



ARTÍCULO

Análisis de prueba de concepto del modelo red neuronal para detección de cambios en coberturas de plantaciones.

Felipe Guzmán Vargas¹; Catalina Zumaeta Guzmán¹; Carlos Bahamondez Villarroel¹; Rodrigo Sagardía Parga¹ & Carlos Buchner Asenjo^{1*}

¹ Instituto Forestal, sede Los Ríos. Valdivia, Chile. cbuchner@infor.cl

*Autor para correspondencia (arial 8)

DOI: <https://doi.org/10.52904/0718-4646.2021.611>

Recibido: 28.08.2024; Aceptado 10.09.2024.

RESUMEN

El seguimiento de los cambios en la cobertura forestal se ha realizado tradicionalmente con sensores remotos, pero en años recientes se ha incorporado el *Machine Learning* (ML), una rama de la Inteligencia Artificial (IA) que desarrolla algoritmos capaces de aprender de los datos y mejorar sin ser programados explícitamente.

El documento presenta una prueba de concepto de ML realizada en la región del Maule. Los resultados del uso de ML en este contexto fueron positivos, con una tasa de precisión del 92% en la métrica de asertividad y del 90% al predecir etiquetas para la provincia de Talca. Se concluye que los sistemas de clasificación basados en redes neuronales son una herramienta importante para optimizar los procesos de clasificación y complementarlos con análisis de datos de alta dimensión, permitiendo que estos se puedan adaptar a diferentes condiciones geográficas. Estos avances permiten una mejor comprensión y gestión de los recursos forestales en el país.

Palabras clave: Cambios en cubierta forestal, *Machine Learning* (ML)

SUMMARY

Monitoring changes in forest cover has traditionally been done with remote sensing, but in recent years Machine Learning (ML), a branch of Artificial Intelligence (AI) that develops algorithms capable of learning from data and improving without being explicitly programmed, has been incorporated.

The paper presents a proof of concept of ML carried out in the Maule region. The results of using ML in this context were positive, with an accuracy rate of 92% in the assertiveness metric and 90% in predicting labels for the province of Talca. It is concluded that neural network-based classification systems are an important tool for optimizing classification processes and complementing them with high-dimensional data analysis, allowing them to be adapted to different geographical conditions. These advances allow a better understanding and management of forest resources in the country.

Key words: Forest cover changes. *Machine Learning* (ML)

INTRODUCCIÓN

Para el seguimiento de los cambios temporales en la cobertura forestal se suele recurrir, por consideraciones de costos, al uso de sensores remotos, a partir de los cuales se buscan patrones característicos que permiten identificar estructuras asociadas a actividades ejecutadas sobre la cobertura de los bosques, por ejemplo, cambios en cobertura por cosechas o cortas intermedias, incendios forestales, nivel de severidad de estos, eventos climáticos extremos, como deslizamientos de tierra, *browning* por sequía extrema, eventos disruptivos de plagas, etc.

En el afán de efectuar monitoreo permanente, se han utilizado técnicas tradicionales de sensores remotos para la detección de cambios. Entre ellas las más comunes son el método de diferencias de imágenes, el uso de componentes principales, el análisis de vectores de cambios, diferencias en post clasificación entre otros (Lu *et al.*, 2004).

Todos estos métodos, independiente de cuál sea el seleccionado, terminan en la aplicación de técnicas de clasificación/regresión que permiten la asignación de clases/valores debidamente definidos por el usuario, por ejemplo, el uso de árboles de regresión, *k-nearest neighbors* y *support vector machine* (Breiman, 2001).

En tal contexto, se ha generalizado en los últimos años el concepto de *Machine Learning* (ML) que se lo reconoce como una rama de la Inteligencia Artificial (IA) que se centra en el desarrollo de algoritmos que pueden aprender de los datos y mejorar su rendimiento en tareas específicas, sin necesidad de ser explícitamente programados para ello (Zelin *et al.*, 2018). Esto lo diferencia de la programación tradicional, en la que las soluciones son programadas con un conjunto de instrucciones específicas que se deben seguir para resolver un problema tipo (Goodfellow *et al.*, 2016). Así, el ML se utiliza en variadas aplicaciones, como, por ejemplo, reconocimiento de voz, reconocimiento de imágenes, clasificación de texto, predicción de precios, recomendación de productos y control de robots.

ANTECEDENTES DE MACHINE LEARNING (ML)

En general se reconocen dos tipos principales de *Machine Learning*, el aprendizaje supervisado y el no supervisado

En el aprendizaje supervisado, el algoritmo de ML recibe un conjunto de datos de entrenamiento que contiene tanto datos de entrada como, datos de salida. El algoritmo aprende a asociar los datos de entrada con los datos de salida. Por ejemplo, un algoritmo de aprendizaje supervisado podría utilizarse para entrenar un sistema de reconocimiento de imágenes de animales.

El sistema sería entrenado con un conjunto de datos de entrenamiento que contenga imágenes de animales en distintos ángulos de visualización (técnica conocida como *augmentation*). El algoritmo aprenderá a asociar e identificar como animal en base a las distintas variantes comprendidas en las imágenes. Cuando se trata de responder a un resultado específico (p.ej. una clase) esto implica tres conceptos básicos:

- Clasificación: Cuando los valores de entrada son divididos en clases que provienen desde el conocimiento previo del operador respecto de los datos y que, el operador los agrupa en una “clase” discreta. Esto habilita el concepto de supervisión.
- Regresión: Equivalente a la clasificación, aunque involucra salidas de variables continuas, no discretas.
- Agrupamiento: Cuando los datos de entrada son divididos en grupos, pero a diferencia de la clasificación, estos grupos no son conocidos de antemano y su agrupación no necesariamente tiene un trasfondo explicativo.

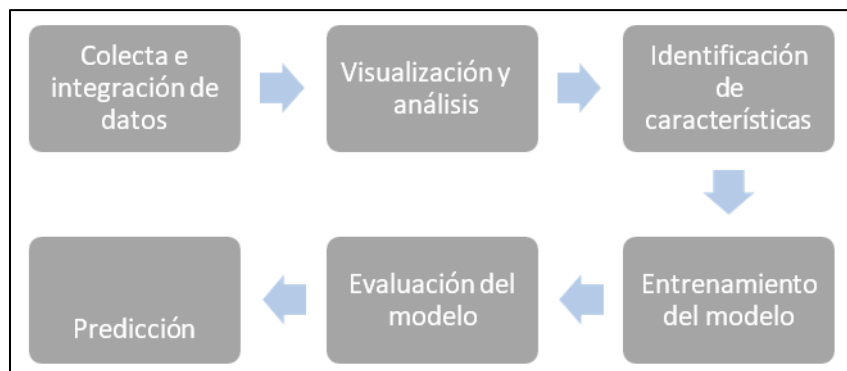
En el aprendizaje no supervisado, el algoritmo de ML recibe un conjunto de datos de entrenamiento que solo contiene datos de entrada. El algoritmo aprende a encontrar patrones en los datos de entrada. Por ejemplo, un algoritmo de aprendizaje no supervisado suele emplearse para agrupar datos por una cierta característica. El algoritmo aprenderá a encontrar patrones en los datos de la característica de interés. Al igual que en el algoritmo de supervisión en este tipo de ML se distinguen tres conceptos:

- Agrupamiento: Los datos son agrupados y separados por algún criterio de similitud.
- Visualización de datos: Utiliza herramientas que permiten visualizar datos en varias dimensiones.
- Modelos: Mediante la definición de un modelo de distribución de probabilidad de los datos de entrada, se habilita la generación de nuevos datos.

En general e independiente del tipo de ML a utilizar (supervisado o no supervisado) se debe considerar un conjunto de pasos metodológicos que aseguren la apropiada aplicación de los algoritmos de ML. Al momento de implementar un algoritmo de ML se visualizan los siguientes pasos:

1. Definición del Problema: Identificar debidamente el problema en la forma lo más simple posible.
2. Colecta de Datos: Reunir y habilitar los datos que serán utilizadas para entrenar el modelo.
3. Exploración de los Datos: Es aconsejable la utilización de métodos estadísticos y técnicas de visualización para reflejar de formas correcta la estructura de los datos.
4. Pre-proceso de Datos: Preparar los datos para modelación realizando normalizaciones y las transformaciones necesarias para aplicarlos al modelo.
5. División de los Datos: Dividir los datos en datos de entrenamiento y de prueba para validar el modelo.
6. Selección de un Modelo: Selección de un modelo de ML que sea apropiado al problema
7. Entrenamiento: Utilizar los datos de entrenamiento para entrenar el modelo, ajuste parámetros para mejorar desempeño del modelo.
8. Evaluar el Modelo: Utilizar los datos de prueba para evaluar el desempeño del modelo en base a una muestra de datos independiente.
9. Afinamiento del Modelo: Finalmente, en base a los resultados obtenidos por la evaluación con datos de prueba, reajustar los parámetros para mejorar la precisión del modelo.

Los pasos metodológicos anteriormente descritos dan origen al flujo conocido como *ML pipeline* y corresponde al flujo lógico de automatización de eventos necesarios de completar para aplicar ML (**Figura 1**).



(Fuente: Adaptado de Novak, 2020)

Figura 1. Diagrama general de un *pipeline* de *Machine Learning*.

Para ejecutar sus algoritmos ML se respalda en un conjunto amplio de disciplinas matemáticas y estadísticas, entre ellas se pueden mencionar:

- Álgebra lineal, incluyendo vectores y matrices, sistema de ecuaciones lineales, factorización de matrices y espacios vectoriales.
- Estadística, incluyendo estadígrafos básicos, test de hipótesis y teoría de estimación e inferencia.
- Geometría, incluyendo vectores y normas, distancias, medidas de similitud, proyecciones ortogonales, algoritmos geométricos, etc.
- Cálculo, incluyendo técnicas de diferenciación y derivadas.
- Probabilidades, incluyendo distribuciones discretas y continuas, distribuciones muestrales, teoría central del límite y ley de los grandes números.
- Regresiones
- Reducción de dimensiones

MATERIAL Y MÉTODO

Área de Prueba

La región del Maule presentaba en diciembre de 2021 una superficie de 412.873 ha de plantaciones, esto es 6.455 ha menos que en el año anterior. Se destaca la provincia de Talca, que concentra el 42% de la superficie regional de plantaciones forestales. En la región la especie predominante es *Pinus radiata*, con el 87% de la superficie plantada, y cuyas plantaciones disminuyeron en 5.532 hectáreas respecto del año 2020. En el mismo periodo la superficie de *Eucalyptus globulus* disminuyó en 758 ha.

En este contexto, para el desarrollo de la prueba de concepto de ML, asociada a variación de cobertura de plantaciones se trabajó en etapas relacionadas a: (i) Adquisición de imágenes satelitales, (ii) preprocesamiento y análisis de las imágenes, (iii) adquisición de características y preparación de los datos; y (iv) clasificación y medición (**Figura 2**).



Figura 2. Etapas para las pruebas de ML con una red neuronal

Adquisición de imágenes satelitales

Las imágenes satelitales utilizadas corresponden a las imágenes de Sentinel-2, provistas por la Agencia Espacial Europea (ESA)¹. La selección de estas imágenes para el proceso de generación de los *dataset* se debe a su disponibilidad y acceso gratuito. Ellas cuentan con una resolución espacial de 10 metros, lo que permite detectar características de las plantaciones forestales en la región de interés. Además, la frecuencia de adquisición de imágenes es alta, garantizando la obtención de datos actualizados en el análisis.

Estructura de la Prueba de Concepto

Para el desarrollo de la prueba se realizaron los siguientes pasos:

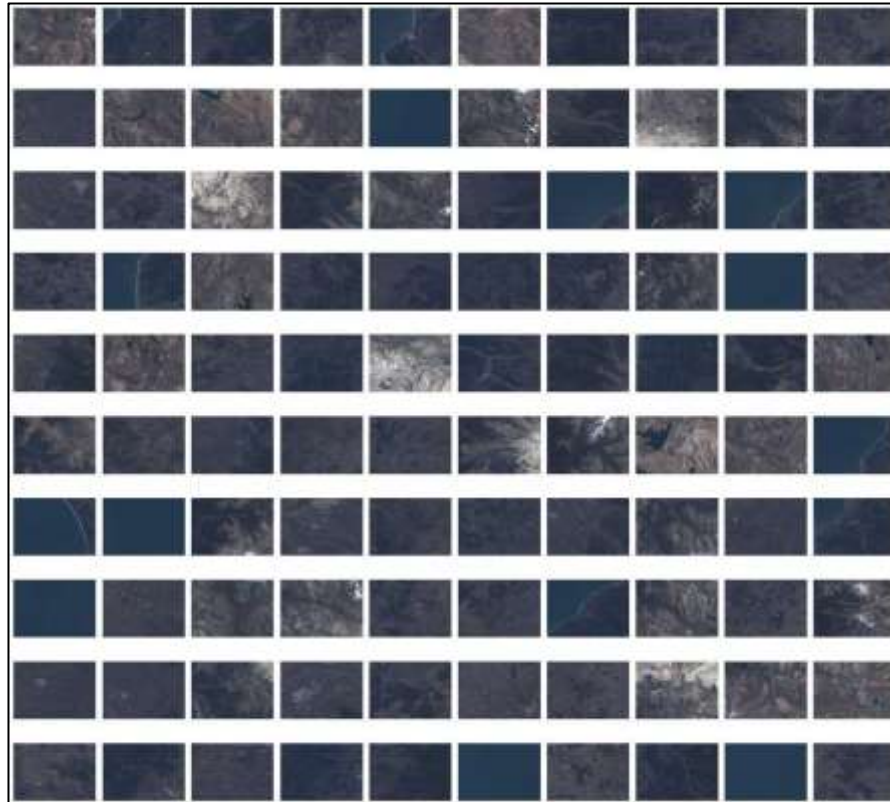
- *Paso 1: Generación de dataset en función de imagen sentinel-2 para la región del Maule.*

Para este caso se utilizaron imágenes del satélite Copernicus/Sentinel-2 correspondientes al periodo del año 2020 entre los meses de enero y diciembre. Los parámetros de selección corresponden a la exclusión de imágenes cuya cobertura de nubes fuese mayor al 20% en el caso de que exista impacto se reduce a través de la mediana. Las líneas de código para la selección de imágenes se presentan en el **Cuadro 1**, en tanto las imágenes obtenidas se muestran en la **Figura 3**.

¹ <https://www.esa.int/>

Cuadro 1. Código de selección de datos.

```
#Usando Sentinel-2  
collection = ee.ImageCollection('COPERNICUS/S2') \  
.filterBounds(geometry) \  
.filterDate('2020-01-01', '2020-12-31') \  
.filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))  
mosaic = collection.median()
```

**Figura 3.** Imágenes obtenidas con la selección regional a través de Geemap.

- *Paso 2: Descarga de Imágenes contenidas en el área de estudio*

Este proceso se realiza para el trabajo posterior, el cual será utilizado como el conjunto de datos en la fase de entrenamiento, la descarga de los datos se lleva a cabo con la librería de *Google Earth Engine* (EE)² la cual permite gestionar la descarga a través de *Google Drive*³.

- *Paso 3: Generación del mosaico*

Se utiliza la librería *Rasterio*⁴ para realizar un mosaico de las imágenes obtenidas a través del paso anterior para ello se emplea el código indicado en el **Cuadro 2**. El mosaico obtenido se muestra en **Figura 4**.

² <https://earthengine.google.com/>

³ <https://drive.google.com/>

⁴ <https://rasterio.readthedocs.io/en/latest/index.html>

Cuadro 2. Código de generación de mosaico

```
for archivo in tqdm(archivos, desc="Leyendo imágenes"):
    rasters.append(rasterio.open(archivo))
mosaico, transform = merge(rasters)
with rasterio.open('/data/mosaico/maule_mosaic.tif', 'w', **rasters[0].meta) as dest:
    dest.write(mosaico)
```



Figura 4. Mosaico para la región del Maule con Sentinel-2.

- Paso 4: Etiquetas de datos para imágenes (vectores)

Para el proceso de etiquetado de imágenes se utilizó las coberturas de cartografía de INFOR y CONAF, las que permitieron obtener las categorías de datos que pasaran a ser las etiquetas del modelo de *Machine Learning*.

En el caso de las coberturas de INFOR, la cartografía seleccionada corresponde a la de plantaciones del año 2021, que posee 16.716 registros polígonos. Esta cobertura ofrece 8 atributos de los cuales se seleccionaron atributos como especie y geometría, esta última, corresponde a la representación espacial de los polígonos en la cobertura.

En la **Figura 5** se presenta un análisis descriptivo de la cobertura de INFOR, en la cual se observa a través de un mapa la distribución geográfica los dos tipos de plantaciones seleccionados para esta prueba, el primero de ellos corresponde a *Pinus radiata* (en verde) y *Eucalyptus globulus* (en rojo). Se aprecia que las plantaciones de *Pinus radiata* son más prevalentes en la región en comparación con *Eucalyptus globulus* debido a la densidad en el mapa.

Adicionalmente se genera un conteo de especies para realizar una comparación cuantitativa del número de ocurrencias (conteo) de diferentes especies. Las especies *Pinus radiata* y *Eucalyptus globulus* tienen los conteos más altos, con *Pinus radiata* siendo el 63% de la data y *Eucalyptus globulus* con un 35%. Las otras

especies poseen conteos mucho menores lo que puede indicar menor presencia en la región.

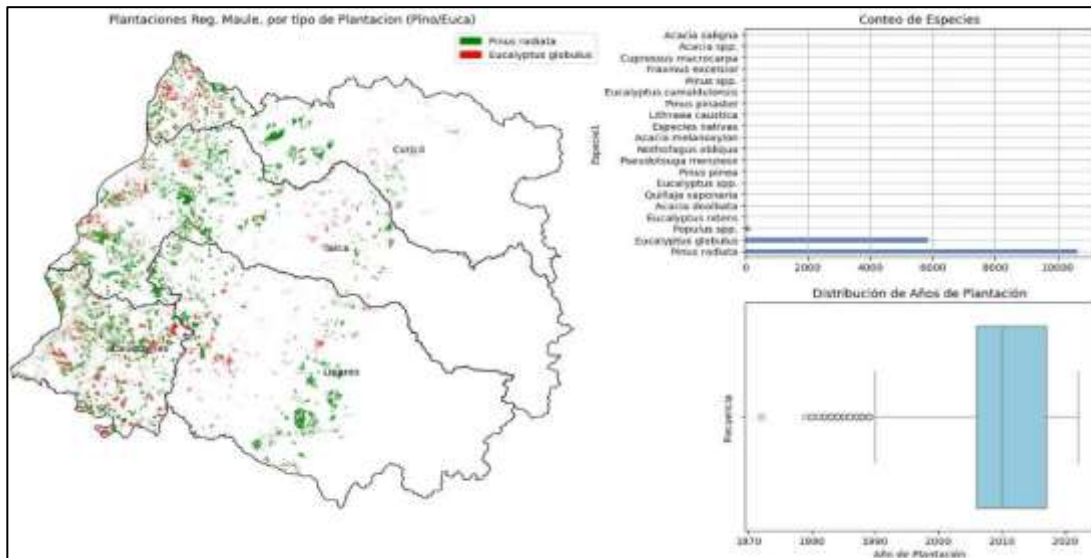


Figura 5. Gráficos descriptivos para cobertura de plantaciones

Por último, se visualiza la información a través de un histograma que muestra la frecuencia de plantaciones a lo largo de los años. Se aprecia un aumento alrededor del año 2000 y una frecuencia más baja en los años anteriores y posteriores a 2000. Algunos valores atípicos (*outliers*) están presentes antes de 1980.

Para el caso del castro vegetacional de CONAF se utiliza la del año 2016 con un total de 117.165 registros, de los cuales los atributos seleccionados corresponden a uso y geometría (Figura 6).

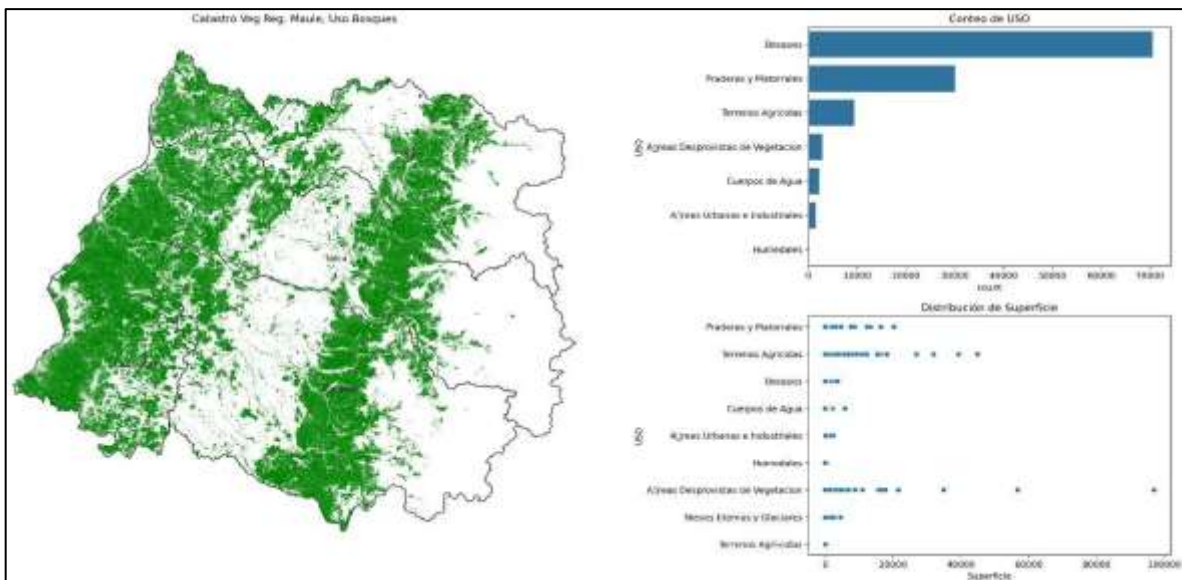


Figura 6. Gráficos descriptivos para cobertura de uso CONAF

De acuerdo al análisis descriptivo para las coberturas de CONAF presentes en la **Figura 6**, se puede señalar que en la representación el mapa muestra la cobertura vegetal o uso de la tierra, de color verde correspondiente al Uso de tipo bosque.

El segundo gráfico, el de barras, permite identificar las categorías de uso predominantes en la región, siendo "Bosques" la mayor cantidad con un 60%, seguida por "Praderas y Matorrales" de un 25% y "Terrenos Agrícolas" 7%, los otros usos de la tierra, como áreas urbanas e industriales, cuerpos de agua y humedales, tienen frecuencias mucho menores.

El tercer gráfico muestra la distribución de superficie por categoría de uso, se aprecia una concentración significativa de puntos en el extremo inferior del eje de superficie, indicando que la mayoría de los polígonos tienen superficies más pequeñas.

Los puntos dispersos a lo largo del eje X sugieren que hay menos polígonos con superficies grandes, y se aprecia una relación entre el tipo de uso y el tamaño de la superficie.

- *Paso 5: Clip de vectores en imágenes*

Una vez seleccionados los tipos y sus características, se obtienen las imágenes de los polígonos según el mosaico generado en los pasos anteriores, para ello se realiza una intersección del vector con la imagen ráster para realizar el proceso clip. Esto se realiza mediante la librería Rasterio usando el código del **Cuadro 3**.

Cuadro 3. Código de proceso clip entre vectores y raster

```
with rasterio.open(mosaic_path) as src: for _, row
    in chunk.iterrows():
        geom = row.geometry
        geom_mask = geometry_mask([geom], transform=src.transform, invert=True, out_shape=(src.height,
        src.width))
        out_image, out_transform = mask(src, [geom], crop=True)
        geom_mask = geometry_mask([geom], transform=out_transform, invert=True,
        out_shape=out_image.shape[1:])
        for i in range(out_image.shape[0]): out_image[i][~geom_mask] =
            src.nodata out_meta = src.meta.copy()
            out_meta.update({
                "driver": "GTiff",
                "height": out_image.shape[1], "width":
                out_image.shape[2], "transform":
                out_transform, "nodata": src.nodata
            })
            clip_name = f"clip_image_{os.path.basename(mosaic_path)}_{tipo}_{row.ID}.tif"
            output_path = os.path.join(output_directory, clip_name)
            with rasterio.open(output_path, "w", **out_meta) as dest: dest.write(out_image)
            print(f"Mosaico recortado guardado en {output_path}")

        results.append({'Polygon_ID':
            row.ID, 'Has_Clip': True,
            'Clip_Name': clip_name,
            'Clip_Path': output_path,
            'Image_Name': os.path.basename(mosaic_path),
            'Image_Path': mosaic_path,
        })
```

Este proceso guarda los resultados en un *dataframe* de pandas para ser analizado de manera posterior, a su vez las imágenes que son recortadas a partir del mosaico siguen manteniendo la información de todas las bandas como su formato (*.tif). Las imágenes se almacenaron en tres carpetas diferentes con el nombre de su categoría, es decir, Pinus, Eucalyptus y Bosque (CONAF). Para el caso de la última categoría, Bosque, se realizó una intersección espacial para determinar los polígonos de plantaciones que no interceptarán con el uso de bosque, esto permite no tener imágenes con dos categorías.

- *Paso 6: Aumento de datos o data augmentation para balanceo de imágenes*

Los resultados de este paso se indican en el **Cuadro 5**. A partir de ellos fue necesario aumentar las imágenes de la categoría con menos cantidad, es decir, *Eucalyptus*. Después, para que la red neuronal empleada no se vea afectada por el aumento de imágenes alteradas, se decide cortar el conjunto de datos en muestras de 10.000 datos por categoría, permitiendo tener un conjunto de datos balanceados con muestras similares.

El paso de aumento de las imágenes se realizó mediante la librería PIL tal como muestra en el código del **Cuadro 6**.

Cuadro 5. Aumento de datos para balanceo de imágenes

Especie	Numero de imágenes
<i>Eucalyptus globulus</i>	5.866
<i>Pinus radiata</i>	10.645
Uso de tipo bosques (CONAF)	70.665

Cuadro 6. Código de funciones para el aumento de datos

```
def random_rotation(image):
    """Aplica una rotación aleatoria a la imagen.""" return
    image.rotate(random.uniform(-40, 40))
def horizontal_flip(image):
    """Aplica una inversión horizontal aleatoria.""" if
    random.random() < 0.5:
        return ImageOps.mirror(image) return image
def vertical_flip(image):
    """Aplica una inversión vertical aleatoria.""" if
    random.random() < 0.5:
        return ImageOps.flip(image)
return ImageOps.flip(image)
```

- *Paso 7: Normalización de imágenes*

Luego de realizar el aumento de datos a las imágenes transformadas de *.tif a *.png se procede a normalizar el tamaño (ancho y alto) de las mismas. Esto permite tener imágenes cuyo tamaño sea inferior o igual a 64X64, y permite que las imágenes se visualicen como se observa en la **Figura 7**.

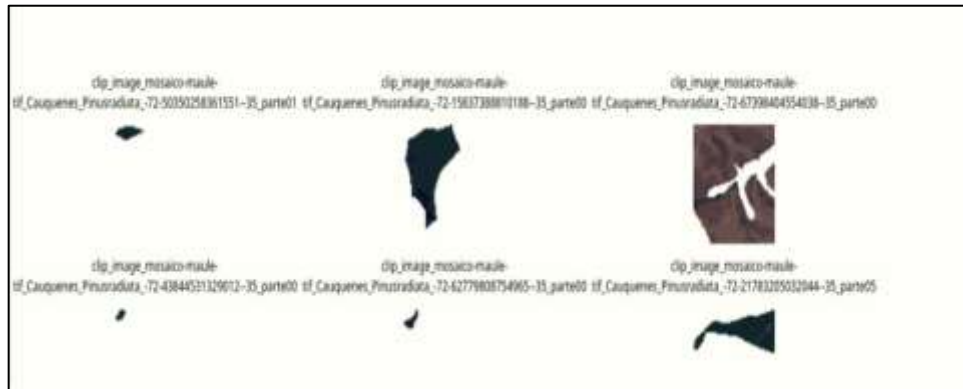


Figura 7. Miniaturas de imágenes inferiores o iguales a 64x64

- Paso 8: Selección del modelo

Para esta prueba de conceptos se utilizó un modelo de red neuronal tipo ResNet (*Residual Networks*), el cual se caracteriza por ser una red neuronal profunda que ha demostrado ser eficaz en diversos campos del aprendizaje automático, incluido el análisis de imágenes satelitales.

Esta red neuronal consta de una arquitectura basada en 50 capas de aprendizaje profundo utilizando conexiones residuales, también llamadas *skip connections*, que permite el flujo de gradientes directamente a través de la red, evitando problemas de desvanecimiento. Sus parámetros iniciales corresponden a 3 categorías, tal como muestra la Figura 8.

```

1 Model: "model_1"
2
3 Layer (type)                Output Shape              Param #                    Connected to
4 -----
5 input_2 (InputLayer)        [(None, 64, 64, 3)]      0
6
7 zero_padding2d_1 (ZeroPadding2D) (None, 76, 76, 3)        0
8
9
10 conv2d_53 (Conv2D)          (None, 32, 32, 64)       9472                       ['zero_padding2d_1[0][0]']
11
12 batch_normalization_53 (Batch Normalization) (None, 32, 32, 64)       256                       ['conv2d_53[0][0]']
13
14
15 activation_49 (Activation)  (None, 32, 32, 64)       0                          ['batch_normalization_53[0][0]']
16
17 max_pooling2d_1 (MaxPooling2D) (None, 15, 15, 64)       0                          ['activation_49[0][0]']
18
19 conv2d_54 (Conv2D)          (None, 15, 15, 64)       4160                       ['max_pooling2d_1[0][0]']
20
21 batch_normalization_54 (Batch Normalization) (None, 15, 15, 64)       256                       ['conv2d_54[0][0]']
22
23
24 activation_56 (Activation)  (None, 15, 15, 64)       0                          ['batch_normalization_54[0][0]']
25
26 ...
27 Total params: 23,593,858
28 Trainable params: 23,548,759
29 Non-trainable params: 53,129
    
```

Figura 8. Resumen de parámetros de la red neuronal ResNet-50

- Paso 9: Aplicación de la red neuronal

Para llevar a cabo esta prueba los datos se separaron en tres categorías: (i) Datos de entrenamiento, para que la red neuronal pueda llevar a cabo el proceso de aprendizaje (70%); (ii). Datos para realizar pruebas

de las predicciones llevadas a cabo por esta red (20%); y (iii) Datos utilizados para realizar la validación, a través de *cross validation* o validación cruzada, que permite generar ajustes en la red en caso de ser necesarios (10%). Para esta división se utiliza la librería de *tensorflow* que posee distintas funciones que permiten seleccionar las categorías directamente desde las carpetas donde residen las imágenes. Esto asigna una etiqueta por cada carpeta siendo la categoría que la red aplicará en cada una de sus iteraciones.

En el **Cuadro 7** se muestra el código para llevar a cabo la selección de la *data train* y *test* para la red neuronal.

Cuadro 7. Código de selección de entrenamiento y test mediante TensorFlow

```
train_dataset = datagen.flow_from_directory(batch_size=batch_size, directory=dataset_url,
shuffle=True, target_size=(img_height,
img_width), subset="training",
class_mode='categorical')

test_dataset = datagen.flow_from_directory(batch_size=16, directory=dataset_url,

shuffle=True, target_size=(img_height,img_width),
subset="validation", class_mode='categorical')
```

RESULTADOS

Los resultados obtenidos en el proceso de prueba de concepto fueron bastante alentadores, permitiendo la obtención de la métrica de *accuracy* o asertividad en un 92%. Los datos se obtuvieron con datos de entrenamiento. Adicionalmente para complementar las pruebas, se utilizó el modelo para predecir las etiquetas de la provincia de Talca, obteniendo un resultado del 90% de asertividad, tal como se muestra en la **Figura 9**. La tasa de asertividad indica que el modelo puede predecir con precisión la información relevante en un 90 % de los casos, lo que demuestra un buen desempeño en la tarea propuesta

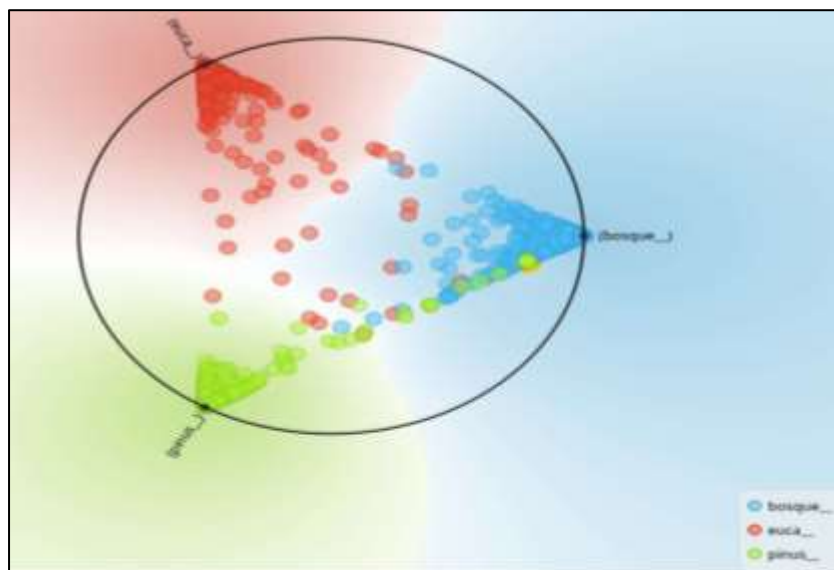


Figura 9. Resultados de predicción para la provincia de Talca, región del Maule.

DISCUSIÓN Y CONCLUSIONES

La importancia de utilizar sistemas de clasificación basados en redes neuronales, para clasificar los tipos de bosques en Chile, radica en que permite modelar la complejidad de los sistemas forestales, captando la interacción entre factores como el bosque, el clima, el suelo y la productividad. Si bien este análisis se basa en las características de solo una región, la metodología expuesta puede ser replicadas en zonas cuyas áreas fuesen similares.

Para complementar este análisis, se pueden incorporar nuevas características de diferentes fuentes al *dataset*, por ejemplo, índices como NDVI (índice de vegetación), datos de campo, información climática u otros que permitan al *dataset* tener más características de las categorías a predecir. Si bien hay estudios actuales que usan otras técnicas para la preparación de los datos, los resultados obtenidos no son significativamente diferentes.

Para adecuar la red neuronal a distintas regiones, se debe tener en cuenta la diversidad de ecosistemas y condiciones climáticas, por lo mismo se aconseja reentrenar el modelo con los datos de diferentes regiones, es decir, ampliar el modelo con más datos. Esto debido a que las redes neuronales pueden ser entrenadas utilizando datos de diversas áreas, lo que permite obtener modelos robustos y precisos

En resumen, los sistemas de clasificación basados en redes neuronales son una herramienta importante para optimizar los procesos de clasificación, y complementarlos con análisis de datos de alta dimensión, permitiendo que estos se puedan adaptar a diferentes condiciones geográficas. Estos avances permiten una mejor comprensión y gestión de los recursos forestales en el país.

REFERENCIAS

- Breiman, L. (2001).** Random forests. *Mach. Learn.*, N° 45. Pp: 5-32. <https://doi.org/10.1023/A:1010933404324>
- Lu, D., Mausel, P., Brondizio, E. & Moran, E. (2004).** Change detection techniques. *International Journal of Remote Sensing*, 25(12): 2365-2401. <https://doi.org/10.1080/0143116031000139863>
- Goodfellow, I., Bengio, J. & Courville, A. (2016).** Deep learning. MIT Press. Cambridge, MA, USA. En: <http://www.deeplearningbook.org>.
- Novak, G. (2020).** Kubeflow: How to automate a machine learning workflow using Kubeflow Pipelines. En: <https://towardsdatascience.com/machine-learning-pipelines-with-kubeflow-4c59ad05522>
- Zelin, L., Changhui, P., Work, T., Candau, JN., DesRochers, A. & Kneeshaw, D. (2018).** Application of machine-learning methods in forest ecology: recent progress and future challenges *Environmental Reviews*, 26(10) <https://doi.org/10.1139/er-2018-0034>