

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Software

Cuantificación automática de malezas utilizando imágenes adquiridas por un dron en campos de cultivos de maíz y/o papa mediante la red neuronal convolucional ResNet.

Trabajo de grado previo a la obtención del título de Ingeniero de Software presentado ante la ilustre Universidad Técnica del Norte.

Autor:

Vinueza Cuaical Kevin Xavier

Director:

PhD. García Santillán Iván Danilo

Ibarra – Ecuador

2024



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100506541-0.		
APELLIDOS Y NOMBRES:	Vinueza Cuaical Kevin Xavier.		
DIRECCIÓN:	Ibarra, El Sagrario.		
EMAIL:	kxvinuezac@utn.edu.ec.		
TELÉFONO FIJO:	062602250.	TELÉFONO MÓVIL:	0969115164.

DATOS DE LA OBRA	
TÍTULO:	Cuantificación automática de malezas utilizando imágenes adquiridas por un dron en campos de cultivos de maíz y/o papa mediante la red neuronal convolucional ResNet.
AUTOR(ES):	Vinueza Cuaical Kevin Xavier.
FECHA:	21/02/2024.
PROGRAMA:	Pregrado.
TÍTULO POR EL QUE OPTA:	Ingeniero de Software.
DIRECTOR:	PhD. García Santillán Iván Danilo.

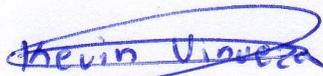
ASESOR:	PhD. Pusedá Chulde Marco Remigio.
----------------	-----------------------------------

2. CONSTANCIAS

El autor(es) manifiesta que la obra objeto de la presente autorización es original y se la desarrollo, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es(son) el(los) titular(es) de los derechos patrimoniales, que lo asume(n) la responsabilidad sobre el contenido de esta saldrá(n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 21 días del mes de febrero de 2024.

EL AUTOR:

A handwritten signature in blue ink that reads "Kevin Vinueza". The signature is written in a cursive style and is underlined with a blue line.

Nombre: Vinueza Cuaical Kevin Xavier

C.I. 100506541-0



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

Carrera de Software

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE TITULACIÓN

En mi calidad de Tutor de Trabajo de Grado presentado por el egresado, **Kevin Xavier Vinueza Cuaical** para optar por el título de Ingeniero de Software, cuyo tema es: **Cuantificación automática de malezas utilizando imágenes adquiridas por un dron en campos de cultivos de maíz y/o papa mediante la red neuronal convolucional ResNet**. Considero que el presente trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador.

En la ciudad de Ibarra, a los 20 días del mes de febrero de 2024

PhD. García Santillán Iván Danilo

DIRECTOR DE TRABAJO DE GRADO

DEDICATORIA

Primeramente, quiero dedicar este trabajo a mi madre, ella ha sido un pilar fundamental para lograr este éxito profesional, sin su ayuda esto no hubiera sido posible, le agradezco por todas las veces que siempre estuvo motivándome a no rendirme y seguir adelante, su amor me motivo a nunca desistir en mis metas, cuando se presentaban situaciones difíciles siempre recordaba las palabras de aliento que ella me daba y estas fueron el motor que me permitió alcanzar y ser lo que soy actualmente, gracias por todas tus enseñanzas, eres la mejor mentora que Dios pudo poner en mi camino, te amo mamá.

Agradezco a mi hermana, que también ha sido una parte fundamental en el logro de esta meta profesional, te agradezco por las veces que me has ayudado en situaciones en que los demás me han dado la espalda, por celebrar mis triunfos como si fueran tuyos y también al consolarme en mis derrotas.

Doy gracias al resto de mi familia por haber formado parte de este camino y hacer su contribución para haber alcanzado este sueño.

AGRADECIMIENTO

Agradezco principalmente a mi madre y mi hermana ya que han sido las personas que han hecho su aporte para que mi persona pueda alcanzar esta meta profesional, de igual manera doy gracias a mis demás familiares por haber sido testigos de cada etapa que fui pasando hasta alcanzar este sueño.

Expreso mis más grandes agradecimientos a la Universidad Técnica del Norte por haber sido una cuna la cual me dio las herramientas y conocimientos necesarios para poder crecer profesionalmente y tener un medio por el cual salir adelante en la vida.

Quiero dar las gracias al PhD. García Santillán Iván Danilo y al PhD. Pusdá Chulde Marco Remigio por haber contribuido en el cumplimiento de este trabajo de titulación.

TABLA DE CONTENIDOS

DEDICATORIA	v
AGRADECIMIENTO	vi
TABLA DE CONTENIDOS	vii
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE TABLAS.....	xvi
RESUMEN.....	xx
ABSTRACT	xxii
INTRODUCCIÓN.....	1
Tema	1
Problema	1
Antecedentes.....	1
Situación Actual.....	1
Prospectiva.....	2
Planteamiento del problema.....	2
Objetivos.....	3
Objetivo General.....	3
Objetivos Específicos	3
Alcance.....	3
Metodología.....	7
Justificación	8
Justificación tecnológica.....	9
Justificación agrícola	9
Justificación económica	9
CAPÍTULO 1	10
Marco Teórico	10

1.1	Fundamentación teórica.....	10
1.1.1	Maíz.....	10
1.1.2	Papa.....	11
1.1.3	Malezas.....	12
1.1.4	Métodos tradicionales para el control de malezas.....	15
1.2	Agricultura de precisión.....	16
1.2.1	Dron (UAV).....	17
1.2.2	Identificación de malezas mediante redes neuronales convolucionales.....	17
1.3	Redes neuronales convolucionales.....	18
1.3.1	Arquitectura.....	18
1.3.2	Red neuronal residual (ResNet).....	20
1.3.3	ResNeXt.....	23
1.4	Metodologías.....	25
1.4.1	KDD.....	25
1.4.2	Kanban.....	26
1.5	Normas ISO.....	28
1.5.1	ISO 25010 (Modelo de calidad de sistemas y software).....	28
1.5.2	ISO 25023 (Medición de la calidad del sistema y del producto de software)....	29
1.6	Aplicación web.....	30
1.6.1	Componentes.....	30
1.6.2	Arquitectura.....	31
1.6.3	Lenguaje de programación.....	31
1.6.4	Framework.....	32
1.6.5	Web Service.....	33
1.6.6	API.....	33
1.7	Trabajos relacionados.....	34

End to End Segmentation of Canola Field Images Using Dilated U-Net	34
Method for Segmentation of Bean Crop and Weeds Based on Improved UperNet	35
Semantic Segmentation of Crops and Weeds with Probabilistic Modeling and Uncertainty Quantification	35
Sugar Beets and Weed Detection using Semantic Segmentation	36
DEEP LEARNING BASED OVERCOMPLETE REPRESENTATIONS FOR PADDY RICE CROP AND WEED SEGMENTATION.....	37
DeepVeg: Deep Learning Model for Segmentation of Weed, Canola, and Canola Flea Beetle Damage.....	37
1.8 Preguntas de investigación	38
CAPÍTULO 2	45
Desarrollo.....	45
2.1 Visión general del proyecto	45
2.1.1 Planificación	45
2.1.2 Entregables del proyecto.	46
2.1.3 Materiales y herramientas.....	46
2.2 Recopilación e inclusión de datos.....	48
2.3 Selección, preprocesamiento y transformación de datos.....	58
2.3.1. Anotación del data set.....	69
2.4 Minería de datos.....	74
2.4.1 Definición del algoritmo de Deep Learning.....	74
2.4.2 Entrenamiento	78
2.5 Evaluación e interpretación	90
2.6 Difusión y uso	93
2.6.1 Backend.....	94
2.6.2 Frontend	95
2.6.3 Implementación de contenedores Docker	98

2.6.4	Despliegue.....	99
CAPÍTULO 3		100
Validación de resultados.....		100
3.1	Validación de resultados del entrenamiento	100
3.1.1	Base line	102
3.1.2	Versión 2.....	105
3.1.3	Versión 3.....	110
3.1.4	Versión 4.....	113
3.1.5	Versión 5.....	117
3.1.6	Versión 6.....	121
3.1.7	Evaluación conjunta de las versiones.....	125
3.2	Validación del nivel de portabilidad de la aplicación web.....	131
3.3	Validación de la problemática	133
3.3.1	Prueba t Student enfocada a determinar si hubo una reducción de tiempo utilizando la aplicación web.....	135
3.3.2	Prueba t Student y coeficiente de determinación aplicada a la cuantificación de la maleza lengua de vaca	137
3.3.3	Prueba t Student y coeficiente de determinación aplicada a la cuantificación de la maleza diente de león	141
3.3.4	Prueba t Student y coeficiente de determinación aplicada a la cuantificación de la maleza kikuyo.....	145
3.3.5	Prueba t Student y coeficiente de determinación aplicada a la cuantificación de la categoría otro tipo de maleza.....	149
3.3.6	Prueba t Student y coeficiente de determinación aplicada a la cuantificación del cultivo de papa	153
3.3.7	Análisis final	157
DISCUSIÓN		159
CONCLUSIONES		166
RECOMENDACIONES.....		168

BIBLIOGRAFÍA 170

ÍNDICE DE FIGURAS

Figura 1	Árbol de problema	2
Figura 2	Arquitectura de la aplicación web	5
Figura 3	Proceso para la identificación y cuantificación de malezas empleando la aplicación web.....	6
Figura 4	Diagrama de la Metodología.....	8
Figura 5	Arquitectura de las redes neuronales convolucionales	19
Figura 6	Red residual. Un bloque de construcción	21
Figura 7	Bloque identidad.....	22
Figura 8	Bloque convolucional	22
Figura 9	ResNet 50.....	22
Figura 10	Bloque residual de ResNet y ResNeXt.....	23
Figura 11	Equivalencias de la arquitectura ResNeXt	24
Figura 12	Diagrama de flujo de la metodología KDD	25
Figura 13	Métricas de la ISO 25023 para la evaluación de la subcaracterística capacidad de ser instalado.....	29
Figura 14	Tablero Kanban para la etapa de recopilación e inclusión de imágenes	49
Figura 15	Preparación del dron para que emprenda vuelo	52
Figura 16	Configuración de los parámetros de DroneDeploy 5.7	52
Figura 17	Imagen capturada por el dron en el terreno 1 en la fecha 20/05/2023	54
Figura 18	Imagen capturada por el dron en el terreno 1 en la fecha 28/05/2023	54
Figura 19	Imagen capturada por el dron en el terreno 1 en la fecha 04/06/2023	55
Figura 20	Imagen capturada por el dron en el terreno 2 en la fecha 04/06/2023	55
Figura 21	Imagen capturada por el dron en el terreno 3 en la fecha 04/06/2023	56
Figura 22	Imagen capturada por el dron en el terreno 4 en la fecha 16-12-2023.....	56
Figura 23	Imagen capturada por el dron en el terreno 5 en la fecha 16-12-2023.....	57
Figura 24	Imagen capturada por el dron en el terreno 6 en la fecha 16-12-2023.....	57

Figura 25 Imagen capturada por el dron en el terreno 7 en la fecha 16-12-2023.....	58
Figura 26 Tablero Kanban para la etapa de selección, procesamiento y transformación de los datos.....	58
Figura 27 Extracción de pequeñas imágenes de 250px por 250px utilizando Adobe Photoshop.....	62
Figura 28 Imagen de 128px por 128px lista para ser anotada.....	63
Figura 29 Ejemplos de plantas de papa.....	64
Figura 30 Ejemplos de plantas de lengua de vaca.....	64
Figura 31 Ejemplos de plantas de kikuyo.....	65
Figura 32 Ejemplos de plantas de diente de león.....	65
Figura 33 Sub imagen de 128px por 128px que contiene más de una especie de planta	69
Figura 34 Tablero Kanban para la anotación del data set.....	70
Figura 35 Anotación de imágenes utilizando Roboflow para segmentación semántica	71
Figura 36 Tablero Kanban para la minería de datos	74
Figura 37 Equivalencias de la arquitectura ResNeXt.....	75
Figura 38 Modificación a la arquitectura ResNet la segmentación semántica.....	76
Figura 40 Arquitectura de ResNeXt50 para segmentación semántica	77
Figura 41 Función de activación Mish, ReLU, SoftPlus y Swish	82
Figura 42 Diferentes tipos de learning rate schedule usando cosine annealing.....	83
Figura 43 Función de decaimiento del learning rate usando cosine annealing con disminución del 70% con respecto al valor inicial del ciclo anterior	84
Figura 44 Arquitectura del modelo versión 2	85
Figura 45 Arquitectura del modelo versión 3	87
Figura 46 Tablero Kanban para la evaluación e interpretación de patrones	90
Figura 47 Preprocesamiento y post procesamiento automático de las imágenes para realizar predicciones.....	92
Figura 48 Tablero Kanban para la etapa de difusión y uso.....	93
Figura 49 Arquitectura de la aplicación web	94

Figura 50 Interfaz de usuario inicial	96
Figura 51 Interfaz de usuario para visualizar las imágenes cargadas	97
Figura 52 Pantalla de espera	97
Figura 53 Interfaz para la visualización de los resultados	98
Figura 54 Dice loss del entrenamiento y validación del modelo base line	102
Figura 55 Mean dice coefficient del entrenamiento y validación del modelo base line	102
Figura 56 Mean IOU del entrenamiento y validación del modelo base line	103
Figura 57 Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo base line	105
Figura 58 Dice loss del entrenamiento y validación del modelo versión 2.....	106
Figura 59 Mean dice coefficient del entrenamiento y validación del modelo versión 2	106
Figura 60 Mean IOU del entrenamiento y validación del modelo versión 2.....	107
Figura 61 Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 2.....	109
Figura 62 Dice loss del entrenamiento y validación del modelo versión 3.....	110
Figura 63 Mean dice coefficient del entrenamiento y validación del modelo versión 3	110
Figura 64 Mean IOU del entrenamiento y validación del modelo versión 3.....	111
Figura 65 Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 3.....	113
Figura 66 Dice loss del entrenamiento y validación del modelo versión 4.....	114
Figura 67 Mean dice coefficient del entrenamiento y validación del modelo versión 4	114
Figura 68 Mean IOU del entrenamiento y validación del modelo versión 4.....	115
Figura 69 Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 4	117
Figura 70 Dice loss del entrenamiento y validación del modelo versión 5.....	118
Figura 71 Mean dice coefficient del entrenamiento y validación del modelo versión 5	118
Figura 72 Mean IOU del entrenamiento y validación del modelo versión 5.....	119

Figura 73 Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 5	121
Figura 74 Dice loss del entrenamiento y validación del modelo versión 6	122
Figura 75 Mean dice coefficient del entrenamiento y validación del modelo versión 6	122
Figura 76 Mean IOU del entrenamiento y validación del modelo versión 6	123
Figura 77 Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 6	125
Figura 78 Dice loss de la etapa de validación de todas las versiones	126
Figura 79 Mean dice coefficient de la etapa de validación de todas las versiones.....	127
Figura 80 Mean IOU de la etapa de validación de todas las versiones	129
Figura 81 Gráfica del coeficiente de determinación para la maleza lengua de vaca	141
Figura 82 Gráfica del coeficiente de determinación para la maleza diente de león	145
Figura 83 Gráfica del coeficiente de determinación para la maleza kikuyo	149
Figura 84 Gráfica del coeficiente de correlación para la categoría otro tipo de maleza	153
Figura 85 Gráfica del coeficiente de determinación para el cultivo de papa	157

ÍNDICE DE TABLAS

Tabla 1 Malezas presentes en los cultivos de maíz del Ecuador.....	13
Tabla 2 Malezas presentes en los cultivos de papa del Ecuador.....	14
Tabla 3 Preguntas de investigación propuestas por el PhD. Iván García.....	38
Tabla 4 Planificación para el desarrollo del proyecto	45
Tabla 5 Entregables del proyecto de titulación	46
Tabla 6 Materiales y herramientas utilizados para el desarrollo	47
Tabla 7 Especificaciones técnicas del dron DJI Mavic 2 Pro	47
Tabla 8 Parámetros del dron elegidos para la captura de imágenes.....	50
Tabla 9 Fechas en las que se capturaron las imágenes en los terrenos elegidos	51
Tabla 10 Número de imágenes de 5472px por 3648px capturadas por fechas en los terrenos.....	53
Tabla 11 Número de imágenes de 5472px por 3648px útiles resultantes	59
Tabla 12 Imágenes de 5472px por 3648px utilizables de todos los terrenos	60
Tabla 13 Especies de plantas presentes en las imágenes.....	63
Tabla 14 Aproximación de sub imágenes de 250px por 250px extraídas por cada terreno..	66
Tabla 15 Número de imágenes de 250px por 250px resultantes después de la etapa de clasificación y extracción	68
Tabla 16 Cantidad de imágenes de 128px por 128px en las que se encuentran presentes los tipos de plantas	69
Tabla 17 Valores numéricos de los píxeles correspondientes a cada tipo de planta	71
Tabla 18 Cantidad de individuos por cada tipo de planta y su porcentaje de representación con respecto a los individuos totales	72
Tabla 19 Número de imágenes del conjunto de entrenamiento y validación.....	73
Tabla 20 Tipos de aumentos de datos aplicados	73
Tabla 21 Valores para las cardinalidades de ResNeXt	75
Tabla 22 Propuestas de autores para la definición de los hiperparámetros de entrenamiento	78

Tabla 23 Codificación one-hot para las categorías.....	79
Tabla 24 Características del entorno de entrenamiento	80
Tabla 25 Valores de los hiperparámetros del modelo base line.....	81
Tabla 26 Valores de los hiperparámetros del modelo versión 2	86
Tabla 27 Valores de los hiperparámetros del modelo versión 3	88
Tabla 28 Valores de los hiperparámetros del modelo versión 4	88
Tabla 29 Valores de los hiperparámetros del modelo versión 5	89
Tabla 30 Valores de los hiperparámetros del modelo versión 6	89
Tabla 31 Valores del dice loss, mean dice coefficient y mean IOU del modelo base line en la época 100 en el entrenamiento y validación.....	103
Tabla 32 IOU de cada categoría del modelo base line utilizando el 20% del conjunto de validación	104
Tabla 33 Tiempo de inferencia y de entrenamiento del modelo base line	104
Tabla 34 Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 2 en la época 100 en el entrenamiento y validación.....	107
Tabla 35 IOU de cada categoría del modelo versión 2 utilizando el 20% del conjunto de validación	108
Tabla 36 Tiempo de inferencia y de entrenamiento del modelo versión 2.....	108
Tabla 37 Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 3 en la época 100 en el entrenamiento y validación.....	112
Tabla 38 IOU de cada categoría del modelo versión 3 utilizando el 20% del conjunto de validación	112
Tabla 39 Tiempo de inferencia y de entrenamiento del modelo versión 3.....	112
Tabla 40 Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 4 en la época 100 en el entrenamiento y validación.....	116
Tabla 41 IOU de cada categoría del modelo versión 4 utilizando el 20% del conjunto de validación	116
Tabla 42 Tiempo de inferencia y de entrenamiento del modelo versión 4.....	116
Tabla 43 Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 5 en la época 100 y 200 en el entrenamiento y validación.....	120

Tabla 44 IOU de cada categoría del modelo versión 5 utilizando el 20% del conjunto de validación	120
Tabla 45 Tiempo de inferencia y de entrenamiento del modelo versión 5.....	120
Tabla 46 Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 6 en la época 100 en el entrenamiento y validación.....	124
Tabla 47 IOU de cada categoría del modelo versión 6 utilizando el 20% del conjunto de validación	124
Tabla 48 Tiempo de inferencia y de entrenamiento del modelo versión 6.....	124
Tabla 49 Valores del dice loss de todas las versiones en la época 100 en la fase de validación	126
Tabla 50 Valores del mean dice coefficient de todas las versiones en la época 100 en la fase de validación	128
Tabla 51 Valores del mean IOU de todas las versiones en la época 100 en la fase de validación	129
Tabla 52 IOU de cada clase de todas las versiones del modelo entrenadas utilizando el 20% del conjunto de validación.....	130
Tabla 53 Resultados de la métrica eficiencia en el tiempo de instalación aplicada a 6 desarrolladores	132
Tabla 54 Resultados de la métrica facilidad de instalación aplicada a 6 desarrolladores ...	132
Tabla 55 Tiempo empleado para la identificación y cuantificación de malezas de forma tradicional en 3 terrenos de papa.....	135
Tabla 56 Tiempo empleado para la identificación y cuantificación de malezas utilizando la aplicación web en 3 terrenos de papa	135
Tabla 57 Aplicación de la prueba t Student a los conjuntos TT y TA.....	136
Tabla 58 Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada a la maleza lengua de vaca	137
Tabla 59 Resultados de la prueba t Student con respecto a la cuantificación de la maleza lengua de vaca utilizando el ground truth y la predicción de la aplicación web	139
Tabla 60 Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada a la maleza diente de león	141
Tabla 61 Resultados de la prueba t Student con respecto a la cuantificación de la maleza diente de león utilizando el ground truth y la predicción de la aplicación web.....	143

Tabla 62 Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada a la maleza kikuyo.....	145
Tabla 63 Resultados de la prueba t Student con respecto a la cuantificación de la maleza kikuyo utilizando el ground truth y la predicción de la aplicación web	148
Tabla 64 Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada a la categoría otro tipo de maleza	149
Tabla 65 Resultados de la prueba t Student con respecto a la cuantificación de la categoría otro tipo de maleza utilizando el ground truth y la predicción de la aplicación web.....	152
Tabla 66 Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada al cultivo de papa	154
Tabla 67 Resultados de la prueba t Student con respecto a la cuantificación del cultivo de papa utilizando el ground truth y la predicción de la aplicación web.....	156
Tabla 68 IOU de cada categoría de todas las versiones del modelo entrenadas utilizando el 20% del conjunto de validación	159
Tabla 69 IOU de cada clase reportado por los trabajos de investigación enfocados en 3 clases	160
Tabla 70 IOU de cada clase reportado por el trabajo de investigación enfocado en 4 clases	161
Tabla 71 Comparativa de modelo ResNeXt50 versión 6 y los modelos de los trabajos de investigación utilizando el IOU del background y cultivo	161
Tabla 72 Mean IOU de los trabajos de investigación	164

RESUMEN

La identificación y cuantificación de malezas es un aspecto muy importante en la agricultura, ya que las personas que se dedican a este oficio deben realizar controles de estas malas hierbas con la finalidad de que sus cultivos presenten las más mínimas consecuencias negativas, las malezas se apoderan de los nutrientes, minerales, espacio físico, luz solar, entre otras cosas, lo cual ocasiona una serie de problemas a los cultivos que van desde baja producción hasta bajos niveles nutricionales.

El proceso de identificación y cuantificación de malezas necesita de una alta cantidad de tiempo, el tiempo empleado es directamente proporcional al área del terreno, además influyen otros aspectos, como nivel de conocimiento del agricultor en identificar los tipos de malezas, forma y tipo de superficie del terreno, clima, entre otros, todos estos factores influyen en la cantidad de tiempo empleado para realizar el proceso de identificación y cuantificación de malas hierbas, muchas de las veces no se hace una cuantificación adecuada por los factores antes mencionados, por lo que en la mayoría de ocasiones se omite este proceso y se hace una estimación muy subjetiva, esto conlleva a que las decisiones tomadas para el control de malezas no sean las óptimas ya que los datos no están adecuadamente calculados.

El propósito de este proyecto es la creación de una aplicación web que implemente un algoritmo de deep learning para la identificación y cuantificación de malezas utilizando imágenes adquiridas por un dron, esto con el fin de ayudar a minimizar el tiempo que toma realizar esta actividad, con esto se realizaría un identificación y cuantificación correcta ya no de manera subjetiva sino totalmente objetivo lo cual ayudará a tomar mejores decisiones para el control de las malas hierbas.

En el capítulo 1 se explica cuáles son las diferencias entre cultivos y malezas, que tipos de malas hierbas existen, cuáles son las propuestas de deep learning para la identificación y cuantificación, que arquitecturas de los algoritmos se están utilizando, cuáles son los mayores

restos en esta áreas, que metodologías se usan para este tipo de proyecto, se analizaran trabajos de investigación relacionados para poder tener una perspectiva más amplia, además servirán de base para poder comparar la propuesta de este trabajo, para finalizar se contestaran preguntas de investigación planteadas por el director de este trabajo.

En el capítulo 2 se detalla como fue el proceso llevado a cabo para la adquisición de los datos y la elección del algoritmo de deep learning, al igual que las distintas versiones propuestas para mejorar los resultados de predicción del algoritmo, también se explicará cómo se creó la aplicación web y que metodologías se utilizaron para todo el proceso.

En el capítulo 3 se hace mucho énfasis en analizar los resultados obtenidos en los entrenamientos de todas las versiones propuestas hasta la elección del mejor modelo para implementarlo en la aplicación web, se evaluará el nivel de portabilidad de la solución tecnológica utilizando las métricas de la ISO 25023 enfocadas a la capacidad de ser instalado, finalmente se utilizará el método estadístico t Student para comprobar que la aplicación web ayuda a disminuir la cantidad de tiempo en realizar la identificación y cuantificación de malezas, y además de comprobar el nivel de confiabilidad.

Finalmente, en la discusión se analiza los resultados obtenidos comparándolos con los presentados en otras propuestas de investigación para poder determinar el nivel de semejanza que existe entre estos.

ABSTRACT

The identification and quantification of weeds is a very important aspect in agriculture, since the people who are dedicated to this profession must carry out controls of these weeds in order that their crops present the slightest negative consequences, weeds take over of nutrients, minerals, physical space, sunlight, among other things, which causes a series of problems for crops ranging from low production to low nutritional levels.

The process of identifying and quantifying weeds requires a high amount of time, the time spent is directly proportional to the area of the land, other aspects also influence, such as the farmer's level of knowledge in identifying the types of weeds, shape and type of surface. of the terrain, climate, among others, all these factors influence the amount of time used to carry out the process of identification and quantification of weeds, many times an adequate quantification is not made due to the aforementioned factors, so in Most of the time this process is omitted and a very subjective estimate is made, this means that the decisions made for weed control are not optimal since the data are not adequately calculated.

The purpose of this project is the creation of a web application that implements a deep learning algorithm for the identification and quantification of weeds using images acquired by a drone, in order to help minimize the time it takes to carry out this activity, with This would be a correct identification and quantification, no longer subjective but totally objective, which will help make better decisions for weed control.

Chapter 1 explains what the differences are between crops and weeds, what types of weeds exist, what are the deep learning proposals for identification and quantification, what algorithm architectures are being used, what are the largest remains in These areas, what methodologies are used for this type of project, related research works will be analyzed in order to have a broader perspective, they will also serve as a basis to compare the proposal of this work, and finally, research questions posed by the project will be answered. director of this work.

Chapter 2 details how the process was carried out to acquire the data and choose the deep learning algorithm, as well as the different versions proposed to improve the prediction results of the algorithm. It will also explain how it was created. the web application and what methodologies were used for the entire process.

In chapter 3, much emphasis is placed on analyzing the results obtained in the training of all the proposed versions until the choice of the best model to implement it in the web application, the level of portability of the technological solution will be evaluated using the ISO metrics. 25023 focused on the ability to be installed, finally the Student t statistical method will be used to verify that the web application helps reduce the amount of time to carry out the identification and quantification of weeds, and in addition to verify the level of reliability.

Finally, in the discussion the results obtained are analyzed, comparing them with those presented in other research proposals in order to determine the level of similarity that exists between these.

INTRODUCCIÓN

Tema

Cuantificación automática de malezas utilizando imágenes adquiridas por un dron en campos de cultivos de maíz y/o papa mediante la red neuronal convolucional ResNet.

Problema

Antecedentes

En el área de la agricultura existen muchos problemas que imposibilitan el desarrollo de actividades agrícolas de manera eficiente, el más común es la presencia de malezas, o malas hierbas como se les conoce en algunas regiones. Las malezas son plantas invasoras que afectan negativamente a los cultivos, entre sus principales características está el apoderamiento de espacio geográfico del terreno y recursos tales como minerales, agua y luz solar, además cumplen la función de ser hospedadoras de plagas. Las consecuencias que generan estas malas hierbas van desde representar grandes pérdidas para los agricultores y empresas que se dedican a este tipo de actividad hasta presentar serios problemas alimenticios no solo para las personas que consumen los productos sino también para los animales ya que los alimentos cosechados tienen mayor probabilidad de presentar deficiencias nutricionales e incluso ser portadores de enfermedades propagadas por las plagas presentes en los cultivos. Este problema se presenta en todos los cultivos del mundo y la granja La Pradera de la UTN no es la excepción. Los expertos en malezas necesitan determinar la densidad poblacional de estas plantas para implementar medidas oportunas al respecto generando el menor grado de impacto negativo en el medio ambiente.

Situación Actual

En la granja La Pradera de la UTN se hace un cálculo manual de la densidad poblacional de las malezas, este procedimiento demanda una gran cantidad de tiempo debido a la extensión

de las áreas de cultivo y en la mayoría de las ocasiones se hace un cálculo subjetivo, por lo que los datos no son precisos y la toma de acciones no es oportuna.

Prospectiva

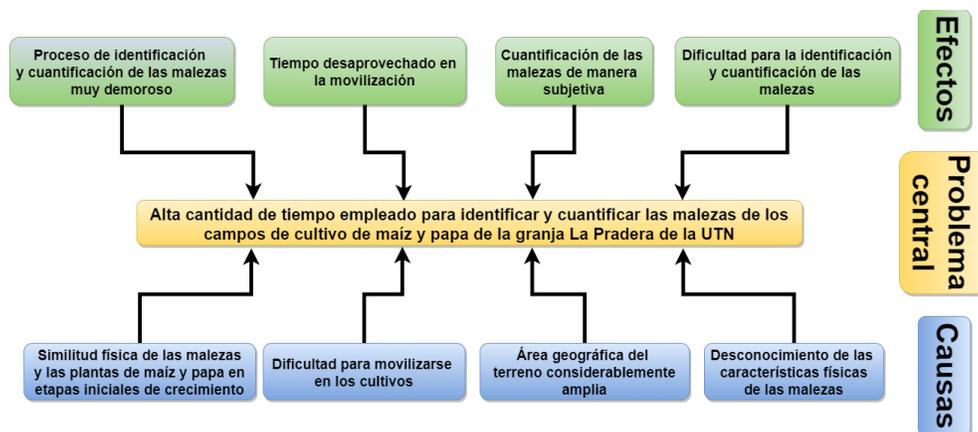
El presente trabajo tiene por finalidad reducir el tiempo que se emplea para calcular la densidad poblacional de las malezas en los cultivos de maíz y/o papa de la granja La Pradera de la UTN, además los resultados serán objetivos, por lo que la toma de decisiones para el control de malezas será oportuna, evitando los gastos innecesarios y reduciendo el impacto ambiental.

Planteamiento del problema

En la granja La Pradera de la UTN el cálculo de la densidad poblacional de las malezas en los cultivos de maíz y papa se lo realiza de manera manual, esto conlleva a que en la mayoría de las ocasiones se lo haga de manera subjetiva debido a la gran cantidad de tiempo que esta actividad demanda, esto repercute en la toma ineficiente de decisiones y una mayor posibilidad de aumentar el grado de impactos negativos en el medio ambiente. El presente trabajo propone una solución utilizando imágenes adquiridas por un dron en los cultivos de maíz y/o papa y empleando la red neuronal convolucional ResNet. En la figura 1 se propone el árbol de problemas planteado para el proyecto.

Figura 1

Árbol de problema



Nota. Elaboración propia.

Objetivos

Objetivo General

Cuantificar automáticamente malezas utilizando imágenes adquiridas por un dron en campos de cultivos de maíz y/o papa mediante la red neuronal convolucional ResNet.

Objetivos Específicos

- Elaborar un marco teórico con respecto a la detección de malezas en cultivos aplicando redes neuronales convolucionales.
- Desarrollar una aplicación web que utilice la arquitectura de ResNet CNN para identificar y cuantificar las malezas en los cultivos de maíz y/o papa cumpliendo la subcaracterística de capacidad de ser instalado inmersa en la característica de portabilidad de la norma ISO 25010.
- Validar los resultados de la propuesta utilizando métricas de inteligencia artificial, estadística descriptiva y métricas establecidas por la norma ISO 25023 para la evaluación de la subcaracterística de capacidad de ser instalado.

Alcance

Este trabajo tiene por finalidad desarrollar una aplicación web que implemente el algoritmo de ResNet CNN para identificar y cuantificar las malezas presentes en campos de cultivos de maíz y/o papa utilizando imágenes adquiridas por un dron. Se aplicará la metodología KDD para el desarrollo del proyecto (Fayyad et al., 1996).

Se desarrollará un marco teórico referencial con respecto a la aplicación de las redes neuronales convolucionales en el contexto de la agricultura para la detección de malezas, para esto se va a utilizar bases de datos bibliográficas proporcionadas por la Universidad Técnica del Norte, además, también se van a responder las siguientes preguntas de investigación planteadas por el director del proyecto de tesis el PhD. Iván Danilo García Santillán. Las preguntas se detallan a continuación:

- ¿Qué drones y cámaras están usando para la tarea de detección automática de malas hierbas?
- ¿Cuál es la altura y velocidad adecuada del dron para capturar imágenes y regiones de interés (ROIs) de alta resolución (px/cm)?
- ¿Qué espacio de color y formato de imágenes resulta adecuado?
- ¿En qué bandas del espectro electromagnético (visible, infrarrojo, ultravioleta) están trabajando?
- ¿Qué cultivos y tipos de malezas se están detectando y dónde?
- ¿Qué conjunto de datos se encuentran disponibles públicamente para el entrenamiento de un modelo de Deep Learning?
- ¿Cómo crear y anotar un conjunto de datos propio?
- ¿Qué arquitectura de CNN se usan para la detección automática de malas hierbas?
- ¿Qué hardware específico o servicios en la nube resulta adecuado para entrenar modelos de DL?
- ¿Qué métricas y gráficas se usan para validar el rendimiento de los modelos de DL?
- ¿Cómo afinar un modelo de DL para mejorar el rendimiento?

Se elaborará un repositorio de imágenes el cuál contendrá un mínimo de 50 imágenes de cultivos de maíz y/o papa con presencia de malezas, las imágenes van a ser adquiridas con un dron en los cultivos de maíz y/o papa en la granja La Pradera de la Universidad Técnica del Norte. La anotación de las imágenes se lo realizará manualmente utilizando una herramienta de software libre.

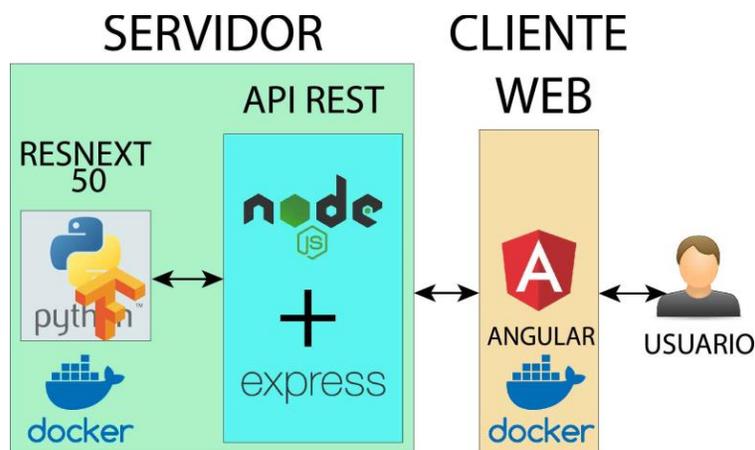
El algoritmo de aprendizaje profundo a utilizar será la ResNet CNN, el lenguaje de programación para el desarrollo de esta red será Python utilizando las librerías de TensorFlow y

Keras, este algoritmo será entrenado utilizando el repositorio previamente elaborado, depurado y anotado. El entorno establecido para el entrenamiento es Google Colaboratory en su versión de paga. Se utilizará la técnica de Grid Search para la etapa de entrenamiento (Syarif et al., 2016).

La arquitectura de la aplicación web será el Cliente-Servidor, para el apartado del Back-End se utilizará NodeJs y Python y la parte del Front-End se desarrollará con el Framework de Angular, además se implementará la tecnología de contenedores Docker. Toda la solución web será desplegada en un servidor cloud. En la figura 2 se representa la arquitectura de la propuesta.

Figura 2

Arquitectura de la aplicación web



Nota. Elaboración Propia.

Debido al alcance y objetivo para el cual se va a desarrollar la aplicación web no se va a hacer uso de ninguna metodología de desarrollo ágil ni tradicional, sin embargo, se va a implementar algunos principios del manifiesto ágil, entre los cuales, Scrum Manager (2021) proporciona la siguiente definición para 4 de ellos:

- Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
- Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.

- El software que funciona es la principal medida del progreso.
- Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.

No se va a optar por ningún gestor de base de datos debido a que la aplicación web está enfocada a que realice la identificación y cuantificación de las malezas al momento que se le proporcione las imágenes de los cultivos y no guardará registros de los resultados ya que no son necesarios para el cumplimiento del objetivo general.

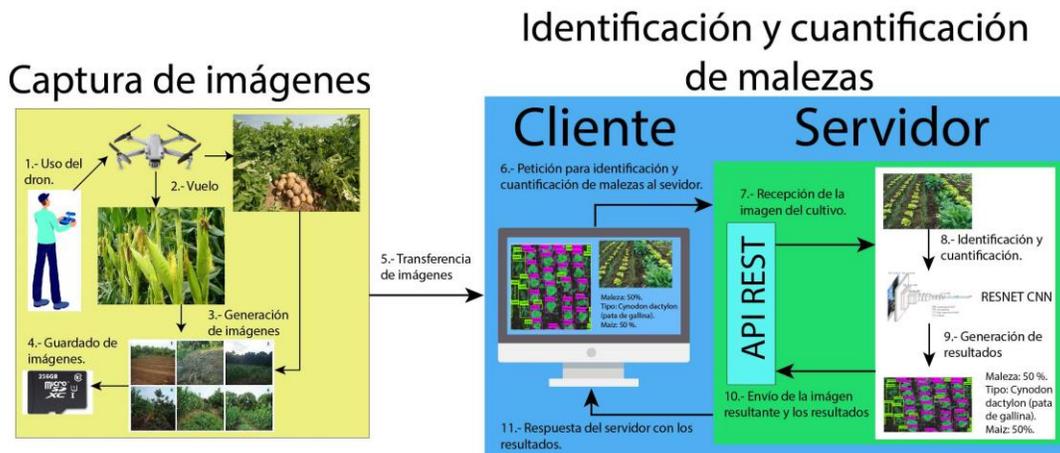
Se aplicará la subcaracterística de capacidad de ser instalado inmersa en la característica de portabilidad de la norma ISO 25010 (ISO 25000, 2022).

Los resultados serán validados utilizando métricas de inteligencia artificial, uso de métodos de estadística descriptiva y métricas establecidas por la norma ISO 25023, las cuales son usadas para la evaluación de la subcaracterística de capacidad de ser instalado, la cual forma parte de la característica de portabilidad de la norma ISO 25010.

En la figura 3 se representa el proceso para la identificación y cuantificación de las malezas mediante la aplicación web.

Figura 3

Proceso para la identificación y cuantificación de malezas empleando la aplicación web



Nota. Elaboración propia.

Metodología

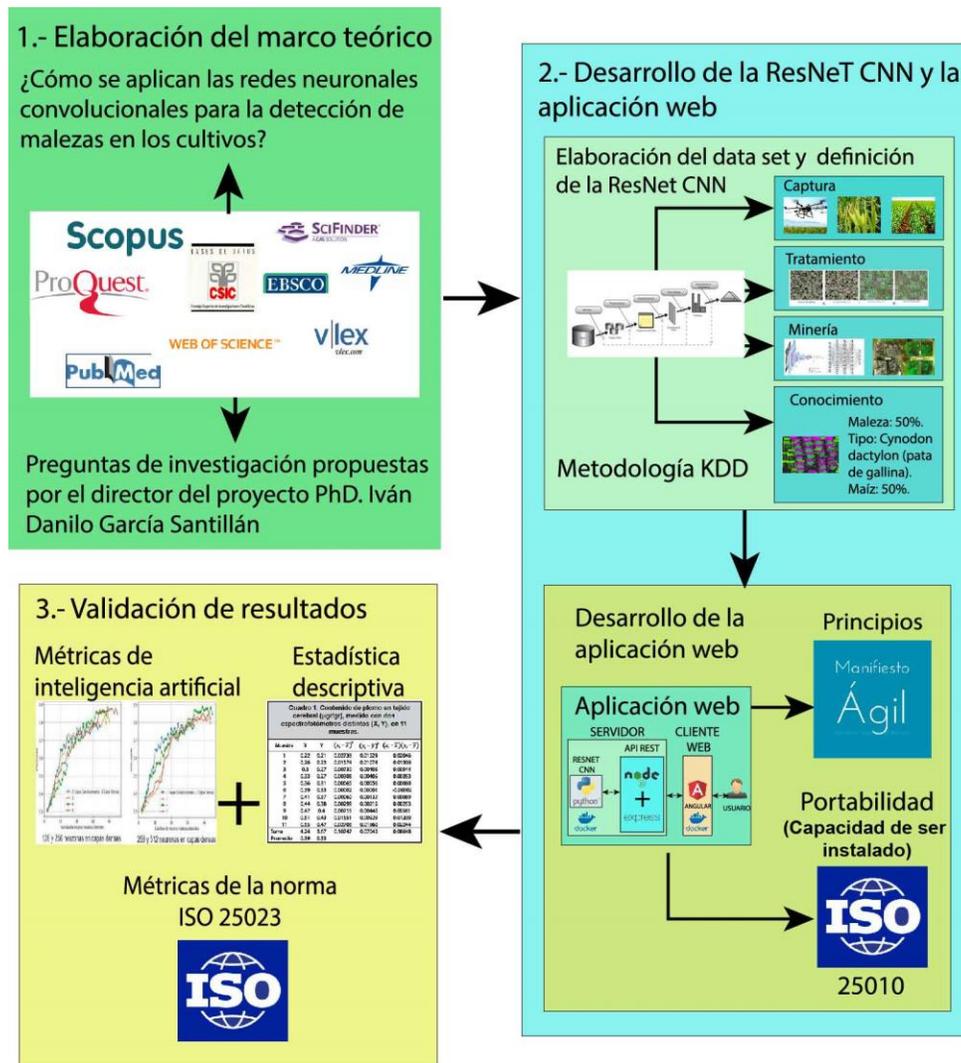
Para cumplir el primer objetivo se va a hacer una recopilación de información de artículos científicos, trabajos de titulación, libros, etc. de las bases bibliográficas proporcionadas por la Universidad Técnica del Norte, la información recopilada debe responder a la pregunta ¿Cómo se aplican las redes neuronales convolucionales para la detección de malezas en los cultivos?, además se deben responder también a las preguntas propuestas por el PhD. Iván Danilo García Santillán.

Para cumplir el segundo objetivo se va a emplear la metodología KDD (Fayyad et al., 1996), esta ayudará a hacer un seguimiento y control del proceso de recolección de los datos, en este caso las imágenes, su depuración, limpieza, anotación, segmentación en datos de entrenamiento, testeo y validación, dentro de la metodología está incluido el entrenamiento de la red neuronal convolucional. Para el apartado del desarrollo de la aplicación web se va a utilizar algunos principios de manifiesto ágil y el modelo arquitectural Cliente-Servidor (Lizama et al., 2016), además se va a cumplir con la subcaracterística de capacidad de ser instalado inmersa en la característica de portabilidad de la norma ISO 25010 (ISO 25000, 2022).

Para cumplir el tercer objetivo se va a emplear métricas de inteligencia artificial y métodos de estadística descriptiva para la validación de los resultados con respecto a la etapa de entrenamiento de la red neuronal convolucional ResNet, así como su confiabilidad de predicción cuando sea implementada en la aplicación web, esto se lo va a realizar juntamente con el director del proyecto y el especialista en malezas. Además, se van a emplear las métricas establecidas por la norma ISO 25023 para la evaluación de la subcaracterística de capacidad de ser instalado de la aplicación web. En la figura 4 se representa la metodología para elaborar el proyecto.

Figura 4

Diagrama de la Metodología



Nota. Elaboración propia.

Justificación

Este trabajo pretende contribuir con el Objetivo de Desarrollo Sostenible N12 “Producción y consumo responsables” específicamente a la meta 12.2 que trata de deducir la cantidad de pérdida de productos durante proceso de siembra (Nation United, 2017).

Justificación tecnológica

La Problemática 7 de las Políticas Agropecuarias para el año 2025 se menciona que existe una baja innovación tecnológica con respecto al área de productividad nacional, debido a esto en el lineamiento 7.4 del Ministerio de Agricultura y Ganadería se determina que se deben desarrollar investigaciones que ayuden a mejorar los sistemas de producción, entre los cuales se encuentran los agro-productivos (Senplades, 2017a).

Justificación agrícola

La Organización de las Naciones Unidas (2017), predice que para el año 2050 va a existir un aumento en la demanda de un 70% de alimentos, lo que supone que el nivel de producción en el ámbito de la agricultura debe aumentar para poder abastecer de alimentos a la población, es por esta razón que se debe hacer uso de la tecnología para mejorar los procesos de la agricultura y aumentar la calidad de los productos.

La agricultura es una actividad que representa el 8% para el Producto Interno Bruto del Ecuador, es por esta razón que se debe hacer mucho énfasis en ayudar a mejorar el rendimiento de producción de esta actividad, esto generaría una sostenibilidad alimenticia a largo plazo (Ministerio de Agricultura y Ganadería, 2019).

Justificación económica

En el artículo 385, literal 3 de la constitución de la república (2008) se establece que se debe fortalecer las actividades de producción nacional haciendo uso de tecnología a innovaciones que permitan mejorar la eficacia y productividad nacional. El Plan Nacional de Desarrollo 2017-2021 en su política 5.6 establece que se debe promover la investigación y desarrollo tecnológico para impulsar un cambio productivo a nivel nacional en el sector público, productivo y en las universidades. (Senplades, 2017b).

CAPÍTULO 1

Marco Teórico

1.1 Fundamentación teórica

1.1.1 Maíz

El maíz, por su nombre científico *Zea mays* L., forma parte del árbol genealógico de las gramíneas, se cree que su ancestro más cercano fue el 'teocintle' debido a su alto grado de semejanza (Zambrano et al., 2021). Estudios mencionan que la domesticación de este cereal se remonta a más de 6 mil años, en regiones pertenecientes al actual país de México (Méndez Barceló, 2016).

Morfología

Alcanza una altura aproximada de 2.50 metros, se caracteriza por la presencia de una flor de género masculino en la zona más alta (Deras, 2020), cuenta con hojas lanceoladas acopladas en los extremos del tallo, una raíz fibrosa y las mazorcas emergen a partir de las flores femeninas localizadas encima de las hojas (Méndez Barceló, 2016).

- **Raíz:** En la etapa de germinación brota una raíz primaria, posteriormente de la corona del tallo emerge una raíz secundaria la cual puede llegar a tener una longitud aproximada de 1.80 metros (Estrada, 2020).
- **Tallo:** Es el soporte de la planta, de este emergen las hojas y las flores, su composición interna es carnosa y presenta una gran cantidad de agua (Guzmán, 2017).
- **Hoja:** Alcanza una longitud promedio de 1 metro, se ubica en la parte lateral del tallo y está formada por la vaina, el plano foliar y el cuello (Badillo, 2016).
- **Inflorescencia:** Presenta inflorescencia masculina y femenina, es una espiga localizada en el ápice del tallo, contiene 2 flores las cuales se encuentran en el

interior de 2 glumas, estas últimas son preservadas por los lemas (Izquierdo, 2012).

- **Grano:** Constituido por una capa externa, en su interior hay proteínas y sustancias que le dan el color que puede ser amarillo o blanco, el endosperma abarca el 85% al 90% del grano (Rodríguez, 2020).

1.1.2 *Papa*

Es un tubérculo comestible, su nombre científico es *Solanum tuberosum* y pertenece a la familia de las solanáceas (Córdor, 2018). Los registros sugieren que las primeras evidencias de papas silvestres se remontan hace 13 mil años, cerca de las costas de lo que hoy es el actual país de Argentina, las primeras manifestaciones de domesticación de esta planta se ubican en las zonas altas de la cordillera de los Andes y en el lago Titicaca, ubicada al norte de Bolivia (Alonso Arce, 2008).

Morfología

Bautista (2015) define las partes que conforman la papa como se menciona a continuación:

- **Tallo:** Emerge del tubérculo madre, puede llegar a alcanzar una altura máxima de 1 metro, en las etapas iniciales es una parte herbácea, sin embargo, puede llegar a presentar características leñosas en la parte inferior.
- **Estolón:** Es un tallo secundario que emerge alrededor del tallo principal.
- **Hoja:** Cuando existe una ramificación del tallo principal pueden llegar a emerger más de 9 hojas, su tamaño depende de la distancia que existe entre la hoja y el tallo principal.
- **Flor:** Su color característico puede variar desde una tonalidad blanca hasta una morada, la cantidad de flores presentes depende del tipo de papa y de las condiciones climáticas y calidad del suelo.

- **Fruto:** Tiene forma redonda, presenta un color que puede variar desde del verde hasta el amarillo, aunque algunos especímenes llegan a presentar tonalidades violetas, su diámetro promedio es de 3 centímetros, puede albergar semillas como no albergar en casos raros, la cantidad máxima puede ser hasta más de 300 semillas.
- **Raíz:** Se ubica en medio del tubérculo madre y el suelo. Sus características físicas más destacables es que es larga, fina y ramificada. Puede alcanzar una profundidad máxima de 0,50 metros y en casos poco comunes hasta 1 metro.
- **Tubérculo:** Forma parte del tallo, cuando se concentran cantidades de reservas minerales, agua, proteínas, grasa e hidratos de carbono nace el tubérculo. Es la parte de interés, lo que se conoce como papa o patata.

1.1.3 Malezas

Son plantas invasoras que afectan negativamente a los cultivos. Algunos de los impactos más importantes son: toma de los recursos; las plantas necesitan agua, nutrientes del suelo y luz solar para poder alcanzar un óptimo desarrollo, las malezas se apoderan de gran parte de dichos recursos, lo que repercute que los cultivos tengan dificultades para desarrollarse (Babaei-Ghaghelestany et al., 2022).

Las malas hierbas pueden adaptarse fácilmente a casi cualquier ecosistema y no necesitan de muchas condiciones favorables para poder sobrevivir, los cultivos dependen de condiciones específicas para poder desarrollarse, por ejemplo: la distancia de separación de cada planta, la longitud dependerá del tipo de producto sembrado, esto significa que las malezas ocuparán los espacios disponibles entre cada planta y por lo tanto el rendimiento del cultivo se verá afectado (Ringselle et al., 2023).

Malezas de los cultivos de maíz en el Ecuador

Las malezas de los cultivos de maíz en el Ecuador se detallan en la tabla 1.

Tabla 1

Malezas presentes en los cultivos de maíz del Ecuador

Nombre científico	Nombre vulgar	Tipo de hoja
Galinsopa spp.	Abuelaquihua.	Ancha.
Spergula arvensis.	Alfarillo.	Ancha.
Escleranthus anus.	Alpatezera.	Ancha.
Lupinus pubescens.	Ashpachocho.	Ancha.
Chenopodium spp.	Ashpaquínoa.	Ancha.
Amaranthus spp.	Bledo.	Ancha.
Silene gallica.	Forastera.	Ancha.
Veronica pérsica.	Golondrina.	Ancha.
Plantago lanceolata	Llantén.	Ancha.
Rumex crispus.	Pacta.	Ancha.
Bidens humilis.	Pacunga.	Ancha.
Stellaria media.	Pajarera.	Ancha.
Ipomea spp.	Porotillo.	Ancha.
Raphanus raphanistrum.	Rábano.	Ancha.
Medicago hispida.	Trebol carretón.	Ancha.
Lepidium bipinnatifidum.	Tzetzera	Ancha.
Brassica campestris.	Nabo.	Ancha.
Cynodon dactylon.	Cabrestillo.	Delgada.
Cyperus spp.	Coquito.	Delgada.
Poa annua.	Poa.	Delgada.
Grammineace.	Saraqihua.	Delgada.

Pennisetum clandestinum.	Kikuyo.	Delgada.
--------------------------	---------	----------

Nota. (Gabela & Cárdenas, 1979).

Malezas de los cultivos de papa en el Ecuador

Las malezas de los cultivos de papa en el Ecuador se detallan en la tabla 2.

Tabla 2

Malezas presentes en los cultivos de papa del Ecuador

Nombre científico	Nombre vulgar	Tipo de hoja
Spergula arvensis L.	Alfarillo.	Ancha.
Scleranthus annuus L.	Alpatereza.	Ancha.
Amaranthus spp.	Bledo.	Ancha.
Polygonum aviculare L.	Cien nudos.	Ancha.
Polygonum nepalense.	Corazón herido (oreja del diablo).	Ancha.
Polygonum segesum H.B.K.	Duraznillo.	Ancha.
Silene gallica L.	Forastera.	Ancha.
Galinsoga ciliata (Raf.) Blade.	Hierba de cuy.	Ancha.
Gflinsonga parviflora Cav.	Sin especificar.	Ancha.
Plansago lanceolara L.	Llantén.	Ancha.
Malvastrum peruvianum (L.) Gray.	Malva blanca.	Ancha.
Malva silvestris L.	Malva morada.	Ancha.
Sinapsis nigra L.	Mostaza.	Ancha.
Brassica napus L.	Nabo.	Ancha.
Brassica campestris L.	Sin especificar.	Ancha.
Rumex crispus L.	Pacta.	Ancha.
Rumex obtusifolius L.	Sin especificar.	Ancha.
Rumex acerosella L.	Pactilia.	Ancha.

Stellaria media (L.) Cyrill.	Pajarera.	Ancha.
Lepidium chinchicara.	Quimbilla.	Ancha.
Raphanus raphanistrum L.	Rábano.	Ancha.
Taraxacum officinale Weber in Wiggers.	Taraxaco.	Ancha.
Lepidium bipinnatifidum Desv.	Tze-Tzera.	Ancha.
Veronica persica Por.	Verónica.	Ancha.
Gramínea (sin identificar).	Cabrestillo.	Delgada.
Gramínea (sin identificar).	Gramma.	Delgada.
Pennisetum clandestinum Hochst ex Chiov.	Kikuyo.	Delgada.
Gramínea (sin identificar).	Saraqihua.	Delgada.

Nota. (Cárdenas, 1987).

1.1.4 Métodos tradicionales para el control de malezas

Control cultural

Se basa en técnicas y conocimientos que han sido transmitidos de generación en generación, algunas son: correcta labranza del terreno antes de la siembra, uso de semillas mejoradas, elección de terrenos que se encuentren a una altura determinada con presencia óptima de niveles en nutrientes, separación adecuada entre cada planta, entre otras (H. Ullah et al., 2023).

Control manual

Es denominada deshierbe en algunas partes del mundo, es una práctica que consiste en extraer las malezas utilizando herramientas como el azadón, este tipo de control tiene una gran desventaja ya que implica un gran número de personas para realizar esta actividad, la dificultad aumenta proporcionalmente con el área del terreno. Como ventaja principal se puede mencionar

que la erradicación de malezas es casi completa, además, que no tiene impactos negativos al medio ambiente ya que no se utilizan químicos (Alagbo et al., 2022).

Control mecánico

Consiste en utilizar maquinaria, comúnmente se usan tractores que tienen acopado en la parte inferior cuchillas que se encargan de la extracción de las malezas, este tipo de control puede ser perjudicial si no se lo ejecuta correctamente ya que el peso del tractor o el vehículo utilizado puede comprimir el suelo (Deras, 2020).

Una desventaja crucial es que no es aplicable a todos los terrenos, en superficies que no presentan uniformidad se debe optar por otros mecanismos de control, su principal ventaja es la rápida extracción y erradicación de malezas en comparación si se lo hiciera manualmente (Fishkis & Koch, 2023).

Control químico

Tiene ventajas atractivas frente a otros métodos de control, elimina de manera más efectiva las malezas debido a que actualmente existen herbicidas especializados para cada tipo (Antonopoulos et al., 2023). Sin embargo, también presenta desventajas, la más significativa es que el uso excesivo y descontrolado también puede afectar al cultivo y provocar que el terreno quede infértil, la aplicación de este control debe hacerse con el asesoramiento de una persona experta para minimizar los impactos negativos (Paušič et al., 2021).

1.2 Agricultura de precisión

Definida como el conjunto de técnicas y herramientas automatizadas para el mejoramiento de la producción agrícola (Aquilani et al., 2022), esto se logra mediante un análisis y toma de decisiones con respecto a los datos obtenidos mediante sensores, lo que permite hacer un uso eficiente de los recursos disponibles como los fungicidas, el agua, semillas, abonos, entre otros (Cisternas et al., 2020).

1.2.1 Dron (UAV)

Se les denomina a los vehículos aéreos no tripulados, el control se lo hace vía remota en bandas de 2.4 GHz y/o 5.8 GHz (Morales et al., 2020). El concepto no es tan reciente ya que Estados Unidos desarrollo esta tecnología en el año 2005 con fines militares (Laghari et al., 2023).

Su nivel de dificultad para el manejo es bajo comparado a otros vehículos aéreos como un helicóptero, razón por la cual está siendo ampliamente utilizado no solo en el ámbito militar sino también en el civil (N. A. Khan et al., 2020), por ejemplo: agricultura de precisión, trabajos fotográficos, búsqueda y rescate, manejo de desastres, entretenimiento, entre otras (Laghari et al., 2023).

Componentes

Los dispositivos más comunes que se pueden acoplar en la mayoría de los modelos de drones son: cámaras, módulos de comunicación, acoplamiento para el transporte de objetos, sensores, etc. Entre los principales sensores se destacan los siguientes: RGB (Red, Green, Blue), multiespectral, hiperespectral, LIDAR (Light Detection and Ranging), infrarrojo térmico (Naveed & Qazi, 2020).

1.2.2 Identificación de malezas mediante redes neuronales convolucionales

Es un área que constantemente está sometiéndose a pruebas y mejoras, un método muy utilizado es la transferencia de aprendizaje, este se fundamenta en que se puede utilizar una red preentrenada y volverla a entrenar con el conjunto de datos de interés, de esta manera se puede obtener una precisión más alta, este enfoque se usa cuando el conjunto de datos es pequeño (Espejo-Garcia, Mylonas, Athanasakos, Fountas, et al., 2020).

Cuantificación de malezas

La cuantificación de malezas es un aspecto muy importante a la hora de tomar decisiones en lo que respecta a su control, permite determinar cuál es el porcentaje representativo con

respecto al área del terreno de interés, es un indicativo que ayuda a determinar el momento más idóneo para realizar el control de malezas y si se utiliza un control químico saber cuál es la dosis adecuada a utilizar (Puerto, 2018).

Debido a que es una tarea que demanda una cantidad de tiempo considerable y es directamente proporcional con la extensión geográfica del terreno, realizar una correcta cuantificación no siempre es viable y por ende se opta por realizarla de manera subjetiva, esto evidentemente no es método preciso y confiable (Osorio Delgado, 2021).

La cuantificación a través de imágenes es un método que mantiene la objetividad y gracias a las redes neuronales convolucionales es una solución más óptima en comparación a realizarlo de manera manual (Puerto, 2018), una de las fórmulas utilizadas es:

$$cobertura = \frac{\text{Área de maleza}}{\text{Área total}} \times 100\%$$

1.3 Redes neuronales convolucionales

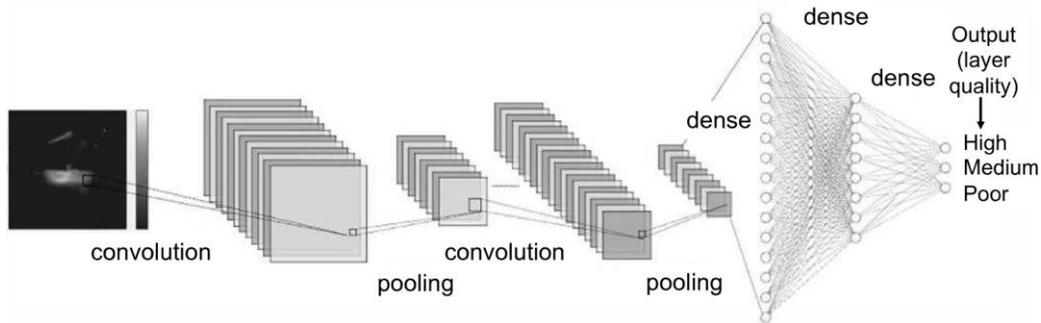
Son algoritmos de Deep Learning comúnmente utilizados en el análisis de imágenes, su cualidad principal es la extracción automática de características a diferencia de los algoritmos de Machine Learning que necesitan como parámetros de entrada dichas características que deben ser determinadas por el analista de datos (Valizadeh & Wolff, 2022), es por esta razón que las redes neuronales convolucionales han tomado gran importancia en los últimos años, además, con el avance del desarrollo de hardware más robusto se ha intensificado en gran medida el uso de estos algoritmos (Coulibaly et al., 2022).

1.3.1 Arquitectura

Las redes neuronales convolucionales se caracterizan por tener 3 tipos de capas, las cuales son: capas de convolución, capas de agrupación y capas completamente conectadas (Coulibaly et al., 2022). En la figura 5 se representa la arquitectura de las redes neuronales convolucionales.

Figura 5

Arquitectura de las redes neuronales convolucionales



Nota. (Valizadeh & Wolff, 2022).

Capas de convolución

Estas capas son las encargadas de la extracción de características de la imagen de entrada, para esto se hace uso de filtros o kernels y al final da como resultado un mapa de características (Valizadeh & Wolff, 2022), el mapa de características puede ser expresado con la siguiente ecuación:

$$Y_k = f(W_k * x)$$

Donde:

- Y_k = Es el mapa de características que servirá de entrada para la capa de agrupación posterior.
- f = Función de activación no lineal.
- W_k = Filtro o kernel.
- x = Imagen de entrada.

Capas de agrupación

Su entrada es el mapa de características, las dimensiones del mapa son reducidos a media que se avanza en la red neuronal convolucional con la finalidad de dar más importancia a las características más relevantes y disminuir el costo computacional, esto se logra mediante los strides que son saltos que se hace en el mapa de características, los cuales deben ser definidos

de acuerdo a las necesidades del problema, los tipos de operaciones más comunes son: agrupación máxima y agrupación promedio (Habib & Qureshi, 2022).

Capas completamente conectadas

Se refiere al perceptrón multicapa, recibe como entrada las características más importantes extraídas en las capas de convolución en formato de un vector, es decir que cada característica es representada por una neurona de entrada, dicha red se encarga de realizar los procesos necesarios para que obtener el resultado que se espera (Habib & Qureshi, 2022). A rasgos generales esta red está conformada por una capa de entrada, capas ocultas y una capa de salida (Habib & Qureshi, 2022).

1.3.2 Red neuronal residual (ResNet)

ResNet es la abreviatura de Residual Network, es un tipo de arquitectura que pertenece a las redes neuronales convolucionales (He et al., 2016).

Las redes neuronales convolucionales tienen un problema denominado desvanecimiento del gradiente cuando las redes son muy profundas (Y. Li et al., 2022). El principal inconveniente es que a medida que el algoritmo de propagación hacia atrás avanza a través de las capas en redes muy profundas el gradiente tienen a ser muy pequeño, lo que significa que los parámetros no pueden ser ajustados correctamente y esto conlleva a que la CNN en cierto momento deje de aprender y se estanque e incluso en algunos casos cometa más errores en las predicciones (K. Zhang et al., 2023).

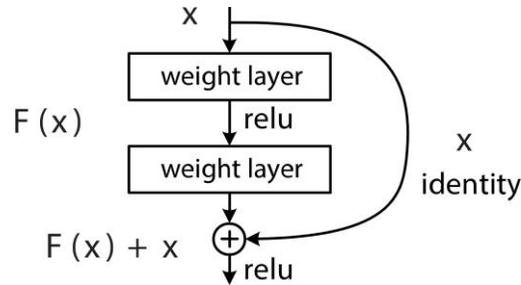
ResNet soluciona ese problema implementando el concepto de conexiones directas, estas conexiones ayudan a que los mapas de características puedan avanzar directamente sin sufrir modificaciones en el proceso, esto ayuda a mantener información importante y que el gradiente no se haga muy pequeño, evidentemente llega un punto en el que la conexión directa se une con el resultado de una capa de convolución, con esta característica una red basada en la arquitectura ResNet tienen mejores resultados mientras más profunda sea (He et al., 2016).

Arquitectura

ResNet propone la segmentación de la red en bloques residuales, a continuación, en la figura 6 se muestra la idea de dicha segmentación:

Figura 6

Red residual. Un bloque de construcción



Nota. (He et al., 2016).

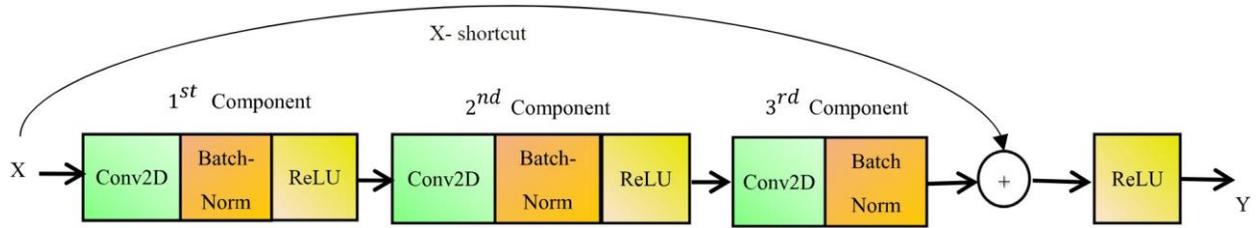
(He et al., 2016) mencionan el significado de apartado de la arquitectura, a continuación, se detalla cada una:

- **X:** Entrada de mapas de características.
- **F(X):** Mapas de características resultado de una capa de convolución.
- **X identity:** Representa la entrada X que avanza por la red de forma directa sin pasar por capas convolucionales.
- **F(X) + X:** Mapas de características resultado de la última capa de convolución de bloque residual a la que se suman los mapas de características de la entrada del bloque residual.

A continuación, se muestra en la figura 7 la representación más completa del bloque de identidad, en la figura 8 el bloque convolucional y en la figura 9 la arquitectura de ResNet 50.

Figura 7

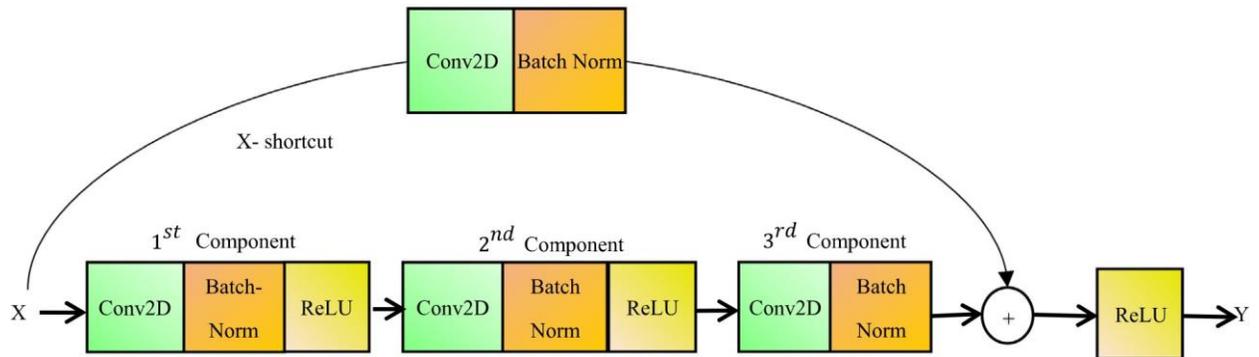
Bloque identidad



Nota. (Shehab et al., 2021).

Figura 8

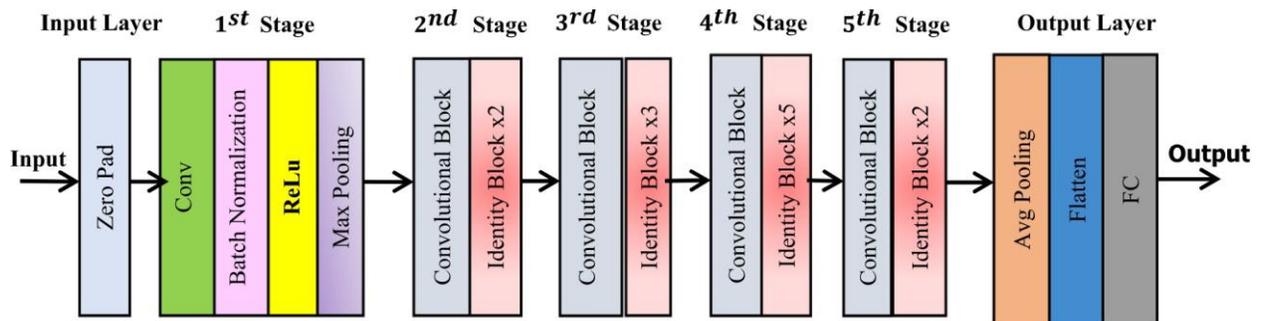
Bloque convolucional



Nota. (Shehab et al., 2021).

Figura 9

ResNet 50



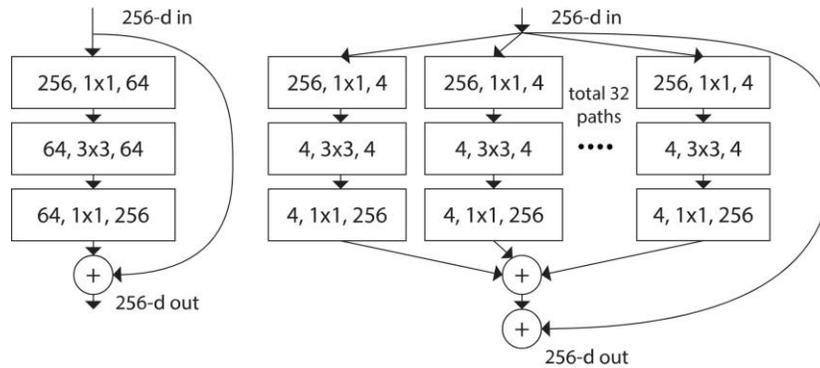
Nota. (Shehab et al., 2021).

1.3.3 ResNeXt

Esta arquitectura toma las bases de ResNet e Inception, ResNet es la base fundamental, se podría decir que es una versión mejorada de esta, la modificación que se le hizo fue implementar el concepto de cardinalidad de Inception, esto se refiere a dividir un canal de flujo, en este caso un bloque convolucional, en varios bloques de convolución paralelos que tengan un menor número de filtros que el bloque convolucional original (Xie et al., 2017). A continuación, en la figura 10 se muestra el diagrama de un bloque residual de ResNet y ResNeXt.

Figura 10

Bloque residual de ResNet y ResNeXt

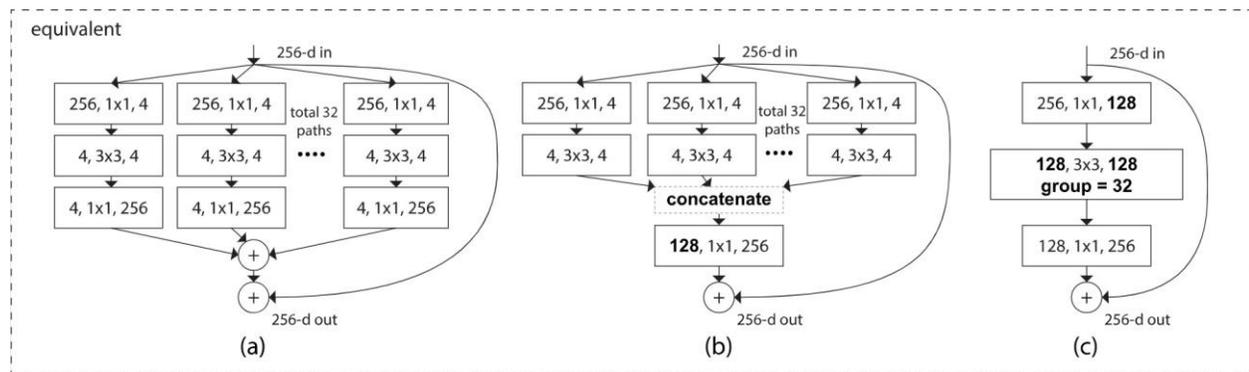


Nota. (Xie et al., 2017).

Según los autores que propusieron esta arquitectura ofrece mejores resultados en comparación que la versión original de ResNet, ya que, aumentar la cardinalidad es más efectivo que aumentar profundidad (Xie et al., 2017). Los autores además ofrecen distintas equivalencias para la implementación de ResNeXt, a continuación, en la figura 11 se muestra las equivalencias.

Figura 11

Equivalencias de la arquitectura ResNeXt



Nota. (Xie et al., 2017).

Beneficios

En el trabajo de (Xie et al., 2017) ellos mencionan que el mayor beneficio es un aumento en la precisión, esto en comparación con la arquitectura ResNet original, ellos llegan a la conclusión de que aumentar más anchura a la red, ósea mayor cardinalidad es mejor para aumentar el nivel de precisión.

Contras

En los beneficios se había mencionado que aumentar la cardinalidad en ResNeXt es mejor ya que permite tener un nivel de precisión más alto, sin embargo, una mayor cardinalidad también conlleva a un aumento en el uso de memoria y consumo de otros recursos como poder de procesamiento, una versión de ResNeXt demasiado profunda y con demasiada anchura podría no ser la mejor opción si se prioriza el tiempo y la versatilidad (Xie et al., 2017), es decir, no sería ideal para implementarla en sistemas embebidos y tampoco utilizarla contextos que se priorice el tiempo antes que la precisión.

1.4 Metodologías

1.4.1 KDD

Significa Descubrimiento de Conocimiento en Base de Datos, es una serie de procesos estructurados que se deben llevar a cabo con el fin de obtener conocimientos valiosos a partir de un conjunto de datos, que por lo general contienen miles o millones de dato (Fayyad et al., 1996). Los conocimientos que se pueden obtener al aplicar esta metodología van desde patrones, tendencias y comportamientos, el descubrimiento oportuno y claro de esta información de valor es útil en la toma de decisiones (Linsel et al., 2020).

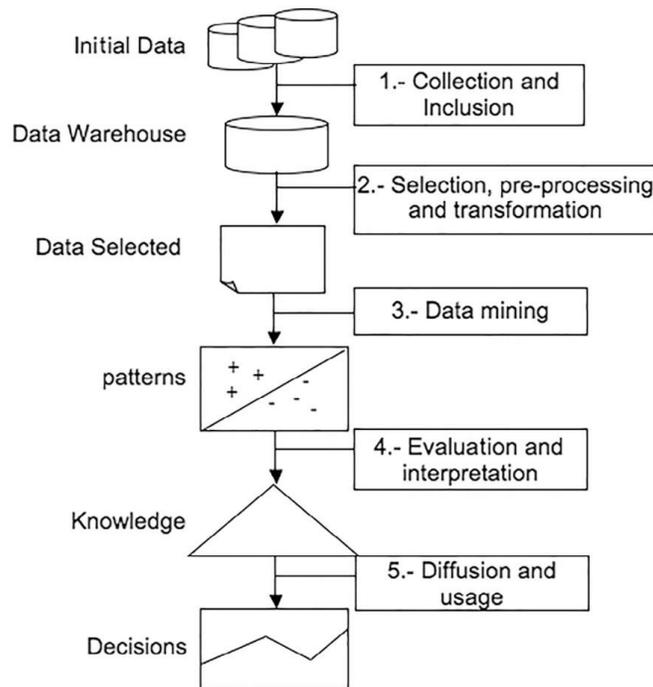
La aplicación de la metodología KDD debe priorizar la novedad, utilidad, generalidad y comprensibilidad a modo de aprovechar todo su potencial (Fayyad et al., 1996).

Etapas

Las etapas de la metodología KDD se plasman en la figura 12.

Figura 12

Diagrama de flujo de la metodología KDD



Nota. (Fernández-Arteaga et al., 2016).

Martarelli & Nagano (2022) proponen la siguiente definición para cada una de las etapas:

- **Recopilación e inclusión:** En esta se recopila la información con mayor grado de significancia, se la debe almacenar en una base de datos que será utilizada en los siguientes procesos.
- **Selección, preprocesamiento y transformación:** Se hace un tratamiento de los datos donde se les da una estructura apropiada para el análisis y procesamiento. Se aplican distintas transformaciones que van desde el reemplazo o eliminación de datos nulos, tratamiento de valores atípicos, conversiones de magnitudes y formatos, etc.
- **Minería de datos:** Esta etapa es la minería de datos, se usa modelos matemáticos y algorítmicos que sean capaces de descubrir patrones relevantes y se manifiesten como algo entendible y útil para la toma de decisiones. Es muy común hacer uso de algoritmos de Machine Learning y Deep Learning ya que estos son muy prácticos para la detección de patrones.
- **Evaluación e interpretación:** Una vez concluida la etapa de minería de datos se analizan los patrones descubiertos y se les da un significado.
- **Difusión y uso:** Se hace uso del conocimiento obtenido para la toma de decisiones en un área de interés, se utilizan herramientas como gráficos, informes, entre otros que faciliten la comprensión del conocimiento encontrado para los usuarios que lo necesiten, también puede ser un punto de partida para la generación de nuevo conocimiento a partir del descubierto.

1.4.2 Kanban

Es un método ágil que tiene como propósito reducir en la medida de lo posible las actividades que no aporten valor tanto para el equipo de trabajo como para el cliente, se basa en

un sistema Pull, esto quiere decir que el trabajo a realizar se ajusta a la demanda de los clientes, esto permite enfocarse en las tareas significativas que aportan valor (SUGIMORI et al., 1977).

A diferencia de Scrum, Kanban pretende entregar el máximo valor en el menor tiempo sin enfocarse en la documentación, no existe un tiempo de entrega común para todas las tareas a realizar, en Scrum está definido por la duración de los Sprints, esto permite que las tareas sean terminadas lo más pronto posible y continuar con las siguientes (Ellis, 2020).

Al tratarse de un método que se enfoca en la percepción visual del proceso de trabajo, es de utilidad para diagnosticar cuellos de botellas y en que etapa del proceso se están dando los problemas, esto permite tomar decisiones rápidas lo que se traduce en una mejor eficiencia en el trabajo, una mejora y evolución continua (Ellis, 2020).

Kiran (2019) describe los siguientes términos del método Kanban:

- **Tablero:** Puede ser físico o virtual, su propósito es proporcionar un área en el que los demás elementos del método Kanban puedan coexistir.
- **Tarjetas:** Son la representación de las tareas que deben ser realizadas, en las tarjetas se debe plasmar toda la información necesaria a modo que la persona a la que se le designe dicha tarea pueda realizarla eficientemente, la información puede ser: descripción de la actividad, tiempo máximo de entrega, encargado de realizarla, etc.
- **Carriles:** Su propósito es establecer una clasificación de las tarjetas, dicha clasificación puede ser: por naturaleza, tipo de actividad, prioridad, etc.
- **WIP:** Es las siglas en inglés de trabajo en proceso, son todas las actividades que se están realizando en cualquier etapa del método Kanban.
- **Limites WIP:** En Kanban un requisito es que las etapas tengan un límite de tareas en proceso, esto tiene como finalidad evitar los cuellos de botella y ser

productivos, si no se establecen límites WIP no se está aplicando el método Kanban.

- **Etapas:** Son secciones del proceso para llevar a cabo una tarea, las más básicas son: requerido, en proceso y hecho. Al tratarse de un método ágil dichas secciones pueden ser modificadas y ajustadas a las necesidades de la empresa o proyecto, lo importantes es mantener por lo menos 3 secciones, ya que de lo contrario no se puede hacer un análisis de la existencia de cuellos de botella.

1.5 Normas ISO

Establecidas por la Organización Internacional de Normalización, se crean con el fin de estandarizar los procesos y productos a nivel mundial, además permiten mejorar la eficiencia, calidad y seguridad de los elementos mencionados. Están conformadas por requisitos, especificaciones y métricas de evaluación que cualquier empresa o asociación que desee adoptarlas debe cumplirlas para poder obtener una certificación (ISO, 2022).

1.5.1 ISO 25010 (*Modelo de calidad de sistemas y software*)

Se enfoca en el modelo de calidad la cual brinda las características y subcaracterísticas que se deben tener en cuenta para la valoración del producto de software, pertenece a la familia 25000 que se enfoca en la evaluación de la calidad del software. Presenta ocho características las cuales son: adecuación funcional, fiabilidad, compatibilidad, facilidad de uso, seguridad, compatibilidad, mantenibilidad y portabilidad (ISO 25000, 2022).

Característica de portabilidad

Esta característica mide el grado que tiene un componente o producto de software de ser trasladado de un entorno a otro (ISO 25000, 2022), se componen de las siguientes subcaracterísticas:

- **Adaptabilidad:** Calidad de adaptación de manera competente en los entornos.

- **Capacidad de ser instalado:** Calidad de instalación y desinstalación en el entorno elegido.
- **Capacidad de ser reemplazado:** Calidad para sustituirlo por otro software que cumpla con el mismo objetivo y entorno que el sustituido.

1.5.2 ISO 25023 (Medición de la calidad del sistema y del producto de software)

Establece las métricas a utilizar para determinar la calidad del producto de software, van ligadas al modelo calidad impuesto por la norma ISO 25010 (ISO, 2016).

Métricas para la subcaracterística de capacidad de ser instalado

En la figura 13 se detalla las métricas a utilizar para evaluar la subcaracterística capacidad de ser instalado.

Figura 13

Métricas de la ISO 25023 para la evaluación de la subcaracterística capacidad de ser instalado

Métricas para la característica de calidad Portabilidad								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Capacidad de ser instalado	Eficiencia en el tiempo de instalación	Externa	¿Cuánto tiempo es requerido para realizar una instalación?	Contar el tiempo total transcurrido al instalar el sistema y contar el número de reintentos al instalar el sistema	$X = A/T$ A = Número de reintentos al instalar el sistema T = Tiempo total transcurrido al instalar el sistema Dónde: $T > 0$	$X = A/T$ El más lejano a 0/t es el mejor	X= Contable/ Contable A= Contable B= Contable	Desarrollador, Tester
	Facilidad de instalación	Externa	¿Puede fácilmente el usuario o el desarrollador instalar el software en un entorno operacional?	Contar el número casos en que los usuarios tuvieron éxito al instalar el sistema cambiando proceso de instalación para su conveniencia y contar el número total de casos en que los usuarios han intentado cambiar el proceso de instalación para su conveniencia	$X = A/B$ A = Número casos en que los usuarios tuvieron éxito al instalar el sistema cambiando proceso de instalación para su conveniencia B = Número total de casos en que los usuarios han intentado cambiar el proceso de instalación para su conveniencia Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Desarrollador, Tester

Nota. (Balseca, 2014).

1.6 Aplicación web

Una aplicación web es un software que tiene la particularidad de ser ejecutado y utilizado mediante un navegador web, esto permite que la aplicación sea accedida desde cualquier dispositivo y sistema operativo que soporte la ejecución de un navegador. Los principales componentes son el cliente, el servidor y el protocolo de comunicación HTTP o HTTPS (Guerra, 2021).

1.6.1 Componentes

Ciente web

Un cliente web es descrito como un software que se encarga de la solicitud de recursos al servidor, es un navegador web en cual está instalado en el dispositivo del cliente ya sea un celular, tableta, computadora, etc. Es usado directamente por el usuario y funciona como interfaz gráfica para que pueda realizar las acciones que desee (Chaucha de la Cruz et al., 2021).

Servidor web

Es un software alojado en un host el cual atiende las solicitudes enviadas por parte del o los clientes, las procesa y envía una respuesta de ser necesario. No es visible para el usuario y en este se procesa el algoritmo de la lógica del negocio programada por un desarrollador (Chaucha de la Cruz et al., 2021).

Protocolo de comunicación

Son reglas estandarizadas que describen como se deben comunicar e interactuar los componentes de un software web. El protocolo más utilizado es HTTP y HTTPS, el primero significa Protocolo de Transferencia de Hipertexto y el segundo es una versión cifrada del primero, este protocolo establece como un navegador web pide y recibe recursos del servidor tales como páginas, recursos multimedia, información, entre otros (Guerra, 2021).

1.6.2 *Arquitectura*

Es un modelo estandarizado de alto nivel que detalla los componentes que conforman un software, a su vez explica la función que desempeña cada uno de ellos y como interactúan con los demás. La principal característica es que no está sujeta a ningún lenguaje de programación o paradigma, su función es estandarizar la manera en cómo se estructura un software, para que sea mantenible y escalable en el tiempo (Ospino, 2018).

Cliente Servidor

El funcionamiento se basa en que el usuario utiliza el cliente web para poder interactuar, dicha interacción se traduce en solicitudes al servidor tales como peticiones de archivos, procesamiento de datos, entre otros, el servidor se encarga de procesar dichas peticiones, en este punto las solicitudes pueden ser o no procesadas, en cualquier caso, el servidor responderá al cliente web el estado de la petición (Chavarria & Gudiño, 2017).

1.6.3 *Lenguaje de programación*

Es un mecanismo que permite establecer comunicación con el hardware de un dispositivo y darle órdenes de que tareas debe realizar. Se trata de una escritura lógica y estructurada que debe respetar una sintaxis en específico, su propósito es permitir a los desarrolladores programar las tareas que se desean realizar, es una capa de abstracción entre el usuario y el hardware, facilitando la comunicación y programación (Catillo & Gómez, 2018).

Python

Es un lenguaje de programación del alto nivel el cual tiene la característica de ser interpretado y de tipado dinámico, soporta el paradigma de programación orientado a objetos y es utilizado para el desarrollo de aplicaciones web, ciencia de datos, machine learning, entre otros (Python TM, 2023).

JavaScript

En sus inicios se pensó como un lenguaje de programación que se ejecutaba solo en el navegador web, sin embargo, en la actualidad también puede ser utilizado como un lenguaje de Back End gracias a Node.js. Es un lenguaje de tipado dinámico e interpretado, soporta el paradigma de programación orientada a objetos y es multiplataforma (JavaScript, 2023).

1.6.4 Framework

Es un entorno definido el cual contiene una serie de normas que el desarrollador debe seguir para desarrollar software, además ofrece librerías y herramientas que facilitan el desarrollo. Una característica para destacar es que estos implementan patrones de diseño y buenas prácticas de desarrollo, por lo que el uso de marcos de trabajo permite que el software desarrollado sea de alta calidad, escalabilidad y mantenibilidad (Ospino, 2018).

Angular

Es un framework para el desarrollo de Front End el cual está desarrollado en TypeScript, se basa en componentes y las aplicaciones desarrolladas en este son de tipo single page, además implementa el patrón de diseño MVC. Fue desarrollado por Google y es de código abierto. Pese a que impone el uso de TypeScript también se puede utilizar JavaScript (Angular, 2022).

Express.js

Es un framework que se ejecuta en el entorno de Node.js y fue pensado para el desarrollo rápido de aplicaciones web, concretamente la parte del Back End. Es un marco de trabajo de JavaScript y es de código abierto. Se integra muy bien con frameworks como Angular y React, además cuenta con mucho apoyo de la comunidad (Express, 2017).

1.6.5 Web Service

Es un conjunto de métodos y técnicas que se utilizan para establecer comunicación a través de la red entre sistemas sin importar el lenguaje de programación utilizado para el desarrollo de dichos sistemas. Su principal propósito es el intercambio de información, para lo cual se utilizan archivos JSON o XML, que son los más utilizados. Pueden soportar tecnologías como REST, SOAP o GraphQL (Al-Musawi et al., 2023).

REST

Hace referencia a un conjunto de principios de arquitectura que son flexibles a las necesidades de los servicios web, se basa en un conjunto de pautas, por ende, la implementación de dichas pautas depende de los desarrolladores. El protocolo de comunicación para su uso es HTTP, utiliza verbos los cuales permiten realizar operaciones como por ejemplo envío de datos como obtención de estos, los verbos más comunes son: GET, POST, PUT, DELETE. El formato más utilizado para el intercambio de datos es JSON por su ligereza y mejor interpretación a comparación de XML (Meng et al., 2009).

1.6.6 API

Es un software que permite la transmisión de datos de un sistema a otro sin que estos sepan la estructura y funcionamiento del otro. Son descritas como contratos, los cuales tienen una serie de pautas que se deben seguir para estructurar las peticiones y respuestas. El principal beneficio es que puede ser utilizada por cualquier tipo de implementación de software, esto hace que no se deba desarrollar un Back End en específico para cada plataforma (Felício et al., 2023).

API REST

Se le denomina API REST o RESTful a la implementación de las pautas impuestas por la definición del Web Service REST (Palma et al., 2022).

1.7 Trabajos relacionados

End to End Segmentation of Canola Field Images Using Dilated U-Net

H. S. Ullah et al. (2021) en su trabajo de investigación mencionan que han propuesto una versión de U-Net modificada para que esta haga un menor uso de memoria y recursos computacionales con la finalidad de poder implementarla en sistemas que tengan limitaciones de hardware. El cultivo donde fue adquirido el data set es una plantación de canola, únicamente se definió las clases background, maleza y cultivo para la anotación del data set, en total se lograron capturar 10000 imágenes en 2 etapas de crecimiento, en la primera etapa se reporta que el cultivo tiene 2 hojas desplegadas y en la segunda más de 5 hojas. Las imágenes fueron recolectadas por un Quad en formato RGB, la resolución no se especifica, solo se menciona que es de alta calidad. Las redes que se probaron fueron: U-Net simple, U-Net(ResNet-50), SegNet(ResNet-50), DeepLabv3+(ResNet-50), HRNet_Mscale y la U-Net modificada. Los hiperparámetros de entrenamiento fueron: número de épocas 200, optimizador Adam, tasa de aprendizaje 0.0001. La distribución del dataset no se especifica como tal, se decidió dividir las imágenes en mosaicos de 800 x 512 para el entrenamiento. Los resultados obtenidos para cada red fueron: U-Net simple mean IOU 0.8971, IOU del background 0.9834, IOU de la canola 0.8665, IOU de las malezas 0.8432; U-Net(ResNet-50) mean IOU 0.8873, IOU del background 0.9815, IOU de la canola 0.8620, IOU de las malezas 0.8183; SegNet(ResNet-50) mean IOU 0.8778, IOU del background 0.9804, IOU de la canola 0.8354, IOU de las malezas 0.8176; DeepLabv3+(ResNet-50) mean IOU 0.8958, IOU del background 0.9829, IOU de la canola 0.8685, IOU de las malezas 0.8359; HRNet_Mscale mean IOU 0.9034, IOU del background 0.9633, IOU de la canola 0.8932, IOU de las malezas 0.8536; U-Net modificada mean IOU 0.8912, IOU del background 0.9823, IOU de la canola 0.8611, IOU de las malezas 0.8299. Como conclusiones se dice que la versión modificada de U-Net es peor en solo 1.2% con respecto a HRNet_Mscale, pero utiliza 57 millones de parámetros menos.

Method for Segmentation of Bean Crop and Weeds Based on Improved UperNet

Qi et al. (2023) proponen una arquitectura PF-UperNet basada en UperNet con la finalidad de disminuir la cantidad de parámetros y aumentar el nivel de precisión. El enfoque fue detectar malezas en cultivos de frijol para lo cual se utilizó con conjunto de datos público elaborado por Jehan-Antoine Vayssade haciendo uso de una cámara multiespectral con un FWHM DE 10 nm, en total se utilizaron 300 imágenes con sus respectivas máscaras, las categorías establecidas fueron background, frijol y maleza, el conjunto de entrenamiento, validación y prueba fue establecido en 70%, 10% y 20% respectivamente. Los hiperparámetros definidos fueron: épocas 400, tamaño de lote 12, optimizador AdamW, tasa de aprendizaje de 0.0002 con un decaimiento de 0.0001. En total se definieron 5 versiones del modelo basado en UperNet, los resultados obtenido para cada modelo fueron: modelo 1 mean IOU 86.37%, IOU del background 96.72%, IOU del frijol 77.92% y IOU de las malezas 84.46%; modelo 2 mean IOU 87.17%, IOU del background 96.93%, IOU del frijol 79.18% y IOU de las malezas 85.4%; modelo 3 mean IOU 87.28%, IOU del background 96.99%, IOU del frijol 79.25% y IOU de las malezas 85.62%; modelo 4 mean IOU 87.36%, IOU del background 96.99%, IOU del frijol 79.48% y IOU de las malezas 85.51%; modelo 5 mean IOU 87.45%, IOU del background 96.91%, IOU del frijol 79.79% y IOU de las malezas 85.65%. Como conclusiones se menciona que la mejor versión del modelo, la versión 5 presenta un mean IOU del 87,45% y una mejora del 1.08% en esta métrica con respecto a UperNet, además de una reducción del 27,92% en la cantidad de parámetros.

Semantic Segmentation of Crops and Weeds with Probabilistic Modeling and Uncertainty Quantification

Celikkan et al. (2023) en su trabajo proponen un método para disminuir la incertidumbre en las predicciones de segmentación semántica probabilística enfocadas a la detección de malezas. Para realizar las experimentaciones se utilizó el conjunto de datos público Sugarbeets2016, este dataset fue elaborado con ayuda de un robot de campo en cultivos de remolacha azucarera en Boon, Alemania, las categorías definidas son: background, cultivo y

maleza. La distribución del conjunto de entrenamiento, validación y pruebas fue 75%, 15% y 15% respectivamente. La arquitectura utilizada fue una combinación de ResNet50 con DeepLabv3. Los hiperparámetros definidos para el entrenamiento fueron: tasa de aprendizaje de 0.001, optimizador Adam, número de épocas 160 y un tamaño de lote de 4. Los resultados obtenidos fueron un mean IOU del 85.77%, IOU de las malezas 63,37%, IOU del cultivo 94.60%, IOU del background 99.33%. Las conclusiones que se da en el trabajo es que el método propuesto mejora el rendimiento de la segmentación competitiva, además que se resaltan las áreas en las que se tiene mayor incertidumbre.

Sugar Beets and Weed Detection using Semantic Segmentation

X.-Z. Hu et al. (2022) proponen en su trabajo de investigación una arquitectura UNet++ con supervisión profunda para mejorar la precisión en la detección de malezas en cultivos de remolacha azucarera. Para hacer la comparativa se entrenaron 3 modelos los cuales son: UNet, UNet++ y UNet con supervisión profunda. El data set está conformado por 1026 imágenes en formato PNG de cultivos de remolacha azucarera con malezas con una resolución de 480px por 360 px, la división del conjunto de entrenamiento y validación es de 80% y 20% respectivamente. Los hiperparámetros elegidos para el entrenamiento fueron: tasa de aprendizaje de $3e-4$, optimizador Adam, número de épocas 200. Los resultados obtenidos al final del entrenamiento fueron: UNet mean IOU 90.81%, IOU del background 92.15%, IOU de las malezas 95.65%, IOU de la remolacha 92.54%; UNet++ mean IOU 92.00%, IOU del background 92.15%, IOU de las malezas 95.66%, IOU de la remolacha 92.54%; UNet ++ con supervisión profunda mean IOU 92.34%, IOU del background 92.15%, IOU de las malezas 95.65%, IOU de la remolacha 92.54%. Como conclusión se afirma que el modelo propuesto de UNet++ con supervisión profunda es mejora ya que su mean IOU es del 92.34%, siendo el valor más alto de los 3 modelos entrenados.

DEEP LEARNING BASED OVERCOMPLETE REPRESENTATIONS FOR PADDY RICE CROP AND WEED SEGMENTATION

Sai et al. (2022) proponen una arquitectura de nominada Kite-Net basada en el modelo TerausNet, esto con el fin de mejorar el nivel de precisión con respecto a escenarios donde exista mucho ruido en las imágenes. Los datos fueron recolectados del Centro de Investigación Agroclimática de la Universidad Estatal de Agricultura Profesor Jayashankar Telangana, el cultivo elegido fue de arroz con cáscara, las imágenes fueron capturados de 2 maneras, la primera utilizando una cámara estática Sony RX 100 y la segunda forma utilizando un dron DJI Inspire-1 Pro UAV, las categorías definidas para la anotación fueron: cultivo, maleza y background. Para el entrenamiento se utilizaron 105 imágenes de 128px por 128px, los hiperparámetros fueron: tamaño de lote de 1 imagen, optimizador Adam, tasa de aprendizaje de 0.009 y un momentum de 0.9. Para poder analizar los resultados se entrenaron 4 modelos y se obtuvieron los siguientes resultados: U-Net mean IOU 0.4248, dice coefficient 0.3895; LinkNet mean IOU 0.4965, dice coefficient 0.4531; TerausNet mean IOU 0.5721, dice coefficient 0.5269; KiU-Net mean IOU 0.6355, dice coefficient 0.6014; modelo propuesto en el trabajo mean IOU 0.7255, dice coefficient 0.6983. Como conclusiones se afirma que el modelo propuesto es mejor a los otros entrenados obteniendo un mean IOU del 72.5% y un dice coefficient del 70%.

DeepVeg: Deep Learning Model for Segmentation of Weed, Canola, and Canola Flea Beetle Damage

Das & Bais (2021) proponen la arquitectura denominada DeepVeg, esta pretende solventar el problema del desbalance de clases que existe en el data set utilizado, las clases definidas para la etiquetación son: background, cultivo de canola, maleza y daños en el cultivo. El conjunto de datos está compuesto por 13000 imágenes RGB, sin embargo, los autores mencionaron que solo se etiquetaron 384 imágenes debido a la falta de tiempo, las imágenes tienen la resolución de 4016px por 6016px, el conjunto de entrenamiento, validación y pruebas se dividió en 70%, 10% y 20% respectivamente. Para el entrenamiento las imágenes fueron

divididas en resolución de 320px por 320px. Los hiperparámetros utilizados fueron: tasa de aprendizaje inicial de 2×10^{-4} con un decaimiento de 75×10^{-2} , optimizador Adam, tamaño de lote igual a 2. Los resultados obtenidos para el modelo propuesto fueron: mean IOU 0.7679, IOU del background 0.9816, IOU del daño en el cultivo 0.6890, IOU de las malezas 0.6575 y el IOU de las plantas de canola 0.7434. Como conclusión se menciona que el modelo propuesto aún tiene dificultades para identificar el daño en el cultivo al mismo tiempo que clasifica los tipos de categorías existentes, se aconseja utilizar datos sintéticos para equilibrar las clases.

1.8 Preguntas de investigación

Las preguntas de investigación planteadas para este trabajo se detallan y contestan en la tabla 3.

Tabla 3

Preguntas de investigación propuestas por el PhD. Iván García

Pregunta	Artículo	Answer
RQ1: ¿Qué drones y cámaras están usando para la tarea de detección automática de malas hierbas?	(Farooq et al., 2018), (Farooq et al., 2019), (Yu et al., 2019), (Asad & Bais, 2020), (Espejo-Garcia, Mylonas, Athanasakos, Fountas, et al., 2020), (Veeranampalayam Sivakumar et al., 2020), (Lam et al., 2021), (Kounalakis et al., 2018).	Drones: DJI Matrice 600, DJI Phantom 3 y 4, DJI Mavic 2 Pro, DJI Spark, DJI Mavic, DJI Inspire 2, Headwall's Nhao Hyperspec. Cámaras: Canon EOS 70D, JAI BM-141GE, XIMEA, SONY Cyber-Shot Digital, Nikon D610, Nikon D700, Canon PowerShot SX600, Nikon D7200, Intel RealSense RGB-D, Canon SD110, Teledyne DALSA Genie Nano, Sony A600, JAI Camera, cámaras monocromáticas, cámaras de celular.

<p>RQ2: ¿Cuál es la altura y velocidad adecuada del dron para capturar imágenes y regiones de interés (ROIs) de alta resolución (px/cm)?</p>	<p>(R. Zhang et al., 2018), (Osorio et al., 2020), (Liang et al., 2019), (Tetila et al., 2020), (Huang et al., 2018), (Beeharry & Bassoo, 2020), (Bah et al., 2019), (Zhou et al., 2021), (Sharpe et al., 2019)</p>	<p>Altura (m): 2, 2.5, 5, 10, 15, 20.</p> <p>Velocidad (km/h): 7, 14, 21.</p> <p>Resolución de obtención (px): 4000 x 6000, 3000 x 4000, 1280 x 720, 1024 x 256, 480 x 360, 640 x 360.</p> <p>Resolución después del procesamiento (px): 64 x 64, 128 x 128, 224 x 224, 228 x 228, 256 x 256, 288 x 288, 360 x 360</p>
<p>RQ3: ¿Qué espacio de color y formato de imágenes resulta adecuado?</p>	<p>(Naveed & Qazi, 2020), (Czymmek et al., 2019), (Lottes et al., 2020)</p>	<p>Espacio de color: RGB, Multiespectral, Hiperespectral.</p> <p>Formato: JPEG, PNG.</p>
<p>RQ4: ¿En qué bandas del espectro electromagnético (visible, infrarrojo, ultravioleta) están trabajando?</p>	<p>(Huang et al., 2020), (Naveed & Qazi, 2020)</p>	<p>Visible.</p>
<p>RQ5: ¿Qué cultivos y tipos de malezas se están detectando y dónde?</p>	<p>(Huang et al., 2020), (Gao et al., 2020), (Veeranampalayam Sivakumar et al., 2020), (Czymmek et al., 2019), (R. Zhang et al., 2018),</p>	<p>Cultivo-maleza:</p> <p>-Arroz: <i>Leptochloa chinensis</i>, <i>Cyperus iria</i>, <i>Digitaria sanguinalis</i> (L). Scop, Barnyard Grass.</p> <p>-Remolacha azucarera: <i>Convolvulus sepium</i>.</p> <p>-Haba de soja: <i>Waterhemp</i>, <i>Palmer amaranthus</i>, <i>common lambsquarters</i>, <i>velvetleaf</i>, <i>foxtail species</i>.</p>

(Sharpe et al., 2020) **-Trigo:** Cirsium Setosum, Descurainia Sophia, Euphorbia Helioscopia, Veronica Didyma, Avena Fatu.

-Zanahoria: Amaranthus retroflexus, Cichorium intybus, Urtica dioica.

-Fresa: Goosegrass.

Localización: Sur de China, Bélgica, Nebraska, norte de Yangling, Alemania, Florida, Australia.

RQ6: ¿Qué conjunto de datos se encuentran públicamente para el entrenamiento de un modelo de Deep Learning?

(Teimouri et al., 2018), (Bosilj et al., 2020), (Binguitcha-Fare & Sharma, 2019), (Arun et al., 2020), (A. Khan et al., 2020), (K. Hu et al., 2020), (Espejo-Garcia, Mylonas, Athanasakos, & Fountas, 2020), (Chebrolu et al., 2017).

-Conjunto de datos de la Universidad de Aarhus Grupo de procesamiento de señales.

-Conjunto de datos elaborado por Teimouri.

-Conjunto de datos elaborado por Bosilj.

-Conjunto de datos elaborado por Chebrolu.

-Crop/Weed Field Image Dataset.

-Rice seeding and weed dataset.

-BoniRob dataset.

-Carrot crop vs. weed dataset.

-Paddy–millet dataset.

-DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning.

-Early crop weed dataset.

-Soybean and weed crops.

RQ7: ¿Cómo crear y anotar un conjunto de datos propio? (Veeranampalayam Sivakumar et al., 2020), (Teimouri et al., 2018), (Chebrolu et al., 2017)

Las imágenes se pueden adquirir a través de cámaras instaladas en:

- Vehículos aéreos no tripulados (UAVs).
- Robots de campo (FRs).
- Vehículos todo terreno (ATVs).
- Satélites.
- Conjunto de datos públicos.

-Estudios demuestran que el formato JPEG es el más apropiado para algoritmos de DL, por lo que las imágenes obtenidas deben convertirse a este formato.

-Posteriormente se debe reducir las dimensiones de las imágenes a fin de no dar carga excesiva a la GPU, por ejemplo, a formato 640px x 360px. Además, dependiendo de las imágenes se puede aplicar distintas técnicas para el preprocesamiento, como reducción de ruido, aumentar la iluminación, etc.

-Si se cuenta con un número reducido de imágenes es conveniente utilizar técnicas para el aumento de datos tales como: rotación, escalado, cizallamiento, volteado, corrección gamma, espacio de color, transformación de espacio de color, inyección de ruido, filtrado de kernel, recorte, traslación.

Existen varios tipos de anotación específicamente para la detección de malezas, entre los cuales se destacan:

-Anotación de nivel de píxel.

-Anotación de cuadros delimitadores.

-Anotación de polígono.

-Anotación de nivel de imagen.

-Etiquetado sintético.

-Un software muy utilizado es LabelMe.

RQ8: ¿Qué arquitectura de CNN se usan para la detección automática de malas hierbas? (Teimouri et al., 2018), (Binguitcha-Fare & Sharma, 2019), (Z. Jiang, 2019), (Yu et al., 2019), (Espejo-Garcia, Mylonas, Athanasakos, Fountas, et al., 2020). YOLOV3, YOLOV5, Faster R-CNN, VGG16, AlexNet, ResNet50, ResNet101, GoogleNet, DetectNet, InceptionV3, VGGNet, Xception, VGNets, Mobilenet, Mask R-CNN.

RQ9: ¿Qué hardware específico o servicios en la nube resulta adecuado para entrenar modelos de DL? (Gao et al., 2020), (Tetila et al., 2020), (Espejo-Garcia, Mylonas, Athanasakos, & Fountas, 2020) Tarjetas gráficas Nvidia.de la serie TITAN.

RQ10: ¿Qué métricas y gráficas se usan para validar el (Arun et al., 2020), (A. Khan et al., 2020), (K. Hu et al., 2020). -Classification Accuracy (CA).
-True Positive (TP).

rendimiento de los 2020), (Espejo-
modelos de DL? Garcia, Mylonas,
Athanasakos, &
Fountas, 2020)

- False Positive Rate (FPR).
- False Negative Rate (FNR).
- Specificity (S).
- Mean Pixel Accuracy (MPA).
- Precision (P).
- Mean Average Precision (mAP).
- F1 Score (F1).
- Confusion Matrix (CM).
- Intersection over Union (IoU).
- Mean Intersection over Union (mIoU).
- Frequency Weighted Intersection over Union (FWIoU).
- Mean Square Error (MSE).
- Root Mean Square Error (RMSE).
- Mean Absolute Error (MAE).
- R2.
- K-fold Cross Validation.
- Receiver Operating Characteristic (ROC) curve.
- Kappa Coefficient.
- Matthews correlation coefficient (MCC).

		-Dice Similarity Coefficient (DSC).
RQ11: ¿Cómo afinar un modelo de DL para mejorar el rendimiento?	(Lottes, Behley, Chebrolu, et al., 2018), (H. Jiang et al., 2020), (Espejo-Garcia, Mylonas, Athanasakos, Fountas, et al., 2020), (Tetila et al., 2020)	-Imágenes de entrenamiento en formato JPG. -Kernel de 3 x 3, 5 x 5, 11 x 11 -Tamaño de lote en múltiplo de 2, por lo general 16 o 32. -Taza de aprendizaje 0,001 o 0,0001. -Función de activación ReLU. -Utilizar una CNN preentrenada para la transferencia de aprendizaje.

Nota. Elaboración propia.

CAPÍTULO 2

Desarrollo

Este proyecto de titulación tiene como finalidad desarrollar una aplicación web para la identificación y cuantificación de malezas en cultivos de papa utilizando imágenes adquiridas por un dron, esta aplicación implementa una red completamente convolucional basada en ResNeXt50. El propósito para el desarrollo de esta aplicación es reducir el tiempo empleado en la cuantificación de malezas y utilizar un método matemático para el cálculo de la densidad porcentual de las malas hierbas.

2.1 Visión general del proyecto

Se utilizó como metodologías a KDD y Kanban, KDD está destinada a todo el proceso que inicia con la captura de las imágenes para conformar el conjunto de datos hasta la difusión y uso de la red neuronal convolucional, Kanban fue utilizada en todas las fases de la metodología KDD para facilitar su ejecución, también sirvió como instrumento de planificación en todo el proceso de desarrollo y puesta en producción de la aplicación web en un servidor cloud para su uso general.

2.1.1 Planificación

La planificación propuesta para el desarrollo del proyecto se muestra en la tabla 4.

Tabla 4

Planificación para el desarrollo del proyecto

Actividades	Tiempo estimado
Captura de imágenes de cultivos de papa con presencia de malezas utilizando un UAV.	7 semanas.
Clasificación y procesamiento de las imágenes.	3 semanas.
Anotación del data set enfocado a la segmentación semántica de los tipos de malezas y papa.	7 semanas.
Programación de la arquitectura ResNext50.	1 semana.

Entrenamiento del algoritmo de deep learning.	1 semana.
Definición de un base line y realización de mejoras para disminuir el nivel de pérdida de la red.	2 semanas.
Desarrollo de la aplicación web para la identificación y cuantificación de malezas mediante imágenes.	1 semana.
Despliegue de la aplicación web en un servidor cloud.	1/2 semana.

Nota. Elaboración propia.

2.1.2 Entregables del proyecto.

Los entregables del proyecto se detallan a continuación en la tabla 5, los cuales serán entregados al director del proyecto el PhD. Iván García una vez finalizado todo el trabajo de titulación.

Tabla 5

Entregables del proyecto de titulación

Entregables	Descripción
Data set.	Data set conformado por imágenes de 128px por 128 px de plantas de papa y malezas con sus respectivas máscaras de segmentación semántica.
Modelo de deep learning ResNext50.	Best model compilado en formato h5.
Métricas de inteligencia artificial e interpretación de estas.	Análisis e interpretación de las métricas asociadas a problemas de segmentación semántica de todas las versiones del modelo entrenadas.

Nota. Elaboración propia.

2.1.3 Materiales y herramientas

Los materiales y herramientas que se utilizaron para el desarrollo se especifican en la tabla 6.

Tabla 6*Materiales y herramientas utilizados para el desarrollo*

Material / Herramienta	Costo estimado (\$)
Hardware	
Laptop.	900.00.
Dron DJI Mavic 2 Pro.	1900.00.
Servidor cloud.	100.00.
Software	
Office 365.	0.00.
Visual Studio Code.	0.00.
Adobe Photoshop.	25.00.
Roboflow.	0.00.
Kaggle.	0.00.
Otros	
Internet.	150.00.
Trasporte.	30.00.
Total	3105.00

Nota. Elaboración propia.

En la tabla 7 se detallan todas las características del dron DJI Mavic 2 Pro.

Tabla 7*Especificaciones técnicas del dron DJI Mavic 2 Pro*

Especificación	Detalles
Sensor	1" Effective Pixels: 20 million CMOS
Lens	FOV: about 77° 35 mm Format Equivalent: 28 mm Aperture: f/2.8–f/11 Shooting Range: 1 m to ∞

ISO Range	Video: 100-6400 Photo: 100-3200 100-12800 (manual)	(auto)
Shutter Speed	Electronic Shutter: 8–1/8000s	
Still Image Size	5472×3648	
Still Photography Modes	Single Burst shooting: 3/5 Auto Exposure Bracketing (AEB): 3/5 bracketed frames at 0.7 EV Bias Interval (JPEG: 2/3/5/7/10/15/20/30/60s RAW:5/7/10/15/20/30/60s)	shot frames
Video Resolution	4K: 3840×2160 2.7K: 2688×1512 FHD: 1920×1080	24/25/30p 24/25/30/48/50/60p 24/25/30/48/50/60/120p
Max Video Bitrate	100Mbps	
Color Mode	Dlog-M (10bit), support HDR video (HLG 10bit)	
Supported System	File FAT32 (≤ 32 GB) exFAT (> 32 GB)	GB)
Photo Format	JPEG / DNG (RAW)	
Video Format	MP4 / MOV (MPEG-4 AVC/H.264, HEVC/H.265)	

Nota. (DJI, 2018).

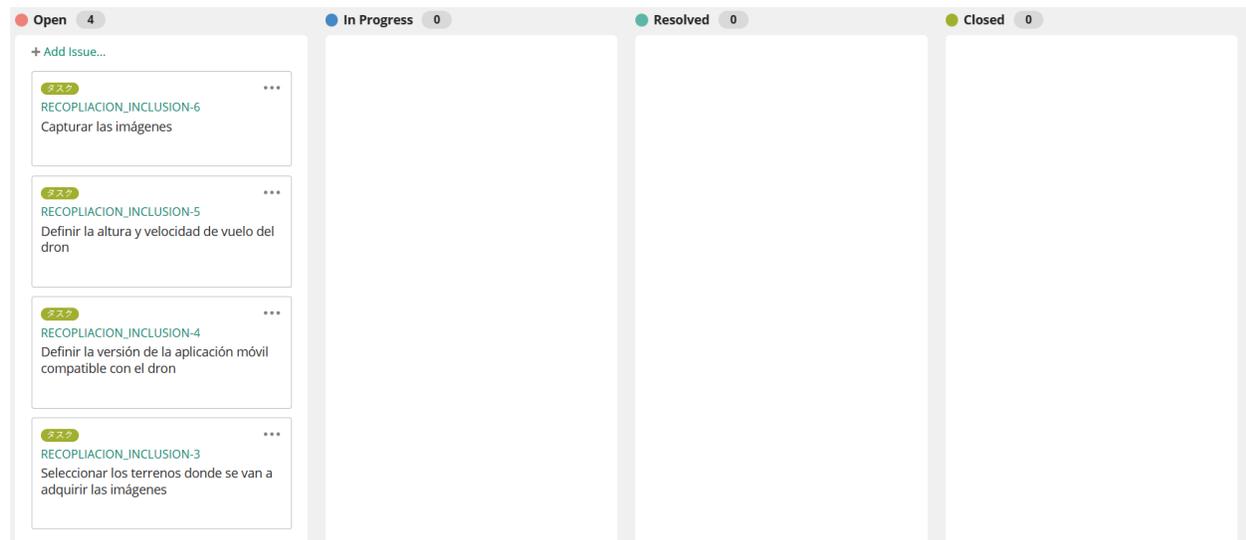
2.2 Recopilación e inclusión de datos

La primera etapa definida en la metodología KDD es la recopilación e inclusión de datos, como se mencionó anteriormente se hizo uso de Kanban para cada etapa de KDD, de tal manera que facilite la realización de las actividades.

En la figura 14 se listan todas las actividades definidas para esta etapa haciendo uso del tablero Kanban.

Figura 14

Tablero Kanban para la etapa de recopilación e inclusión de imágenes



Nota. Elaboración propia.

La primera actividad que se realizó fue seleccionar los terrenos para la adquisición de las imágenes. Al inicio se tuvo planeado utilizar los terrenos que conforman la granja La Pradera de la Universidad Técnica del Norte, sin embargo, en el tiempo planificado para la captura de imágenes dichos terrenos no tenían cultivos de maíz o papa, que son los dos tipos escogidos para este estudio, por lo tanto, como plan alternativo se decidió optar por terrenos ubicados en la provincia del Carchi de cultivos de papa.

Se planificó visitar terrenos en la zona de Cuesaca, San Gabriel y el Ángel. Específicamente se eligieron 7 terrenos, 2 ubicados en San Gabriel y 1 en Cuesaca y 4 en el Ángel.

La segunda actividad que se realizó fue la definición de la versión de la aplicación móvil compatible con el dron, actualmente existen muchas versiones para los drones de la marca DJI, sin embargo, no todas las versiones son compatibles con todos los modelos de drones, por lo que se hizo una verificación de las compatibles, al final se decidió elegir la DroneDeploy 5.7.

DroneDeploy 5.7 ofrece muchas configuraciones especiales que las otras no disponen, entre las cuales está la programación del recorrido para la zona geográfica, esto es de utilidad

porque el dron puede volar de forma automática por el terreno sin necesidad de manejarlo manualmente, además permite establecer la velocidad de vuelo, la altura y ráfaga de captura (DroneDeploy, 2024).

La tercera actividad realizada fue la definición de la altura y velocidad de vuelo del dron, para poder determinar estas dos variables se tomó muy en cuenta todos los trabajos de investigación con respecto a la detección de malezas utilizando imágenes adquiridas por un dron vistos en el capítulo 1, las alturas más comunes en la mayoría de los trabajos son 5 y 10 metros.

Se optó por la altura de 9 metros debido a que se tiene que tomar menos fotografías para cubrir la misma área en comparación a la de 5 metros, además fue en la que mejor se podría hacer una distinción de los tipos de plantas. Con respecto a la velocidad se eligió la de 1m/s. En la tabla 8 se detalla los parámetros utilizados para la toma de las imágenes.

Tabla 8

Parámetros del dron elegidos para la captura de imágenes

Parámetro	Valor
Altura de vuelo.	9 metros.
Velocidad de vuelo.	1 metro/segundo.
Resolución de imagen.	5472px por 3648px.
Relación px/cm.	4 px/cm.

Nota. Elaboración propia.

La última actividad planificada para esta fase es la captura de imágenes, como se mencionó anteriormente se eligieron 7 terrenos para la toma de imágenes, a continuación, en la tabla 9 se especifican las fechas en las que se capturaron las imágenes como también la ubicación de los terrenos.

Tabla 9

Fechas en las que se capturaron las imágenes en los terrenos elegidos

Terreno y ubicación	Fecha
Terreno 1 - San Gabriel, comunidad de Chutan Bajo, 77°51'03''W (Propiedad del padre del Ing. Marco Pusdá).	20/05/2023.
	28/05/2023.
	04/06/2023.
Terreno 2 - San Gabriel, 77°48'403''W (Propiedad anónima).	04/06/2023.
Terreno 3 – Cuesaca, Avenida de la Prehistoria, 77°52'40''W (Propiedad del tío del estudiante Kevin Vinueza).	17/06/2023.
Terreno 4 - El Ángel, barrio San Vicente, 0°38'21''N (Propiedad anónima).	16/12/2023.
Terreno 5 - El Ángel, barrio San Vicente, 77°55'59''W (Propiedad anónima).	16/12/2023.
Terreno 6 - El Ángel, barrio San Vicente, 77°55'59''W (Propiedad anónima).	16-12-2023.
Terreno 7 - El Ángel, barrio San Vicente, 0°38'13''N (Propiedad anónima).	16-12-2023.

Nota. Elaboración propia.

Para realizar la adquisición de las imágenes se hizo uso de un cuadrado de 50cm por 50 cm de color rojo, su función es ayudar a determinar la relación cm/px de las imágenes, aunque DroneDeploy ya ofrece ese dato, además de ser como una base para el vuelo y aterrizaje del dron. En la figura 15 se muestra cómo se preparó el dron para que emprenda el vuelo.

Figura 15

Preparación del dron para que emprenda vuelo

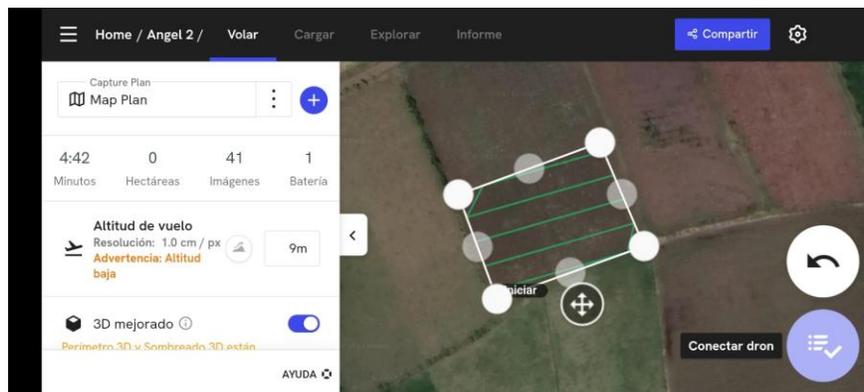


Nota. Elaboración propia.

Como se mencionó anteriormente se hizo uso de la aplicación DroneDeploy 5.7, para programar el recorrido que debe hacer el dron, velocidad de vuelo y altura. Primeramente, se conecta el mando vía cable usb tipo c al dispositivo Android que tenga instalado la aplicación, en este caso fue un celular Xiaomi Note 10 2022, se establece la comunicación con el dron y se configura los parámetros antes mencionados. En la figura 16 se indica cual es la interfaz de usuario para la configuración de los parámetros.

Figura 16

Configuración de los parámetros de DroneDeploy 5.7



Nota. Elaboración propia.

En la tabla 10 se hace un resumen del número de imágenes capturadas en las fechas que se visitó los 3 terrenos.

Tabla 10

Número de imágenes de 5472px por 3648px capturadas por fechas en los terrenos

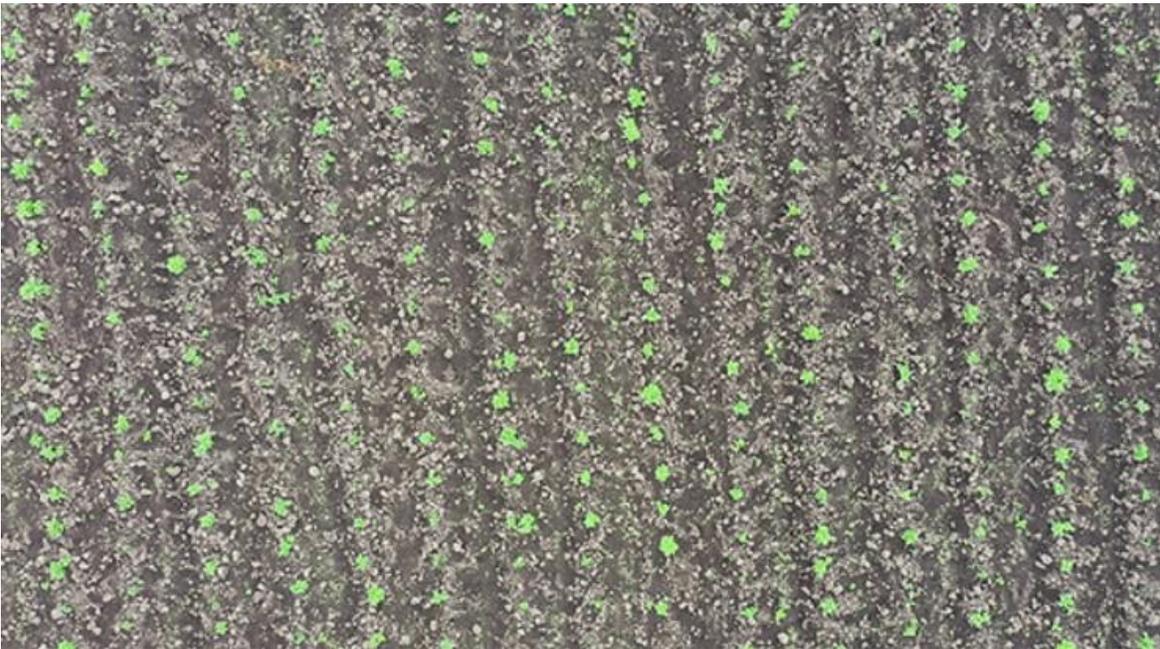
Terreno	Fecha y hora	Número de imágenes (5472px por 3648px)
Terreno 1 - San Gabriel, comunidad de Chutan Bajo, 77°51'03"W (Propiedad del padre del Ing. Marco Pusdá).	20/05/2023, 9:10 - 9:55.	539.
	28/05/2023, 10:07 - 10:42.	409.
	04/06/2023, 11:25 - 12:38.	926.
Terreno 2 - San Gabriel, 77°48'403"W (Propiedad anónima).	04/06/2023, 14:12 - 14:20.	197.
Terreno 3 – Cuesaca, Avenida de la Prehistoria, 77°52'40"W (Propiedad del tío del estudiante Kevin Vinueza).	17/06/2023, 8:58 - 9:20.	209.
Terreno 4 - El Ángel, barrio San Vicente, 0°38'21"N (Propiedad anónima).	16/12/2023, 10:23 - 10:28.	61.
Terreno 5 - El Ángel, barrio San Vicente, 77°55'59"W (Propiedad anónima).	16/12/2023, 10:52 - 10:55.	28.
Terreno 6 - El Ángel, barrio San Vicente, 77°55'59"W (Propiedad anónima).	16-12-2023, 11:18 - 11:25.	53.
Terreno 7 - El Ángel, barrio San Vicente, 0°38'13"N (Propiedad anónima).	16-12-2023, 11:51 - 11:55.	25.
Total.		2447.

Nota. Elaboración propia.

En las figuras 17, 18, 19, 20, 21, 22, 23, 24 y 25 se muestra una imagen capturada por cada terreno, en el caso del terreno 1 se muestra por cada fecha visitada.

Figura 17

Imagen capturada por el dron en el terreno 1 en la fecha 20/05/2023



Nota. Elaboración propia.

Figura 18

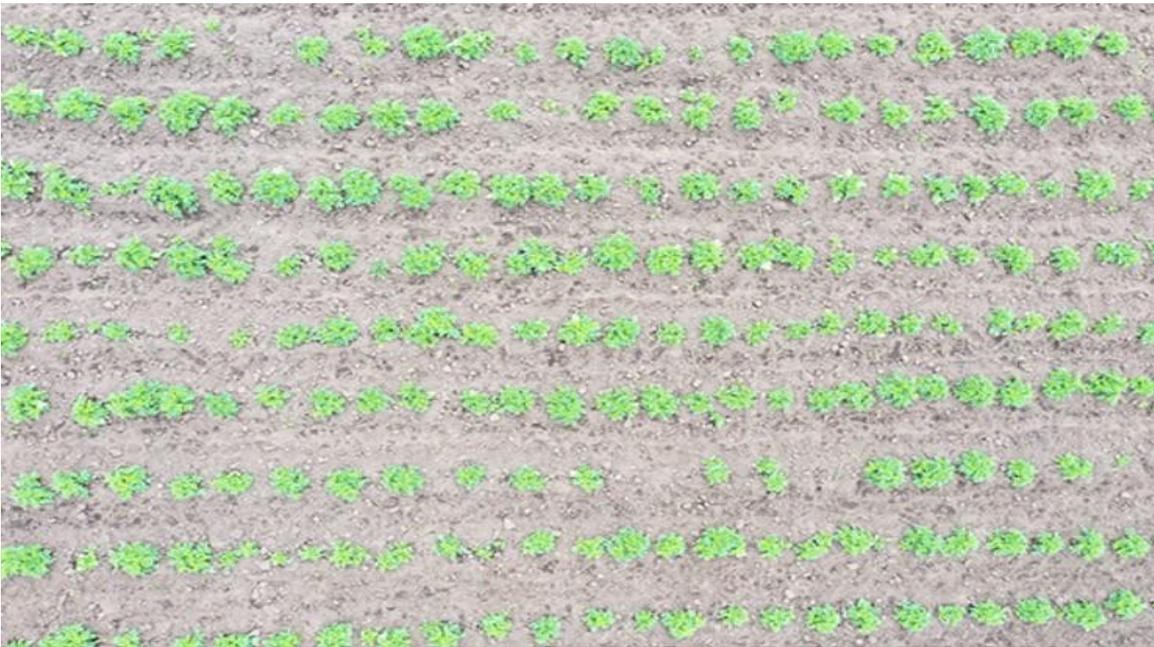
Imagen capturada por el dron en el terreno 1 en la fecha 28/05/2023



Nota. Elaboración propia.

Figura 19

Imagen capturada por el dron en el terreno 1 en la fecha 04/06/2023



Nota. Elaboración propia.

Figura 20

Imagen capturada por el dron en el terreno 2 en la fecha 04/06/2023



Nota. Elaboración propia.

Figura 21

Imagen capturada por el dron en el terreno 3 en la fecha 04/06/2023



Nota. Elaboración propia.

Figura 22

Imagen capturada por el dron en el terreno 4 en la fecha 16-12-2023



Nota. Elaboración propia.

Figura 23

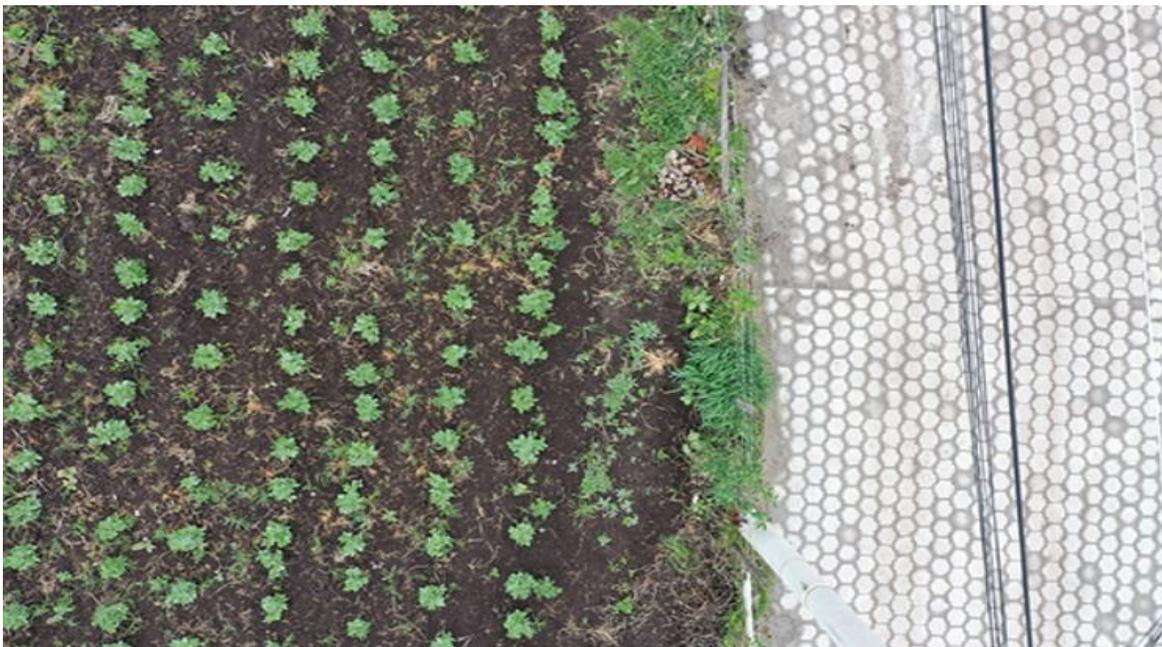
Imagen capturada por el dron en el terreno 5 en la fecha 16-12-2023



Nota. Elaboración propia.

Figura 24

Imagen capturada por el dron en el terreno 6 en la fecha 16-12-2023



Nota. Elaboración propia.

Figura 25

Imagen capturada por el dron en el terreno 7 en la fecha 16-12-2023



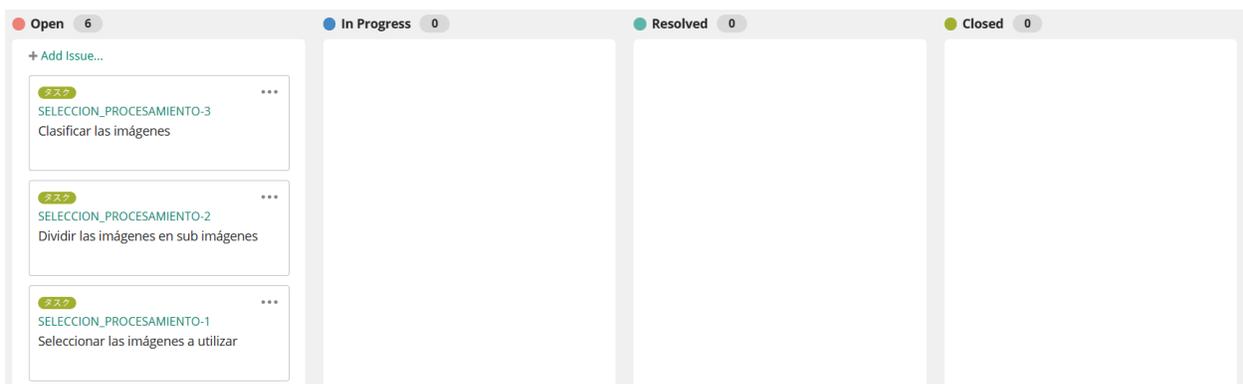
Nota. Elaboración propia.

2.3 Selección, preprocesamiento y transformación de datos

La segunda etapa de la metodología KDD es la selección, procesamiento y transformación de datos, a continuación, en la figura 26 se indica cual es el tablero Kanban para esta etapa.

Figura 26

Tablero Kanban para la etapa de selección, procesamiento y transformación de los datos



Nota. Elaboración propia.

En total se capturaron 2447 imágenes durante toda la fase de recolección de datos, para esta fase de la metodología KDD la primera actividad planificada fue la selección de las imágenes a utilizar, el criterio tomado en cuenta fue que solo se iban a utilizar imágenes que tengan presencia de maleza significativa y que se pueda hacer una distinción clara de las especies.

Las imágenes capturadas en el terreno 1 lamentablemente no cuentan con presencia significativa de malezas tendiendo a ser nula, por lo que fueron descartadas, en el terreno 3 había abundancia de malezas, sin embargo, no se podía hacer una distinción clara de las especies presentes en el terreno, ya que al estar superpuestas y tamaño pequeño hacía que su categorización sea demasiado complicada incluso para los agricultores, por esta razón también se decidió desechar todas las imágenes de este terreno.

Descartando las imágenes antes mencionadas en total quedaron 364, de estas se eliminaron aquellas que en su nomenclatura no eran múltiplos de 3, esto está ligado a la velocidad y altura que se voló el dron, que fue de 1m/s y 9 metros respectivamente, cada 3 imágenes se capturaban zonas distintas del terreno, por lo que se eliminó las restantes, después de realizar este proceso quedaron un total de 121.

De las 121 imágenes se descartaron aquellas en las cuales no se podía hacer una distinción clara del tipo de malezas existentes debido a problemas de desenfoque, luego de esto quedaron 108 imágenes utilizables. En la tabla 11 se hace un resumen de las imágenes resultantes después de aplicar cada criterio de descarte explicado en los anteriores párrafos.

Tabla 11

Número de imágenes de 5472px por 3648px útiles resultantes

Criterio de eliminación	Número de imágenes utilizables resultantes (5472px por 3648px)
Eliminación de todas las imágenes adquiridas en el terreno 1 y 3.	364.
Eliminación de aquellas que en su nomenclatura no eran múltiplos de 3.	121.

Eliminación de aquellas que presentaban desenfoque.	108.
Número de imágenes final.	108.

Nota. Elaboración propia.

En la tabla 12 se detallan el número de imágenes utilizables de todos los terrenos visitados.

Tabla 12

Imágenes de 5472px por 3648px utilizables de todos los terrenos

Terreno	Número de imágenes capturadas (5472px por 3648px)	Número de imágenes utilizables (5472px por 3648px)
Terreno 1 - San Gabriel, comunidad de Chutan Bajo, 77°51'03"W (Propiedad del padre del Ing. Marco Pusdá).	539.	0.
	409.	0.
	926.	0.
Terreno 2 - San Gabriel, 77°48'403"W (Propiedad anónima).	197.	55.
Terreno 3 - Cuesaca, Avenida de la Prehistoria, 77°52'40"W (Propiedad del tío del estudiante Kevin Vinueza).	209.	0.
Terreno 4 - El Ángel, barrio San Vicente, 0°38'21"N (Propiedad anónima).	61.	19.
Terreno 5 - El Ángel, barrio San Vicente, 77°55'59"W (Propiedad anónima).	28.	9.
Terreno 6 - El Ángel, barrio San Vicente, 77°55'59"W (Propiedad anónima).	53.	17.
Terreno 7 - El Ángel, barrio San Vicente, 0°38'13"N (Propiedad anónima).	25.	8.
Total.	2447.	108.

Nota. Elaboración propia.

ResNext50 es una arquitectura considerablemente profunda por lo que necesita una imagen de entrada de dimensiones pequeñas, se planificó que funcione con imágenes de 128 x 128 píxeles en formato RGB, por esta razón se optó por extraer pequeñas imágenes de las dimensiones especificadas a partir de las 108 imágenes de 5472px por 3648px.

Para la tarea de extraer las pequeñas imágenes se optó por utilizar el software Adobe Photoshop 2020 que es de dominio para el autor de este trabajo de titulación.

Previamente se hizo un estudio de las resoluciones de imágenes que se estaban utilizando en otros trabajos de investigación con respecto a la detección de malezas utilizando imágenes adquiridas por un dron, existen muchas propuestas, sin embargo, en el trabajo de T et al. (2019) utilizaron la dimensión de 250 x 250 píxeles, dicha resolución se adapta al tamaño de las plantas de papa y maleza, es decir que cubre toda el área conformada por una planta.

Cabe recalcar que las dimensiones de las imágenes admitidas por ResNext50 para este trabajo de titulación en particular son de 128 x 128 píxeles, por lo que después de extraer una imagen de 250 x 250 píxeles se debe hacer una redimensión, una ventaja de extraer imágenes de 250 x 250 píxeles es que al momento de realizar la redimensión no se pierde mucha información, aproximadamente por 4 píxeles cuadrados se obtiene 1 píxel cuadrado.

El proceso de extracción de las imágenes se la realizó de forma manual, Adobe Photoshop provee de múltiples herramientas, entre las cuales se utilizó la “Herramienta de rectángulo”, esta permite crear rectángulos o cuadrados (Adobe, 2024), se configuro la dimensión de 250 x 250 píxeles que da como resultado un cuadrado.

Con el cuadrado establecido se fue posicionando manualmente en cada una de las plantas de tal manera que cubra la mayor parte de dicha planta. A continuación, en la figura 27 se muestra como fue este proceso.

Figura 27

Extracción de pequeñas imágenes de 250px por 250px utilizando Adobe Photoshop



Nota. Elaboración propia.

La imagen resultante tiene la dimensión de 250 x 250 píxeles, posteriormente se redimensionó a 128 x 128 píxeles utilizando nuevamente Adobe Photoshop, esto debido a que al ser un software de edición robusto y con bastante tiempo en el mercado usa algoritmos optimizados que mantienen gran calidad en la imagen.

Como se mencionó anteriormente después de hacer la selección de todas las imágenes adquiridas por el dron en total se obtuvieron 108 imágenes en 4k utilizables, durante la extracción se dio prioridad a las plantas que sean cubiertas casi en su totalidad por el cuadrado de 250 x 250 píxeles y también se optó por elegir las plantas que tengan diferencias significativas con las demás.

En la figura 28 se muestra una imagen de 128 x 128 píxeles resultado del redimensionamiento de las de 250 x 250 píxeles, la cual está lista para ser anotada.

Figura 28

Imagen de 128px por 128px lista para ser anotada



Nota. Elaboración propia.

A continuación, en la tabla 13 se especifica cuáles son los tipos de plantas que tienen presencia significativa en las imágenes para la anotación del data set.

Tabla 13

Especies de plantas presentes en las imágenes

Planta	Nombre científico	Tipo de hoja
Lengua de vaca.	Rumex crispus L.	Ancha.
Diente de león.	Taraxacum officinale.	Ancha.
Kikuyo.	Pennisetum clandestinum.	Delgada.
Papa.	Solanum tuberosum.	Ancha.

Nota. Elaboración propia.

A continuación, en las figuras 29, 30, 31 y 32 se muestran varios ejemplos de cada tipo de plantas mencionadas en la tabla 13.

Figura 29

Ejemplos de plantas de papa



Nota. Elaboración propia.

Figura 30

Ejemplos de plantas de lengua de vaca



Nota. Elaboración propia.

Figura 31

Ejemplos de plantas de kikuyo



Nota. Elaboración propia.

Figura 32

Ejemplos de plantas de diente de león



Nota. Elaboración propia.

Evidentemente en los terrenos existen muchos más tipos de especies, sin embargo, las plantas mencionadas anteriormente tienen una presencia muy significativa en las imágenes, es decir, en casi todo el data set se encuentran las mencionadas anteriormente.

Las otras especies fueron catalogadas como “otras”, por varios motivos; uno es la poca presencia, hay casos en los que una especie está presente en una parte muy concreta de un solo terreno, ofreciendo un número muy pequeño de individuos, si se etiquetan con su categoría correspondiente existiría un desbalance de clases significativo y esto provocaría que en el entrenamiento la red no pueda aprender correctamente. Otra razón es su difícil distinción ya sea por poca visibilidad, desconocimiento de la especie o superposición con otras plantas.

Cabe recalcar que en cada una de las 108 imágenes no se encuentran todos los tipos de plantas, la razón es porque son de distintos terrenos, por esta cuestión se hizo un vistazo rápido de las imágenes utilizables para determinar cuántas imágenes pequeñas se pueden extraer de cada categoría de planta. A continuación, en la tabla 14 se detalla el número aproximado de pequeñas imágenes extraídas de cada terreno.

Tabla 14

Aproximación de sub imágenes de 250px por 250px extraídas por cada terreno

Terreno	Total de imágenes (5472px por 3648px) de los terrenos	Lengua de vaca	Diente de león	Kikuyo	Papa	Otras
Terreno 2.	55.	De 55 imágenes sub imágenes aproximadamente por cada una.	No hay presencia.	No hay presencia.	De 55 imágenes sub imágenes aproximadamente por cada una.	De 8 imágenes sub imagen por cada una.
Terreno 4.	19.	No hay presencia.	No hay presencia.	De 19 imágenes sub	No se extrajo	De 7 imágenes sub

				imágenes aproximadamente por cada una.	aisladamente.	imagen por cada una.
Terreno 5.	9.	No hay presencia.	De 3 imágenes sub imágenes aproximadamente.	De 19 imágenes sub imágenes aproximadamente por cada una.	No se extrajo aisladamente.	De 2 imágenes sub imágenes por cada una.
Terreno 6.	17.	No hay presencia.	De 10 imágenes sub imágenes aproximadamente.	De 19 imágenes sub imágenes aproximadamente por cada una.	No se extrajo aisladamente.	De 8 imágenes sub imágenes por cada una.
Terreno 7.	8.	De 2 imágenes sub imágenes aproximadamente por cada una.	De 4 imágenes sub imágenes aproximadamente.	De 19 imágenes sub imágenes aproximadamente por cada una.	No se extrajo aisladamente.	De 3 imágenes sub imagen por cada una.

Nota. Elaboración propia.

La tarea de clasificación de las imágenes fue llevada a cabo en paralelo con la tarea de dividir las imágenes en sub imágenes, una vez extraída y redimensionada se procedía a clasificarla.

En la tabla 15 se detalla el número de imágenes extraídas y clasificadas listas para la anotación.

Tabla 15

Número de imágenes de 250px por 250px resultantes después de la etapa de clasificación y extracción

Categoría de la planta	Número de imágenes (250px por 250px)
Lengua de vaca.	500.
Diente de león.	160.
Kikuyo.	300.
Otras.	40.
Papa.	500.
Total.	1500.

Nota. Elaboración propia.

Cabe hacer una importante aclaración, las imágenes finales para la fase de anotación detalladas en la tabla 15 no están conformadas únicamente por una sola categoría de planta, durante la extracción se hizo mucho énfasis en elegir las zonas que estén densamente pobladas, todo esto con el fin de anotar imágenes que se asemejen a un entorno real, ya que las plantas se superponen unas con otras o están muy juntas.

Otra razón para extraer sub imágenes que contengan más de una categoría es para aumentar la diversidad, todo esto con el fin de que la red neuronal convolucional tenga más ejemplares que aprender. A continuación, en la figura 33 se muestra una sub imagen que contiene más de una especie.

Figura 33

Sub imagen de 128px por 128px que contiene más de una especie de planta



Nota. Elaboración propia.

En la tabla 16 se muestra cual es la cantidad de imágenes en las que se encuentran presentes cada una de las categorías de plantas en las imágenes del data set.

Tabla 16

Cantidad de imágenes de 128px por 128px en las que se encuentran presentes los tipos de plantas

Tipo de planta	Número de imágenes con respecto a todo el data set (128px por 128px)	Porcentaje con respecto a todo el data set (1500 imágenes de 128px por 128px)
Lengua de vaca.	524.	35%.
Diente de león.	209.	13%.
Kikuyo.	540.	36%.
Otras.	574.	38%.
Papa.	607.	40%.

Nota. Elaboración propia.

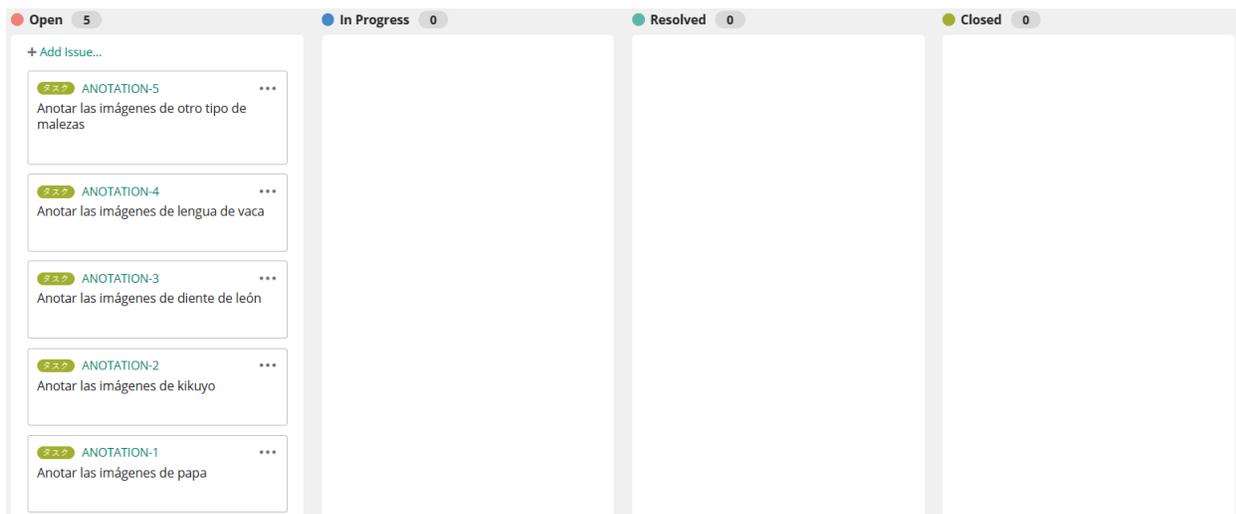
2.3.1. Anotación del data set

La anotación del data set forma parte de la etapa de selección, preprocesamiento y transformación de los datos, sin embargo, tiene una importancia muy significativa, por lo que

para describir como fue el proceso llevado a cabo se creó esta sección. En la figura 34 se visualiza el tablero Kanban con las tareas correspondientes para esta sección.

Figura 34

Tablero Kanban para la anotación del data set



Nota. Elaboración propia.

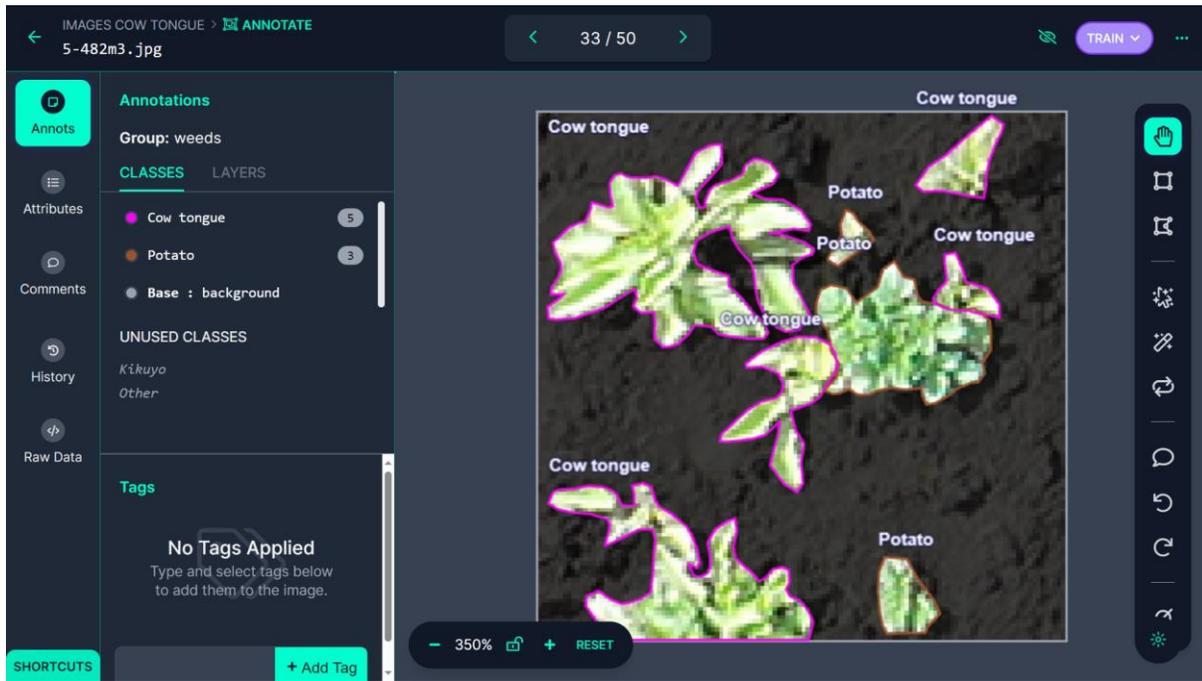
Para llevar a cabo todas las tareas de anotación se hizo uso de la plataforma Roboflow, esta es muy popular, de fácil aprendizaje y uso, además ofrece múltiples enfoques de anotación de imágenes desde bounding box hasta segmentación semántica (Roboflow, 2024), esta última es el enfoque para este proyecto.

Por cada tipo de planta se creó un proyecto individual dentro de la plataforma para facilitar la anotación de las imágenes y también para tener una distribución equitativa con respecto a la división de las imágenes en conjunto de entrenamiento y validación, Roboflow ofrece herramientas que facilitan la etiquetación de las imágenes, algunas de forma manual y otras haciendo uso de algoritmos de inteligencia artificial para la detección de objetos, se optó hacerlo de forma manual para mejorar la confiabilidad de la anotación.

A continuación, en la figura 35 se muestra la interfaz gráfica que proporciona Roboflow para la anotación de imágenes.

Figura 35

Anotación de imágenes utilizando Roboflow para segmentación semántica



Nota. Elaboración propia.

Como se sabe la segmentación semántica consiste en determinar la categoría a la que pertenece un píxel, a continuación, en la tabla 17 se especifica que valor numérico Roboflow asignó a las distintas categorías de plantas.

Tabla 17

Valores numéricos de los píxeles correspondientes a cada tipo de planta

Tipo de planta	Valor numérico
Background.	0.
Lengua de vaca.	1.
Diente de león.	2.
Kikuyo.	3.
Otra.	4.
Papa.	5.

Nota. Elaboración propia.

En la tabla 16 se especifica el número de imágenes en el que se encuentran los distintos tipos de plantas identificadas, este enfoque puede ayudar a tener una perspectiva global de la distribución de las clases en el data set, sin embargo, Roboflow ofrece la opción de contabilizar el número de individuos por cada categoría, esta media es la más idónea para determinar la distribución de cada especie en todo el data set, por lo tanto, en la tabla 18 se detalla el número de individuos por cada categoría y su porcentaje de población con respecto al número de individuos totales.

Tabla 18

Cantidad de individuos por cada tipo de planta y su porcentaje de representación con respecto a los individuos totales

Tipo de planta	Cantidad de individuos	Porcentaje de representación con respecto a los individuos totales (6774)
Lengua de vaca.	978.	14.43%.
Diente de león.	321.	4.73%.
Kikuyo.	2670.	39.41%.
Otras.	997.	14.71%.
Papa.	1808.	26.69%.
Total.	6774.	100%.

Nota. Elaboración propia.

Una vez finalizada la anotación de las imágenes se procedió a exportarlas como también con sus respectivas máscaras de segmentación semántica. Se optó por separar las imágenes en conjunto de entrenamiento y validación con un 80% y 20% respectivamente. A continuación, en la tabla 19 se detalla el número de imágenes correspondientes a cada conjunto de datos.

Tabla 19*Número de imágenes del conjunto de entrenamiento y validación*

Tipo de planta	Conjunto de entrenamiento 80%	Conjuntos de validación 20%
Lengua de vaca.	400.	100.
Diente de león.	128.	32.
Kikuyo.	240.	60.
Otra.	32.	8.
Papa.	400.	100.
Total.	1200.	300.

Nota. Elaboración propia.

Un punto a tomar en cuenta es que en el periodo que fue llevado a cabo este proyecto las bibliotecas de Tensorflow para la generación de datos dan por finalizado el soporte y ya no son compatibles con la versión más reciente de este framework, por esta razón se decidió hacer un aumento de datos utilizando opencv y python antes de realizar el entrenamiento. A continuación, en la tabla 20 se detalla los tipos de aumentos de datos elegidos con la respectiva cantidad de imágenes resultantes del conjunto de entrenamiento.

Tabla 20*Tipos de aumentos de datos aplicados*

Tipo de aumento	Cantidad de imágenes obtenidas (128px por 128px)
Sin aumento (Imágenes originales).	1200.
Aumento del brillo 25%.	1200.
Disminución del brillo 25%.	1200.
Desenfoco gaussiano con un kernel de 5x5.	1200.
Rotación 90 grados sentido horario con aumento del brillo 25%.	1200.

Rotación 180 grados sentido horario con disminución del brillo 25%.	1200.
Rotación 270 grados sentido horario.	1200.
Total.	8400.

Nota. Elaboración propia.

2.4 Minería de datos

La siguiente etapa de la metodología KDD es la minería de datos, la cual consiste en utilizar algoritmos y métodos matemáticos los cuales permitan analizar los datos y extraer patrones y tendencias, esta información es útil para la toma de decisiones.

Para el contexto de este trabajo esta etapa se divide en 2 secciones, la primera tiene que ver con la programación del algoritmo de deep learning y la segunda con el entrenamiento, cada sección es representada como una actividad en el tablero Kanban, a continuación, en la figura 36 se muestra cuáles son las actividades para esta fase.

Figura 36

Tablero Kanban para la minería de datos



Nota. Elaboración propia.

2.4.1 Definición del algoritmo de Deep Learning

El algoritmo elegido es ResNeXt50, este fue seleccionado por el tutor del proyecto el PhD. Iván García.

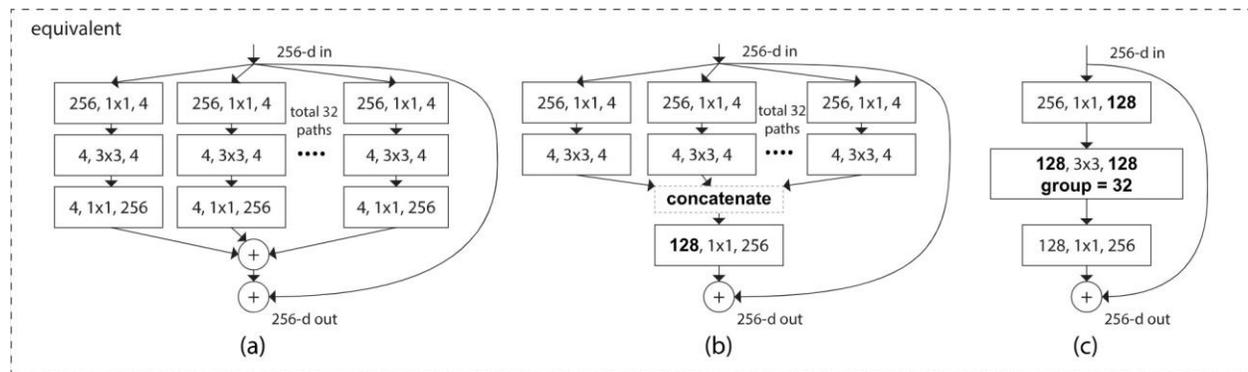
ResNext50 es una variante de ResNet50 que se le aplica el concepto de cardinalidad de Inception, según sus autores ofrece mejor rendimiento que su versión base (Xie et al., 2017),

esto evidentemente es muy útil en la tarea de identificación de malezas debido a la complejidad y dificultad que esta tarea tiene.

La arquitectura ResNeXt puede tener 3 tipos de estructuras equivalentes, en la figura 37 se detalla cuáles son.

Figura 37

Equivalencias de la arquitectura ResNeXt



Nota. (Xie et al., 2017).

Para este trabajo se optó por elegir la segunda forma descrita en la figura 37 con cardinalidad de 2, debido a la simplicidad que resulta programarla al igual que eficiencia de memoria y potencia computacional. Los autores también definieron en su trabajo el número de filtros para cada valor de cardinalidad, a continuación, en la tabla 38 se detalla cuáles son los valores para cada una.

Tabla 38

Valores para las cardinalidades de ResNeXt

Cardinality C	1	2	4	8	32
Width of bottleneck d.	64	40	24	14	4
Width of group conv.	64	80	96	112	128

Nota. (Xie et al., 2017).

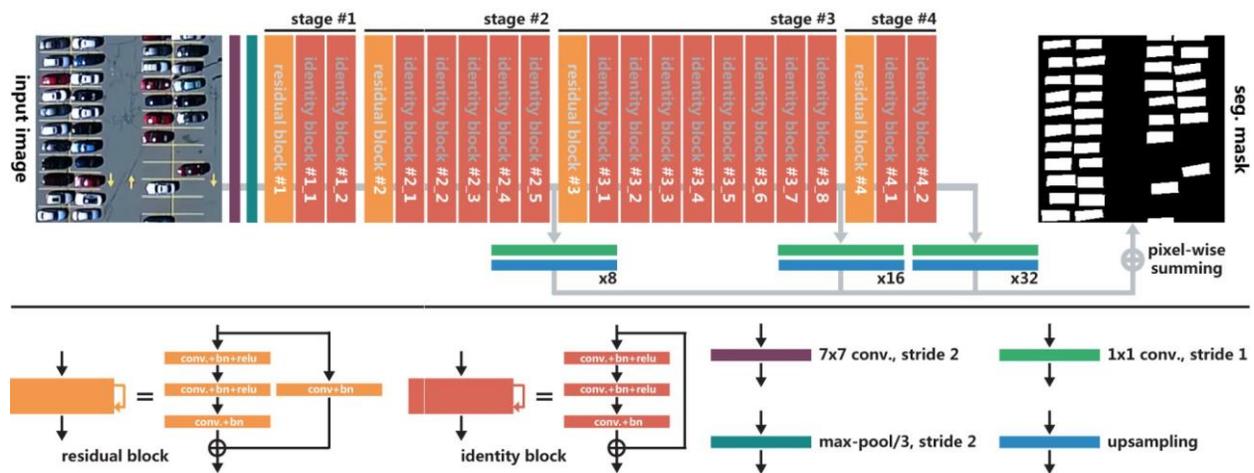
El tema de este trabajo es la identificación y cuantificación de malezas, para realizar la cuantificación se optó por utilizar el método de cobertura, el cual necesita el número de píxeles

total de la imagen y el número de píxeles que conforman las plantas a cuantificar, por lo que se decidió realizar una modificación a la arquitectura ResNeXt para que en lugar de enfocarse a la clasificación se enfoque a la segmentación semántica para obtener dicha cantidad de píxeles.

Se tomó en cuenta la modificación propuesta por Mou & Zhu (2018), a continuación, en la figura 39 se muestra cual es la propuesta del autor.

Figura 39

Modificación a la arquitectura ResNet la segmentación semántica



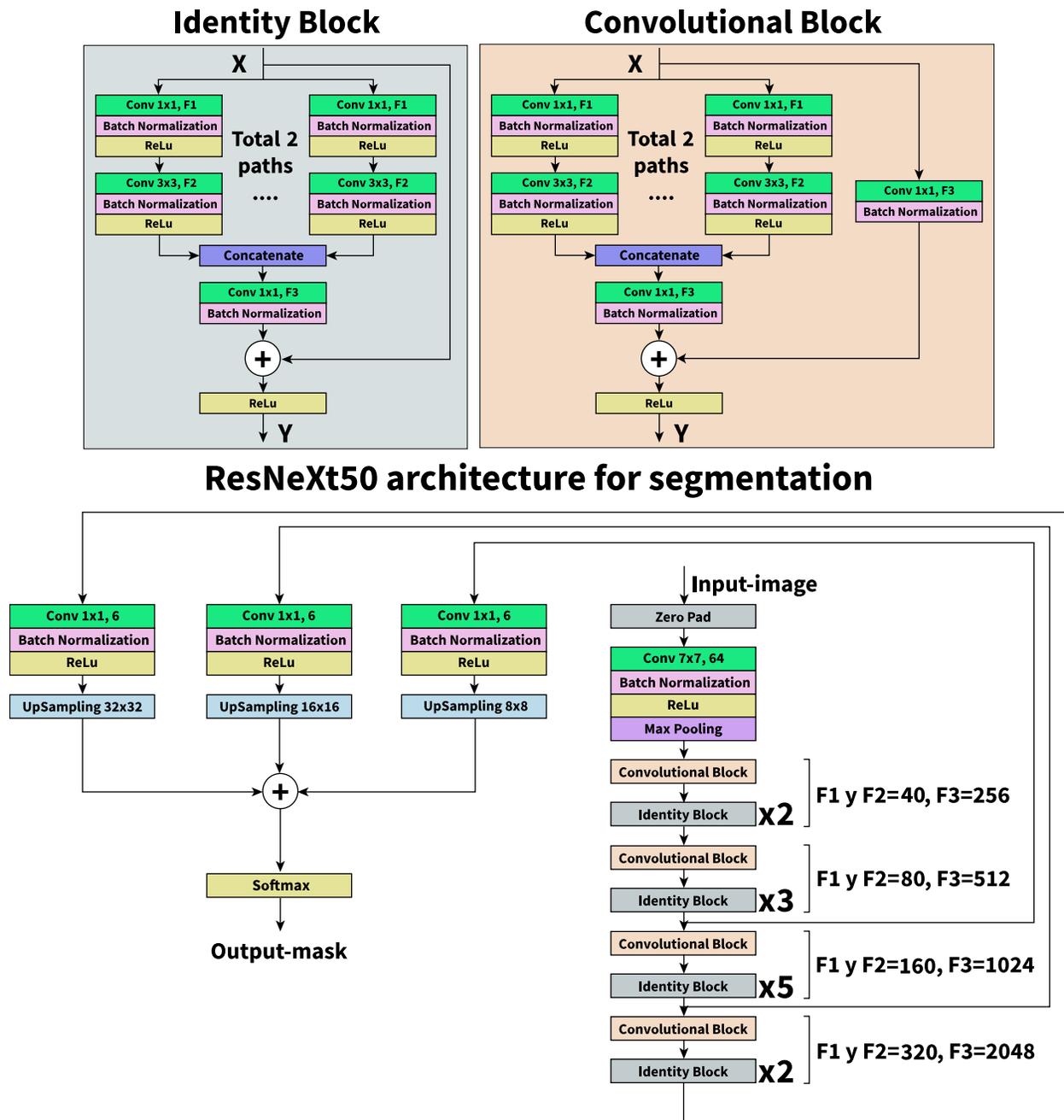
Nota. (Mou & Zhu, 2018).

Las librerías utilizadas para la programación fueron Tensorflow y Keras. Keras ofrece muchas versiones y tipos de algoritmos de machine y deep learning compiladas a disposición del usuario, sin embargo, para este proyecto se optó por programarla desde cero debido a las modificaciones explicadas anteriormente para que pueda realizar la segmentación semántica.

A continuación, en la figura 40 se muestra cual fue la arquitectura programada para ResNeXt50 enfocada en la segmentación semántica.

Figura 40

Arquitectura de ResNeXt50 para segmentación semántica



Nota. Elaboración propia.

2.4.2 Entrenamiento

Se hizo una búsqueda de trabajos de investigación acerca de la detección de malezas mediante segmentación semántica, a continuación, en la tabla 22 se resumen los hiperparámetros utilizados por los autores.

Tabla 22

Propuestas de autores para la definición de los hiperparámetros de entrenamiento

Autores	Tamaño de lote	Número de épocas	Tasa de aprendizaje	Optimizador	Distribución del data set
(X. Li et al., 2023)	2.	100.	5e-04.	Adam.	50% entrenamiento, 38% validación y 12% pruebas.
(X.-Z. Hu et al., 2022)	No especificado.	200.	3e-4.	Adam.	80% para entrenamiento y 20% para validación.
(Lottes, Behley, Milioto, et al., 2018)	10.	200.	0,01.	RMSPROP.	No especificado
(Weyler et al., 2023)	4.	4096.	0,0001.	Adam.	70% entrenamiento, 15% validación y 15% pruebas.
(Sa et al., 2018)	6.	640.	0,001.	SGD.	80% entrenamiento y 20% validación.
(Gonçalves et al., 2023)	8.	20.	0,0001.	Adam.	90% entrenamiento y 10% validación.
(H. S. Ullah et al., 2021)	No especificado.	200.	0,0001.	Adam.	70% entrenamiento, 15% validación

y 15% pruebas.

Nota. Elaboración propia.

Las imágenes se mantuvieron en formato RGB para que ResNeXt50 pueda hacer una mejor discriminación de las distintas clases de plantas ya que las tonalidades varían entre cada planta.

Para definir el ground truth se optó por utilizar la codificación one-hot, esta ofrece la ventaja de que permite determinar una categoría asignándole el valor de 1 y a las otras 0, al momento de interpretar las predicciones se identificaría el índice que contenga el valor más alto del array y ese índice determinaría la categoría a la que pertenece un píxel. A continuación, en la tabla 23 se especifica cuál es la codificación one-hot para las categorías.

Tabla 23

Codificación one-hot para las categorías

Categoría	Codificación one-hot a nivel de píxel – Valor del píxel en la máscara
Background.	1 0 0 0 0 0 – 0.
Lengua de vaca.	0 1 0 0 0 0 – 1.
Diente de león.	0 0 1 0 0 0 – 2.
Kikuyo.	0 0 0 1 0 0 – 3.
Otra.	0 0 0 0 1 0 – 4.
Papa.	0 0 0 0 0 1 – 5.

Nota. Elaboración propia.

El entorno seleccionado para el entrenamiento fue la plataforma Kaggle, esta ofrece una versión gratuita que proporciona tanto CPU, GPU o TPU, lo cual es beneficioso para el entrenamiento ya que el tiempo en llevar a cabo este proceso será menor en comparación a si se eligiera entorno la computadora del estudiante. A continuación, en la tabla 24 se detalla las características de hardware elegidas.

Tabla 24*Características del entorno de entrenamiento*

Componente hardware	Detalle
CPU.	Multihilo, frecuencia y cantidad de núcleos no especificado.
RAM.	29 GB.
GPU.	P100.
VRAM.	16 GB.
Almacenamiento.	73 GB HDD.

Nota. Elaboración propia.

Para este trabajo de titulación se decidió hacer múltiples mejoras al modelo base con el fin de obtener mejores valores en las métricas, de tal manera que al final se elija el modelo más idóneo para implementarlo en la aplicación web. En las siguientes secciones se detalla cual fue el proceso para cada una de las versiones de mejora.

Base line

El modelo base line es el que se especifica en la figura 39, este modelo inicial es el resultado de la implementación de la propuesta realizada por Mou & Zhu (2018) para segmentación semántica en la arquitectura ResNeXt50.

En la tabla 22 se hizo un resumen de los valores asignados a los hiperparámetros de entrenamiento utilizados por los autores que se enfocaban a la segmentación semántica para detección de malezas, una tendencia que se pudo observar es un valor bajo en lo que respecta al tamaño del lote, basándose en los fundamentos teóricos de deep learning un tamaño de lote más pequeño conlleva a que se hagan más iteraciones en una sola época, es decir los parámetros se podrán actualizar más veces, además de hacer un menor uso de memoria RAM durante el entrenamiento.

Otra particularidad es el bajo valor del learning rate de las distintas propuestas, haciendo una analogía esto estaría estrechamente relacionado al tamaño del lote, un learning rate bajo significaría que se haga un descenso más controlado para buscar el mínimo global de la función de pérdida, sin embargo, si se hacen pocas actualizaciones durante el entrenamiento posiblemente no se llegue nunca hasta el mínimo global, por esta razón la elección de un tamaño de lote bajo significaría un mayor número de iteraciones y por lo tanto mayor probabilidad de alcanzar el mínimo global.

En la tabla 25 se detalla el valor de los hiperparámetros elegidos para el entrenamiento del modelo base line.

Tabla 25

Valores de los hiperparámetros del modelo base line

Hiperparámetros	Valor
Tamaño del lote.	8.
Número de épocas.	100.
Tasa de aprendizaje.	0.0001.
Optimizador.	Adam.

Nota. Elaboración propia.

Versión 2

En el modelo base line una particularidad que se puede observar en la arquitectura es que las capas de upsampling son de 8x8, 16x16 y 32x32, esto supone un incremento muy rápido en la resolución de los mapas de características resultantes, por lo que la máscara predicha no se acoplará perfectamente a las siluetas de las plantas y tendrá una forma muy cuadrática.

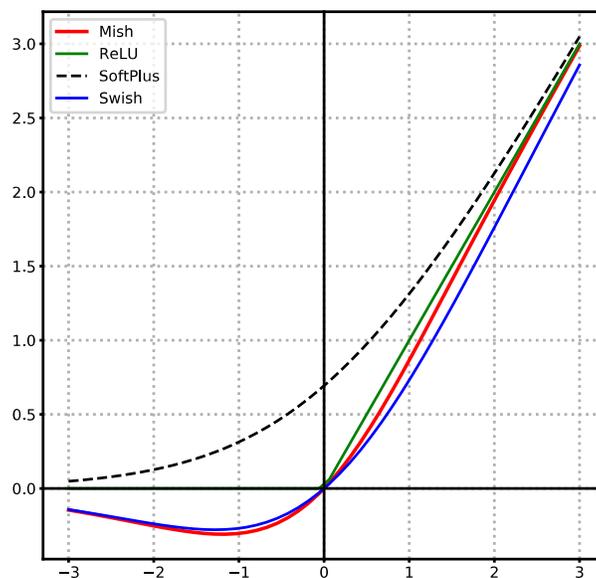
Para solventar el problema se optó por modificar la parte del decoder, la modificación importante fue hacer una redimensión de los mapas de características gradualmente en múltiplos de 2, es decir, por ejemplo, en lugar de utilizar una única capa de upsampling de 8x8 se empleó

3 capas de upsampling de 2x2, esto ayudaría a que la máscara final se acople de mejor manera a la silueta de las plantas.

Además, se optó por cambiar la función de activación, en la arquitectura original se emplea ReLU, para esta versión 2 se propuso utilizar la función de activación Mish, esta proporciona un mayor rango de valores para el eje de las ordenadas, ya que su forma le permite albergar valores negativos, por su parte ReLU está limitada a valores de $[0, \infty]$ (Misra, 2019), a continuación, en la figura 41 se muestra la forma de Mish en comparación con otras funciones de activación.

Figura 41

Función de activación Mish, ReLU, SoftPlus y Swish



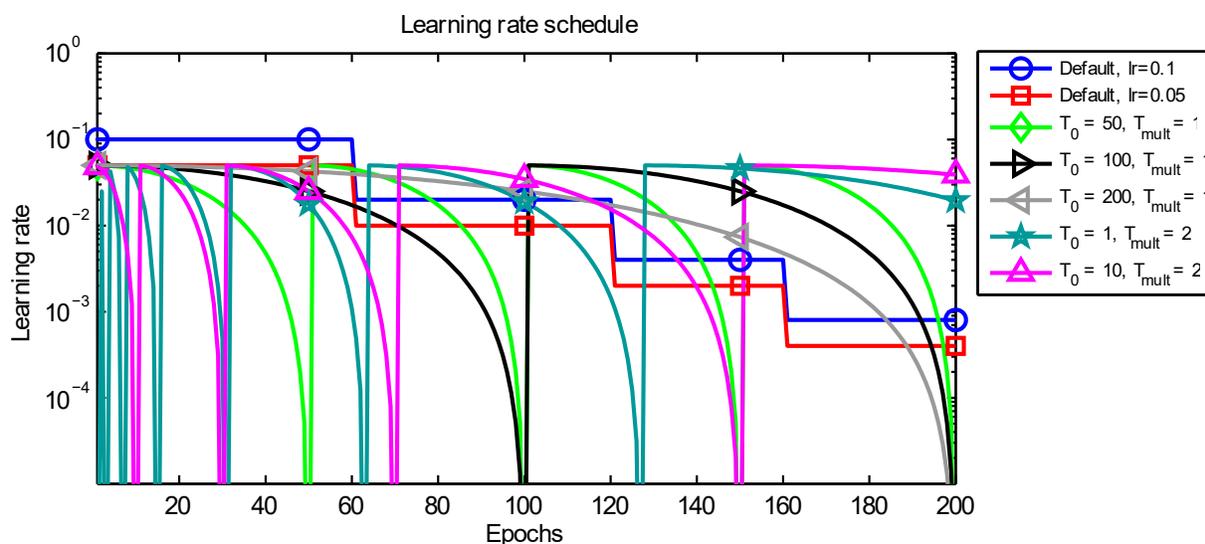
Nota. (Misra, 2019).

Otra modificación realizada con respecto al modelo base line es en el learning rate, este hiperparámetro era fijo, es decir que su valor siempre se mantenía igual durante todo el entrenamiento, está demostrado que se obtiene mejores resultados si el learning rate es variable con tendencia al decaimiento, además, implementar una función que permita que en cierto punto del entrenamiento este aumente su valor ayuda a desestancarse de mínimos locales y tener mayor probabilidad de alcanzar el mínimo global (Smith, 2015).

En la figura 42 se plasma la propuesta del paper titulado “SGDR: STOCHASTIC GRADIENT DESCENT WITH WARM RESTARTS” para la política de decaimiento del learning rate durante el entrenamiento usando cosine annealing.

Figura 42

Diferentes tipos de learning rate schedule usando cosine annealing



Nota. (Loshchilov & Hutter, 2016).

Como se puede observar en la figura 42 existen diferentes tipos de políticas, todas persiguen el mismo propósito, el cual es que el learning rate varíe durante el entrenamiento.

Sin embargo, se decidió hacer una modificación personalizada a esta política, como se puede observar el valor al inicio es alto y a media que se ejecutan las épocas este valor disminuye, sin embargo, en un punto concreto el learning rate recupera su valor inicial y empieza el ciclo nuevamente, esto supone que efectivamente se tenga mayor probabilidad de abandonar un posible mínimo local, pero aumentar drásticamente el valor supone que los parámetros se desajusten.

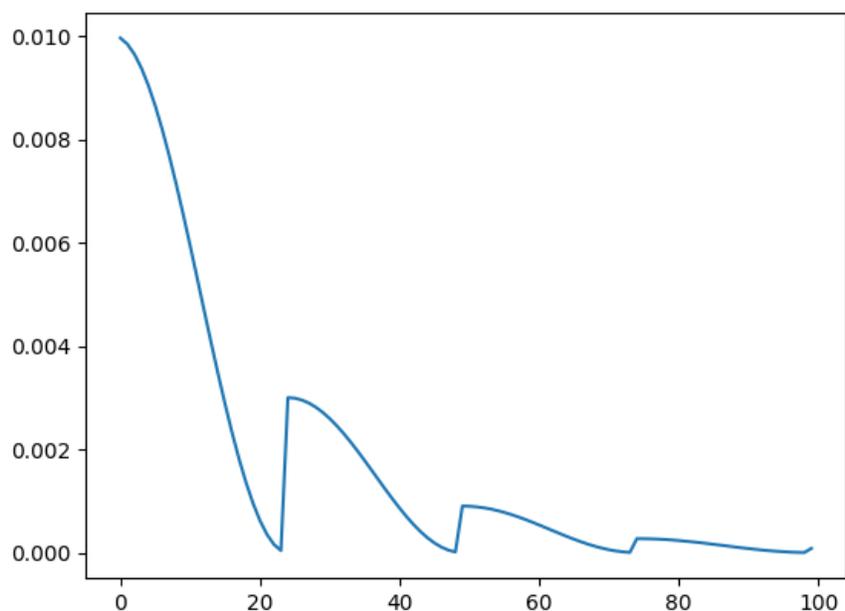
La modificación realizada es que al inicio de cada ciclo el learning rate disminuya un 70% con respecto al valor inicial del ciclo anterior, con esto al final de cada ciclo se estaría incrementando el valor del learning rate para abandonar un posible mínimo local, el learning rate

inicial del siguiente ciclo al representar el 30% del anterior haría que el desajuste de los parámetros hasta ese punto no sea muy significativo haciendo que la red no aumente significativamente su error inicialmente, posteriormente se prevé que se estabilice y siga disminuyendo su error durante el entrenamiento.

A continuación, en la figura 43 se detalla la modificación realizada a política de decaimiento del learning rate usando cosine annealing.

Figura 43

Función de decaimiento del learning rate usando cosine annealing con disminución del 70% con respecto al valor inicial del ciclo anterior

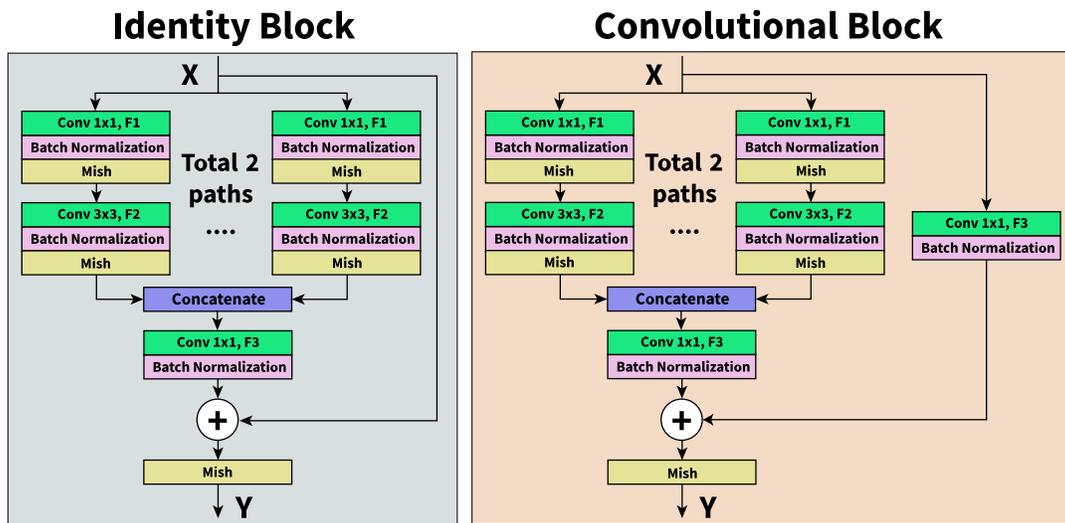


Nota. Elaboración propia.

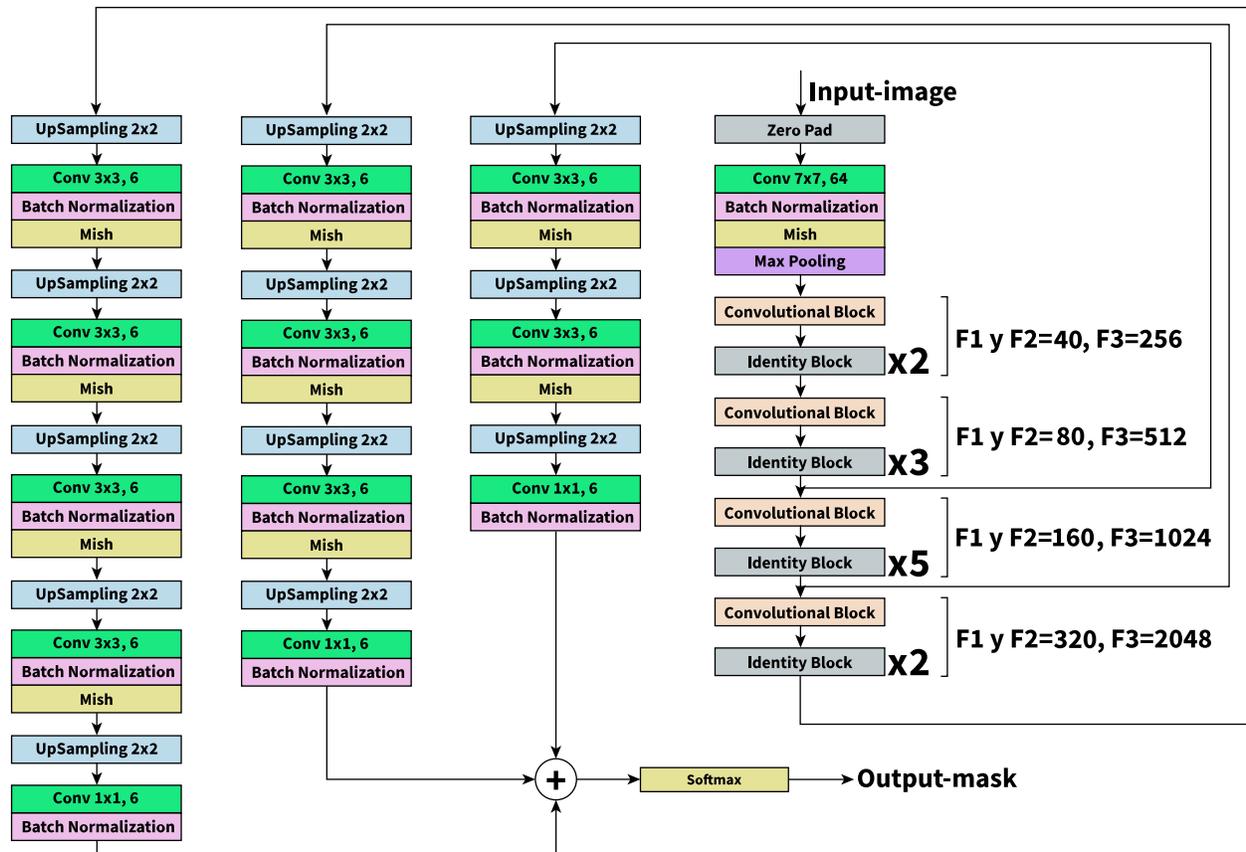
En la figura 44 se muestra la arquitectura propuesta para la versión 2.

Figura 44

Arquitectura del modelo versión 2



ResNeXt50 architecture for segmentation



Nota. Elaboración propia.

A continuación, en la tabla 26 se detalla el valor de los hiperparámetros elegidos para el entrenamiento de la versión 2.

Tabla 26

Valores de los hiperparámetros del modelo versión 2

Hiperparámetros	Valor
Tamaño de lote.	8.
Número de épocas.	100.
Tasa de aprendizaje.	0.001, utilizando la función cosine annealing personalizada con 10 ciclos y una disminución del 70% de la tasa de aprendizaje con respecto al ciclo anterior.
Optimizador.	Adam.

Nota. Elaboración propia.

Versión 3

Para definir la tercera versión del modelo se tomó como base la arquitectura de la versión 2, en la segunda arquitectura se puede observar que en la parte del decoder por cada capa de upsampling se utiliza una capa convolucional con un número de kernels igual al número de clases a predecir.

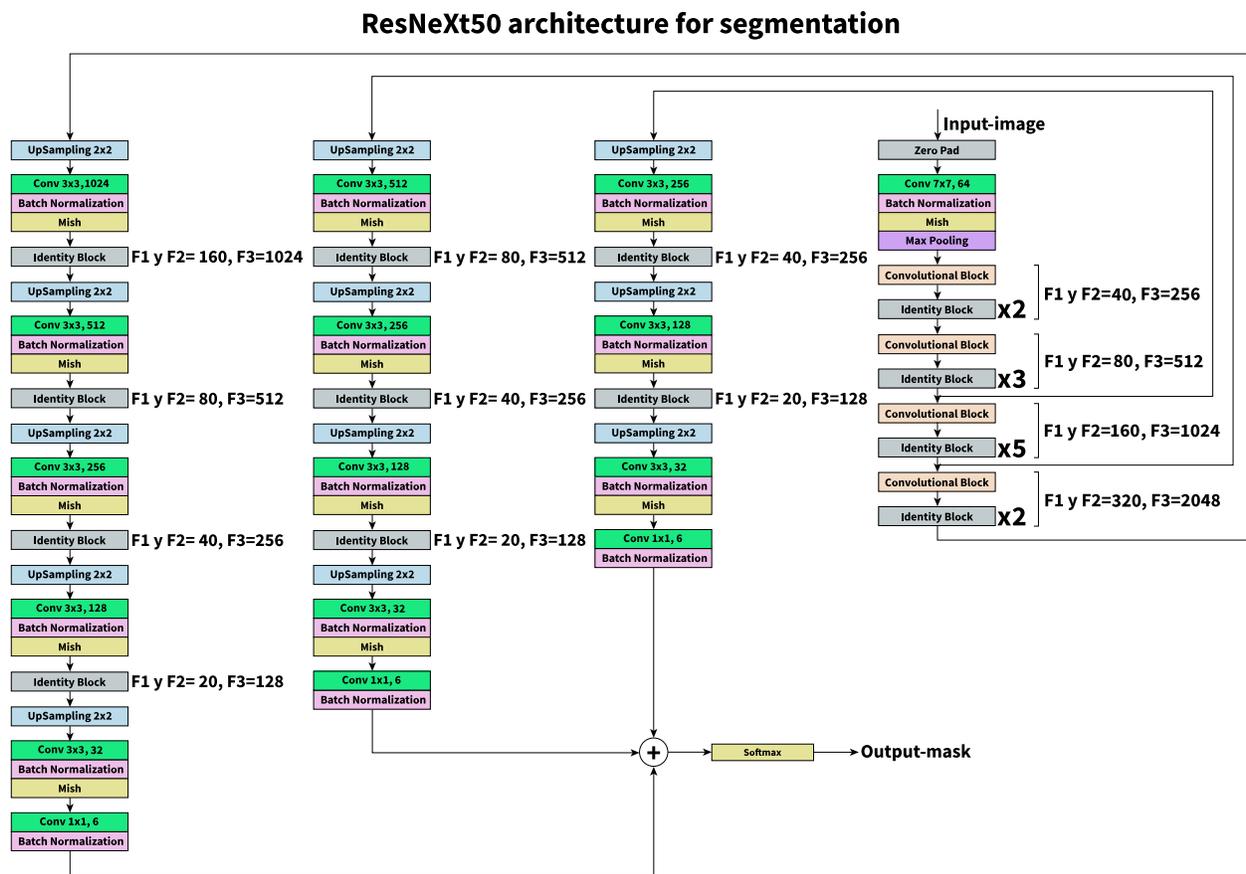
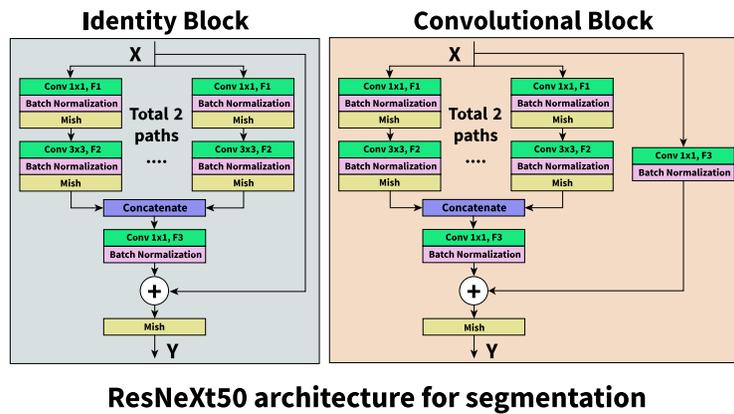
La parte del decoder es muy importante porque con esta se puede lograr obtener la máscara de segmentación semántica, si se hace más robusta esta parte la máscara predicha será más confiable.

Haciendo uso de los fundamentos de la arquitectura ResNeXt, el modelo versión 3 implementa los bloques de identidad de ResNeXt para que pueda hacer una extracción más eficiente de característica en la parte del decoder y así mejorar las predicciones.

En la figura 45 se muestra la arquitectura propuesta para la versión 3.

Figura 45

Arquitectura del modelo versión 3



Nota. Elaboración propia.

En la tabla 27 se muestra los valores de los hiperparámetros elegidos para en entrenamiento del modelo versión 3.

Tabla 27*Valores de los hiperparámetros del modelo versión 3*

Hiperparámetros	Valor
Tamaño de lote.	8.
Número de épocas.	100.
Tasa de aprendizaje.	0.001, utilizando la función cosine annealing personalizada con 10 ciclos y una disminución del 70% de la tasa de aprendizaje con respecto al ciclo anterior.
Optimizador.	Adam.

Nota. Elaboración propia.**Versión 4**

Se usó la misma arquitectura propuesta para la versión 3, en este caso ya no se hicieron modificaciones a la arquitectura debido a que ya es lo suficientemente robusta, para esta versión y las posteriores se propone cambiar los valores de los hiperparámetros de entrenamiento para encontrar la mejor combinación. En la tabla 28 se describe los hiperparámetros elegidos para el entrenamiento.

Tabla 28*Valores de los hiperparámetros del modelo versión 4*

Hiperparámetros	Valor
Tamaño de lote	16.
Número de épocas	100.
Tasa de aprendizaje.	0.00001.
Optimizador.	Adam.

Nota. Elaboración propia.**Versión 5**

La versión 5 utiliza el mismo modelo de arquitectura de la versión 3, lo único que se hace para esta propuesta es utilizar diferentes valores para los hiperparámetros.

En la tabla 29 se detallan los valores de los hiperparámetros utilizados para entrenar la versión 5.

Tabla 29

Valores de los hiperparámetros del modelo versión 5

Hiperparámetros	Valor
Tamaño de lote	32
Número de épocas	200.
Tasa de aprendizaje.	0.0001, utilizando la función cosine annealing personalizada con 10 ciclos y una disminución del 70% de la tasa de aprendizaje con respecto al ciclo anterior.
Optimizador.	Adam.

Nota. Elaboración propia.

Versión 6

El modelo de arquitectura es tomado de la versión 3, en esta fase solo se cambia el valor de los hiperparámetros. En la tabla 30 se detalla los valores de los hiperparámetros utilizados para el entrenamiento.

Tabla 30

Valores de los hiperparámetros del modelo versión 6

Hiperparámetros	Valor
Tamaño de lote	2.
Número de épocas	100.
Tasa de aprendizaje.	0.001, utilizando la función cosine annealing personalizada con 10 ciclos y una disminución del 70% de la tasa de aprendizaje con respecto al ciclo anterior.
Optimizador.	Adam.

Nota. Elaboración propia.

2.5 Evaluación e interpretación

El método de cobertura utilizado por Puerto (2018) necesita 2 valores, el primero es la cantidad de píxeles total de la imagen y el segundo es el número de píxeles del objeto de interés al que se le desea realizar la cuantificación, en este caso cada tipo de planta, a continuación, se indica cual es la ecuación propuesta:

$$cobertura = \frac{\text{Área de maleza}}{\text{Área total}} \times 100\%$$

En la figura 46 se plasma cual es la actividad planificada para esta fase de la metodología KDD.

Figura 46

Tablero Kanban para la evaluación e interpretación de patrones



Nota. Elaboración propia.

ResNeXt50 para este trabajo se enfocó en la segmentación semántica, por lo que las predicciones resultantes de la red son matrices de dimensión (128,128,6), donde, el primer y segundo valor representan el alto y ancho de la imagen (128 x 128) y el tercer parámetro es un array de 6 elementos, cada elemento representa la probabilidad de la clase a la que un píxel puede pertenecer, la sumatoria de los valores de estos elementos debe ser 1 ya que en la capa final de ResNeXt50 se utilizó la función de activación softmax.

El beneficio de haber utilizado la codificación one-hot es que cada clase es representada por el número 1 y las demás con 0 en un array, por lo que se llega a la conclusión de que el índice que contenga el valor más alto del array es el que determina a que clase pertenece un píxel.

Como se sabe la segmentación semántica consiste en que a partir de una imagen se obtenga una máscara, dicha máscara establece la categoría a la que pertenecen los píxeles, sin embargo, lo idóneo es editar la imagen original de tal manera que se evidencie cuáles son las clases detectadas, básicamente aplicar la máscara a la imagen original, el color de cada clase es el que permite distinguir las distintas categorías de plantas.

En la etapa de recolección de datos se hizo énfasis en que las imágenes capturadas tiene una resolución de 5472px por 3648px, a estas le hizo la extracción de múltiples imágenes de 250px por 250px y a estas últimas se las redimensionó a 128px por 128px para la anotación y entrenamiento. Evidentemente cuando se desee hacer la identificación y cuantificación de malezas de un terreno no se pueden tomar imágenes de resolución de 128px por 128px por parte del dron ya que no se visualizaría absolutamente nada.

La manera de solventar esta particularidad es que la imagen de 5472px por 3648px sea dividida en pequeñas imágenes de 250px por 250px y a estas últimas hacer una redimensión a 128px por 128px, tal y como se hizo en la etapa de selección, preprocesamiento y transformación de datos para que sean compatibles con la entrada de ResNeXt50.

Una forma de llegar al mismo resultado es hacer una redimensión de la imagen original y luego hacer la extracción de las sub imágenes de 128px por 128px. Para obtener las dimensiones de la imagen redimensionada se utilizó la relación 250/128, aproximadamente de un valor de 1,953, con esto ya no es necesario primero hacer la extracción de sub imágenes de 250px por 250px y luego redimensionarlas a 128px por 128px ya que este proceso se estaría haciendo directamente. A continuación, se muestran las ecuaciones utilizadas para calcular el alto y ancho de la imagen redimensionada a partir de la original.

$$\text{Alto} = \text{alto de la imagen original} \times (250/128)$$

$$\text{Ancho} = \text{ancho de la imagen original} \times (250/128)$$

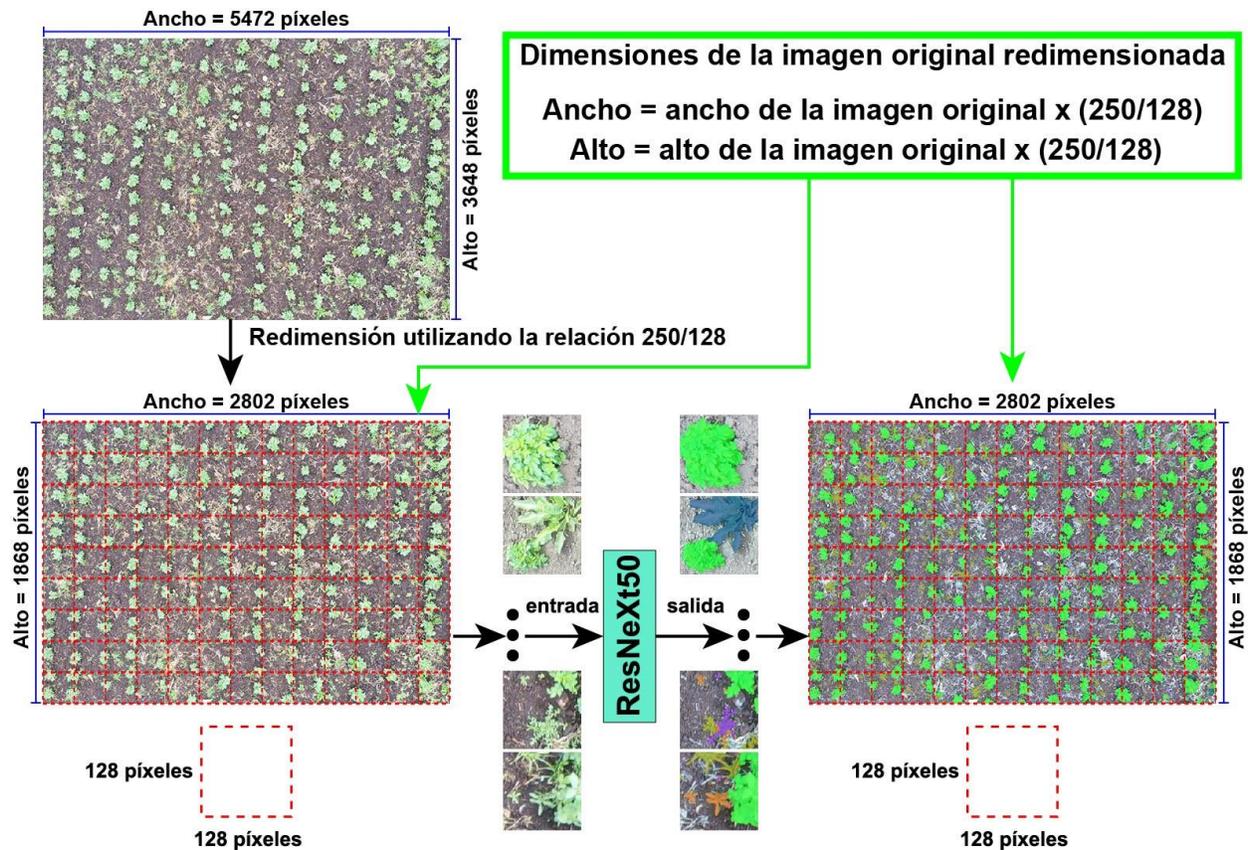
Una vez aplicado el preprocesamiento explicado anteriormente ya se puede hacer las predicciones utilizando ResNeXt50, finalizado el proceso se aplica la máscara de segmentación

semántica a la imagen original para que se pueda distinguir las categorías de plantas encontradas, sin embargo, la agrupación de las máscaras resultantes para formar una sola máscara no tendrá las mismas dimensiones de la imagen original ya que se aplicó una redimensión a esta última para poder dividir la imagen en sub imágenes y realizar las predicciones, aplicando la ecuación de cobertura explicada al inicio de esta fase de KDD da como resultado un valor porcentual, esta medida es relativa por lo que hacer una redimensión a la máscara final no supondrá un cambio en el resultado, por esta razón se decidió aplicar la máscara de segmentación a la imagen redimensionada para que sea compatible.

En la figura 47 se detalla cual es el proceso explicado anteriormente.

Figura 47

Preprocesamiento y post procesamiento automático de las imágenes para realizar predicciones



Nota. Elaboración propia.

Cuando se agrupa las máscaras predichas para obtener la máscara final se hace el conteo de los píxeles pertenecientes a cada categoría de planta y se aplica el método de cobertura para cada clase, el resultado es un valor porcentual que representa el porcentaje de área que ocupa un tipo de planta con respecto al área total de la imagen.

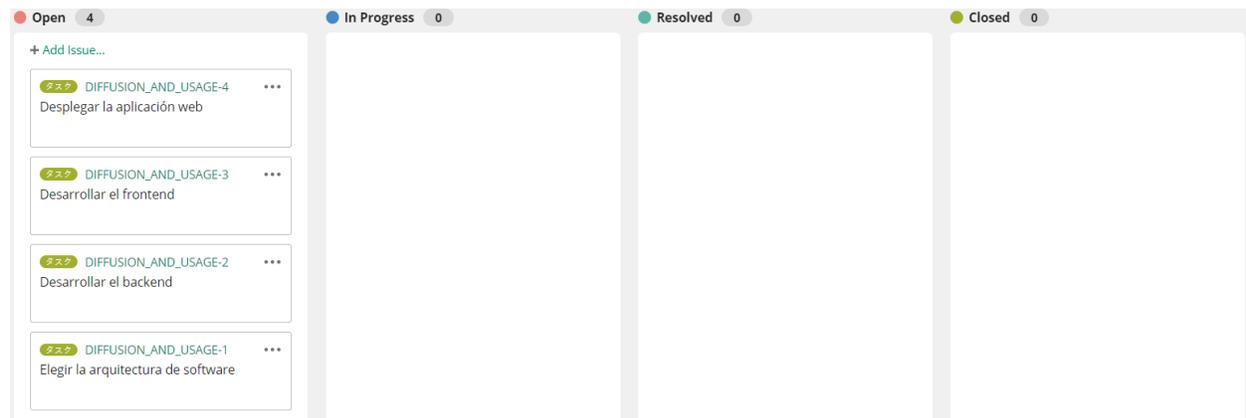
2.6 Difusión y uso

La última etapa de la metodología KDD es el uso y difusión de conocimiento, básicamente se trata de hacer uso de herramientas ya sea tablas, gráficos, informes, entre otros, que faciliten la comprensión del conocimiento adquirido a los usuarios que lo necesiten, por esta razón en la propuesta para realizar este proyecto se planteó el desarrollo de una aplicación web para la identificación y cuantificación de malezas mediante imágenes.

En la figura 48 se muestra las actividades planificadas para esta etapa haciendo uso del tablero Kanban

Figura 48

Tablero Kanban para la etapa de difusión y uso



Nota. Elaboración propia.

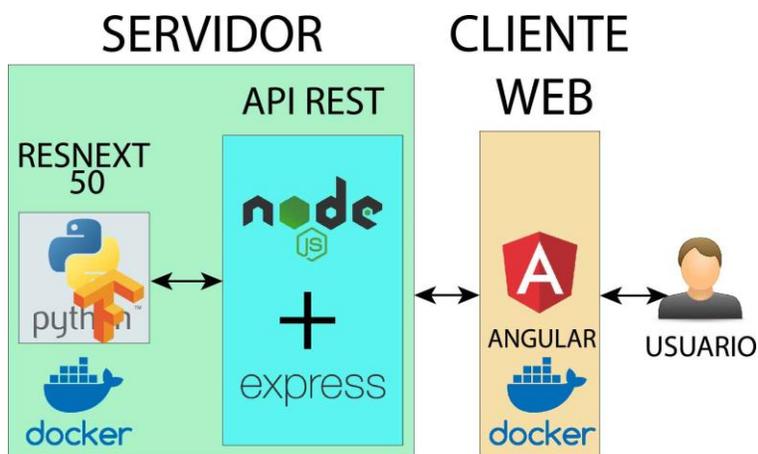
Para desarrollar la primera actividad de esta etapa se optó por elegir el modelo de arquitectura cliente-servidor debido al contexto del proyecto, ResNeXt50 necesita de un entorno que tenga muy buenas prestaciones de hardware para que pueda ejecutarse de la manera óptima, es por esa razón que el servidor debe de encargarse en su totalidad de esta tarea,

mientras que el cliente únicamente funcionaría como un medio por el cual el usuario envía las imágenes al servidor y visualizaría los resultados.

A continuación, en la figura 49 se muestra la arquitectura propuesta.

Figura 49

Arquitectura de la aplicación web



Nota. Elaboración propia.

2.6.1 Backend

El backend se compone de dos apartados fundamentales, el API REST y la red neuronal convolucional ResNeXt50, el API REST tiene como propósito recibir y enviar las imágenes al cliente angular, mientras que ResNeXt50 procesa las imágenes y obtiene los valores cuantitativos de las predicciones.

API REST

Se utilizó el entorno de nodejs junto con el framework de express para el desarrollo de la API REST. Primeramente, se tuvo que crear un middleware para que permita almacenar las imágenes en un directorio temporal, este middleware recibe las imágenes en forma de FormData, FormData es un tipo de estructura que es compatible con angular, por lo su uso es viable para este proyecto.

En el middleware se procesan las imágenes transformándolas de FormData a imágenes tal cual fueron cargadas por el usuario, es decir manteniendo su misma extensión y calidad. El directorio en donde se almacenan las imágenes contiene dos subcarpetas; en la primera se almacenan las imágenes cargadas desde la interfaz de usuario, y en la segunda se almacenan las imágenes que fueron procesadas por ResNeXt50 junto con los resultados de la cuantificación en un archivo JSON.

Nativamente no se pueden comunicar de forma directa nodejs con Python, sin embargo, nodejs ofrece una biblioteca llamada "child-process", esta permite ejecutar procesos en segundo plano, el proceso en segundo plano que se ejecuta es un archivo de python el cual contiene toda la lógica para que ResNext50 pueda realizar las predicciones de las imágenes.

Una vez obtenidos los resultados se comprimen todas las imágenes y son enviadas al cliente angular para que visualice los resultados, cuando finalizado el envío de los datos automáticamente se elimina el directorio temporal ya que no es necesario que se mantenga dicha información en el servidor.

Python y ResNeXt50

En la sección anterior se mencionó que para poder ejecutar ResNeXt50 se tuvo que programar toda la lógica en un solo archivo de python ya que la comunicación no puede ser directa con nodejs.

El archivo contiene los métodos que permiten hacer el preprocesamiento y post procesamiento de las imágenes explicado en la etapa de difusión y uso, al igual que calcular los valores de la cuantificación.

2.6.2 Frontend

Para realizar la identificación y cuantificación de malezas el usuario únicamente necesita una interfaz que le permita cargar las imágenes del cultivo, previsualizarlas, solicitar el proceso de identificación y cuantificación y visualizar los resultados, por esta razón se optó por

implementar todas estas funcionalidades en un único componente y utilizando una única plantilla HTML.

Haciendo una recapitulación rápida del apartado del backend se había mencionado que el formato para realizar el envío de las imágenes era FormData, cuando el usuario carga las imágenes a través del formulario y selecciona la opción de predecir, las imágenes se transforman a formato FormData y posteriormente se hace el envío al backend.

Como se sabe la aplicación web no va a ser utilizada por un único usuario a la vez, sino que pueden existir muchos simultáneamente, por esta razón antes de realizar el envío se calcula un número aleatorio de 10 dígitos el cual servirá como id de la transacción realizada, este id serviría tanto para el envío de los datos como para la recepción de los resultados, el identificador es enviado en la cabecera de la solicitud.

La recepción de los resultados es automática, cuando el backend termina de realizar las predicciones empaqueta los resultados y los envía al cliente angular, en este caso el formato que recibe el frontend es un archivo comprimido, para lo cual se programaron métodos que permitan descomprimir el archivo y visualizar los resultados.

En la figura 50 se visualiza la interfaz de usuario inicial.

Figura 50

Interfaz de usuario inicial



Nota. Elaboración propia.

En la figura 51 se muestra la interfaz que permite visualizar las imágenes cargadas.

Figura 51

Interfaz de usuario para visualizar las imágenes cargadas



Nota. Elaboración propia.

En la figura 52 se visualiza la pantalla de espera hasta que todas las imágenes sean procesadas.

Figura 52

Pantalla de espera

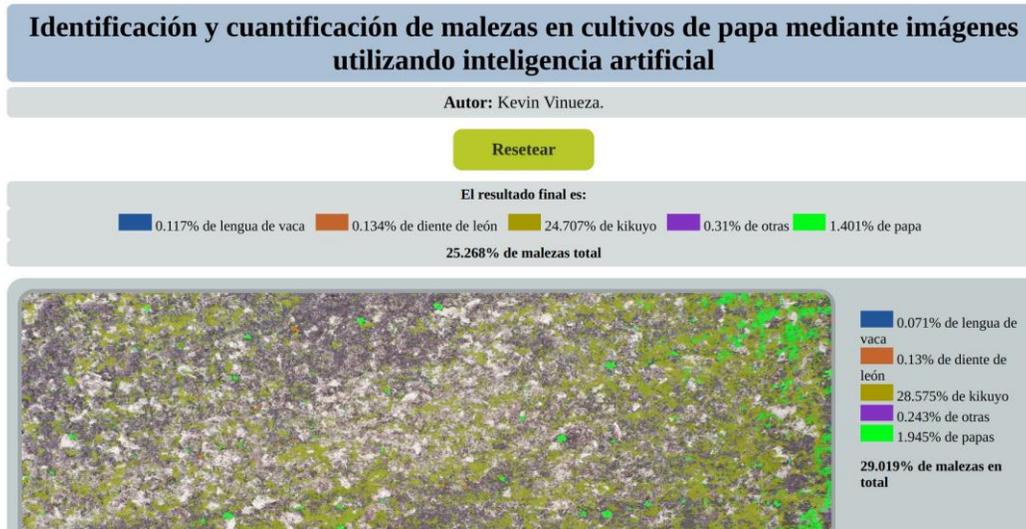


Nota. Elaboración propia.

En la figura 53 se visualiza la interfaz de visualización de los resultados.

Figura 53

Interfaz para la visualización de los resultados



Nota. Elaboración propia.

2.6.3 Implementación de contenedores Docker

Una de las propuestas para realizar este trabajo de titulación es priorizar el aspecto de la portabilidad del sistema, como se mencionó en secciones anteriores se utilizaron múltiples tecnologías para el desarrollo de la aplicación web, si se hiciera un despliegue tradicional la puesta en marcha sería muy complicada, ya que se debería hacer una configuración del entorno exhaustiva y por lo tanto el nivel de portabilidad sería muy bajo.

Por esta razón se decidió implementar Docker en la parte del backend y frontend, esta tecnología tiene la ventaja de que permite tener configurado todo el entorno en una imagen Docker, y para su puesta en marcha en otro sistema solo se debe crear un contenedor a partir de esa imagen y el sistema ya estaría listo para usar sin tener que configurar cosas adicionales (Docker, 2024a).

2.6.4 Despliegue

Para poder desplegar la aplicación web se decidió utilizar Azure, esta plataforma ofrece la compatibilidad con Docker (Microsoft, 2023), por lo que para realizar el despliegue las imágenes deben estar subidas a un repositorio para su posterior descargar.

Se optó por elegir el repositorio Docker Hub, este repositorio es gratuito y fácil de utilizar (Docker, 2024b), lo primero fue crear un proyecto individual para el backend y el frontend, luego con ayuda de la documentación oficial se procedió a subir las imágenes.

En la plataforma de Azure se crearon 2 proyectos, uno para cada imagen Docker, esta plataforma dispone de un servicio llamado “App Service”, que es la enfocada al despliegue de aplicaciones web, la configuración es bastante sencilla, lo único que se debe hacer es elegir el tipo de despliegue, la ubicación del servidor y las características de recursos de hardware a utilizar. En pocos minutos la aplicación web estaría lista para ser utilizada por los usuarios.

CAPÍTULO 3

Validación de resultados

En la propuesta de este trabajo de titulación se había definido hacer 3 tipos de validaciones, la primera tiene que ver con los resultados obtenidos de los entrenamientos de las versiones del modelo propuestas en el capítulo 2, la segunda validación es con respecto a las métricas de la norma ISO 25023 enfocadas a la característica capacidad de ser instalado, estas métricas permiten dar un criterio de que tan portable es la aplicación web, la última validación corresponde a comparar los tiempos para realizar la identificación y cuantificación de malezas de forma manual versus el tiempo empleado por la aplicación web, esto para determinar si la propuesta tecnológica solventado la problemática de este proyecto, que es la alta cantidad de tiempo empleado para realizar esta actividad.

Debido a la extensión de cada evaluación se ha decidido dividir el capítulo 3 en 3 secciones.

3.1 Validación de resultados del entrenamiento

En el capítulo 2 se habían propuesto 6 versiones de ResNeXt50, esto con el fin de probar algunas modificaciones a la arquitectura original y en la combinación de distintos valores de los hiperparámetros propuestos por autores que realizaron trabajos enfocados a la segmentación semántica de malezas, al final de esta comparativa se decidió elegir el mejor modelo para implementarlo en la aplicación web.

Las métricas elegidas para este tipo de enfoque son: dice loss, mean dice coefficient y mean IOU. H. S. Ullah et al. (2021) en su trabajo recopila las ecuaciones de cada métrica las cuales son:

$$Dice\ loss(Y_{pred}, Y_{true}) = 1 - \frac{1}{C} \sum_{c=1}^C \frac{2 \sum_i^N Y_{pred_{i,c}} Y_{true_{i,c}}}{\sum_i^N Y_{pred_{i,c}}^2 + \sum_i^N Y_{true_{i,c}}^2}$$

$$\text{Mean dice coefficient}(Y_{\text{pred}}, Y_{\text{true}}) = \frac{1}{C} \sum_{c=1}^C \frac{2 \sum_i^N Y_{\text{pred},i,c} Y_{\text{true},i,c}}{\sum_i^N Y_{\text{pred},i,c}^2 + \sum_i^N Y_{\text{true},i,c}^2}$$

$$\text{Mean IOU}(Y_{\text{pred}}, Y_{\text{true}}) = \frac{1}{C} \sum_{c=1}^C \frac{\sum_i^N Y_{\text{pred},i,c} \cap Y_{\text{true},i,c}}{\sum_i^N Y_{\text{pred},i,c} \cup Y_{\text{true},i,c}}$$

Donde:

- Y_{pred} es la máscara predicha por la capa softmax en codificación one-hot.
- Y_{true} es la máscara real, ground truth en codificación one-hot.
- C es el número de clases.
- N es el número de píxeles.
- i es el inicio del intervalo hasta N , su valor inicial es 1.

Dice loss es una función de pérdida utilizada para problemas de segmentación semántica, su propósito es evaluar el nivel de superposición y semejanza que existe entre la máscara de segmentación de la predicción y el ground truth, su rango está limitado desde $[0,1]$, si el valor se acerca a 0 significa que la semejanza de ambas máscaras es mayor (Celikkan et al., 2023).

Mean dice coefficient, es el coeficiente utilizado para calcular la función de pérdida dice, se le denomina mean porque calcula el dice coefficient promedio de todas las clases, su rango está limitado desde $[0,1]$, si el valor se acerca a 1 la semejanza entre la máscara predicha y el ground truth será mayor (H. S. Ullah et al., 2021).

Mean IOU, IOU es las siglas de Intersection over Union, su función es calcular el grado de exactitud entre la máscara predicha y el ground truth, mean es porque calcula el IOU promedio de todas las clases, su rango está limitado desde $[0,1]$, si el valor se acerca a 1 la exactitud será mayor (Qi et al., 2023).

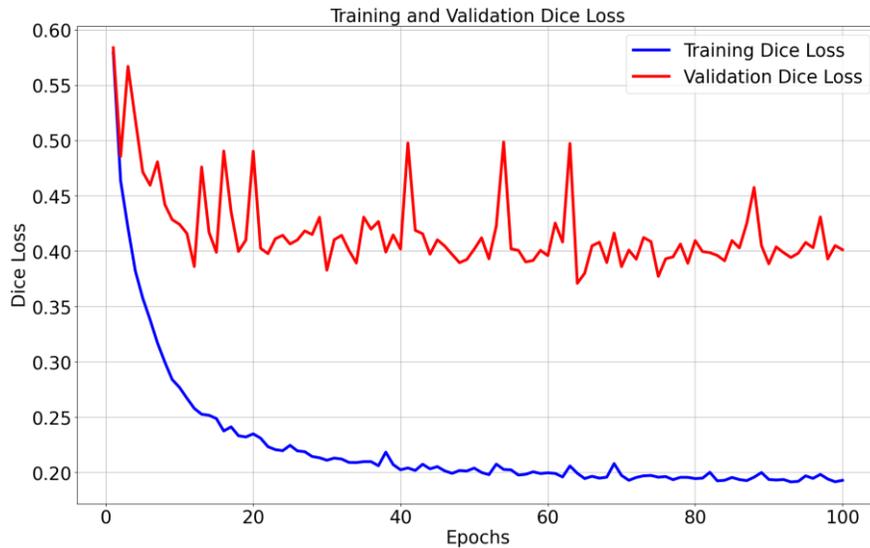
A continuación, en las siguientes secciones se da una interpretación completa de los resultados obtenidos por cada versión del modelo y al final una comparativa global a fin de elegir el mejor modelo.

3.1.1 Base line

A continuación, en las figuras 54, 55 y 56 se muestran las gráficas de dice loss, mean dice coefficient y mean IOU respectivamente de la etapa de entrenamiento y validación.

Figura 54

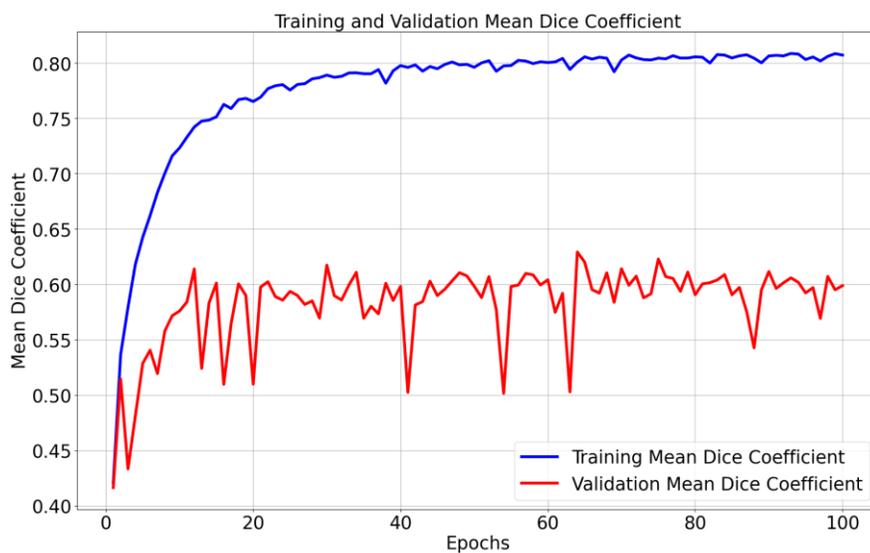
Dice loss del entrenamiento y validación del modelo base line



Nota. Elaboración propia.

Figura 55

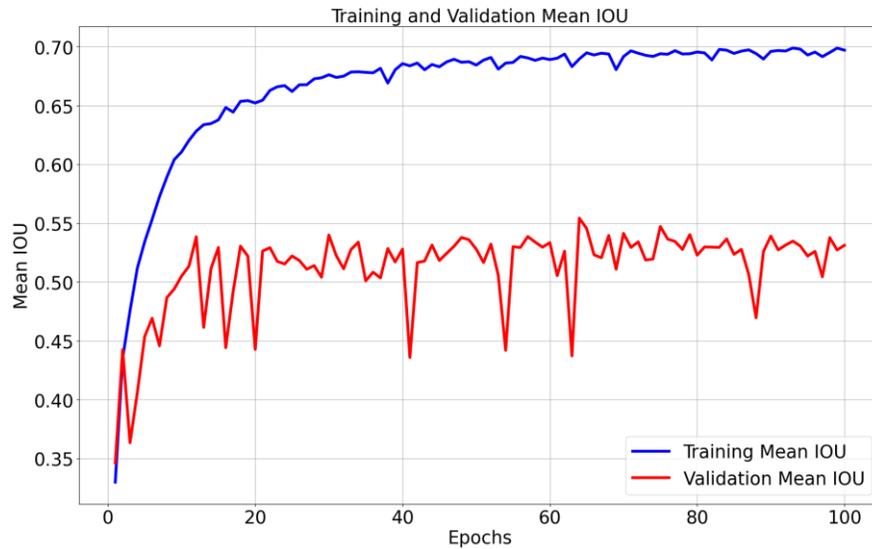
Mean dice coefficient del entrenamiento y validación del modelo base line



Nota. Elaboración propia.

Figura 56

Mean IOU del entrenamiento y validación del modelo base line



Nota. Elaboración propia.

Con respecto a las 3 métricas se puede observar que existe una notoria diferencia entre el conjunto de entrenamiento y validación, en el conjunto de entrenamiento se nota que tiene un rendimiento decente, pero en el de validación el panorama cambia, en este el rendimiento es peor, además experimenta muchos picos de subida y bajada, lo cual hace que sea muy inestable, este fenómeno se observa durante las 100 épocas, mientras que en el conjunto de entrenamiento los picos no son tan pronunciados. En la tabla 31 se detalla el valor de cada métrica alcanzado en la época 100.

Tabla 31

Valores del dice loss, mean dice coefficient y mean IOU del modelo base line en la época 100 en el entrenamiento y validación

Conjunto de entrenamiento	Dice loss (época 100)	Mean dice coefficient (época 100)	Mean IOU (época 100)
Entrenamiento.	0.192.	0.807.	0.697.
Validación.	0.401.	0.598.	0.531.

Nota. Elaboración propia.

A continuación, en la tabla 32 se detalla el IOU por cada categoría utilizando el 20% del conjunto de validación.

Tabla 32

IOU de cada categoría del modelo base line utilizando el 20% del conjunto de validación

IOU del background	IOU de lengua de vaca	IOU del diente de león	IOU del kikuyo	IOU de otro tipo de malezas	IOU de papa	Mean IOU
0.911.	0.711.	0.505.	0.536.	0.009.	0.834.	0.584.

Nota. Elaboración propia.

A continuación, en la tabla 33 se detalla el tiempo de inferencia y de entrenamiento.

Tabla 33

Tiempo de inferencia y de entrenamiento del modelo base line

Detalle (en base al entorno de entrenamiento)	Tiempo (segundos)
Tiempo de entrenamiento de una sola época.	76.
Tiempo de todo el entrenamiento (100 épocas).	7600.
Tiempo de inferencia de una sola imagen de 128px por 128px.	0.071.
Tiempo de inferencia de una imagen de 5472px por 3648px (330 sub imágenes de 128px por 128px, primero se extrajo de 250px por 250px y luego se redimensionó a 128px por 128px).	0.700.

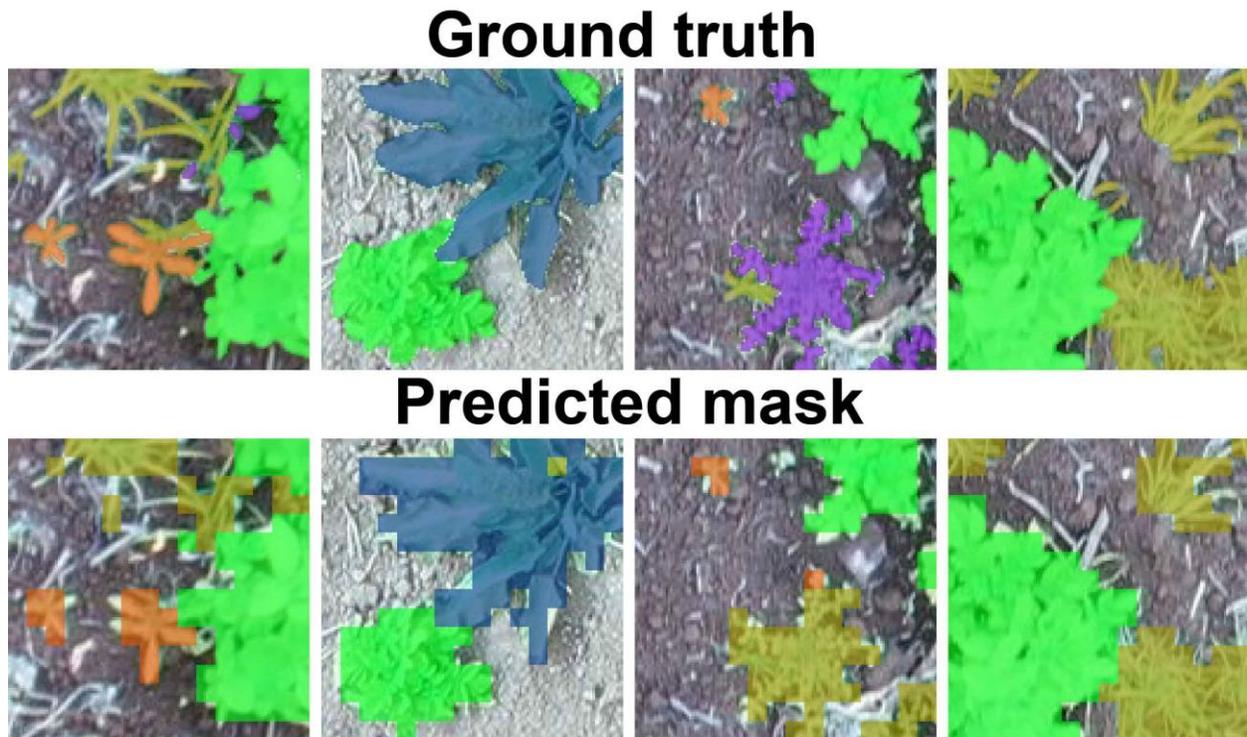
Nota. Elaboración propia.

Los resultados de las 3 métricas se deben a que el modelo base line tiene una particularidad en la parte del decoder, en esta se utilizan capas de upsampling que suben la resolución de los mapas de características demasiado rápido, por lo que el resultado es una silueta cuadrática en lugar de una que se asemeje a la forma de las plantas, las capas de upsampling utilizadas son 8x8, 16x16 y 32x32, esto supone un incremento muy acelerado en la resolución.

A continuación, en la figura 57 se muestra el ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px.

Figura 57

Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo base line



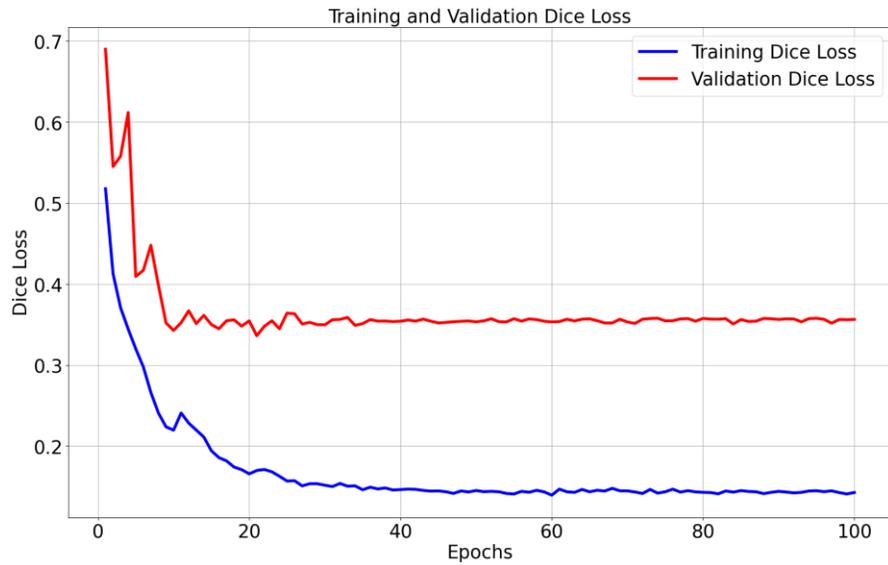
Nota. La maleza kikuyo es identificada por el color amarillo mostaza, el diente de león por el color naranja, la lengua de vaca por el color azul, el cultivo de papa por el color verde y la maleza que pertenece a la categoría otra por el color morado. Elaboración propia.

3.1.2 Versión 2

A continuación, en las figuras 58, 59 y 60 se muestran las gráficas de dice loss, mean dice coefficient y mean IOU respectivamente de la etapa de entrenamiento y validación.

Figura 58

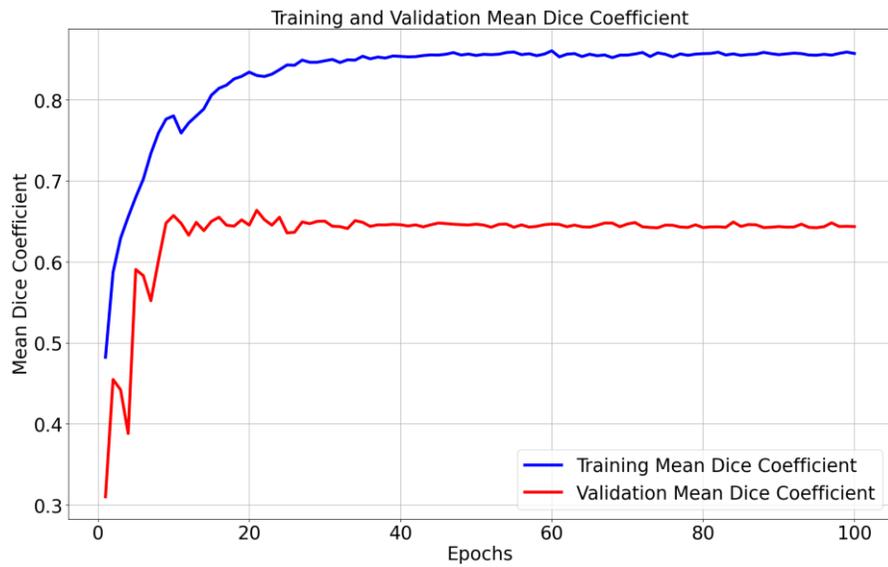
Dice loss del entrenamiento y validación del modelo versión 2



Nota. Elaboración propia.

Figura 59

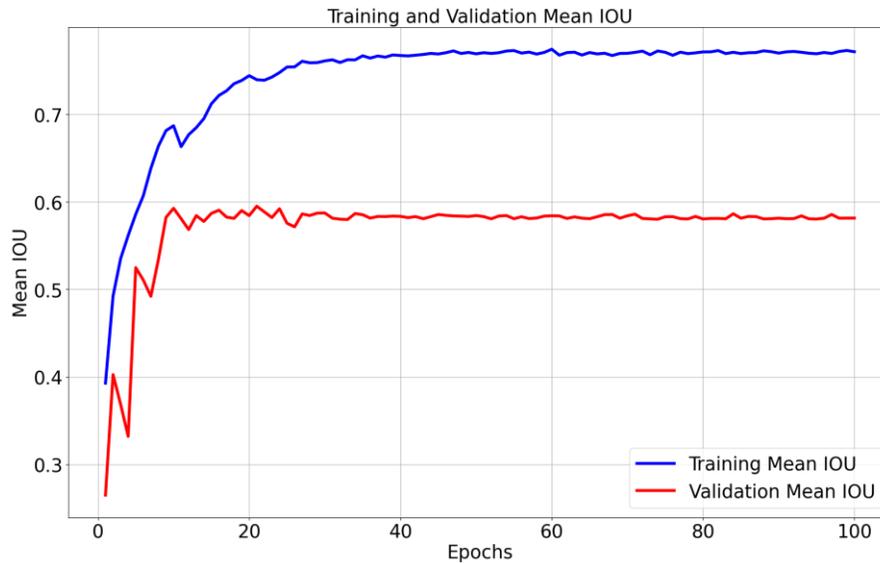
Mean dice coefficient del entrenamiento y validación del modelo versión 2



Nota. Elaboración propia.

Figura 60

Mean IOU del entrenamiento y validación del modelo versión 2



Nota. Elaboración propia.

Con respecto a las 3 métricas se puede notar que existe una mejoría tanto en el conjunto de entrenamiento como el de validación, una mejoría con respecto a los valores reportados por la versión base line, además en el conjunto de validación se puede verificar que los picos inicialmente si son muy pronunciados, pero a partir aproximadamente de la época 10 ya se estabilizan. En la tabla 34 se detalla el valor de cada métrica alcanzado en la época 100.

Tabla 34

Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 2 en la época 100 en el entrenamiento y validación

Conjunto de entrenamiento	Dice loss (época 100)	Mean dice coefficient (época 100)	Mean IOU (época 100)
Entrenamiento.	0.142.	0.857.	0.771.
Validación.	0.356.	0.643.	0.581.

Nota. Elaboración propia.

A continuación, en la tabla 35 se detalla el IOU por cada categoría utilizando el 20% del conjunto de validación.

Tabla 35

IOU de cada categoría del modelo versión 2 utilizando el 20% del conjunto de validación

IOU del background	IOU de lengua de vaca	IOU del diente de león	IOU del kikuyo	IOU de otro tipo de malezas	IOU de papa	Mean IOU
0.952.	0.836.	0.476.	0.658.	0.124.	0.901.	0.658.

Nota. Elaboración propia.

A continuación, en la tabla 36 se detalla el tiempo de inferencia y de entrenamiento.

Tabla 36

Tiempo de inferencia y de entrenamiento del modelo versión 2

Detalle (en base al entorno de entrenamiento)	Tiempo (segundos)
Tiempo de entrenamiento de una sola época.	74.
Tiempo de todo el entrenamiento (100 épocas).	7400.
Tiempo de inferencia de una sola imagen de 128px por 128px.	0.081.
Tiempo de inferencia de una imagen de 5472px por 3648px (330 sub imágenes de 128px por 128px, primero se extrajo de 250px por 250px y luego se redimensionó a 128px por 128px).	0.734.

Nota. Elaboración propia.

La razón de esta mejora es que en la arquitectura de la versión 2 se implementan capas de upsampling gradualmente hasta llegar a la resolución de salida, es decir 128 x 128, en lugar de hacer el aumento de resolución bruscamente como en el modelo base line, por ejemplo, en el modelo base line para la tercera estación se utiliza un único upsampling de 8 x 8, en la versión 2 se utilizan 3 capas upsampling de 2 x 2.

Además, otras de las mejoras importantes es la sustitución de la función de activación ReLU por Mish y la programación de una función de learning rate con decaimiento,

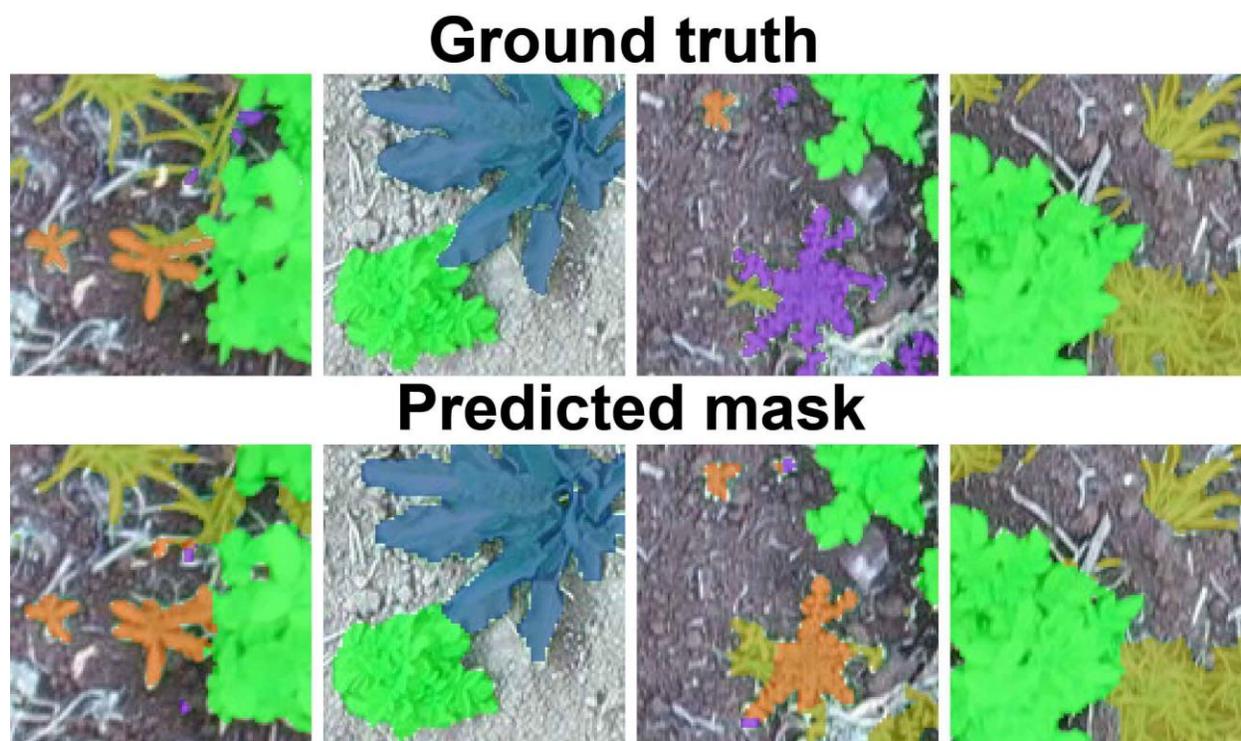
específicamente la función cosine annealing con una reducción del 70% del learning rate inicial con respecto al valor inicial del ciclo anterior.

Todas estas mejoras han dado como resultado un aumento en el rendimiento del modelo versión 2 y lo más importante, la máscara de segmentación semántica se asemeja de mejor manera a las siluetas de las plantas.

A continuación, en la figura 61 se muestra el ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px.

Figura 61

Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 2



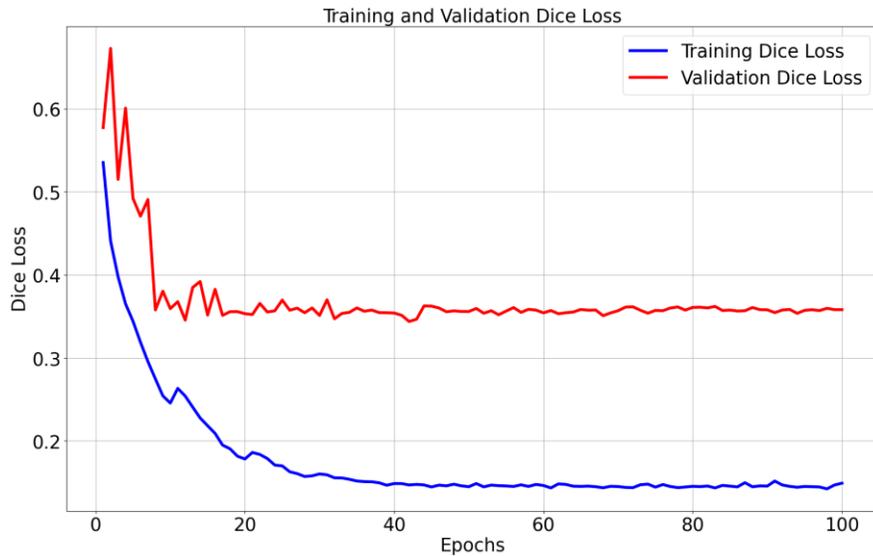
Nota. La maleza kikuyo es identificada por el color amarillo mostaza, el diente de león por el color naranja, la lengua de vaca por el color azul, el cultivo de papa por el color verde y la maleza que pertenece a la categoría otra por el color morado. Elaboración propia.

3.1.3 Versión 3

A continuación, en las figuras 62, 63 y 64 se muestran las gráficas de dice loss, mean dice coefficient y mean IOU respectivamente de la etapa de entrenamiento y validación.

Figura 62

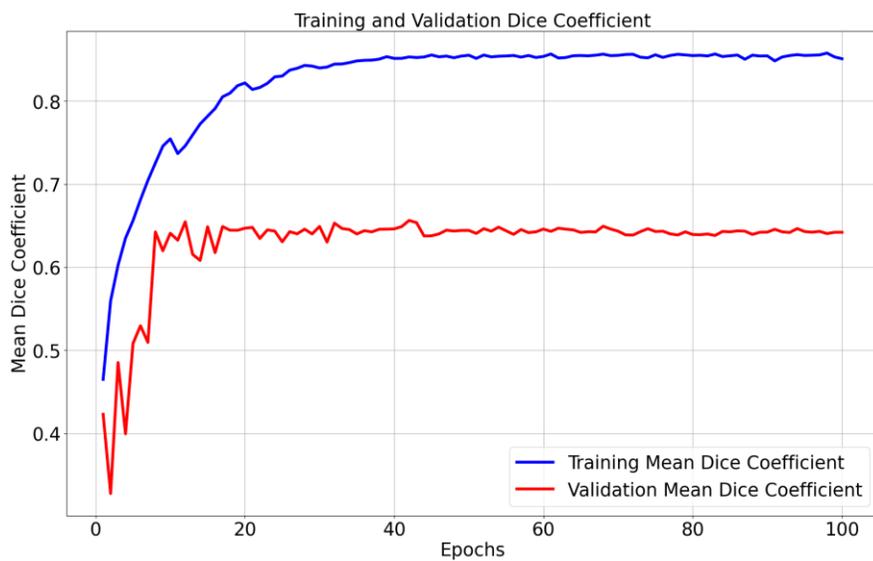
Dice loss del entrenamiento y validación del modelo versión 3



Nota. Elaboración propia.

Figura 63

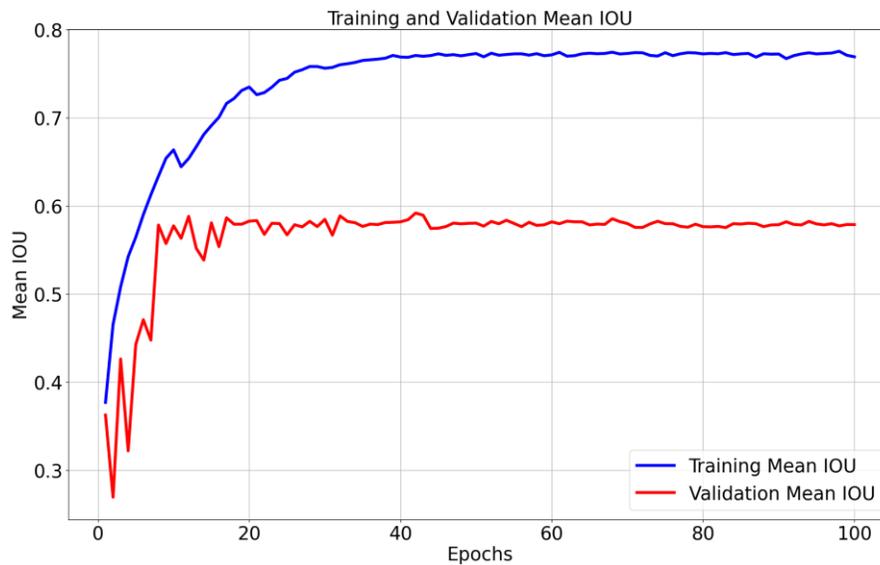
Mean dice coefficient del entrenamiento y validación del modelo versión 3



Nota. Elaboración propia.

Figura 64

Mean IOU del entrenamiento y validación del modelo versión 3



Nota. Elaboración propia.

Se puede observar que en las 3 métricas los valores de esta versión 3 son casi idénticos con el de la versión 2. Se podría decir que la razón está relacionada con la distribución de las clases del data set, la arquitectura propuesta fue pensada para mejorar la parte del decoder encargada de generar la máscara de segmentación semántica, por lo que la robustez con respecto a la versión 2 es superior.

Por su parte el data set no presenta un balance de clases idóneo debido a la naturaleza del proyecto, en la sección de comparación de todos los modelos se hará más énfasis en el impacto del desbalance de clases. El desbalance de clases puede ser una razón del porque no ha existido una mejoría con respecto a la versión anterior. En la tabla 37 se detalla el valor de cada métrica alcanzado en la época 100.

Tabla 37

Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 3 en la época 100 en el entrenamiento y validación

Conjunto de entrenamiento	Dice loss (época 100)	Mean dice coefficient (época 100)	Mean IOU (época 100)
Entrenamiento.	0.149.	0.850.	0.768.
Validación.	0.358.	0.641.	0.578.

Nota. Elaboración propia.

A continuación, en la tabla 38 se detalla el IOU por cada categoría utilizando el 20% del conjunto de validación.

Tabla 38

IOU de cada categoría del modelo versión 3 utilizando el 20% del conjunto de validación

IOU del background	IOU de lengua de vaca	IOU del diente de león	IOU del kikuyo	IOU de otro tipo de malezas	IOU de papa	Mean IOU
0.956.	0.769.	0.478.	0.671.	0.242.	0.888.	0.668.

Nota. Elaboración propia.

A continuación, en la tabla 39 se detalla el tiempo de inferencia y de entrenamiento.

Tabla 39

Tiempo de inferencia y de entrenamiento del modelo versión 3

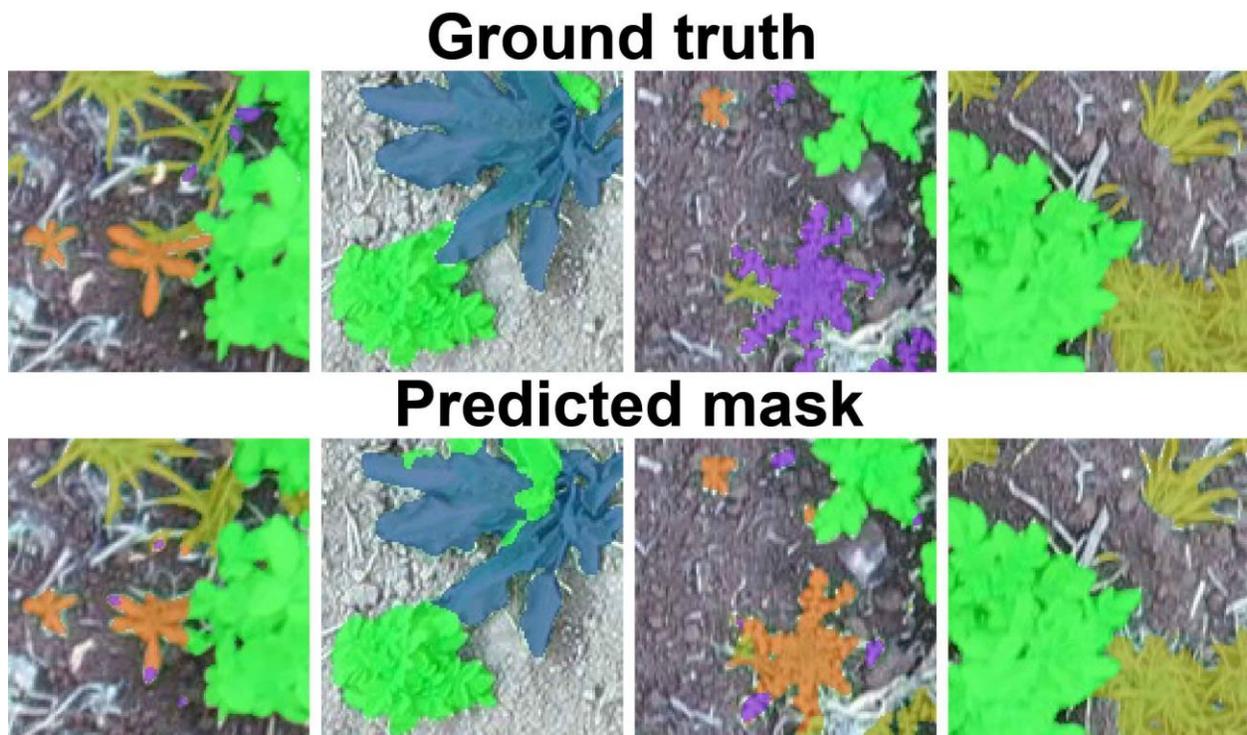
Detalle (en base al entorno de entrenamiento)	Tiempo (segundos)
Tiempo de entrenamiento de una sola época.	172.
Tiempo de todo el entrenamiento (100 épocas).	17200.
Tiempo de inferencia de una sola imagen de 128px por 128px.	0.109.
Tiempo de inferencia de una imagen de 5472px por 3648px (330 sub imágenes de 128px por 128px, primero se extrajo de 250px por 250px y luego se redimensionó a 128px por 128px).	1.672.

Nota. Elaboración propia.

A continuación, en la figura 65 se muestra el ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px.

Figura 65

Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 3



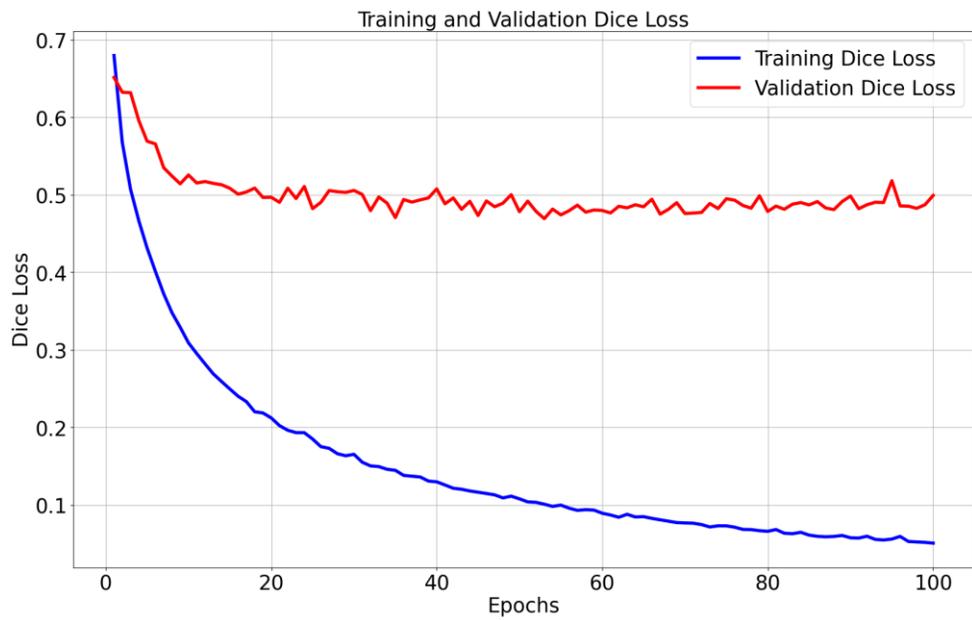
Nota. La maleza kikuyo es identificada por el color amarillo mostaza, el diente de león por el color naranja, la lengua de vaca por el color azul, el cultivo de papa por el color verde y la maleza que pertenece a la categoría otra por el color morado. Elaboración propia.

3.1.4 Versión 4

A continuación, en las figuras 66, 67 y 68 se muestran las gráficas de dice loss, mean dice coefficient y mean IOU respectivamente de la etapa de entrenamiento y validación.

Figura 66

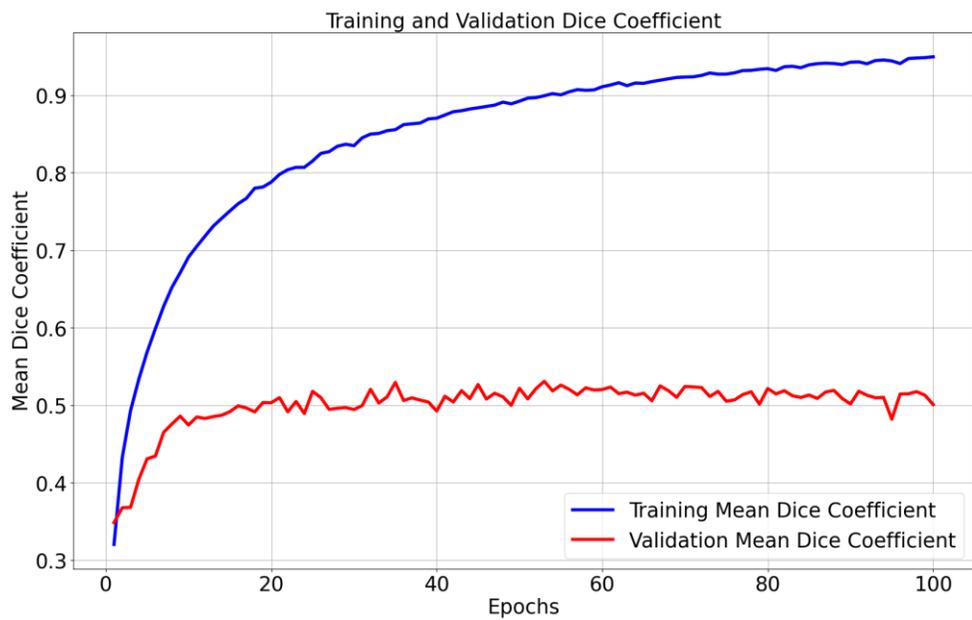
Dice loss del entrenamiento y validación del modelo versión 4



Nota. Elaboración propia.

Figura 67

Mean dice coefficient del entrenamiento y validación del modelo versión 4



Nota. Elaboración propia.

Figura 68

Mean IOU del entrenamiento y validación del modelo versión 4



Nota. Elaboración propia.

A partir de la versión 4 en adelante la arquitectura utilizada es la misma de la versión 3, desde este punto se decidió experimentar con distintos valores de los hiperparámetros para observar cuál era su impacto en los resultados.

Como se puede observar en las gráficas de las 3 métricas existe una clara evidencia de overfitting ya que la brecha entre los resultados del conjunto de entrenamiento y validación es cada vez mayor a medida que avanza cada época, la explicación que se puede dar para este comportamiento es que se definió un learning rate estático de 0.0001 en lugar del dinámico con decaimiento utilizado en los anteriores modelos, además de un tamaño de lote más grande, pasando de 8 a 16.

En este punto se puede evidenciar la importancia de un learning rate dinámico, por esta razón para las posteriores versiones se decidió volver a implementar la función de decaimiento cosine annealing, además, para probar el impacto que tiene el tamaño del lote se decidió duplicar la cantidad utilizada en esta versión para entrenar la versión 5. En la tabla 40 se detalla el valor de cada métrica alcanzado en la época 100.

Tabla 40

Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 4 en la época 100 en el entrenamiento y validación

Conjunto de entrenamiento	Dice loss (época 100)	Mean dice coefficient (época 100)	Mean IOU (época 100)
Entrenamiento.	0.050.	0.949.	0.906.
Validación.	0.499.	0.500.	0.446.

Nota. Elaboración propia.

A continuación, en la tabla 41 se detalla el IOU por cada categoría utilizando el 20% del conjunto de validación.

Tabla 41

IOU de cada categoría del modelo versión 4 utilizando el 20% del conjunto de validación

IOU del background	IOU de lengua de vaca	IOU del diente de león	IOU del kikuyo	IOU de otro tipo de malezas	IOU de papa	Mean IOU
0.949.	0.577.	0.460.	0.546.	0.003.	0.817.	0.559.

Nota. Elaboración propia.

A continuación, en la tabla 42 se detalla el tiempo de inferencia y de entrenamiento.

Tabla 42

Tiempo de inferencia y de entrenamiento del modelo versión 4

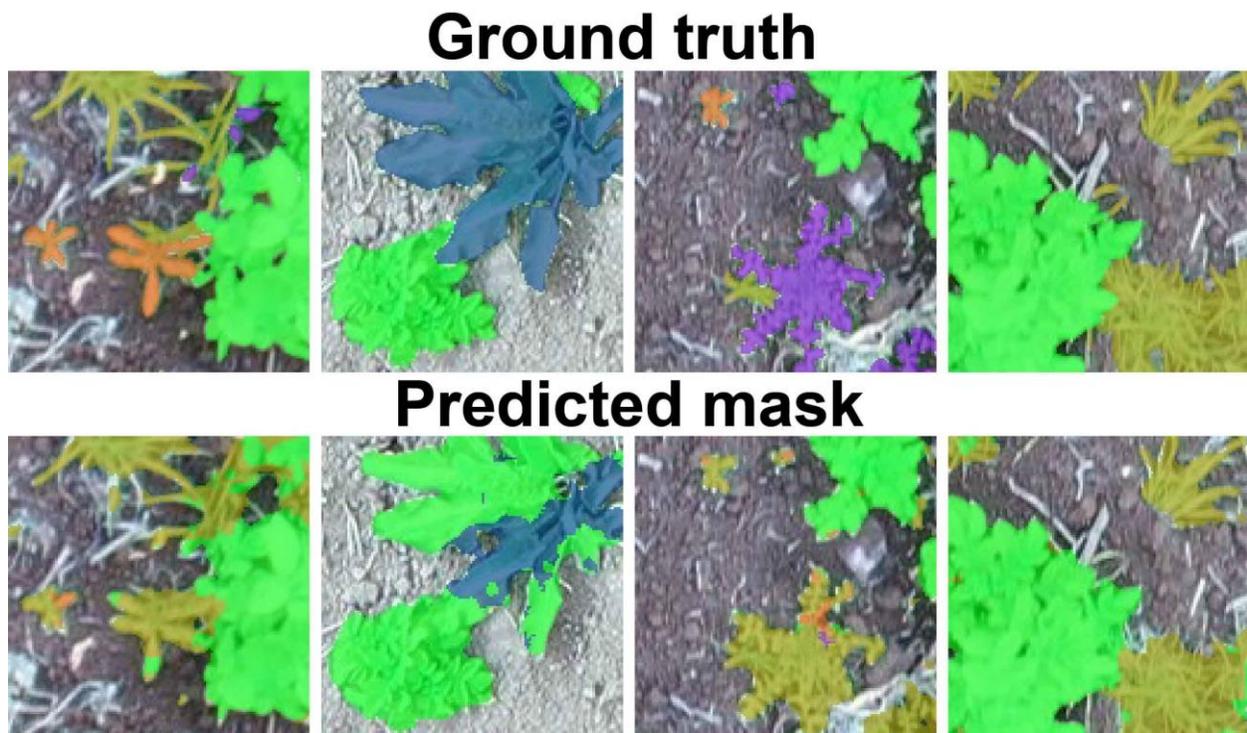
Detalle (en base al entorno de entrenamiento)	Tiempo (segundos)
Tiempo de entrenamiento de una sola época.	132.
Tiempo de todo el entrenamiento (100 épocas).	13200.
Tiempo de inferencia de una sola imagen de 128px por 128px.	0.094.
Tiempo de inferencia de una imagen de 5472px por 3648px (300 sub imágenes de 128px por 128px, primero de extrajo de 250px por 250px y luego se redimensionó a 128px por 128px).	1.708.

Nota. Elaboración propia.

A continuación, en la figura 69 se muestra el ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px.

Figura 69

Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 4



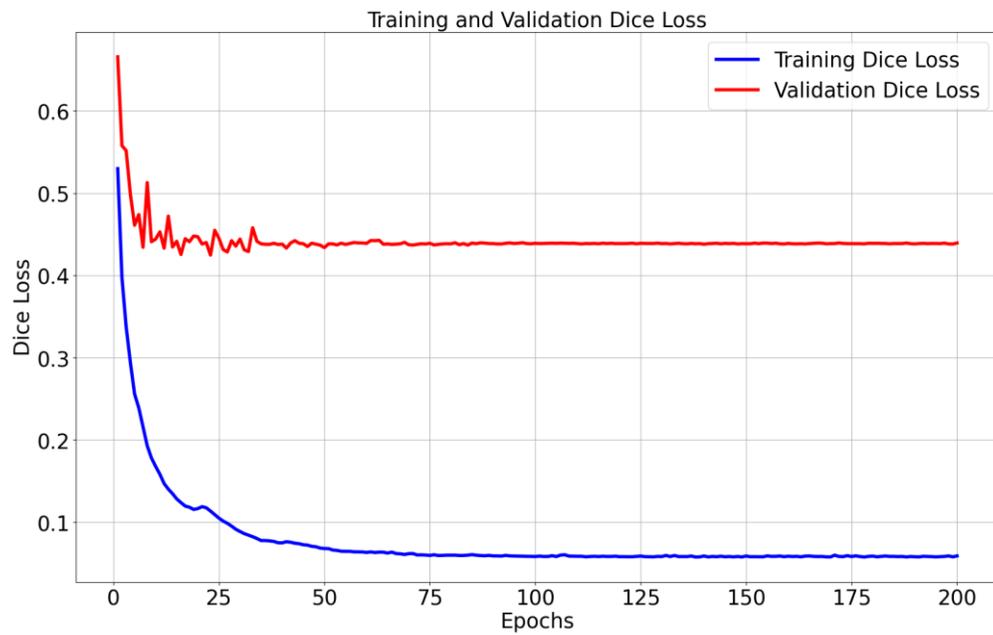
Nota. La maleza kikuyo es identificada por el color amarillo mostaza, el diente de león por el color naranja, la lengua de vaca por el color azul, el cultivo de papa por el color verde y la maleza que pertenece a la categoría otra por el color morado. Elaboración propia.

3.1.5 Versión 5

A continuación, en las figuras 70, 71 y 72 se muestran las gráficas de dice loss, mean dice coefficient y mean IOU respectivamente de la etapa de entrenamiento y validación.

Figura 70

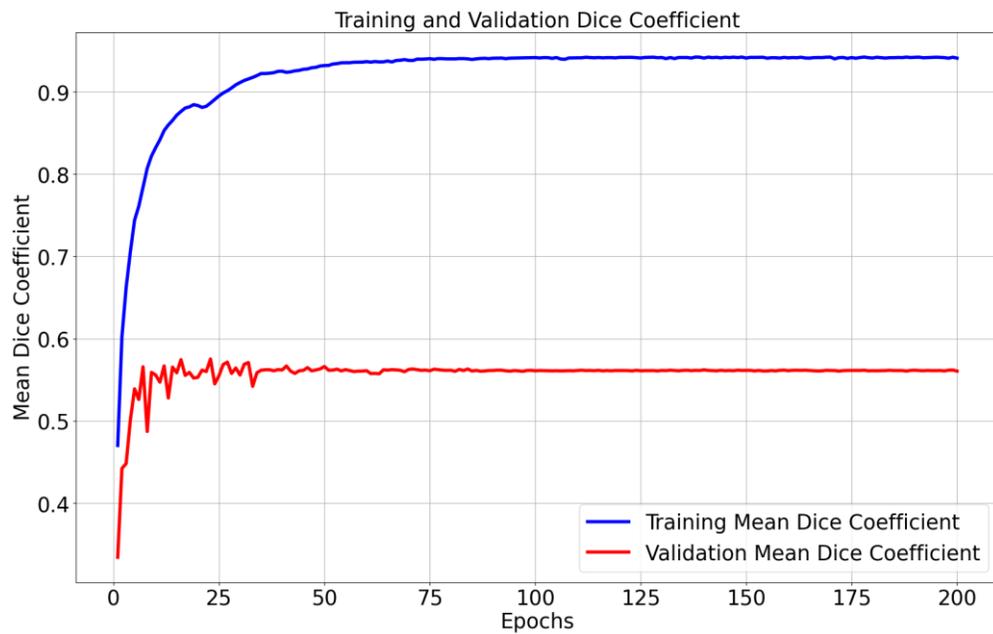
Dice loss del entrenamiento y validación del modelo versión 5



Nota. Elaboración propia.

Figura 71

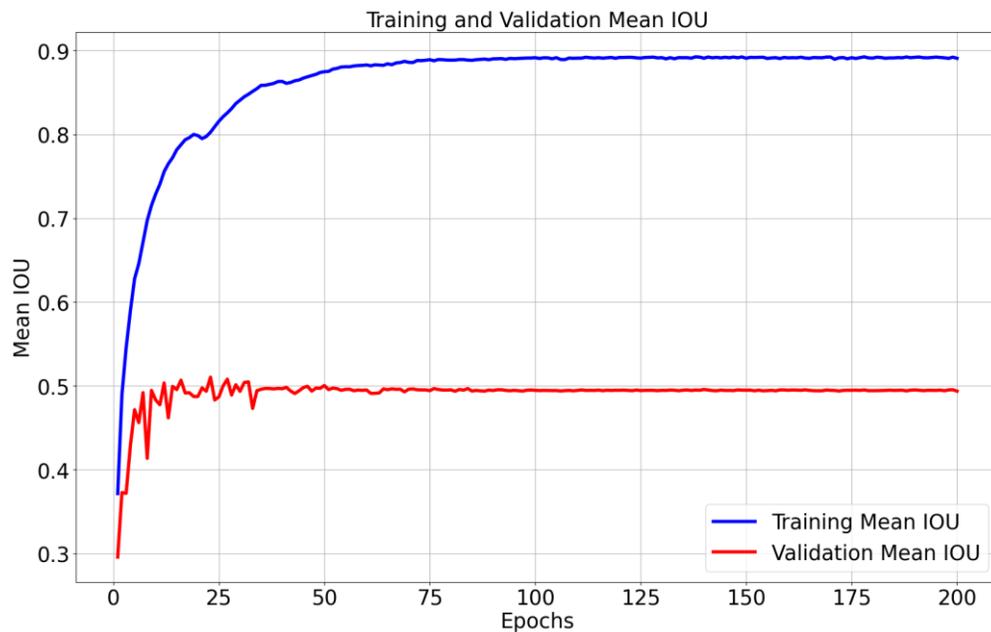
Mean dice coefficient del entrenamiento y validación del modelo versión 5



Nota. Elaboración propia.

Figura 72

Mean IOU del entrenamiento y validación del modelo versión 5



Nota. Elaboración propia.

Haciendo una analogía con respecto a las versiones anteriores se puede llegar a una conclusión, a mayor valor del tamaño de lote mayor presencia de overfitting, esta puede ser una razón del porque los demás investigadores utilizan un tamaño de lote pequeño, por esta razón para la siguiente y última versión se ha decidió utilizar el tamaño de lote más pequeño aconsejado, que es 2, otro aspecto observado es que aumentar el número de épocas de 100 a 200 no ha supuesto ninguna mejora, ya que el rendimiento máximo se lo consigue aproximadamente por la época 100.

Analizando las 3 métricas se puede observar que en el conjunto de entrenamiento se consiguen resultados bastantes buenos, sin embargo, con el conjunto de validación son pésimos, esto indicaría que el modelo no puede generalizar bien datos nuevos, por lo que su confiabilidad sería mala al momento de hacer predicciones. Esto es una clara evidencia de overfitting.

En la tabla 43 se detalla el valor de cada métrica alcanzado en la época 100 y 200 para notar que no existe diferencia significativa.

Tabla 43

Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 5 en la época 100 y 200 en el entrenamiento y validación

Conjunto de entrenamiento	Dice loss (época 100 y 200)	Mean dice coefficient (época 100 y 200)	Mean IOU (época 100 y 200)
Entrenamiento.	0.058 – 0.059.	0.941 – 0.940.	0.891 – 0.890.
Validación.	0.439 – 0.439.	0.560 – 0.560.	0.494 – 0.493.

Nota. Elaboración propia.

A continuación, en la tabla 44 se detalla el IOU por cada categoría utilizando el 20% del conjunto de validación.

Tabla 44

IOU de cada categoría del modelo versión 5 utilizando el 20% del conjunto de validación

IOU del background	IOU de lengua de vaca	IOU del diente de león	IOU del kikuyo	IOU de otro tipo de malezas	IOU de papa	Mean IOU
0.946.	0.694.	0.580.	0.596.	0.031.	0.853.	0.617.

Nota. Elaboración propia.

A continuación, en la tabla 45 se detalla el tiempo de inferencia y de entrenamiento.

Tabla 45

Tiempo de inferencia y de entrenamiento del modelo versión 5

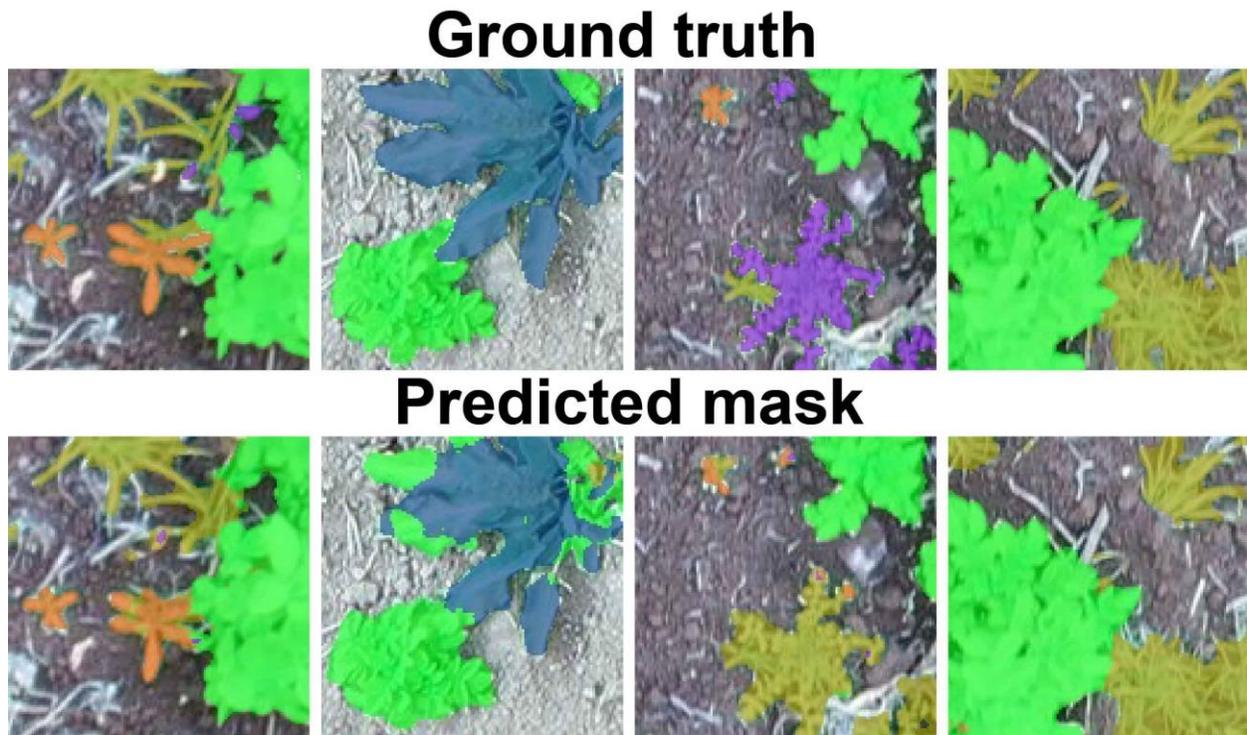
Detalle (en base al entorno de entrenamiento)	Tiempo (segundos)
Tiempo de entrenamiento de una sola época.	113.
Tiempo de todo el entrenamiento (200 épocas).	22600.
Tiempo de inferencia de una sola imagen de 128px por 128px.	0.092.
Tiempo de inferencia de una imagen de 5472px por 3648px (330 sub imágenes de 128px por 128px, primero se extrajo de 250px por 250px y luego se redimensionó a 128px por 128px).	1.695.

Nota. Elaboración propia.

A continuación, en la figura 73 se muestra el ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px.

Figura 73

Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 5



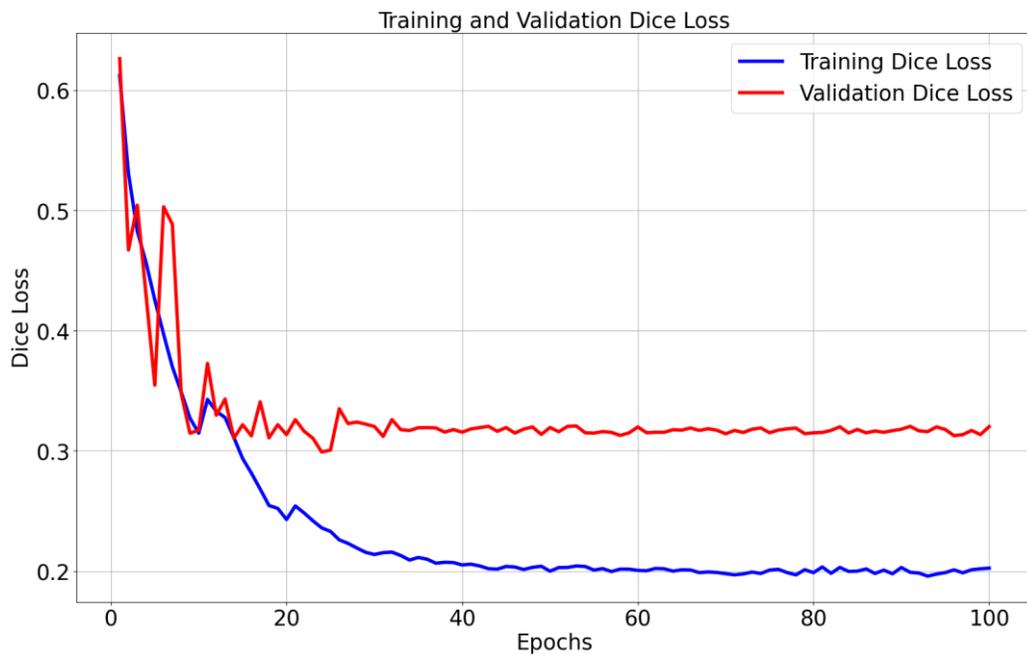
Nota. La maleza kikuyo es identificada por el color amarillo mostaza, el diente de león por el color naranja, la lengua de vaca por el color azul, el cultivo de papa por el color verde y la maleza que pertenece a la categoría otra por el color morado. Elaboración propia.

3.1.6 Versión 6

A continuación, en las figuras 74, 75 y 76 se muestran las gráficas de dice loss, mean dice coefficient y mean IOU respectivamente de la etapa de entrenamiento y validación.

Figura 74

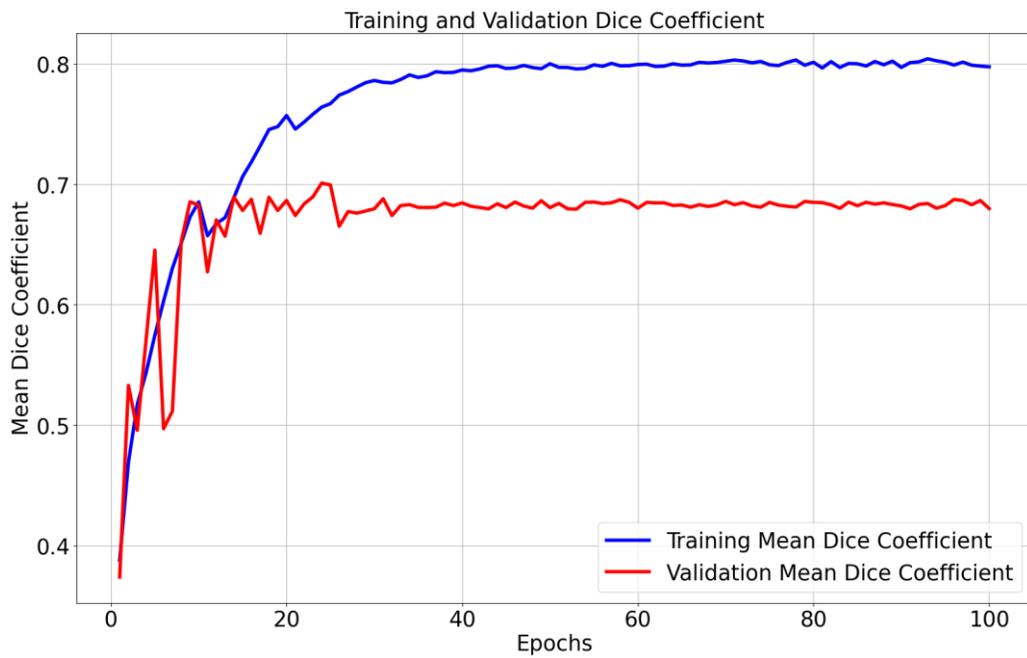
Dice loss del entrenamiento y validación del modelo versión 6



Nota. Elaboración propia.

Figura 75

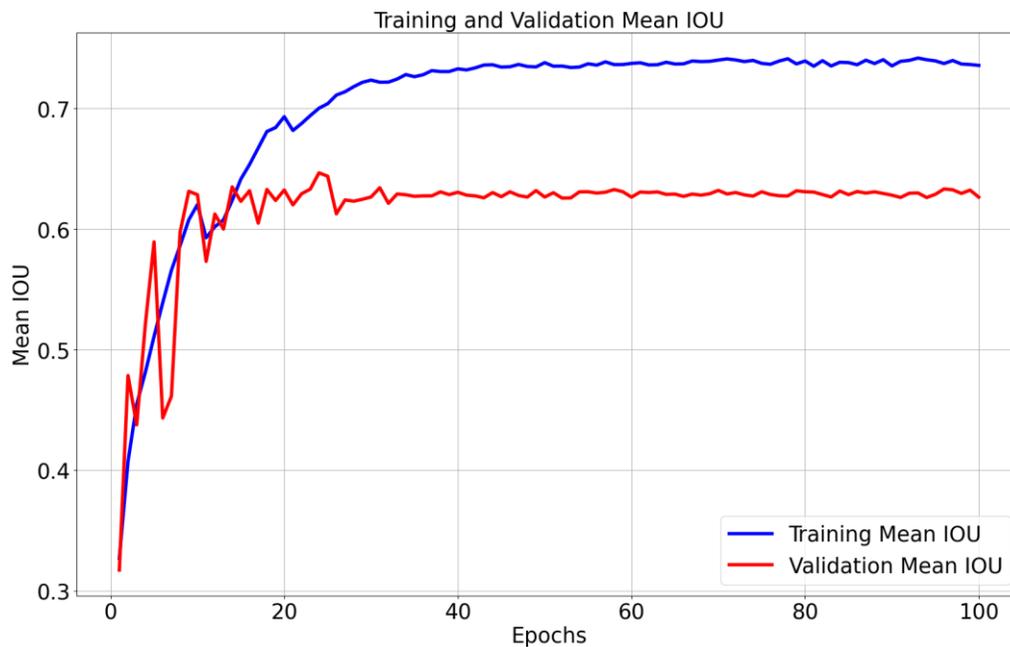
Mean dice coefficient del entrenamiento y validación del modelo versión 6



Nota. Elaboración propia.

Figura 76

Mean IOU del entrenamiento y validación del modelo versión 6



Nota. Elaboración propia.

Con respecto a las 3 métricas se puede decir que ha habido una mejora significativa en el conjunto de validación, ya que la brecha entre el conjunto de entrenamiento y validación es menor en comparación con las demás versiones.

Si bien es cierto en el conjunto de entrenamiento el rendimiento ha bajado con respecto a los demás modelos, el aumento de rendimiento del conjunto de validación supone que el modelo es mejor prediciendo datos que aún no ha visto, por lo que sería mejor que los demás modelos.

Se puede evidenciar que una disminución en el tamaño del lote ayuda a que los resultados mejoren en lo que respecta a la generalización.

En la tabla 46 se detalla el valor de cada métrica alcanzado en la época 100.

Tabla 46

Valores del dice loss, mean dice coefficient y mean IOU del modelo versión 6 en la época 100 en el entrenamiento y validación

Conjunto de entrenamiento	Dice loss (época 100)	Mean dice coefficient (época 100)	Mean IOU (época 100)
Entrenamiento.	0.202.	0.797.	0.735.
Validación.	0.320.	0.679.	0.626.

Nota. Elaboración propia.

A continuación, en la tabla 47 se detalla el IOU por cada categoría utilizando el 20% del conjunto de validación.

Tabla 47

IOU de cada categoría del modelo versión 6 utilizando el 20% del conjunto de validación

IOU del background	IOU de lengua de vaca	IOU del diente de león	IOU del kikuyo	IOU de otro tipo de malezas	IOU de papa	Mean IOU
0.956.	0.826.	0.583.	0.703.	0.434.	0.908.	0.735.

Nota. Elaboración propia.

A continuación, en la tabla 48 se detalla el tiempo de inferencia y de entrenamiento.

Tabla 48

Tiempo de inferencia y de entrenamiento del modelo versión 6

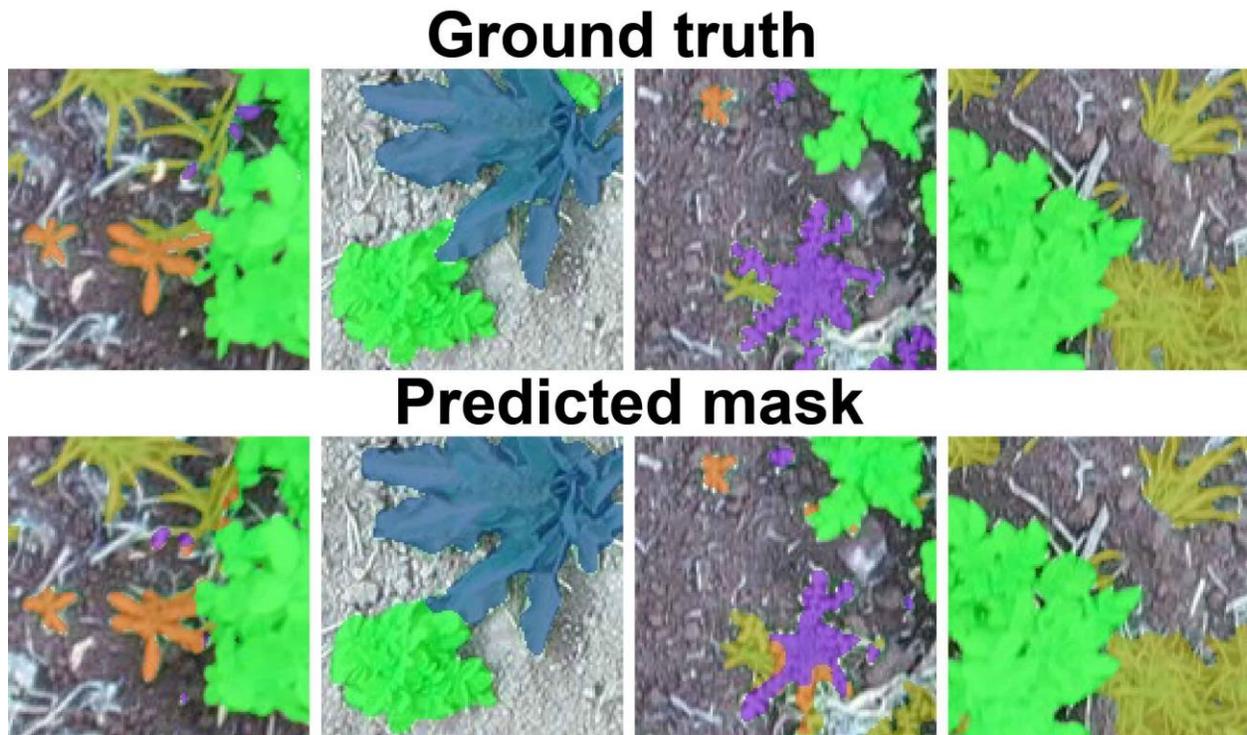
Detalle (en base al entorno de entrenamiento)	Tiempo (segundos)
Tiempo de entrenamiento de una sola época.	465.
Tiempo de todo el entrenamiento (100 épocas).	46500.
Tiempo de inferencia de una sola imagen de 128px por 128px.	0.094.
Tiempo de inferencia de una imagen de 5472px por 3648px (330 sub imágenes de 128px por 128px, primero se extrajo de 250px por 250px y luego se redimensionó a 128px por 128px).	1.698.

Nota. Elaboración propia.

A continuación, en la figura 77 se muestra el ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px.

Figura 77

Ground truth y el resultado de la predicción de 4 imágenes de 128px por 128px utilizando el modelo versión 6



Nota. La maleza kikuyo es identificada por el color amarillo mostaza, el diente de león por el color naranja, la lengua de vaca por el color azul, el cultivo de papa por el color verde y la maleza que pertenece a la categoría otra por el color morado. Elaboración propia.

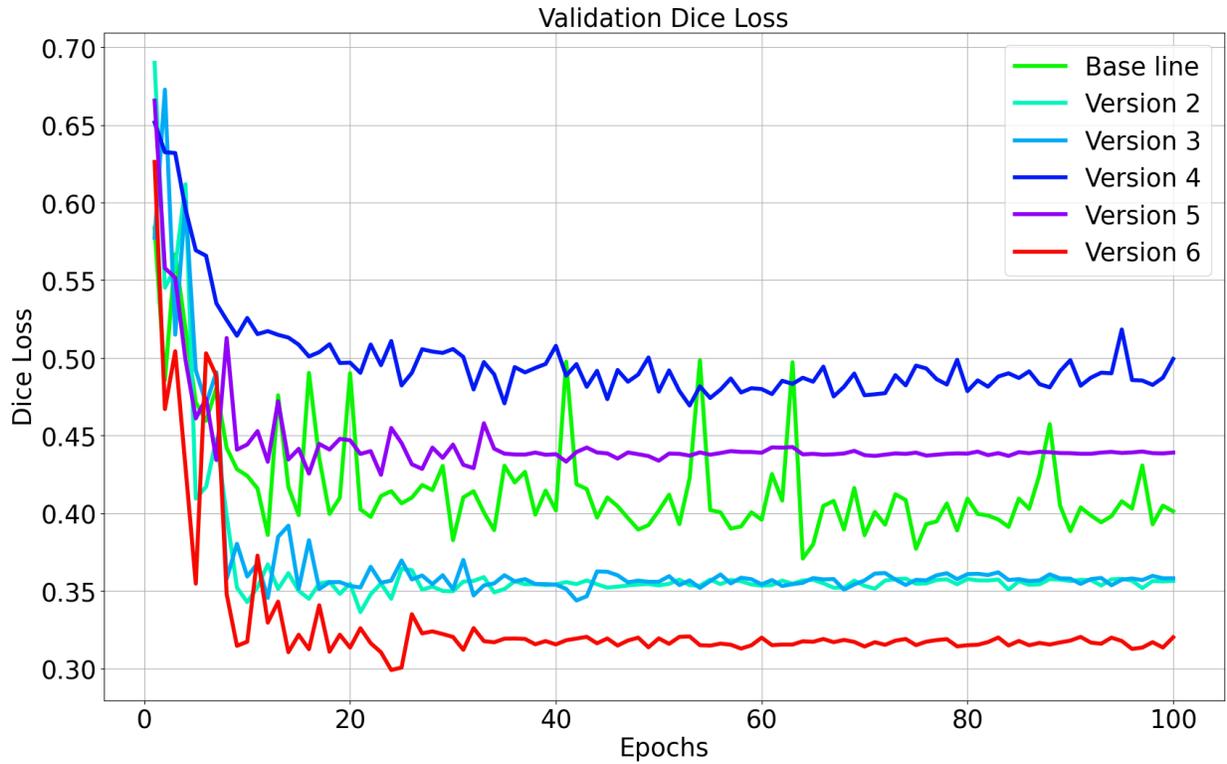
3.1.7 Evaluación conjunta de las versiones

Una vez analizado individualmente las métricas de cada versión se procede a analizarlas conjuntamente para poder elegir la versión más idónea para implementar en la aplicación web.

En la figura 78 se muestra la gráfica de la métrica dice loss de todas las versiones en la etapa de validación.

Figura 78

Dice loss de la etapa de validación de todas las versiones



Nota. Elaboración propia.

En la tabla 49 se hace un resumen de los valores del dice loss de todas las versiones en la época 100.

Tabla 49

Valores del dice loss de todas las versiones en la época 100 en la fase de validación

Versión	Dice loss (época 100)
Base line.	0.401.
Versión 2.	0.356.
Versión 3.	0.358.
Versión 4.	0.499.
Versión 5.	0.439.
Versión 6.	0.320.

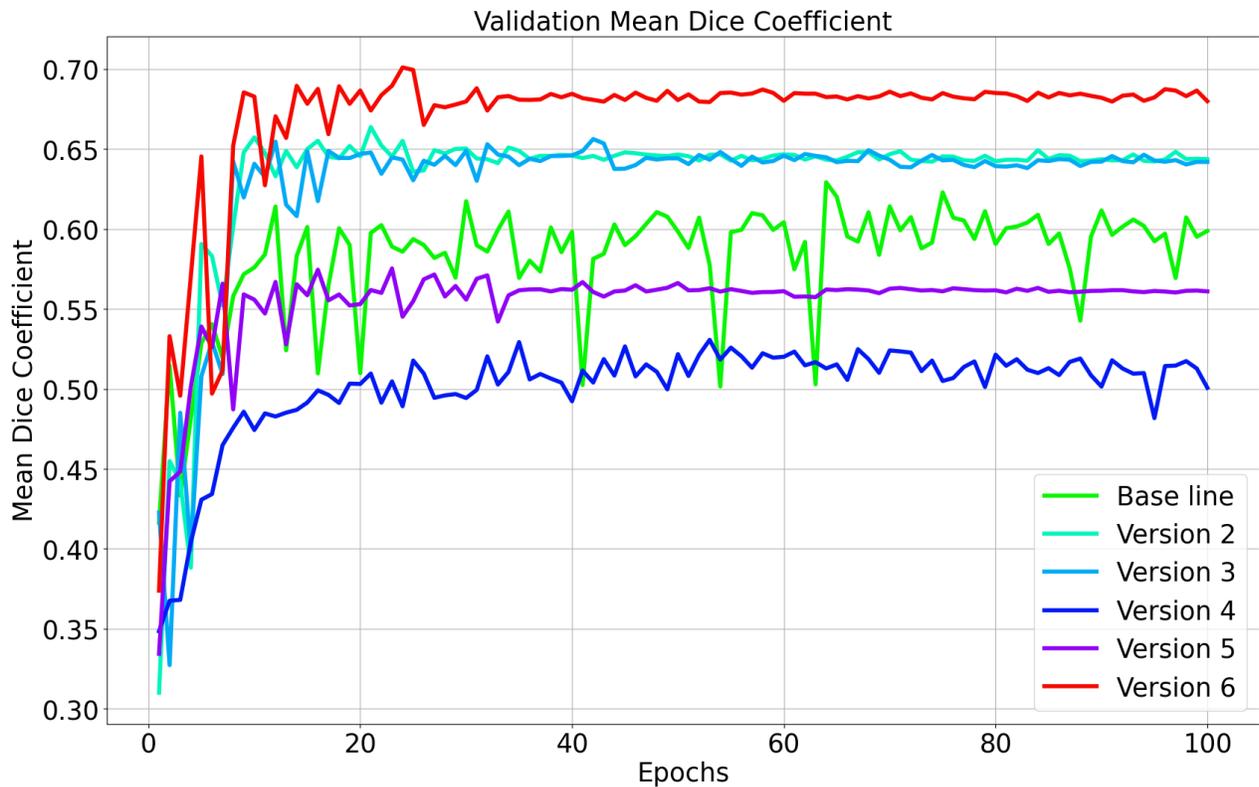
Nota. Elaboración propia.

En la tabla 49 se puede notar que la mejor versión es la sexta, ya que su valor dice loss en la época 100 es de 0.320, es la más baja en comparación con las demás, analizando los resultados se puede decir que la sexta versión puede hacer una mayor generalización, esto significa que con datos nuevos tiene mayor precisión en las predicciones.

En la figura 79 se muestra la gráfica de la métrica mean dice coefficient de todas las versiones

Figura 79

Mean dice coefficient de la etapa de validación de todas las versiones



Nota. Elaboración propia.

En la tabla 50 se hace un resumen de los valores del mean dice coefficient de todas las versiones en la época 100.

Tabla 50

Valores del mean dice coefficient de todas las versiones en la época 100 en la fase de validación

Versión	Mean dice coefficient (época 100)
Base line.	0.598.
Versión 2.	0.643.
Versión 3.	0.641.
Versión 4.	0.500.
Versión 5.	0.560.
Versión 6.	0.679.

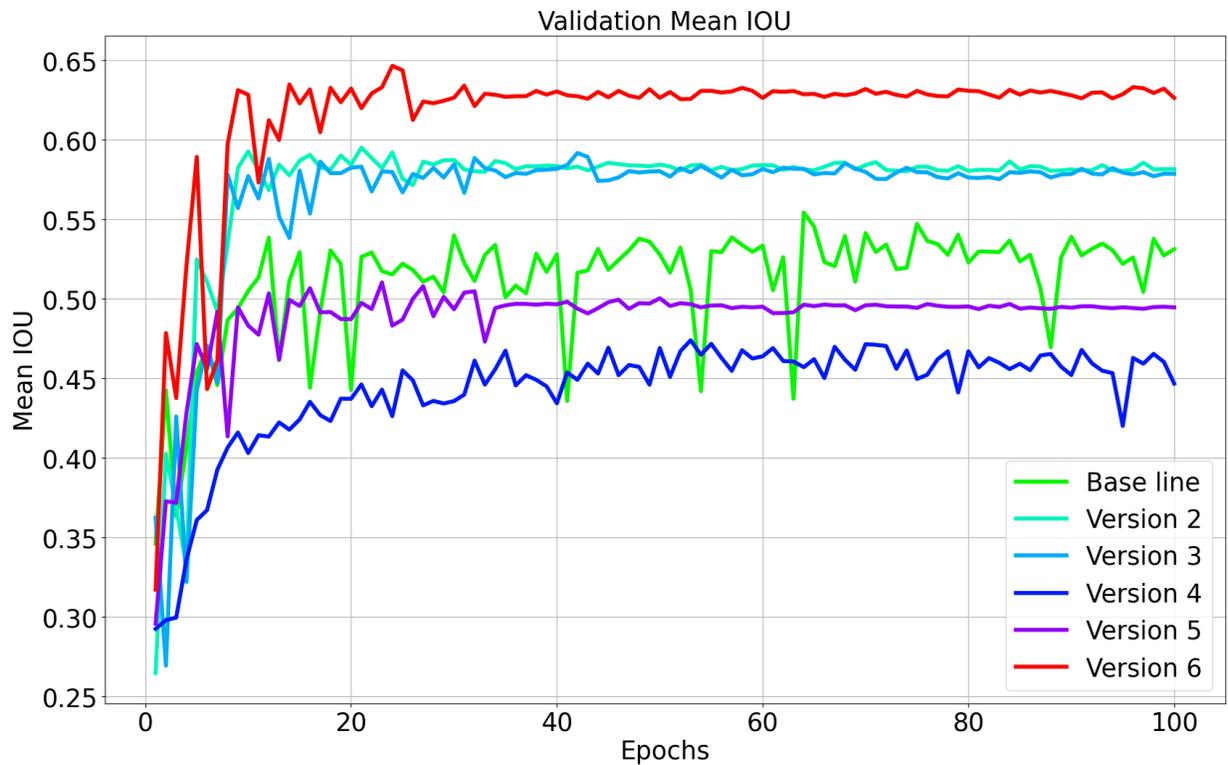
Nota. Elaboración propia.

En la tabla 50 se puede observar que la sexta versión es la mejor también en esta métrica, su valor de 0.679 es superior a las demás, se da el mismo criterio que se dio en el análisis de la métrica dice loss, la sexta versión tiene una mayor capacidad de generalización que las demás, lo que la hace más idónea para ser el mejor modelo.

En la figura 80 se muestra la gráfica correspondiente a la métrica mean IOU de todas las versiones.

Figura 80

Mean IOU de la etapa de validación de todas las versiones



Nota. Elaboración propia.

En la tabla 51 se hace un resumen de los valores del mean IOU de todas las versiones en la época 100.

Tabla 51

Valores del mean IOU de todas las versiones en la época 100 en la fase de validación

Versión	Mean IOU (época 100)
Base line.	0.531.
Versión 2.	0.581.
Versión 3.	0.578.
Versión 4.	0.446.
Versión 5.	0.494.
Versión 6.	0.626.

Nota. Elaboración propia.

En la métrica mean IOU de la etapa de validación se puede observar que la sexta versión sigue siendo la mejor, con un valor de 0.626 es superior a las demás.

En la sección anterior que se enfocaba al análisis de cada versión del modelo de forma individual, específicamente en el análisis de los resultados del modelo versión 3 se mencionó que el desbalance de clases del data set afecta negativamente a los valores obtenidos de las métricas. Lo idea es que el número de individuos por cada clase sea equilibrado, sin embargo, debido a la naturaleza del proyecto esto no pudo ser así.

Debido a que no se definió un porcentaje del data set para el conjunto de pruebas se decidió tomar el 20% del conjunto de validación para poder obtener los valores del IOU de cada clase a fin de evidenciar la afectación del desbalance de clases. A continuación, en la tabla 52 se muestra el IOU de cada clase de todas las versiones del modelo entrenadas.

Tabla 52

IOU de cada clase de todas las versiones del modelo entrenadas utilizando el 20% del conjunto de validación

Versión	IOU del background	IOU de lengua de vaca	IOU del diente de león	IOU del kikuyo	IOU de otras malezas	IOU de papa	Mean IOU
Base line	0.911.	0.711.	0.505.	0.536.	0.009.	0.834.	0.584.
Versión 2.	0.952.	0.836.	0.476.	0.658	0.124.	0.901.	0.658.
Versión 3.	0.956.	0.769.	0.478.	0.671.	0.242.	0.888.	0.668.
Versión 4.	0.949.	0.577.	0.460.	0.546.	0.003.	0.817.	0.559.
Versión 5.	0.946.	0.694.	0.580.	0.596.	0.031.	0.853.	0.617.
Versión 6.	0.956.	0.826.	0.583.	0.703.	0.434.	0.908.	0.735.

Nota. Elaboración propia.

Como se puede observar en la tabla 52 las clases que mayor IOU presentan son las del background y papa, la que más se ve perjudicada es la clase de otro tipo de malezas, esta presenta el IOU más bajo de todas, con los datos plasmados en la tabla 52 se puede observar que el desbalance de clases afecta en los resultados de todos los modelos entrenados, el que mejor puntaje presenta es la versión 6 con un mean IOU de 0.735, superando a todas las demás versiones.

Una vez analizado las versiones de forma individual y en conjunto se ha llegado a la conclusión de que la versión 6 del modelo es la mejor, ya que en los análisis realizados es la que mejores resultados ha obtenido, siendo superior a las demás versiones, por esta razón se decidió definirla como el mejor modelo y fue la que se implementó en la aplicación web.

3.2 Validación del nivel de portabilidad de la aplicación web

Una de las propuestas definidas para este trabajo de titulación es priorizar la característica de portabilidad de la aplicación web, específicamente la sub característica “capacidad de ser instalado”, para lo cual se había utilizado la tecnología de contenedores Docker, debido a los beneficios que esta presenta en cuanto a la portabilidad.

La norma ISO 25023 define para esta sub característica 2 métricas, las cuales son: eficiencia en el tiempo de instalación y facilidad de instalación, las fórmulas y la manera de aplicación fueron abordadas en el capítulo 1.

Los recursos que propone utilizar son usuarios o desarrolladores que hagan la instalación del software, en este caso se ha optado por desarrolladores ya que la aplicación al utilizar Docker necesita de personas que sepan utilizar esta tecnología.

En total se ha elegido a 6 personas que son conocidos del autor de este proyecto para poder obtener los valores de las métricas y dar un criterio de la portabilidad de la aplicación web.

A continuación, en la tabla 53 se muestran los resultados de la métrica eficiencia en el tiempo de instalación aplicada a los 6 desarrolladores.

Tabla 53*Resultados de la métrica eficiencia en el tiempo de instalación aplicada a 6 desarrolladores*

Desarrollador / Plataforma	Parámetro A(Número de reintentos al instalar el sistema)	Parámetro T(Tiempo total transcurrido al instalar el sistema, minutos)	Resultado (A/T)
D1 / Azure	0	10	0
D2 / AWS	0	12	0
D3 / Azure	0	9	0
D4 / AWS	0	11	0
D5 / Azure	0	11	0
D6 / Azure	0	9	0

Nota. Elaboración propia.

Como se puede observar en la tabla 53 todos los desarrolladores no hicieron reintentos para poder desplegar la aplicación web, esto quiere decir que el despliegue lo hicieron a la primera, el tiempo es casi similar en todas las mediciones, el mínimo registrado es 9 y el máximo es 12. En este caso al no haber reintentos en las 6 mediciones el resultado de A/T es 0. La diferencia de tiempo puede deberse a la disponibilidad del ancho de banda en la plataforma de Docker Hub en el momento del despliegue.

La segunda métrica se denomina “facilidad de instalación”, a continuación, en la tabla 54 se muestra las mediciones obtenidas.

Tabla 54*Resultados de la métrica facilidad de instalación aplicada a 6 desarrolladores*

Desarrollador / Plataforma	Parámetro A(Número de casos en los que se tuvo éxito al cambiar la forma de instalación)	Parámetro B(Número total de casos en los que se cambió la forma de instalación)	Resultado(A/B)
D1 / Azure	1	1	1

D2 / AWS	1	1	1
D3 / Azure	1	1	1
D4 / AWS	1	1	1
D5 / Azure	1	1	1
D6 / Azure	1	1	1

Nota. Elaboración propia.

Debido a que las plataformas de AWS y Azure tienen muy bien automatizados sus servicios el despliegue se lo puede hacer de una forma, que es únicamente especificando el nombre de la imagen con su respectivo tag, no se debe hacer ninguna otra configuración para poder desplegar la aplicación web.

Como se puede observar los resultados para todas las mediciones es 1, según la ISO 25023 para esta métrica se dice que si el resultado se acerca al 1 es mejor, por lo que se puede llegar a la conclusión de que los resultados obtenidos son perfectos.

Habiendo analizado las 2 métricas enfocadas a la característica “capacidad de ser instalado” se puede llegar a la conclusión de que la portabilidad de la aplicación web en este aspecto es excelente.

3.3 Validación de la problemática

La problemática que pretende solucionar este proyecto es reducir la cantidad de tiempo empleado para la identificación y cuantificación de malezas en cultivos de papa.

Los agricultores para poder llevar a cabo esta tarea recorren toda la zona de interés haciendo una observación visual y utilizando los conocimientos que sus antepasados les han transmitido, básicamente es una medición muy subjetiva.

La cantidad de tiempo empleado aumenta proporcionalmente con respecto al área del terreno, además existen otros factores que pueden determinar un aumento del tiempo, por ejemplo: tipo de superficie del terreno, experiencia del agricultor y clima.

Para poder determinar si la aplicación web ayuda a solventar esta problemática se ha decidido tomar como muestra 3 terrenos y hacer la comparativa del tiempo empleado realizando la identificación y cuantificación de manera tradicional versus el tiempo tomado por la aplicación web.

Para poder realizar la comparación fundamentada en un método estadístico se ha decidido aplicar la prueba t Student, esta prueba ayuda a determinar si existe una diferencia entre las medias de 2 grupos (Student, 1908).

Lind et al. (2012) en su trabajo señalan que la ecuación para calcular el valor t es:

$$t = \frac{d}{S_d/\sqrt{n}}$$

Donde:

- d representa la media de la diferencia.
- S_d representa la desviación estándar de la diferencia.
- n representa el número de muestras.

Se define la hipótesis nula y la alterna, donde:

$$H_0: d = 0$$

$$H_1: d \neq 0$$

H_0 corresponde a la hipótesis nula, esta establece que no existe diferencia significativa en la media de la diferencia, por su parte H_1 que es la hipótesis alterna establece que si existe diferencia significativa.

Además, es importante medir el nivel de confiabilidad de la aplicación web, esto para poder determinar si es viable su uso en un entorno real. Debido a que se van a analizar muchos aspectos se ha decidido dividir a esta sección en varias fases, a continuación, se presentan cada una de ellas.

3.3.1 Prueba t Student enfocada a determinar si hubo una reducción de tiempo utilizando la aplicación web

A continuación, en la tabla 55 se muestra los datos de los terrenos al igual que el tiempo empleado para realizar la identificación y cuantificación de malezas de forma tradicional.

Tabla 55

Tiempo empleado para la identificación y cuantificación de malezas de forma tradicional en 3 terrenos de papa

Terreno	Extensión (m ²)	Tiempo (minutos)
Terreno 1.	800.	66.
Terreno 2.	650.	54.
Terreno 3.	730.	60.

Nota. Elaboración propia.

En la tabla 56 se muestran los resultados con respecto al tiempo empleado para la identificación y cuantificación de malezas utilizando la aplicación web de los 3 terrenos. Cabe recalcar que el tiempo empleado fue tomado en cuenta desde que se vuela el dron para que capture las fotografías hasta que se obtienen los resultados en la aplicación web.

Tabla 56

Tiempo empleado para la identificación y cuantificación de malezas utilizando la aplicación web en 3 terrenos de papa

Terreno	Extensión (m ²)	Tiempo (minutos)
Terreno 1.	800.	50.
Terreno 2.	650.	39.
Terreno 3.	730.	46.

Nota. Elaboración propia.

Excel ofrece un complemento el cual permite realizar la prueba t Student de forma automática, a continuación, en la tabla 57 se muestra los resultados. TT representa el conjunto de tiempo en identificar y cuantificar las malezas de forma tradicional y TA el conjunto de tiempo utilizando la aplicación web. El nivel de confianza elegido fue del 95%.

Tabla 57*Aplicación de la prueba t Student a los conjuntos TT y TA*

Medida	TT	TA
Media	60	45
Varianza	36	31
Observaciones	3	3
Coeficiente de correlación de Pearson	0,987829161	
Diferencia hipotética de las medias	0	
Grados de libertad	2	
Estadístico t	25,98076211	
P(T<=t) una cola	0.000739098	
Valor crítico de t (una cola)	2,91998558	
P-valor	0,001478197	
Valor crítico de t (dos colas)	4,30265273	

Nota. Elaboración Propia.

Para poder rechazar la hipótesis nula se debe cumplir la siguiente condición:

$$P - \text{valor} \leq \alpha$$

Remplazando las variables con los resultados obtenidos se tiene que:

$$0.001 \leq 0.05$$

Se puede observar que la condición se cumple, por lo tanto, la hipótesis nula se rechaza y se acepta la alterna, la cual menciona que si existe una diferencia significativa en la media de la diferencia.

Como conclusión se puede afirmar que existe una disminución del tiempo significativa utilizando la aplicación web para identificar y cuantificar las malezas en comparación a realizar este proceso de forma tradicional.

3.3.2 Prueba t Student y coeficiente de determinación aplicada a la cuantificación de la maleza lengua de vaca

Para realizar este análisis estadístico para comparar la confiabilidad de la aplicación web se utilizó el 20% del conjunto de validación como muestra, esto es debido a que los agricultores no tienen un fundamento técnico para realizar la cuantificación de malezas de manera adecuada, por lo que sus resultados son muy subjetivos ya que se basan en la percepción visual y esto tiene mucha variación dependiendo de cada agricultor y su nivel de experiencia.

Por la razón antes mencionada se decidió utilizar el 20% del conjunto de validación como muestra, utilizando el ground truth que son las máscaras de segmentación semántica, con estos datos ya se puede hacer una comparación objetiva con las predicciones de la aplicación web, el 20% representa 60 imágenes. A continuación, en la tabla 58 se muestra la cuantificación de la maleza lengua de vaca utilizando el ground truth y la predicción de la aplicación web.

Tabla 58

Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada a la maleza lengua de vaca

Número de imagen	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
1	0.000	0.000
2	0.000	0.000
3	0.000	0.000
4	0.000	0.000
5	0.000	0.000
6	0.000	0.000
7	0.000	0.000
8	11.060	11.640
9	0.000	0.000
10	0.000	0.000
11	12.400	11.460

12	0.000	0.000
13	0.000	0.000
14	33.380	32.710
15	0.000	0.000
16	0.000	0.000
17	12.870	12.050
18	0.000	0.000
19	5.600	4.940
20	0.000	0.000
21	11.610	12.580
22	0.000	0.000
23	8.470	8.470
24	0.000	0.000
25	1.700	1.060
26	0.000	0.000
27	0.000	0.000
28	30.020	30.520
29	0.000	0.000
30	10.430	11.500
31	1.220	0.000
32	0.000	0.000
33	27.000	25.900
34	0.000	0.000
35	42.950	39.740
36	12.520	10.140
37	0.000	0.000
38	18.190	17.800
39	9.890	9.860
40	7.000	7.450
41	0.000	0.000
42	14.530	14.010

43	14.670	13.590
44	11.960	13.100
45	11.530	10.720
46	39.700	31.410
47	0.000	0.000
48	0.000	0.030
49	0.000	0.000
50	0.000	0.000
51	0.000	0.000
52	0.000	0.000
53	0.000	0.000
54	0.000	0.000
55	0.000	0.120
56	0.000	0.000
57	0.000	0.000
58	0.000	0.000
59	0.000	0.000
60	0.000	0.000

Nota. Elaboración propia.

En la tabla 59 se muestra el resultado de la aplicación de la prueba t Student.

Tabla 59

Resultados de la prueba t Student con respecto a la cuantificación de la maleza lengua de vaca utilizando el ground truth y la predicción de la aplicación web

Medida	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
Media	5.811666667	5.513333333
Varianza	106.2409633	92.4400226
Observaciones	60	60
Coeficiente de correlación de Pearson	0.994610678	

Diferencia hipotética de las medias	0
Grados de libertad	59
Estadístico t	1.857297129
P(T<=t) una cola	0.034131094
Valor crítico de t (una cola)	1.671093032
P-valor	0.068262188
Valor crítico de t (dos colas)	2.000995378

Nota. Elaboración propia.

Para poder rechazar la hipótesis nula se debe cumplir la siguiente condición:

$$P - \text{valor} \leq \alpha$$

Remplazando las variables con los resultados obtenidos se tiene que:

$$0.068 \leq 0.05$$

Como la condición no se cumple se puede decir que no existe una diferencia significativa, por lo tanto, para las predicciones de la maleza lengua de vaca los resultados sí serían confiables.

El coeficiente de determinación, también conocido como R^2 es una medida estadística denominada bondad del ajuste que permite determinar que tan bien se ajustan los datos a un modelo de regresión, su rango está denotado desde [0,1], si el resultado se acerca a 1 la fuerza de ajuste será mayor, por lo tanto, los datos serán se acoplarán mejor al modelo (Colin Cameron & Windmeijer, 1997).

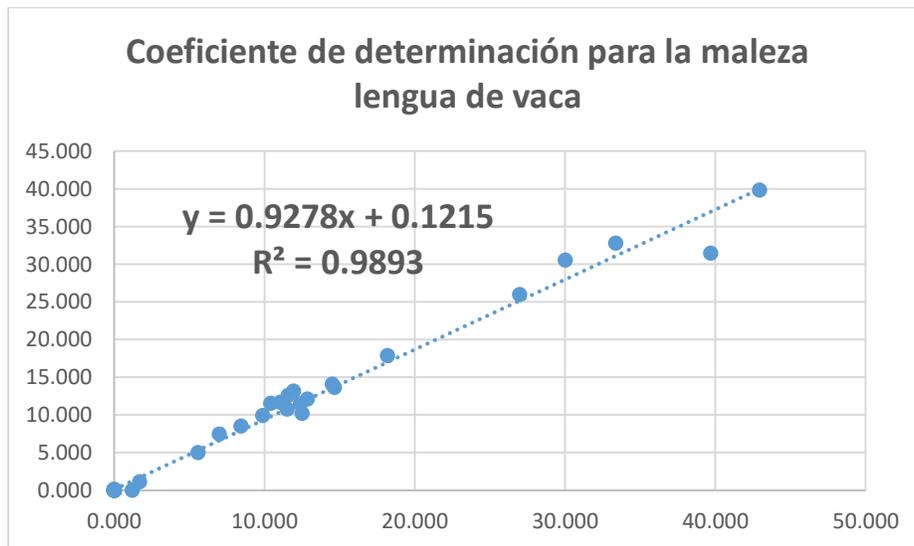
Lind et al. (2012) mencionan que los rangos para la interpretación del coeficiente de determinación son:

- **Bajo:** $0 \leq R^2 < 0.3$.
- **Moderado:** $0.3 \leq R^2 \leq 0.7$.
- **Alto:** $0.7 < R^2 \leq 1$.

A continuación, en la figura 81 se muestra la gráfica R^2 para la maleza lengua de vaca.

Figura 81

Gráfica del coeficiente de determinación para la maleza lengua de vaca



Nota. Elaboración propia.

El resultado de R^2 es 0.9893, haciendo uso de los rangos de interpretación se puede afirmar que la fuerza de ajuste es alta, por lo tanto, la confiabilidad de la aplicación web para identificar y cuantificar la maleza lengua de vaca es alta.

3.3.3 Prueba t Student y coeficiente de determinación aplicada a la cuantificación de la maleza diente de león

A continuación, en la tabla 60 se muestra la cuantificación de la maleza diente de león utilizando el ground truth y la predicción de la aplicación web.

Tabla 60

Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada a la maleza diente de león

Número de imagen	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
1	5.350	5.550
2	4.600	7.620
3	1.630	1.810
4	4.950	6.330

5	8.640	8.640
6	4.350	4.230
7	0.000	0.000
8	0.000	0.000
9	0.000	0.000
10	0.000	0.000
11	0.000	0.000
12	0.000	0.000
13	0.000	0.000
14	0.000	0.000
15	0.000	0.000
16	0.000	0.000
17	0.000	0.000
18	0.000	0.000
19	0.000	0.000
20	0.000	0.000
21	0.000	0.000
22	0.000	0.000
23	0.000	0.000
24	0.000	0.000
25	0.000	0.000
26	0.000	0.000
27	0.000	0.000
28	0.000	0.000
29	0.000	0.000
30	0.000	0.000
31	0.000	0.000
32	0.000	0.000
33	0.000	0.000
34	0.000	0.000
35	0.000	0.000

36	0.000	0.000
37	0.000	0.000
38	0.000	0.000
39	0.000	0.000
40	0.000	0.000
41	0.000	0.000
42	0.000	0.000
43	0.000	0.000
44	0.000	0.000
45	0.000	0.000
46	0.000	0.000
47	6.980	6.770
48	0.870	2.650
49	7.060	1.140
50	0.000	0.000
51	0.000	0.000
52	0.000	0.730
53	0.000	0.000
54	0.000	0.000
55	0.000	1.520
56	0.000	0.000
57	0.000	0.000
58	0.000	0.000
59	0.000	0.000
60	0.000	0.000

Nota. Elaboración propia.

En la tabla 61 se muestra el resultado de la aplicación de la prueba t Student.

Tabla 61

Resultados de la prueba t Student con respecto a la cuantificación de la maleza diente de león utilizando el ground truth y la predicción de la aplicación web

Medida	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
Media	0.7405	0.783166667
Varianza	4.015838729	4.151723701
Observaciones	60	60
Coeficiente de correlación de Pearson	0.891997152	
Diferencia hipotética de las medias	0	
Grados de libertad	59	
Estadístico t	-0.351683849	
P(T<=t) una cola	0.36316416	
Valor crítico de t (una cola)	1.671093032	
P-valor	0.726328321	
Valor crítico de t (dos colas)	2.000995378	

Nota. Elaboración propia.

Para poder rechazar la hipótesis nula se debe cumplir la siguiente condición:

$$P - \text{valor} \leq \alpha$$

Reemplazando las variables con los resultados obtenidos se tiene que:

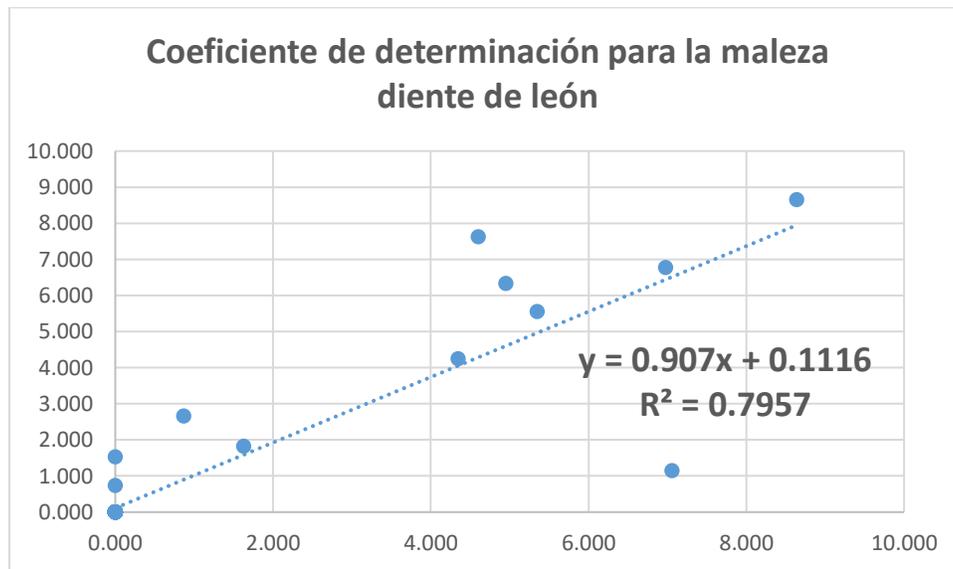
$$0.72 \leq 0.05$$

Como la condición no se cumple se puede decir que no existe una diferencia significativa, por lo tanto, para las predicciones de la maleza diente de león los resultados sí serían confiables.

A continuación, en la figura 82 se muestra la gráfica R² para la maleza diente de león.

Figura 82

Gráfica del coeficiente de determinación para la maleza diente de león



Nota. Elaboración propia.

El resultado de R^2 es 0.7957, haciendo uso de los rangos de interpretación se puede afirmar que la fuerza de ajuste es alta, por lo tanto, la confiabilidad de la aplicación web para identificar y cuantificar la maleza diente de león es alta.

3.3.4 Prueba t Student y coeficiente de determinación aplicada a la cuantificación de la maleza kikuyo

A continuación, en la tabla 62 se muestra la cuantificación de la maleza kikuyo utilizando el ground truth y la predicción de la aplicación web.

Tabla 62

Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada a la maleza kikuyo

Número de imagen	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
1	8.800	9.350
2	0.100	0.150
3	0.560	1.490

4	9.000	8.730
5	0.000	0.340
6	4.570	4.610
7	0.000	0.000
8	0.000	0.000
9	0.000	0.000
10	0.000	0.000
11	0.000	0.000
12	0.000	0.180
13	0.000	0.000
14	0.000	0.000
15	0.000	0.000
16	0.000	0.000
17	0.000	0.140
18	0.000	0.000
19	0.000	0.000
20	0.000	0.000
21	0.000	0.040
22	0.000	0.000
23	0.000	0.000
24	0.000	0.000
25	0.000	0.000
26	0.000	0.000
27	0.000	0.000
28	0.000	0.000
29	0.000	0.000
30	0.000	0.520
31	0.000	0.000
32	0.000	0.000
33	0.000	0.000
34	0.000	0.000

35	0.000	0.000
36	1.580	0.000
37	0.000	0.000
38	0.000	0.000
39	0.000	0.000
40	0.000	0.000
41	0.000	0.000
42	0.000	0.000
43	0.410	0.000
44	0.000	0.000
45	0.000	0.000
46	0.000	0.000
47	4.040	5.980
48	0.980	5.420
49	1.400	0.590
50	13.310	12.490
51	21.580	22.200
52	16.790	14.480
53	20.380	21.510
54	16.380	17.050
55	19.620	19.800
56	13.310	14.430
57	14.450	14.780
58	15.490	15.230
59	29.750	29.390
60	16.320	16.080

Nota. Elaboración propia.

En la tabla 63 se muestra el resultado de la aplicación de la prueba t Student.

Tabla 63

Resultados de la prueba t Student con respecto a la cuantificación de la maleza kikuyo utilizando el ground truth y la predicción de la aplicación web

Medida	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
Media	3.813666667	3.916333333
Varianza	52.53500328	52.84224395
Observaciones	60	60
Coeficiente de correlación de Pearson	0.993953555	
Diferencia hipotética de las medias	0	
Grados de libertad	59	
Estadístico t	-0.995932563	
P(T<=t) una cola	0.161675043	
Valor crítico de t (una cola)	1.671093032	
P-valor	0.323350087	
Valor crítico de t (dos colas)	2.000995378	

Nota. Elaboración propia.

Para poder rechazar la hipótesis nula se debe cumplir la siguiente condición:

$$P - valor \leq \alpha$$

Remplazando las variables con los resultados obtenidos se tiene que:

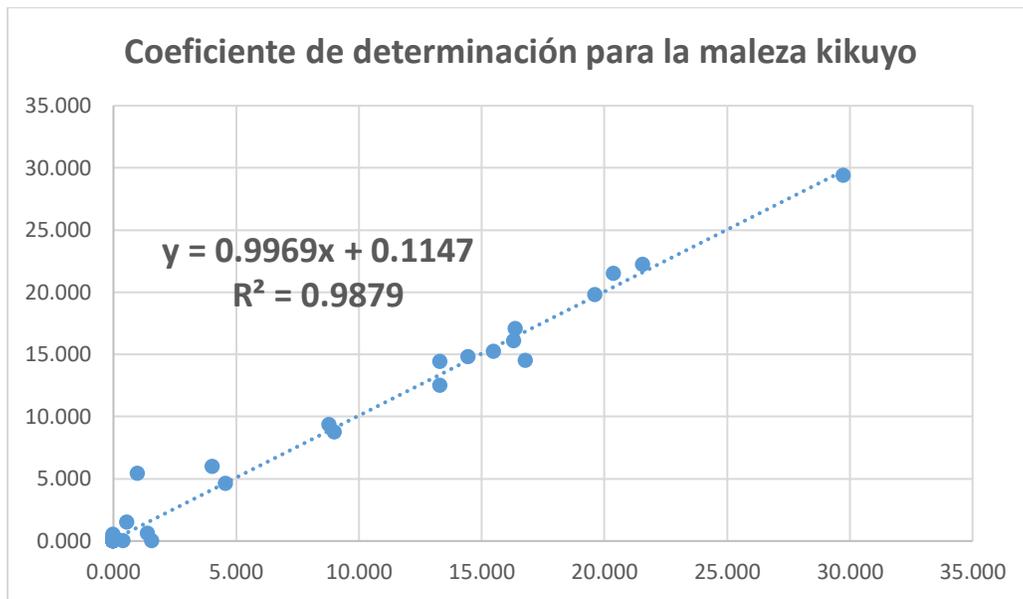
$$0.32 \leq 0.05$$

Como la condición no se cumple se puede decir que no existe una diferencia significativa, por lo tanto, para las predicciones de la maleza kikuyo los resultados sí serían confiables.

A continuación, en la figura 83 se muestra la gráfica R² para la maleza kikuyo.

Figura 83

Gráfica del coeficiente de determinación para la maleza kikuyo



Nota. Elaboración propia.

El resultado de R^2 es 0.9879, haciendo uso de los rangos de interpretación se puede afirmar que la fuerza de ajuste es alta, por lo tanto, la confiabilidad de la aplicación web para identificar y cuantificar la maleza kikuyo es alta.

3.3.5 Prueba t Student y coeficiente de determinación aplicada a la cuantificación de la categoría otro tipo de maleza

A continuación, en la tabla 64 se muestra la cuantificación de la categoría otro tipo de maleza utilizando el ground truth y la predicción de la aplicación web.

Tabla 64

Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada a la categoría otro tipo de maleza

Número de imagen	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
1	0.000	0.000
2	0.700	1.060
3	6.620	6.100

4	0.710	0.620
5	1.400	1.160
6	0.000	0.000
7	0.000	0.000
8	0.000	0.000
9	0.000	0.000
10	0.000	0.000
11	0.000	0.000
12	0.000	0.000
13	0.000	0.000
14	0.000	0.000
15	0.000	0.000
16	0.180	0.000
17	0.400	0.090
18	0.000	0.000
19	0.210	0.240
20	0.510	0.370
21	1.940	0.010
22	0.000	0.000
23	0.540	0.250
24	0.000	0.000
25	1.510	1.320
26	0.000	0.000
27	0.000	0.000
28	0.000	0.000
29	0.000	0.000
30	0.910	0.000
31	0.000	0.000
32	0.000	0.000
33	0.830	0.020
34	0.000	0.000

35	0.000	0.000
36	0.000	0.460
37	0.000	0.000
38	0.270	0.000
39	0.000	0.000
40	0.600	0.000
41	0.000	0.000
42	0.000	0.000
43	0.000	0.330
44	0.000	0.000
45	0.110	0.220
46	0.000	0.000
47	0.230	0.000
48	12.260	7.340
49	1.520	1.860
50	0.000	0.000
51	0.000	0.000
52	0.270	0.260
53	1.230	0.000
54	0.000	0.000
55	0.690	0.240
56	0.660	0.580
57	0.000	0.000
58	0.000	0.000
59	0.000	0.000
60	0.000	0.000

Nota. Elaboración propia.

En la tabla 65 se muestra el resultado de la aplicación de la prueba t Student.

Tabla 65

Resultados de la prueba t Student con respecto a la cuantificación de la categoría otro tipo de maleza utilizando el ground truth y la predicción de la aplicación web

Medida	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
Media	0.571666667	0.3755
Varianza	3.239536158	1.555547203
Observaciones	60	60
Coeficiente de correlación de Pearson	0.952896733	
Diferencia hipotética de las medias	0	
Grados de libertad	59	
Estadístico t	2.113466192	
P(T<=t) una cola	0.01939978	
Valor crítico de t (una cola)	1.671093032	
P-valor	0.038799559	
Valor crítico de t (dos colas)	2.000995378	

Nota. Elaboración propia.

Para poder rechazar la hipótesis nula se debe cumplir la siguiente condición:

$$P - valor \leq \alpha$$

Remplazando las variables con los resultados obtenidos se tiene que:

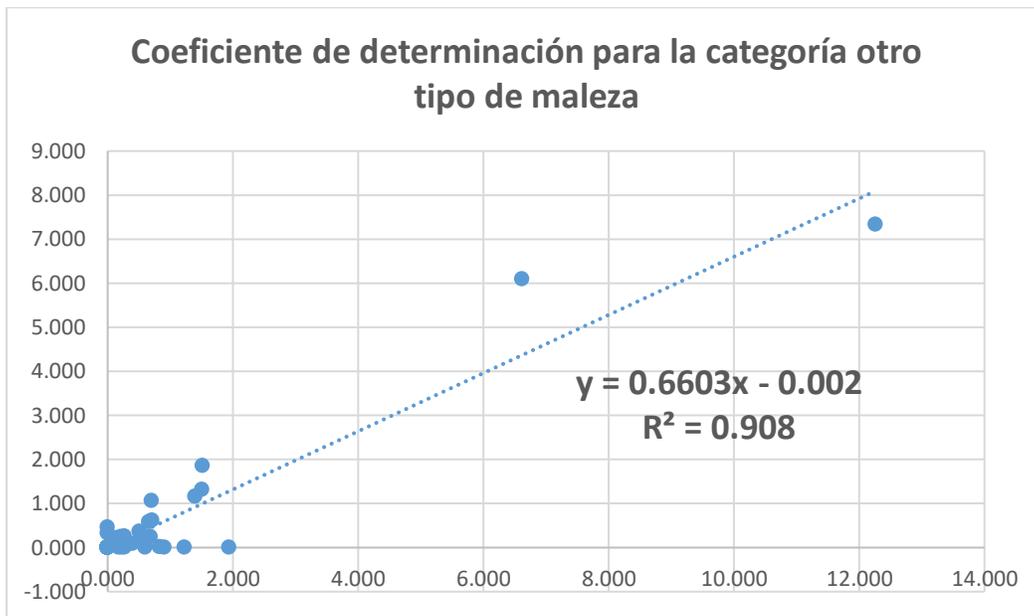
$$0.038 \leq 0.05$$

Como la condición si se cumple se puede decir que sí existe una diferencia significativa, por lo tanto, para las predicciones de la categoría otro tipo de maleza los resultados no serían muy confiables.

A continuación, en la figura 84 se muestra la gráfica R² para la categoría otro tipo de maleza.

Figura 84

Gráfica del coeficiente de correlación para la categoría otro tipo de maleza



Nota. Elaboración propia.

El resultado de R^2 es 0.908, haciendo uso de los rangos de interpretación se puede afirmar que la fuerza de ajuste es alta, sin embargo, en la prueba t Student se demostró que si existe una diferencia significativa entre el ground truth y las predicciones de la aplicación web, por lo tanto, como conclusión final se puede afirmar que la confiabilidad de las predicciones con respecto a la categoría otro tipo de malezas no es muy buena, ya que puede que en ocasiones los resultados sean erróneos, esto puede deberse a que para esta categoría hubo muy pocas imágenes, además, en el capítulo 2 se mencionó que la categoría otro tipo de malezas es para aquellas que son muy difíciles de identificar y para las que no sean diente de león, kikuyo y lengua de vaca, esto produjo que ResNeXt50 no encontrará los patrones adecuados para clasificar esta categoría.

3.3.6 Prueba t Student y coeficiente de determinación aplicada a la cuantificación del cultivo de papa

A continuación, en la tabla 66 se muestra la cuantificación del cultivo de papa utilizando el ground truth y la predicción de la aplicación web.

Tabla 66

Cuantificación utilizando el ground truth y la predicción de la aplicación web de 60 imágenes de 128px por 128px enfocada al cultivo de papa

Número de imagen	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
1	13.650	13.720
2	37.190	33.090
3	22.510	22.550
4	24.280	25.080
5	0.000	0.000
6	10.550	10.670
7	51.750	54.600
8	9.330	9.190
9	52.860	54.420
10	63.580	63.380
11	6.990	6.840
12	20.170	19.960
13	52.200	53.280
14	0.000	0.570
15	35.680	35.720
16	37.480	39.010
17	0.000	0.990
18	50.690	51.210
19	0.000	0.000
20	27.480	28.660
21	0.000	0.000
22	34.770	33.640
23	0.000	0.000
24	14.960	14.800
25	11.630	11.410
26	17.690	17.350

27	52.760	52.560
28	1.310	1.080
29	30.590	30.260
30	0.000	0.000
31	8.590	9.450
32	16.570	17.920
33	0.000	0.530
34	31.120	31.950
35	15.220	18.030
36	0.380	1.130
37	31.850	33.090
38	1.720	0.570
39	0.000	0.290
40	6.900	7.460
41	41.940	43.640
42	0.000	0.000
43	7.540	8.630
44	0.000	0.000
45	1.710	1.620
46	0.270	6.450
47	5.130	5.730
48	10.190	10.100
49	16.960	22.680
50	0.760	0.480
51	0.000	0.000
52	0.000	0.000
53	0.670	0.850
54	0.800	0.680
55	16.820	14.930
56	4.630	4.790
57	4.360	4.270

58	0.000	0.000
59	1.820	1.820
60	1.350	1.290

Nota. Elaboración propia.

En la tabla 67 se muestra el resultado de la aplicación de la prueba t Student.

Tabla 67

Resultados de la prueba t Student con respecto a la cuantificación del cultivo de papa utilizando el ground truth y la predicción de la aplicación web

Medida	Cuantificación utilizando el ground truth	Cuantificación utilizando la predicción de la aplicación web
Media	15.12333333	15.54033333
Varianza	315.6116972	320.8208711
Observaciones	60	60
Coeficiente de correlación de Pearson	0.996831943	
Diferencia hipotética de las medias	0	
Grados de libertad	59	
Estadístico t	-2.262883632	
P(T<=t) una cola	0.013668842	
Valor crítico de t (una cola)	1.671093032	
P-valor	0.27337683	
Valor crítico de t (dos colas)	2.000995378	

Nota. Elaboración propia.

Para poder rechazar la hipótesis nula se debe cumplir la siguiente condición:

$$P - \text{valor} \leq \alpha$$

Remplazando las variables con los resultados obtenidos se tiene que:

$$0.27 \leq 0.05$$

cálculo de manera objetiva, por el contrario de lo que se hace utilizando los métodos tradicionales que se basan en la perspectiva subjetiva de cada agricultor.

DISCUSIÓN

Se ha propuesto una versión de ResNeXt50 enfocada a la segmentación semántica haciendo uso de la propuesta de Mou & Zhu (2018), esta versión fue el base line, a partir de esta se fueron haciendo modificaciones a la arquitectura a fin de hacerla más robusta, posteriormente se probaron combinaciones de los valores de los hiperparámetros utilizados por otros investigadores enfocados a la segmentación semántica de malezas, el resultado fue la propuesta de 6 versiones, de las cuales se hizo la elección de la más idónea para implementarla en la aplicación web, la cual es un instrumento para realizar la identificación y cuantificación automática de malezas utilizando imágenes adquiridas por un dron.

Se utilizó el 20% del conjunto de validación para realizar las pruebas para calcular el IOU de cada categoría con el fin de definir la mejor versión, en la tabla 68 se muestra cuáles fueron los resultados.

Tabla 68

IOU de cada categoría de todas las versiones del modelo entrenadas utilizando el 20% del conjunto de validación

Versión	IOU del background	IOU de lengua de vaca	IOU del diente de león	IOU del kikuyo	IOU de otras malezas	IOU de papa	Mean IOU (6 clases)
Base line	0.911.	0.711.	0.505.	0.536.	0.009.	0.834.	0.584.
Versión 2.	0.952.	0.836.	0.476.	0.658.	0.124.	0.901.	0.658.
Versión 3.	0.956.	0.769.	0.478.	0.671.	0.242.	0.888.	0.668.
Versión 4.	0.949.	0.577.	0.460.	0.546.	0.003.	0.817.	0.559.
Versión 5.	0.946.	0.694.	0.580.	0.596.	0.031.	0.853.	0.617.
Versión 6.	0.956.	0.826.	0.583.	0.703.	0.434.	0.908.	0.735.

Nota. Elaboración propia.

En base a los resultados obtenidos se decidió definir a la sexta versión como el mejor modelo, ya que esta es la que presenta mejores valores en comparación con las demás versiones, la versión 6 presenta un valor de IOU del background de 0.956, un IOU de lengua de vaca de 0.826, un IOU del diente de león de 0.583, un IOU del kikuyo de 0.703, un IOU de otras malezas de 0.434, un IOU de papa de 0.908 y un mean IOU de 0.735. La versión 6 es la que tiene mayores capacidades para la generalización, por lo que su precisión al procesar imágenes nunca vistas será mayor y fue la que se implementó en la aplicación web.

Se hizo una búsqueda de trabajos de investigación enfocados a la segmentación semántica de malezas, en total se encontraron 6 propuestas, de las cuales 5 se enfocan únicamente en 3 clases, que son background, cultivo y maleza, solo 1 trabajo utilizó 4 clases, que son las mencionadas anteriormente y una adicional enfocada a detectar daños en el cultivo. A continuación, en las tablas 69 y 70 se muestra los resultados reportados por los investigadores del IOU de cada clase.

Tabla 69

IOU de cada clase reportado por los trabajos de investigación enfocados en 3 clases

Autores	Modelo	Cultivo	Número de malezas	IOU del background	IOU del cultivo	IOU de la maleza	Mean IOU(3 clases)
(H. S. Ullah et al., 2021)	U-Net modificada.	Canola.	1 (Sin especificar).	0.982	0.861	0.829	0.891
(Qial., 2023)	PF-UperNet.	Frijol.	1 (Sin especificar).	0.969	0.797	0.856	0.874
(Celikkan et al., 2023)	DeepLabv3.	Remolacha azucarera.	1 (Sin especificar).	0.993	0.946	0.633	0.857
(X.-Z. Hu et al., 2022)	UNet++ con supervisión	Remolacha azucarera.	1 (Sin especificar).	0.921	0.925	0.956	0.923

	profund a.							
(Sai et al., 2022)	Kite-Net.	Arroz.	1 (Sin especificar).	No especificado	No especificado.	No especificado	No especificado	0.725

Nota. Elaboración propia.

Tabla 70

IOU de cada clase reportado por el trabajo de investigación enfocado en 4 clases

Autores	Modelo	Cultivo	Número de malezas	IOU del background	IOU del cultivo	IOU de la maleza	IOU del daño del cultivo	Mean IOU (4 clases)
(Das Bais, 2021)	& DeepVe g.	Canola	1 (Sin especificar).	0.981	0.743	0.657	0.689	0.767

Nota. Elaboración propia.

Como se puede observar solo 1 trabajo de investigación reporta que utilizaron 4 clases, del resto solo se limitaron a 3, para poder dar un criterio de los valores obtenidos por ResNeXt50 versión 6 en comparación con las propuestas de los investigadores se ha decidido realizar un promedio de las malezas lengua de vaca, kikuyo y diente de león, no se tomó en cuenta a la categoría otro tipo de malezas debido a que estas presentan un valor de IOU muy bajo, además es la que menos individuos(plantas) presenta en todo el data set. A continuación, en la tabla 71 se muestra una tabla de la comparativa.

Tabla 71

Comparativa de modelo ResNeXt50 versión 6 y los modelos de los trabajos de investigación utilizando el IOU del background y cultivo

Autores	Modelo	IOU del background	IOU del cultivo	IOU de la maleza
(H. S. Ullah et al., 2021).	U-Net modificada.	0.982.	0.861.	0.829.

(Qi et al., 2023).	PF-UperNet.		0.969.	0.797.	0.856.
(Celikkan et al., 2023).	DeepLabv3.		0.993.	0.946.	0.633.
(X.-Z. Hu et al., 2022).	UNet++ con supervisión profunda.		0.921.	0.925.	0.956.
(Sai et al., 2022).	Kite-Net.		No especificado	No especificado.	No especificado.
(Das & Bais, 2021).	DeepVeg.		0.981.	0.743.	0.657.
Kevin Vinueza (autor del trabajo de titulación y de ResNeXt50 modificada).	ResNeXt50 modificada (versión 6).		0.956.	0.908.	0.704. (promedio de las 3 clases de malezas lengua de vaca, kikuyo y diente de león)

Nota. Elaboración propia.

Como se puede observar en la tabla 71 ResNeXt50 en su versión 6 con respecto al IOU del background está un poco por debajo de los modelos U-Net modificada, PF-UperNet, DeepLabv3 y DeepVeg, siendo inferior en 0.026, 0.013, 0.037 y 0.025 respectivamente, la diferencia no es muy significativa. El modelo versión 6 es superior al modelo UNet++ con supervisión profunda la cual presenta un IOU del background de 0.921, la superioridad es de 0.035. Como se puede evidenciar el valor obtenido por ResNext50 en su versión 6 está muy a la par que los modelos de los demás investigadores.

En el IOU del cultivo el modelo versión 6 es inferior a los modelos DeepLabv3 y UNet++ con supervisión profunda con 0.038 y 0.017 respectivamente, pero, es superior a los modelos U-Net modificada, PF-UperNet y DeepVeg con 0.047, 0.111 y 0.165 respectivamente. Con respecto al IOU del cultivo se puede decir que ResNeXt50 en su versión 6 es superior a 3 modelos con diferencias significativas para 2 de ellos, específicamente para PF-UperNet y DeepVeg.

Con respecto al IOU de la maleza ResNeXt50 en su versión 6 es superada con gran diferencia por U-Net modificada, PF-UperNet, y UNet++ con supervisión profunda con 0.125,

0.152 y 0.252 respectivamente. En este aspecto se puede observar que la diferencia de resultados es muy significativa para los 3 modelos mencionados anteriormente, el que mayor puntuación presenta es U-Net modificada con 0.956, esta enorme brecha se puede justificar que es debido a que estos modelos fueron entrenados únicamente con 3 clases, a diferencia de las 6 que se utilizó en este trabajo de titulación, los autores únicamente definieron a la clase maleza sin hacer distinción de tipos, cosa que en este trabajo si se hizo definiendo un total de 4 que son: lengua de vaca, kikuyo, diente de león y la categoría otras. Haber establecido muchas categorías de malezas hizo que existieran más clases, además el data set no contaba con un balance de clases idóneo debido a la naturaleza del proyecto. Sin embargo, a pesar de lo mencionado anteriormente ResNeXt50 en su versión 6 supera a 2 modelos, concretamente a DeepLabv3 y DeepVeg, siendo superior con 0.071 y 0.047 respectivamente.

En el trabajo de Das & Bais (2021) hacen mucho énfasis en que el desbalance de clases perjudica enormemente los resultados, el entrenamiento lo hicieron con un data set el cual no tenía un balance de clases idóneo, además, fueron los únicos que definieron 4 clases y esa es la razón del porque obtuvieron un valor de 0.767 en el mean IOU, cabe recalcar que el mean IOU de ResNeXt50 en su versión 6 es de 0.735, que está un poco por debajo.

El data set con el que se entrenó las 6 versiones propuestas no estaba equilibrado, la categoría lengua de vaca representa un 14.43% con respecto al número de individuos (plantas) presentes en todo el data set, kikuyo un 39.41%, diente de león un 4.73%, papa un 26.69% y la categoría otras un 14.71%. Con esto se puede dar una justificación del porque se obtuvo valores bajos en el IOU de las malezas ya que las características del data set son similares al de los autores Das & Bais (2021) con respecto al desbalance de clases.

A continuación, en la tabla 72 se muestra el mean IOU de ResNeXt50 versión 6 y de los modelos de los autores.

Tabla 72*Mean IOU de los trabajos de investigación*

Autores	Modelo	Mean IOU
(H. S. Ullah et al., 2021)	U-Net modificada.	0.891.
(Qi et al., 2023)	PF-UperNet.	0.874.
(Celikkan et al., 2023)	DeepLabv3.	0.857.
(X.-Z. Hu et al., 2022)	UNet++ con supervisión profunda.	0.923.
(Sai et al., 2022)	Kite-Net.	0.725.
(Das & Bais, 2021)	DeepVeg.	0.767.
Kevin Vinueza (autor del trabajo de titulación y de ResNeXt50 modificada)	ResNeXt50 modificada (versión 6).	0.735.

Nota. Elaboración propia.

Para ResNeXt50 se evidencia que es significativamente inferior a los modelos U-Net modificada, PF-UperNet, DeepLabv3, UNet++ con supervisión profunda y un poco por debajo de DeepVeg, sin embargo, es un poco superior la Kite-Net con 0.010 a su favor. Como se hizo mucho énfasis en los párrafos anteriores, haber entrenado a ResNeXt50 con 6 clases y presentando un desbalance de clases en el data set conllevó a que se obtengan los resultados presentados, DeepVeg tiene un valor un poco superior en el mean IOU, este fue entrenado con un data set en similares condiciones de desbalance de clases, con la única diferencia de que se definieron únicamente 4 clases, si se hubiera optado únicamente por 3 clases muy probablemente los resultados hubieran sido muy similares a los modelos con mejor puntaje como por ejemplo UNet++ con supervisión profunda y U-Net modificada.

Para trabajos futuros se aconseja que se priorice el balance de clases, ya que esto permitiría que el modelo pueda aprender de mejor manera teniendo un equilibrio en los patrones para cada clase, se puede utilizar el data set elaborado en este trabajo de titulación e intentar

balancear las clases, especialmente para la categoría otro tipo de malezas y diente de león, además también aumentar el número de imágenes a fin de tener más variedad.

También se aconseja buscar más trabajos de investigación para poder tener un campo visual más amplio acerca de los principales desafíos en este contexto y tomar muy en cuenta las propuestas de los autores para hacerles frente, se aconseja probar más combinaciones en los hiperparámetros y en la medida de lo posible hacer más modificaciones a la arquitectura a fin de obtener mejores resultados.

Lo más recomendable para entrenar este tipo de algoritmos es utilizar en la medida de lo posible GPUs dedicadas ya que esto permitirá reducir el tiempo de entrenamiento y lo más importante, tener la posibilidad de realizar más pruebas con el fin de buscar la mejor combinación de los hiperparámetros y modificaciones en la arquitectura.

CONCLUSIONES

- El desarrollo de una red neuronal convolucional basada en la arquitectura ResNeXt para segmentación semántica ha permitido realizar la identificación y cuantificación automática efectiva de malezas en cultivos de papa mediante imágenes capturadas por un dron manteniendo una confiabilidad muy alta en los resultados.
- La búsqueda de trabajos de investigación enfocados a la identificación de malezas mediante segmentación semántica utilizando redes neuronales convolucionales ha sido muy útil para el desarrollo de este trabajo, ya que se tuvo un panorama muy claro de cuáles son los principales retos que existen en esta área, las arquitecturas más utilizadas, las propuestas de cada autor para mejorar la precisión de los algoritmos, entre otros. Las preguntas de investigación más relevantes que ayudaron a desarrollar este proyecto fue la recopilación de las alturas a las que se capturaban las imágenes, esto permitió elegir la altura más adecuada, que fue de 9 metros, al igual que la velocidad de vuelo, la cual correspondía a 1 m/s, esta altura permitió abarcar más área de terreno en una foto y mantener un nivel de calidad de imagen alto. Otra pregunta muy importante fue determinar que métricas se utilizan para la evaluación de los modelos, esto permitió identificar aquellas que eran utilizadas para segmentación semántica. Finalmente, la pregunta que se considera la más importante es aquella en la que se plantea como afinar un modelo, en esta se responde que es adecuado trabajar con tamaños de lote de 2^n ya que según los investigadores da muy buenos resultados, otra es utilizar imágenes en formato JPG ya que según los autores se dice tiene mejores resultados en redes que utilizan la arquitectura ResNet.
- La red neuronal convolucional ResNeXt50 pasó por varias versiones propuestas en las cuales se hacían cambios en la arquitectura y se probaban distintas combinaciones de los hiperparámetros utilizados por otros autores, esto ayudó a definir un modelo base y

en consecuencia surgieron más versiones hasta llegar al mejor modelo. La aplicación web simplifica la identificación y cuantificación de malezas utilizando imágenes adquiridas por un dron ya que dispone de una interfaz con las funcionalidades necesarias para llevar a cabo el proceso, además, se priorizó el aspecto de la portabilidad con la finalidad que pueda ser migrada o desplegada sin muchas complicaciones.

- Las versiones propuestas del modelo fueron analizadas a fin de elegir la más idónea para implementarla en la aplicación web, la mejor versión fue la sexta ya que tiene mayor capacidad de generalización a nuevos datos lo que da una mayor confiabilidad en las predicciones en comparación con las demás versiones, además, se hizo una comparación contra las propuestas de otros investigadores a fin de determinar el nivel de semejanza en los resultados de las pruebas, se llegó a la conclusión de que los resultados obtenidos por ResNeXt50 son buenos tomando en cuenta que se decidió clasificar a las malezas en cuatro tipos, cosa que los demás investigadores no hicieron y solo se limitaron a utilizar la categoría maleza para todas las malas hierbas. La reducción del tiempo utilizando la aplicación web es evidente en comparación con el tiempo empleado en realizar la identificación y cuantificación de forma manual, esto supone que las decisiones que se tomen para el control de malezas sean más efectivas ya que los resultados se obtendrán en menos tiempo y lo más importante de forma objetiva, además su uso es viable ya que al aplicar la prueba t Student y el coeficiente de determinación los resultados demostraron que el nivel de confianza para las predicciones es alta. La aplicación web tiene un nivel de portabilidad muy alto, esto se debe a que se utilizó la tecnología de contenedores docker, lo que facilita el despliegue o migración sin tener que hacer configuraciones de entorno o de librerías.

RECOMENDACIONES

- Se debería probar con otro tipo de arquitecturas como por ejemplo U-Net, SegNet, DeepLab y PSPNet por mencionar algunas a fin de mejorar los resultados de las métricas.
- La búsqueda de trabajos de investigación es fundamental para realizar un trabajo que siga el mismo propósito, por esta razón se debería priorizar los más recientes ya que esto significaría en la mayoría de las ocasiones que se están tomando propuestas nuevas, se aconseja optar por trabajos que no sean más antiguos a 5 años. Sería aconsejable aumentar más preguntas de investigación ya que esto permitiría tener un conocimiento más amplio de cuáles son los principales retos que se tiene en esta área y cuáles son las propuestas para hacerles frente.
- Se deberían hacer más propuestas para la búsqueda del mejor modelo, intentando con más combinaciones de los hiperparámetros y proponiendo más modificaciones a la arquitectura. La aplicación web al estar destinada a usuarios que en la mayoría de los casos tienen nulos conocimientos tecnológicos debería priorizar también la facilidad de uso.
- Para obtener mejores resultados se aconseja aumentar el número de imágenes del data set priorizando el equilibrio de clases, las imágenes deberían tomarse en más terrenos, en distintas épocas y en distinto clima a fin tener una gran variedad de escenarios, se debería optar por realizar el entrenamiento utilizando GPU dedicada ya que esto reduciría el tiempo del entrenamiento y por ende permitiría hacer más pruebas ya sea probando distintos valores de los hiperparámetros y modificaciones a la arquitectura. Para facilitar el despliegue o migración de la aplicación web se debería optar por servicios cloud como Azure, AWS o Google Cloud por poner unos ejemplos, ya que estos tienen muy bien automatizado el proceso con respecto a docker. Se debería optar en la medida de lo

posible por elegir servicios que ofrezcan GPUs, ya que son más rápidas en comparación con CPUs, esto reduciría el tiempo para realizar la identificación y cuantificación de malezas.

BIBLIOGRAFÍA

- Adobe. (2024). *Formación en Photoshop*.
<https://creativecloud.adobe.com/es/learn/app/photoshop>
- Alagbo, O. O., Akinyemiju, O. A., & Chauhan, B. S. (2022). Weed Management in Rainfed Upland Rice Fields under Varied Agro-Ecologies in Nigeria. *Rice Science*, 29(4), 331.
<https://doi.org/10.1016/j.rsci.2021.11.004>
- Al-Musawi, N. A., Al-Shammary, D., & Al-foudi, A. S. (2023). A Survey on Web Service approaches targeting network traffic reduction. *Procedia Computer Science*, 218, 1033–1035. <https://doi.org/10.1016/j.procs.2023.01.083>
- Alonso Arce, F. (2008). *El cultivo de la patata (2a. ed.)* (2nd ed., p. 17). Mundi-Prensa.
<https://elibro.net/es/ereader/utnorte/35825>
- Angular. (2022, February 28). *Introduction to Angular concepts*.
<https://angular.io/guide/architecture#introduction-to-angular-concepts>
- Antonopoulos, N., Kanatas, P., Gazoulis, I., Tataridas, A., Ntovakos, D., Ntaoulis, V., Zavra, S.-M., & Travlos, I. (2023). Hot foam: Evaluation of a new, non-chemical weed control option in perennial crops. *Smart Agricultural Technology*, 3, 1–2.
<https://doi.org/10.1016/j.atech.2022.100063>
- Aquilani, C., Confessore, A., Bozzi, R., Sirtori, F., & Pugliese, C. (2022). Review: Precision Livestock Farming technologies in pasture-based livestock systems. *Animal*, 16(1), 1–2.
<https://doi.org/10.1016/j.animal.2021.100429>
- Arun, R. A., Umamaheswari, S., & Jain, A. V. (2020). Reduced U-Net Architecture for Classifying Crop and Weed using Pixel-wise Segmentation. *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 1–6.
<https://doi.org/10.1109/INOCON50539.2020.9298209>
- Asad, M. H., & Bais, A. (2020). Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network. *Information Processing in Agriculture*, 7(4), 535–545. <https://doi.org/10.1016/j.inpa.2019.12.002>

- Babaei-Ghaghelestany, A., Alebrahim, M. T., Farzaneh, S., & Mehrabi, M. (2022). The anticancer and antibacterial properties of aqueous and methanol extracts of weeds. *Journal of Agriculture and Food Research*, 10, 2. <https://doi.org/10.1016/j.jafr.2022.100433>
- Badillo, A. (2016). *Evaluación del aporte de gallinaza fresca en el rendimiento de cultivo de maíz (Zea mais) variedad INIAP 122, en dosis diferentes, en la provincia de malchinguí, cantón Pedro Moncayo, provincia de Pichincha* [Universidad Nacional de Loja]. <https://dspace.unl.edu.ec/jspui/handle/123456789/10735>
- Bah, M. D., Hafiane, A., Canals, R., & Emile, B. (2019). Deep features and One-class classification with unsupervised data for weed detection in UAV images. *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 1–5. <https://doi.org/10.1109/IPTA.2019.8936091>
- Balseca, E. (2014). *Evaluación de calidad de productos software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000* [Escuela Politécnica Nacional]. <http://bibdigital.epn.edu.ec/handle/15000/9113>
- Bautista, A. (2015). *Evaluación de la aplicación de cuatro tipos de abonos orgánicos, en la productividad del cultivo de papa Solanum tuberosum, variedad chola, en San Agustín, Parroquia Pintag, cantón Quito, provincia Pichincha* [Universidad Nacional de Loja]. <https://dspace.unl.edu.ec/jspui/bitstream/123456789/17323/1/TESIS%20PAPA%20PATHY%201.pdf>
- Beeharry, Y., & Bassoo, V. (2020). Performance of ANN and AlexNet for weed detection using UAV-based images. *2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM)*, 163–167. <https://doi.org/10.1109/ELECOM49001.2020.9296994>
- Binguitcha-Fare, A.-A., & Sharma, P. (2019). Crops and weeds classification using Convolutional Neural Networks via optimization of transfer learning parameters. *International Journal of Engineering and Advanced Technology (IJEAT)*, 8(5), 2284–2293.
- Bosilj, P., Aptoula, E., Duckett, T., & Cielniak, G. (2020). Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. *Journal of Field Robotics*, 37(1), 7–19. <https://doi.org/10.1002/rob.21869>

- Cárdenas, J. (1987). *Manual de control de malezas en papa* (Vol. 9, p. 28). INIAP, Estación Experimental Santa Catalina, Departamento de Control de Malezas. <http://repositorio.iniap.gob.ec/jspui/handle/41000/808>
- Catillo, H., & Gómez, B. (2018). *Desarrollo de una aplicación arquitectura cliente servidor para la automatización del flujo de trabajo del mantenimiento de partes y componentes en flotas de vehículos y equipos de transporte de carga pesada* [Universidad de Guayaquil]. <http://repositorio.ug.edu.ec/handle/redug/34219>
- Celikkan, E., Saberioon, M., Herold, M., & Klein, N. (2023). Semantic Segmentation of Crops and Weeds with Probabilistic Modeling and Uncertainty Quantification. *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 582–592. <https://doi.org/10.1109/ICCVW60793.2023.00065>
- Chaucha de la Cruz, Eijman, R., & Cesar, J. (2021). *Arquitectura cliente servidor en una red de telecomunicaciones 802.11N para la actualización de la base de datos del registro de atención del centro de salud del distrito de Colcabamba* [Universidad Nacional de Huancavelica]. <http://repositorio.unh.edu.pe/handle/UNH/4178>
- Chavarria, B., & Gudiño, E. (2017). *Implementación de un servidor web y un diseño de una página utilizando herramientas de software libre para el dispensario “Sagrada Familia” de la ciudad de Guayaquil* [Universidad Politécnica Salesiana Sede Guayaquil]. <https://dspace.ups.edu.ec/bitstream/123456789/14162/1/GT001840.pdf>
- Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., & Stachniss, C. (2017). Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research*, 36(10), 1045–1052. <https://doi.org/10.1177/0278364917720510>
- Cisternas, I., Velásquez, I., Caro, A., & Rodríguez, A. (2020). Systematic literature review of implementations of precision agriculture. *Computers and Electronics in Agriculture*, 176, 1–2. <https://doi.org/10.1016/j.compag.2020.105626>
- Colin Cameron, A., & Windmeijer, F. A. G. (1997). An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*, 77(2), 329–342. [https://doi.org/10.1016/S0304-4076\(96\)01818-0](https://doi.org/10.1016/S0304-4076(96)01818-0)

- Cóndor, B. (2018). *Identificación de papas producidas y cultivadas en la provincia de Tungurahua: sus características y sugerencia de usos en la cocina diaria* [Universidad San Francisco de Quito USFQ]. <https://repositorio.usfq.edu.ec/bitstream/23000/7756/1/140374.pdf>
- Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D., & Traore, D. (2022). Deep Convolution Neural Network sharing for the multi-label images classification. *Machine Learning with Applications*, 10, 2. <https://doi.org/10.1016/j.mlwa.2022.100422>
- Czymbek, V., Harders, L. O., Knoll, F. J., & Hussmann, S. (2019). Vision-Based Deep Learning Approach for Real-Time Detection of Weeds in Organic Farming. *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 1–5. <https://doi.org/10.1109/I2MTC.2019.8826921>
- Das, M., & Bais, A. (2021). DeepVeg: Deep Learning Model for Segmentation of Weed, Canola, and Canola Flea Beetle Damage. *IEEE Access*, 9, 119367–119380. <https://doi.org/10.1109/ACCESS.2021.3108003>
- Deras, H. (2020). *Guía técnica El cultivo del maíz* (M. Sorto, N. Menjívar, & L. Reyes, Eds.; pp. 9–18). IICA. <http://repiica.iica.int/docs/b3469e/b3469e.pdf>
- DJI. (2018, August 23). *MAVIC 2 Specs*. <https://www.dji.com/global/mavic-2/info>
- Docker. (2024a). *Overview of Docker Build*. <https://docs.docker.com/build/>
- Docker. (2024b). *Overview of Docker Hub*. <https://docs.docker.com/docker-hub/>
- DroneDeploy. (2024). *La guía de hardware de drones y robots 2024*. <https://dronedeploy.com/resources/ebooks/the-drone-and-robot-hardware-guide-2024>
- Ellis, G. (2020). Workflow—Kanban and Kamishibai: Just-In-Time Rationalization. In *Improve* (p. 409). Elsevier. <https://doi.org/10.1016/B978-0-12-809519-5.00021-1>
- Espejo-Garcia, B., Mylonas, N., Athanasakos, L., & Fountas, S. (2020). Improving weeds identification with a repository of agricultural pre-trained deep neural networks. *Computers and Electronics in Agriculture*, 175, 1–7. <https://doi.org/10.1016/j.compag.2020.105593>

- Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S., & Vasilakoglou, I. (2020). Towards weeds identification assistance through transfer learning. *Computers and Electronics in Agriculture*, 171, 1–10. <https://doi.org/10.1016/j.compag.2020.105306>
- Estrada, M. (2020). *Efecto de tres programas de nutrición en el cultivo de maíz (Zea mays L.), en tres híbridos Emblema, Advanta y Gladiador, en el Cantón Montalvo, Provincia Los Ríos*. [Universidad Católica de Santiago de Guayaquil]. <http://repositorio.ucsg.edu.ec/bitstream/3317/15555/1/T-UCSG-PRE-TEC-AGRO-176.pdf>
- Express. (2017). *Express Fast, unopinionated, minimalist web framework for Node.js*. <https://expressjs.com/>
- Farooq, A., Hu, J., & Jia, X. (2018). Weed Classification in Hyperspectral Remote Sensing Images Via Deep Convolutional Neural Network. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 3816–3819. <https://doi.org/10.1109/IGARSS.2018.8518541>
- Farooq, A., Jia, X., Hu, J., & Zhou, J. (2019). Knowledge Transfer via Convolution Neural Networks for Multi-Resolution Lawn Weed Classification. *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 1–5. <https://doi.org/10.1109/WHISPERS.2019.8920832>
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). *From Data Mining to Knowledge Discovery in Databases* (© AAAI) (Vol. 17). www.ffly.com/
- Felício, D., Simão, J., & Datia, N. (2023). RapiTest: Continuous Black-Box Testing of RESTful Web APIs. *Procedia Computer Science*, 219, 538. <https://doi.org/10.1016/j.procs.2023.01.322>
- Fernández-Arteaga, V., Tovilla-Zárate, C. A., Fresán, A., González-Castro, T. B., Juárez-Rojop, I. E., López-Narváez, L., & Hernández-Díaz, Y. (2016). Association between completed suicide and environmental temperature in a Mexican population, using the Knowledge Discovery in Database approach. *Computer Methods and Programs in Biomedicine*, 135, 220. <https://doi.org/10.1016/j.cmpb.2016.08.002>

- Fishkis, O., & Koch, H.-J. (2023). Effect of mechanical weeding on soil erosion and earthworm abundance in sugar beet (*Beta vulgaris* L.). *Soil and Tillage Research*, 225, 1–2. <https://doi.org/10.1016/j.still.2022.105548>
- Gabela, F., & Cárdenas, J. (1979). *Control de malezas en maíz de la sierra* (Vol. 105, p. 9). INIAP, Estación Experimental Santa Catalina, Departamento de Control de Malezas. <http://repositorio.iniap.gob.ec/handle/41000/201>
- Gao, J., French, A. P., Pound, M. P., He, Y., Pridmore, T. P., & Pieters, J. G. (2020). Deep convolutional neural networks for image-based *Convolvulus sepium* detection in sugar beet fields. *Plant Methods*, 16(1), 1–12. <https://doi.org/10.1186/s13007-020-00570-z>
- Gonçalves, É. C., Almeida, G. P. de, Silva, E. L. Da, Schein, T. T., Evald, P. J. D. de O., & Drews-Jr, P. L. J. (2023). Line Detection and Segmentation of Annual Crops Using Hybrid Method. *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*, 472–477. <https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10332920>
- Guerra, D. (2021). *Desarrollo de aplicación web progresiva para la gestión de pedido aplicada a la empresa de ropa HOXTONX* [Universidad Técnica de Ambato]. <https://repositorio.uta.edu.ec/jspui/handle/123456789/32673>
- Guzmán, D. (2017). *Etapas fenológicas del maíz (Zea mays L.) VAR. Tusilla bajo las condiciones climáticas del cantón Cumandá, provincia de Chimborazo* [Universidad Técnica de Ambato]. <https://repositorio.uta.edu.ec/jspui/handle/123456789/25123>
- Habib, G., & Qureshi, S. (2022). Optimization and acceleration of convolutional neural networks: A survey. *Journal of King Saud University - Computer and Information Sciences*, 34(7), 4250–4251. <https://doi.org/10.1016/j.jksuci.2020.10.004>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 773–774. <https://doi.org/10.1109/CVPR.2016.90>
- Hu, K., Coleman, G., Zeng, S., Wang, Z., & Walsh, M. (2020). Graph weeds net: A graph-based deep learning method for weed recognition. *Computers and Electronics in Agriculture*, 174, 1–9. <https://doi.org/10.1016/j.compag.2020.105520>

- Hu, X.-Z., Jeon, W.-S., & Rhee, S.-Y. (2022). Sugar Beets and Weed Detection using Semantic Segmentation. *2022 International Conference on Fuzzy Theory and Its Applications (IFUZZY)*, 1–4. <https://doi.org/10.1109/IFUZZY55320.2022.9985222>
- Huang, H., Deng, J., Lan, Y., Yang, A., Deng, X., Wen, S., Zhang, H., & Zhang, Y. (2018). Accurate Weed Mapping and Prescription Map Generation Based on Fully Convolutional Networks Using UAV Imagery. *Sensors*, *18*(10), 3299. <https://doi.org/10.3390/s18103299>
- Huang, H., Lan, Y., Yang, A., Zhang, Y., Wen, S., & Deng, J. (2020). Deep learning versus Object-based Image Analysis (OBIA) in weed mapping of UAV imagery. *International Journal of Remote Sensing*, *41*(9), 3446–3479. <https://doi.org/10.1080/01431161.2019.1706112>
- ISO. (2016, June). *ISO/IEC 25023:2016 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality*. <https://www.iso.org/standard/35747.html>
- ISO. (2022). *International Organization for Standardization*. <https://www.iso.org/about-us.html>
- ISO 25000. (2022). *ISO/IEC 25010*. <https://iso25000.com/index.php/normas-iso-25000/iso-25010?start=6>
- Izquierdo, R. (2012). *Evaluación del cultivo de maíz (Zea mays), como complemento a la alimentación de bovinos de leche en épocas de escasez de alimento*. Cayambe - Ecuador [Universidad politécnica salesiana sede Quito]. <http://dspace.ups.edu.ec/handle/123456789/1832>
- JavaScript. (2023). *JavaScript*. <https://www.javascript.com/>
- Jiang, H., Zhang, C., Qiao, Y., Zhang, Z., Zhang, W., & Song, C. (2020). CNN feature based graph convolutional network for weed and crop recognition in smart farming. *Computers and Electronics in Agriculture*, *174*, 1–11. <https://doi.org/10.1016/j.compag.2020.105450>
- Jiang, Z. (2019). A Novel Crop Weed Recognition Method Based on Transfer Learning from VGG16 Implemented by Keras. *IOP Conference Series: Materials Science and Engineering*, *677*(3), 1–17. <https://doi.org/10.1088/1757-899X/677/3/032073>

- Khan, A., Ilyas, T., Umraiz, M., Mannan, Z. I., & Kim, H. (2020). CED-Net: Crops and Weeds Segmentation for Smart Farming Using a Small Cascaded Encoder-Decoder Architecture. *Electronics*, 9(10), 1602. <https://doi.org/10.3390/electronics9101602>
- Khan, N. A., Jhanjhi, N. Z., Brohi, S. N., & Nayyar, A. (2020). Emerging use of UAV's: secure communication protocol issues and challenges. In *Drones in Smart-Cities* (pp. 37–55). Elsevier. <https://doi.org/10.1016/B978-0-12-819972-5.00003-3>
- Kiran, D. R. (2019). Just in time and kanban. In *Production Planning and Control* (pp. 373–377). Elsevier. <https://doi.org/10.1016/B978-0-12-818364-9.00026-3>
- Kounalakis, T., Malinowski, M. J., Chelini, L., Triantafyllidis, G. A., & Nalpantidis, L. (2018). A Robotic System Employing Deep Learning for Visual Recognition and Detection of Weeds in Grasslands. *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, 1–6. <https://doi.org/10.1109/IST.2018.8577153>
- Laghari, A. A., Jumani, A. K., Laghari, R. A., & Nawaz, H. (2023). Unmanned aerial vehicles: A review. *Cognitive Robotics*, 3, 8–10. <https://doi.org/10.1016/j.cogr.2022.12.004>
- Lam, O. H. Y., Dogotari, M., Prüm, M., Vithlani, H. N., Roers, C., Melville, B., Zimmer, F., & Becker, R. (2021). An open source workflow for weed mapping in native grassland using unmanned aerial vehicle: using *Rumex obtusifolius* as a case study. *European Journal of Remote Sensing*, 54(sup1), 71–88. <https://doi.org/10.1080/22797254.2020.1793687>
- Li, X., Duan, F., Hu, M., Hua, J., & Du, X. (2023). Weed Density Detection Method Based on a High Weed Pressure Dataset and Improved PSP Net. *IEEE Access*, 11, 98244–98255. <https://doi.org/10.1109/ACCESS.2023.3312191>
- Li, Y., Qian, R., & Li, K. (2022). Inter-patient arrhythmia classification with improved deep residual convolutional neural network. *Computer Methods and Programs in Biomedicine*, 214, 5. <https://doi.org/10.1016/j.cmpb.2021.106582>
- Liang, W.-C., Yang, Y.-J., & Chao, C.-M. (2019). Low-Cost Weed Identification System Using Drones. *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)*, 260–263. <https://doi.org/10.1109/CANDARW.2019.00052>

- Lind, D. A., Wathen, S. A., & Marchal, W. G. (2012). *ESTADÍSTICA APLICADA A LOS NEGOCIOS Y A LA ECONOMÍA* (J. Mares & M. Teresa, Eds.; 15th ed., pp. 392–488). McGraw Hill.
- Linsel, A., Bär, K., Haas, J., Hornung, J., Greb, M. D., & Hinderer, M. (2020). GeoReVi: A knowledge discovery and data management tool for subsurface characterization. *SoftwareX*, 12, 2. <https://doi.org/10.1016/j.softx.2020.100597>
- Lizama, O., Kindley, G., & Jeria Morales, I. J. (2016). *Redes de computadores arquitectura Cliente-Servidor*. Universidad Técnica Federico Santa María.
- Loshchilov, I., & Hutter, F. (2016). *SGDR: Stochastic Gradient Descent with Warm Restarts*. 2. <http://arxiv.org/abs/1608.03983>
- Lottes, P., Behley, J., Chebrolu, N., Milioto, A., & Stachniss, C. (2018). Joint Stem Detection and Crop-Weed Classification for Plant-Specific Treatment in Precision Farming. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8233–8238. <https://doi.org/10.1109/IROS.2018.8593678>
- Lottes, P., Behley, J., Chebrolu, N., Milioto, A., & Stachniss, C. (2020). Robust joint stem detection and crop-weed classification using image sequences for plant-specific treatment in precision farming. *Journal of Field Robotics*, 37(1), 20–34. <https://doi.org/10.1002/rob.21901>
- Lottes, P., Behley, J., Milioto, A., & Stachniss, C. (2018). Fully Convolutional Networks With Sequential Information for Robust Crop and Weed Detection in Precision Farming. *IEEE Robotics and Automation Letters*, 3(4), 2870–2877. <https://doi.org/10.1109/LRA.2018.2846289>
- Martarelli, N. J., & Nagano, M. S. (2022). How to undertake reviews of large collections of articles and establish main contributions: an ontology-based literature review approach. *International Journal of Information Management Data Insights*, 2(2), 5. <https://doi.org/10.1016/j.jjime.2022.100091>
- Méndez Barceló, A. (2016). *Percepción de los productores de maíz (Zea mays, Lin.) sobre sus plagas claves: principales aspectos agroecológicos en área agrícolas de Venezuela* (pp. 6–13). Editorial Universitaria. <https://elibro.net/es/ereader/utnorte/100769>

- Meng, J., Mei, S., & Yan, Z. (2009). RESTful Web Services: A Solution for Distributed Data Integration. *2009 International Conference on Computational Intelligence and Software Engineering*, 1–4. <https://doi.org/10.1109/CISE.2009.5365234>
- Microsoft. (2023, November 22). *Key Azure services for developers*. <https://learn.microsoft.com/en-us/azure/developer/intro/azure-developer-key-services>
- Ministerio de Agricultura y Ganadería. (2019, September 9). *Agricultura, la base de la economía y la alimentación*.
- Misra, D. (2019). *Mish: A Self Regularized Non-Monotonic Activation Function*. 1–2. <http://arxiv.org/abs/1908.08681>
- Morales, J., Rodriguez, G., Huang, G., & Akopian, D. (2020). Toward UAV Control via Cellular Networks: Delay Profiles, Delay Modeling, and a Case Study Within the 5-mile Range. *IEEE Transactions on Aerospace and Electronic Systems*, 56(5), 4132–4151. <https://doi.org/10.1109/TAES.2020.2987406>
- Mou, L., & Zhu, X. X. (2018). Vehicle Instance Segmentation From Aerial Image and Video Using a Multitask Learning Residual Fully Convolutional Network. *IEEE Transactions on Geoscience and Remote Sensing*, 56(11), 6702. <https://doi.org/10.1109/TGRS.2018.2841808>
- Nation United. (2017). *Sustainable development goals*.
- Naveed, M., & Qazi, S. (2020). *QoE Analysis of Real-Time Video Streaming over 4G-LTE for UAV-Based Surveillance Applications* (Vol. 1198, pp. 443–458). Springer, Singapore. https://doi.org/10.1007/978-981-15-5232-8_38
- Osorio Delgado, A. K. (2021). *Método para la estimación de maleza en cultivos de lechuga utilizando aprendizaje profundo e imágenes multiespectrales*. Universidad Nacional de Colombia.
- Osorio, K., Puerto, A., Pedraza, C., Jamaica, D., & Rodríguez, L. (2020). A Deep Learning Approach for Weed Detection in Lettuce Crops Using Multispectral Images. *AgriEngineering*, 2(3), 471–488. <https://doi.org/10.3390/agriengineering2030032>

- Ospino, M. (2018). *Arquitectura de software para las gestión y proyección financiera de planes de negocio* [Universidad de Cartagena]. <https://repositorio.unicartagena.edu.co/bitstream/handle/11227/8503/TESIS%20DE%20GRADO%20RUIZ%20HERN%C3%81NDEZ.pdf?sequence=1&isAllowed=y>
- Palma, F., Olsson, T., Wingkvist, A., & Gonzalez-Huerta, J. (2022). Assessing the linguistic quality of REST APIs for IoT applications. *Journal of Systems and Software*, 191, 1. <https://doi.org/10.1016/j.jss.2022.111369>
- Paušič, A., Tojnko, S., & Lešnik, M. (2021). Permanent, undisturbed, in-row living mulch: A realistic option to replace glyphosate-dominated chemical weed control in intensive pear orchards. *Agriculture, Ecosystems & Environment*, 318, 1–2. <https://doi.org/10.1016/j.agee.2021.107502>
- Puerto, A. (2018). *Clasificación y cuantificación de maleza en cultivos de hortalizas por medio de procesamiento de imágenes digitales multiespectrales* [Universidad Nacional de Colombia]. <https://repositorio.unal.edu.co/handle/unal/62958>
- Python TM. (2023). *Python*. <https://www.python.org/>
- Qi, M., Gao, H., Wang, T., Du, B., Li, H., Zhong, W., & Tang, Y. (2023). Method for Segmentation of Bean Crop and Weeds Based on Improved UperNet. *IEEE Access*, 11, 143804–143814. <https://doi.org/10.1109/ACCESS.2023.3344520>
- Ringselle, B., Bakken, A. K., Höglind, M., Jørgensen, M., & Tørresen, K. S. (2023). Effects of integrated grassland renewal strategies on annual and perennial weeds in the sowing year and subsequent production years. *European Journal of Agronomy*, 146, 1–2. <https://doi.org/10.1016/j.eja.2023.126799>
- Roboflow. (2024, January). *Create a Project*. <https://docs.roboflow.com/datasets/create-a-project>
- Rodríguez, G. (2020). *Evaluación de la competencia de malezas en cultivo de maíz duro (Zea mays L.) en la zona Ventanas - Los Ríos* [Universidad Agraria del Ecuador]. <https://repositorio.espam.edu.ec/bitstream/42000/741/1/TMA164.pdf>

- Sa, I., Chen, Z., Popovic, M., Khanna, R., Liebisch, F., Nieto, J., & Siegwart, R. (2018). weedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming. *IEEE Robotics and Automation Letters*, 3(1), 588–595. <https://doi.org/10.1109/LRA.2017.2774979>
- Sai, G. U., Tejasri, N., Kumar, A., & Rajalakshmi, P. (2022). Deep Learning Based Overcomplete Representations for Paddy Rice Crop and Weed Segmentation. *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 6077–6080. <https://doi.org/10.1109/IGARSS46834.2022.9884107>
- Scrum Manager. (2021, April 26). *El manifiesto ágil*. https://www.scrummanager.com/bok/index.php?title=El_manifiesto_%C3%A1gil
- Senplades. (2017a). *Agenda Zonal Zona-1*. <https://www.Planificacion.Gob.Ec/Wp-Content/Uploads/Downloads/2015/10/Agenda-Zona-1.Pdf>.
- Senplades. (2017b). *Buen vivir 2017-2021*. <http://www.Planificacion.Gob.Ec/Wp-Content/Uploads/Downloads/2017/07/Plan-Nacional-Para-El-Buen-Vivir-2017-2021.Pdf>.
- Sharpe, S. M., Schumann, A. W., & Boyd, N. S. (2019). Detection of Carolina Geranium (*Geranium carolinianum*) Growing in Competition with Strawberry Using Convolutional Neural Networks. *Weed Science*, 67(2), 239–245. <https://doi.org/10.1017/wsc.2018.66>
- Sharpe, S. M., Schumann, A. W., & Boyd, N. S. (2020). Goosegrass Detection in Strawberry and Tomato Using a Convolutional Neural Network. *Scientific Reports*, 10(1), 9548. <https://doi.org/10.1038/s41598-020-66505-9>
- Shehab, L. H., Fahmy, O. M., Gasser, S. M., & El-Mahallawy, M. S. (2021). An efficient brain tumor image segmentation based on deep residual networks (ResNets). *Journal of King Saud University - Engineering Sciences*, 33(6), 408. <https://doi.org/10.1016/j.jksues.2020.06.001>
- Smith, L. N. (2015). *Cyclical Learning Rates for Training Neural Networks*. <http://arxiv.org/abs/1506.01186>
- Student. (1908). The Probable Error of a Mean. *Biometrika*, 6(1), 1. <https://doi.org/10.2307/2331554>

- SUGIMORI, Y., KUSUNOKI, K., CHO, F., & UCHIKAWA, S. (1977). Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system. *International Journal of Production Research*, 15(6), 553–564. <https://doi.org/10.1080/00207547708943149>
- Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 14(4), 75–76. <https://doi.org/10.12928/telkomnika.v14i4.3956>
- T, S., T, S., G S, S. G., S, S., & R, K. (2019). Performance Comparison of Weed Detection Algorithms. *2019 International Conference on Communication and Signal Processing (ICCSP)*, 0844. <https://doi.org/10.1109/ICCSP.2019.8698094>
- Teimouri, N., Dyrmann, M., Nielsen, P., Mathiassen, S., Somerville, G., & Jørgensen, R. (2018). Weed Growth Stage Estimator Using Deep Convolutional Neural Networks. *Sensors*, 18(5), 1–11. <https://doi.org/10.3390/s18051580>
- Tetila, E. C., Machado, B. B., Menezes, G. K., Da Silva Oliveira, A., Alvarez, M., Amorim, W. P., De Souza Belete, N. A., Da Silva, G. G., & Pistori, H. (2020). Automatic Recognition of Soybean Leaf Diseases Using UAV Images and Deep Convolutional Neural Networks. *IEEE Geoscience and Remote Sensing Letters*, 17(5), 903–907. <https://doi.org/10.1109/LGRS.2019.2932385>
- Ullah, H., Khan, N., & Khan, I. A. (2023). Complementing cultural weed control with plant allelopathy: Implications for improved weed management in wheat crop. *Acta Ecologica Sinica*, 43(1), 28. <https://doi.org/10.1016/j.chnaes.2021.06.006>
- Ullah, H. S., Asad, M. H., & Bais, A. (2021). End to End Segmentation of Canola Field Images Using Dilated U-Net. *IEEE Access*, 9, 59741–59753. <https://doi.org/10.1109/ACCESS.2021.3073715>
- Valizadeh, M., & Wolff, S. J. (2022). Convolutional Neural Network applications in additive manufacturing: A review. *Advances in Industrial and Manufacturing Engineering*, 4, 1–2. <https://doi.org/10.1016/j.aime.2022.100072>

- Veeranampalayam Sivakumar, A. N., Li, J., Scott, S., Psota, E., J. Jhala, A., Luck, J. D., & Shi, Y. (2020). Comparison of Object Detection and Patch-Based Classification Deep Learning Models on Mid- to Late-Season Weed Detection in UAV Imagery. *Remote Sensing*, *12*(13), 1–22. <https://doi.org/10.3390/rs12132136>
- Weyler, J., Läbe, T., Magistri, F., Behley, J., & Stachniss, C. (2023). Towards Domain Generalization in Crop and Weed Segmentation for Precision Farming Robots. *IEEE Robotics and Automation Letters*, *8*(6), 3310–3317. <https://doi.org/10.1109/LRA.2023.3262417>
- Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5987–5995. <https://doi.org/10.1109/CVPR.2017.634>
- Yu, J., Schumann, A. W., Cao, Z., Sharpe, S. M., & Boyd, N. S. (2019). Weed Detection in Perennial Ryegrass With Deep Learning Convolutional Neural Network. *Frontiers in Plant Science*, *10*. <https://doi.org/10.3389/fpls.2019.01422>
- Zambrano, J. L., Velásquez, J., Peñaherrera, D., Sangoquiza, C., Cartagena, Y., Villacrés, E., Garcés, S., Ortíz, R., León, J., Campaña, D. †, López, V., Asaquibay, C., Nieto, M., Sanmartín, G., Pintado, P., Yáñez, C., & Racines, M. (2021). *Guía para la producción sustentable de maíz en la Sierra ecuatoriana Quito-Ecuador* (Vol. 122). INIAP. <https://repositorio.iniap.gob.ec/bitstream/41000/5796/1/GUIA%20CULTIVO%20DE%20MAIZ%202021-1.pdf>
- Zhang, K., Yang, X., Cao, H., Thé, J., Tan, Z., & Yu, H. (2023). Multi-step forecast of PM_{2.5} and PM₁₀ concentrations using convolutional neural network integrated with spatial–temporal attention and residual learning. *Environment International*, *171*, 1–7. <https://doi.org/10.1016/j.envint.2022.107691>
- Zhang, R., Wang, C., Hu, X., Liu, Y., Chen, S., & Su, B. (2018). Weed location and recognition based on UAV imaging and deep learning. *International Journal of Precision Agricultural Aviation*, *1*(1), 23–29. <https://doi.org/10.33440/j.ijpaa.20200301.63>
- Zhou, J., Zhou, J., Ye, H., Ali, M. L., Chen, P., & Nguyen, H. T. (2021). Yield estimation of soybean breeding lines under drought stress using unmanned aerial vehicle-based imagery and

convolutional neural network. *Biosystems Engineering*, 204, 90–103.
<https://doi.org/10.1016/j.biosystemseng.2021.01.017>