

SISTEMA DE CLASIFICACIÓN DE COBERTURAS EN IMÁGENES TOMADAS POR DRONES USANDO
TÉCNICAS DE DEEP LEARNING

Jhon Jairo Tello Dagua

Nota de Aceptación

Certificamos que el presente Trabajo de Grado
Satisface, en alcances y calidad, todos los requisitos
Que demanda un Trabajo de Grado de Maestría.



Louis Reymondin
Director



Gloria Inés Álvarez Vargas
Jurado



Alejandro Coca Castro
Jurado

Aprobado en cumplimiento de los requisitos exigidos
por la Pontificia Universidad Javeriana Cali, para optar
el título de Magister en Ingeniería de Software.



HERNÁN CAMILO ROCHA NIÑO Ph. D.
Decano Facultad de Ingeniería y Ciencias



JUAN CARLOS MARTÍNEZ ARIAS
Director Posgrados de Ingeniería y Ciencias

Acta de Correcciones al Documento de Trabajo de Grado

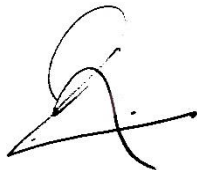
Santiago de Cali, 03 de marzo de 2021

Autor: Jhon Jairo Tello Dagua

Título del Trabajo de Grado: “SISTEMA DE CLASIFICACIÓN DE COBERTURAS EN IMÁGENES TOMADAS POR DRONES USANDO TÉCNICAS DE DEEP LEARNING”

Director: Dr. Louis Reymondin

Como indica el artículo 2.13 de las Directrices para Trabajo de Grado de Maestría, he verificado que el estudiante indicado arriba ha implementado todas las correcciones que los Jurados del Proyecto de Trabajo de Grado definieron que se efectuaran, como consta en el Acta de Evaluación correspondiente.



Firma del Director del Trabajo de Grado

DATOS DEL ESTUDIANTE

1. NOMBRE COMPLETO: Jhon Jairo Tello Dagua
2. DIRECCIÓN: Km 8 La Buitrera vía al Rosario
3. TELEFONO DE CONTACTO: 3175756684
4. CORREO ELECTRÓNICO: jjtello@javerianacali.edu.co
5. PROFESIÓN: Especialista en Ingeniería de Software
6. UNIVERSIDAD: Pontificia Universidad Javeriana Cali
7. EMPRESA: Centro Internacional de Agricultura Tropical (CIAT)
8. CARGO: Analista de Sistemas

RESUMEN

El mapeo de la cobertura terrestre es un tema de investigación importante en la ciencia del cambio en el uso de la tierra y la planificación del paisaje. Las actividades humanas cambian constantemente los patrones de cobertura de la tierra e influyen en los procesos biofísicos del territorio. Tradicionalmente la detección, clasificación y el monitoreo de coberturas de la tierra se ha llevado a cabo a través de la teledetección satelital. La capacidad para detectar, clasificar y cuantificar depende en gran parte de la capacidad del sensor y técnicas de clasificación entre las que se destacan la supervisada y no supervisada. Estas técnicas dependen de la calidad del algoritmo utilizado para discriminar las categorías. Con el uso de los drones es posible en la actualidad contar con grandes conjuntos de imágenes de alta resolución que contienen una gran cantidad de información que se puede explorar. Estas imágenes tienen el potencial de contener varios tipos de características como: bosques, casas, edificios, cultivos, carreteras, entre muchos otros. Esta investigación se enfocará en explorar el uso de técnicas de Deep Learning y redes neuronales convolucionales para la clasificación automatizada de coberturas de la tierra usando imágenes de alta resolución tomadas por drones. Se usará la librería de Deep Learning, Deeplearning4j para implementar un prototipo para la clasificación de coberturas. Como este tipo de modelos requiere que el conjunto de datos de entrenamiento sea bastante grande para un rendimiento más óptimo también dentro de la investigación se creará una plataforma para el etiquetado de imágenes por parte de expertos y una base de datos que estará en continuo crecimiento a medida que se etiqueten nuevas imágenes.

ABSTRACT

Land cover mapping is an important research topic in the science of land use change and landscape planning. Human activities constantly change land cover patterns and influence the biophysical processes of the territory. Traditionally, the detection, classification and monitoring of land cover has been carried out through satellite remote sensing. The ability to detect, classify and quantify depends largely on sensor capacity and classification techniques, including supervised and unsupervised techniques. These techniques depend on the quality of the algorithm used to discriminate the categories. With the use of drones, it is now possible to have large sets of high-resolution images that contain a large amount of information that can be explored. These images have the potential to contain several types of characteristics such as forests, houses, buildings, crops, roads, among many others. This research will focus on exploring the use of deep learning techniques and convolutional neural networks for the automated classification of land covers using high-resolution images taken by drones. The Deep Learning library, Deeplearning4j will be used to implement a prototype for the classification of coverage. As this type of model requires that the training data set is large enough for a more optimal performance, also within the investigation, a platform will be created for the labeling of images by experts and a database that will be in continuous growth. as new images are tagged.

SISTEMA DE CLASIFICACIÓN DE COBERTURAS EN IMÁGENES TOMADAS POR DRONES USANDO TÉCNICAS DE DEEP LEARNING

Jhon Jairo Tello Dagua
Código 8936442

Trabajo de grado para optar al título de
Magister en Ingeniería de Software

Director
Dr. Louis Reymondin

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN INGENIERIA DE SOFTWARE
SANTIAGO DE CALI, *DICIEMBRE* DE 2020

AGRADECIMIENTOS

Me gustaría agradecer primero a mi esposa e hijo y al resto de mi familia, por el apoyo en todo lo que hago, a mi abuela que ahora desde el cielo nos cuida.

A mi director Louis Reymondin por su orientación y apoyo, al director de la Maestría en Ingeniería de Software Juan Carlos Martinez por su paciencia, apoyo y motivación para seguir adelante y terminar el proyecto.

A la Pontificia Universidad Javeriana Cali, directivos y profesores por todo el conocimiento brindado a través de la Maestría en Ingeniería de Software.

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	9
2. DEFINICION DEL PROBLEMA DE INVESTIGACIÓN	11
2.1 PLANTEAMIENTO DEL PROBLEMA	11
3. OBJETIVOS DEL PROYECTO.....	13
3.1 OBJETIVO GENERAL	13
3.2 OBJETIVOS ESPECÍFICOS.....	13
3.3 RESULTADOS ESPERADOS	13
4. JUSTIFICACIÓN DEL TRABAJO DE GRADO	14
5. MARCO DE REFERENCIA	15
5.1 ESTADO DEL ARTE.....	15
5.1.1 Cobertura y uso de la tierra	15
5.1.2 Definición de las coberturas de la tierra.....	15
5.1.3 Métodos utilizados en la clasificación de imágenes.....	16
5.1.3.1 Clasificación basada en píxel	17
5.1.3.2 Clasificación basada en subpixel.....	18
5.1.3.3 Clasificadores contextuales	18
5.1.3.4 Clasificadores basados por campo.....	18
5.1.3.5 Clasificación orientada a objetos	19
5.1.4 Software existente orientado a la clasificación de imágenes	19
5.1.5 Sistemas existentes para el etiquetado de imágenes.....	23
5.1.6 Validación de la clasificación de imágenes.....	25
5.1.6.1 Matriz de confusión	25
5.1.6.2 Coeficiente de Validación Kappa.....	26
5.1.7 Redes neuronales artificiales.....	27
5.1.7.1 Características de las redes neuronales	31
5.1.7.2 Mecanismo de aprendizaje.....	32
5.1.8 Deep Learning	34
5.1.9 Redes Neuronales convolucionales.....	36
5.1.9.1 Clasificación de imágenes con redes neuronales convolucionales CNN	38
5.1.9.2 Operaciones principales de redes convolucionales.....	39



5.1.9.3 Funcionamiento De La Red Convolutacional	40
5.1.9.4 Antecedentes de uso de redes neuronales convolucionales y deep learning.....	41
6. METODOLOGIA DE LA INVESTIGACIÓN.....	44
6.1 Recopilación de las imágenes de alta resolución con drones.....	46
6.1.1 Generación de Ortomosaicos usando pix4D.....	47
6.2 Etapa de educación de requisitos	49
6.2.1 Diagrama de contexto	49
6.2.2 Técnicas de obtención de requisitos	51
6.2.3 Documentación de requisitos	52
Atributos de Requisitos Funcionales.....	52
Prioridad de Requisitos	53
Estado de Requisitos	53
Estado de Negociación de Requisitos	54
Requisitos Funcionales	54
Plantilla de Requisitos Funcionales:	69
Requisitos No Funcionales	69
USA - Usabilidad	69
SEG - Seguridad.....	71
POR - Portabilidad.....	72
DIS - Disponibilidad	73
DES - Desempeño.....	74
Plantilla de Requisitos No Funcionales:.....	74
6.2.4 Priorización de requisitos	75
6.2.5 Negociación de requisitos	75
6.3 Arquitectura de software y diseño de la solución	77
6.4 Modelo de datos	81
6.5 Creación de las submuestras.....	82
6.6 Codificación de la aplicación web para el etiquetado de imágenes de alta resolución.....	84
Interfaz gráfica de usuario aplicación web	84
6.7 Prototipo de clasificación de coberturas.....	92

El dataset	93
La arquitectura	96
Desarrollo del prototipo de clasificación de coberturas	98
Estructura y parámetros de la CNN	100
6.8 Pruebas aplicación web y prototipo	103
6.9 Evaluación del prototipo de clasificación de coberturas.....	105
6.10 Prueba del modelo de clasificación en una imagen de alta resolución	107
7. GLOSARIO DE TERMINOS ESPECIALES	108
8. CONCLUSIONES	110
9. TRABAJO FUTURO	111
10. ANEXOS	112
11. REFERENCIAS BIBLIOGRÁFICAS	113

LISTA DE FIGURAS

Figura 1. Elementos principales de una neurona (Hernández, 2014)	28
Figura 2. Esquema general de una Red Neuronal Artificial (Hernández, 2014).....	30
Figura 3. Redes neuronales organizadas en capas que constan de una serie de nodos interconectados. Las redes pueden tener decenas o cientos de capas ocultas ("Aprendizaje profundo: Tres cosas que es necesario saber", 2018).....	36
Figura 4. Ejemplo de una red con muchas capas convolucionales. Se aplican filtros a cada imagen de entrenamiento con distintas resoluciones, y la salida de cada imagen convolucionada se emplea como entrada para la siguiente capa ("Aprendizaje profundo: Tres cosas que es necesario saber", 2018).	37
Figura 5. Lado izquierdo lo que el ojo humano percibe, lado derecho lo que la computadora percibe (Sorokina, 2018).	38
Figura 6. Secuencia de transformaciones en una red convolucional. Fuente: "Convolutional Neural Network (CNN)", 2018	40
Figura 7. Fases del proceso metodológico.....	46
Figura 8. Captura de imágenes dron.....	47
Figura 9. Posprocesamiento de imágenes dron usando PIX4D.....	47
Figura 10. Ortomosaicos seleccionados para la generación de las 10000 muestras....	48
Figura 11. Diagrama de contexto del sistema	50
Figura 12. Arquitectura de alto nivel.....	78
Figura 13. Vista de Arquitectura y segmentos de capa	78
Figura 14. Patrón de diseño MVC de Laravel. Fuente: (¿"Why Use Laravel as a Development Platform?", 2020)	80
Figura 15. Diagrama entidad-relación aplicación web de etiquetado	82

Figura 16. Diagrama de clases modulo para la generaci3n de submuestras	84
Figura 17. Pantalla inicial de la aplicaci3n web de etiquetado de im3genes	85
Figura 18. Login de usuario.....	85
Figura 19. Vista de los m3dulos disponibles desde un usuario administrador	86
Figura 20. Modulo usuarios.....	86
Figura 21. Vista creaci3n de un usuario nuevo	86
Figura 22. Modulo roles.....	87
Figura 23. Vista creaci3n de un nuevo rol.....	87
Figura 24. Modulo labels.....	88
Figura 25. Vista creaci3n de una nueva etiqueta	88
Figura 26. M3dulo experimentos	89
Figura 27. Vista creaci3n de un nuevo experimento	90
Figura 28. Vista de mostrar un experimento creado.....	91
Figura 29. Vista de una muestra etiquetada.....	92
Figura 30. Par3metros de configuraci3n para el entrenamiento del modelo	97
Figura 31. Diagrama de paquetes prototipo de clasificaci3n de coberturas	99
Figura 32. Diagrama de clases prototipo de clasificaci3n de coberturas. Ver anexo 3	100
Figura 33. Casos de prueba aplicaci3n web y prototipo.....	105
Figura 34. Ejemplo de una muestra etiquetada como NA	106
Figura 35. Matriz de confusi3n prototipo de clasificaci3n de coberturas	107
Figura 36. Prueba del prototipo de clasificaci3n.....	108

LISTA DE TABLAS

Tabla 1. Tipos de exactitud para la validaci3n de la clasificaci3n de im3genes (V3squez, 2011).....	26
Tabla 2. Categor3a de concordancias de validaci3n para el coeficiente Kappa (V3squez, 2011).....	27
Tabla 3. Cantidad de muestras tomadas por clase, muestras seleccionadas para entrenamiento y pruebas por clase	94
Tabla 4. Estructura y par3metros utilizados en la CNN.....	101

LISTA DE ANEXOS

Anexo 1. Formato de entrevista utilizado	112
Anexo 2. Formato y resultado del proceso de priorizaci3n de requisitos	112
Anexo 3. Diagrama de clases prototipo de clasificaci3n de coberturas completo	112

Palabras clave: Deep learning, redes neuronales convolucionales, reconocimiento de im3genes, ingenier3a de software, clasificaci3n de coberturas

SIGLAS Y ACRÓNIMOS

BSD*	Distribución de Software de Berkeley
CIAT	Centro Internacional de Agricultura Tropical
CNN*	Red Neuronal Convolutiva
CRUD*	Crear, Leer, Actualizar, Borrar
SGBD	Sistema de Gestión de Base de Datos
DCNN*	Red Neuronal Convolutiva Profunda
GPU*	Unidad de Procesamiento Gráfico
IA	Inteligencia Artificial
IR	Ingeniería de Requisitos
MVC	Modelo Vista Controlador
PHP*	Preprocesador de Hipertexto
RNA	Red Neuronal Artificial
SIG	Sistemas de Información Geográfica
UAV*	Vehículo aéreo no tripulado
UI*	Interfaz de Usuario

* _ Por sus siglas en inglés.

1. INTRODUCCIÓN

El monitoreo de cambios de cobertura vegetal y uso de suelo resulta ser un tema de gran interés en el campo de la conservación ya que permite estudiar el estado de coberturas de alta importancia como el bosque húmedo tropical de la amazonia, que en la última década ha sido intervenido drásticamente para dar paso a cultivos de expansión continua como la palma de aceite, minería y ganadería. Actualmente este monitoreo se lleva a cabo realizando un análisis de imágenes de satélite, como es el caso del sistema Terra-i desarrollado en el CIAT, el cual utiliza imágenes de baja resolución espacial (250 m) y alta temporalidad (cada 16 días) para el monitoreo de cambios de cobertura a nivel pantropical (Reymondin, et al 2012).

Las coberturas de la tierra proveen información valiosa que puede ser utilizada en diferentes niveles (país, departamentos, municipios, áreas protegidas, áreas

indígenas) procesos como mapas de vegetación, conflictos de uso de la tierra, ordenamiento territorial, monitoreo de cuencas, levantamiento y seguimiento de indicadores ambientales como porcentaje de cobertura boscosa, inventarios forestales, porcentaje de cobertura boscosa y tasa de deforestación de bosques, entre otros (Coca-Castro, et al 2013).

Tradicionalmente, la generación de información cartográfica sobre usos o coberturas del suelo ha implicado técnicas de fotointerpretación apoyadas en revisiones de campo. La fuente principal de información son fotografías aéreas de escalas similares a las que actualmente ofrecen las imágenes de satélite de alta resolución espacial. El problema de esta metodología es que es muy intensiva en mano de obra especializada, requiere mucho tiempo y está notablemente influenciada por la subjetividad de los técnicos (Cánovas, 2012).

Hasta mediados de la década de los 90, los métodos de clasificación supervisada aplicados al análisis de imágenes de satélite se basaban principalmente en técnicas estadísticas convencionales, como la clasificación por máxima verosimilitud o por mínima distancia. Aunque estas técnicas pueden dar buenos resultados, su capacidad para resolver problemas de confusión entre clases es muy limitada.

Como consecuencia, en los últimos años, a raíz de los avances experimentados en el campo tecnológico de la computación, se han propuesto estrategias alternativas, particularmente el uso de redes neuronales artificiales, árboles de decisión, máquinas de vectores soporte o métodos derivados de la teoría de la lógica borrosa (Tso y Mather, 2009).

En este proyecto se utilizarán técnicas de Deep Learning y redes neuronales convolucionales (S. Lawrence, et al 1997) para la clasificación de coberturas utilizando imágenes de alta resolución tomadas por drones UAV las cuales serán etiquetadas y almacenadas en un base de datos que serán usadas para alimentar el modelo.

2. DEFINICION DEL PROBLEMA DE INVESTIGACIÓN

2.1 PLANTEAMIENTO DEL PROBLEMA

Con el fin de tener mayor veracidad en los resultados del análisis de las imágenes satelitales es importante evaluar la exactitud de las detecciones correspondientes a cambios de cobertura realizando procesos detallados de validación y verificación en campo. Estas rectificaciones en campo resultan ser dispendiosas y de alto costo.

Actualmente en el CIAT se está utilizando los drones como alternativa de validación. Con el drone se logra capturar imágenes de alta calidad las cuales son utilizadas para validar el cambio dado por el satélite, determinar el tipo de cobertura presente y área de cambio en la zona validada. Esta además es una solución económicamente viable para validación de áreas mayores a 5 ha debido a que se puede cubrir toda el área en menor tiempo.

Los vuelos se realizan de forma autónoma realizando misiones de áreas de 270m * 270m, la altitud del vuelo oscila entre los 90m a 120m sobre el nivel del suelo, aproximadamente.

El proceso de identificación y clasificación de los tipos de cobertura presente en cada imagen es realizado por un investigador experto el cual se encarga de organizar las imágenes, realizar la revisión de cada una de ellas, registrando manualmente el tipo de cobertura (bosque, arbustos, herbáceas, suelo desnudo, pastizales, entre otras) encontrado en cada una, este proceso puede tomar uno o varios días dependiendo la cantidad de imágenes a evaluar ocasionando demoras en la consolidación de los datos para los análisis posteriores que deben hacerse.

De acuerdo con los investigadores sería de gran ayuda contar con una herramienta que realice el proceso de clasificación de coberturas de forma automática y eficiente, permitiéndoles contar con más tiempo para los pasos siguientes como determinar dinámicas y agentes de cambio en la zona. También ayudaría a poder analizar zonas de interés de mayor tamaño donde podamos tener un conocimiento más amplio del comportamiento al interior de un territorio.

Los enfoques actuales para el reconocimiento de objetos hacen uso de los métodos de aprendizaje automático. Para mejorar el rendimiento de estos métodos se puede recopilar conjuntos de datos más grandes, entrenar modelos más potentes y utilizar mejores técnicas para evitar el sobreajuste (overfitting). Hasta hace algunos años, los conjuntos de datos de las imágenes etiquetadas eran relativamente pequeños, del orden de decenas de miles de imágenes. Las tareas de reconocimiento simple se pueden resolver bastante bien con conjuntos de este tamaño.

Los objetos en entornos realistas exhiben una considerable variabilidad, por lo que para aprender a reconocerlos es necesario utilizar conjuntos de entrenamientos grandes y para aprender sobre muchos objetos en gran cantidad de imágenes, es necesario un modelo con una gran capacidad de aprendizaje (Krizhevsky, et al 2012).

Un ejemplo sobre el uso de la aplicación de estas técnicas es el reconocimiento de caras de Facebook y las personas a quienes pertenecen, utilizando una gran cantidad de imágenes etiquetadas, haciendo uso del espectro visible y utilizando técnicas de deep learning.

Este proyecto propone el uso de técnicas de deep learning y redes neuronales convolucionales (CNN) para la clasificación de coberturas en imágenes de alta resolución tomadas con drones, creando una herramienta que permita a los investigadores del CIAT desarrollar esta tarea de una manera eficiente con un nivel de exactitud alto y que pueda ser usada en otros centros de investigación y universidades.

2.2 Formulación del problema

¿Cómo apoyar a los investigadores del CIAT en la clasificación de coberturas en imágenes de alta resolución tomadas con drones de una manera eficiente?

Sub-preguntas de investigación

P1. ¿Cuáles son los tipos de coberturas que deben ser clasificadas?

P2. ¿Cómo facilitar la interpretación de imágenes por parte de expertos?

P3. ¿Qué modelo de datos usar para almacenar las imágenes?

P4. ¿Cuál es la librería de clasificación más adecuada para la solución del problema?

P5. ¿Cómo evaluar el rendimiento de la solución propuesta?

3. OBJETIVOS DEL PROYECTO

3.1 OBJETIVO GENERAL

Desarrollar un prototipo de software para la clasificación de coberturas en imágenes tomadas por drones en el CIAT usando técnicas de deep learning y redes neuronales convolucionales.

3.2 OBJETIVOS ESPECÍFICOS

O1. Definir con los investigadores del CIAT el esquema de clasificación de coberturas que deben ser tenidas en cuenta por el prototipo de software. (P1)

O2. Desarrollar una plataforma para el etiquetado de imágenes de alta resolución por parte de expertos. (P3)

O3. Definir un modelo de datos que permita el manejo eficiente de las imágenes y datos asociados. (P2)

O4. Desarrollar un prototipo para la clasificación de coberturas basado en la librería de clasificación seleccionada. (P4)

O5. Realizar pruebas necesarias para evaluar el rendimiento de la herramienta. (P5)

3.3 RESULTADOS ESPERADOS

Los principales resultados de esta propuesta se resumen de la siguiente manera:

R1. Documento de especificación de requerimientos que incluya el prototipo de clasificación de coberturas y la plataforma para la interpretación de imágenes de alta resolución. (O1)

R2. Plataforma para el etiquetado de imágenes de alta resolución por parte de expertos. (O2)

R3. Base de datos con aproximadamente 10000 imágenes etiquetadas para el entrenamiento del modelo. (O3)

R4. Prototipo funcional para la clasificación automatizada de coberturas. (O4)

R5. Un informe presentando el Análisis realizado de la eficiencia del prototipo desarrollado. (O5)

4. JUSTIFICACIÓN DEL TRABAJO DE GRADO

La sinergia de los análisis de las imágenes de satélite, uso del dron y datos colectados en campo, permite monitorear las dinámicas de cambio en las coberturas y uso de suelo en las zonas evaluadas, además permite evaluar la exactitud de las detecciones de un modelo como Terra-i y calcular el área de cambio exacto por pixel.

Adicionalmente la identificación automatizada de las causas de deforestación permite tener un panorama claro en menor tiempo de lo que está ocurriendo en el territorio y apoyar la toma de medidas o implementación de políticas que se consideren necesarias.

Contar con esta información permite a los investigadores comprender mejor cuál es el comportamiento de estos cambios y así poder entregar análisis más completos a los tomadores de decisiones. Adicionalmente poder analizar zonas de interés de mayor tamaño donde podamos tener un conocimiento más amplio del comportamiento al interior de un territorio.

Con el desarrollo de esta herramienta se busca incorporar una mayor sofisticación en la clasificación de las coberturas presentes en estas imágenes que ayude a realizar de un manera más eficiente este proceso y que también permita alimentar una base de conocimiento con el objetivo de que la información recopilada pueda ser utilizada por otros proyectos dentro del CIAT y otras instituciones de investigación por ejemplo en la planificación y ordenamiento de los territorios, manejo sostenible de recursos naturales, seguridad alimentaria, etc.

5. MARCO DE REFERENCIA

5.1 ESTADO DEL ARTE

5.1.1 Cobertura y uso de la tierra

Los sistemas ecológicos, están compuestos por una parte visible, estructura y composición, llamado “fenosistema”; y por otra parte no visible, denominada “criptosistema”. La identificación, delimitación y caracterización general de los diferentes tipos de ecosistemas se realiza a través de los aspectos constituyentes del fenosistema, que hace referencia a la morfología de la superficie terrestre, geoforma y su recubrimiento, cobertura terrestre (agua, suelo desnudo, hielo, cobertura vegetal), en este último es donde se enfoca esta investigación. La cobertura terrestre se reconoce mediante dos términos: cobertura y uso de la tierra. El primero, hace referencia a los aspectos presentes de la superficie de la tierra, independiente de su origen (natural o antrópico) involucra la fisonomía y composición de la cobertura vegetal, que son modelados espacialmente de forma continua (gradientes) o discreta (mosaico), representados a partir de mapas temáticos donde ilustran la variación de la superficie de acuerdo a un nivel de abstracción. El segundo término, se aplica a los tipos de ocupación o utilización que hace el hombre de una cobertura, de forma temporal o permanente (Etter, 1991).

5.1.2 Definición de las coberturas de la tierra

En los últimos años, la metodología más usada para clasificar las coberturas y usos del suelo, es la europea denominada “CORINE (Coordination of information of the Environment) Land Cover”, que surgió a partir de un proyecto iniciado en 1985 para los países de la unión Europea, con el fin de capturar datos de tipo numérico y espacial, generando una base de datos de Europa, a escala 1:100.000, basados en las imágenes del satélite LANDSAT (Vásquez, 2011). En Colombia, esta metodología fue adaptada en el año 2006, para la cuenca Magdalena – Cauca, convenio Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM), Corporación Autónoma Regional del Río Grande de la Magdalena (CORMAGDALENA) y el Instituto Geográfico Agustín Codazzi (IGAC). Se organizó la clasificación de las coberturas y usos de la tierra, escala 1:100.000, con imágenes satelitales del sensor Landsat.

Independiente del método utilizado para identificar las coberturas y usos de la tierra con los productos de sensores remotos, en Colombia con el propósito de generar el mapa de coberturas y usos de la tierra se lleva a cabo en tres fases: 1) selección de la fuente de visualización (imagen, escala), 2) estructuración de leyenda preliminar de cobertura y uso 3) captura digital de la cobertura. La determinación y cuantificación de la cobertura y uso de la tierra, incluye los cambios que ocurren en ésta en el espacio y el tiempo y adicionalmente, las interacciones con la comunidad, estas son fundamentales para predecir el dinamismo que existe entre los componentes de un paisaje (Vásquez, 2011).

5.1.3 Métodos utilizados en la clasificación de imágenes

Una imagen obtenida por un sensor remoto, es una matriz rectangular que se compone de un número determinado de píxeles, los cuales representan un área específica en el espacio, y esta a su vez representa un valor de energía electromagnética emitida por una superficie en particular.

La investigación en sensores remotos ha sido foco de atención de la comunidad de teledetección porque los resultados obtenidos de esta clasificación son un insumo importante para diversas aplicaciones ambientales y socioeconómicas.

De acuerdo con Lu & Weng (2007) la clasificación por detección remota es un proceso complejo y requiere la consideración de muchos factores. Indicando que los principales pasos de la clasificación de imágenes pueden incluir la determinación de un sistema de clasificación adecuado, selección de muestras de entrenamiento, preprocesamiento de imágenes, extracción de características, selección de enfoques de clasificación adecuados, procesamiento posterior a la clasificación y evaluación de precisión. La necesidad del usuario, la escala del área de estudio, la condición económica y las habilidades del investigador son factores importantes que influyen en la selección de los datos obtenidos por teledetección, el diseño del procedimiento de clasificación y la calidad de los resultados de la clasificación.

En el campo de clasificación de imágenes en los últimos años se han utilizado varias tecnologías como las redes neuronales, sistemas expertos y la lógica difusa. Según Lu & Weng (2007) existen diferentes métodos de clasificación y técnicas para mejorar el rendimiento del proceso de clasificación. Entre algunos de los sistemas de clasificación que mencionan se encuentran: los orientados a objetos, por campo, contextuales, basados en subpixel y basados en pixel.

5.1.3.1 Clasificación basada en píxel

Los clasificadores basados en pixel son los más tradicionales, estos desarrollan una firma espectral combinando el espectro de todos los píxeles del conjunto de entrenamiento para una característica dada. Estos algoritmos de clasificación pueden ser supervisados y no supervisados (Melgani & Bruzzone, 2004).

Los clasificadores supervisados suponen que existe un conjunto de datos distribuido normalmente, y que los parámetros estadísticos (por ejemplo, el promedio del vector y la matriz de covarianza) generados a partir de las muestras de entrenamiento son representativos. Una desventaja principal radica en la dificultad de integrar datos espectrales con datos auxiliares. Son muy utilizados debido a su robustez y la disponibilidad en casi cualquier software de procesamiento de imágenes (DeFries et al., 2000; Hermes et al., 1999; Melgani & Bruzzone, 2002).

Los clasificadores no supervisados, no necesitan parámetros estadísticos para separar las clases de las imágenes. Investigaciones previas han indicado que los clasificadores no paramétricos pueden proporcionar mejores resultados de clasificación que los clasificadores paramétricos. Entre los enfoques de clasificación no paramétricos más utilizados se encuentran redes neuronales, árboles de decisión, máquinas de vectores de soporte y sistemas expertos (Sathya & Annamma, 2013).

La red neuronal tiene varias ventajas, su naturaleza no paramétrica, capacidad de límite de decisión arbitraria, fácil adaptación a diferentes tipos de datos y estructuras de entrada, valores difusos de salida y generalización para su uso con múltiples imágenes, lo que la convierte en una excelente opción para la clasificación de la cobertura terrestre. El perceptrón multicapa es tipo de red neuronal más utilizada en

la clasificación de imágenes. Sin embargo, la variación en la dimensionalidad de un conjunto de datos y las características de los conjuntos de entrenamiento y prueba pueden disminuir la precisión de la clasificación de imágenes (Lu & Weng ,2007).

5.1.3.2 Clasificación basada en subpixel

Esta metodología asume que la información espectral de cada píxel es la combinación lineal o no lineal de un grupo de materiales puros (vegetación, suelo, etc.), es decir un píxel está definido por un conjunto de valores espectrales que representan más de un objeto, existiendo la relación muchos a muchos, caso diferente a la clasificación basada en píxel, el cual es uno a muchos (Asner et al., 2009).

Lu & Weng (2007), mencionan y describen las características de diferentes clasificadores subpixel, tales como: ángulo espectral, esperanza – maximización, C-medias difuso, de mezclado espectral no lineal, entre otros. Una desventaja principal de la clasificación de subpíxel radica en la dificultad para evaluar la precisión.

5.1.3.3 Clasificadores contextuales

Es una técnica de clasificación, en la que se considerada conjuntamente las características espectrales y espaciales de la imagen, este clasificador tiene en cuenta la información espacial entre pixeles vecinos. Un clasificador contextual puede usar técnicas de suavizado, campos aleatorios de Markov, estadísticas espaciales, lógica difusa, segmentación o redes neuronales para mejorar sus resultados, siendo útil para la clasificación de imágenes de alta resolución. Esta metodología ha sido usada por diferentes algoritmos y clasificadores para mejorar la precisión de la clasificación, como la clasificación orientada a objetos, que es uno de los parámetros importantes para el proceso de la segmentación. (Lu & Weng, 2007).

5.1.3.4 Clasificadores basados por campo

El método consiste en subdividir una imagen en parcelas llamadas campos, utilizando la información vectorial, cada una de ellas, será la mínima unidad de la clasificación, de esta forma evita las variaciones de la información espectral de cada clase (Lu & Weng, 2007), esta clasificación es muy similar a la orientada por objetos

metodológicamente, ya que se basa en los límites de la información vectorial de la parcelación y la otra en la información propia de cada objeto para la segmentación.

Los sistemas de información geográfica (SIG) proporcionan un medio para implementar la clasificación por campo a través de la integración de datos vectoriales y datos de tipo matricial (raster), reduciendo los ruidos causados en otras clasificaciones debido a la diversidad de los paisajes en las coberturas de la tierra (Lu & Weng, 2007).

5.1.3.5 Clasificación orientada a objetos

La clasificación orientada a objetos es una metodología que se enfoca en encontrar un objeto (relación muchos a uno – píxeles a objeto y no por píxel como se realiza en la clasificación “por-píxel”) utilizando las formas geométricas, la información espectral y el análisis de vecindad definidos para un objeto en particular. El objeto es referido a la cobertura que se desea extraer de una clasificación (Al-Khudhairy et al., 2005).

Este tipo de clasificadores, no se basa exclusivamente en la información espectral que contiene un píxel, adicionalmente tiene en cuenta, las relaciones de formas y continuidades espaciales que se presenta para un objeto particular, aumentando los criterios y variables de la clasificación. En el proceso de la determinación de los objetos, la “escala” determina la presencia o ausencia de las clases, de tal manera que el tamaño del mismo afectaría el tamaño de la clasificación. La clasificación ha sido utilizada durante las últimas dos décadas, para determinar patrones específicos de una imagen, definiendo “el objeto” como, el conjunto de píxeles que enmarcan un solo contexto, tales como: edificaciones, zonas urbanas, puentes, caminos, coberturas, cultivos, bosques, minas, entre otras, ayudando de forma global, a la diferenciación entre los objetos que contiene una imagen (Vásquez, 2011).

5.1.4 Software existente orientado a la clasificación de imágenes

La teledetección es el proceso de detección y monitoreo de características físicas de un área, los equipos sensoriales como drones o satélites están equipados con cámaras especiales para recopilar imágenes que nos ayuden a detectar

características de la cobertura de la tierra. Para comprender mejor estas imágenes, se utiliza el software de procesamiento de imágenes.

El software de procesamiento de imágenes se usa para editar, analizar y mejorar una imagen analógica o digital para extraer información útil de ella. Existen en el Mercado una buena cantidad de software de procesamiento de imágenes. Por ejemplo, ERDAS IMAGINE, ENVI, IDRISI, eCognition y PCI. Los más usados en la actualidad son ERDAS IMAGINE, ENVI y eCognition. A continuación, se describen las características principales.

ERDAS (Earth Resources Data Analysis System)

Fue creado en Atlanta, USA, 1979. Es un software de procesamiento de imágenes digitales utilizado principalmente para el estudio y análisis de imágenes satelitales. Se puede usar para extraer valores numéricos digitales de los píxeles, importar y exportar rasters, combinar varias bandas de imágenes para realizar análisis detallados de diversos objetos e información utilizando la técnica de reconocimiento de patrones. También es posible realizar análisis de uso del suelo y cobertura de la tierra ("ERDAS VS ENVI: ¿Which is The Best Remote Sensing Image Processing Software?", 2018).

Ventajas

- Buen soporte.
- Interface amigable.
- Ampliamente usado en procesamiento de imágenes.
- Posee potentes algoritmos y funciones de procesamiento de datos.
- Se pueden integrar varias fuentes de datos.
- Es más usado en procesamiento geométrico.
- Muy usado en clasificación de uso del suelo.

Desventajas

- Alto costo.
- En clasificación no supervisada solo proporciona un algoritmo Isodata.
- El algoritmo Isodata requiere 6 parámetros.
- En clasificación supervisada proporciona 3 algoritmos.
- Para la realización de corrección geométrica necesita por lo menos 6 puntos.

ENVI (ENvironment for Visualizing Images)

Es un completo paquete de software para la visualización, análisis y despliegue de todo tipo de información geoespacial.

Fue creado en Boulder, USA, 1991. ENVI usa métodos científicos probados y procesos automatizados para ayudarle a transformar imágenes geoespaciales en conocimiento. Soporta múltiples sensores y tipos de datos. Las herramientas para análisis de ENVI pueden ser fácilmente desplegadas en la nube para obtener rápidamente resultados usando solo un botón ("ERDAS VS ENVI: ¿Which is The Best Remote Sensing Image Processing Software?", 2018).

Ventajas

- En clasificación no supervisada proporciona dos algoritmos Isodata.
- ENVI posee una mejor configuración de parámetros, opciones de clasificación y métodos más diversos.
- El algoritmo Isodata que posee ENVI requiere 10 parámetros.
- En clasificación supervisada proporciona 6 algoritmos.
- Para la realización de corrección geométrica necesita por lo menos 9 puntos.
- Es ampliamente usado para análisis espectral.

Desventajas

- Alto costo
- El procesamiento con grandes conjuntos de datos es costoso en tiempo y recursos

ECognition

eCognition developer es un potente entorno de desarrollo para el análisis de imágenes basado en objetos. Se utiliza en ciencias de la tierra para desarrollar conjuntos de reglas (o aplicaciones para eCognition Architect) para el análisis automático de datos de teledetección.

Se aplica a todas las tareas comunes de teledetección, como el mapeo de vegetación, la extracción de características, la detección de cambios y el reconocimiento de objetos. El enfoque basado en objetos facilita el análisis de todas las fuentes de datos comunes, como datos satelitales de resolución media a alta, fotografía aérea de alta a muy alta resolución, lidar, radar e incluso datos hiperespectrales (Kampouraki et al., 2008).

Ventajas

- Objetivo (las reglas y parámetros elegidos son subjetivos, pero las reglas se aplican a toda la imagen objetivamente).
- Representación multiescala.
- Conexión jerárquica entre multiescala.
- Uso de forma contexto y relaciones de vecindad.
- Reglas transferibles: límites reproducidos automáticamente a través diferentes conjuntos de datos.
- Métodos rápidos.

Desventajas

- Identificación de objetos de imagen, no objetos reales.
- Incapacidad para incluir o ignorar características de forma inteligente.
- Fusión de objetos reales debido a la confusión espectral (Kampouraki et al., 2008).

5.1.5 Sistemas existentes para el etiquetado de imágenes

La necesidad de datos de entrenamiento etiquetados para el reconocimiento de objetos ha aumentado debido a la difusión de enfoques de aprendizaje profundo. Estos enfoques a menudo requieren de decenas de miles de imágenes etiquetadas para mejorar las tareas de reconocimiento. Crear estos grandes conjuntos de imágenes de entrenamiento desde diferentes entornos y anotar los objetos grabados es una tarea significativa que requiere tiempo, experiencia y recursos.

Actualmente con el avance rápido de la tecnología se han desarrollado herramientas para hacer esta tarea de una forma más eficiente, a continuación, se presentan algunas herramientas que incluyen el servicio de etiquetado.

Amazon Rekognition

Amazon Rekognition facilita la adición de análisis de imagen y video a sus aplicaciones con tecnología probada, altamente escalable y de aprendizaje profundo que no requiere experiencia en aprendizaje automático para su uso. Con las etiquetas personalizadas de Amazon Rekognition, puede identificar los objetos y escenas en imágenes que son específicas para sus necesidades empresariales. Por ejemplo, puede encontrar su logotipo en publicaciones de redes sociales, identificar sus productos en los estantes de las tiendas, clasificar piezas de máquinas en una línea de montaje, distinguir plantas en buen estado o infectadas, o detectar sus personajes animados en videos ("Amazon Rekognition Image", 2021).

Clarifai

Clarifai ofrece una solución integral para modelar datos no estructurados para todo el ciclo de vida de la IA. Las soluciones de reconocimiento de imágenes, videos y texto

que ofrecen se basan en la plataforma de aprendizaje automático más avanzada y son fácilmente accesibles a través de diferentes dispositivos. Clarifai tiene un servicio llamado Scribe Labeler para crear datos de entrenamiento de alta calidad utilizando los flujos de trabajo de extremo a extremo y herramientas automatizadas, todo en una sola plataforma empresarial AI por conveniencia, velocidad y precisión ("Scribe Data Labeler | AI Data Annotation Tool | Clarifai", 2021).

Cloud Vision API

Con la API de Vision de Google Cloud, se ofrecen potentes modelos de aprendizaje automático previamente entrenados mediante las API de REST y RPC. Asigna etiquetas a las imágenes y las clasifica rápidamente en millones de categorías predefinidas. Detecta objetos y rostros, lee texto impreso y escrito a mano, y compila metadatos valiosos en tu catálogo de imágenes. Permite analizar imágenes en un sinnúmero de maneras, desde reconocer contenido explícito, hasta detectar emociones en las caras dentro de las imágenes. La API de Vision permite el etiquetado manual de imágenes y también puede usar el etiquetado automático que puede detectar y extraer información sobre las entidades de una imagen en un amplio grupo de categorías. Las etiquetas pueden identificar objetos generales, ubicaciones, actividades, especies de animales, productos entre otros ("Vision AI | Obtén información valiosa de las imágenes a través del AA", 2021).

Collect Earth Online

Collect Earth Online es un sistema de interpretación y visualización de imágenes satelitales de código abierto y personalizado. Es usado principalmente como herramienta en proyectos que requieren cobertura del suelo y/o datos de referencia del uso del suelo. Collect Earth Online promueve la coherencia en la localización, interpretación y etiquetado de parcelas de datos de referencia para su uso en la clasificación y seguimiento del cambio de cobertura y uso de la tierra. La funcionalidad completa de Collect Earth Online, incluida la compilación colaborativa de bases de datos de puntos de referencia ("Collect Earth Online: Open Foris", 2021).

Google Earth Engine

GEE combina un catálogo de varios petabytes de imágenes satelitales y conjuntos de datos geospaciales con capacidades de análisis a escala planetaria y lo pone a disposición de científicos, investigadores y desarrolladores para detectar cambios, mapear tendencias y cuantificar diferencias en la superficie de la Tierra.

GEE también proporciona una API y otras herramientas para permitir el análisis de grandes conjuntos de datos. Dentro de esta herramienta se puede hacer etiquetado de imágenes satelitales de forma manual. GEE está alojado en la nube, es gratuito para investigación, educación y para usos sin ánimo de lucro ("Google Earth Engine", 2021).

5.1.6 Validación de la clasificación de imágenes

Para validar los métodos de clasificación, se sugiere realizar el análisis de la exactitud o aciertos de las clasificaciones, comparándola con otra que el investigador declare como absoluta, proceso realizado utilizando una "Matriz de Confusión" y el "Coeficiente Kappa".

5.1.6.1 Matriz de confusión

La matriz de confusión (C) o contingencia, permite comparar dos clasificaciones: una de base definida por el usuario y la que se desea evaluar. Posteriormente se construye una comparación matricial de clases realizadas de la clasificación, ubicada generalmente en diferentes sectores o en la totalidad del mapa, comparando las clases de cada clasificación.

Con la matriz de confusión se obtienen tres tipos de exactitud: exactitud global, exactitud del usuario, exactitud del productor, cada uno de estos está definido en la Tabla 1 (Vásquez, 2011).

Tabla 1. Tipos de exactitud para la validación de la clasificación de imágenes (Vásquez, 2011)

Exactitud	Explicación
Global (EG)	Indica la exactitud del conjunto de las clases del método a evaluar. $EG = \frac{\sum_{i=1}^r X_{ii}}{N}$, X_{ii} : diagonal mayor de C, N: número total de puntos de muestreo, r: número de filas de la matriz
De Usuario (EU)	Es el acertamiento de una clase en particular, respecto a toda la clasificación $EU = \frac{X_{ii}}{X_{i+}} \times 100$, X_{ii} : diagonal de dicha fila, X_{i+} : totales marginales de la fila i
De Productor (EP)	Es el acertamiento de una clase en particular, respecto a la clasificación a evaluar $EP = \frac{X_{ii}}{X_{+i}} \times 100$, X_{ii} : diagonal de dicha columna, X_{+i} : totales marginales de la columna i

Estas medidas de exactitud, se basan únicamente en resultados parciales de la matriz, por lo que no aprovechan todos los datos en su totalidad, al mismo tiempo se toman resultados aleatorios, que pueden conducir a interpretaciones sesgadas del método. Para corregir los tipos de exactitud, se utiliza el “Coeficiente Kappa”.

5.1.6.2 Coeficiente de Validación Kappa

Este coeficiente, propuesto por Jacob Cohen en 1960, es un coeficiente estadístico que mide la concordancia de dos metodologías. Es utilizado para ver la similitud de dos clasificaciones de imágenes, realizado por medio de la ecuación (1):

$$CK = \frac{\sum_{i,j=1}^r x_{ij} - \sum_{i,j=1}^r (\sum x_i \cdot \sum x_j)}{N^2 - \sum_{i,j=1}^r (\sum x_i \cdot \sum x_j)}$$

(1) Vásquez, 2011

En donde r es el número de filas en la matriz; x_{ij}, número de píxeles de la fila i, columna j, es decir la diagonal mayor; x_i, fila i; x_j, columna j y N el total de píxeles de la matriz. El valor de coeficiente puede variar de 0 a 1, entre más cercano esté a 1, quiere decir que la concordancia de los dos métodos es muy alta.

De acuerdo al valor, éste índice se puede categorizar en seis clases (Tabla 2) de acuerdo a esta clasificación se puede rechazar o aceptar la nueva clasificación a evaluar.

Tabla 2. Categoría de concordancias de validación para el coeficiente Kappa (Vásquez, 2011)

Rango	Concordancia
0	Nula
0.01 - 0.02	Leve
0.21 - 0.40	Aceptable
0.41 - 0.60	Moderada
0.61 - 0.80	Considerable
0.81 - 1.00	Casi perfecta

5.1.7 Redes neuronales artificiales

Las redes neuronales artificiales se componen de unidades simples de procesamiento, las cuales son llamadas “neuronas”, desde el punto de vista biológico una neurona es un interruptor entre varias señales de entrada y una de salida. Este interruptor se activa si las entradas a la neurona alcanzan un determinado umbral de activación, entonces el pulso de salida será emitido por la neurona como resultado de su activación, en la figura 1, se muestran los componentes de una neurona.

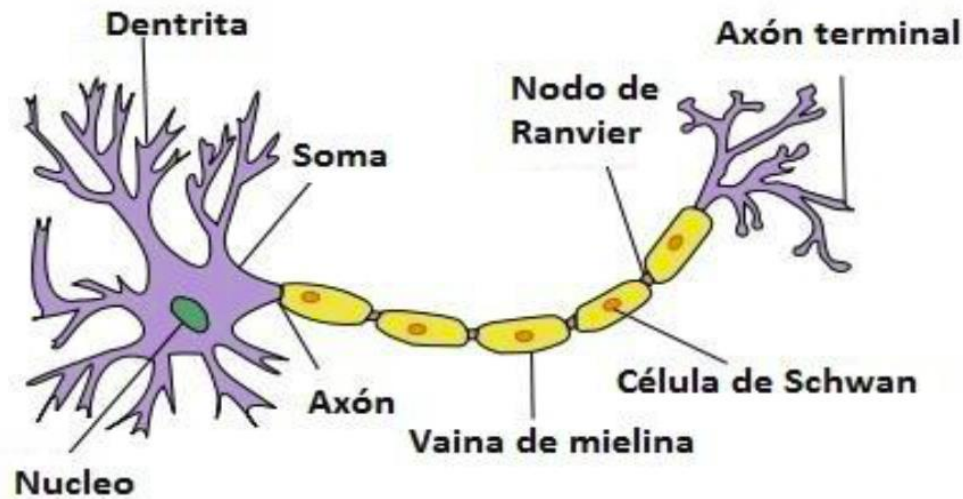


Figura 1. Elementos principales de una neurona (Hernández, 2014)

El punto de entrada de las neuronas son las dendritas, las cuales reciben la información mediante conexiones especiales llamadas sinapsis. Después la señal es transmitida a través del axón, desde el cual la señal es transformada mediante un proceso químico, para después ser transmitida a las neuronas conectadas a esta (Hernández, 2014).

El modelo biológico de las neuronas puede ser trasladado a un proceso técnico, donde las consideraciones más importantes se mencionan a continuación:

Entrada Vectorial: La entrada de una neurona artificial se compone de varios elementos, de tal forma que puede definirse como un vector, ecuación (2).

$$\vec{x}, x \subseteq \vec{x}, x \in \mathcal{R} \quad (2) \text{ Hernández, 2014}$$

La Sinapsis cambia las señales de entrada: En las redes neuronales artificiales, las entradas son multiplicadas por un número (“peso sináptico”), este conjunto de pesos representa la información almacenada de una red neuronal artificial, ecuación (3).

$$\sum_i w_i x_i$$

(3) Hernández, 2014

Salida Escalar: La salida de una neurona artificial es un escalar, lo cual significa que una neurona está compuesta de un solo componente, el cual suma las señales de entrada a la neurona y produce solo una señal una señal de salida, ecuación (4).

$$y = f\left(\sum_i w_i x_i\right)$$

(4) Hernández, 2014

Características no lineales: las entradas de una neurona artificial no son proporcionales a la salida de la misma.

Pesos ajustables: los pesos que ponderan las entradas de la neurona son variables. Esto añade una gran dinamicidad a las redes neuronales artificiales, debido a la gran cantidad de conocimiento que se encuentra almacenado en sus pesos (Hernández, 2014).

A continuación, se muestra una representación gráfica de una red neuronal artificial:

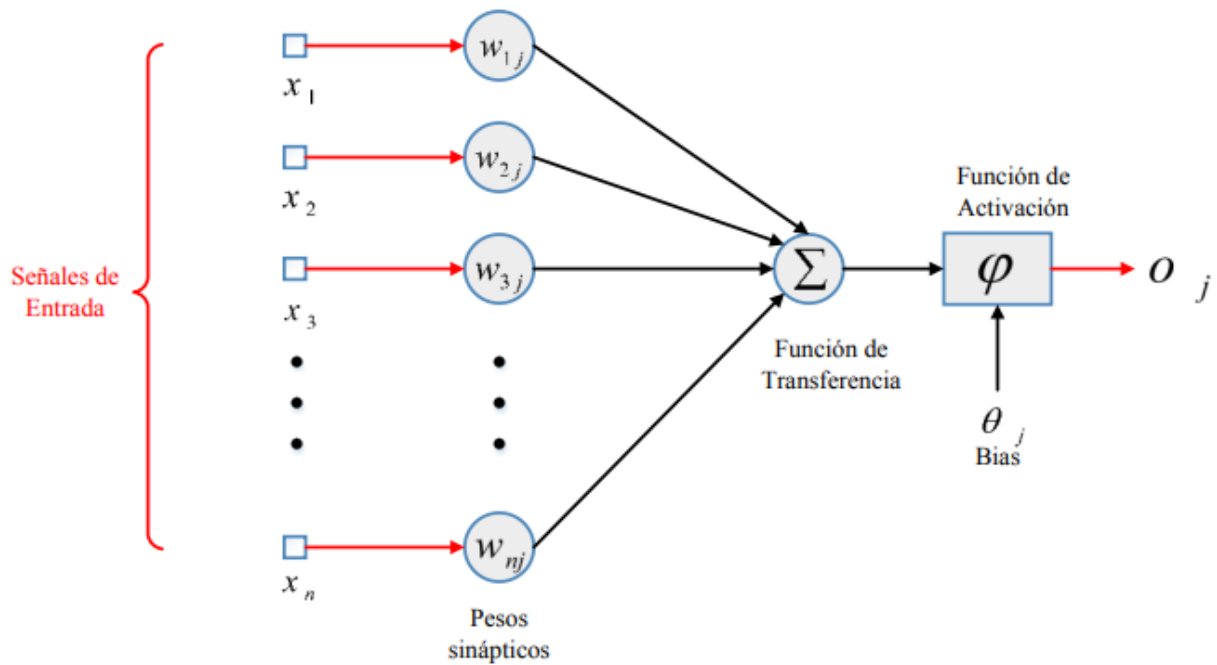


Figura 2. Esquema general de una Red Neuronal Artificial (Hernández, 2014).

Un aspecto importante de las redes neuronales artificiales es su capacidad de familiarizarse con los problemas, esto se da como resultado de un entrenamiento, una vez este entrenamiento ha finalizado, la red neuronal será capaz de resolver nuevos problemas pertenecientes a la misma clase de entrenamiento. Este proceso es conocido como generalización (Hernández, 2014).

El concepto de Redes Neuronales Artificiales surgió en un intento de Warren McCulloch y Walter Pitts en 1943 por simular los procesos que ocurren en el cerebro. Hoy en día existe una gran cantidad de redes neuronales, las cuales pueden ser divididas en:

Redes Neuronales de Primera Generación, también conocidas con el nombre de perceptrones, estas se componen de dos secciones una sumatoria y un nivel de umbral. La parte de la sumatoria recibe la información de entrada ponderada por los pesos sinápticos y realiza la función umbral sobre el resultado de la suma, los datos de entrada y salida pueden ser 0's y 1's (Hernández, 2014).

Redes Neuronales de Segunda Generación se componen de neuronas que realizan el cómputo en dos etapas, la suma de los valores recibidos a través de la sinapsis, y una función Sigmoide, cuya entrada es la suma de los parámetros de entrada ponderados por los pesos sinápticos, esta función se denomina como función de transferencia (Hernández, 2014).

Redes Neuronales de Tercera Generación o redes neuronales pulsantes enfocadas aplicadas en temas de comportamiento adaptativo en robots autónomos. (Hernández, 2014)

5.1.7.1 Características de las redes neuronales

Existen cuatro aspectos que caracterizan una red neuronal: su topología, el mecanismo de aprendizaje, tipo de asociación realizada entre la información de entrada y salida, y la forma de representación de estas informaciones (Zapata & Díaz, 2008).

Topología

La arquitectura de las redes neuronales consiste en la organización y disposición de las neuronas formando capas más o menos alejadas de la entrada y salida de la red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas.

Redes Monocapa

Se establecen conexiones laterales, cruzadas o autorecurrentes entre las neuronas que pertenecen a la única capa que constituye la red. Se utilizan en tareas relacionadas con lo que se conoce como autoasociación; por ejemplo, para generar informaciones de entrada que se presentan a la red incompletas o distorsionadas.

Redes Multicapa

Son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas. Una forma de distinguir la capa a la que pertenece la neurona, consiste en

fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. De acuerdo al tipo de conexión se distinguen las redes feedforward, y las redes feedforward/feedback (Zapata & Díaz, 2008).

5.1.7.2 Mecanismo de aprendizaje

Es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios producidos durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre las neuronas, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor diferente de cero, una conexión se destruye cuando su peso pasa a ser cero. Se puede afirmar que el proceso de aprendizaje ha finalizado (la red ha aprendido) cuando los valores de los pesos permanecen estables.

Un criterio para diferenciar las reglas de aprendizaje se basa en considerar si la red puede aprender durante su funcionamiento habitual, o si el aprendizaje supone la desconexión de la red.

Otro criterio suele considerar dos tipos de reglas de aprendizaje: las de aprendizaje supervisado y las de aprendizaje no supervisado, estas reglas a clasificaciones que se realizan de las RNA. (Zapata & Díaz, 2008).

La diferencia entre estos dos tipos de RNA estriba en la existencia o no de un agente externo (supervisor) que controle el aprendizaje de la red.

Redes con Aprendizaje Supervisado

El proceso de aprendizaje se realiza a través de un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor comprueba la salida de la red y en el caso de que ésta no coincide con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida se aproxime a la deseada. Se consideran tres diferentes formas de llevar a cabo este tipo de aprendizaje:

Aprendizaje por corrección de error: Consiste en ajustar los pesos en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, en función del error.

Aprendizaje por refuerzo: Se basa en la idea de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada. El papel del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito= +1 o fracaso= -1), y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades.

Aprendizaje estocástico: Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad (Zapata & Díaz, 2008).

Redes con Aprendizaje No Supervisado

Estas redes no requieren influencia externa para ajustar los pesos de las conexiones entre neuronas. La red no recibe ninguna información por parte del entorno si la salida generada es o no correcta, así que existen varias posibilidades en cuanto a la interpretación de la salida de estas redes.

La salida puede representar en algunos casos el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se han presentado en el pasado. En otro caso podría realizar una codificación de los datos de entrada, generando en la salida una versión codificada de la entrada, con menos bits, pero manteniendo la información relevante de los datos, también algunas redes con aprendizaje no supervisado lo que realizan es un mapeo de características, obteniéndose en las neuronas de salida una disposición geométrica de tal forma que si se presentan a la red informaciones similares, siempre sean afectadas neuronas de salidas próximas entre sí, en la misma zona del mapa. En general este tipo de aprendizaje se suelen considerar dos tipos:

Aprendizaje competitivo y cooperativo: Las neuronas compiten (y cooperan) unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje se pretende que cuando se presente a la red cierta información de entrada, solo una de las neuronas se active (alcance su valor de respuesta máximo). Por lo tanto, las neuronas compiten por activarse, quedando finalmente una, o una por grupo, como neurona vencedora. (Zapata & Díaz, 2008).

Las últimas tendencias en el concepto de redes neuronales vinieron acompañadas con el aumento en el poder de computación y el incremento de la experiencia. Esto impulsó el desarrollo de lo que conocemos como Deep learning o aprendizaje profundo, donde las estructuras tecnológicas de las redes neuronales se han vuelto cada vez más complejas y capaces de resolver una amplia gama de tareas que antes no podían ser resueltas eficazmente como la clasificación de imágenes (Zapata & Díaz, 2008).

5.1.8 Deep Learning

El aprendizaje profundo (Deep Learning) es una técnica de aprendizaje automático que enseña a los ordenadores a hacer lo que resulta natural para las personas: aprender mediante ejemplos. El aprendizaje profundo es una tecnología clave presente en los vehículos sin conductor que les permite reconocer una señal de stop o distinguir entre un peatón y una farola. Resulta fundamental para el control mediante voz en dispositivos tales como teléfonos, tabletas, televisores y altavoces manos libres. El aprendizaje profundo atrae mucha atención últimamente, y hay razones de peso para ello. Está consiguiendo resultados que antes no eran posibles.

Con el aprendizaje profundo, un modelo informático aprende a realizar tareas de clasificación directamente a partir de imágenes, texto o sonido. Los modelos de aprendizaje profundo pueden obtener una precisión de vanguardia que, en ocasiones, supera el rendimiento humano. Los modelos se entrenan mediante un amplio conjunto de datos etiquetados y arquitecturas de redes neuronales que contienen muchas capas ("Aprendizaje profundo: Tres cosas que es necesario saber", 2018).

Aunque las primeras teorías sobre el aprendizaje profundo se desarrollaron en la década de los ochenta, existen dos razones principales por las que solo ha empezado a resultar útil recientemente:

El aprendizaje profundo requiere grandes cantidades de datos etiquetados. Por ejemplo, para el desarrollo de un vehículo sin conductor se necesitan millones de imágenes y miles de horas de vídeo.

El aprendizaje profundo requiere una potencia de cálculo significativa. Las GPU de alto rendimiento tienen una arquitectura paralela que resulta eficiente para el aprendizaje profundo. En combinación con clusters o con el cálculo en la nube, esto permite a los equipos de desarrollo reducir el tiempo necesario para el entrenamiento de una red de aprendizaje profundo de semanas a horas o incluso menos.

La mayor parte de los métodos de aprendizaje emplean arquitecturas de redes neuronales, por lo que, a menudo, los modelos de aprendizaje profundo se denominan redes neuronales profundas.

El término “profundo” suele hacer referencia al número de capas ocultas en la red neuronal. Las redes neuronales tradicionales sólo contienen dos o tres capas ocultas, mientras que las redes profundas pueden tener hasta 150. La figura 3 muestra como está organizada una red neuronal profunda.

Los modelos de Deep Learning se entrenan mediante el uso de extensos conjuntos de datos etiquetados y arquitecturas de redes neuronales que aprenden directamente a partir de los datos, sin necesidad de una extracción manual de características ("Aprendizaje profundo: Tres cosas que es necesario saber", 2018).

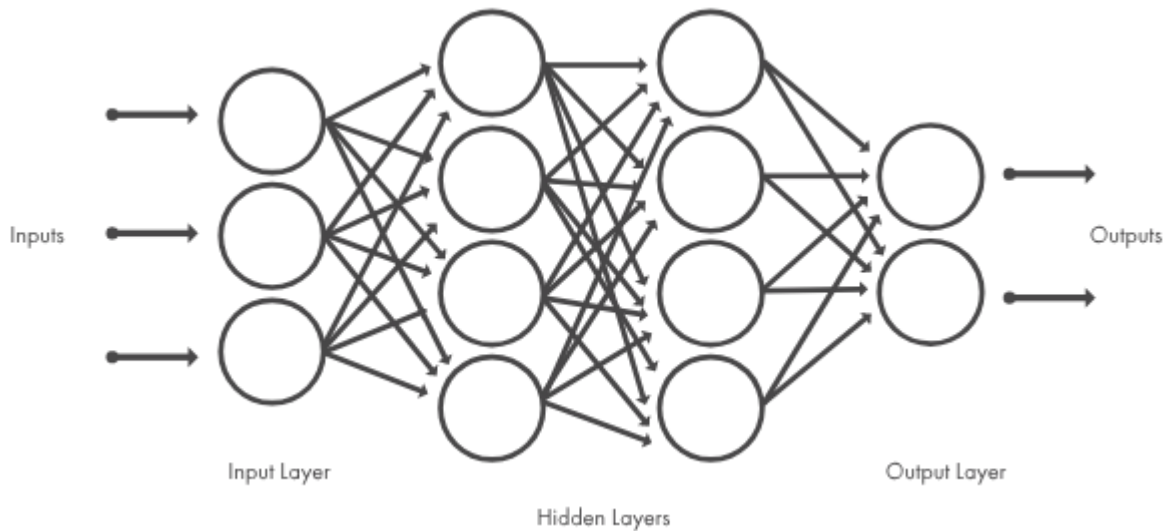


Figura 3. Redes neuronales organizadas en capas que constan de una serie de nodos interconectados. Las redes pueden tener decenas o cientos de capas ocultas ("Aprendizaje profundo: Tres cosas que es necesario saber", 2018).

5.1.9 Redes Neuronales convolucionales

Uno de los tipos más populares de redes neuronales profundas son las conocidas como redes neuronales convolucionales (CNN o ConvNet). Una CNN convoluciona las características aprendidas con los datos de entrada y emplea capas convolucionales 2D, lo cual hace que esta arquitectura resulte adecuada para procesar datos 2D, tales como imágenes.

Las redes neuronales convolucionales (CNN por sus siglas en inglés) es una arquitectura especial de redes neuronales artificiales, propuesta por Yann LeCun en 1988. Las redes neuronales convolucionales utilizan algunas características de la corteza visual. Uno de aplicaciones más populares de esta arquitectura es la clasificación de imágenes. Algunos ejemplos de su uso son:

Facebook las usa para algoritmos de etiquetado automático (Taigman et al., 2014),

Amazon las usa para generar recomendaciones de productos (Hidasi, et al., 2015)

Google las usa en la clasificación de videos a gran escala (Karpathy, et al., 2014).

Conducción autónoma (Chen et al., 2015)

Investigación médica (Havaei et al., 2015)

Las CNN eliminan la necesidad de una extracción de características manual, por lo que no es necesario identificar las características utilizadas para clasificar las imágenes. La CNN funciona mediante la extracción de características directamente de las imágenes. Las características relevantes no se entrenan previamente; se aprenden mientras la red se entrena con una colección de imágenes. Esta extracción de características automatizada hace que los modelos aprendizaje profundo sean muy precisos para áreas de visión artificial, como por ejemplo la clasificación de objetos (Figura 4).

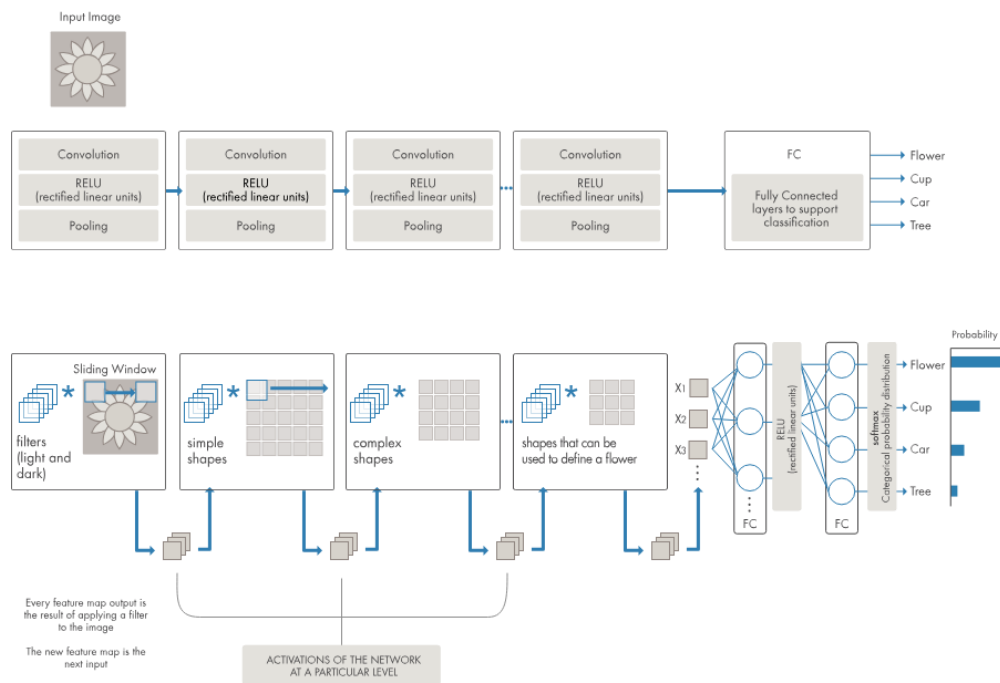


Figura 4. Ejemplo de una red con muchas capas convolucionales. Se aplican filtros a cada imagen de entrenamiento con distintas resoluciones, y la salida de cada imagen convolucionada se emplea como entrada para la siguiente capa ("Aprendizaje profundo: Tres cosas que es necesario saber", 2018).

Las CNN aprenden a detectar diferentes características de una imagen mediante decenas o cientos de capas ocultas. Cada capa oculta aumenta la complejidad de las características de la imagen aprendidas. Por ejemplo, la primera capa oculta podría aprender cómo detectar bordes, mientras la segunda aprende cómo detectar formas más complejas propias de la forma del objeto que se intenta reconocer (Sorokina, 2018).

5.1.9.1 Clasificación de imágenes con redes neuronales convolucionales CNN

La tarea principal de la clasificación de imágenes es la recepción de la imagen de entrada y después la definición de su clase. Esta habilidad es desarrollada por las personas desde su nacimiento, quienes pueden determinar fácilmente los elementos específicos presentes en una imagen. Pero el computador ve las imágenes de una manera bastante diferente (figura 5).

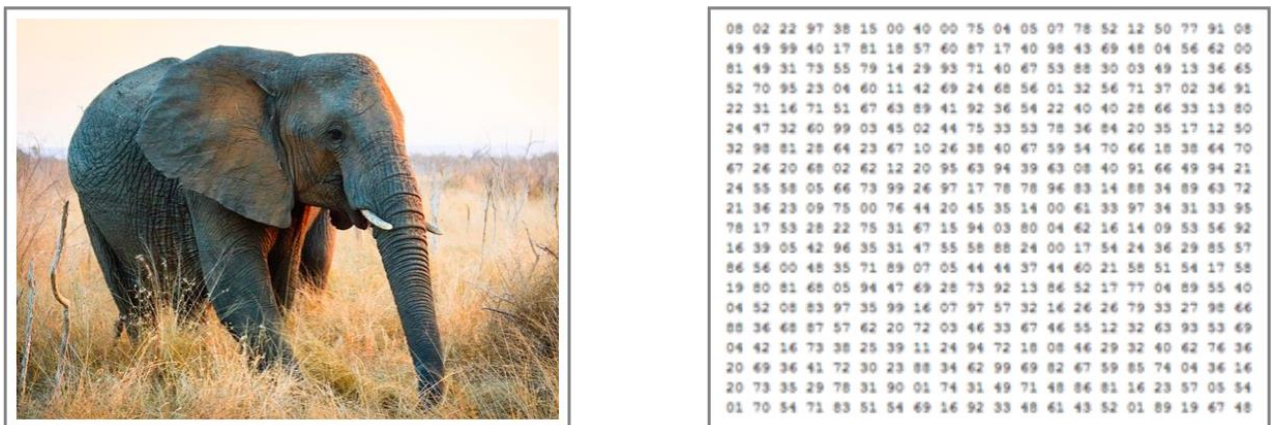


Figura 5. Lado izquierdo lo que el ojo humano percibe, lado derecho lo que la computadora percibe (Sorokina, 2018).

En lugar de la imagen, el computador ve un arreglo de píxeles. Por ejemplo, en una imagen de 300 x 300 píxeles, el tamaño del arreglo será de 300 x 300 x 3, donde el primer valor es el ancho, el segundo valor es el alto y el tercer valor es el canal RGB.

Al computador se le asigna un valor de 0 a 255 a cada uno de estos valores. Estos valores describen la intensidad del píxel en cada punto.

Para resolver este problema, el computador busca las características del nivel base. Esto en relación a la comprensión humana son características tales como el tronco o las orejas grandes. Para el computador, estas características son límites o curvaturas. Luego, a través de los grupos de capas convolucionales, la computadora construye conceptos más abstractos. En un nivel más detallado esto quiere decir que la imagen se pasa a través de una serie de capas convolucionales, no lineales, capas agrupadas y capas completamente conectadas, y luego genera la salida (Sorokina, 2018).

5.1.9.2 Operaciones principales de redes convolucionales

Existen cuatro operaciones principales que conforman las redes convolucionales.

Capa convolucional

Su principal objetivo es extraer características de una imagen. Consiste en un conjunto de filtros entrenables que realizan producto punto con los valores de la capa precedente. En la práctica, los valores de los filtros son aprendidos para su activación al encontrar ciertas características. Al ser colocados en cascada se obtienen diferentes niveles de abstracción.

Rectificador Lineal de Unidad

Son utilizados después de cada convolución. Son una operación que reemplaza los valores negativos por cero y su propósito es agregar no linealidad al modelo, eliminando la relación proporcional entre la entrada y salida.

Pooling

Algoritmo utilizado para reducir las dimensiones, con el objetivo de disminuir los tiempos de procesado reteniendo la información más importante.

Capa totalmente conectada (Dense Layer)

Realiza la clasificación basado en las características extraídas por las capas de convolución y las reducidas por pooling. En esta capa todos los nodos están conectados con la capa precedente.

5.1.9.3 Funcionamiento De La Red Convolutiva

Las capas de convolución y las de pooling se encargan de extraer características mientras que la capa totalmente conectada actúa como clasificador. Para el funcionamiento de este modelo debemos proceder al entrenamiento. Esto implica: 1) Inicializar todos los parámetros o pesos con valores aleatorios; 2) Utilizar una imagen de entrenamiento y utilizarla en el modelo; 3) Calcular el error total de las probabilidades resultantes del modelo y finalmente; 4) Propagar hacia atrás para calcular el error de gradiente de todos los pesos en la red y utilizar gradiente descendiente para actualizar estos valores y minimizar el error de salida.

La Figura 6 muestra la secuencia de transformaciones involucradas en una red convolutiva típica.

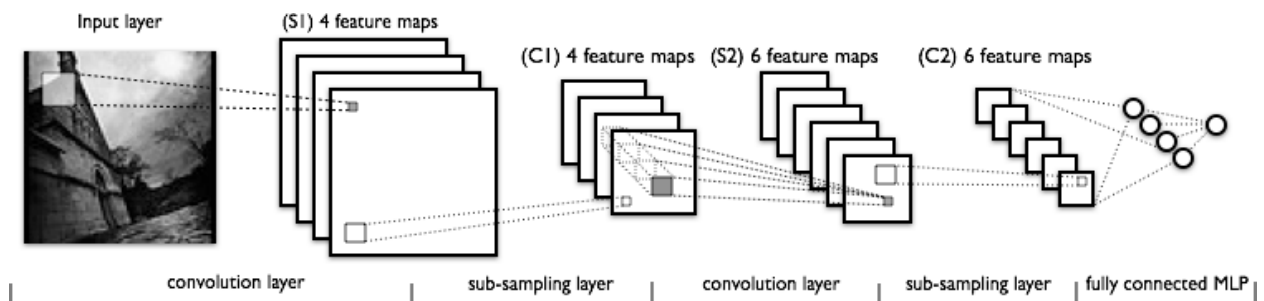


Figura 6. Secuencia de transformaciones en una red convolutiva. Fuente: "Convolutional Neural Network (CNN)", 2018

En la imagen se puede observar de izquierda a derecha:

- La imagen de entrada real que se escanea en busca de características. El rectángulo de luz es el filtro que pasa sobre él.
- Mapas de activación apilados uno encima de otro, uno para cada filtro que emplee. El rectángulo más grande es un parche para ser muestreado.
- Los mapas de activación se condensaron a través de una reducción de resolución.
- Un nuevo conjunto de mapas de activación creados al pasar filtros sobre la primera pila de la muestra reducida.

- La segunda reducción de resolución, la cual condensa el segundo conjunto de mapas de activación.
- Finalmente, una capa completamente conectada que clasifica el resultado con una etiqueta por nodo.

La salida del modelo da una probabilidad de que una clase dada esté presente en una imagen dada.

5.1.9.4 Antecedentes de uso de redes neuronales convolucionales y deep learning

Los métodos de clasificación de imágenes descritos en la sección 5.1.3 generalmente usan solo información espectral de las imágenes. Las imágenes de alta resolución (resolución espacial de 2m y más) contienen más detalles sobre el terreno, esto implica que la cantidad de píxeles que se deben analizar es más grande. La cobertura terrestre tiene varios tipos, y se ve afectada por el ruido, la iluminación, estación, entre otros factores, lo cual hace que se presenten dificultades cuando se quiere hacer una clasificación en estas imágenes de alta resolución.

Teniendo en cuenta la perspectiva del reconocimiento de patrones la selección y extracción de características representativas es el cuello de botella para mejorar la precisión. Esto quiere decir que con el uso de un conjunto específico de características no se puede lograr la clasificación de todo tipo de objetos terrestres.

La clasificación automática de la cobertura terrestre de las imágenes de teledetección de alta resolución es una capacidad muy deseable en la actualidad, pero presenta muchos desafíos debido al gran volumen y variedad de imágenes satelitales y aéreas recopiladas a diario. Esto se complica aún más por la gran variedad de características y escalas relacionadas con los seres humanos presentes en áreas geográficamente diversas en la superficie de la tierra.

Las redes neuronales convolucionales (CNN) y el aprendizaje profundo han experimentado un resurgimiento en los últimos años, realizando finalmente la promesa de su desarrollo conceptual en la década de 1980 a partir de estudios como (Fukushima et al.,1983). Las técnicas de aprendizaje profundo, como Caffe

framework que proporciona a los científicos y profesionales de multimedia un marco limpio y modificable para el uso de algoritmos avanzados de aprendizaje y una colección de modelos de referencia. El framework es una biblioteca C++ con licencia BSD con Python y enlaces MATLAB para entrenamiento y despliegue de propósito general, redes neuronales convolucionales y otros modelos profundos eficientes en arquitecturas de productos básicos. Caffe permite la experimentación entre plataformas para facilidad de desarrollo e implementación de prototipos de máquinas en ambientes en la nube (Jia et al., 2014).

En las técnicas de deep learning, las redes neuronales convolucionales se ha logrado resultados notables en la clasificación de imágenes, reconocimiento y tiene el puntaje más alto en muchas bases de datos visuales como ImageNet, análisis de patrones, modelado estadístico, Computational Learning Visual Object Classes (PASCAL VOC), and Microsoft Common Objects in Context (MS-COCO). (Fu, G et al., 2017).

GoogLeNet es una red cuya calidad se evalúa en el contexto de la clasificación y la detección. La red fue diseñada teniendo en mente la eficiencia computacional y la practicidad, de modo que la inferencia se puede ejecutar en dispositivos individuales, incluso aquellos con recursos computacionales limitados, especialmente con una huella de memoria baja. La red tiene 22 capas de profundidad al contar solo capas con parámetros (o 27 capas si también se cuenta el agrupamiento). El número total de capas (bloques de construcción independientes) utilizados para la construcción de la red es de aproximadamente 100 (Szegedy et al., 2014).

ResNet es un framework de marco de aprendizaje residual para facilitar el entrenamiento de redes que son sustancialmente más profundas que las utilizadas previamente. Reformulando explícitamente las capas como funciones de aprendizaje residuales con referencia a las capas de entrada, en lugar de aprender funciones sin referencia. Proporcionan evidencia empírica que muestra que mediante estas técnicas son más fáciles de optimizar y pueden ganar precisión manejando una mayor profundidad (He et al., 2016).

TensorFlow es una biblioteca de software de código abierto para el cálculo numérico que utiliza gráficos de flujo de datos. Los nodos de gráfico representan operaciones matemáticas, mientras que los bordes de gráfico representan los conjuntos de datos multidimensionales (tensores) que fluyen entre ellos. Esta arquitectura flexible le permite implementar computación en una o más CPU o GPU en una computadora de escritorio, servidor o dispositivo móvil sin reescribir el código. TensorFlow también incluye TensorBoard, un kit de herramientas de visualización de datos ("tensorflow/tensorflow", 2018).

Eclipse Deeplearning4j es la primera biblioteca de aprendizaje profundo distribuida, de código abierto, de grado comercial, escrita para Java y Scala. Integrado con Hadoop y Apache Spark, DL4J lleva a IA a entornos empresariales para su uso en GPU y CPU distribuidas (Chris V. Nicholson, 2018).

Los métodos y framework mencionados anteriormente se han establecido recientemente como los métodos de mayor rendimiento para una amplia variedad de tareas de aprendizaje automático.

Este resurgimiento en el aprendizaje profundo ha creado muchas vías nuevas y prometedoras de investigación. Las redes neuronales profundas (DCNN) son una arquitectura neuronal profunda común que intenta imitar la capacidad del cerebro para aprender y desarrollar representaciones de características jerárquicas dentro de una fase de extracción de características multinivel. Estas características visuales aprendidas proporcionan a la fase de clasificación las características necesarias para discriminar imágenes en sus respectivas clases. El aprendizaje profundo a menudo supera a los extractores convencionales de funciones de visión artificial, como los mencionados anteriormente (Scott et al., 2016).

Por ejemplo, GoogLeNet (Szegedy et al., 2014) fue el ganador en ILSVRC o ImageNet Large Scale Visual Recognition Challenge en 2014, usando un subconjunto de imágenes de ImageNet de 1.2 millones de entrenamiento, clasificadas en más de 100 categorías, 50.000 de validación y 150000 de prueba, alcanzando una precisión de 93.5%. En el mismo evento en 2015 ResNet (He et al., 2016), usando un conjunto

de datos igual entrenando una red neuronal de 152 capas logró una tasa de error de 3.57%. CaffeNet (Jia et al., 2014) también ha sido aplicado en bases de datos como MNIST (Mixed National Institute of Standards and Technology), una base de datos de dígitos escritos a mano, un total de 60 000 imágenes de dígitos se utilizaron como conjunto de entrenamiento y 10 000 otras constituyeron el conjunto de prueba, que se utilizó luego en la etapa de predicción de dígitos (categorización), alcanzando una precisión en la clasificación de 94% (Kovalev et., 2016).

Sin embargo, DCNN requiere conjuntos de datos de entrenamiento etiquetados muy grandes para permitir que las redes complejas aprendan suficiente de extracción de características y clasificación simultáneamente. Desafortunadamente, los grandes conjuntos de datos de imágenes con etiquetas no están disponibles. Para mitigar la disponibilidad limitada de los datos de entrenamiento etiquetados, se puede acelerar el aprendizaje aplicando el aprendizaje por transferencia (TL) también conocido como transferencia inductiva (Scott et al., 2016) de modelos preconfigurados de DCNN, CaffeNet (Jia et al., 2014), GoogLeNet (Szegedy et al., 2014) y ResNet (He et al., 2016), y se puede ampliar el entrenamiento con imágenes de teledetección. De hecho, Scott et al, (2016) mostraron que, al inicializar el aprendizaje profundo con un modelo preestablecido, se disminuye en gran medida la cantidad de tiempo de entrenamiento requerido al reutilizar y perfeccionar los pesos ya aprendidos en la fase de extracción de características.

6. METODOLOGIA DE LA INVESTIGACIÓN

Esta investigación consiste en el desarrollo de un prototipo de software para la clasificación de coberturas en imágenes que son tomadas por drones empleando técnicas de deep learning conjuntamente con la definición de una red neuronal convolucional. Se opta por el uso de estas redes neuronales puesto que en la literatura consultada presenta un alto rendimiento en la clasificación de imágenes (Krizhevsky et al., 2012; Karpathy et al., 2014; Taigman et al., 2014). Con el deep learning, un modelo informático puede aprender a realizar tareas de clasificación directamente a partir de imágenes. Los modelos de aprendizaje profundo pueden obtener una precisión de vanguardia que, en ocasiones, supera el rendimiento

humano. Los modelos se entrenan mediante un amplio conjunto de datos etiquetados y arquitecturas de redes neuronales que contienen varias capas.

La elección de las técnicas de deep learning y la construcción de la red neuronal convolucional se orientan en esta investigación a obtener un modelo tal que, dada una imagen, este pueda detectar automáticamente las diferentes coberturas seleccionadas por los investigadores (árboles, frutales, café, suelo desnudo, cultivo, pastizales) presentes en la imagen, haciendo más eficiente este proceso, permitiendo contar con la información en menor tiempo para análisis posteriores por parte de los investigadores del CIAT.

Para el ciclo de vida de la aplicación web de etiquetado y del prototipo de clasificación de coberturas se utilizó el modelo incremental, la principal característica es entregar el software en partes pequeñas pero utilizables llamadas incrementos. En general cada incremento se construye sobre aquel que ya ha sido entregado. Aunque al principio ya teníamos una idea clara de lo que se buscaba y teníamos los principales requisitos del sistema se decidió ir realizando la construcción del sistema haciendo entregas teniendo en cuenta los distintos módulos que se identificaron en los requisitos como son:

- Módulo de usuarios
- Módulo de roles
- Módulo de etiquetas
- Módulo de experimentos
- Prototipo de clasificación de coberturas

Para tener una idea general y clara de las diferentes fases del proyecto en la figura 7 se puede observar las diferentes etapas involucradas.

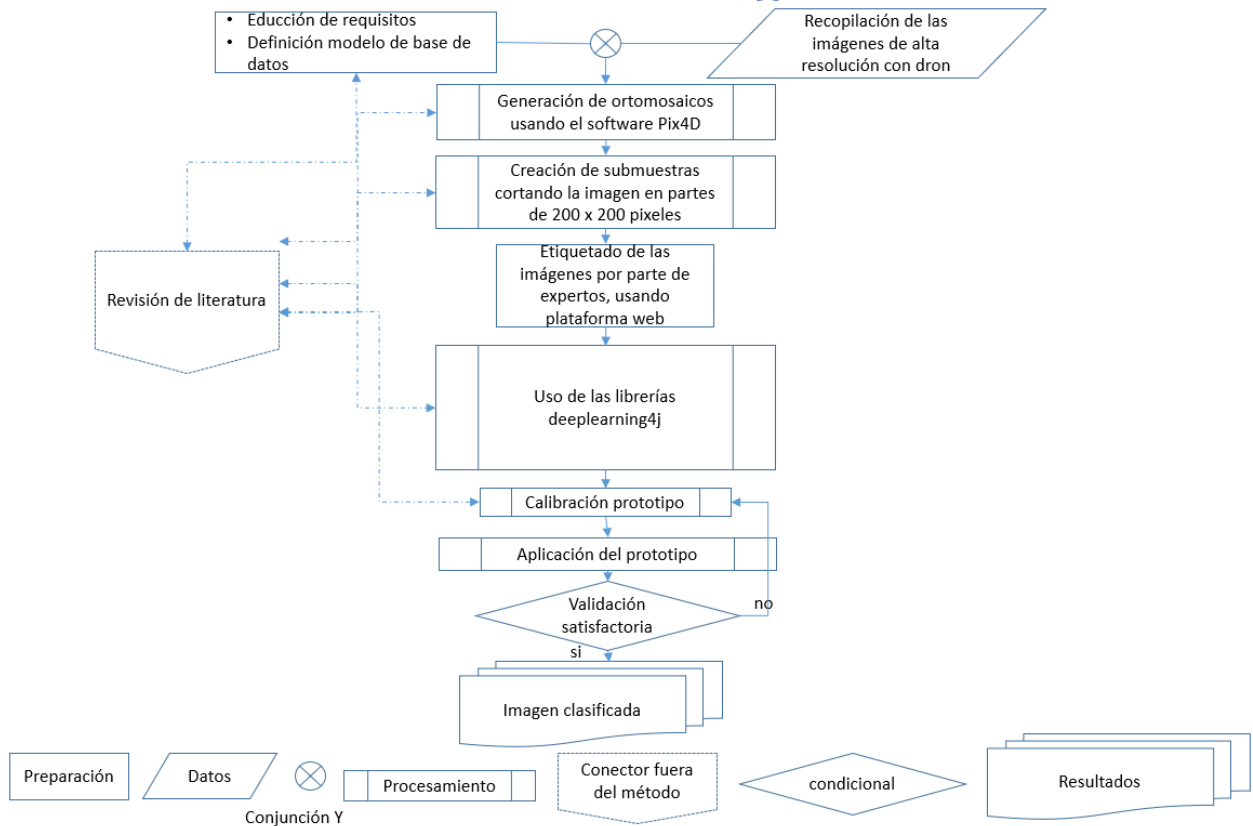


Figura 7. Fases del proceso metodológico.

6.1 Recopilación de las imágenes de alta resolución con drones

Las primeras imágenes a utilizar en este proyecto son imágenes RGB de alta resolución tomadas por drones Phantom 3 advanced, Phantom 4 y Mavick Air las cuales fueron tomadas en los años 2017, 2018 y 2019 y pertenecen al Valle del Cauca, Colombia, Ahuachapán, El Salvador y Ocotepeque, Honduras, estas imágenes fueron tomadas entre los 100m a 150m de altura y cubren un área entre 6.7 y 7.3 ha. La figura 8 muestra cómo se realiza la captura de imágenes con el uso del dron y la aplicación pix4D Capture.

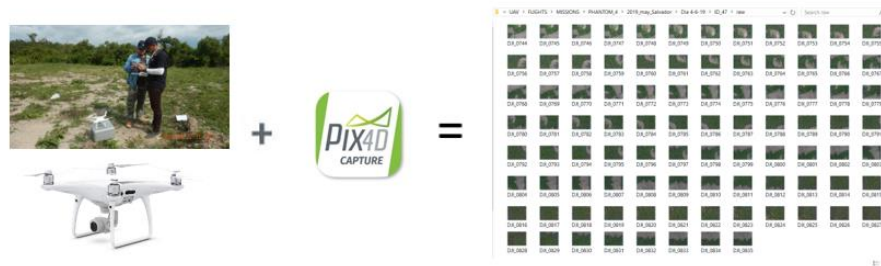


Figura 8. Captura de imágenes dron

6.1.1 Generación de Ortomosaicos usando pix4D

En la figura 9 se muestra el proceso realizado para el tratamiento de las imágenes tomadas utilizando el software Pix4D para generar ortomosaicos en formato geotiff, estos son generados sin la utilización de correcciones radiométricas.

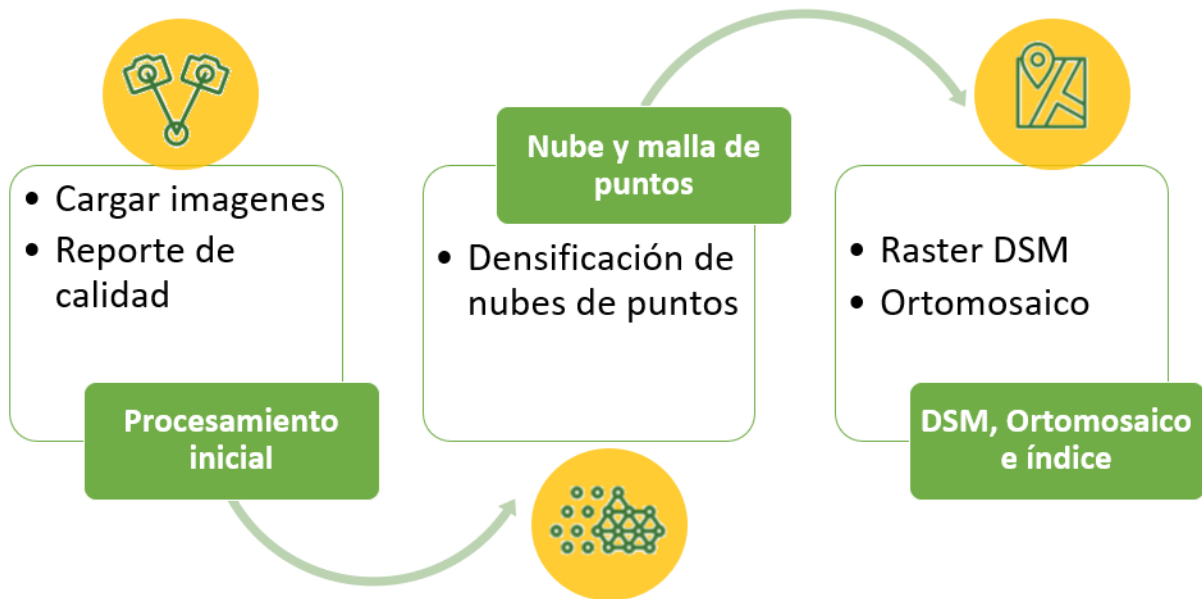


Figura 9. Posprocesamiento de imágenes dron usando PIX4D.

Los mosaicos que se muestran en la figura 10 son la base para realizar el primer muestreo, estos fueron seleccionados teniendo en cuenta los siguientes criterios:

- El mosaico debe tener buena resolución.
- No tener demasiados huecos de información.
- Contener algunas de las clases de interés definidas por los expertos.
- Tener diferentes alturas.
- El área del vuelo debe ser mayor a 6 ha.

Imagen	Lugar	Año	Altura(m)	Area de vuelo	Formato	peso
	Valle del Cauca, Colombia	2017	100	260 m * 260 m	GeoTIFF	102MB
	Ocotepeque, Honduras	2018	150	270 m * 270 m	GeoTIFF	103MB
	Ocotepeque, Honduras	2018	120	260 m * 260 m	GeoTIFF	175MB
	Ahuachapan, El Salvador	2019	110	271 m * 273 m	GeoTIFF	287MB
	Ahuachapan, El Salvador	2019	120	271 m * 273 m	GeoTIFF	203MB
	Ahuachapan, El Salvador	2019	110	271 m * 273 m	GeoTIFF	298 MB

Figura 10. Ortomosaicos seleccionados para la generación de las 10000 muestras.

6.2 Etapa de educación de requisitos

Esta etapa consiste en identificar cuáles son los requisitos que debe satisfacer cierto sistema.

Durante esta etapa se ejecutarán las siguientes actividades:

- Diagrama de contexto del problema.
- Identificación de principales interesados.
- Definición de las clases de coberturas que deben incluirse en el prototipo de clasificación de coberturas.
- Entrevistas y reuniones con los interesados para validar la información.
- Documentación del proceso.

El objetivo principal de esta etapa es plasmar de forma clara y concisa las necesidades de los stakeholders del proyecto en términos de la aplicación web de etiquetado de imágenes tomadas por drones y el prototipo de clasificación de coberturas que se va a desarrollar.

6.2.1 Diagrama de contexto

Cuando inicia el proceso de ingeniería de requisitos el primer paso que se debe realizar es identificar los Stakeholders del sistema. Para esto se realizó la construcción del diagrama de contexto, en donde se puede visualizar de una forma más clara los Stakeholders del sistema, también identificar la información que entra y sale entre el sistema y cada uno de los Stakeholders.

A continuación, se puede visualizar el diagrama de contexto para el sistema:

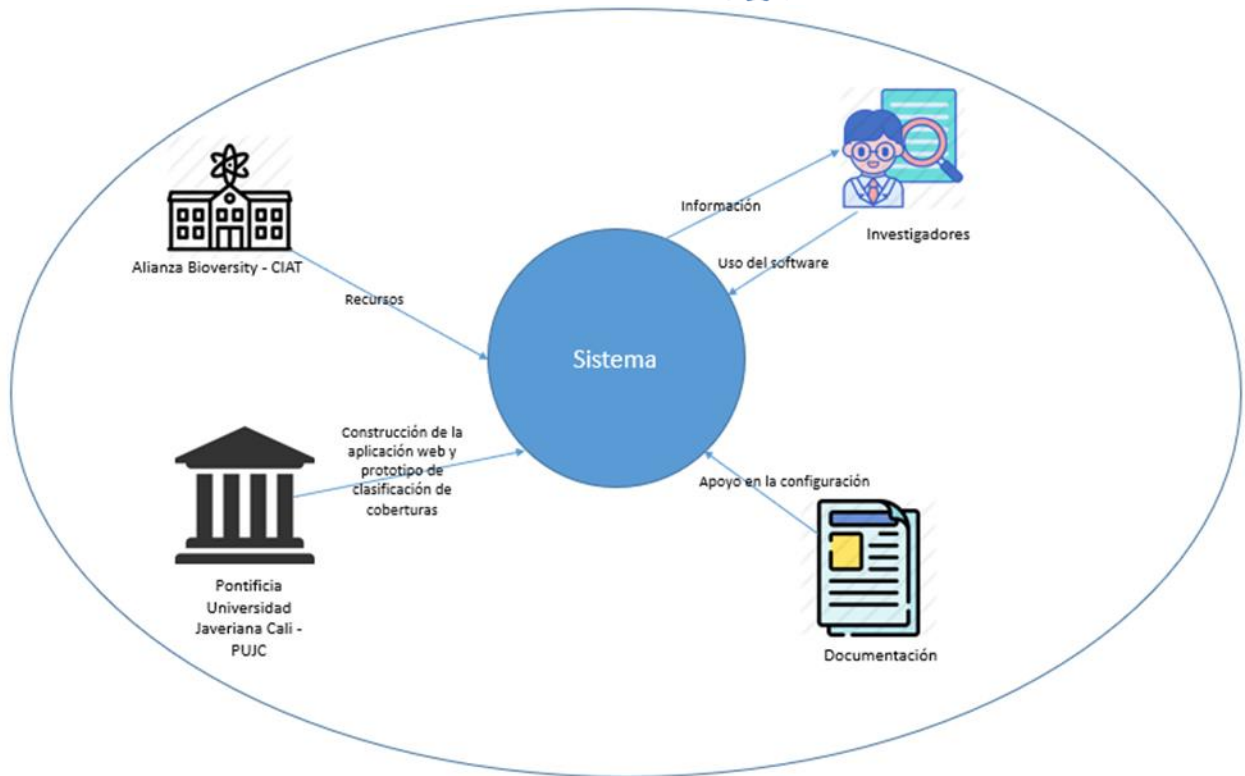


Figura 11. Diagrama de contexto del sistema

En la figura 11 se muestra el diagrama de contexto donde se pueden visualizar los stakeholders principales que son los investigadores que hacen parte del proyecto Terra-i del CIAT. Ellos directamente van aportar diferentes requisitos asociados a sus necesidades de acuerdo con su experiencia en el proceso de clasificación de coberturas.

Los investigadores del proyecto Terra-i juegan un papel muy importante dentro del sistema debido a que son las personas que harán uso del sistema inicialmente, ellos son las personas que tienen el mayor conocimiento en sensores remotos, clasificación de coberturas, reconocimiento de patrones y experiencia en campo. Este conocimiento le permite al experto producir información confiable a través de procesos de toma y comparación de información espacial a través de imágenes de alta resolución. Esta información base nos muestra una idea clara de que los usuarios clave para la información son personas con amplio conocimiento en análisis espacial y principalmente en identificación de coberturas.

Adicionalmente el desarrollo del prototipo se basó principalmente en el artículo ImageNet Classification with Deep Convolutional Neural Networks (Krizhevsky, et al 2012).

6.2.2 Técnicas de obtención de requisitos

En el proceso de obtención de requisitos del sistema se aplicaron diferentes técnicas de educación, en las que se tuvieron en cuenta los stakeholders identificados previamente.

Las técnicas utilizadas se describen a continuación:

- Entrevista

Para conocer el contexto e identificar las necesidades de los investigadores se realizaron entrevistas a los tres principales investigadores dentro del proyecto, en esta primera etapa se abordó de forma personal a cada uno para hacerles un conjunto de preguntas previamente escritas con el fin de ampliar el contexto. A través de llamadas virtuales se consultó a los tres investigadores uno de ellos ubicado en Vietnam y dos en Colombia. La experiencia de cada uno de ellos en sensores remotos y en el tema específico de clasificación de coberturas usando imágenes de dron, revelaron necesidades y deseos que posteriormente fueron usados para definir requisitos del sistema.

La entrevista tuvo una duración de 1 hora. El objetivo principal de la entrevista fue definir si es útil tener una herramienta que ayude en el proceso de clasificación de coberturas en imágenes tomadas por dron y cuantas y que clases incluir en la primera etapa de implementación del prototipo de clasificación. Adicionalmente la entrevista también sirvió para identificar algunas funcionalidades que se incluyeron en la aplicación web.

En el anexo1 se puede observar el modelo de la entrevista que se utilizó.

- Documentos de apoyo a la investigación

El documento que se tomó como referencia principal para identificar requisitos iniciales tanto para la aplicación web de etiquetado y para el prototipo fue el artículo ImageNet Classification with Deep Convolutional Neural Networks. Este artículo sirvió como fuente de información relevante para poder definir la estructura base sobre la cual se desarrolló el prototipo de clasificación de coberturas.

6.2.3 Documentación de requisitos

Los requisitos de software son las condiciones o características que un sistema debe cumplir, definiendo lo que lo que el sistema debe hacer y lo que no. Es por esto que una buena gestión de los requisitos es uno de los aspectos clave para el éxito de los proyectos de software.

La Ingeniería de Requisitos (IR) tiene un papel muy importante en el desarrollo de software dado tiene como objetivo entender y documentar los deseos y necesidades de los implicados, gestionándolos para minimizar el riesgo de entregar un sistema que no cumple las necesidades o los deseos del cliente.

Se elaboraron un conjunto de requisitos del sistema a partir del análisis del contexto y las entrevistas realizadas a los expertos del proyecto Terra-i de la Alianza Bioversity-CIAT.

Los requisitos funcionales y no funcionales fueron escritos en lenguaje natural, la documentación de cada uno de los requisitos está basada en los campos base del estándar internacional IEEE 29148 ("IEEE/ISO/IEC 29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering", 2020)

Atributos de Requisitos Funcionales

La especificación formal de los requisitos funcionales, se definió el siguiente formato, que contiene los atributos relevantes para proveer más detalles acerca del requisito, y facilitar

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
Código único que permite identificar el requisito. Ejemplo: RF-01 RF: Requisito funcional.	Indica si el requisito es Funcional o No Funcional.	Fecha de última actualización del requisito.	Número de versión del requisito.
NOMBRE	Nombre corto del requisito.		
DESCRIPCIÓN	Descripción del requisito utilizando lenguaje natural.		
PRIORIDAD	Prioridad del requisito: Alta, Media, Baja. (Ver: Prioridad de Requisitos).		



ESTADO	Estado del Requisito: Propuesto, Aprobado, Implementado, Verificado, Diferido, Eliminado, Fusionado, Rechazado. (Ver: Estado de Requisitos).
RESTRICCIONES	Restricciones para la ejecución del proceso descrito por el requisito.
REFERENCIAS	Referencias a otros requisitos, documentos o anexos. Utilizar el identificador del requisito, nombre del documento o anexo al que se hace referencia y proveer una descripción de la referencia.
FUENTE	Nombre de la Fuente del requisito. Permite identificar quién o qué originó el requisito.
AUTOR	Nombre del autor del requisito.
RESPONSABLE	Nombre del responsable del requisito.
ESTADO DE NEGOCIACIÓN	Estado de negociación del requisito (Ver: Estado de Negociación de Requisitos).
INFORMACIÓN ADICIONAL	Información, comentarios, notas adicionales relevantes para el requisito.
HISTORIAL DE CAMBIOS	Historial de cambios del requisito.

Prioridad de Requisitos

En la siguiente tabla se describen los diferentes niveles de prioridad de los requisitos:

ESTADO	DESCRIPCIÓN
Alta	El requisito tiene alta prioridad y deberá ser implementado en las primeras iteraciones de desarrollo del producto.
Media	El requisito tiene prioridad media, y podrá ser implementado en iteraciones posteriores.
Baja	El requisito tiene baja prioridad, y podrá ser implementado en las últimas iteraciones de desarrollo del producto.

Estado de Requisitos

En la siguiente tabla se describen los diferentes estados en los que se puede encontrar un requisito:

ESTADO	DESCRIPCIÓN
Propuesto	Alguien propuso el requisito.
Aprobado	El requisito fue aprobado y asignado a una línea base.
Implementado	El código fue diseñado, escrito, y probado con pruebas unitarias.



Verificado	El requisito pasó las pruebas después de la integración con el producto.
Diferido	El requisito será implementado en futuras versiones.
Eliminado	El requisito no será implementado.
Fusionado	El requisito fue fusionado con otro.
Rechazado	La idea del requisito no fue aprobada.

Estado de Negociación de Requisitos

En esta lista se describen los diferentes estados de negociación en los que se puede encontrar un requisito:

ESTADO	DESCRIPCIÓN
No Negociado	El requisito no ha requerido ser negociado.
Conflicto	El requisito presenta conflictos y requiere ser negociado con los stakeholders.
Negociado	El requisito fue negociado con los stakeholders.

Requisitos Funcionales

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-01	Funcional	23/04/2020	1
NOMBRE	Autenticación de usuarios.		
DESCRIPCIÓN	El sistema deberá manejar autenticación para el ingreso de usuarios, esta autenticación se realizará con los campos correo electrónico y contraseña.		
PRIORIDAD	Alta		
ESTADO	Verificado		
RESTRICCIONES			
REFERENCIAS			
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 - Requisito creado. 04/05/2020 - Se realizó la actualización del requisito		



CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-02	Funcional	23/04/2020	1
NOMBRE	Módulos del sistema		
DESCRIPCIÓN	El sistema deberá contener los siguientes módulos: Usuarios, Roles, Etiquetas y Experimentos		
PRIORIDAD	Alta		
ESTADO	Verificado		
RESTRICCIONES			
REFERENCIAS			
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 - Requisito creado.		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-03	Funcional	23/04/2020	1
NOMBRE	Administración de usuarios		
DESCRIPCIÓN	El módulo de usuarios del sistema deberá permitirle al usuario administrador la creación, actualización, consulta y eliminación de usuarios		
PRIORIDAD	Alta		
ESTADO	Verificado		
RESTRICCIONES	El usuario Administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-02		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 - Requisito creado.		



CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-04	Funcional	23/04/2020	1
NOMBRE	Administración de roles		
DESCRIPCIÓN	El módulo de Roles del sistema deberá permitirle al usuario administrador la creación, actualización y consulta de roles		
PRIORIDAD	Alta		
ESTADO	Verificado		
RESTRICCIONES	El usuario administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-02		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado. 04/05/2020 - Se realizó la actualización del requisito		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-05	Funcional	29/03/2017	1
NOMBRE	Administración de etiquetas		
DESCRIPCIÓN	El módulo de etiquetas del sistema deberá permitirle al usuario administrador la creación, actualización y consulta de etiquetas		
PRIORIDAD	Alta		
ESTADO	Verificado		
RESTRICCIONES	El usuario administrador debe estar autenticado en el sistema.		
REFERENCIAS			
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado 04/05/2020 - Se realizó la actualización del requisito		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-06	Funcional	23/04/2020	1
NOMBRE	Administración de experimentos		
DESCRIPCIÓN	El módulo de experimentos del sistema deberá permitirle al usuario administrador la creación, actualización, consulta y eliminación de experimentos		
PRIORIDAD	Alta		
ESTADO	Verificado		
RESTRICCIONES	El usuario administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-02		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-07	Funcional	23/04/2020	1
NOMBRE	Manejo de roles		
DESCRIPCIÓN	El sistema deberá manejar inicialmente los siguientes roles: Administrador y usuario		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	El usuario administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-04		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado 04/05/2020 - Se realizó la actualización del requisito		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-08	Funcional	23/04/2020	1
NOMBRE	Definición de alto y ancho de imágenes de muestra		
DESCRIPCIÓN	El Sistema deberá permitirle al usuario administrador en el momento de la creación de un experimento definir el alto y ancho de las imágenes de muestra a generar		
PRIORIDAD	alta		
ESTADO	Verificado		
RESTRICCIONES	El usuario administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-09	Funcional	23/04/2020	1
NOMBRE	Definición de tamaño de muestra y conjunto de entrenamiento		
DESCRIPCIÓN	El Sistema deberá permitirle al usuario administrador en el momento de la creación de un experimento definir el tamaño de la muestra y el tamaño de conjunto de entrenamiento		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	El usuario administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-10	Funcional	23/04/2020	1
NOMBRE	Selección de etiquetas		
DESCRIPCIÓN	El Sistema deberá permitirle al usuario administrador en el momento de la creación de un experimento seleccionar las etiquetas que quiera usar para el experimento		
PRIORIDAD	alta		
ESTADO	Verificado		
RESTRICCIONES	El usuario administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-11	Funcional	23/04/2020	1
NOMBRE	Visualización de mapas base		
DESCRIPCIÓN	El Sistema deberá permitirle al usuario administrador en el momento de la creación de un experimento visualizar los mapas base (imágenes geotiff) disponibles		
PRIORIDAD	alta		
ESTADO	Verificado		
RESTRICCIONES	El usuario administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-12	Funcional	23/04/2020	1
NOMBRE	Selección de mapas base		
DESCRIPCIÓN	El Sistema deberá permitirle al usuario administrador en el momento de la creación de un experimento seleccionar uno o más mapas base (imágenes geotiff) disponibles		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	El usuario administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-13	Funcional	23/04/2020	1
NOMBRE	Selección de etiquetas por muestra		
DESCRIPCIÓN	El Sistema deberá permitirle al usuario seleccionar una o más etiquetas al momento de etiquetar una muestra		
PRIORIDAD	alta		
ESTADO	Verificado		
RESTRICCIONES	El usuario con rol administrador o rol usuario debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-14	Funcional	23/04/2020	1
NOMBRE	Etiquetado multiusuario		
DESCRIPCIÓN	El Sistema deberá permitirle a uno o varios usuarios realizar el etiquetado colaborativo sobre un mismo experimento		
PRIORIDAD	baja		
ESTADO	Verificado		
RESTRICCIONES	El usuario con rol administrador o rol usuario debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-15	Funcional	23/04/2020	1
NOMBRE	Conteo muestras etiquetadas		
DESCRIPCIÓN	El Sistema deberá permitirles a los usuarios ver la cantidad de muestras que han sido etiquetadas en un experimento		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	El usuario con rol administrador o rol usuario debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		



CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-16	Funcional	23/04/2020	1
NOMBRE	Finalización del proceso de etiquetado		
DESCRIPCIÓN	El Sistema deberá resaltar en color verde los experimentos que hayan sido etiquetados en su totalidad		
PRIORIDAD	baja		
ESTADO	Verificado		
RESTRICCIONES	El usuario con rol administrador o rol usuario debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-17	Funcional	23/04/2020	1
NOMBRE	Descarga de archivo etiquetado		
DESCRIPCIÓN	El Sistema deberá permitirles a los usuarios descargar en formato json el archivo de un experimento		
PRIORIDAD	alta		
ESTADO	Verificado		
RESTRICCIONES	El usuario con rol administrador o rol usuario debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		



CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-18	Funcional	23/04/2020	1
NOMBRE	Mensaje de creación de un experimento		
DESCRIPCIÓN	El Sistema deberá mostrarle al usuario que crea el experimento un mensaje o etiqueta que diga que el experimento ha sido creado		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	El usuario con rol administrador o rol usuario debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-19	Funcional	23/04/2020	1
NOMBRE	Almacenamiento de información de etiquetado		
DESCRIPCIÓN	El Sistema deberá guardar la información de cada muestra etiquetada como: ruta de la imagen base de donde se obtuvo la muestra, etiqueta o etiquetas y la ubicación de la muestra etiquetada (xOffset, y Offset)		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	La información debe ser almacenada en una base de datos		
REFERENCIAS	RF-06		
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	23/04/2020 – Requisito creado		



CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-20	Funcional	27/04/2020	1
NOMBRE	Configuración entrenamiento del prototipo de clasificación de coberturas		
DESCRIPCIÓN	El Prototipo deberá permitir configurar los parámetros de entrada para el proceso de entrenamiento como: archivo de etiquetado, ruta de salida del modelo, tamaño de lote, cores, épocas, save max y save step		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES			
REFERENCIAS			
FUENTE	Entrevista y paper		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL	El archivo de etiquetado usado por el prototipo es el que se descarga de la aplicación web de etiquetado una vez está finalizado		
HISTORIAL DE CAMBIOS	27/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-21	Funcional	27/04/2020	1
NOMBRE	Almacenamiento imágenes de muestra		
DESCRIPCIÓN	El Prototipo deberá permitir guardar las imágenes etiquetadas clasificadas en carpetas por cada etiqueta usada y las rutas de estas imágenes en una base de datos		
PRIORIDAD	alta		
ESTADO	Verificado		
RESTRICCIONES			
REFERENCIAS			
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	27/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-22	Funcional	27/04/2020	1
NOMBRE	Generación modelo de clasificación de coberturas		
DESCRIPCIÓN	El Prototipo deberá generar el modelo de clasificación de coberturas resultado del proceso de entrenamiento, almacenándolo en una carpeta de modelos		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES			
REFERENCIAS			
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	27/04/2020 – Requisito creado		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-23	Funcional	27/04/2020	1
NOMBRE	Clasificación de coberturas		
DESCRIPCIÓN	El Prototipo deberá permitir recibir una imagen en formato geotiff guardando en una carpeta de salida los resultados del proceso de clasificación		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES			
REFERENCIAS			
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	27/04/2020 – Requisito creado		



CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-24	Funcional	27/04/2020	1
NOMBRE	Configuración de permisos por rol.		
DESCRIPCIÓN	El sistema deberá permitirle al usuario administrador configurar si el permiso es total o no.		
PRIORIDAD	baja		
ESTADO	Verificado		
RESTRICCIONES	El administrador debe estar autenticado en el sistema.		
REFERENCIAS	RF-04		
FUENTE	Jhon Jairo Tello Dagua		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	27/04/2020 – Requisito creado.		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-25	Funcional	27/04/2020	1
NOMBRE	Validación de permisos de permisos de usuario.		
DESCRIPCIÓN	El sistema deberá validar los permisos de las acciones habilitadas para el usuario, a partir de la configuración de seguridad de su rol, antes de permitirle acceder una funcionalidad o realizar una acción.		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	El usuario debe estar autenticado en el sistema.		
REFERENCIAS			
FUENTE	Jhon Jairo Tello Dagua		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL	RF-04, RF-03		
HISTORIAL DE CAMBIOS	27/04/2020 – Requisito creado.		



CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-26	Funcional	27/04/2020	1
NOMBRE	Generación aleatoria de imágenes.		
DESCRIPCIÓN	El sistema deberá permitir la generación aleatoria de imágenes de muestra para ser etiquetadas.		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	El usuario debe estar autenticado en el sistema.		
REFERENCIAS			
FUENTE	Jhon Jairo Tello Dagua		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL	RF-09		
HISTORIAL DE CAMBIOS	27/04/2020 – Requisito creado.		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-27	Funcional	27/04/2020	1
NOMBRE	Idioma.		
DESCRIPCIÓN	El sistema deberá tener la mayoría de sus textos en idioma inglés		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES			
REFERENCIAS			
FUENTE	Entrevista		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	27/04/2020 – Requisito creado.		



CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-28	Funcional	18/05/2020	1
NOMBRE	Cargar experimento		
DESCRIPCIÓN	El sistema deberá permitir a un usuario registrado cargar un experimento para ver las muestras etiquetadas		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	El usuario debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Proceso de negociación		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	18/05/2020 – Requisito creado.		

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-29	Funcional	18/05/2020	1
NOMBRE	Mostrar experimento		
DESCRIPCIÓN	El sistema deberá permitir a un usuario registrado mostrar un experimento para ver los parámetros de configuración del experimento		
PRIORIDAD	media		
ESTADO	Verificado		
RESTRICCIONES	El usuario debe estar autenticado en el sistema.		
REFERENCIAS	RF-06		
FUENTE	Proceso de negociación		
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	18/05/2020 – Requisito creado.		

Plantilla de Requisitos Funcionales:

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RF-00	Funcional/ No funcional	dd/mm/aaaa	1
NOMBRE			
DESCRIPCIÓN			
PRIORIDAD	Alta/Media/Baja		
ESTADO	Propuesto, Aprobado, Implementado, Verificado, Diferido, Eliminado, Fusionado, Rechazado		
RESTRICCIONES			
REFERENCIAS			
FUENTE			
AUTOR	Jhon Jairo Tello Dagua		
RESPONSABLE			
ESTADO DE NEGOCIACIÓN	Negociado /No Negociado		
INFORMACIÓN ADICIONAL			
HISTORIAL DE CAMBIOS	dd/mm/aaaa – Requisito creado.		

Requisitos No Funcionales

USA - Usabilidad

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RNF-USA-001	No Funcional	27/04/2020	1
DESCRIPCIÓN:	El sistema deberá ser fácil de usar para los usuarios investigadores.		
FUENTE DEL ESTÍMULO:	Usuario investigador.		
ESTÍMULO:	Uso de la aplicación para el etiquetado de un conjunto de imágenes.		
AMBIENTE:	Operación normal.		
ARTEFACTO:	Aplicación.		
RESPUESTA:	<ul style="list-style-type: none"> - La aplicación tiene un diseño minimalista con el fin de que el usuario se puede ubicar fácilmente. - La aplicación utiliza mensajes y colores para identificar que ciertos procesos se están realizando o ya están terminados. 		

	<ul style="list-style-type: none"> - La aplicación es consistente y utiliza estándares visuales. - La aplicación previene errores validando la información ingresada por el usuario.
MEDICIÓN DE RESPUESTA:	Después de 30 minutos utilizando la aplicación junto con el investigador, el usuario puede realizar el proceso de etiquetado de forma fluida
PRIORIDAD	Alta
ESTADO	Verificado
RESPONSABLE	Jhon Jairo Tello Dagua
ESTADO DE NEGOCIACIÓN	Negociado
INFORMACIÓN ADICIONAL	Se tiene como referencia los principios de usabilidad de acuerdo a las Heurísticas de Jakob Nielsen.

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RNF-USA-002	No Funcional	29/03/2017	1
DESCRIPCIÓN:	El sistema deberá permitirle al usuario investigador crear fácilmente un experimento.		
FUENTE DEL ESTÍMULO:	Usuario investigador		
ESTÍMULO:	Uso del sistema para la creación de un experimento.		
AMBIENTE:	Operación normal.		
ARTEFACTO:	Sistema.		
RESPUESTA:	<ul style="list-style-type: none"> - La aplicación provee una retroalimentación visual de las interacciones del usuario. - El sistema muestra los campos resaltados de manera que el usuario sabe en qué parte del formulario se encuentra. - El sistema muestra mensajes de error en caso de que quede algún campo vacío así se puede identificar fácilmente que información falta. - La aplicación es consistente y utiliza estándares visuales. - La aplicación previene errores validando la información ingresada por el usuario. 		
MEDICIÓN DE RESPUESTA:	Después de 30 minutos utilizando el sistema, el usuario habrá aprendido como utilizarlo.		
PRIORIDAD	Alta		
ESTADO	Verificado		
RESPONSABLE	Jhon Jairo Tello Dagua		



ESTADO DE NEGOCIACIÓN	Negociado
INFORMACIÓN ADICIONAL	Se tiene como referencia los principios de usabilidad de acuerdo a las Heurísticas de Jakob Nielsen.

SEG - Seguridad

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RNF-SEG-001	No Funcional	29/03/2017	1
DESCRIPCIÓN:	El sistema debería garantizar la creación de usuarios, roles y etiquetas solo sea realizado por un usuario administrador		
FUENTE DEL ESTÍMULO:	Usuario administrador		
ESTÍMULO:	Usuario administrador hace ingreso al sistema.		
AMBIENTE:	Operación normal.		
ARTEFACTO:	Sistema.		
RESPUESTA:	El sistema valida que el usuario se encuentra autenticado. El sistema valida que el usuario tenga un rol administrador. El sistema rechaza la solicitud si el usuario no está autorizado.		
MEDICIÓN DE RESPUESTA:	Siempre que se trate de acceder a los módulos de usuarios, roles y etiquetas sin un usuario valido, la solicitud es rechazada.		
PRIORIDAD	Alta		
ESTADO	Verificado		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RNF-SEG-002	No Funcional	29/03/2017	1
DESCRIPCIÓN:	Toda la Información almacenada en el sistema debe estar protegida contra accesos no autorizados.		
FUENTE DEL ESTÍMULO:	Usuario no autenticado		
ESTÍMULO:	Usuario no autenticado trata de hacer ingreso al sistema.		
AMBIENTE:	Operación normal.		
ARTEFACTO:	Sistema.		
RESPUESTA:	El sistema valida que el usuario se encuentra autenticado.		

	El sistema rechaza la solicitud si el usuario no está autorizado.
MEDICIÓN DE RESPUESTA:	Siempre que un usuario no autenticado trate de acceder a cualquiera de los módulos del sistema, la solicitud de acceso es rechazada.
PRIORIDAD	Alta
ESTADO	Verificado
RESPONSABLE	Jhon Jairo Tello Dagua
ESTADO DE NEGOCIACIÓN	Negociado
INFORMACIÓN ADICIONAL	

POR - Portabilidad

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RNF-POR-001	No Funcional	29/03/2017	1
DESCRIPCIÓN:	El sistema de etiquetado deberá ser accesible a través de diferentes tipos de dispositivos (Tablet, PC, Teléfono Inteligente).		
FUENTE DEL ESTÍMULO:	Usuario.		
ESTÍMULO:	Ejecución de Sistema en dispositivo.		
AMBIENTE:	Operación normal.		
ARTEFACTO:	Sistema.		
RESPUESTA:	Si el dispositivo es soportado, la aplicación se ejecuta.		
MEDICIÓN DE RESPUESTA:	El sistema se ejecuta y funciona adecuadamente en el dispositivo.		
PRIORIDAD	Alta		
ESTADO	Verificado		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RNF-POR-002	No Funcional	29/03/2017	1
DESCRIPCIÓN:	El sistema deberá adaptar el contenido de las actividades a diferentes tamaños de pantalla para ser visualizados en varios tipos de dispositivos como teléfonos, tablets y computadores portátiles y de escritorio.		
FUENTE DEL ESTÍMULO:	Usuario.		
ESTÍMULO:	Ejecución de aplicación en dispositivo.		
AMBIENTE:	Operación normal.		
ARTEFACTO:	Aplicación.		
RESPUESTA:	El sistema ajusta el tamaño y distribución de los elementos en pantalla de acuerdo al tamaño del dispositivo.		
MEDICIÓN DE RESPUESTA:	La aplicación y su contenido se visualiza adecuadamente en el dispositivo.		
PRIORIDAD	baja		
ESTADO	Verificado		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			

DIS - Disponibilidad

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RNF-DIS-001	No Funcional	29/03/2017	1
DESCRIPCIÓN:	El sistema deberá funcionar correctamente en los principales navegadores web		
FUENTE DEL ESTÍMULO:	Usuario.		
ESTÍMULO:	Ejecución de aplicación en navegadores como Chrome, Firefox, Edge.		
AMBIENTE:	Operación con conexión a internet.		
ARTEFACTO:	Sistema.		
RESPUESTA:	El sistema abre la página de inicio.		
MEDICIÓN DE RESPUESTA:	El sistema permite navegar sin problemas a través de las páginas disponibles.		
PRIORIDAD	media		
ESTADO	Aprobado		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	Negociado		
INFORMACIÓN ADICIONAL			

DES - Desempeño

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
RNF-DES-001	No Funcional	29/03/2017	1
DESCRIPCIÓN:	El tiempo de carga de las páginas en cada módulo no deberá ser mayor a 5 segundos.		
FUENTE DEL ESTÍMULO:	Usuario investigador		
ESTÍMULO:	Abrir un módulo disponible ej. experimentos		
AMBIENTE:	Operación normal.		
ARTEFACTO:	Sistema.		
RESPUESTA:	Se despliega el contenido del módulo seleccionado		
MEDICIÓN DE RESPUESTA:	Al ejecutar una actividad el tiempo de carga es menor o igual a 5 segundos.		
PRIORIDAD	media		
ESTADO	Verificado		
RESPONSABLE	Jhon Jairo Tello Dagua		
ESTADO DE NEGOCIACIÓN	No Negociado		
INFORMACIÓN ADICIONAL			

Plantilla de Requisitos No Funcionales:

CÓDIGO	CLASIFICACIÓN	FECHA	VERSIÓN
Código único que permite identificar el requisito. Ejemplo: RNF-DES-001 RNF: Requisito no funcional. DES: Grupo de funcionalidad, Ej. Seguridad (SEG), desempeño (DES), portabilidad (POR) 001: Consecutivo del requisito dentro del grupo de funcionalidad	No Funcional	dd/mm/aaaa	1

DESCRIPCIÓN:	Descripción del requisito usando lenguaje natural
FUENTE DEL ESTÍMULO:	Actor que genera el estímulo
ESTÍMULO:	Condición o acción que necesita ser considerada
AMBIENTE:	Condiciones en las que se encuentra el sistema al presentarse el estímulo
ARTEFACTO:	Componente o destinatario que recibe el estímulo
RESPUESTA:	Actividad que se realiza una vez detectado el estímulo
MEDICIÓN DE RESPUESTA:	Métrica definida para el tipo de estímulo
PRIORIDAD	Alta/Media/Baja
ESTADO	Propuesto, Aprobado, Implementado, Verificado, Diferido, Eliminado, Fusionado, Rechazado
RESPONSABLE	
ESTADO DE NEGOCIACIÓN	Negociado/No Negociado
INFORMACIÓN ADICIONAL	

6.2.4 Priorización de requisitos

Durante el proceso de gestión de requisitos de un desarrollo de un producto o servicio, además de realizar una correcta obtención y escritura de los requisitos, adquiere una gran importancia realizar una priorización de los requisitos. La realización de este proceso es muy importante porque nos ayuda en la planificación de la construcción del software y conseguir darle un valor más alto al proyecto a un menor costo.

La técnica de priorización de requisitos escogida para este proyecto fue Escala de tres niveles: Prioridad Alta, Media, Baja. La priorización fue realizada por los stakeholders Louis Reymondin, Paula Paz y Jorge Perez

En el anexo 2 se encuentra el enlace al formato con el resultado del proceso de priorización realizado.

6.2.5 Negociación de requisitos

El día 04 y 18 de mayo de 2020, a través de una llamada por Skype se realizó un proceso de socialización y negociación con los investigadores del proyecto Terra-i CIAT, acerca de algunos de los requisitos funcionales y no funcionales, con el objetivo de llegar a acuerdos sobre el planteamiento de cada uno de estos.

El proceso se realizó de la siguiente forma:

1. Cada investigador realizó una revisión rápida de requisitos.
2. Se realizó una selección por rondas de 2 requisitos por cada grupo para realizar el proceso de negociación. En total se realizaron 20 rondas.
3. Cada investigador presentó sus argumentos sobre aceptar o ajustar requisitos, se realizaron algunas aclaraciones a inquietudes y comentarios sobre los requisitos presentados.
4. La moderación fue realizada por Jhon Jairo Tello.
5. Se realizaron los acuerdos respectivos entre todos los participantes, realizando los cambios acordados a los requisitos negociados.
6. Se realizó la documentación del proceso de negociación.

En la realización del ejercicio de negociación de requisitos no hubo problemas entre stakeholders.

A continuación, relacionamos los requisitos que durante el proceso de negociación tuvieron algún comentario y actualización.

1. Autenticación (Ingreso del investigador al sistema)

En la negociación de este requisito todos los investigadores estuvieron de acuerdo.

Se realizó la actualización del requisito mejorando la redacción del requisito

Después de realizar este cambio se actualizó el requisito: **RF-01**

2. Manejo de roles

En la negociación de este requisito se decidió en conjunto definir que los primeros roles a crear sean Administrador y usuario.

Después de realizar este cambio se actualizó el requisito: **RF-07**

3. Administración de roles

En la negociación de este requerimiento se generó una discusión sobre si incluir o no la opción de borrar para los roles. Para nuestro caso se llegó al acuerdo de que no se tenga esa opción por ahora, si llega a ser necesario el borrado de un rol este se va a realizar a nivel de administración de la base de datos.

Después de realizar este cambio se actualizó el requisito: **RF-04**.

4. Administración de etiquetas

En la negociación de este requerimiento se generó una discusión sobre si incluir o no la opción de borrar etiquetas creadas. Para nuestro caso se llegó al acuerdo de que no se tenga esa opción por ahora, si llega a ser necesario el borrado de un rol este se va a realizar a nivel de administración de la base de datos.

Después de realizar este cambio se actualizó el requisito: **RF-05**.

5. Cargar experimento

Este requisito salió en medio de una de las discusiones del proceso de negociación y no lo teníamos contemplado hasta ese momento.

Después de realizar este proceso se creó el requisito: **RF-27**.

6. Mostrar experimento

Este requisito salió en medio de una de las discusiones del proceso de negociación y no lo teníamos contemplado hasta ese momento.

Después de realizar este proceso se creó el requisito: **RF-28**.

6.3 Arquitectura de software y diseño de la solución

Después de terminar el proceso de ingeniería de requisitos, se procede a crear una propuesta que aborde las preguntas de la investigación, en esta etapa se realizó el diseño de una arquitectura de software que desde el punto de vista conceptual pueda dar solución a la creación de la aplicación web para el etiquetado de imágenes tomadas por drones. La arquitectura sobre la que implementó la aplicación web es una arquitectura en capas (Figura12).

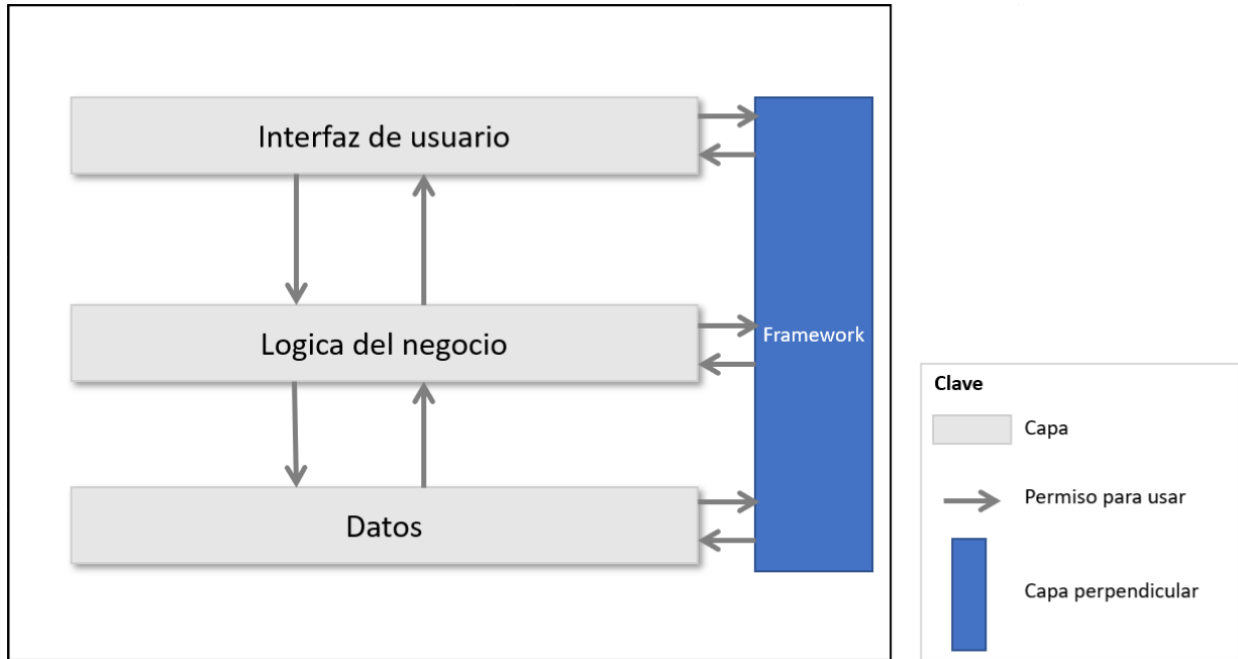


Figura 12. Arquitectura de alto nivel

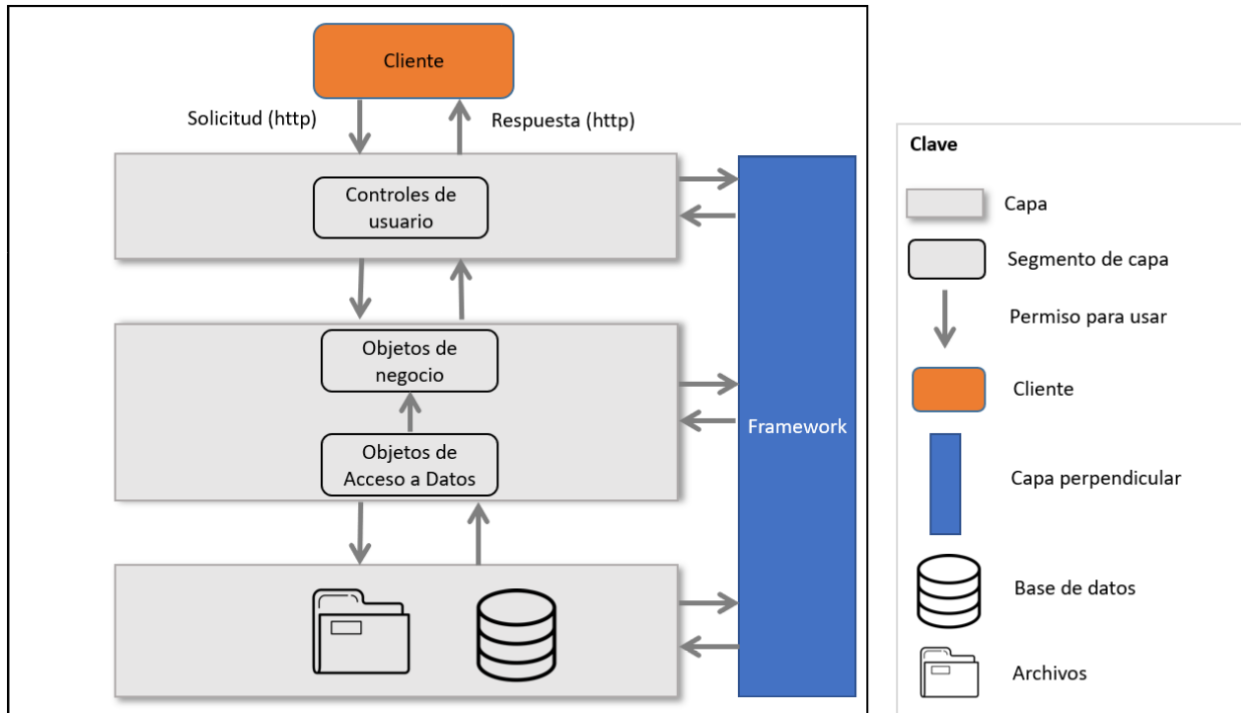


Figura 13. Vista de Arquitectura y segmentos de capa

La arquitectura por capas que se muestra en la figura 13, muestra que el sistema que se comunica con sus clientes a través de internet. Las personas interactúan con la plataforma a través de un front-end en la capa de interfaz de usuario, mientras que en la capa de negocio se alojarán los objetos relacionados con el negocio y el acceso a datos y por último la capa de datos donde se alojarán los registros de imágenes etiquetadas, imágenes y archivos que se irán generando a través del uso del sistema. Para que los clientes interactúen con el sistema deberán tener credenciales de acceso. La aplicación web cuenta con un módulo para la gestión de usuarios, roles, etiquetas, experimentos y finalmente para la generación de las imágenes de muestra y el almacenamiento en una base de datos de los registros. Los objetos de acceso a datos se encargan de gestionar la conexión a la base de datos y hacer uso del CRUD (create, read, update, delete) de los datos. Las peticiones son tratadas por el controlador, el cual es el encargado de manejar el flujo de la aplicación, en este se hacen usos de los modelos, los cuales son necesarios para acceder a la base de datos. El controlador devuelve la vista la cual se muestra en el navegador.

Para agilizar el proceso de desarrollo una vez ya se tiene la arquitectura de la solución se decidió utilizar un framework de desarrollo web que sea compatible con la arquitectura, eficiente y con el que ya se tiene un poco de experiencia, este framework es Laravel.

Laravel es un framework de código abierto que utiliza el modelo MVC (Figura 14). Fue desarrollado por Taylor Otwell en conjunto con un gran grupo de contribuidores cuya finalidad es “desarrollar aplicaciones y servicios web de forma simple y elegante” (Cíceri, 2019).

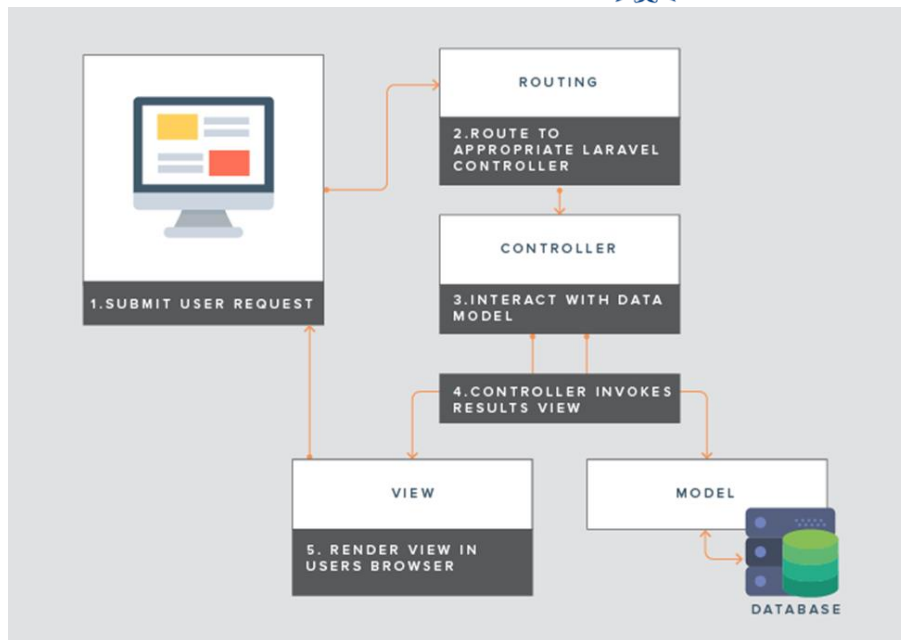


Figura 14. Patrón de diseño MVC de Laravel. Fuente: (¿” Why Use Laravel as a Development Platform?”, 2020)

Modelo

El modelo se encarga de gestionar los datos de la aplicación, responde a la solicitud desde la vista y también responde a las instrucciones del controlador para actualizarse.

Vista

El componente de vista se utiliza para toda la lógica de la interfaz de usuario de la aplicación. Los componentes de la UI como cuadros de texto, desplegables, botones, entre otros con los que interactúa el usuario final.

Controlador

El controlador es responsable de controlar la lógica de la aplicación y actúa como coordinador entre la Vista y el Modelo. El controlador recibe información de los usuarios a través de la vista, luego procesa los datos del usuario con la ayuda del modelo y devuelve los resultados a la vista.

Enrutamiento

El enrutamiento define cómo la aplicación procesará y responderá a la solicitud HTTP entrante. Puede utilizar URL que describan correctamente la acción del controlador a la que debe enrutarse la solicitud. El enrutamiento permite el uso de URL que describen las acciones del usuario y que los usuarios pueden comprender más fácilmente.

Datos

El último componente de la arquitectura es el que hace referencia a los datos. Esta parte es donde se guardan los datos de forma estructurada y es la encargada de acceder a ellos. Está formada por un gestor de base de datos que realiza el almacenamiento de datos, reciben solicitudes de almacenamiento y recuperación de información desde la capa de negocio.

Para la definición de esta capa se utilizó un modelo de base de datos relacional, el cual es uno de los modelos más conocidos y utilizados. En este modelo la información se almacena en tablas, donde las columnas hacen referencia a atributos de la tabla y las filas son los registros. Las tablas se relacionan entre sí a través de atributos llamados llaves.

6.4 Modelo de datos

El Sistema de Gestión de Bases de Datos (SGBD) que se utilizó fue PostgreSQL el cual es un sistema de gestión de base de datos relacional orientado a objetos y de código abierto, el principal motivo de usar PostgreSQL fue que este cuenta con una extensión que añade soporte de objetos geográficos y permite realizar análisis mediante consultas SQL espaciales o mediante conexión a aplicaciones SIG (Sistemas de Información Geográfica).

En un modelo de base de datos definimos cómo se almacenan los datos y cuál es la relación entre ellos. Los modelos de datos se representan mediante un diagrama de base de datos (Figura 15).

A continuación, se presenta el diagrama entidad-relación de la aplicación web:

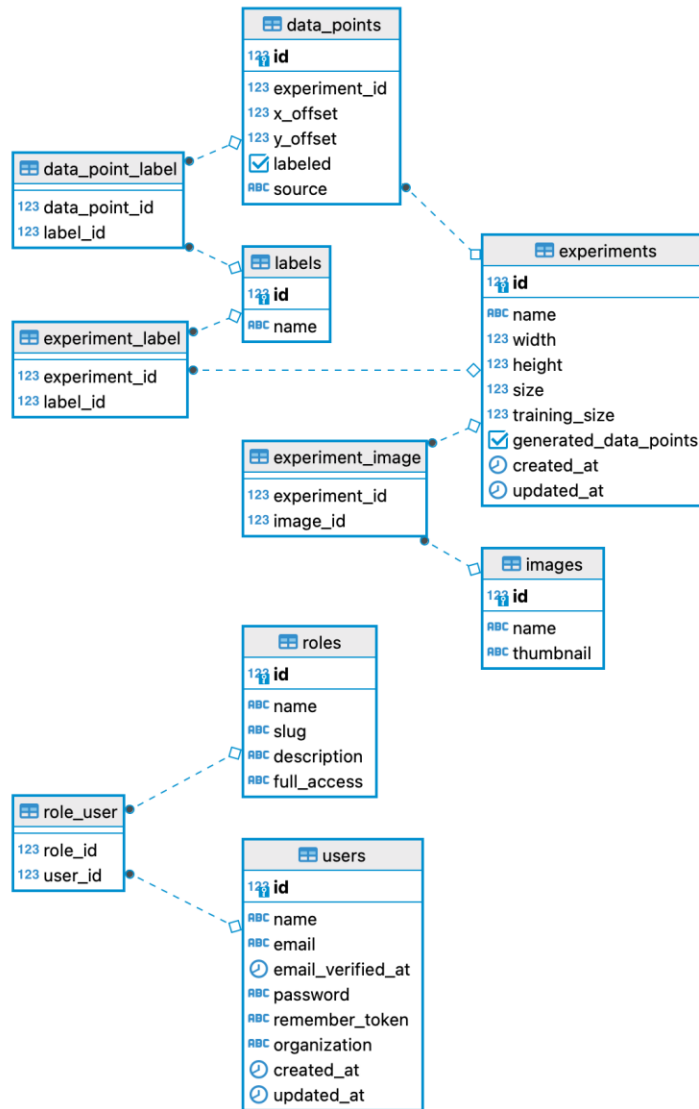


Figura 15. Diagrama entidad-relación aplicación web de etiquetado

6.5 Creación de las submuestras

La creación de las submuestras que se van a etiquetar a través de la aplicación web de etiquetado se genera a través de un componente desarrollado en lenguaje java el cual se encuentra integrado al prototipo de clasificación de coberturas. Las submuestras se obtienen de las imágenes en formato geotiff descritas en la sección 6.1.1 Estas imágenes obtenidas son seleccionadas aleatoriamente y tienen un tamaño de 200 pixeles x 200 pixeles equivalente a un área de 6m x 6m. El

subconjunto de imágenes una vez etiquetadas será guardado en carpetas y las rutas en una base de datos. En el siguiente paso, las imágenes previamente muestreadas van a ser clasificadas por expertos y serán el insumo de entrada para entrenar los modelos de clasificación. A continuación, se muestra el diagrama de clases para el módulo de generación de submuestras (Figura 16).

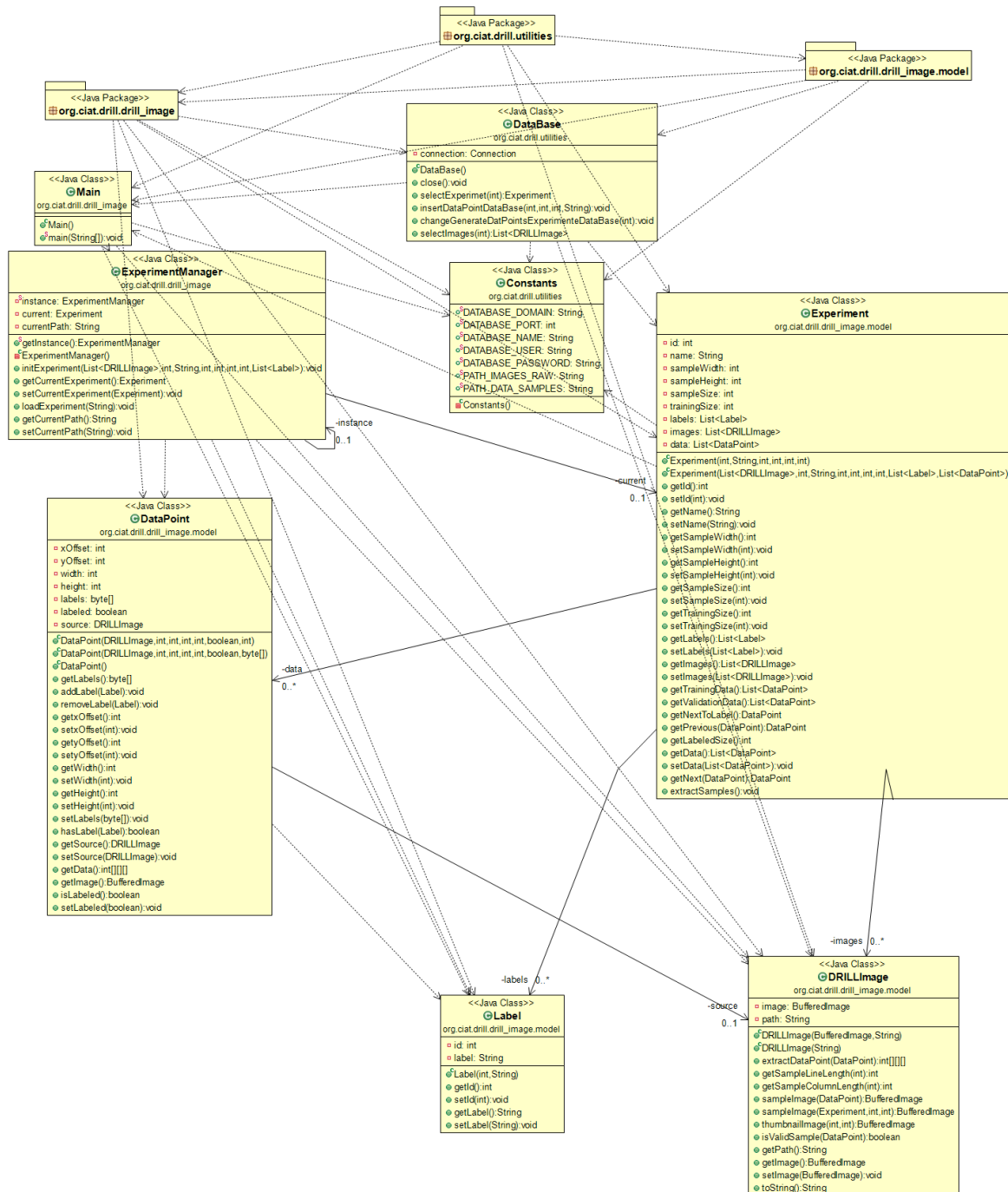


Figura 16. Diagrama de clases modulo para la generación de submuestras

6.6 Codificación de la aplicación web para el etiquetado de imágenes de alta resolución
Para el desarrollo de la aplicación web se utilizó el lenguaje de programación PHP, esto debido a que el framework de desarrollo que se eligió fue Laravel y está basado en este lenguaje. El prototipo de clasificación de coberturas fue construido bajo el lenguaje Java.

Las tecnologías utilizadas en el desarrollo de la herramienta fueron:

- JavaScript (Bootstrap)
- CSS
- HTML
- Framework Laravel 8
- Composer 2.0
- PosrgreSQL
- Java

Interfaz gráfica de usuario aplicación web

La interfaz gráfica de usuario (GUI por sus siglas en inglés) es el medio por el que los usuarios acceden a las funcionalidades de la aplicación web de etiquetado de imágenes de alta resolución a continuación se presentan las imágenes de los diferentes módulos que hacen parte de la aplicación web que tiene por nombre Image GEO y se encuentra alojada temporalmente en una máquina virtual Linux de 8GB de memoria RAM y 4vCPUs con la siguiente url <https://imagegeo.proyectosbeta.net/> .

Image GEO

Figura 17. Pantalla inicial de la aplicación web de etiquetado de imágenes

Login

E-Mail Address

Password

Login

Figura 18. Login de usuario

Dashboard

Figura 19. Vista de los módulos disponibles desde un usuario administrador

Users

New

Name	Email	Role			
Paula Paz	papaz@cgiar.org	User	Show	Edit	Delete
admin	admin@admin.com	Admin	Show	Edit	Delete

Figura 20. Modulo usuarios

New user

Name

Email

Organization

Password

Role ▾

Menu Save

Figura 21. Vista creación de un usuario nuevo

Roles

[New](#)

Name	Slug	Total access		
Admin	admin	Yes	Show	Edit
User	user	No	Show	Edit

Figura 22. Modulo roles

New role

Name

Slug

Description

Total access
 Yes No

[Menu](#) [Save](#)

Figura 23. Vista creación de un nuevo rol

The screenshot shows a web interface for managing labels. At the top, there is a 'Labels' header and a 'New' button. Below is a table with the following data:

Name	Show	Edit
Arboles	Show	Edit
Frutales	Show	Edit
Café	Show	Edit
Suelo desnudo	Show	Edit
Cultivo	Show	Edit
Pastizales	Show	Edit
Vías	Show	Edit

Figura 24. Modulo labels

The screenshot shows a form titled 'New label'. It contains a text input field with the placeholder text 'Name'. Below the input field are two buttons: 'Menu' and 'Save'.

Figura 25. Vista creación de una nueva etiqueta

Experiments

New experiment

Name	Size	Training size	Labeled	Total						
test6	20	10		Processing	Show	Labeled	Load	Train	Delete	Download drill
First_test_10000	12000	10000	439	11917	Show	Labeled	Load	Train	Delete	Download drill
test_terra4	20	15	20	20	Show	Labeled	Load	Train	Delete	Download drill
test_terra3	20	15	0	20	Show	Labeled	Load	Train	Delete	Download drill
test2	10000	8000	0	10000	Show	Labeled	Load	Train	Delete	Download drill

Figura 26. Módulo experimentos

New experiment

Name

Width

Height

Size

Training size

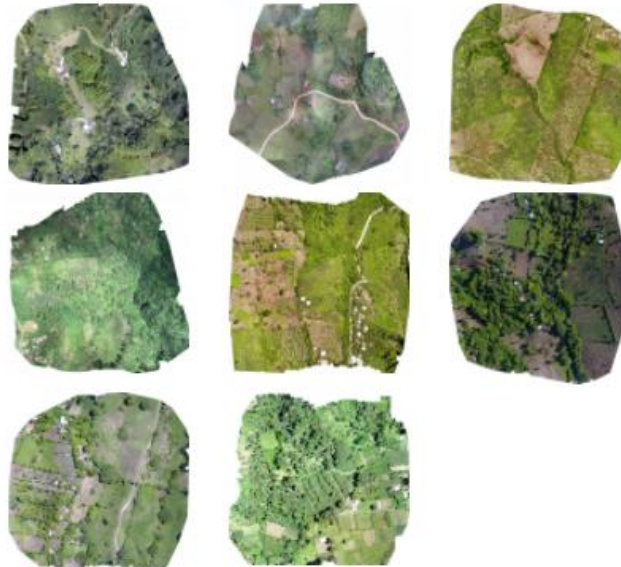
Labels

Please select the labels that you are going to use in the experiment

- | | | |
|--|-----------------------------------|-------------------------------------|
| <input type="checkbox"/> Arboles | <input type="checkbox"/> Frutales | <input type="checkbox"/> Café |
| <input type="checkbox"/> Suelo desnudo | <input type="checkbox"/> Cultivo | <input type="checkbox"/> Pastizales |
| <input type="checkbox"/> Vías | <input type="checkbox"/> Casas | <input type="checkbox"/> NA |

Base maps

Select the images to use in the experiment



Menu Save

Figura 27. Vista creación de un nuevo experimento

Show experiment

Name

Width

Height

Size

Training size

Labels

- | | | |
|---|--|--|
| <input checked="" type="checkbox"/> Arboles | <input checked="" type="checkbox"/> Frutales | <input checked="" type="checkbox"/> Café |
| <input checked="" type="checkbox"/> Suelo desnudo | <input checked="" type="checkbox"/> Cultivo | <input checked="" type="checkbox"/> Pastizales |
| <input checked="" type="checkbox"/> Vías | <input checked="" type="checkbox"/> Casas | <input checked="" type="checkbox"/> NA |

Base maps



Menu

Figura 28. Vista de mostrar un experimento creado

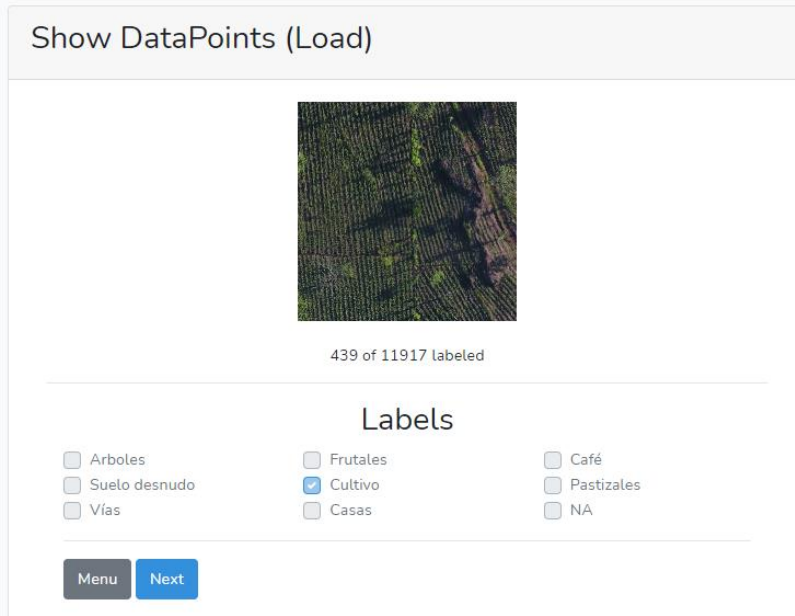


Figura 29. Vista de una muestra etiquetada

6.7 Prototipo de clasificación de coberturas

El prototipo de clasificación de coberturas en imágenes tomadas por drone fue desarrollado en lenguaje java usando la librería DL4J, basado en el paper ImageNet Classification with Deep Convolutional Neural Networks y bajo la supervisión del científico del CIAT Dr. Louis Reymondin. De acuerdo con el paper para mejorar el reconocimiento de objetos se pueden recopilar conjuntos más grandes de datos, aprender modelos más potentes y utilizar mejores técnicas para prevenir el sobreajuste (Krizhevsky, et al 2012).

Los objetos en entornos realistas presentan una variabilidad considerable, por lo que para aprender a reconocerlos es necesario utilizar conjuntos de entrenamiento mucho más grandes (Krizhevsky, et al 2012).

La capacidad de una red neuronal convolucional CNN se puede controlar variando su profundidad y amplitud y tiene la ventaja de que es más fácil de entrenar que una red neuronal feedforward estándar con capas de tamaño similar.

Una desventaja de implementar este tipo de modelos es que computacionalmente es costosa, pero esto ha ido disminuyendo gracias a que actualmente se puede contar con GPU que ayudan a que la tarea de entrenamiento de CNN interesantes y grandes sea más eficiente.

El objetivo de desarrollar este prototipo es mostrar la utilidad de una CNN en el proceso de clasificación de coberturas en imágenes tomadas por dron.

El dataset

La aplicación web de etiquetado de imágenes tomadas por dron fue la base para la obtención de imágenes de entrada, en el paper de referencia la cantidad de ejemplos etiquetados fue de 1.2 millones (Krizhevsky, et al 2012). Para este ejercicio se etiquetaron 10000 muestras a través de la aplicación web. Las muestras seleccionadas aleatoriamente provienen de las imágenes recopiladas y creadas en la sección 6.1.1 y presentadas en la figura 10. El tamaño de cada una de las muestras se definió de $200 * 200$ pixeles en esta primera prueba, esto principalmente porque para estas imágenes proveen una buena resolución para reconocer las clases de interés y es lo suficientemente pequeña para que el modelo pueda ser entrenado en poco tiempo.

Las etiquetas a utilizar fueron clases de interés para los investigadores, estas fueron:


- Arboles
- Frutales
- Café
- Suelo desnudo
- Cultivo
- Pastizales








- Vías
- Casas
- NA


Como se mencionó anteriormente el número total de muestras fue de 10000, el tamaño de entrenamiento fue definido como el 80% que correspondería a 8000 muestras y el tamaño del conjunto de pruebas fue definido como el 20% que correspondería a 2000 muestras. En la tabla 3 podemos observar que el total de muestras seleccionado para entrenamiento fue de 7459 el de pruebas fue de 1865, esta diferencia se debe a que la librería DL4J empleada para la codificación del modelo emplea funciones como FileSplit, ParentPathLabelGenerator y RandomPathFilter que ayudan a introducir más aleatoriedad en la decisión probabilística y, por lo tanto, aumentará el rendimiento del modelo (métricas de precisión). La aleatorización ayuda a superar los problemas de sobreajuste y el uso de estas funciones también hace que se puedan obtener lotes óptimos para el entrenamiento.

La tabla 3 presenta la cantidad de muestras totales que fueron tomadas por cada clase, cantidad de muestras seleccionadas para entrenamiento y la cantidad de muestras seleccionadas para validación, adicionalmente se incluye un ejemplo de cada cobertura.

Tabla 3. Cantidad de muestras tomadas por clase, muestras seleccionadas para entrenamiento y pruebas por clase

Clases	Cantidad de muestras totales por clase	Cantidad de muestras seleccionadas para entrenamiento	Cantidad de muestras seleccionadas para pruebas	Ejemplo de cobertura por clase
Árboles	1651	1161	271	

Café	1067	794	154	
Casas	291	212	42	
Cultivos	632	426	114	
Frutales	1856	1385	387	
Pastizales	1448	1158	301	
Suelo desnudo	1169	935	228	
Vias	215	151	37	

NA	1671	1237	331	
Total	10000	7459	1865	

La arquitectura

La arquitectura de nuestra red se resume contiene ocho capas aprendidas: cinco convolucionales y tres completamente conectadas, adicionalmente dentro del modelo se usan dos capas de normalización y tres de agrupación o max pooling. Algunas de las características novedosas o inusuales de la arquitectura de la red de acuerdo con el paper se describen a continuación:

No linealidad ReLU

La forma estándar de modelar la salida de una neurona f como una función de su entrada x es $f(x) = \tanh(x)$ o $f(x) = (1 + e^{-x})^{-1}$. En términos de tiempo de entrenamiento con descenso de gradiente, estas no linealidades de saturación son mucho más lentas que la no linealidad de no saturación $f(x) = \max(0; x)$. Las neuronas con esta no linealidad se definen como Unidades Lineales Rectificadas (ReLU). Las redes neuronales convolucionales profundas con ReLU se entrenan varias veces más rápido que sus equivalentes con unidades tanh. El aprendizaje más rápido tiene una gran influencia en el rendimiento de grandes modelos entrenados en grandes conjuntos de datos. Esta función de activación es utilizada después de cada convolución, la operación que realiza es reemplazar los valores negativos por cero (Krizhevsky, et al 2012).

Entrenamiento del modelo en servidor con GPU

El entrenamiento del modelo se realizó en un servidor Dell EMC PowerEdge R730, 2 procesadores Intel Xeon E5-2650 v4 2.2 GHz 12 cores y con 100GB de memoria RAM de las cuales para el entrenamiento se usaron 30GB, este servidor se encuentra en las instalaciones del CIAT. En el entrenamiento del modelo se utilizaron un total de 10000 muestras etiquetadas distribuidas en las 9 clases definidas por los investigadores del CIAT. El conjunto de entrenamiento y pruebas se definió a razón de 80% y 20% (Figura 30). Se ejecutaron primero 50 épocas las cuales tomaron aproximadamente 4 días, después de esto usando el modelo resultante de esta primera ejecución se volvieron a correr nuevamente otras 50 épocas, el tamaño de lote usado fue 256 registros.

```
[main] INFO org.nd4j.linalg.factory.Nd4jBackend - Loaded [CpuBackend] backend
[main] INFO org.nd4j.nativeblas.NativeOpsHolder - Number of threads used for NativeOps: 4
[main] INFO org.reflections.Reflections - Reflections took 71 ms to scan 11 urls, producing 29 keys and 189 values
[main] INFO org.nd4j.nativeblas.Nd4jBlas - Number of threads used for BLAS: 4
[main] INFO org.nd4j.linalg.api.ops.executioner.DefaultOpExecutioner - Backend used: [CPU]; OS: [Windows Server 2016]
[main] INFO org.nd4j.linalg.api.ops.executioner.DefaultOpExecutioner - Cores: [16]; Memory: [26.7GB];
[main] INFO org.nd4j.linalg.api.ops.executioner.DefaultOpExecutioner - Blas vendor: [OPENBLAS]
[main] INFO org.reflections.Reflections - Reflections took 1872 ms to scan 129 urls, producing 5512 keys and 24422 values
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - Number number of classes : 9
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - Number of samples : 10000
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - Training split : 0.8
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - Batch size : 256
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - Epochs : 50
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - Number of cores : 12
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - ===== Load data =====
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - Train data size : 7459
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - Test data size : 1865
[main] INFO org.reflections.Reflections - Reflections took 99 ms to scan 11 urls, producing 371 keys and 1443 values
[main] INFO org.ciat.drill.deep.dl4j.DL4JTrainer - ===== Train model =====
```

Figura 30. Parámetros de configuración para el entrenamiento del modelo

Reducción de sobreajuste

Para reducir el sobreajuste en la red se utiliza el dropout o abandono el cual es un método que desactiva un numero de neuronas de una red neuronal de forma aleatoria. En cada iteración de la red neuronal dropout desactivara diferentes neuronas, las neuronas desactivadas no se toman en cuenta para el **forwardpropagation** ni para el **backwardpropagation** lo que obliga a las neuronas cercanas a no depender tanto de las neuronas desactivadas. Este método ayuda a reducir el **overfitting** ya que las neuronas cercanas suelen aprender patrones que se relacionan y estas relaciones

pueden llegar a formar un patrón muy específico con los datos de entrenamiento, con dropout esta dependencia entre neuronas es menor en toda la red neuronal, de esta manera las neuronas necesitan trabajar mejor de forma solitaria y no depender tanto de las relaciones con las neuronas vecinas.

Desarrollo del prototipo de clasificación de coberturas

El prototipo de clasificación de coberturas se desarrolló en lenguaje java usando la librería DeepLearning4j DL4J la cual es una librería open-source y basado en (Krizhevsky, et al 2012).

Cada flujo de trabajo de aprendizaje automático consta de al menos dos partes. El primero es cargar sus datos y prepararlos para usarlos en el aprendizaje. Este proceso es conocido como ETL (extraer, transformar, cargar). DataVec es la biblioteca que usa DL4J para la creación de canalizaciones de datos. La segunda parte es el propio sistema de aprendizaje. Ese es el núcleo algorítmico de DL4J. Gracias a sus diferentes backends, esta librería nos permite utilizar tanto CPU como GPU.

A continuación, se muestran el diagrama de paquetes (Figura 31) y diagrama de clases del prototipo de clasificación de coberturas (Figura 32).



Figura 31. Diagrama de paquetes prototipo de clasificación de coberturas

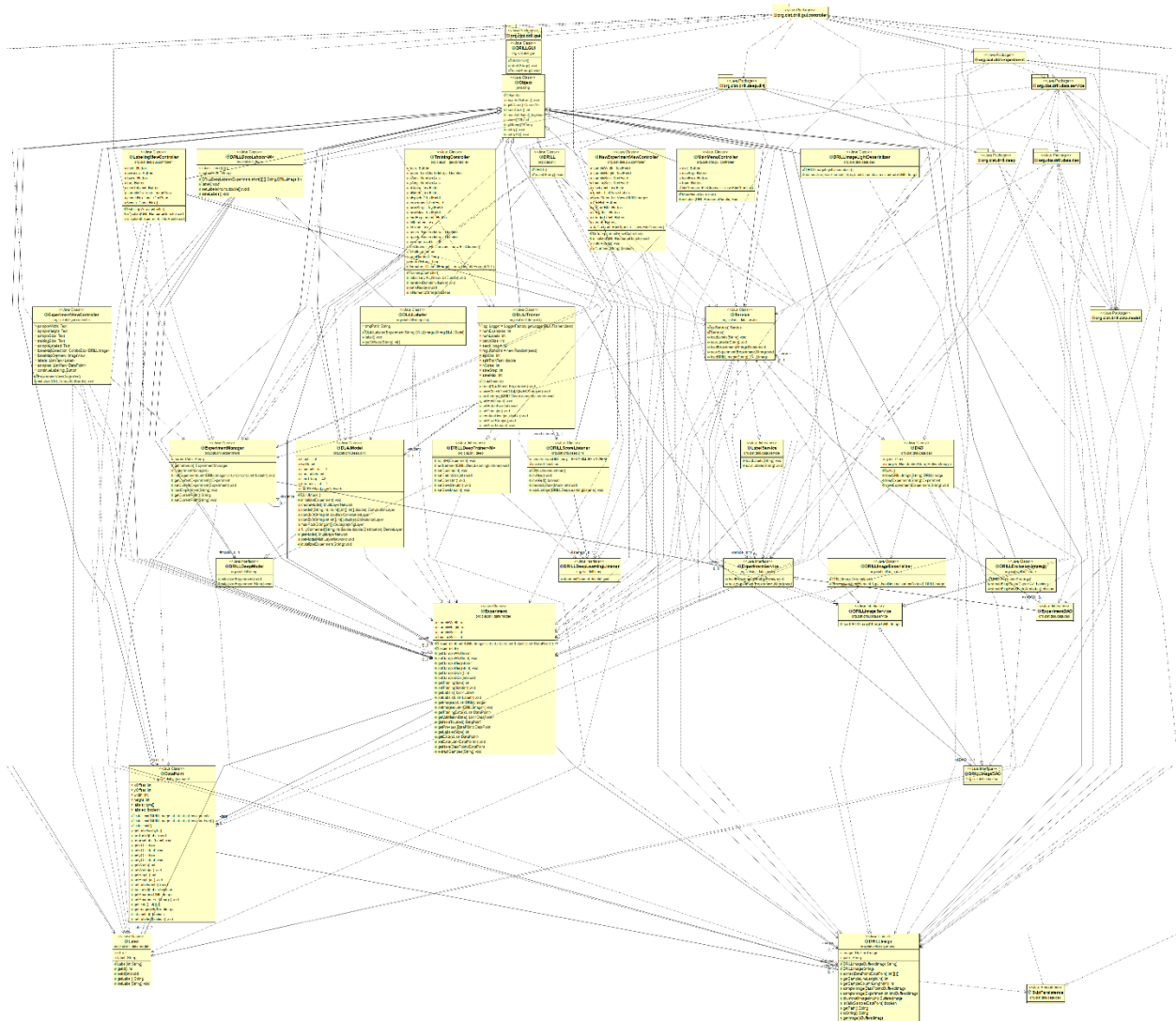


Figura 32. Diagrama de clases prototipo de clasificación de coberturas. Ver anexo 3

Estructura y parámetros de la CNN

Seguendo (Krizhevsky, et al 2012), la red que se implementó está constituida de 13 capas con la siguiente configuración (Tabla 4):

Tabla 4. Estructura y parámetros utilizados en la CNN

Capa	Tipo	Parámetros
0	Convolución	Filters: 96 kernel: 11 x 11 stride: 4 x 4 padding: 3 x 3
1	Normalización	
2	Max Pool	kernel: 3 x 3 stride: 2 x 2
3	Convolución	Filters: 256 kernel: 5 x 5 stride: 1 x 1 padding: 2 x 2 Bias: 1
4	Normalización	
5	Max Pool	kernel: 3 x 3 stride: 2 x 2
6	Convolución	Filters: 384 kernel: 3 x 3 stride: 1 x 1 padding: 1 x 1 Bias: 0
7	Convolución	Filters: 384 kernel: 3 x 3 stride: 1 x 1 padding: 1 x 1 Bias: 1
8	Convolución	Filters: 256 kernel: 3 x 3 stride: 1 x 1 padding: 1 x 1 Bias: 1
9	Max Pool	kernel: 3 x 3

		stride: 2 x 2
10	Capa completamente conectada	Filters: 4096 Bias: 1 dropout: 0.5
		Gaussian Distribution: 0 y 0.005
11	Capa completamente conectada	Filters: 4096 Bias: 1 dropout: 0.5
		Gaussian Distribution: 0 y 0.005
12	Capa de salida	Softmax: 9 Loss function: Negative Log Likelihood

La primera capa convolucional filtra la imagen de entrada de 200 x 200 x 3 con 96 núcleos o kernels de tamaño 11 x 11 x 3, un stride de 4 pixeles, este parámetro ayuda a que la operación convolucional sea más rápida y un padding de 3 pixeles, este se encarga de agregar pixeles alrededor de una imagen.

La segunda capa convolucional toma como entrada la salida con la respuesta normalizada y el paso de max pool de la primera capa convolucional y la filtra con 256 núcleos o kernels de tamaño 5 x 5, con un stride de 1 y un padding de 2.

La tercera, cuarta y quinta capas convolucionales están conectadas entre sí sin cualquier capa intermedia de agrupación o normalización. La tercera capa convolucional tiene 384 núcleos o kernels de tamaño 3 x 3, con un stride de 1 y un padding de 1 conectados a las salidas con la respuesta normalizada y el paso de max pool de la segunda capa

convolucional. La cuarta capa convolucional tiene 384 núcleos o kernels de tamaño 3×3 , con un stride de 1 y un padding de 1, la quinta capa convolucional tiene 256 núcleos o kernels de tamaño 3×3 , con un stride de 1 y un padding de 1. Las capas completamente conectadas tienen 4096 neuronas cada una (Krizhevsky, et al 2012).

Los procesos de normalización y agrupación se realizan para ayudar a disminuir la tasa de error y evitar el sobreajuste.

La salida de la última capa completamente conectada se alimenta a un softmax de 9 neuronas que produce una distribución sobre las etiquetas de las 9 clases usadas y se usa la función de pérdida Negative Log Likelihood.

El entrenamiento del modelo se realizó usando el algoritmo de optimización descenso de gradiente estocástico con un tamaño de lote de 256 ejemplos, un impulso de 0.9 y un descenso de peso de 0,0005. Esta pequeña variación en los pesos es importante para que el modelo aprenda, esto quiere decir que ayuda a que las funciones de error calculen el error necesario para la modificación de los pesos sin caer en overfitting (Krizhevsky, et al 2012).

Los pesos en cada capa fueron inicializados a partir de una distribución gaussiana de media cero con una desviación estándar de 0,01. Se inicializaron los sesgos de las neuronas en la segunda, cuarta y quinta capas convolucionales, así como en las capas ocultas completamente conectadas, con la constante 1. Esta inicialización acelera las primeras etapas del aprendizaje al proporcionar a las ReLU entradas positivas. Los sesgos de neuronas en las capas restantes fueron inicializados con la constante 0. La tasa de aprendizaje usada fue igual para todas las capas y se inicializó en 0.01. El parámetro de dropout se definió en 0.5, este incide principalmente en la reducción del sobreajuste (Krizhevsky, et al 2012).

Esta red fue entrenada con 50 épocas constituidas por 30 lotes de 256 registros cada una.

6.8 Pruebas aplicación web y prototipo

El siguiente plan de pruebas se realiza con el fin de evaluar la aplicación web para el etiquetado de imágenes tomadas por drones que es una aplicación creada para el apoyo

a los investigadores en el proceso de interpretación de imágenes de alta resolución y clasificación de coberturas.

Tipo de aplicación: Aplicación Web y aplicación de escritorio

Para realizar las pruebas del prototipo funcional se definieron y ejecutaron 21 casos de pruebas para validar las principales funcionalidades la aplicación web y del prototipo (Figura 33).

Las funcionalidades que se probaron fueron:

Autenticación

Creación de usuarios

Creación de roles

Creación de etiquetas

Creación de experimentos

Consulta de usuarios

Consulta de roles

Consulta de etiquetas

Consulta de experimentos

Actualización de usuarios

Actualización de roles

Actualización de etiquetas

Eliminación de usuarios

Eliminación de experimentos

Visualización de un experimento

Etiquetado de muestras

Descarga de archivo etiquetado finalizado

Entrenamiento de un modelo

Clasificación de coberturas

ID	Nombre Caso de prueba	Descripción caso de prueba	Resultado
CP01	Autenticación de un usuario con rol administrador	Un usuario registrado con rol de administrador ingresa en la aplicación web	Exitoso
CP02	Autenticación de un usuario con rol usuario	Un usuario registrado con rol de usuario ingresa en la aplicación web	Exitoso
CP03	Creación de un usuario con rol administrador	Se ingresa al sistema con un usuario administrador y se crea un usuario nuevo con rol administrador	Exitoso
CP04	Creación de un usuario con rol usuario	Se ingresa al sistema con un usuario administrador y se crea un usuario nuevo con rol usuario	Exitoso
CP05	Creación de un rol administrador	Se ingresa al sistema con un usuario administrador y se crea un nuevo rol administrador	Exitoso
CP06	Creación de un rol usuario	Se ingresa al sistema con un usuario administrador y se crea un nuevo rol usuario	Exitoso
CP07	Creación de una etiqueta nueva	Se ingresa al sistema con un usuario administrador y se crea una nueva etiqueta	Exitoso
CP08	Creación de un nuevo experimento	Se ingresa al sistema con un usuario administrador y se crea un nuevo experimento	Exitoso
CP09	Consulta de usuarios	Se ingresa al sistema con un usuario administrador y abre el modulo de usuarios	Exitoso
CP10	Consulta de roles	Se ingresa al sistema con un usuario administrador y abre el modulo de roles	Exitoso
CP11	Consulta de etiquetas	Se ingresa al sistema con un usuario administrador y abre el modulo de etiquetas	Exitoso
CP12	Consulta de experimentos	Se ingresa al sistema con un usuario registrado abre el modulo de experimentos	Exitoso
CP13	Actualización de usuarios	Se ingresa al sistema con un usuario administrador y abre el modulo de usuarios, se selecciona el boton editar de un usuario y se cambia el correo	Exitoso
CP14	Actualización de roles	Se ingresa al sistema con un usuario administrador y abre el modulo de roles, se selecciona el boton editar de un rol y se cambia el nombre	Exitoso
CP15	Actualización de etiquetas	Se ingresa al sistema con un usuario administrador y abre el modulo de etiquetas, se selecciona el boton editar de una etiqueta y se cambia el nombre	Exitoso
CP16	Eliminación de usuarios	Se ingresa al sistema con un usuario administrador y abre el modulo de usuarios, se selecciona el boton eliminar de un usuario	Exitoso
CP17	Actualización de etiquetas	Se ingresa al sistema con un usuario administrador y abre el modulo de etiquetas, se selecciona el boton eliminar de una etiqueta	Exitoso
CP18	Etiquetado de una muestra	Se ingresa al sistema con un usuario registrado, se abre el modulo de experimentos, se selecciona el boton etiquetado de un experimento y se etiqueta una submuestra	Exitoso
CP19	Descarga de archivo etiquetado	Se ingresa al sistema con un usuario registrado, se abre el modulo de experimentos, se selecciona el boton descarga de un experimento y se etiqueta una submuestra	Exitoso
CP20	Entrenamiento de un modelo	Se ingresa en el prototipo los parametros del modelo archivo de etiquetado, ruta de salida del modelo, batchsize, cores, epochs, save max y save step, y se ejecuta el	Exitoso
CP21	Clasificar coberturas	Se ingresa en el prototipo una imagen en formato geotiff, ruta del modelo y ruta de salida y se ejecuta el proceso de clasificación	Exitoso

Figura 33. Casos de prueba aplicación web y prototipo

6.9 Evaluación del prototipo de clasificación de coberturas

En esta fase se realizó una medición del desempeño del prototipo de clasificación de coberturas usando la librería DeepLearning4G DL4G para esto se utilizaron métricas de precisión por clase, Exhaustividad (Recall) por clase y el Accuracy global.

Adicionalmente se tomó el tiempo aproximado de entrenamiento de un modelo y el tiempo de aplicar este modelo a una imagen geotiff de 6 ha aproximadamente.

Las clases a evaluar fueron: arboles, frutales, café, suelo desnudo, cultivo, pastizales, vías, casas y NA. La figura 34 muestra un ejemplo de la clase NA que corresponde a

zonas donde la imagen de alta resolución pueda tener huecos de información o la imagen esté borrosa usualmente cuando se procesa una imagen geotiff usando Pix4D los bordes tienden a verse distorsionados.



Figura 34. Ejemplo de una muestra etiquetada como NA

El entrenamiento del modelo se realizó con los siguientes parámetros:

Número de clases: 9

Número de muestras: 10000

Tamaño de entrenamiento: 80%

Tamaño del lote: 256

Épocas: 100

Numero de núcleos: 12

Memoria: 30GB

Número de iteraciones realizada en este proceso fue de: 20320 iteraciones

Tiempo de duración del entrenamiento del modelo: 8 días aproximadamente

Matriz de confusión	Arboles	Café	Casas	Cultivos	Frutales	Pastizales	Suelo desnudo	Vías	NA	Total	Precisión
Arboles	184	32	0	2	28	5	0	0	20	271	0.68
Café	20	110	1	3	15	3	0	0	2	154	0.71
Casas	1	0	35	1	1	0	2	0	2	42	0.83
Cultivos	0	3	0	95	4	5	5	0	2	114	0.83
Frutales	20	8	1	0	339	8	9	0	2	387	0.88
Pastizales	2	1	0	16	8	251	12	1	10	301	0.83
Suelo desnudo	5	1	1	2	14	7	197	0	1	228	0.86
Vías	0	0	2	0	0	1	3	29	2	37	0.78
NA	9	5	4	2	7	5	7	1	291	331	0.88
Total	241	160	44	121	416	285	235	31	332	1865	
Recall	0.76	0.69	0.80	0.79	0.81	0.88	0.84	0.94	0.88		

Figura 35. Matriz de confusión prototipo de clasificación de coberturas

La matriz de confusión en la figura 35 presenta las diferentes clases evaluadas y las métricas de precisión y Recall en las cuales se evidencia que aún se debe entrenar un poco más el modelo porque hay confusión entre las clases árboles y café.

El Accuracy del modelo es del 82%. Esto podría mejorar corriendo el modelo con más épocas.

6.10 Prueba del modelo de clasificación en una imagen de alta resolución

La figura 36 muestra la prueba del prototipo de clasificación de coberturas aplicado a una imagen geotiff de alta resolución de un sobrevuelo realizado en El Salvador en 2019 que cubre un área de aproximadamente 7ha, en la parte inferior de la imagen se muestran los resultados de la clasificación por cada una de las clases evaluadas. La ejecución del proceso de clasificación para esta imagen duró aproximadamente 5 minutos.

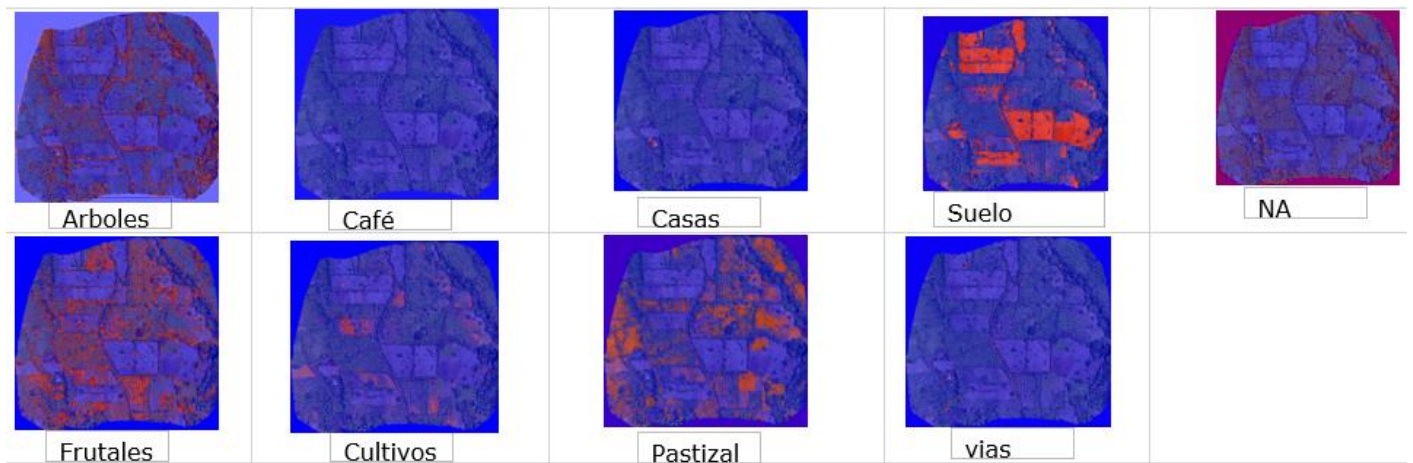


Figura 36. Prueba del prototipo de clasificación

7. GLOSARIO DE TERMINOS ESPECIALES

CNN: Convolutional Neural Network (Red Neuronal Convocucional)

Drones: También conocidos como unmanned aerial vehicle (UAV), son aviones o multirrotores controlados de forma remota o autónoma que siguen una línea de vuelo preprogramada (Torres et al., 2013).

Época: Es un hiperparámetro que define el número de veces que el algoritmo de aprendizaje va a trabajar a través de todo el conjunto de datos de entrenamiento.

Una época significa que cada muestra del conjunto de datos de entrenamiento ha tenido la oportunidad de actualizar los parámetros internos del modelo.

Experimento: Es un conjunto de etiquetas y muestreos correspondientes a una o varias imágenes geotiff.

Lote o Batch: Es una cantidad de muestras procesadas antes de que se actualice el modelo.

Imagen digital: Es una representación bidimensional de una imagen a partir de una matriz numérica, frecuentemente en binario (unos y ceros). Dependiendo de si la resolución de la imagen es estática o dinámica, puede tratarse de una imagen matricial (o mapa de bits) o de un gráfico vectorial. El mapa de bits es el formato más utilizado en informática ("Imagen digital", 2018).

Ortomosaico: Es un producto de imagen fotogramétricamente ortorrectificado organizado como mosaico a partir de una colección de imágenes, donde la distorsión geométrica se ha corregido y donde se ha realizado un balance de color de las imágenes para producir un dataset de mosaico continuo ("Generar un ortomosaico | Documentación", 2021).

Pixel: Es la unidad mínima de color que hace parte de una imagen digital.

ReLU: Rectified Linear Unit esta función transforma los valores introducidos anulando los valores negativos y dejando los positivos tal y como entran.

Teledetección: Es una técnica que permite adquirir imágenes desde la superficie terrestre desde sensores instalados en plataformas a distancia y su posterior tratamiento en el contexto de una determinada aplicación (Perry, 2018).

Terra-i: Es un sistema de alerta temprana para monitorear los cambios en el hábitat en toda América Latina y el resto de trópicos.

8. CONCLUSIONES

Una vez finalizado el proceso de desarrollo de la aplicación web para el etiquetado de imágenes de alta resolución se puede concluir lo siguiente:

Contar con los conocimientos en el proceso de ingeniería de requisitos y arquitectura de software hace que sea posible tener una mayor claridad sobre el camino a seguir para brindar una solución que permita a los stakeholders tener una satisfacción mayor sobre el producto final.

Usar un framework de desarrollo permite que la productividad se incremente y la solución final sea construida bajo buenas prácticas de desarrollo, esto también tiene un impacto en la facilidad del mantenimiento de las aplicaciones a futuro.

Identificar los stakeholders más importantes es clave a la hora de realizar una recopilación de los requisitos que ayuden a tener una solución más completa.

Es un apoyo importante para los stakeholders contar con una herramienta donde puedan crear una base de información de coberturas de interés que además permite que este trabajo se pueda realizar en línea y hacer trabajo colaborativo.

Con respecto al prototipo de clasificación de coberturas usando técnicas de deep learning se puede concluir:

En el desarrollo de este trabajo se ha logrado crear un conjunto de datos de coberturas desde cero y este a su vez ha sido usado para el entrenamiento de un prototipo de clasificación de coberturas basado en una CNN obteniendo un modelo donde las métricas de evaluación del mismo resultaron ser satisfactorias

El entrenamiento de una CNN requiere de un conjunto grande de datos, pero además de esto el conjunto de datos debe ser de calidad para que el entrenamiento no se vea afectado y el modelo resultante no sea eficiente.

Es un apoyo importante para los stakeholders contar con una herramienta donde el proceso de clasificación de coberturas pueda hacerse de una forma más eficiente cuando se usan imágenes de alta resolución.

9. TRABAJO FUTURO

Aplicación web de etiquetado de imágenes

El proyecto de investigación realizado es la primera parte de la construcción de una aplicación web para el etiquetado de imágenes dron de alta resolución. El prototipo de clasificación de coberturas actualmente funciona como una herramienta de escritorio que la cual recibe como entrada el archivo etiquetado para luego hacer el proceso de entrenamiento y clasificación. El siguiente paso será integrar completamente el prototipo de clasificación de coberturas para que sea posible hacer proceso de entrenamiento y ejecución del modelo directamente desde la aplicación web.

En la siguiente fase se puede explorar como integrar el uso de otros modelos dentro de la aplicación web, esto sería útil para probar en línea como es el comportamiento de diferentes modelos en una imagen.

Prototipo de clasificación de coberturas

Se deben seguir creando ejemplos etiquetados que ayuden al modelo a mejorar pues estas redes van mejorando a medida que se tienen más cantidad de imágenes. También se debería correr el proceso de entrenamiento con más épocas para ver que tanto mejora el modelo.

Se pueden usar los experimentos creados para probar otras arquitecturas de CNN ya existentes.

Explorar otras técnicas como el transfer learning, esto puede ser integrado al prototipo con el fin de acelerar el aprendizaje.

Esta herramienta también puede ser usada como apoyo en otras actividades como hacer análisis de fronteras agrícolas, línea base de coberturas, análisis de zonas sembradas de cultivos como caña, arroz, palma, entre otros.

10. ANEXOS

Anexo 1. Formato de entrevista utilizado

Preguntas Entrevista

Nombres y Apellidos:

Profesión:

Edad:

Correo

1. ¿Cuántos años lleva en el proyecto Terra-i?
2. ¿Cuántos años de experiencia tiene en el uso de sensores remotos?
3. ¿Qué herramientas para análisis espacial conoce?
4. ¿Cuáles son las principales dificultades al momento de hacer el análisis de cambios en coberturas usando imágenes de alta resolución?
5. En su experiencia personal. ¿Qué nivel de dificultad observa cuando debe hacer identificación de coberturas en imágenes de alta resolución como las de drones?
6. En su experiencia personal. ¿Qué nivel de importancia le daría a una herramienta que ayude en la identificación de coberturas en imágenes de alta resolución tomadas con drones?
7. ¿Cuáles coberturas le parece son las que principalmente se deberían incluir en la clasificación de coberturas en imágenes de alta resolución tomadas con drones?
8. ¿Qué funcionalidades piensa se deberían incluir en la plataforma de etiquetado web?

Anexo 2. Formato y resultado del proceso de priorización de requisitos

https://drive.google.com/file/d/1W2zh9ydhFEeO32BoprB6F_8OWmFOplZ7/view?usp=sharing

Anexo 3. Diagrama de clases prototipo de clasificación de coberturas completo

<https://drive.google.com/file/d/1PZqd5PcGyRRb7i-MfKOKfzATgWlhoUjp/view?usp=sharing>

11. REFERENCIAS BIBLIOGRÁFICAS

Al-Khudhair, D. H. A., Caravaggi, I., & Giada, S. (2005). Structural damage assessments from Ikonos data using change detection, object-oriented segmentation, and classification techniques. *Photogrammetric Engineering & Remote Sensing*, 71(7), 825-837.

Amazon Rekognition Image. (2021). Consultado 20 febrero 2021, desde <https://aws.amazon.com/es/rekognition/image-features/?nc=sn&loc=3&dn=2>

Aprendizaje Profundo: Tres cosas que es necesario saber. (2018). Consultado 15 diciembre 2018, desde <https://la.mathworks.com/discovery/deep-learning.html>

Asner, G. P., Knapp, D. E., Balaji, A., & Páez-Acosta, G. (2009). Automated mapping of tropical deforestation and forest degradation: CLASlite. *Journal of Applied Remote Sensing*, 3(1), 033543.

Cánovas García, F. (2012). Análisis de imágenes basado en objetos (OBIA) y aprendizaje automático para la obtención de mapas de coberturas del suelo a partir de imágenes de muy alta resolución espacial. Aplicación en la Unidad de Demanda Agraria n.º 28, Cabecera del Argos. *Revista de Teledetección*, 38, 89-90.

Chen, C., Seff, A., Kornhauser, A., & Xiao, J. (2015, December). Deepdriving: Learning affordance for direct perception in autonomous driving. In *Computer Vision (ICCV), 2015 IEEE International Conference on* (pp. 2722-2730). IEEE.

Chris V. Nicholson, S. (2018). *Deeplearning4j: Open-source, Distributed Deep Learning for the JVM*. Consultado 15 de diciembre desde <https://deeplearning4j.org/>

Cíceri, M. (2019). *Introducción a Laravel: Aplicaciones robustas ya gran escala*. RedUsers.

Coca-Castro, A., Reymondin, L., Bellfield, H., & Hyman, G. (2013). Land use status and trends in Amazonia. Report for Global Canopy Programme and International Center for Tropical Agriculture as part of the Amazonia Security Agenda.

Collect Earth Online: Open Foris. (2021). Consultado 20 febrero 2021, desde <http://www.openforis.org/tools/collect-earth-online.html>

Defries, R. S., Hansen, M. C., Townshend, J. R., Janetos, A. C. and Loveland, T. R. (2000), A new global 1-km dataset of percentage tree cover derived from remote sensing. *Global Change Biology*, 6: 247-254.

European Environment Agency, 2013. CORINE Land Cover (CLC) 2006, Version 17, Copenhagen K, Denmark. CRC/TR32 Database (TR32DB).

ERDAS VS ENVI: Which is The Best Remote Sensing Image Processing Software? (2018). Consultado 15 de diciembre desde <http://monde-geospatial.com/erdas-vs-envi-best-remote-sensing-image-processing-software/>

Etter, A. Introducción a la ecología del Paisaje. 1991. Instituto Geográfico Agustín Codazzi.

Fukushima, K., Miyake, S., & Ito, T. (1983). Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (5), 826-834.

Fu, G., Liu, C., Zhou, R., Sun, T., & Zhang, Q. (2017). Classification for high resolution remote sensing imagery using a fully convolutional network. *Remote Sensing*, 9(5), 498.

Generar un ortomosaico | Documentación. (2021). Consultado 20 febrero 2021, desde <https://pro.arcgis.com/es/pro-app/latest/help/data/imagery/generate-an-orthomosaics-using-the-orthomosaic-wizard.htm>

Google Earth Engine. (2021). Consultado 20 febrero 2021, desde <https://earthengine.google.com/>

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y.,... & Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35, 18-31.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hernández, G. (2014). Aplicación de redes neuronales para la identificación de objetos en tiempo real en imágenes tomadas por un quadrotor.

Hermes, L., Frieauff, D., Puzicha, J., & Buhmann, J. M. (1999). Support vector machines for land usage classification in Landsat TM imagery. In *Geoscience and Remote Sensing Symposium, 1999. IGARSS'99 Proceedings. IEEE 1999 International* (Vol. 1, pp. 348-350). IEEE.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Imagen digital. (2018). Consultado 15 de diciembre desde https://es.wikipedia.org/wiki/Imagen_digital

IEEE/ISO/IEC 29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering. (2020). Retrieved 13 April 2020, from <https://standards.ieee.org/standard/29148-2011.html>

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 675-678). ACM.

Kampouraki, M., Wood, G. A., & Brewer, T. R. (2008). Opportunities and limitations of object-based image analysis for detecting urban impervious and vegetated surfaces using true-colour aerial photography. In *Object-Based Image Analysis* (pp. 555-569). Springer, Berlin, Heidelberg.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).

Kovalev, V., Kalinovsky, A., & Kovalev, S. (2016). Deep learning with theano, torch, caffe, tensorflow, and deeplearning4j: Which one is the best in speed and accuracy?

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks, 8(1), 98-113.

Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. International journal of Remote sensing, 28(5), 823-870.

Melgani, F., & Bruzzone, L. (2002). Support vector machines for classification of hyperspectral remote-sensing images. In Geoscience and Remote Sensing Symposium, 2002. IGARSS'02. 2002 IEEE International (Vol. 1, pp. 506-508). IEEE.

Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. IEEE Transactions on geoscience and remote sensing, 42(8), 1778-1790.

Perry, A. (2018). Diseño del proceso de clasificación supervisada de cubiertas del suelo en imágenes satelitales. Consultado 15 de diciembre desde <http://repositorio.uchile.cl/bitstream/handle/2250/145336/Dise%C3%B1o-del-proceso-de-clasificacion-supervisada-de-cubiertas-de-suelo-en-imagenes-satelitales.pdf?sequence=1&isAllowed=y>

Reymondin, L., Jarvis, A., Perez-Urbe, A., Touval, J., Argote, K., Rebetez, J.,... & Mulligan, M. (2012). A methodology for near real-time monitoring of habitat change at

continental scales using MODIS-NDVI and TRMM. Submitted Remote Sensing of Environment.

Sathya, R., & Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2), 34-38.

Scott, G. J., England, M. R., Starns, W. A., Marcum, R. A., & Davis, C. H. (2017). Training deep convolutional neural networks for land–cover classification of high-resolution imagery. *IEEE Geoscience and Remote Sensing Letters*, 14(4), 549-553.

Scribe Data Labeler | AI Data Annotation Tool | Clarifai. (2021). Consultado 20 febrero 2021, desde <https://www.clarifai.com/products/label>

Sorokina, K. (2018). Image Classification with Convolutional Neural Networks. Retrieved from <https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8>

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2014). Going deeper with convolutions, CoRR abs/1409.4842. URL <http://arxiv.org/abs/1409.4842>.

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1701-1708).

tensorflow/tensorflow.(2018).Consultado 15 de Diciembre desde <https://github.com/tensorflow/tensorflow>

Torres-Sánchez, J., Peña-Barragán, J. M., de Castro-Megías, A. I., & López-Granados, F. Puesta a punto de un vehículo aéreo no tripulado (UAV) para detección de malas hierbas en fase temprana: resolución espacial y altura de vuelo.

Tso, B., & Mather, P. (2009). *Classification Methods for Remotely Sensed Data*", CRC Presst (pp. 37-38). ISBN 1-4200-9072-0.

Vásquez, A. F. R. (2011). Metodología para detectar cambios en el uso de la tierra utilizando los principios de la clasificación orientada a objetos, estudio de caso piedemonte de Villavicencio, Meta.

Vision AI | Obtén información valiosa de las imágenes a través del AA. (2021). Consultado 20 febrero 2021, desde <https://cloud.google.com/vision?hl=es-419>

Why Use Laravel as a Development Platform? (2020). Consultado 25 mayo 2020, desde <https://freshinup.com/why-use-laravel-as-a-development-platform/>

Zapata Garrido, L. A., & Díaz Mojica, H. F. (2008). Predicción del tipo de cambio peso-dólar utilizando Redes Neuronales Artificiales (rna). *Pensamiento & Gestión*, (24), 29-42.