246.54	117.98	10.10

Fuente: Elaboración propia

ANEXO Nº 10 – RENDIMIENTO DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA IZQUIERDA EN DÍA SOLEADO UTILIZANDO EL ALGORITMO DBSCAN.

Ítem	Imágenes	Tiempo De Respuesta (Ms)	Consumo De Memoria Ram (%)	Consumo De CPU (%)
1	IMG_1938.jpg	9.57	95.60	4.00
2	IMG_1939.jpg	27.73	95.40	3.00
3	IMG_1940.jpg	44.00	95.80	7.60
4	IMG_1941.jpg	59.02	95.80	6.70
5	IMG_1942.jpg	75.64	96.00	5.20
6	IMG_1944.jpg	91.18	95.80	8.50
7	IMG_1945.jpg	108.71	96.00	4.30
8	IMG_1950.jpg	124.43	95.80	5.90
9	IMG_1951.jpg	141.86	96.00	6.40
10	IMG_1953.jpg	157.39	96.10	3.80
11	IMG_1955.jpg	173.89	95.50	2.20
12	IMG_1960.jpg	190.22	94.90	6.60
13	IMG_1961.jpg	206.89	95.40	5.30
14	IMG_1962.jpg	223.37	95.00	4.00

15	IMG_1963.jpg	239.99	95.40	4.90
16	IMG_1964.jpg	257.32	96.60	4.20
17	IMG_1965.jpg	272.98	97.30	6.00
18	IMG_1966.jpg	290.47	95.10	5.00
19	IMG_1968.jpg	308.01	95.20	7.00
20	IMG_1972.jpg	324.43	95.20	3.00
21	IMG_1973.jpg	339.74	97.50	4.20
22	IMG_1974.jpg	355.50	95.10	5.50
23	IMG_1975.jpg	371.10	95.00	6.60
24	IMG_1976.jpg	388.03	96.80	4.70
25	IMG_1980.jpg	403.73	95.00	9.40
	Promedio	207.40	95.73	5.36

Fuente: Elaboración propia

ANEXO Nº 11 - RENDIMIENTO DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA DERECHA EN DÍA SOLEADO UTILIZANDO EL ALGORITMO DBSCAN.

Ítem	Imágenes	Tiempo De Respuesta (Ms)	Consumo De Memoria RAM (%)	Consumo De CPU (%)
1	IMG_2000.jpg	6.85	116.30	26.20
2	IMG_2002.jpg	32.37	117.50	14.70
3	IMG_2003.jpg	47.60	116.30	26.20
4	IMG_2004.jpg	63.53	115.90	16.60
5	IMG_2012.jpg	89.24	117.40	36.80
6	IMG_2013.jpg	105.12	116.10	4.40
7	IMG_2014.jpg	128.84	116.00	3.90
8	IMG_2015.jpg	146.03	117.40	9.70
9	IMG_2016.jpg	171.45	116.00	23.60
10	IMG_2017.jpg	185.55	116.00	3.20
11	IMG_2018.jpg	200.14	115.90	3.70
12	IMG_2019.jpg	215.28	117.10	7.90
13	IMG_2020.jpg	235.93	115.80	3.30

14	IMG_2021.jpg	259.47	115.70	6.70
15	IMG_2022.jpg	284.06	117.20	11.70
16	IMG_2023.jpg	305.15	116.50	13.90
17	IMG_2025.jpg	330.50	114.70	4.90
18	IMG_2026.jpg	346.62	112.40	3.20
19	IMG_2030.jpg	361.81	111.60	8.80
20	IMG_2031.jpg	379.59	111.50	5.10
21	IMG_2032.jpg	396.53	112.90	13.80
22	IMG_2033.jpg	417.32	111.40	7.60
23	IMG_2034.jpg	438.92	111.40	10.20
24	IMG_2035.jpg	467.17	112.70	18.60
25	IMG_2036.jpg	491.98	112.20	7.90
	Promedio	244.28	114.96	11.70

Fuente: Elaboración propia

ANEXO Nº 12 – RENDIMIENTO TOTAL DE 100 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ADELANTE, HACIA ATRÁS, EN POSICION EN DIAGONAL HACIA LA IZQUIERDA Y EN POSICION EN DIAGONAL HACIA LA DERECHA EN DIA NUBLADO Y EN DIA SOLEADO UTILIZANDO ALGORITMOS K-MEANS Y DBSCAN.

Rendimiento total en segmentación de imágenes digitales de plantas de arroz.		
	K-MEANS	DBSCAN
Tiempo De Respuesta (ms)	155.14	233.40
Consumo De Memoria Ram	67.66%	109.24%
Consumo De CPU	9.78%	13.32%

Fuente: Elaboración propia

**ANEXO Nº 13 - SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ADELANTE EN DÍA NUBLADO
UTILIZANDO EL ALGORITMO K-MEANS.**

Ítem	Imágenes	Verdaderos Positivos	Falsos Positivos	Verdaderos Negativos	Falsos Negativos	Exactitud	Sensibilidad	Especificidad
1	IMG_2142.JPG	161	181	479839	70	0.97	0.70	1.00
2	IMG_2143.JPG	189	240	479811	71	0.96	0.73	1.00
3	IMG_2144.JPG	44	438	479956	118	0.89	0.27	1.00
4	IMG_2145.JPG	48	206	479952	81	0.95	0.37	1.00
5	IMG_2146.JPG	50	166	479950	62	0.96	0.45	1.00
6	IMG_2148.JPG	290	210	479710	32	0.96	0.90	1.00
7	IMG_2149.JPG	165	129	479835	7	0.94	0.96	1.00
8	IMG_2150.JPG	64	66	479936	7	0.93	0.90	1.00
9	IMG_2151.JPG	127	83	479873	11	0.95	0.92	1.00
10	IMG_2152.JPG	90	131	479910	34	0.97	0.73	1.00
11	IMG_2153.JPG	120	88	479880	15	0.98	0.89	1.00
12	IMG_2157.JPG	96	154	479904	20	0.91	0.83	1.00

13	IMG_2158.JPG	200	221	479800	39	0.91	0.84	1.00
14	IMG_2159.JPG	121	107	479879	11	0.93	0.92	1.00
15	IMG_2160.JPG	188	137	479812	10	0.95	0.95	1.00
16	IMG_2161.JPG	197	118	479803	17	0.98	0.92	1.00
17	IMG_2162.JPG	287	192	479713	43	0.97	0.87	1.00
18	IMG_2163.JPG	231	201	479769	31	0.94	0.88	1.00
19	IMG_2164.JPG	122	164	479878	39	0.87	0.76	1.00
20	IMG_2165.JPG	173	213	479827	28	0.92	0.86	1.00
21	IMG_2166.JPG	33	46	479967	2	0.85	0.94	1.00
22	IMG_2167.JPG	115	143	479885	12	0.91	0.91	1.00
23	IMG_2168.JPG	123	136	479877	34	0.86	0.78	1.00
24	IMG_2169.JPG	156	200	479844	31	0.85	0.83	1.00
25	IMG_2170.JPG	186	203	479814	30	0.86	0.86	1.00
Promedio						0.93	0.80	1.00

Fuente: Elaboración propia



ANEXO Nº 14 - SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ATRÁS EN DÍA NUBLADO UTILIZANDO EL ALGORITMO K-MEANS.

Ítem	Imágenes	Verdaderos Positivos	Falsos Positivos	Verdaderos Negativos	Falsos Negativos	Exactitud	Sensibilidad	Especificidad
1	IMG_2116.JPG	92	100	479908	6	0.93	0.94	1.00
2	IMG_2118.JPG	51	116	479949	2	0.90	0.96	1.00
3	IMG_2119.JPG	100	108	479900	3	0.90	0.97	1.00
4	IMG_2120.JPG	67	107	479933	4	0.91	0.94	1.00
5	IMG_2121.JPG	145	130	479855	16	0.89	0.90	1.00
6	IMG_2122.JPG	45	74	479955	2	0.89	0.96	1.00
7	IMG_2123.JPG	147	125	479853	4	0.95	0.97	1.00
8	IMG_2124.JPG	88	89	479912	3	0.88	0.97	1.00
9	IMG_2125.JPG	232	147	479768	7	0.96	0.97	1.00
10	IMG_2126.JPG	130	126	479870	1	0.94	0.99	1.00
11	IMG_2127.JPG	193	223	479807	93	0.97	0.67	1.00

12	IMG_2128.JPG	238	236	479762	64	0.97	0.79	1.00
13	IMG_2129.JPG	138	149	479862	1	0.93	0.99	1.00
14	IMG_2130.JPG	211	146	479789	10	0.95	0.95	1.00
15	IMG_2131.JPG	263	186	479737	13	0.94	0.95	1.00
16	IMG_2132.JPG	214	139	479786	29	0.97	0.88	1.00
17	IMG_2133.JPG	194	333	479806	99	0.96	0.66	1.00
18	IMG_2134.JPG	66	536	479934	286	0.88	0.19	1.00
19	IMG_2135.JPG	34	490	479966	224	0.86	0.13	1.00
20	IMG_2136.JPG	87	344	479913	166	0.93	0.34	1.00
21	IMG_2137.JPG	133	310	479867	188	0.94	0.41	1.00
22	IMG_2138.JPG	290	301	479710	112	0.97	0.72	1.00
23	IMG_2139.JPG	186	175	479814	8	0.92	0.96	1.00
24	IMG_2140.JPG	163	149	479837	6	0.94	0.96	1.00
25	IMG_2141.JPG	198	143	479802	6	0.96	0.97	1.00
Promedio						0.93	0.81	1.00

Fuente: Elaboración propia

ANEXO Nº 15 - SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA IZQUIERDA EN DÍA SOLEADO UTILIZANDO EL ALGORITMO K-MEANS.

Ítem	Imágenes	Verdaderos Positivos	Falsos Positivos	Verdaderos Negativos	Falsos Negativos	Exactitud	Sensibilidad	Especificidad
1	IMG_1938.JPG	53	121	479947	2	0.83	0.96	1.00
2	IMG_1939.JPG	51	131	479949	5	0.86	0.91	1.00
3	IMG_1940.JPG	52	92	479948	2	0.82	0.96	1.00
4	IMG_1941.JPG	51	129	479949	4	0.84	0.93	1.00
5	IMG_1942.JPG	74	179	479926	2	0.84	0.97	1.00
6	IMG_1944.JPG	124	197	479876	12	0.88	0.91	1.00
7	IMG_1945.JPG	69	108	479931	7	0.87	0.91	1.00
8	IMG_1950.JPG	142	217	479858	7	0.90	0.95	1.00
9	IMG_1951.JPG	129	283	479871	142	0.95	0.48	1.00
10	IMG_1953.JPG	217	270	479783	86	0.97	0.72	1.00
11	IMG_1955.JPG	152	329	479848	145	0.95	0.51	1.00

12	IMG_1960.JPG	71	137	479929	8	0.79	0.90	1.00
13	IMG_1961.JPG	81	155	479919	2	0.81	0.98	1.00
14	IMG_1962.JPG	137	168	479863	10	0.88	0.93	1.00
15	IMG_1963.JPG	114	161	479886	4	0.82	0.97	1.00
16	IMG_1964.JPG	93	128	479907	9	0.84	0.91	1.00
17	IMG_1965.JPG	101	148	479899	10	0.84	0.91	1.00
18	IMG_1966.JPG	96	141	479904	8	0.85	0.92	1.00
19	IMG_1968.JPG	171	222	479829	7	0.92	0.96	1.00
20	IMG_1972.JPG	55	105	479945	2	0.86	0.96	1.00
21	IMG_1973.JPG	103	243	479897	7	0.89	0.94	1.00
22	IMG_1974.JPG	112	196	479888	12	0.90	0.90	1.00
23	IMG_1975.JPG	174	178	479826	5	0.95	0.97	1.00
24	IMG_1976.JPG	74	111	479926	1	0.94	0.99	1.00
25	IMG_1980.JPG	152	313	479848	136	0.95	0.53	1.00
Promedio						0.88	0.88	1.00

Fuente: Elaboración propia

ANEXO Nº 16 - SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA DERECHA EN DÍA SOLEADO UTILIZANDO EL ALGORITMO K-MEANS.

Ítem	Imágenes	Verdaderos Positivos	Falsos Positivos	Verdaderos Negativos	Falsos Negativos	Exactitud	Sensibilidad	Especificidad
1	IMG_2000.JPG	64	434	479936	193	0.92	0.25	1.00
2	IMG_2002.JPG	109	445	479891	200	0.93	0.35	1.00
3	IMG_2003.JPG	62	508	479938	231	0.90	0.21	1.00
4	IMG_2004.JPG	42	434	479958	202	0.91	0.17	1.00
5	IMG_2012.JPG	328	279	479672	103	0.97	0.76	1.00
6	IMG_2013.JPG	286	233	479714	84	0.97	0.77	1.00
7	IMG_2014.JPG	85	515	479915	283	0.90	0.23	1.00
8	IMG_2015.JPG	126	420	479874	236	0.93	0.35	1.00

9	IMG_2016.JPG	230	250	479770	79	0.97	0.74	1.00
10	IMG_2017.JPG	153	351	479847	193	0.94	0.44	1.00
11	IMG_2018.JPG	256	381	479744	138	0.96	0.65	1.00
12	IMG_2019.JPG	336	219	479664	32	0.97	0.91	1.00
13	IMG_2020.JPG	335	275	479665	95	0.97	0.78	1.00
14	IMG_2021.JPG	147	453	479853	199	0.94	0.42	1.00
15	IMG_2022.JPG	286	241	479714	80	0.97	0.78	1.00
16	IMG_2023.JPG	289	205	479711	92	0.97	0.76	1.00
17	IMG_2025.JPG	176	404	479824	178	0.95	0.50	1.00
18	IMG_2026.JPG	304	303	479696	61	0.97	0.83	1.00
19	IMG_2030.JPG	296	189	479704	30	0.98	0.91	1.00
20	IMG_2031.JPG	260	268	479740	61	0.98	0.81	1.00
21	IMG_2032.JPG	364	265	479636	26	0.98	0.93	1.00

22	IMG_2033.JPG	224	145	479776	8	0.96	0.97	1.00	
23	IMG_2034.JPG	296	294	479704	59	0.91	0.83	1.00	
24	IMG_2035.JPG	210	256	479790	34	0.91	0.86	1.00	
25	IMG_2036.JPG	167	154	479833	15	0.85	0.92	1.00	
						Promedio	0.95	0.65	1.00

Fuente: Elaboración propia

**ANEXO Nº 17 - SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ADELANTE EN DÍA NUBLADO
UTILIZANDO EL ALGORITMO DBSCAN.**

Ítem	Imágenes	Verdaderos Positivos	Falsos Positivos	Verdaderos Negativos	Falsos Negativos	Exactitud	Sensibilidad	Especificidad
1	IMG_2142.jpg	0	75	480000	69	0.51	0.00	1.00
2	IMG_2143.jpg	0	9	480000	33	0.71	0.00	1.00
3	IMG_2144.jpg	0	10	480000	21	0.63	0.00	1.00
4	IMG_2145.jpg	0	9	480000	23	0.65	0.00	1.00
5	IMG_2146.jpg	0	9	480000	17	0.65	0.00	1.00
6	IMG_2148.jpg	0	6	480000	43	0.69	0.00	1.00
7	IMG_2149.jpg	0	20	480000	50	0.64	0.00	1.00
8	IMG_2150.jpg	0	25	480000	30	0.63	0.00	1.00
9	IMG_2151.jpg	0	9	480000	74	0.65	0.00	1.00
10	IMG_2152.jpg	0	9	480000	34	0.63	0.00	1.00
11	IMG_2153.jpg	0	10	480000	38	0.66	0.00	1.00
12	IMG_2157.jpg	0	28	480000	239	0.46	0.00	1.00
13	IMG_2158.jpg	0	36	480000	253	0.47	0.00	1.00

14	IMG_2159.jpg	0	34	480000	154	0.55	0.00	1.00	
15	IMG_2160.jpg	0	49	480000	141	0.59	0.00	1.00	
16	IMG_2161.jpg	0	18	480000	74	0.59	0.00	1.00	
17	IMG_2162.jpg	0	15	480000	61	0.64	0.00	1.00	
18	IMG_2163.jpg	0	32	480000	75	0.62	0.00	1.00	
19	IMG_2164.jpg	0	14	480000	70	0.62	0.00	1.00	
20	IMG_2165.jpg	0	39	480000	83	0.61	0.00	1.00	
21	IMG_2166.jpg	0	23	480000	95	0.60	0.00	1.00	
22	IMG_2167.jpg	0	16	480000	70	0.61	0.00	1.00	
23	IMG_2168.jpg	2	42	479998	147	0.61	0.01	1.00	
24	IMG_2169.jpg	1	38	479999	116	0.64	0.01	1.00	
25	IMG_2170.jpg	0	35	480000	125	0.61	0.00	1.00	
						Promedio	0.61	0.00	1.00

Fuente: Elaboración propia

ANEXO Nº 18 - SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ATRÁS EN DÍA NUBLADO UTILIZANDO EL ALGORITMO DBSCAN.

Ítem	Imágenes	Verdaderos Positivos	Falsos Positivos	Verdaderos Negativos	Falsos Negativos	Exactitud	Sensibilidad	Especificidad
1	IMG_2116.jpg	0	75	480000	98	0.48	0.00	1.00
2	IMG_2118.jpg	0	88	480000	397	0.64	0.00	1.00
3	IMG_2119.jpg	0	123	480000	255	0.62	0.00	1.00
4	IMG_2120.jpg	0	86	480000	148	0.58	0.00	1.00
5	IMG_2121.jpg	0	75	480000	191	0.54	0.00	1.00
6	IMG_2122.jpg	1	58	479999	260	0.51	0.00	1.00
7	IMG_2123.jpg	0	71	480000	45	0.37	0.00	1.00
8	IMG_2124.jpg	0	41	480000	186	0.53	0.00	1.00
9	IMG_2125.jpg	0	51	480000	155	0.57	0.00	1.00
10	IMG_2126.jpg	0	67	480000	147	0.54	0.00	1.00
11	IMG_2127.jpg	0	47	480000	130	0.60	0.00	1.00
12	IMG_2128.jpg	0	35	480000	71	0.60	0.00	1.00
13	IMG_2129.jpg	0	46	480000	103	0.60	0.00	1.00

14	IMG_2130.jpg	0	50	480000	89	0.58	0.00	1.00	
15	IMG_2131.jpg	0	29	480000	69	0.64	0.00	1.00	
16	IMG_2132.jpg	0	59	480000	105	0.32	0.00	1.00	
17	IMG_2133.jpg	0	76	480000	132	0.29	0.00	1.00	
18	IMG_2134.jpg	0	43	480000	83	0.32	0.00	1.00	
19	IMG_2135.jpg	0	24	480000	33	0.50	0.00	1.00	
20	IMG_2136.jpg	0	23	480000	19	0.58	0.00	1.00	
21	IMG_2137.jpg	0	28	480000	56	0.59	0.00	1.00	
22	IMG_2138.jpg	0	44	480000	111	0.62	0.00	1.00	
23	IMG_2139.jpg	0	43	480000	153	0.59	0.00	1.00	
24	IMG_2140.jpg	0	41	480000	71	0.57	0.00	1.00	
25	IMG_2141.jpg	0	46	480000	109	0.53	0.00	1.00	
						Promedio	0.53	0.00	1.00

Fuente: Elaboración propia

ANEXO Nº 19 - SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA IZQUIERDA EN DÍA SOLEADO UTILIZANDO EL ALGORITMO DBSCAN.

Ítem	Imágenes	Verdaderos Positivos	Falsos Positivos	Verdaderos Negativos	Falsos Negativos	Exactitud	Sensibilidad	Especificidad
1	IMG_1938.jpg	0	10	480000	20	0.40	0.00	1.00
2	IMG_1939.jpg	0	4	480000	28	0.40	0.00	1.00
3	IMG_1940.jpg	0	1	480000	17	0.51	0.00	1.00
4	IMG_1941.jpg	0	11	480000	61	0.37	0.00	1.00
5	IMG_1942.jpg	0	18	480000	59	0.40	0.00	1.00
6	IMG_1944.jpg	0	1	480000	49	0.52	0.00	1.00
7	IMG_1945.jpg	0	3	480000	30	0.48	0.00	1.00
8	IMG_1950.jpg	0	2	480000	38	0.56	0.00	1.00
9	IMG_1951.jpg	0	7	480000	52	0.71	0.00	1.00
10	IMG_1953.jpg	0	4	480000	59	0.66	0.00	1.00
11	IMG_1955.jpg	0	2	480000	24	0.67	0.00	1.00
12	IMG_1960.jpg	0	11	480000	32	0.45	0.00	1.00

13	IMG_1961.jpg	0	10	480000	38	0.51	0.00	1.00	
14	IMG_1962.jpg	0	8	480000	20	0.50	0.00	1.00	
15	IMG_1963.jpg	0	20	480000	110	0.49	0.00	1.00	
16	IMG_1964.jpg	0	2	480000	35	0.60	0.00	1.00	
17	IMG_1965.jpg	0	7	480000	65	0.55	0.00	1.00	
18	IMG_1966.jpg	0	5	480000	88	0.56	0.00	1.00	
19	IMG_1968.jpg	0	1	480000	28	0.55	0.00	1.00	
20	IMG_1972.jpg	0	2	480000	35	0.47	0.00	1.00	
21	IMG_1973.jpg	0	12	480000	53	0.40	0.00	1.00	
22	IMG_1974.jpg	0	3	480000	18	0.52	0.00	1.00	
23	IMG_1975.jpg	0	8	480000	57	0.48	0.00	1.00	
24	IMG_1976.jpg	0	1	480000	30	0.54	0.00	1.00	
25	IMG_1980.jpg	0	10	480000	50	0.55	0.00	1.00	
						Promedio	0.51	0.00	1.00

Fuente: Elaboración propia

ANEXO Nº 20 - SEGMENTACIÓN DE 25 IMÁGENES DIGITALES DE PLANTAS DE ARROZ EN POSICIÓN DIAGONAL HACIA LA DERECHA EN DÍA SOLEADO UTILIZANDO EL ALGORITMO DBSCAN.

Ítem	Imágenes	Verdaderos Positivos	Falsos Positivos	Verdaderos Negativos	Falsos Negativos	Exactitud	Sensibilidad	Especificidad
1	IMG_2000.jpg	0	34	480000	351	0.32	0.00	1.00
2	IMG_2002.jpg	0	48	480000	84	0.73	0.00	1.00
3	IMG_2003.jpg	0	16	480000	81	0.80	0.00	1.00
4	IMG_2004.jpg	0	14	480000	76	0.25	0.00	1.00
5	IMG_2012.jpg	0	12	480000	17	0.75	0.00	1.00
6	IMG_2013.jpg	0	15	480000	42	0.81	0.00	1.00
7	IMG_2014.jpg	1	9	479999	45	0.79	0.02	1.00
8	IMG_2015.jpg	0	2	480000	63	0.25	0.00	1.00
9	IMG_2016.jpg	0	14	480000	24	0.81	0.00	1.00
10	IMG_2017.jpg	0	21	480000	48	0.81	0.00	1.00
11	IMG_2018.jpg	0	11	480000	54	0.78	0.00	1.00

12	IMG_2019.jpg	0	8	480000	98	0.11	0.00	1.00
13	IMG_2020.jpg	0	8	480000	73	0.11	0.00	1.00
14	IMG_2021.jpg	0	9	480000	50	0.26	0.00	1.00
15	IMG_2022.jpg	0	9	480000	35	0.47	0.00	1.00
16	IMG_2023.jpg	0	9	480000	39	0.23	0.00	1.00
17	IMG_2025.jpg	0	16	480000	59	0.82	0.00	1.00
18	IMG_2026.jpg	0	9	480000	69	0.78	0.00	1.00
19	IMG_2030.jpg	0	16	480000	121	0.58	0.00	1.00
20	IMG_2031.jpg	0	17	480000	42	0.58	0.00	1.00
21	IMG_2032.jpg	0	23	480000	160	0.44	0.00	1.00
22	IMG_2033.jpg	0	10	480000	63	0.50	0.00	1.00
23	IMG_2034.jpg	0	11	480000	98	0.29	0.00	1.00
24	IMG_2035.jpg	0	12	480000	63	0.41	0.00	1.00
25	IMG_2036.jpg	0	10	480000	56	0.37	0.00	1.00
Promedio						0.52	0.00	1.00

Fuente: Elaboración propia

ANEXO N° 21 – SEGMENTACION TOTAL DE 100 IMÁGENES DIGITALES DE PLANTAS DE ARROZ HACIA ADELANTE, HACIA ATRÁS, EN POSCION EN DIAGONAL HACIA LA IZQUIERDA Y EN POSICION EN DIAGONAL HACIA LA DERECHA EN DIA NUBLADO Y EN DIA SOLEADO UTILIZANDO ALGORITMOS K-MEANS Y DBSCAN.

Resultado total en segmentación de 100 imágenes de plantas de arroz.		
	K-MEANS	DBSCAN
Exactitud	92.25%	54.25%
Sensibilidad	78.50%	0
Especificidad	100%	100%

Fuente: Elaboración propia

NOMBRE DEL TRABAJO

Sandoval_Odar_william_Tesis - copia.docx

AUTOR

William Sandoval Odar

RECuento DE PALABRAS

23773 Words

RECuento DE CARACTERES

123920 Characters

RECuento DE PÁGINAS

144 Pages

TAMAÑO DEL ARCHIVO

18.3MB

FECHA DE ENTREGA

Mar 22, 2024 8:44 AM GMT-5

FECHA DEL INFORME

Mar 22, 2024 8:46 AM GMT-5**● 7% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 6% Base de datos de Internet
- Base de datos de Crossref
- 3% Base de datos de trabajos entregados
- 2% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Coincidencia baja (menos de 8 palabras)
- Material citado