



Reconocimiento de zonas de cultivo de café a partir de imágenes satelitales utilizando aprendizaje de máquina.

Gabriel Eduardo Alarcón Alarcón

Universidad Militar Nueva Granada

Facultad de Ingeniería

Bogotá D.C, Colombia

Año 2022

Reconocimiento de zonas de cultivo de café a partir de imágenes satelitales utilizando aprendizaje de máquina.

Gabriel Eduardo Alarcón Alarcón

Trabajo de grado presentado como requisito parcial para optar al título de:

Ingeniero en Telecomunicaciones

Director (a):

PhD, MSC Ing. Diego Renza Torres

Codirector (a):

PhD, MSC Ing. Dora María Ballesteros

Línea de Investigación:

Visión por computador

Grupo de Investigación:

GISSIC

Universidad Militar Nueva Granada

Facultad de Ingeniería

Bogotá D.C, Colombia

Año 2022

Agradecimientos

Agradezco enormemente a mis padres Adriana Alarcón y Henry Alarcón por ser ese apoyo incondicional en todas las etapas de mi vida, por permitirme ser quien soy y acompañarme en este camino arduo que pronto finalizaré, a Dios por darme la oportunidad de vivir y disfrutar cada uno de mis días en la universidad junto con mis compañeros y especialmente mi novia Daniela Mendoza, para ella también un inmenso agradecimiento por motivarme y hacer mis días realmente especiales y muy alegres, además de ser esa pareja ideal en todo el sentido, buscando siempre lo mejor para mí.

Finalmente, un agradecimiento a mis directores de opción de grado Diego Renza Torres y Dora María Ballesteros, ya que sin su gran disposición y vocación nada de esto habría sido posible, grandes maestros y personas de los cuales aprendí bastante en muchos sentidos, quedaré inmensamente agradecido con todas las personas aquí mencionadas que son parte fundamental en mi vida y lo que soy.

“VENI, VIDI, VICI.”

Resumen

En este trabajo de grado se evalúan dos estrategias de diseño de modelos de clasificación de imágenes basados en aprendizaje profundo, aplicado a “*Brazilian coffee scenes*”, correspondiente con imágenes satelitales. El objetivo es determinar en qué imágenes satelitales se tiene presencia de cultivo de café, utilizando modelos diseñados por AutoML, o por transferencia de aprendizaje. Al finalizar la investigación, se comparan los resultados de las mejores soluciones, en términos de *accuracy*, obtenidas por cada estrategia de diseño, y se selecciona el mejor modelo de los evaluados en esta investigación.

Palabras clave: Aprendizaje profundo, cultivos de café, teledetección, transferencia de aprendizaje, AutoML, *Brazilian coffee scenes*.

Abstract

This study evaluates two strategies for the design of image classification models based on deep learning, applied to "Brazilian coffee scenes", corresponding to satellite images. The objective is to determine which satellite images correspond to a coffee growing area, using models designed by AutoML or by transfer learning. At the end of the research, the results of the best solutions, in terms of accuracy, obtained by each design strategy are compared, and the best model of those evaluated in this research is selected.

Keywords: Deep learning, coffee crops, remote sensing, transfer learning, AutoML, Brazilian coffee scenes.

Tabla de contenido

1. Contexto (definición de objetivos y tareas)	12
1.1 Introducción	12
1.2 Planteamiento del problema	14
1.2.1 Identificación	14
1.2.2 Descripción	15
1.3 Pregunta de investigación	15
1.4 Justificación	15
1.5 Objetivos	16
1.5.1 Objetivo general	16
1.5.2 Objetivos específicos	16
1.6 Metodología	17
1.7 Alcance o delimitación de la investigación	20
2. Marco teórico y revisión de estado del arte	21
2.1 Marco teórico	21
2.2 Documentación del estado del arte	29
2.2.1 Trabajos que utilizan el conjunto de datos <i>Brazilian Coffee Scenes</i>	30
2.2.2 Trabajos relacionados con clasificación de imágenes satelitales.	34
3. Comprensión y preparación de los datos	38
3.1 Conjunto de datos Brazilian Coffee Scenes	38
3.2 Aplicación de aumento de datos	40
4. Implementación y evaluación de modelos	44
4.1 Modelado con AutoML (KerasTuner)	44
4.1.1 Modelo base	44
4.1.2 Espacio de búsqueda inicial y aumento de datos.	45
4.1.3 Ampliación de espacio de búsqueda	46
4.1.4 Optimización de recursos de computadora y división de modelos	47
4.1.5 Recopilación y obtención de los mejores modelos	48
4.1.6 Evaluación del mejor modelo obtenido con AutoML	50
4.2 Modelado con transferencia de aprendizaje	52
4.2.1 Modelo base.	52
4.2.2 Aumento de datos y determinación del espacio de búsqueda.	53
4.2.3 Reentrenamiento de los mejores modelos.	56
4.2.4 Obtención y evaluación del mejor modelo.	57
5. Análisis de resultados	61

6. Conclusiones	64
7. Bibliografía	66

LISTA DE FIGURAS

Figura 1. Proceso de convolución de capas convolucionales.	24
Figura 2. Funcionamiento AutoML. [20].....	27
Figura 3. Ejemplo esquema de matriz de confusión.	28
Figura 2. Matriz de confusión obtenida en el modelo implementado en [21].....	31
Figura 3. Esquema de la arquitectura total del modelo implementado en [22].....	33
Figura 4. Análisis grafico de los resultados obtenidos en [22].....	34
Figura 5. Gráfico de comparación de arquitecturas usadas en [23]	35
Figura 6. Escenas de la clase café discriminadas por zonas.....	38
Figura 7. Escenas de la clase no café discriminadas por zonas.	39
Figura 8. Distribución de imágenes del dataset	39
Figura 9. Aplicación de rotation range en aumento de datos.	40
Figura 10. Aplicación de zoom range en aumento de datos.	41
Figura 11. Aplicación de height shift range en aumento de datos.....	42
Figura 12. Aplicación de width shift range en aumento de datos.	42
Figura 13. Aplicación de vertical flip en aumento de datos.	43
Figura 14. Arquitectura de modelo base.....	44
Figura 15. Esquema arquitectura del modelo completo de AutoML seleccionado.	49
Figura 16. Resultados de clasificación de las imágenes de test: a) valores reales, b) valores estimados por el modelo obtenido con AutoML.....	50
Figura 17. Matriz de confusión del mejor modelo.	51
Figura 18. Arquitecturas Resnet50, Xception, VGG19 y VGG16 (de izquierda a derecha). Tomado de [27].	53
Figura 19. Arquitectura Xception del mejor modelo obtenido por medio de TL.	58
Figura 20. Resultados de clasificación de las imágenes de test estimadas con el modelo obtenido con Transferencia de aprendizaje: a) valores reales, b) valores obtenidos en la predicción.....	59
Figura 21. Matriz de confusión del mejor modelo obtenido con TL.	60

LISTA DE TABLAS

Tabla 1. Estado del arte: investigaciones con el dataset brazilian coffee scenes (Transferencia de aprendizaje con Res2Net para clasificación de escenas de percepción remota)	30
Tabla 2. Parámetros evaluados en [16]	31
Tabla 3. Precisión del modelo implementado en [16].	31
Tabla 4. Arquitecturas utilizadas en los modelos implementados en [16]	31
Tabla 5. Estado del arte: investigaciones con el dataset brazilian coffee scenes (Clasificación de imágenes satelitales multi escala utilizando un enfoque de aprendizaje profundo.)	32
Tabla 6. Parámetros importantes evaluados en el desarrollo del [17]	33
Tabla 7. Resultados de accuracy obtenidos en los modelos de [17]	34
Tabla 8. Estado del arte: investigaciones empleando imágenes satelitales (Clasificación de múltiples etiquetas de imágenes satelitales de teledetección.)	35
Tabla 9. Parámetros y arquitecturas evaluadas en [18]	36
Tabla 10. Estado del arte: investigaciones empleando imágenes satelitales (Teledetección para la monitorización de plantas solares fotovoltaicas en Brasil mediante segmentación semántica profunda.)	36
Tabla 11. Resultados en accuracy de los modelos implementados en [19]	37
Tabla 12. Hiperparámetros y resultados modelo base.	45
Tabla 13. Espacio de búsqueda inicial.	46
Tabla 14. Mejor modelo obtenido con aumento de datos y espacio de búsqueda inicial.	46
Tabla 15. Hiperparámetros y conjunto de valores del espacio de búsqueda de KerasTuner.	46
Tabla 16. Mejor modelo ampliación espacio de búsqueda.	47
Tabla 17. Tabla 13a y 13b. Hiperparámetros y conjunto de valores de dos espacios de búsqueda para optimización de recursos: a) Modelo 1, b) Modelo 2.	47
Tabla 18. Tabla 14. Hiperparámetros, valores y resultados de los mejores modelos obtenidos con AutoML (usando Keras Tuner).	48
Tabla 19. Hiperparámetros y valores del mejor modelo obtenido con AutoML (usando Keras Tuner)	49
Tabla 20. Parámetros modelo base.	52

Tabla 21.	Resultados Accuracy en validación para VGG16, VGG19, Xception y Resnet50.	52
Tabla 22.	Hiperparámetros y características de modelos evaluados por transferencia de aprendizaje.	54
Tabla 23.	Resultados de rendimiento de los modelos por transferencia de aprendizaje evaluados.	55
Tabla 24.	Modelos seleccionados de la fase 4.2.2	56
Tabla 25.	Resultados de los modelos de la fase 4.2.3	56
Tabla 26.	Hiperparámetros del mejor modelo de TL.	57
Tabla 27.	Resultados del mejor modelo de TL.	57
Tabla 28.	Hiperparámetros comunes AutoML y Transferencia de aprendizaje.	61
Tabla 29.	Val_accuracy mejor modelo por técnica.	62
Tabla 30.	Precisión en validación vs precisión en predicción mejores modelos.	63

1.Contexto (definición de objetivos y tareas)

1.1 Introducción

El aprendizaje de máquina es una rama de la inteligencia artificial, que por medio de algoritmos busca encontrar patrones de acuerdo con los datos que le son suministrados y que representan la base del problema. Todo esto con el fin de crear algoritmos con la capacidad de modelar y generar una respuesta acorde con un problema específico, donde dichas decisiones están estructuradas fundamentalmente en bases matemáticas y estadísticas tomadas de la recopilación de los datos que se le asignan [1].

El crecimiento tan acelerado y en masa que tiene la información puede hacer que para una persona sea muy difícil el procesar y analizar la misma, ante esto el aprendizaje de máquina constituye una alternativa para abordar este problema, ya que la rapidez con que puede analizar datos y los múltiples modelos que pueden ser aplicables generan un alto valor en su utilización, obteniendo resultados realmente precisos y veloces. En un caso típico de aprendizaje automático, se dispone de un conjunto de datos de entrada que se utilizarán para la recopilación de las características necesarias para resolver el problema mediante el algoritmo, incluyendo posibles etapas de pre-procesamiento que permitan extraer las características adecuadamente.

A la hora de realizar implementaciones utilizando aprendizaje de máquina se hace necesario entender cómo funciona y los procesos que involucra. Inicialmente se pueden discriminar tres procesos: el primero de ellos, constituye el fin mismo del algoritmo y está relacionado con dotar (entrenar) a la máquina con la capacidad de determinar el resultado de la problemática (por ejemplo el resultado de una clasificación); el segundo corresponde a un proceso de evaluación del error donde se determina la precisión y la calidad del modelo empleando diversos elementos como la función de pérdida y una función de optimización que permite que el aprendizaje sea iterativo; y un tercer proceso complementario a los dos anteriores

que se encarga de seleccionar el mejor modelo, y es aquí donde juegan un papel muy importante los hiperparámetros, que corresponden a aquellos valores que la máquina por sí sola no aprende, sino por el contrario son valores que se deben definir buscando encontrar los que mejores resultados aporten al rendimiento del modelo [2].

Más allá de esto, es importante considerar la calidad y cantidad de los datos usados en el aprendizaje de estos algoritmos, ya que el entrenamiento se realiza mediante la interacción de los modelos con estos, brindándole la capacidad al algoritmo de generalizar la tarea específica para la que se implementa. Por ejemplo, para el caso de esta investigación (reconocimiento de zonas de cultivo de café a partir de imágenes satelitales), se requieren imágenes que involucren tanto zonas de cultivos de café como también zonas con otros cultivos, las cuales deben involucrar información espectral (ej. bandas de color) y que sean de diferentes regiones geográficas. Para ello, se alimenta el modelo con imágenes provenientes del dataset *Brazilian Coffee Scenes*, que consta de 2876 imágenes de dos clases (cultivo/no cultivo), en una relación del 50% [3], y que serán la base de información e implementación de esta investigación.

Para la búsqueda y selección del algoritmo, se usarán dos métodos que permitirán evaluar diversos modelos y sus comportamientos: técnicas de transferencia de aprendizaje y técnicas de AutoML (*Automated Machine Learning*).

La transferencia de aprendizaje es una técnica en la que se emplean modelos pre-entrenados que tienen una relación cercana al problema que se quiere resolver. Al tener modelos de similares condiciones ya previamente entrenados, es posible tomar su arquitectura y parámetros y emplearlos como base para el modelo que pretende dar solución a la problemática planteada. Esto genera una reducción en los tiempos de entrenamiento y un posible mejor rendimiento al enfrentar a la máquina a una problemática de la cual, por decirlo así, ya tendría experiencia para resolver.

En relación con AutoML, estas técnicas están orientadas a facilitar tareas de aprendizaje automático, por ejemplo, en relación con la búsqueda y selección de hiperparámetros. En este sentido, las herramientas de AutoML se encargan de simplificar muchas de las tareas involucradas en el ciclo de vida de un proyecto de ML, reduciendo tiempos y ayudando a la selección de un determinado modelo.

En resumen, esta investigación tiene como fin usar el aprendizaje de máquina, específicamente con técnicas de transferencia de aprendizaje y AutoML, para seleccionar un modelo que permita el reconocimiento y clasificación de zonas de cultivo de café a partir de imágenes satelitales usando un conjunto de datos de dos clases. De acuerdo con los modelos obtenidos, se evaluarán sus resultados considerando hiperparámetros, arquitecturas y rendimiento, en función de la resolución de la problemática presentada hasta aquí.

1.2 Planteamiento del problema

1.2.1 Identificación

Actualmente, Colombia es el tercer país productor de café y el mayor productor de café suave del mundo [4], por lo que el papel que tienen las regiones y zonas cafeteras en el país es de suma importancia. Esto hace necesaria la implementación de herramientas para tener un seguimiento a los cultivos de café específicamente en todo lo que concierne a la expansión o disminución del tamaño destinado a la siembra de este.

En muchas ocasiones la generación de inventarios de forma manual se hace compleja por temas de ubicación y seguridad, aumentando los gastos y riesgos haciendo que el levantamiento y procesamiento de este tipo de información sea muy dispendioso, lo que a su vez no facilita la obtención de datos totalmente acertados.

1.2.2 Descripción

De acuerdo con lo anterior, en nuestro país la importancia de los cultivos de café cada vez se hace mayor, por lo cual el control en cuanto a cantidad, tamaño y zonas destinadas al cultivo de café es indispensable. Esto involucra la necesidad de contar con herramientas que permitan brindar una identificación y seguimiento claros de las zonas en el país que se aprovechan para esta labor, a la vez que minimicen los problemas que se generan cuando este tipo de identificación se hace de manera presencial y manual, ya sea por temas de tiempo, acceso a las zonas o de seguridad. Si bien la investigación y lo descrito en este trabajo gira en torno a un ámbito colombiano el alcance del proyecto estipula el uso del dataset *Brazilian Coffee Scenes* el cual ya está establecido y cuenta con imágenes que permiten el desarrollo correcto de los procesos de modelado.

1.3 Pregunta de investigación

Bajo el contexto anterior, se planteó la siguiente pregunta de investigación:

¿Qué hiperparámetros y/o arquitectura debe tener un modelo basado en aprendizaje profundo para permitir la identificación de cultivos de café, haciendo uso de imágenes de satélite multiespectrales?

1.4 Justificación

En Colombia, el cultivo de café forma parte fundamental de la economía, dado que representa cerca de 3 billones de pesos en exportaciones y equivale al 25% del PIB agrícola del país. Esta planta fue introducida a principios del siglo XIX por el Orinoco, luego por los Santanderes y a partir de allí, ingresó a lo que se conoce hoy como la zona cafetera central. Posteriormente pasó al norte y luego se expandió hacia el sur de Colombia [4]. Es decir, su presencia en distintas regiones del país es indispensable, por lo cual la identificación y seguimiento de este cultivo puede servir de insumo para la generación de políticas de impacto en el sector agrícola del país. Una evaluación automática de su crecimiento a lo largo y ancho del país puede

abordarse por medio del aprendizaje de máquina y el uso de imágenes satelitales que faciliten realizar el reconocimiento de las zonas de cultivo de café.

El uso de este tipo de datos facilita tareas relacionadas con el levantamiento de información en las zonas cafeteras, ya que la captura de datos se realiza de forma remota y periódica, lo que permite contrastar información de diferentes épocas. Lo anterior como una ventaja respecto a los procesos de recopilación de información presencial en la zona, que pueden representar mayores gastos y riesgos por la dificultad de acceso a las mismas dados los problemas en seguridad que se presentan en el país.

Por su parte, la apropiación de nuevas tecnologías como es el aprendizaje de máquina para la solución de problemas locales constituye una herramienta de apoyo en actividades gubernamentales. En particular, la automatización de tareas y el uso de información a gran escala, puede constituir un soporte en la toma de decisiones. Por ejemplo, este tipo de modelos podría en un futuro apoyar tareas de monitoreo e identificación de zonas con cultivos ilícitos, teniendo en cuenta las problemáticas que afronta el país en este tipo de escenarios.

1.5 Objetivos

1.5.1 Objetivo general

Reconocer zonas de cultivo de café a partir de imágenes satelitales utilizando aprendizaje de máquina.

1.5.2 Objetivos específicos

- Evaluar modelos basados en aprendizaje de máquina para el reconocimiento de zonas de cultivo de café, utilizando una herramienta AutoML.
- Evaluar modelos utilizando transferencia de aprendizaje para el reconocimiento de cultivos de café a partir de imágenes satelitales.

1.6 Metodología

La metodología que fue empleada en este trabajo para su seguimiento y realización está basada en la metodología propuesta por IBM denominada “metodología fundamental para la ciencia de datos” [5]. Esta busca proporcionar una estrategia de orientación, que sea independiente de las tecnologías, los volúmenes de datos o los enfoques involucrados. Así mismo, dicha metodología puede articularse con nuevas prácticas en la ciencia de datos, como el uso de grandes volúmenes de datos, la incorporación de la analítica de texto en el modelado predictivo, o el aprendizaje profundo.

La metodología empleada en el desarrollo de este proyecto se presenta en la Figura 1, y las seis fases de la metodología se definen a continuación.



Metodología utilizada en el desarrollo de la investigación

1. **Definición de objetivos y tareas:** La primera fase de esta metodología consiste en entender el problema que se va a abordar, articulado con la forma en que se va a abordar (objetivos y tareas que fueron propuestos). Esto con el fin de determinar los requisitos del problema, e identificar posibles técnicas de aprendizaje automático que puedan ser utilizadas para resolver el problema. La realización de esta primera fase viene dada por los siguientes pasos:

- Determinar los objetivos del proyecto y los criterios de éxito de este.
- Valorar la situación: requisitos, supuestos, restricciones, contingencias y valoración de la problemática a resolver.
- Identificar los objetivos del procesamiento de los datos: metas del procesamiento y criterios de éxito de este.
- Plan del proyecto: valoración de sus enfoques y herramientas necesarias.

2. Documentación del estado del arte: En esta fase se realiza el proceso de levantamiento de información de diversas fuentes, en los temas concernientes a la investigación. Aquí se consideran trabajos que hayan empleado el conjunto de datos *Brazilian coffee scenes*, así como trabajos que hayan realizado clasificación de imágenes satelitales y que hayan empleado técnicas como transferencia de aprendizaje y AutoML.

3. Comprensión y preparación de los datos: En esta fase se establecen todos y cada uno de los pasos a los que serán sometidos el conjunto de datos que son base para la realización del proyecto:

- Comprensión: por medio de la comprensión se hará el primer acercamiento en cuanto a la problemática que se estableció en la fase 1. Se busca hacer la exploración de todo el conjunto de datos obtenidos, entendiendo su relevancia y evaluando la calidad de dichos datos.
- Preparación: durante esta etapa se hace énfasis como su nombre lo indica en hacer la preparación al conjunto de datos. El conjunto de imágenes se divide en tres grupos: entrenamiento, validación, y pruebas externas.

4. Implementación y evaluación de modelos: consiste en realizar el entrenamiento y validación de modelos de clasificación, diseñados a partir de las dos estrategias de diseño, para el conjunto de imágenes correspondiente.

Las actividades en esta fase son:

- Se definen las técnicas y herramientas que se usarán para los diferentes modelos.

- Construcción del modelo con las técnicas determinadas, modificación de los hiperparámetros y descripción del modelo usado en el momento.
- Se genera todo lo que obedece al protocolo con el que se harán las respectivas pruebas del modelo.

5. Análisis de resultados: Esta fase corresponde a todas las acciones que recopilan los resultados, comparaciones y análisis de todo lo obtenido al momento de poner en práctica la fase 4 de la metodología.

6. Documentación: En esta fase se realiza todo el proceso de escritura del trabajo y se reportan los datos más importantes en la ejecución de la investigación mediante un documento, el cual se publicará en el repositorio de la Universidad Militar Nueva Granada. En dicha documentación se evidencian los procesos metodológicos desarrollados, los modelos creados y los resultados obtenidos a lo largo del trabajo investigativo.

1.7 Alcance o delimitación de la investigación

El presente trabajo tuvo como fin el reconocimiento de cultivos de café haciendo uso específicamente de imágenes por teledetección y el uso de la inteligencia artificial, mediante imágenes de satélite de un dataset ya establecido (*Brazilian Coffee Scenes*) [6]. Este es un dataset balanceado que incluye zonas con presencia mayoritaria de cultivos de café (50%) y zonas de no presencia de cultivo de café (50%). El reconocimiento de cultivos se hará mediante aprendizaje de máquina determinando el modelo que mejor se adapte a dicho enfoque y limitando el proyecto a:

- Reconocimiento de zonas específicamente de cultivo de café de 5 zonas brasileras.
- Entrenar y evaluar modelos de aprendizaje automático por medio de imágenes satelitales.
- Utilizar herramientas de aprendizaje automático como lo son la transferencia de aprendizaje y AutoML.

2. Marco teórico y revisión de estado del arte

2.1 Marco teórico

- **Inteligencia Artificial (IA)**

La inteligencia artificial “es una amplia rama de la informática que se ocupa de la construcción de máquinas inteligentes capaces de realizar tareas que normalmente requieren inteligencia humana. La IA es una ciencia interdisciplinaria con múltiples enfoques, donde los avances en el aprendizaje automático y el aprendizaje profundo están creando un cambio de paradigma en prácticamente todos los sectores de la industria tecnológica” [6].

La inteligencia artificial tiene cuatro enfoques:

- Pensar como humano.
- Actuar como humano.
- Pensar racionalmente.
- Actuar racionalmente.

Los dos primeros hacen alusión a los procesos que involucran el pensamiento y el razonamiento, y los últimos dos al comportamiento. Muchos de los sistemas de inteligencia artificial trabajan bien sea con aprendizaje automático, aprendizaje profundo y reglas en general [6].

- **Aprendizaje de Máquina**

El aprendizaje de máquina o aprendizaje automático es una rama de la inteligencia artificial que consiste en construir y entrenar modelos predictivos por medio de unos datos de entrada denominados Dataset, y a partir de estos modelos poder realizar predicciones con datos nuevos y desconocidos, pero con características como las utilizadas para realizar el entrenamiento. Una definición alternativa habla de “una habilidad indispensable para hacer sistemas capaces de identificar patrones entre los datos para hacer predicciones. Esta tecnología está presente en un sinnúmero de aplicaciones como las recomendaciones de Netflix o Spotify, las respuestas inteligentes de Gmail o asistentes como Siri y Alexa” [7].

El *machine learning*, como se le conoce en inglés, se caracteriza por utilizar datos de ejemplo en un programa informático que le permita realizar inferencias de nuevas muestras de datos que no han sido utilizadas previamente. Esta capacidad de aprendizaje se emplea en diferentes aspectos de gran relevancia hoy en día, involucrando una serie de algoritmos capaces de analizar grandes cantidades de datos para deducir cuál es el resultado óptimo para un determinado problema [7].

- **Visión por computador**

La visión por computador es una disciplina científica en la que se emplean métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real, esto con el fin de generar información numérica o simbólica para que los datos obtenidos puedan ser tratados por un ordenador de manera óptima [8].

Similar a la capacidad de los seres humanos para comprender los entornos por medio del uso de la visión y el cerebro, así mismo se pretende que suceda un efecto similar en cuanto a los ordenadores. Para ello y muy de la mano con el aprendizaje de máquina se emplean herramientas geométricas, estadísticas y demás para el cumplimiento de esta función. Aquí, *“la adquisición de los datos se consigue por varios medios como secuencias de imágenes, vistas desde varias cámaras de video o datos multidimensionales desde un escáner médico, imágenes satelitales, aéreas y de diversas formas”* [8].

- **Deep learning (DL)**

El DL es una técnica que lleva a cabo el *“proceso de aprendizaje automático usando una red neuronal artificial que se compone de un número de niveles jerárquicos. En el nivel inicial de la jerarquía la red aprende algo simple y luego envía esta información al siguiente nivel. El siguiente nivel toma esta información sencilla, la combina, compone una información un poco más compleja, y se lo pasa al tercer nivel, y así sucesivamente”* [9].

Esta técnica es de gran utilidad para la implementación de modelos con el fin de predecir, aplicar y descubrir diferentes tipos de patrones en casos de la vida real y con un amplio volumen de datos [9].

- **Datos**

En el machine learning los datos son el recurso primordial que alimenta los diversos algoritmos, permitiendo a partir de estos tomar decisiones respondiendo a comportamientos encontrados en estos datos, un dato o conjunto de datos que no estén procesados no representan ningún tipo de valor, es por esto por lo que deben someterse a un análisis científico, volviéndolos útiles, rentables y construyendo valor. [10]

Para el caso de esta investigación en particular los datos corresponden a un conjunto de imágenes, una imagen esta compuesta por una serie de valores numéricos denominados píxeles, que son los utilizados por los algoritmos para determinar las características comunes de a estos datos y llegar a predecir situaciones específicas que el usuario en el análisis del problema desea resolver a través de ellos.

- **Red neuronal convolucional (CNN)**

Una red neuronal convolucional *“es un tipo de red neuronal artificial donde las neuronas corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria, consisten en múltiples capas de filtros convolucionales de una o más dimensiones. Después de cada capa, por lo general se añade una función para realizar un mapeo”* [11].

Como cualquier red empleada para clasificación, al principio estas redes tienen una fase de extracción de características, compuesta de neuronas convolucionales, luego hay una reducción por muestreo y al final tendremos neuronas tipo perceptrón más sencillas para realizar la clasificación final sobre las características previamente extraídas [11].

En una red neuronal convolucional en gran parte de los casos se tiene la combinación entre: capas convolucionales, capas pooling y capas densas.

- **Capas convolucionales**

Las capas convolucionales son la capa fundamental para la construcción de una CNN, contiene un conjunto de filtros configurables en los cuales sus parámetros deben aprenderse cuando el modelo esté en entrenamiento, generalmente dichos

filtros deben ser de escala mas pequeña que el tamaño de la imagen del conjunto de datos, cada uno de estos se encarga de hacer una convolución con la imagen generando un mapeo y realizando una operación de producto escalar entre cada elemento del filtro y de la entrada. A modo de ejemplo se tiene la figura 2 que muestra un proceso de convolución, para la generación del mapa se empieza calculando la convolución entre el filtro con la parte marcada en color azul y repitiendo este proceso para cada uno de los valores de los pixeles de la imagen entrante, cada filtro configurado representa un mapa de activación que formara un conjunto con el resto de los mapas obtenidos de los demás filtros. [12]

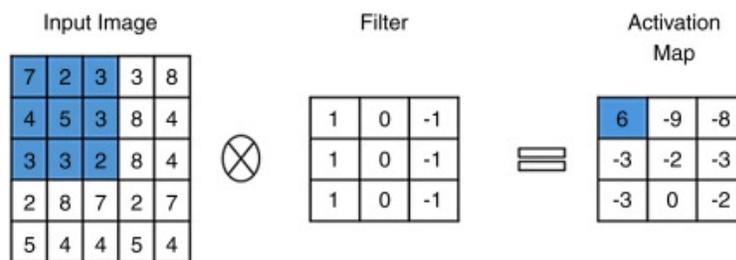


Figura 1. Proceso de convolución de capas convolucionales. [12]

- **Capas pooling**

La función principal de las capas pooling es la reducción de las dimensiones de los mapas de características, función que permite al modelo reducir la cantidad de parámetros que debe aprender y por tanto el recurso computacional también disminuye, las capas pooling resumen las características que existen en una región del mapa de características creado en las capas convolucionales, generando operaciones con características resumidas brindando mayor disposición a la variación en las características de imágenes de entrada. [13]

- **Hiperparámetros**

Para el caso de un modelo de machine learning los hiperparámetros son aquellos valores que un algoritmo por si solo no es capaz de aprender y deben ser configurados y ajustados para controlar el proceso de entrenamiento del modelo.

Es aquí donde el ajuste de hiperparámetros brinda una ayuda importante al resultado y rendimiento del modelo ya que en gran medida se depende de dichos valores, la optimización de hiperparámetros es el proceso en el que se pretende encontrar la configuración de hiperparámetros que produzcan el mejor rendimiento, suele ser un proceso manual y costoso computacionalmente. [14]

- **Algoritmos de optimización**

De forma general se puede definir a los algoritmos de optimización como los métodos robustos que se emplean para el diseño y resolución de problemas complejos con elevado número de elementos, restricciones y variables y en los que, por lo general, coexisten una o varias soluciones óptimas que no pueden aproximarse por los métodos tradicionales propios de la programación lineal [15].

- **Tasa de aprendizaje**

La tasa de aprendizaje representa un hiperparámetro de gran importancia, es un valor que indica qué tanto desplazamiento deben tener los parámetros para alcanzar el error mínimo en el entrenamiento de los modelos, es decir, la tasa de aprendizaje hace referencia a la longitud de un paso a otro, al querer dar pasos de gran tamaño se puede llegar al punto de menor error de manera más rápida pero existe el riesgo que al dar pasos tan amplios dicho error se pase por alto, mientras que al buscar dar pasos pequeños el error muy seguramente se encuentre pero también genera que el modelo demore mucho más tiempo en encontrarlo [16].

- **Dropout**

Este tipo de capa busca eliminar cierto número de neuronas en la fase de entrenamiento todo esto con el fin de evitar problemas de sobreajuste en los modelos, es decir, cuando se tienen capas fully connected cada una de las neuronas de la red se vuelve dependiente de las otras quitando la posibilidad de potenciar el poder de cada neurona de forma individual dejando el modelo muy ajustado a un solo tipo de red, esto se evita por medio del dropout.

- **Arquitecturas**

Las arquitecturas CNN son aquellas conformaciones de redes CNN previamente establecidas por equipos de investigación con el fin de generar los mejores resultados de precisión de un modelo, basados generalmente en el dataset ImageNet que es tal vez el dataset más popular con alrededor de 14 millones de imágenes y 22 mil clases [17], entre las arquitecturas más importantes y que se aplican a esta investigación se encuentran: VGG, Resnet, Xception.

- **Transferencia de aprendizaje**

La transferencia de aprendizaje es un enfoque del aprendizaje de máquina que se centra en el aprovechamiento de un aprendizaje previamente adquirido en la resolución de un problema, para posteriormente aplicarlo a un problema distinto, pero del que guarda relación. Por ejemplo, el conocimiento adquirido al aprender a reconocer cultivos de café podría aplicarse al reconocimiento de cultivos de coca [18].

Otra forma de definir esta técnica es la forma de transferir información y conocimiento de una tarea simple y/o con más datos a una tarea compleja con menos datos de aprendizaje. Cuando se involucran elementos de aprendizaje automático en la mayor proporción de los casos se encargan de la resolución de una sola tarea, a diferencia de transferencia de aprendizaje en donde un solo programa puede dedicarse a resolver multiplicidad de tareas [19].

Diferentes experimentos realizados han demostrado una mejora en *“la precisión de las redes neuronales y las redes neuronales convolucionales a través del aprendizaje por transferencia tanto en la primera época (antes de cualquier aprendizaje, es decir, en comparación con la inicialización de pesos aleatoria estándar) como en la asíntota (el final del proceso de aprendizaje)”* [20]. Es decir, los algoritmos se mejoran con la exposición a otro dominio. Además, el usuario final de un modelo previamente entrenado puede cambiar la estructura de capas completamente conectadas para lograr un rendimiento superior.

- **AutoML**

AutoML es un proceso que consiste en automatizar las tareas al aplicar ML a problemas del mundo real, en diferentes fase del proceso, dado que “ *cubre la canalización completa desde el conjunto de datos sin procesar hasta el modelo de aprendizaje automático implementable*” [21]. De aquí que este tipo de soluciones facilita la aplicación de ML, incluso si el usuario no las domina en profundidad, y permitiendo desarrollos en un menor tiempo [21].

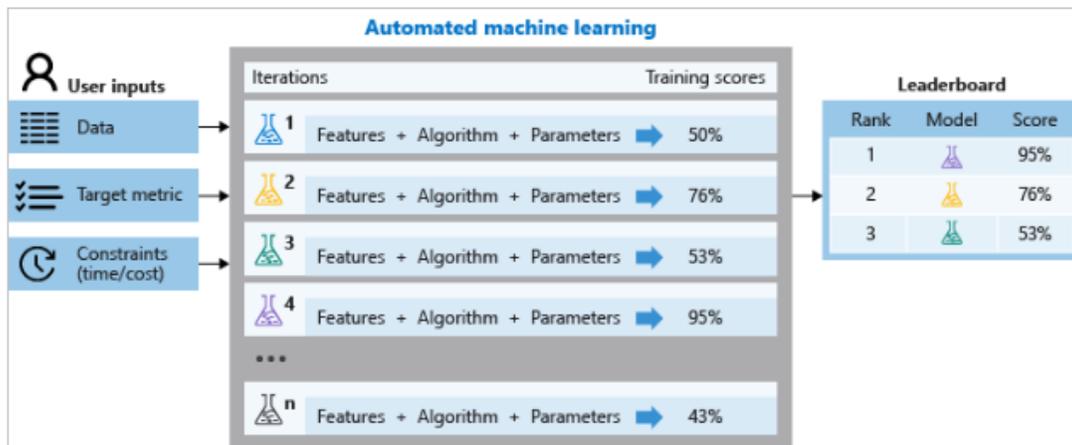


Figura 2. Funcionamiento AutoML. [22]

Como se aprecia en la figura 2 el funcionamiento de la estrategia de diseño AutoML, consta de un proceso de intervención del usuario que supervisa y configura los parámetros del modelo, un proceso en el que el modelo reúne los diferentes parámetros y los ejecuta, y el último proceso en el que se determinan los mejores resultados obtenidos; en el primer proceso se determina el problema que se quiere abordar, para el caso de esta investigación es un problema de clasificación binaria, a partir de esta identificación se determina lo necesario para llevar a cabo el AutoML, los datos que son la base fundamental del modelo, así como los parámetros que se tendrán en cuenta para que el modelo ejecute de forma aleatoria obteniendo resultados de la mayor cantidad posible de combinación de parámetros determinando cual de estas es mejor, aquí es importante el tercer proceso en el que se recopilan los modelos con los parámetros que mejores resultados de entrenamiento arrojan, es por esto que AutoML es una estrategia muy apropiada para reducir tiempo y recurso computacional para el usuario que desea dar solución a un problema por medio del aprendizaje de máquina [22].

- **Métricas de clasificación**

Al hablar de métricas de clasificación se habla de aquellas medidas que permiten asegurar que un modelo cumple con las condiciones y rendimiento esperados en sus procesos de entrenamiento y validación, y que al someterlo a evaluación los valores que se obtienen son los esperados.

- **Matriz de confusión**

Corresponde a una tabla descriptiva del rendimiento de un modelo en su fase de testeo, esta consta de 4 valores representativos que muestran donde el modelo está confundiendo su clasificación. La figura 3 representa de manera gráfica como está diseñada una matriz de confusión.

1. **“True Positives (TP):** cuando la clase real del punto de datos era 1 (Verdadero) y la predicha es también 1 (Verdadero)
2. **Verdaderos Negativos (TN):** cuando la clase real del punto de datos fue 0 (Falso) y el pronosticado también es 0 (Falso).
3. **False Positives (FP):** cuando la clase real del punto de datos era 0 (False) y el pronosticado es 1 (True).
4. **False Negatives (FN):** Cuando la clase real del punto de datos era 1 (Verdadero) y el valor predicho es 0 (Falso).” [23]

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figura 3. Ejemplo esquema de matriz de confusión.

- **Accuracy**

Es una métrica utilizada de acuerdo con los valores obtenidos por el rendimiento del modelo, se refiere a la cantidad de aciertos en la clasificación que tiene el modelo de acuerdo con la clase en base al total de clasificaciones realizadas y se deriva de la matriz de confusión, está dada a partir de la siguiente ecuación.

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

- **Precisión**

Al igual que la métrica anterior permite obtener valores de acuerdo con el rendimiento del modelo y este valor se enfoca en considerar la clase positiva, determina de todos los valores que en predicción fueron positivos cuantos de ellos realmente lo son, viene dada por la siguiente ecuación.

$$Precisión = \frac{TP}{TP + FP}$$

2.2 Documentación del estado del arte

Las temáticas relacionadas con la presente investigación están directamente relacionadas con el uso de *datasets* de reconocimiento de zonas de café, clasificación de imágenes satelitales, aplicación de técnicas de transferencia de aprendizaje y AutoML. De aquí que, el análisis del estado del arte se presenta agrupado en dos enfoques principales: el primero hace referencia a ciertas investigaciones que hicieron uso del dataset *Brazilian Coffee Scene*, el cual será utilizado para la realización del trabajo de grado; mientras que, el segundo enfoque se centra en investigaciones que usaron como temática la clasificación de escenas de imágenes satelitales haciendo uso de *deep learning*, transferencia de aprendizaje o AutoML.

2.2.1 Trabajos que utilizan el conjunto de datos *Brazilian Coffee Scenes*

En esta sección se involucran una serie de investigaciones en las que se evidencia el trabajo realizado y aplicado en el uso del *dataset Brazilian Coffee Scenes*, sus temáticas principales, conclusiones y resultados arrojados en cada una de ellas:

- **Transferencia de aprendizaje con Res2Net para clasificación de escenas de percepción remota**

<i>Transfer Learning with Res2Net for Remote Sensing Scene Classification</i>	
Referencia	Das, A., & Chandran, S. (2021). Transfer learning with RES2Net for remote sensing scene classification. Paper presented at the Proceedings of the Confluence 2021: 11th International Conference on Cloud Computing, Data Science and Engineering [24].
Tema	Transferencia de aprendizaje, clasificación de imágenes remotas
Palabras clave	Classification (of information); Data Science; Deep learning; Image segmentation; Transfer learning, Brazilian coffees; Classification accuracy; Learning architectures; Multi-scale features; Novel architecture; Region of interest.
Propósito	Hacer uso de la transferencia de aprendizaje para la clasificación de imágenes satelitales todo esto por medio de la evaluación de hiperparámetros y la implementación de la arquitectura Res2Net, utilizando 3 conjuntos de datos entre ellos <i>Brazilian Coffee Scene</i> .
Conclusiones	Para la realización de la investigación se utilizó la arquitectura Res2Net que maneja varias capas convolucionales, se evaluaron dos hiperparámetros que corresponden a la escala y profundidad, variando dichos valores para determinar los más efectivos junto con el Res2Net-50 obteniendo una precisión de 93.25% clasificando el conjunto de datos <i>Brazilian Coffee Scene</i> .
Desarrollo de la investigación	Implementación de la arquitectura Res2Net que según el artículo funciona como una columna vertebral para transferir los pesos entrenados y extraer las funciones por medio de bloques obteniendo las características de las imágenes de entrada, utilizando como hiperparámetros principales W, H y C que corresponden a anchura altura y profundidad. Para realizar la transferencia de aprendizaje utiliza el extractor de funciones de Res2Net-50 que se inicializa con pesos ya entrenados, aumentando la precisión de la clasificación. Asimismo, se usaron dos técnicas: una en la que se emplea un congelamiento en el extractor de características, entrenando solo los pesos actuales y representando menos tiempo y recursos de computación y en la segunda técnica se entrena todo el modelo con los pesos ya anteriormente entrenados.
Hiperparámetros utilizados	Optimizador Adam, tasa de aprendizaje de 0.0001, 60 épocas y una pérdida de entropía y el uso de la técnica one-hot, aplicando técnicas en el conjunto de datos para aumentarlo.

Tabla 1. Estado del arte: investigaciones con el dataset brazilian coffee scenes (Transferencia de aprendizaje con Res2Net para clasificación de escenas de percepción remota)

Resultados que obtuvo la investigación:

Valores definidos para la escala y profundidad:

	Depth (w)	Scale (s)
A	14	8
B	26	4
C	26	6
D	26	8
E	48	2

Tabla 2. Parámetros evaluados en [24]

Exactitud (*accuracy*) del dataset *Brazilian Coffee Scene* empleando las técnicas mencionadas en el desarrollo:

Combination	Technique 1	Technique 2
A	75.62	93.25
B	78.76	92.99
C	77.23	92.36
D	80.75	92.96
E	76.77	92.75

Tabla 3. Precisión del modelo implementado en [24].

Comparación con arquitecturas distintas a la empleada en el artículo:

Method	Accuracy
VGG16 [41]	85.36
OverFeat _S + Caffe [29]	79.01
OverFeat _L + Caffe [29]	79.15
OverFeat _S + OverFeat _L [29]	83.04
D-DSML-CaffeNet [42]	91.13
Proposed Method	93.25

Tabla 4. Arquitecturas utilizadas en los modelos implementados en [24]

Ejemplo del uso de Matrices de confusión para evidenciar el resultado arrojado para la carpeta 1 del dataset:



Figura 2. Matriz de confusión obtenida en el modelo implementado en [24]

- **Clasificación de imágenes satelitales multi escala utilizando un enfoque de aprendizaje profundo.**

Multiscale satellite image classification using deep learning approach	
Referencia	Laban, N., Abdellatif, B., Ebied, H. M., Shedeed, H. A., & Tolba, M. F. (2020). Multiscale satellite image classification using deep learning approach doi:10.1007/978-3-030-20212-5_9 [25].
Tema	Clasificación de imágenes satelitales multiescala, redes neuronales convolucionales.
Propósito	La investigación propone un modelo con técnicas mejoradas en cuanto a redes convolucionales y la clasificación de imágenes de satélite, todo esto por medio de la modificación de la escala que presentan las imágenes originales de los conjuntos de datos y en este caso se utiliza el dataset <i>Brazilian Coffee Scene</i> , procurando la mejora del rendimiento de los modelos.
Conclusiones	El artículo demuestra que al utilizar imágenes con una modificación mediante el uso de un método que minimiza la escala de la imagen lo máximo posible, esta obtiene como resultado una mayor precisión, demostrando que las técnicas empleadas mejoran el rendimiento de la clasificación al hacer modificación de escalas en los datasets y afirma que se obtienen resultados óptimos en comparación con el uso de capas anteriores.
Desarrollo de la investigación	La investigación se basa en el uso de tensorflow y Keras para implementar un modelo que utiliza 3 capas convolucionales; la primera aplica 32 filtros 7x7 a la capa de entrada y una función de activación ReLU, la segunda y tercera capa utilizan 64 filtros de 3x3. Luego de estas capas, se modifica la escala de las imágenes para determinar la eficacia de los modelos, empezando desde una escala de 8x8 píxeles hasta 64x64. Aquí se determina una relación entre la escala y el costo computacional, así como los tiempos de entrenamiento.
Hiper-parámetros utilizados	3 capas convolucionales una de 32 filtros 7x7, el resto de 64 filtros 3x3, 3 capas de pooling, una capa de dropout, 2 capas <i>fully connected</i> , la última con función de activación <i>softmax</i> , una función de pérdida <i>crossentropy</i> , 100 épocas y tasa de aprendizaje de 0.001.

Tabla 5. Estado del arte: investigaciones con el dataset brazilian coffee scenes (Clasificación de imágenes satelitales multi escala utilizando un enfoque de aprendizaje profundo.)

Resultados.

La siguiente figura muestra la arquitectura de red propuesta para la clasificación:

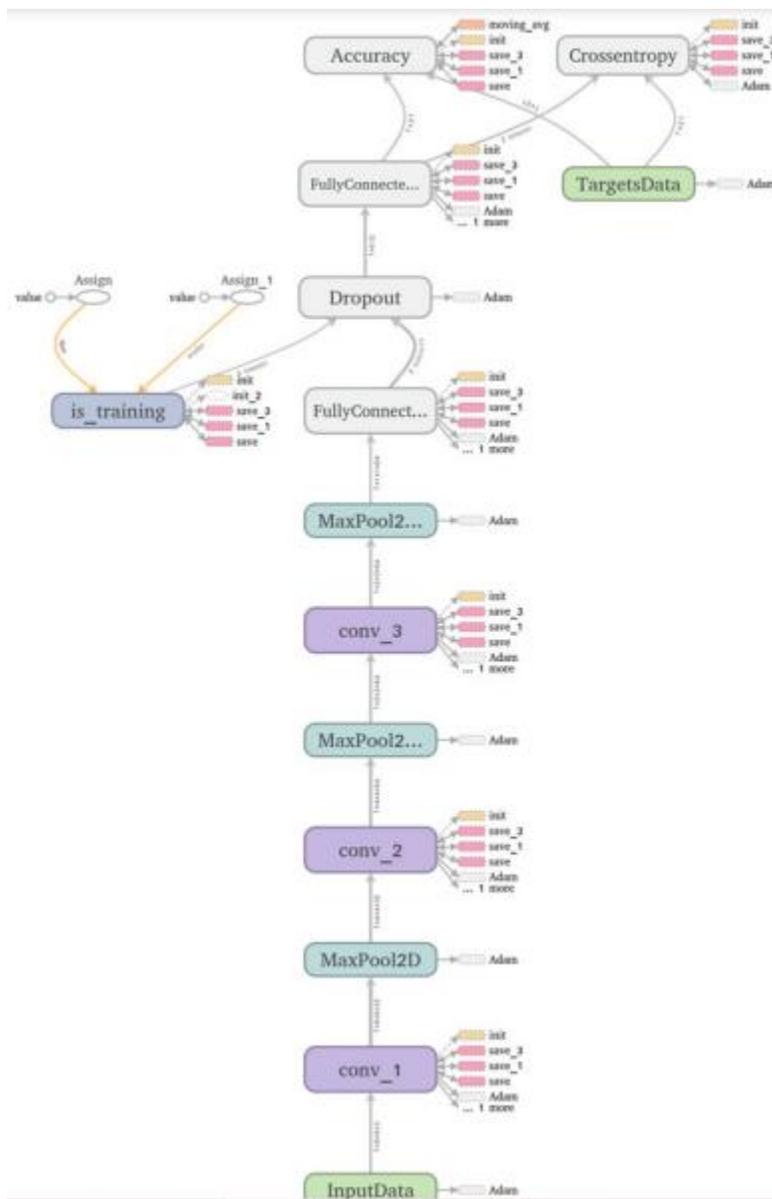


Figura 3. Esquema de la arquitectura total del modelo implementado en [25]

Las escalas utilizadas en el desarrollo de la investigación se muestran a continuación:

Brazilian Coffee Scenes Dataset	64×64	8×8	16×16	32×32	64×64
---------------------------------	-------	-----	-------	-------	-------

Tabla 6. Parámetros importantes evaluados en el desarrollo del [25]

Finalmente, los resultados de precisión por escala se relacionan en la siguiente tabla y la subsecuente gráfica.

Dataset	Scale 1		Scale 2		Scale 3		Scale 4	
	After 100 time unit	After 100 epoch	After 100 time unit	After 100 epoch	After 100 time unit	After 100 epoch	After 100 time unit	After 100 epoch
UCMerced Land-use Dataset	18.8	70.8	91.5	99.4	89.4	96.6	88.4	98
WHU-RS Dataset	60.7	89.7	64.9	95.9	56.4	97.7	39.7	97.4
Brazilian Coffee Scenes Dataset	78.9	94.6	88.7	96.8	88.3	98	82.6	99.3

Tabla 7. Resultados de *accuracy* obtenidos en los modelos de [25]

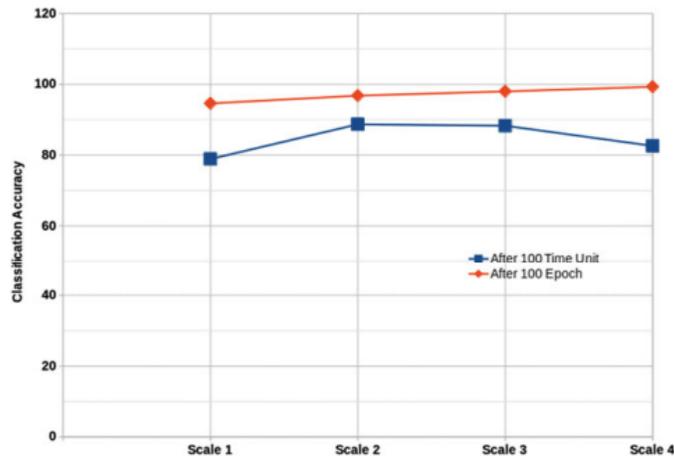


Figura 4. Análisis grafico de los resultados obtenidos en [25]

2.2.2 Trabajos relacionados con clasificación de imágenes satelitales.

En esta segunda sección se recopilan investigaciones enfocadas en la clasificación de imágenes satelitales haciendo uso de diversas técnicas obteniendo información clave de sus procesos y resultados.

- **Clasificación de múltiples etiquetas de imágenes satelitales de teledetección.**

<i>Multilabel classification of remote sensed satellite imagery</i>	
Referencia	Kumar, A., Abhishek, K., Kumar Singh, A., Nerurkar, P., Chandane, M., Bhirud, S., . . . Busnel, Y. (2021). Multilabel classification of remote sensed satellite imagery. Transactions on Emerging Telecommunications Technologies [26].
Tema	Clasificación de imágenes satelitales, Aplicación de Deep learning

Palabras clave	Classification models; Convolutional networks; Critical researches; Feature engineerings; Multi-label classifications; Performance metrics; Scene classification; Very high resolution
Propósito	Realizar la clasificación de imágenes de origen satelital que contienen múltiples clases y evaluar los modelos entrenados con diferentes arquitecturas e hiperparámetros, por medio del dataset UC Merced con imágenes en alta definición, y buscando una optimización de los resultados involucrando menos épocas en su composición
Conclusiones	Fueron empleadas 15 arquitecturas de aprendizaje profundo para la evaluación de modelos, con base en técnicas de transferencia de aprendizaje para manejo de grandes volúmenes de datos y uso de ajuste fino. Este estudio arrojó que Inception V3 obtuvo los mejores resultados con el 85% de precisión, mientras que la arquitectura con el peor rendimiento fue ResNet con un 67.21% de precisión
Desarrollo de la investigación	Para el desarrollo de los distintos modelos se utilizó un total de 15 arquitecturas, todas con iguales condiciones de hiperparámetros. El objetivo fue evaluar el rendimiento de los modelos respecto a la clasificación de las 21 clases del conjunto de datos. Se emplearon de igual forma arquitecturas previamente entrenadas para manejar un volumen de datos menor y facilitar la resolución de las temáticas. De igual forma se probaron los diferentes modelos para obtener la cantidad de falsos positivos y negativos de las arquitecturas
Hiper-parámetros utilizados	Xception, VGG 16, VGG 19, 5 arquitecturas de ResNet, InceptionV3, MobileNet y DenseNet con variaciones, además de optimizador RMSprop, un Dropout de 0.4, métrica <i>accuracy</i> , función de pérdida binary cross entropy, y capas adicionales de Average pooling

Tabla 8. Estado del arte: investigaciones empleando imágenes satelitales (Clasificación de múltiples etiquetas de imágenes satelitales de teledetección.)

Resultados Obtenidos:

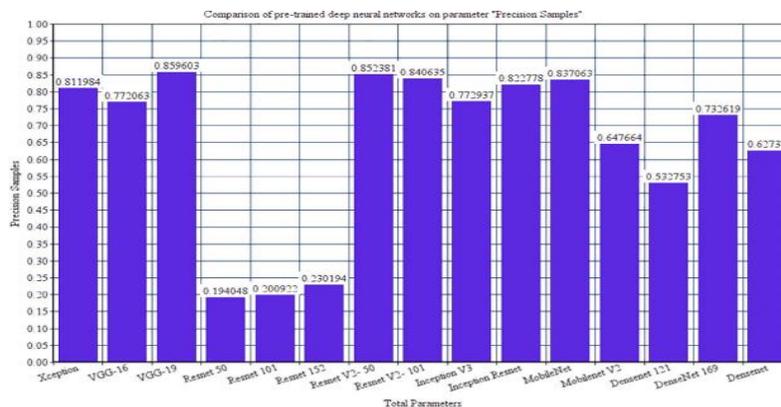


Figura 5. Gráfico de comparación de arquitecturas usadas en [26]

Resultados de las variables aplicadas y la precisión de las arquitecturas evaluadas:

Model	Macro-precision	Precision samples	Macro-recall	Recall samples	Macro-F ₁	F ₁ samples	Total parameters	Nontrainable	Trainable	Threshold
Xception	0.88	0.81	0.7	0.76	0.75	0.76	20896313	20861480	34833	0.1
VGG16	0.76	0.77	0.74	0.85	0.72	0.79	14723409	14714688	8721	0.1
VGG19	0.8	0.86	0.55	0.65	0.61	0.72	20033105	20024384	8721	0.18
ResNet50	0.19	0.19	0.98	0.98	0.28	0.31	23622545	23587712	34833	0.001
ResNet101	0.19	0.2	0.9	0.96	0.27	0.32	42693009	42658176	34833	0.001
ResNet152	0.18	0.23	0.73	0.88	0.25	0.35	58405777	58370944	34833	0.001
ResNet50V2	0.92	0.85	0.58	0.63	0.66	0.7	23599633	23564800	34833	0.15
ResNet101V2	0.9	0.84	0.62	0.69	0.69	0.73	42661393	42626560	34833	0.1
InceptionV3	0.85	0.77	0.75	0.84	0.75	0.77	21837617	21802784	34833	0.05
InceptionResNet	0.87	0.82	0.69	0.73	0.72	0.74	54362865	54336736	26129	0.08
MobileNet	0.81	0.84	0.69	0.77	0.71	0.78	3246289	3228864	17425	0.1
MobileNetV2	0.72	0.65	0.67	0.81	0.61	0.68	2279761	2257984	21777	0.05
DenseNet121	0.59	0.53	0.85	0.95	0.64	0.65	7054929	7037504	17425	0.04
DenseNet169	0.81	0.73	0.63	0.74	0.63	0.71	12671185	12642880	28305	0.1
DenseNet201	0.75	0.63	0.72	0.87	0.63	0.70	18354641	18321984	32657	0.05

Tabla 9. Parámetros y arquitecturas evaluadas en [26]

- **Teledetección para la monitorización de plantas solares fotovoltaicas en Brasil mediante segmentación semántica profunda.**

Remote sensing for monitoring photovoltaic solar plants in brazil using deep semantic segmentation	
Referencia	Chen, J., Huang, H., Peng, J., Zhu, J., Chen, L., Tao, C., & Li, H. (2021). Contextual information-preserved architecture learning for remote-sensing scene classification. IEEE Transactions on Geoscience and Remote Sensing, doi:10.1109/TGRS.2021.3063287 [27]
Tema	Transferencia de aprendizaje, Redes neuronales convolucionales Imágenes de teledetección
Palabras clave	Deep learning, semantic segmentation, solar panel, remote sensing, image segmentation, overall accuracies, overlapping pixels
Propósito	Esta investigación tuvo como propósito identificar y clasificar plantas de energía solar con el fin de llevar el control de estos dispositivos, con base en clasificación de imágenes satelitales, y usando diversas técnicas del aprendizaje profundo donde se evaluaron los mejores modelos
Conclusiones	Se compararon diversos modelos de DL para la clasificación de imágenes de plantas solares comparando específicamente cuatro arquitecturas. Se implementó transferencia de aprendizaje, y se determinó que la arquitectura U-net presentó el mejor desempeño junto con la utilización de un Backbone Eff-b7. Se obtuvo un <i>accuracy</i> de 98%, pero el costo computacional que representa el trabajo fue desafiante
Desarrollo de la investigación	Se evaluaron las 4 arquitecturas que fueron utilizadas para el modelado junto con los <i>backbone</i> . Luego de un ajuste, se logró contrarrestar la baja cantidad de imágenes y su resolución. También se utilizó la función de pérdida <i>Dice loss</i> y se aplicó transferencia de aprendizaje
Hiper-parámetros utilizados	Se utilizaron cuatro tipos de arquitecturas, a la vez que se consideraron parámetros como: backbone, accuracy, IoU, F1-score and Epoch Period. En el caso de la arquitectura U-net se usaron 4 backbones efficient-net-b7 - net-b0 y ResNet 101 - 50. De igual forma se utilizaron 300 épocas, optimizador Adam, tasa de aprendizaje de 0.005 y un tamaño del lote de 5

Tabla 10. Estado del arte: investigaciones empleando imágenes satelitales (Teledetección para la monitorización de plantas solares fotovoltaicas en Brasil mediante segmentación semántica profunda.)

Resultados obtenidos: precisión de acuerdo con las arquitecturas y backbones evaluados:

Architecture	Backbone	Accuracy (%)	IoU (%)	F-Score (%)	Epoch Period (s)
U-net	Eff-b7	98.08	91.17	95.38	12
	Eff-b0	98.05	90.97	95.27	5
	R-101	97.96	90.58	95.06	5
	R-50	97.98	90.70	95.12	4
DeepLabv3+	Eff-b7	97.83	89.98	94.73	26
	Eff-b0	97.77	89.82	94.64	5
	R-101	97.46	88.47	93.88	7
	R-50	97.02	86.63	92.84	6
PSPNet	Eff-b7	97.35	88.03	93.64	5
	Eff-b0	96.73	85.43	92.14	3
	R-101	97.06	86.98	93.04	3
	R-50	97.23	87.60	93.39	3
FPN	Eff-b7	97.38	87.99	93.61	12
	Eff-b0	97.45	88.21	93.73	5
	R-101	97.58	89.21	94.30	6
	R-50	97.25	87.74	93.47	5

Tabla 11. Resultados en *accuracy* de los modelos implementados en [27]

3. Comprensión y preparación de los datos

En esta sección se presentan las características de los datos a utilizar, la distribución de estos y las técnicas aplicadas a los datos que facilitan la generalización de los algoritmos.

3.1 Conjunto de datos Brazilian Coffee Scenes

Brazilian Coffee Scenes es un *dataset* a base de una composición de escenas (imágenes) de zonas brasileras con enfoque en cultivos de café. Fue creado por Rubens Lamparelli a partir de imágenes del sensor SPOT 5 en el año 2005, y actualmente es de acceso libre para su utilización y manipulación, o para investigación, como es el caso del presente trabajo [3].

Brazilian Coffee Scenes cuenta con un total de 2876 imágenes de 64x64 píxeles en las bandas rojo, verde e infrarrojo de 4 diferentes zonas brasileras: Arceburgo, Guaranesia, Guaxupé y Monte Santo como se aprecia en las figuras 6 y 7. Al contener imágenes tomadas en regiones montañosas, el conjunto de datos recopila escenas con diferentes edades de plantas de café y/o con distorsiones espectrales causadas por sombras. Está conformado específicamente por dos diferentes clases de imágenes: la clase Café y la clase no café, cada una de las cuales cuentan con el 50% de la distribución total de imágenes del dataset es decir 1438 imágenes para cada clase [3].

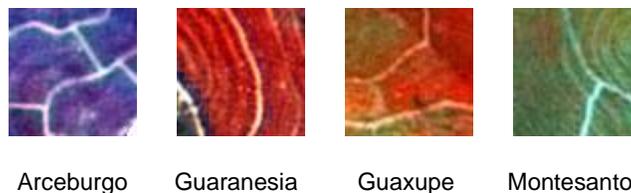


Figura 6. Escenas de la clase café discriminadas por zonas.

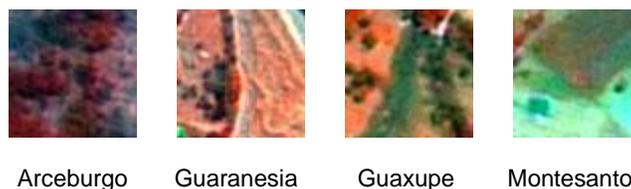


Figura 7. Escenas de la clase no café discriminadas por zonas.

En la construcción del *dataset*, la determinación de si una imagen pertenece a una u otra clase siguió el parámetro de clasificarlas según el porcentaje de píxeles que cuentan con zonas de cultivo de café, por lo que para que una imagen perteneciera a la clase café al menos el 80% de sus píxeles requería contener características de zonas de cultivo de café, y para considerarla clase no café menos del 10%.

Teniendo en cuenta lo anterior, para este trabajo el *dataset* se organizó en tres carpetas distintas, así: entrenamiento, validación y test. Esta distribución involucró un 70% de imágenes para entrenamiento, que se utilizarán para el aprendizaje del modelo y obtención de características por parte del algoritmo; 15% para validación, el objetivo de esta carpeta es tanto para monitorear el funcionamiento del modelo sino para realizar una correcta selección de hiperparámetros que permitan mejorar significativamente los mismos; y por último, otro folder con el 15% de las imágenes para pruebas en la que se disponen las que servirán como testeo del modelo final para determinar su rendimiento con imágenes que no ha conocido previamente y evaluar de forma apropiada los modelos. A continuación, se muestra el diagrama de la distribución del *dataset* para este trabajo:

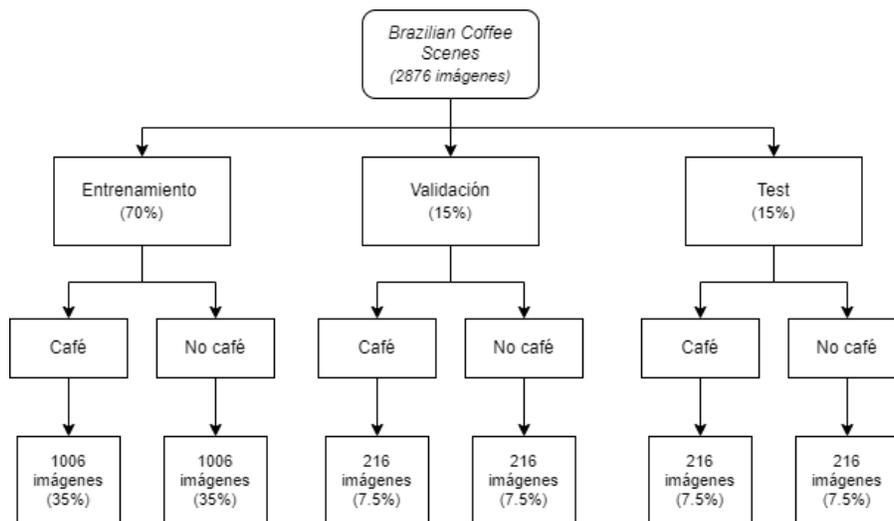


Figura 8. Distribución de imágenes del dataset

Dado que el *dataset* está conformado por imágenes de diversas zonas cafeteras de Brasil con características variadas, fue necesario incluir en cada una de las carpetas mencionadas imágenes de todas las zonas, esto con el fin de que, al realizar el

entrenamiento, validación o prueba de los modelos se tuviera la variación de características que presentan las diferentes zonas del *dataset*.

3.2 Aplicación de aumento de datos

Por medio del uso de técnicas de aumento de datos es posible incrementar la variabilidad del *dataset*, en términos, por ejemplo, de nivel de brillo de la imagen, ubicación de la zona de interés, entre otras. Estas nuevas características le permiten al algoritmo aprender de manera amplia y variada a partir del conjunto de datos, buscando una mejor generalización de este. Sin embargo, del conjunto de transformaciones disponibles para aumento de datos en imágenes, no todas son apropiadas para todos los problemas. En este sentido, se evaluaron las diferentes operaciones de transformación de datos disponibles en Keras/Tensorflow seleccionando las siguientes operaciones [28] [29]:

- **Rotation range:** esta operación genera una rotación de la imagen nueva con respecto a la imagen original en función del valor numérico que se le asigne (representado en grados, para el caso específico se utilizó un valor de 40 grados). Algunos ejemplos de la rotación realizada se muestran en la figura 9.

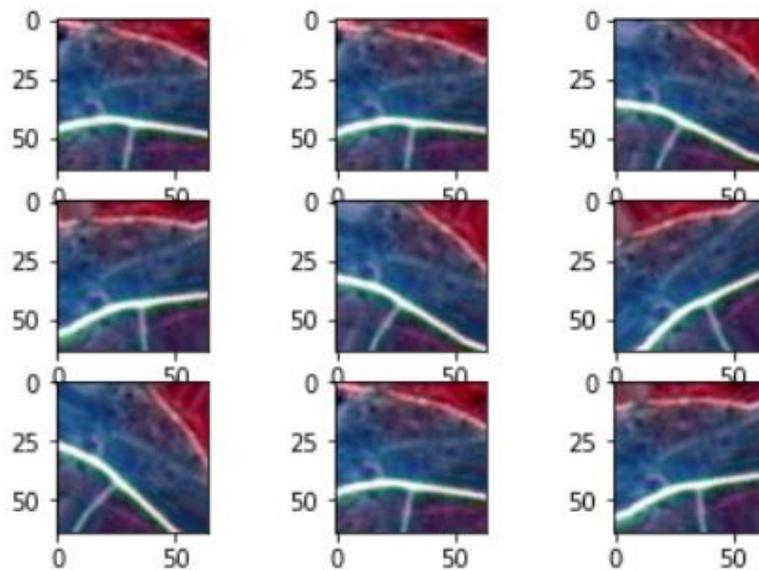


Figura 9. Aplicación de *rotation range* en aumento de datos.

- **Zoom range:** esta técnica de aumento de datos busca generar un zoom a la imagen original en un porcentaje determinado que resalte las características de los pixeles en un área determinada de la imagen. Para el caso de un zoom del 30% se obtiene el siguiente resultado:

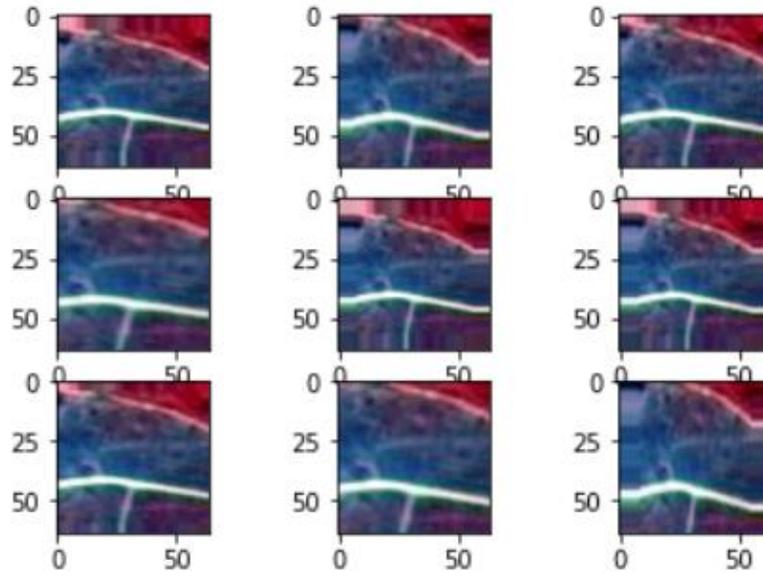


Figura 10. Aplicación de zoom range en aumento de datos.

- **Height shift range y width shift range:** la aplicación de estas técnicas busca generar un desplazamiento en la imagen original, tanto vertical como horizontalmente (ver ejemplos en Figuras 11 y 12).

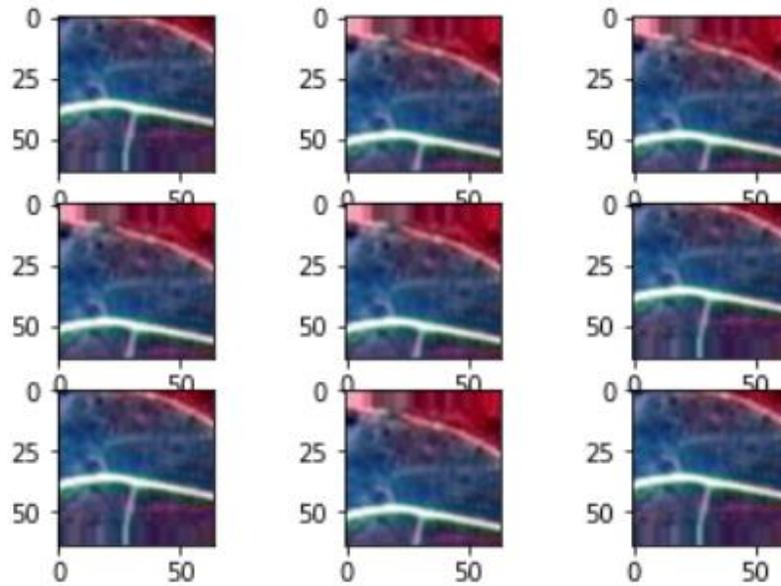


Figura 11. Aplicación de height shift range en aumento de datos.

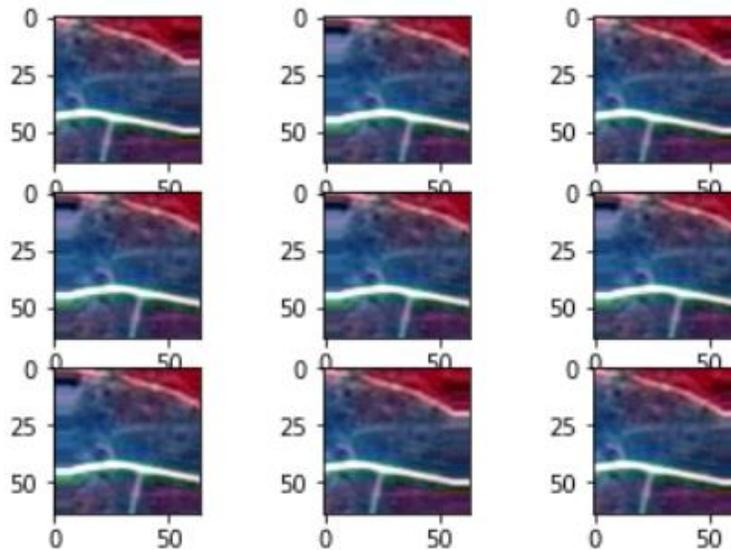


Figura 12. Aplicación de width shift range en aumento de datos.

- **Vertical flip y horizontal flip:** El efecto del flip es generar en la imagen un reflejo vertical u horizontal, es decir, las características presentes de la imagen original que se encuentren en la parte superior se ubicarán en la parte inferior en la nueva imagen y de igual forma, las de la derecha se ubicarán en la parte izquierda para el flip horizontal (ver ejemplos en Figura 13).

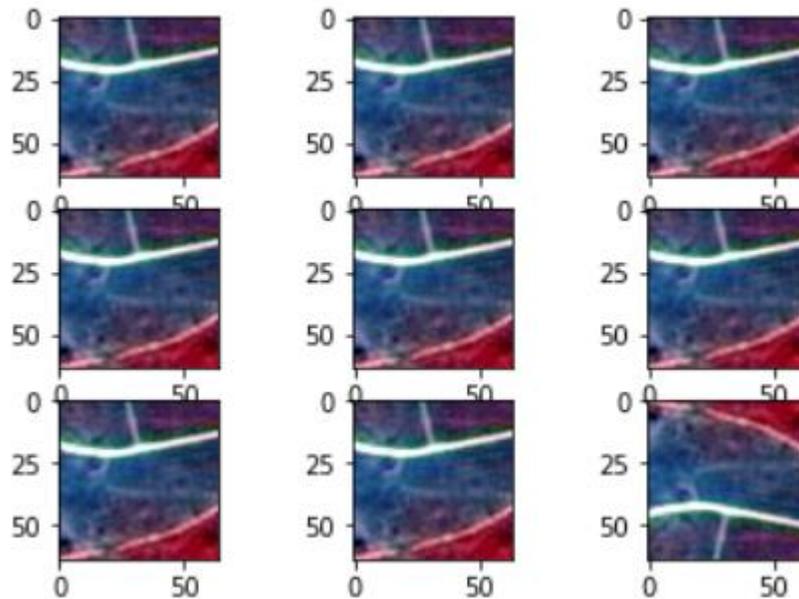


Figura 13. Aplicación de vertical flip en aumento de datos.

En el proceso de valoración y selección de operaciones sobre la imagen para el aumento de datos, se evaluaron otras opciones que finalmente se descartaron, considerando que no aportaban de manera significativa o por el contrario distorsionaban o dañaban las imágenes. Incluso con algunas operaciones las características de la imagen cambian en gran medida, lo que puede degradar el rendimiento de los algoritmos. Algunas de las operaciones evaluadas corresponden a *channel shift range*, *samplewise*, *contrast*, *color augmentation*, *saturation*, *adding noise* y *fill mode*.

4. Implementación y evaluación de modelos

En la sección número 4 se involucran las acciones correspondientes al desarrollo de los objetivos documentando los procesos que se llevaron a cabo en cada uno de los pasos realizados.

4.1 Modelado con AutoML (KerasTuner)

4.1.1 Modelo base

Para el desarrollo de la fase de modelamiento y cumplimiento del primer objetivo específico, se partió de un modelo base de aprendizaje profundo evaluando su comportamiento en entrenamiento y validación inicial, para determinar posteriormente el progreso a medida que se implementen variantes en el mismo. Este modelo base se presenta en la Figura 21.

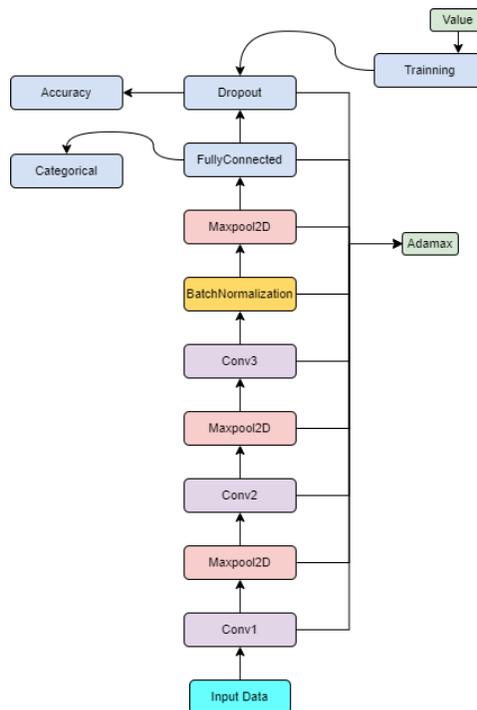


Figura 14. Arquitectura de modelo base.

La propuesta del modelo base cuenta con 3 capas convolucionales de 32, 64 y 128 filtros respectivamente, así como capas de *Maxpooling* y *Batch normalization*, capa

de dropout y optimizador, tasa de aprendizaje y función de pérdida seleccionadas de manera superficial con el fin de solo evidenciar el comportamiento inicial del modelo.

Para la obtención de resultados iniciales en la aplicación del modelo base se dispuso del entrenamiento de los datos con solo la distribución del dataset ilustrado en la figura 15, sin ningún tipo de aumento de datos ni realizando espacios de búsqueda de hiperparámetros, evaluando solo 1 valor por cada parámetro, con el fin de determinar el comportamiento y avance de los modelos a medida que se fueron implementando las demás fases. A continuación, se reflejan los datos y el resultado a nivel de *accuracy* en validación del modelo base:

Capas conv	Maxpooling	BatchNormalization	Tasa de aprendizaje	Optimizador	Accuracy en validación
3	True	True	0.001	Adamax	75%

Tabla 12. Hiperparámetros y resultados modelo base.

4.1.2 Espacio de búsqueda inicial y aumento de datos.

A partir del modelo base y sus resultados iniciales, se hizo necesario implementar una fase inicial que involucró la definición de un espacio de búsqueda, determinando valores para los distintos hiperparámetros. Para ello se utilizó AutoML, específicamente la herramienta KerasTuner.

Mediante KerasTuner se estableció la variación de algunos hiperparámetros generando rangos en valores como tasa de aprendizaje, filtros en capas convolucionales y tamaño del *kernel*, procurando ver el cambio y el nivel de *accuracy* que va a entregar el modelo con KerasTuner, a diferencia del modelo base.

De igual forma se recopiló la información de parámetros y rendimiento que aportó este modelo ya con la implementación de KerasTuner y nuevos parámetros:

trials	Capas conv	Filtros	MaxPooling	BatchNormalization	Tasa de aprendizaje	Kernel	Optimizador
10	3	(32/64) (64/128) (128/256)	True	True	0.01 - 0.001	3x3 5x5	Adamax

Tabla 13. Espacio de búsqueda inicial.

Aumento de datos	Filtros	Tasa de aprendizaje	Kernel	Accuracy (validación)
<i>True</i>	conv1(32) conv2(64) conv3(128)	0.001	3x3	82%

Tabla 14. Mejor modelo obtenido con aumento de datos y espacio de búsqueda inicial.

Con lo anterior se puede evidenciar que, al implementar el aumento de datos y la fase inicial de KerasTuner, el *accuracy* en validación del modelo mejoró significativamente llegando al 82%.

4.1.3 Ampliación de espacio de búsqueda

Posterior a la fase de implementación inicial de KerasTuner se procedió con la optimización del modelo añadiendo mayor número de valores posibles de hiperparámetros para evaluar y entrenar el modelo. Así mismo, se configuró el número de *trials* (intentos) para obtener diversidad de modelos y evaluar mayor cantidad de parámetros diferentes cuando se esté entrenando y validando automáticamente.

Para lo anterior, se configuró una cantidad de *trials* de 100, para obtener un total de hasta 3000 combinaciones de hiperparámetros, a partir de los siguientes espacios de búsqueda:

Hiperparámetros	Valores
Tasa de aprendizaje	(1e-1, 1e-2, 1e-3, 1e-4 y 1e-5)
Filtros por capa conv	(32-128) (128-256) (256-512)
Tamaño de Kernel	3x3 3x5 5x5
Dropout	[0.3 0.5]
Optimizadores	Adamax, Adagrad, RMSprop Adam

Tabla 15. Hiperparámetros y conjunto de valores del espacio de búsqueda de KerasTuner.

En esta sección el valor máximo de *accuracy* obtenido con los datos de validación fue del 88%. Este rendimiento se alcanzó con la siguiente combinación de hiperparámetros, de acuerdo con lo reportado por la herramienta *Keras Tuner*:

Filtros	Tasa de aprendizaje	Kernel	Dropout	Optimizador	Val_accuracy
conv1(41) conv2(147) conv3(256)	0.001	3x3	0.3	Adam	88%

Tabla 16. Mejor modelo ampliación espacio de búsqueda.

4.1.4 Optimización de recursos de computadora y división de modelos

Al implementar la fase anterior la cantidad de recursos computacionales y el tiempo de ejecución aumentó de manera considerable, por lo que se buscó la manera de reducir el número de combinaciones de hiperparámetros a evaluar. En este sentido, se segmentó el espacio de búsqueda con dos nuevos modelos cada uno con un espacio de búsqueda más reducido, como se presenta a continuación:

a)	b)																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Hiperparámetros</th> <th>Valores</th> </tr> </thead> <tbody> <tr> <td>Tasa de aprendizaje</td> <td>(1e-2, 1e-3)</td> </tr> <tr> <td>Filtros por capa conv</td> <td>(32-64) (64-128) (256-512)</td> </tr> <tr> <td>Tamaño de Kernel</td> <td>3x3 3x5</td> </tr> <tr> <td>Dropout</td> <td>0.3</td> </tr> <tr> <td>Optimizadores</td> <td>Adamax, Adagrad</td> </tr> </tbody> </table>	Hiperparámetros	Valores	Tasa de aprendizaje	(1e-2, 1e-3)	Filtros por capa conv	(32-64) (64-128) (256-512)	Tamaño de Kernel	3x3 3x5	Dropout	0.3	Optimizadores	Adamax, Adagrad	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Hiperparámetros</th> <th>Valores</th> </tr> </thead> <tbody> <tr> <td>Tasa de aprendizaje</td> <td>(1e-4, 1e-5)</td> </tr> <tr> <td>Filtros por capa conv</td> <td>(64) (64-128) (128-256)</td> </tr> <tr> <td>Tamaño de Kernel</td> <td>5x5</td> </tr> <tr> <td>Dropout</td> <td>0.5</td> </tr> <tr> <td>Optimizadores</td> <td>RMSprop, Adam</td> </tr> </tbody> </table>	Hiperparámetros	Valores	Tasa de aprendizaje	(1e-4, 1e-5)	Filtros por capa conv	(64) (64-128) (128-256)	Tamaño de Kernel	5x5	Dropout	0.5	Optimizadores	RMSprop, Adam
Hiperparámetros	Valores																								
Tasa de aprendizaje	(1e-2, 1e-3)																								
Filtros por capa conv	(32-64) (64-128) (256-512)																								
Tamaño de Kernel	3x3 3x5																								
Dropout	0.3																								
Optimizadores	Adamax, Adagrad																								
Hiperparámetros	Valores																								
Tasa de aprendizaje	(1e-4, 1e-5)																								
Filtros por capa conv	(64) (64-128) (128-256)																								
Tamaño de Kernel	5x5																								
Dropout	0.5																								
Optimizadores	RMSprop, Adam																								

Tabla 17. Hiperparámetros y conjunto de valores de dos espacios de búsqueda para optimización de recursos: a) Modelo 1, b) Modelo 2.

En esta sección el valor máximo obtenido y por tanto el mejor modelo en cuanto a resultado de *accuracy* en validación fue del 92%, las características y recopilación de los modelos de esta fase se relacionan a continuación (subsección 4.1.5).

4.1.5 Recopilación y obtención de los mejores modelos

Del conjunto de pruebas realizadas con KerasTuner, se seleccionan los cinco mejores modelos, en términos de *accuracy* de validación.

Modelo	1	2	3	4	5
<i>Val_accuracy</i>	92.36%	90.74%	91.44%	88.66%	91.66%
<i>Batch normalization</i>	False	True	False	True	True
<i>No. de capas</i>	3	2	3	2	3
<i>No. de filtros (capa 1)</i>	61	32	57	32	32
<i>Kernel</i>	5	3	5	3	3
<i>No. de filtros (capa 2)</i>	89	64	71	64	64
<i>Kernel</i>	5	3	3	3	3
<i>No. de filtros (capa 3)</i>	184	0	230	0	128
<i>Kernel</i>	3x3	5x5	3x3	3x3	5x5
<i>Dropout (capa 3)</i>	0.3	0.5	0.3	0.5	0.3
<i>Optimizador</i>	Adamax	RMSprop	Adamax	Adam	Adamax
<i>Tasa de aprendizaje</i>	0.001	0.001	0.001	0.001	0.001

Tabla 18. Hiperparámetros, valores y resultados de los mejores modelos obtenidos con AutoML (*usando Keras Tuner*).

De igual manera y teniendo en cuenta lo anterior, el modelo con mejor desempeño en cuanto a *accuracy* de validación obtuvo un 92.36%, correspondiente al modelo 1 de la tabla anterior.

Por lo cual, con la estrategia de diseño de AutoML, se selecciona el modelo que tiene los siguientes hiperparámetros de arquitectura y de entrenamiento:

Hiperparametros	Valores
Tasa de aprendizaje	0,001
Filtros por capa conv	(61) (89) (184)
Tamaño de Kernel	3x3
Dropout	0.3
Optimizadores	Adamax
Batch Normalization	True
MaxPooling	True

Tabla 19. Hiperparámetros y valores del mejor modelo obtenido con AutoML (usando Keras Tuner)

La siguiente figura muestra la arquitectura del modelo final para la clasificación binaria en el caso de esta investigación:

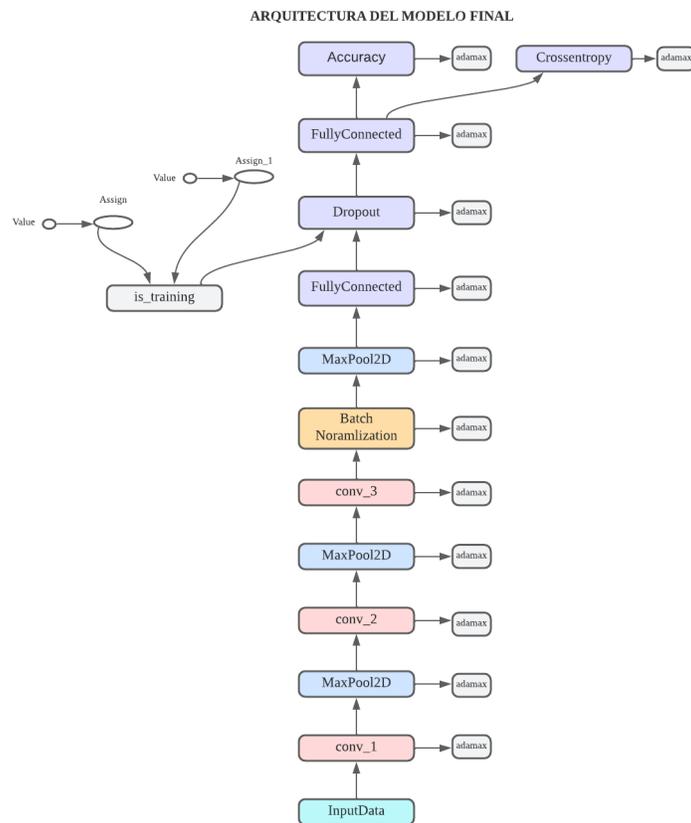


Figura 15. Esquema arquitectura del modelo completo de AutoML seleccionado.

A partir de los resultados anteriores, se crea la matriz de confusión del modelo, la cual corresponde con una gráfica en la cual se contabilizan la cantidad de predicciones correctas de la clase 0 y de la clase 1, así como la cantidad de clasificaciones incorrectas de cada una de las clases [19]. A continuación, se presenta la matriz de confusión obtenida:

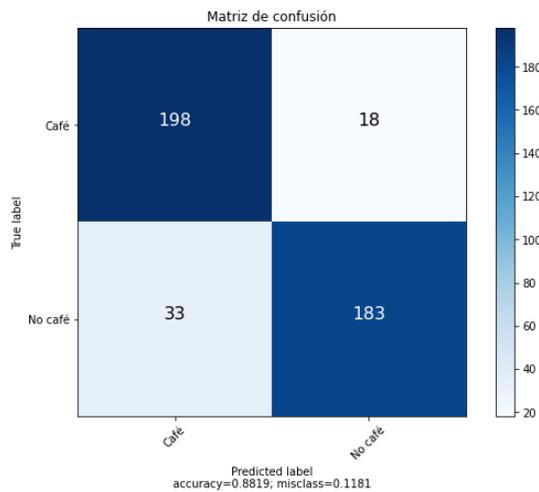


Figura 17. Matriz de confusión del mejor modelo.

La anterior figura evidencia que de las 216 imágenes pertenecientes a la clase café el modelo clasificó correctamente 198 de ellas (TP: verdaderos positivos), e incorrectamente clasificó 18 como no café (FN: falsos negativos). Adicionalmente, de las 216 imágenes de la clase no café, el modelo clasificó de manera acertada 183 (TN: verdaderos negativos) y 33 de manera errónea (FP: falsos positivos).

A partir de los valores de la matriz de confusión, se calcula el *accuracy* y la precisión del modelo, obteniendo:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} = \frac{198 + 183}{432} = 88.19\%$$

$$Precisión = \frac{TP}{TP + FP} = \frac{198}{198 + 33} = 85.7\%$$

4.2 Modelado con transferencia de aprendizaje

4.2.1 Modelo base.

Para esta fase relacionada con el segundo objetivo específico, se procedió a crear el modelo base, de forma similar a cómo se realizó en el desarrollo del primer objetivo específico.

Para el modelo base se tuvieron en cuenta 4 arquitecturas: VGG16, Resnet50, Xception y VGG19 (Tabla 16). Se realizó el entrenamiento de todos los parámetros y capas de las arquitecturas, sin ningún tipo de congelamiento ni recorte a las mismas. Se incluyó el aumento de datos explicado previamente, y el optimizador Adamax.

Los hiperparámetros se presentan a continuación:

Arquitecturas	Optimizador	Tasa de aprendizaje (por defecto)	Épocas
VGG16	Adamax	0.001	30
VGG19			
Xception			
Resnet50			

Tabla 20. Parámetros modelo base.

Al entrenar las arquitecturas de la tabla 21 se obtuvieron los siguientes resultados de *accuracy* en validación:

Arquitectura	Val_Accuracy
VGG16	90%
VGG19	88%
Xception	90.5%
Resnet50	89%

Tabla 21. Resultados Accuracy en validación para VGG16, VGG19, Xception y Resnet50.

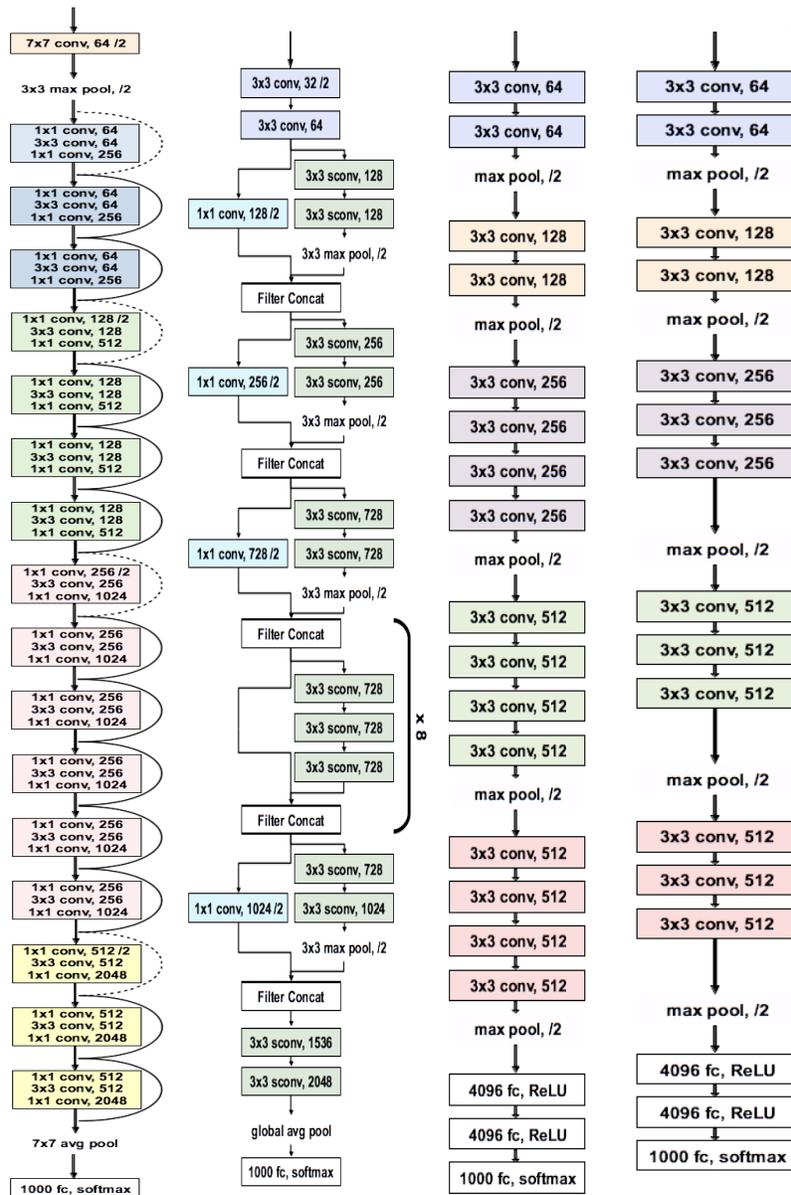


Figura 18. Arquitecturas Resnet50, Xception, VGG19 y VGG16 (de izquierda a derecha). Tomado de [30].

4.2.2 Aumento de datos y determinación del espacio de búsqueda.

Para esta fase y ya con un modelo base definido y unos resultados sobre los cuales partir, se generó un espacio de búsqueda sobre el cual poder realizar pruebas y documentar resultados. Este espacio de búsqueda consistió en la implementación de 27 modelos, obtenidos de la combinación de optimizadores, arquitecturas y

puntos de recorte del modelo pre-entrenado transferido. Los hiperparámetros utilizados se detallan en la Tabla 22.

Hiperparámetros y características utilizadas para esta fase:

Arquitectura	Optimizador	Bloque	Épocas	Tasa de aprendizaje
VGG16	Adagrad	block4_pool	50	0.001
VGG16	Adamax	block4_pool		
VGG16	Adam	block4_pool		
VGG16	Adagrad	block2_pool		
VGG16	Adamax	block2_pool		
VGG16	Adam	block2_pool		
VGG16	Adagrad	block5_pool		
VGG16	Adamax	block5_pool		
VGG16	Adam	block5_pool		
Resnet50	Adagrad	conv4_block6_add		
Resnet50	Adamax	conv4_block6_add		
Resnet50	Adam	conv4_block6_add		
Resnet50	Adagrad	conv2_block3_add		
Resnet50	Adamax	conv2_block3_add		
Resnet50	Adam	conv2_block3_add		
Resnet50	Adagrad	conv5_block3_add(todo)		
Resnet50	Adamax	conv5_block3_add(todo)		
Resnet50	Adam	conv5_block3_add(todo)		
Xception	Adagrad	add_1		
Xception	Adamax	add_1		
Xception	Adam	add_1		
Xception	Adagrad	add_6		
Xception	Adamax	add_6		
Xception	Adam	add_6		
Xception	Adagrad	block14_senpconv2_add		
Xception	Adamax	block14_senpconv2_add		
Xception	Adam	block14_senpconv2_add		

Tabla 22. Hiperparámetros y características de modelos evaluados por transferencia de aprendizaje.

A cada arquitectura se le aplicaron 2 distintos puntos de recorte, el primer punto en los bloques iniciales: para el caso de VGG16 el block2_pool, en Resnet50 el bloque conv2_block3_add, y para Xception el bloque add_1. Para el caso del segundo recorte se buscaron las capas intermedias de cada arquitectura teniendo en cuenta su estructura por bloques y recortando en los puntos correctos, en el caso de VGG16 se conservó la arquitectura y pesos hasta el bloque block4_pool, para Resnet50 hasta el bloque conv4_block6_add y para Xception hasta el bloque intermedio add_6. Como tercer punto de congelamiento se determinó utilizar la arquitectura completa (sin incluir las capas top); de esta forma, en VGG16 se conservó la

arquitectura y pesos hasta el bloque “block5_pool”, en Resnet50 hasta el bloque “conv5_block3_add” y en Xception hasta el bloque “block14_sepconv2_add”.

Resultados obtenidos en cuanto a *accuracy* en validación:

Arquitectura	Optimizador	Recorte por bloques	Val_accuracy
VGG16	Adagrad	block4_pool	90.28%
	Adamax	block4_pool	91%
	Adam	block4_pool	90.28%
	Adagrad	block2_pool	89.1%
	Adamax	block2_pool	92.59%
	Adam	block2_pool	91.6%
	Adagrad	block5_pool	93,40%
	Adamax	block5_pool	89,10%
	Adam	block5_pool	92,84%
Resnet50	Adagrad	conv4_block6_add	81%
	Adamax	conv4_block6_add	81,25%
	Adam	conv4_block6_add	80%
	Adagrad	conv2_block3_add	91,00%
	Adamax	conv2_block3_add	92%
	Adam	conv2_block3_add	92,13%
	Adagrad	conv5_block3_add(todo)	91,40%
	Adamax	conv5_block3_add(todo)	94%
	Adam	conv5_block3_add(todo)	92%
Xception	Adagrad	add_1	90%
	Adamax	add_1	91,80%
	Adam	add_1	91,40%
	Adagrad	add_6	89,50%
	Adamax	add_6	90,20%
	Adam	add_6	90,74%
	Adagrad	block14_sepconv2_add	87,00%
	Adamax	block14_sepconv2_add	93,30%
	Adam	block14_sepconv2_add	92,36%

Tabla 23. Resultados de rendimiento de los modelos por transferencia de aprendizaje evaluados.

De la Tabla 23 se identifica que existe una mejora entre algunos de los modelos en relación con el modelo base presentado en la Tabla 21. Por ejemplo, el modelo base VGG16 había alcanzado un *accuracy* cercano al 90%, mientras que con el ajuste de hiperparámetros aumenta hasta el 93.4%. En el caso de Xception el resultado en *accuracy* obtenido con el modelo base fue del 90.5%, mientras que con en el ajuste y búsqueda alcanza un 93.3%, de igual forma para el caso de Resnet50 su rendimiento inicial fue de del 89% y al aplicar el ajuste alcanza 94%.

Una vez consolidados y analizados los resultados de esta fase, se seleccionaron los siguientes modelos (Tabla 24) de acuerdo con su rendimiento:

Arquitectura	Optimizador	Recorte por bloques	Val_accuracy
VGG16	Adagrad	block5_pool (toda la arquitectura)	93,40%
Resnet50	Adamax	conv5_block3_add (toda la arquitectura)	94%
Xception	Adamax	block14_sepconv2_add (toda la arquitectura)	93,30%

Tabla 24. Modelos seleccionados de la fase 4.2.2

4.2.3 Reentrenamiento de los mejores modelos.

En esta fase se utilizan los tres modelos seleccionados previamente, pero ahora, se realiza un recorte al modelo pre-entrenado. Se aumenta la cantidad de épocas de entrenamiento y se fija la tasa de aprendizaje en 0.001. Adicionalmente, con el propósito de disminuir el sobre-ajuste y como buena práctica de diseño, se incluye un bloque de *dropout* para diferentes valores entre 0.2 y 0.5. Estos modelos y sus resultados de rendimiento se presentan en la Tabla 25.

Arquitectura	Optimizador	Recorte de Bloque	Épocas	Tasa de aprendizaje	Dropout	Accuracy en validación
Xception	Adamax	block14_senpconv2_add	100	0.001	0.2	94%
					0.3	95%
					0.5	93,30%
Resnet50	Adamax	conv5_block3_add			0.2	93,28%
					0.3	93,10%
					0.5	93,50%
VGG16	Adagrad	block5_pool			0.2	93,90%
					0.3	93%
					0.5	92,13%

Tabla 25. Resultados de los modelos de la fase 4.2.3

De acuerdo con los resultados anteriores, la mejor configuración corresponde a Xception con ajuste fino y los hiperparámetros relacionados en la Tabla. Su rendimiento con datos de validación es del 95% (*Accuracy*).

4.2.4 Obtención y evaluación del mejor modelo.

La anterior fase permitió obtener el mejor modelo haciendo uso de transferencia de aprendizaje, a partir de la evaluación de un modelo base, aumento de datos y determinación del espacio de búsqueda que arrojó como resultado el modelo descrito en las siguientes figuras:

Hiperparámetros del mejor modelo:

Arquitectura	Optimizador	Épocas	Tasa de aprendizaje	Dropout
Xception	Adamax	100	0.001	0.3

Tabla 26. Hiperparámetros del mejor modelo de TL.

Resultados del mejor modelo:

Arquitectura	Recorte por bloques	Optimizador	Val_accuracy Mejor modelo
Xception	block14_senpconv2_add	Adamax	95%

Tabla 27. Resultados del mejor modelo de TL.

Arquitectura mejor modelo Xception.

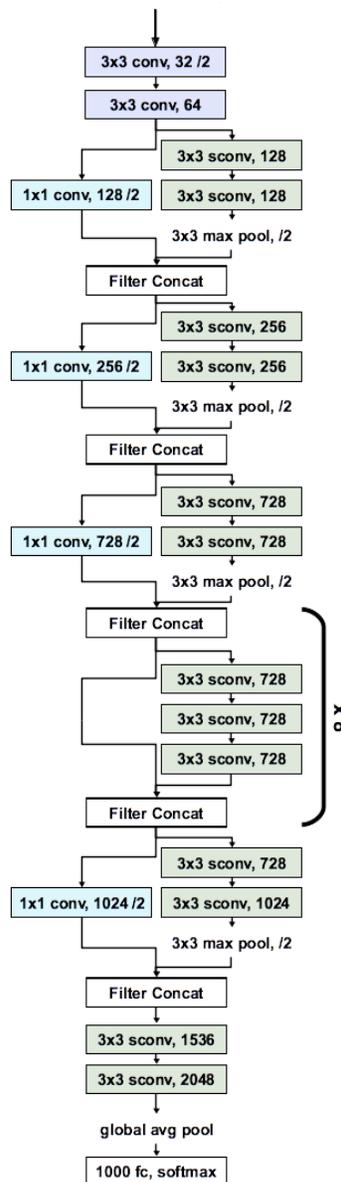


Figura 19. Arquitectura Xception del mejor modelo obtenido por medio de TL.[30].

Con el modelo de la Figura 19, el paso final consistió en realizar la predicción al conjunto de imágenes de *test*, de forma similar a la evaluación realizada con el mejor modelo por AutoML.

Para evaluar de manera objetiva estos resultados, de nuevo se calcula la matriz de confusión, con el fin de visualizar de manera más clara la predicción realizada con los modelos por transferencia de aprendizaje, obteniendo el siguiente resultado:

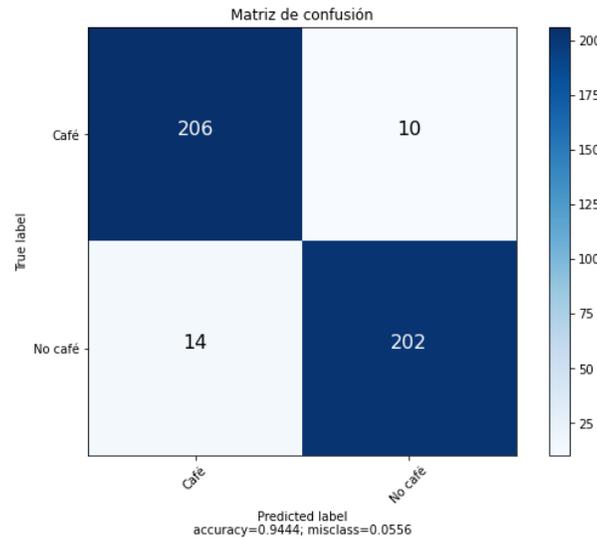


Figura 21. Matriz de confusión del mejor modelo obtenido con TL.

La anterior figura evidencia que de las 216 imágenes pertenecientes a la clase café el modelo obtuvo 206 verdaderos positivos (imágenes de clase café clasificadas correctamente), e incorrectamente clasificó 10 imágenes como no café (falsos negativos); de igual manera, de 216 imágenes de la clase no café el modelo clasificó de manera acertada 202 (verdaderos negativos) y 14 de manera errónea (falsos positivos).

Teniendo en cuenta lo anterior, se calcula el *accuracy* en predicción que arroja un valor del 94.4%, resultado que muestra el porcentaje de aciertos del clasificador. Así mismo el modelo alcanzó un 93.6% de precisión, lo que refleja el comportamiento en la predicción del modelo solamente para la clase café.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} = \frac{206 + 202}{432} = 94.4\%$$

$$Precisión = \frac{TP}{TP + FP} = \frac{206}{206 + 14} = 93.6\%$$

5. Análisis de resultados

Comparación de resultados entre AutoML y transferencia de aprendizaje

En las anteriores secciones se apreció la recopilación de resultados y la selección de los hiperparámetros que permitieron obtener el mejor modelo usando las técnicas tanto de AutoML como de transferencia de aprendizaje, a partir de estos resultados se pueden generar una serie de análisis y comparaciones entre las técnicas empleadas, los hiperparámetros usados en cada una de ellas y por tanto seleccionar un modelo con mejor comportamiento en general.

En cuanto a selección de valores para los hiperparámetros, ambos modelos obtuvieron valores similares de tasa de aprendizaje, optimizador y dropout, así como el uso de aumento de datos. Es decir, se puede inferir que al implementar un modelo de clasificación para el caso específico del dataset *Brazilian coffee scenes* es importante tener en cuenta los siguientes valores que arrojaron los mejores resultados de precisión a la hora de clasificar:

Hiperparámetros comunes entre AutoML y transferencia de aprendizaje

Hiperparámetros	Estrategias empleadas	
	AutoML	Transferencia de aprendizaje
Optimizador	Adamax	Adamax
Tasa de aprendizaje	0.001	0.001
Dropout	0.3	0.3
Aumento de datos	True	True
Maxpooling	True	True

Tabla 28. Hiperparámetros comunes AutoML y Transferencia de aprendizaje.

De igual manera la forma de implementación de ambas técnicas es distinta ya que mientras una de ellas, como es el caso de AutoML, empleó técnicas de selección de hiperparámetros de manera automática a partir de rangos de valores configurados para obtener así gran cantidad de combinaciones y por tanto mucha variedad de modelos como lo fueron cerca de 3000 de acuerdo a la cantidad de trials y parámetros que se dispusieron en cada fase de esta estrategia en particular, mientras que para transferencia de aprendizaje la configuración y prueba de valores de cada hiperparámetro así como la selección de arquitecturas se realizó de forma manual; por la razón anterior la diferencia entre la cantidad de modelos entrenados en una estrategia de diseño respecto de la otra fue muy amplia, ya que para el caso de transferencia de aprendizaje se llegaron a entrenar cerca de 80 modelos en las diferentes fases versus los cerca de 3000 de AutoML; otra de las notables diferencias entre ambas herramientas es la forma de entrenar el modelo, mientras que para el caso de transferencia de aprendizaje se dispuso de la selección de arquitecturas como Xception, VGG16 o Resnet50 que son arquitecturas ya establecidas, evaluadas y consideradas en muchos problemas de clasificación de todo tipo y con niveles de profundidad y entrenamiento muy elaborados, en el caso de AutoML el diseño del modelo y capas implementadas fue determinado y realizado recopilando información e intentando evaluar qué tanto avanzaba el modelo a medida que se implementaba pero de forma más autóctona siendo éstas algunas de las razones del posible porqué de los resultados.

Teniendo en cuenta lo anterior y los resultados obtenidos en numerales ya explicados se pudo determinar lo siguiente:

Técnicas empleadas	AutoML	Transferencia de aprendizaje
Resultados mejor modelo	92,36%	95%

Tabla 29. Val_accuracy mejor modelo por técnica.

De acuerdo con la Tabla 29 y con lo obtenido a la hora de hacer las pruebas con las imágenes dispuestas para test en la que ambos modelos tanto para el caso de AutoML como el de transferencia de aprendizaje los resultados que se generaron entre el valor de accuracy en testeo como el valor arrojado en el accuracy en validación para cada estrategia fue muy similar, observar Tabla 26:

Estrategia de diseño	Resultados mejores modelo	
	Accuracy en validación	Accuracy en predicción
AutoML	92,30%	88%
Transferencia de aprendizaje	95%	94%

Tabla 30. Precisión en validación vs precisión en predicción mejores modelos.

Consecuente a lo obtenido en los resultados y determinando mediante un análisis de estos se establece que para el caso de esta investigación el modelo optimo y con gran precisión a la hora de generar las pruebas para la clasificación de imágenes de cultivo de café a través del uso del dataset *Brazilian coffee scenes* es por medio de la estrategia de diseño transferencia de aprendizaje en conjunto con los Hiperparámetros descritos en la Tabla 26.

6. Conclusiones

- Se implementó un modelo de clasificación obtenido por medio de la técnica AutoML y el uso del dataset *Brazilian coffee scenes* que permite reconocer imágenes satelitales de cultivo de café con un *accuracy* del 89%.
- Se implementó un modelo de clasificación obtenido mediante transferencia de aprendizaje y el uso del dataset *Brazilian coffee scenes* que permite reconocer imágenes satelitales de cultivo de café con un *accuracy* del 94%.
- En ambas estrategias de diseño se realizó ajuste de hiperparámetros tanto de arquitectura como de entrenamiento, a partir de un espacio de búsqueda automática (con AutoML) o manual (por transferencia de aprendizaje).
- Para ambas estrategias de diseño empleadas, el ajuste de hiperparámetros incrementa de forma significativa el valor de *accuracy* del modelo base, en promedio del 8%.
- En términos de profundidad, los mejores resultados se obtienen al transferir la arquitectura completa, y no de forma parcial, con una diferencia entre los valores de *accuracy* de aproximadamente el 2%.
- A la hora de diseñar los modelos de clasificación, el optimizador impacta significativamente el desempeño del modelo. En esta investigación se encontró que para la clasificación de zonas de cultivo de café con el dataset *Brazilian coffee scene*, Adamax es el optimizador más adecuado entre los evaluados en esta investigación.
- En cuanto a arquitectura, entre las arquitecturas evaluadas por transferencia de aprendizaje la de mejor rendimiento fue Xception, por lo que se infiere que las arquitecturas paralelas son muy adecuadas para manejar problemas de clasificación de este tipo, sin

dejar de lado que arquitecturas secuenciales como VGG16 presentaron un comportamiento competitivo.

- Para el caso de esta investigación y con los resultados obtenidos se concluye que la estrategia de diseño de modelos más apropiada para realizar una clasificación de zonas de cultivo de café con imágenes satelitales por medio del dataset *Brazilian coffee scene* fue a través de transferencia de aprendizaje.

7. Bibliografía

- [1] M. X. Bastidas, «reconoSERID,» 29 Mayo 2020. [En línea]. Available:
] <https://reconoserid.com/que-es-el-aprendizaje-de-maquina-y-en-que-casos-deberia-usarlo-en-su-negocio/>.
- [2] IBM, «IBM,» 15 07 2020. [En línea]. Available:
] <https://www.ibm.com/cloud/learn/machine-learning>.
- [3] C. D. Silva, «Brazilian Coffee Scenes Dataset,» 12 Noviembre 2017. [En línea].
] Available: <http://patreo.dcc.ufmg.br/2017/11/12/brazilian-coffee-scenes-dataset/>.
- [4] Bancolombia, «Guía completa: Cómo cultivar Café,» 30 Agosto 2018. [En
] línea]. Available:
<https://www.grupobancolombia.com/wps/portal/negocios/actualizate/sostenibilidad/guia-cultivo-cafe-colombia>.
- [5] IBM, «Metodología Fundamental para la ciencia de datos,» Junio 2015. [En
] línea]. Available: <https://www.ibm.com/downloads/cas/6RZMKDN8>.
- [6] Anonimo, «What is Artificial Intelligence? How Does AI Work?,» [En línea].
] Available: <https://builtin.com/artificial-intelligence>.
- [7] BBVA, «'Machine learning': ¿qué es y cómo funciona?,» 08 Noviembre 2019.
] [En línea]. Available: <https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/>.
- [8] L. G. S. a. G. C, Computer Vision, 2001.
]
- [9] C. G. Moreno, «Indra,» 2019. [En línea]. Available:
] <https://www.indracompany.com/es/blogneo/deep-learning-sirve#:~:text=Por%20lo%20tanto%2C%20la%20principal,datos%2C%20no%20es%20Machine%20Learning>.
- [1] UPBE, «INTELIGENCIA ARTIFICIAL: LA IMPORTANCIA DE LOS DATOS,»
0] 2020. [En línea]. Available: <https://www.upbe.ai/blog/inteligencia-artificial-datos/>. [Último acceso: 05 09 2022].

- [1] J. Barrios, «Health big data,» 2022. [En línea]. Available:
1] <https://www.juanbarrios.com/redes-neurales-convolucionales/#:~:text=c%C3%A9lula%20sea%20activada.-,C%C3%B3mo%20están%20construidas%20y%20c%C3%B3mo%20funcionan,un%20mapeo%20causal%20no%2Dlineal.>
- [1] F.-X. Sakib Mostafaa, «Diagnosis of autism spectrum disorder with
2] convolutional autoencoder and structural MRI images,» 2021. [En línea]. Available:
<https://www.sciencedirect.com/science/article/pii/B978012822822700003X>.
[Último acceso: 05 09 2022].
- [1] savyakhosla, «CNN | Introducción a la capa de agrupación,» 24 08 2022. [En
3] línea]. Available: <https://www.geeksforggeeks.org/cnn-introduction-to-pooling-layer/>. [Último acceso: 05 09 2022].
- [1] Microsoft, 06 06 2022. [En línea]. Available: <https://docs.microsoft.com/es-es/azure/machine-learning/how-to-tune-hyperparameters>. [Último acceso: 11 09 2022].
- [1] tradingsys ORG, «tradingsys,» 16 Febrero 2012. [En línea]. Available:
5] <https://www.tradingsys.org/metodos-de-optimizacion-algoritmos-geneticos/#:~:text=En%20general%20podemos%20definir%20los,pueden%20aproximarse%20por%20los%20m%C3%A9todos.>
- [1] Interactive Chaos, [En línea]. Available:
6] <https://interactivechaos.com/es/manual/tutorial-de-deep-learning/tasa-de-aprendizaje>. [Último acceso: 11 09 2022].
- [1] J. Deng, «ImageNet,» [En línea]. Available:
7] <https://paperswithcode.com/dataset/imagenet>. [Último acceso: 10 09 2022].
- [1] W. Jeremy, V. Dan y W. Sean, Spring Research Presentation: A Theoretical
8] Foundation for Inductive Transfer, 2007.
- [1] Creative Commons Attribution 4.0 , «Machine Learning Glossary,» 28
9] Septiembre 2021. [En línea]. Available: https://developers.google.com/machine-learning/glossary?hl=es_419#t. [Último acceso: 21 Abril 2022].
- [2] G. K. Thommen y B. Roland, Self-organizing maps for storage and transfer of
0] knowledge in reinforcement learning, 2019.

- [2 H. F. H. H. L.-B. K. Thornton C, Auto-WEKA: Combined Selection and
1] Hyperparameter Optimization of Classification Algorithms, 2013.
- [2 Microsoft, «¿Qué es el aprendizaje automático automatizado (AutoML)?», 29
2] 08 2022. [En línea]. Available: <https://docs.microsoft.com/es-es/azure/machine-learning/concept-automated-ml>. [Último acceso: 04 09 2022].
- [2 Desconocido, «Machine Learning: Selección Métricas de clasificación», 19 01
3] 2019. [En línea]. Available: <https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/>. [Último acceso: 11 09 2022].
- [2 A. & Das y S. Chandran, «Transfer learning with RES2Net for remote sensing
4] scene classification.,» de *11th International Conference on Cloud Computing, Data*, 2021.
- [2 N. A. B. E. H. M. S. H. A. & Laban y M. F. Tolba, «Multiscale satellite image
5] classification using deep learning approach,» 2020.
- [2 A. A. K. K. S. A. N. P. C. M. B. Kumar, «Multilabel classification of remote
6] sensed satellite imagery.,» de *Transactions on Emerging Telecommunications Technologies.*, 2021.
- [2 J. H. H. P. J. Z. J. C. L. T. C. & Chen y H. Li, «Contextual information-
7] preserved architecture learning for remote-sensing scene classification.,» de *IEEE Transactions on Geoscience and Remote Sensing.*, 2021.
- [2 TensorFlow, «tf.keras.preprocessing.image.ImageDataGenerator,» 03 marzo
8] 2022. [En línea].
- [2 Keras, «Image augmentation layers,» [En línea]. Available:
9] https://keras.io/api/layers/preprocessing_layers/image_augmentation/random_flip/. [Último acceso: 21 04 2022].
- [3 T. J. C. E. R. R. Z. F. A. F. Matheus M. Leonardo*, «Deep Feature-based
0] Classifiers for Fruit Fly Identification (Diptera: Tephritidae),» de *Conference: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2018.