

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN

TECNOLOGÍAS DE LA INFORMACIÓN



"CLASIFICACIÓN DE IMÁGENES
HIPERESPECTRALES"



Junio -2024

AUTOR: Francisco Javier Quiles Ruiz

DIRECTOR/ES: Héctor Francisco Migallon Gomis

Otoniel Mario López Granado

Miguel Onofre Martínez Rach

ÍNDICE

CAPÍTULO 1 INTRODUCCIÓN	6
RESUMEN DEL PROYECTO	7
ESTADO DEL ARTE	8
CAPITULO 2 PREELIMINARES	10
2.1 TECNOLOGÍA HIPERESPECTRAL.....	13
2.2 PREPROCESAMIENTO DE LOS DATOS.....	15
2.2.1 NORMALIZACIÓN DE REFERENCIA	18
2.2.2 FILTRO SAVIZKY-GOLAY	18
2.2.3 NORMALIZACIÓN MÁXIMA	19
2.2.4 (SNV) NORMALIZACIÓN ESTÁNDAR VARIANTE.....	19
2.2.5 (MSC) MULTIPLICATIVE SCATTER CORRECTION.....	20
2.2.6 (EMSC) ELABORATED MULTIPLICATIVE SCATTER CORRECTION	20
2.2.7 (ASLS) ASYMMETRIC LEAST SQUARES.....	21
2.2.8 (WLS) WEIGHED LEAST SQUARES	22
2.2.9 (MC) MEAN CENTERING	23
2.2.10 (TSC) TWO DIMENSION SAVINZKY-GOLAY	23
2.2.11 IMAGE SEGMENTATION BY SPECTRAL ANGLE	24
2.2.12 DERIVADAS DE NORRIS WILLIS	25
2.2.13 HYPERESPECTRAL IMAGE CORRECTION WITH DOUBLE STANDARD REFLECTANCE.....	26
2.2.14 DERIVADA LOGARÍTMICA	27
2.2.15 HYPER-HUE SATURATION INTENSITY (HHSI).....	28
2.2.16 BACKGROUND BALANCING	29
2.3 TÉCNICAS DE ANÁLISIS HIPERESPECTRAL PARA LA CLASIFICACIÓN DE MATERIALES	29
2.3.1 APRENDIZAJE AUTOMÁTICO	30

2.3.2 ARTIFICIAL NEURAL NETWORK	31
2.3.3 PERCEPTRÓN MULTICAPA	34
2.3.4 SUPPORTED VECTOR MACHINE	35
2.3.5 (CNN) CONVOLUTIONAL NEURAL NETWORK.....	36
2.3.6 (KNN) K-NEAREST NEIGHBOURGS	40
2.3.7 (PLS-DA) PARTIAL LEAST SQUARES DISCRIMINANT ANALYSIS...	41
2.3.8 CLASIFICACIÓN BAYESIANA	42
2.3.9 ARBOLES DE DECISION	45
2.3.10 RANDOM FOREST	45
2.3.11 (PCA) PRINCIPAL COMPONENT ANALYSIS.....	46
2.3.12 HIERARCHICAL MODEL	47
2.3.13 HRNet+OCR (HIGH RESOLUTION NET + OBJECT CONTEXTUAL REPRESENTATION).....	48
2.3.14 (RBNN) RADIAL BASIS NEURAL NETWORK	49
2.4 MATERIALES	50
CAPÍTULO 3 SISTEMA DE ADQUISICIÓN DE DATOS	52
3.1 INFRAESTRUCTURA HARDWARE	53
3.2 ESTRUCTURA DE DATOS OBTENIDOS	53
CAPÍTULO 4 TÉCNICAS DE CLASIFICACIÓN	62
4.1 INTRODUCCIÓN.....	63
4.2 PERCEPTRÓN MULTICAPA	63
CAPÍTULO 5 ANÁLISIS DE TÉCNICAS DE CLASIFICACIÓN.....	65
5.1 PERCEPTRÓN MULTICAPA	66
5.1.1 FUENTE DE DATOS.....	66
5.1.2 CONFIGURACIÓN DE RED.....	67
5.1.3 OPTIMIZADOR.....	67
5.1.4 RESULTADOS.....	68

CAPÍTULO 6 CONCLUSIONES	70
BIBLIOGRAFÍA	72
ÍNDICE DE FIGURAS	78
ÍNDICE DE TABLAS	79
ANEXO I: CÓDIGO EMPLEADO PARA LA REALIZACIÓN DEL TRABAJO.....	80



CAPÍTULO 1

INTRODUCCIÓN

Este es el primer capítulo en el cual se describe brevemente el concepto de imagen hiperespectral y cual es el problema que se viene a solucionar. Se detalla también cual es el estado del arte que será analizado mas adelante en este mismo documento.



RESUMEN DEL PROYECTO

La alta demanda y uso extensivo de plásticos en la vida moderna está asociada con un impacto económico significativo y una seria huella ecológica. La producción de plásticos conlleva un alto consumo de energía y emisiones de CO₂, así como una gran necesidad de recursos fósiles (limitados). Debido a la alta durabilidad de los plásticos, grandes cantidades de basura plástica se acumulan en vertederos desbordantes y los desechos plásticos flotan en los océanos del mundo. Por otra parte, la incineración para generar energía libera aún más CO₂ y sustancias tóxicas a la atmósfera.

El reciclaje de productos plásticos después de su ciclo de vida puede contribuir significativamente a la reducción de los impactos ambientales y económicos. Para producir productos de reciclaje de alta calidad, se requieren composiciones monofraccionales de polímeros de desecho; es decir, composiciones formadas por un solo tipo de plástico, en lugar de ser una mezcla de diferentes tipos de plásticos reciclados. Existen varios métodos para clasificar y recoger este tipo de polímeros, desde separación manual, clasificación mecánica, flotación por densidad o tecnología por infrarrojos.

Sin embargo, las tecnologías de medición existentes, como la espectroscopia de infrarrojo cercano, presentan limitaciones en la clasificación de mezclas complejas y diferentes grados de polímeros. Las tecnologías más recientemente inventadas basadas en espectroscopia de infrarrojo medio, espectroscopia Raman o espectroscopia asistida por láser aún se encuentran en desarrollo y se espera que sean bastante costosas.

Según Gruber et al. (2019), la clasificación de residuos plásticos negros utilizando imágenes de fluorescencia y aprendizaje automático ha demostrado ser eficaz para mejorar los procesos de reciclaje [1].

En este trabajo se clasifican residuos en función de su composición química procesando imágenes hiperespectrales (HSI). Las imágenes hiperespectrales consisten en una imagen a la cual se le agregan más información en su tercera dimensión. Así, mientras que una imagen normal consta de 3 dimensiones: ancho, altura y en su tercera dimensión encontramos tres espectros; en una HSI la tercera dimensión la componen muchas bandas. En el caso de este estudio en concreto son 224.

ESTADO DEL ARTE

El reciclaje de materiales es muy importante dado los recursos finitos, el aumento de demanda y las condiciones de extracción y producción dañinas para el medio ambiente. Esto se aplica en particular a los plásticos, que actualmente se fabrican en su mayoría a partir de petróleo crudo. La tasa de reciclaje de materiales de plástico en todo el mundo en 2017 era solo del 9%, otro 11% se utiliza para el reciclaje térmico, mientras que el resto se deposita en vertederos [2]. La mayoría de los plásticos reciclados a menudo se procesan en productos secundarios de calidad inferior. La razón de esto es la falta de tecnologías de sensores adecuadas para clasificar los plásticos, lo que permitiría separar las mezclas de plástico con suficiente pureza (a veces hasta el 99.9%).

Las tecnologías de separación estándar utilizadas para clasificar los plásticos, como la clasificación mediante métodos de flotación-hundimiento [3], la separación electrostática [4], o la clasificación mediante imágenes hiperespectrales en el rango espectral del infrarrojo cercano [5][6], están en constante desarrollo. Otros métodos como la fluorescencia de rayos X (XRF) [7] son adecuados para clasificar los plásticos negros, pero están limitados a un pequeño número de combinaciones de plástico.

La tecnología clave detrás de diversas aplicaciones como el análisis de la composición del suelo en campos de cultivo es la HSI (Imágenes Hiperespectrales) con información espectral y espacial detallada, lo que hace que la HSI sea una poderosa fuente de información para métodos avanzados de clasificación como vecinos más cercanos (KNN), supported vector machine (SVM), bosques aleatorios (RF), redes neuronales (CNN) y enfoques de aprendizaje profundo (DL, por sus siglas en inglés). La comparación de estos métodos de clasificación mencionados se da en diversos estudios y muestra que no hay un clasificador que proporcione el mejor rendimiento, y que la calidad del resultado depende de factores como la disponibilidad de muestras de entrenamiento, requisitos de procesamiento, ajuste de parámetros y velocidad del algoritmo.

Otro aspecto a considerar en el contexto de los clasificadores mencionados es la Maldición de la Dimensionalidad. Si se utilizan HSI de alta dimensionalidad directamente

como entrada, la precisión de clasificación disminuye, mientras que el esfuerzo computacional del modelo tiende a aumentar exponencialmente. Para evitar este problema, es esencial realizar una reducción de dimensionalidad. La reducción de dimensiones implica que los algoritmos deben extraer automáticamente un conjunto de valores espectrales característicos que representen las características decisivas de los objetos en cuestión, a partir de toda la secuencia de una firma espectral. La cantidad de estudios existentes demuestra que no existe una metodología satisfactoria, robusta y confiable, y que este tema es uno de los principales problemas abiertos en la imagen espectral. Como consecuencia, este paso recomendado conlleva una pérdida de información en el espacio espectral, ya que las bandas seleccionadas no ofrecen una descripción precisa de la firma espectral original debido a la reducción de dimensionalidad y la posible eliminación de características espectrales relevantes [8].



CAPITULO 2

PREELIMINARES

Actualmente existen estudios que utilizan la tecnología hiperspectral, con el objetivo de analizar cual es la mejor forma optima de proceder en nuestro proyecto en el presente capitulo se realiza un análisis sobre las técnicas de preprocesamiento y de análisis hiperspectral utilizadas previamente.



En los últimos años, la tecnología de imágenes hiperespectrales ha madurado cada vez más. Con características como múltiples canales espectrales, alta resolución espectral, fuerte continuidad de bandas y "unidad de mapeo" (la capacidad de mapear información espectral a una imagen espacial), la tecnología de imágenes hiperespectrales se utiliza ampliamente en teledetección, agricultura, biomedicina y otros campos [9][10]. Los datos adquiridos mediante técnicas de imágenes hiperespectrales se denominan imágenes hiperespectrales. Su análisis permite adquirir características morfológicas de atributos externos y características espectrales de atributos internos de los componentes de la muestra, lo que permite la clasificación del objetivo. Sin embargo, debido a la alta correlación de información entre las bandas de las imágenes hiperespectrales, los siguientes factores pueden interferir en las aplicaciones prácticas:

- Información de fondo redundante en las imágenes hiperespectrales: este tipo de imágenes recoge gran cantidad de información espectral detallada que puede contener redundancias innecesarias que no aportan valor adicional. Es decir, el procesamiento de datos no relevantes puede conllevar a una menor eficiencia del análisis y a un costoso tiempo de realización [11][12].
- Ruido de corriente oscura: se caracteriza por ser un componente no deseado en las imágenes hiperespectrales que pueden inducir errores y afectar a la calidad de los datos espectrales. Principalmente se debe a un espacio de medición que no es claro, un ejemplo sería la existencia de partículas en suspensión en el aire de la zona a medir [13] [14] [15].
- Ruido dimensional espacial y espectral: por un lado, el ruido dimensional se basa en variaciones no deseadas en la respuesta de los píxeles. Por otro lado, el ruido espectral puede deberse a la interferencia atmosférica y errores en las mediciones, entre otros [16][17].

Por lo tanto, se requieren operaciones de preprocesamiento y análisis adecuados antes de clasificar las imágenes hiperespectrales para evitar los fenómenos descritos. Actualmente, se utilizan métodos como umbralización, segmentación de regiones y algoritmos de cuencas hidrográficas para eliminar la información de fondo redundante de las imágenes hiperespectrales [18][19]. El método de corrección en blanco y negro es el método más utilizado para corregir el ruido en las corrientes oscuras [20][21]. Dado que la imagen

hiperespectral es una imagen bidimensional en la dimensión espacial, las bandas individuales se pueden eliminar utilizando métodos clásicos de eliminación de ruido de imágenes; la dimensión espectral se suele suprimir mediante filtrado de Savitzky-Golay [22][23].

Aunque los métodos de preprocesamiento mencionados anteriormente se utilizan ampliamente en la clasificación de imágenes hiperespectrales, todavía presentan algunas limitaciones. Cuando la diferencia entre la escala de grises de la región donde se encuentra el objetivo de interés en la imagen hiperespectral y la escala de grises entre los píxeles circundantes es pequeña o la superposición del rango de grises es grande, el efecto de la segmentación por umbral a menudo no es satisfactorio [24].

La segmentación de regiones puede llevar a una sobresegmentación o subsegmentación cuando los parámetros no se manipulan correctamente. Algunos algoritmos tratan de erosionar de manera sucesiva varias imágenes, lo cual tiene una alta velocidad en la segmentación de imágenes, pero es muy propenso a la sobresegmentación [25].

El método de corrección en blanco y negro se divide en dos pasos: corrección en blanco y corrección en negro. En la corrección en negro, la fluctuación de la corriente oscura cambia constantemente debido a factores ambientales como la temperatura [26].

Aunque la eliminación de ruido en el dominio espacial o espectral por separado puede filtrar la mayor parte del ruido, no puede manejar bien las imágenes hiperespectrales con una relación señal-ruido baja. Por lo tanto, se necesita la eliminación de ruido en conjunto de las dimensiones espacial y espectral de las imágenes hiperespectrales [27].

En relación al problema de la información de fondo redundante en las imágenes hiperespectrales, diversos artículos han propuesto un análisis conjunto de componentes principales basado en el ángulo espectral. Este enfoque utiliza el método de umbralización del histograma, logrando que solo unos pocos píxeles no se segmenten correctamente. En segundo lugar, para resolver el problema de la inestabilidad del ruido de corriente oscura, en el artículo [9] se propone el método de corrección de placas de reflectancia estándar doble en lugar del método de corrección en blanco y negro. Se utiliza una placa de

reflectancia estándar con un 50% de reflectancia, cuya información espectral se conoce, como muestra objetivo, y se utilizan placas de reflectancia estándar con un 2% y un 98% de reflectancia como placas de calibración. La curva espectral de reflectancia después de la corrección de la placa de reflectancia bicriterio se acerca más a la verdadera curva espectral de reflectancia de la muestra y el resultado corregido mejora la precisión total de clasificación de la imagen hiperespectral en un 1,88%. Para realizar el filtrado conjunto de dimensiones espaciales y espectrales de las imágenes hiperespectrales, este trabajo propone un método conjunto de dimensiones espaciales-espectrales (TSG) basado en el principio de filtrado de Savitzky-Golay. En este estudio, se utilizó la imagen hiperespectral de la Universidad de Pavia como imagen experimental para evaluar la calidad de la imagen antes y después del filtrado SG/TSG. Los resultados muestran que, aunque el filtrado TSG modifica más la imagen original que el filtrado SG, aún conserva la mayoría de las características originales y tiene menos información de ruido después del filtrado. Al comparar los efectos del filtrado SG/TSG en los resultados de clasificación, se encontró que la precisión total de clasificación después del filtrado TSG fue significativamente mayor que después del filtrado SG, alcanzando el 98,69% [9].

2.1 TECNOLOGÍA HIPERESPECTRAL

El término "imágenes hiperespectrales" fue acuñado por Alexander F. H. Goetz en un artículo de 1985 sobre teledetección que describe el inicio de la tecnología. La HSI se desarrolló como una técnica de imágenes para identificar materiales en la superficie de la Tierra desde aeronaves y satélites. La HSI proporciona una alta resolución espectral en comparación con los sistemas de imágenes multiespectrales, como los proporcionados por los satélites Landsat. Esta mejora en la resolución espectral permitió una mejor identificación de los materiales en la superficie, especialmente los minerales en los suelos.

La gran cantidad de datos también ayudó a superar desafíos en la teledetección, como la interferencia atmosférica y la presencia de materiales mixtos en la superficie terrestre. Aunque se desarrolló en la década de 1970, la HSI no tuvo una aplicación amplia hasta que los avances técnicos permitieron almacenar y procesar fácilmente los datos [28].

Una imagen hiperespectral se basa en los mismos principios que una imagen tradicional de rojo-verde-azul (RGB). Una imagen se puede representar como una matriz con I filas y J columnas, lo que da las dimensiones $I \times J$. Estas dos dimensiones espaciales son el tamaño de la imagen. Cada entrada en la matriz es sinónimo de un píxel. A pesar del uso de "cuadrados" para representar los píxeles, los píxeles son en realidad medidas puntuales. Estos "cuadrados" se refieren más correctamente como resolución espacial, donde cada medida puntual (píxel) representa un área de un tamaño discreto. Cada píxel representa una posición en el espacio real que refleja y absorbe luz en todo el espectro electromagnético. La luz reflejada se registra como un número que indica la intensidad [29].

Si solo se muestrea una longitud de onda, el resultado es una imagen en escala de grises donde alta intensidad se representa como blanco y baja intensidad como negro. Una imagen en color utiliza tres bandas de longitud de onda para describir la cantidad de luz en una región del espectro electromagnético que está siendo reflejada por un objeto. Estas bandas de color se encuentran dentro del espectro de luz visible para el ojo humano (400-800 nm) y corresponden a la luz visible en rojo, verde y azul. Estas regiones se eligen debido a la sensibilidad del ojo humano a estos fotopigmentos. Cada píxel en una imagen se expresa como una combinación de tres valores de intensidad que representan rojo, verde y azul (RGB), y la matriz ahora tiene tres dimensiones [30].

Cuando el número de bandas espectrales es mayor a 100, el cubo de datos resultante se vuelve hiperespectral. El artículo fundacional que describe HSI por Goetz et al. [8] define HSI como "la adquisición de imágenes en cientos de bandas espectrales contiguas y registradas, de tal manera que para cada píxel se pueda derivar un espectro de radiancia".

Esta definición excluye la imagen multiespectral, ya que las bandas no son necesariamente contiguas. El resultado de este muestreo contiguo se visualiza mejor como un espectro. Este espectro es una representación altamente detallada de las diferentes longitudes de onda de la luz reflejada por un objeto, y por lo tanto permite la clasificación de píxeles según su composición química en lugar de solo su color [10] [31].

2.2 PREPROCESAMIENTO DE LOS DATOS

Una vez destacada la importancia y puesto el contexto el porqué del preprocesamiento de datos, se realizará una descripción de los métodos más utilizados en el preprocesamiento de imágenes hiperespectrales.

Los datos hiperespectrales pueden ser tratados según diferentes puntos de vista espaciales-espectrales. La mayoría de los métodos de aprendizaje profundo tempranos solo explotan los datos píxel por píxel (enfoques unidimensionales), trabajando en la dirección espectral. Esto se puede hacer extrayendo firmas espectrales de píxeles individuales o de grupos de ellos que rodean un píxel central o que pertenecen a un área de objeto. El enfoque posterior generalmente requiere algún conocimiento a priori y una fase de preprocesamiento para detectar el objeto de interés (mediante segmentación). Por ejemplo, en el artículo [32], se utiliza una transformación de distancia del coseno espectral para identificar y ponderar los píxeles que pertenecen a objetos de interés en una aplicación biomédica.

La reducción de dimensionalidad se utiliza para abordar la redundancia de la información espectral. De las diferentes técnicas de reducción de dimensionalidad, el Análisis de Componentes Principales (PCA) sigue siendo una forma clásica de proceder. Dependiendo del contexto, también se pueden utilizar otros enfoques, como Análisis de Componentes Independientes (ICA, por sus siglas en inglés) y autoencoders apilados. En el caso de aplicar un proceso bidireccional, generalmente también se realiza una reducción de dimensionalidad preliminar [33] [34].

Los datos de HSI, se pueden manejar en su totalidad con el objetivo de extraer características espaciales y espectrales (tridimensionales). Algunos de estos enfoques aún utilizan una etapa de preprocesamiento para acondicionar los datos, pero a menudo el objetivo final es trabajar directamente en los hipercubos "crudos". Dado que esto puede ser un método computacionalmente costoso y complejo, a menudo se prefiere operar con conjuntos de datos 3D (es decir, subvolúmenes, analizando diversas bandas) [35].

En la figura 1 podemos ver un conjunto de datos unificado, mientras que en la figura 2 vemos como es este mismo conjunto de datos, pero con los datos separados en diversas capas.

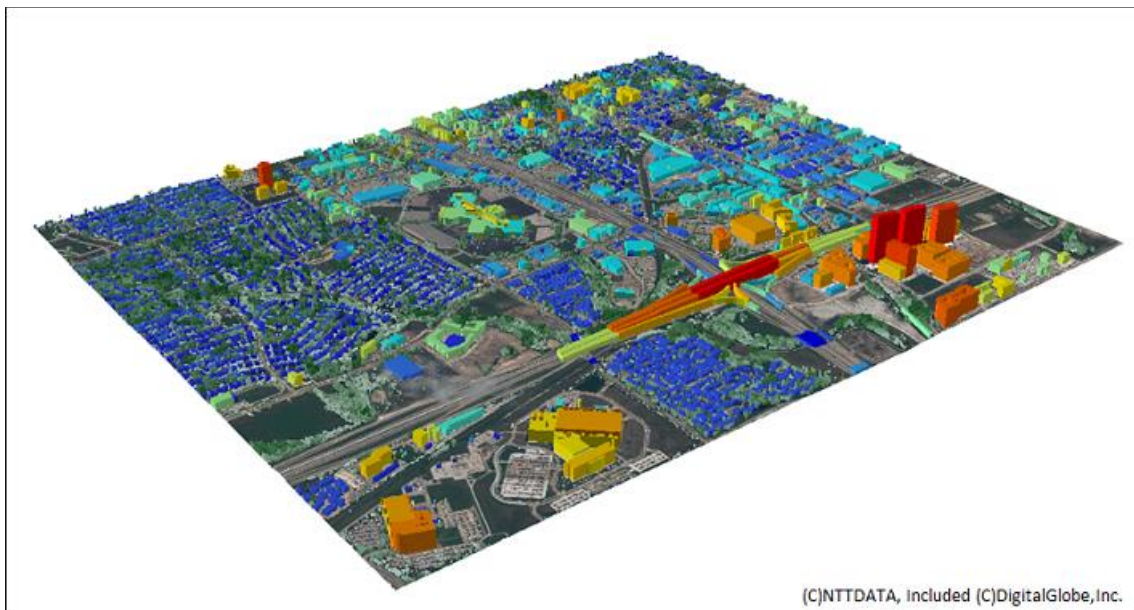


Figura 1 Conjunto de datos unificado

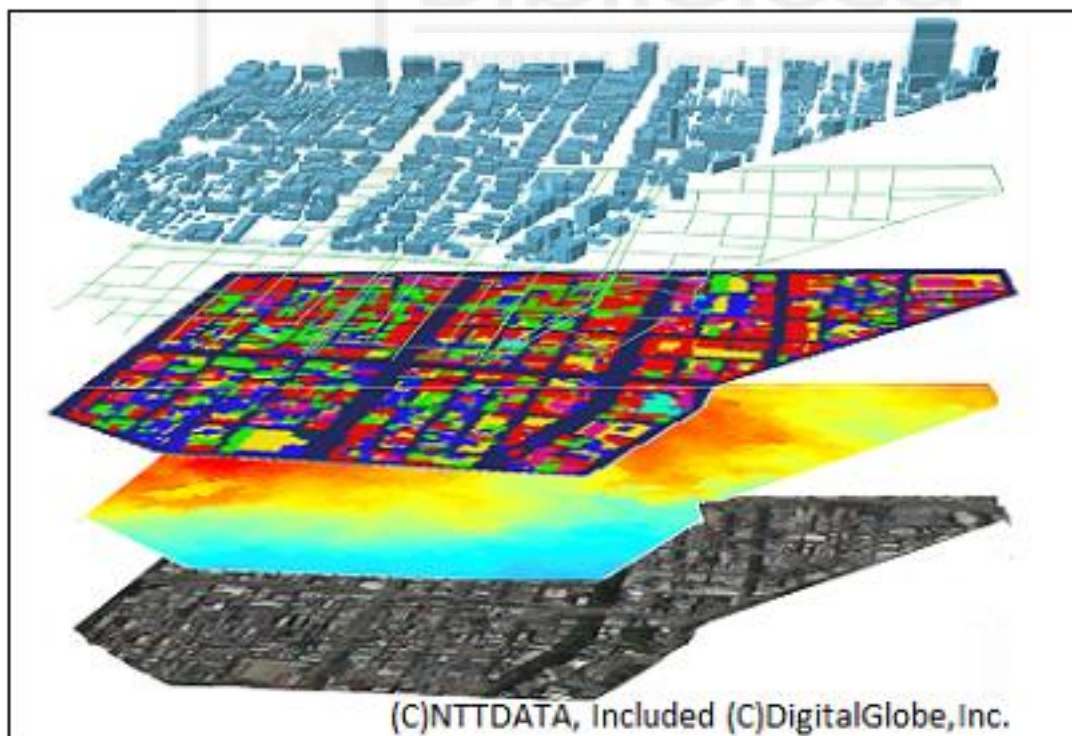


Figura 2 Conjunto de datos separados en capas

Como en la espectroscopia clásica, la calidad de las medidas en bruto a menudo se ve afectada por el ruido o por variaciones instrumentales que pueden alterar el análisis y las conclusiones que se pueden obtener con un posterior procesamiento de datos. Por lo tanto, es obligatorio realizar un buen preprocesamiento espectral del cubo de datos antes de aplicar cualquier otro método, incluso cuando se va a monitorear una sola longitud de onda. Las medidas hiperespectrales basadas en espectroscopia NIR, mid-IR y Raman suelen verse afectadas por dos factores: el ruido espectral (debido a la relación señal-ruido del espectrómetro) y la deriva de línea base (debido a la dispersión de la luz en el NIR y mid-IR, y a la influencia de la fluorescencia en el espectro Raman). Para reducir el ruido, generalmente se utilizan técnicas de suavizado basadas en el enfoque de Savitzky-Golay.

Otro enfoque que se utiliza ampliamente se basa en el ajuste gradual del canal espectral a una función polinómica (las funciones de segundo orden son las más comunes), dependiendo de la selección de un tamaño de ventana. Menos utilizado es el filtrado basado en wavelet o el análisis de componentes principales (PCA) para la eliminación de ruido de los espectros NIR y mid-IR [36]. Para corregir las derivas de línea base en los espectros NIR y mid-IR, se emplean con frecuencia métodos como el variado normal estándar, la corrección de dispersión multiplicativa y la corrección extendida de dispersión multiplicativa. Los primeros y segundos derivados de Savitzky-Golay también se utilizan para la eliminación del ruido y la corrección de línea base. Además, pueden aumentar algunas diferencias entre los espectros. Se han sugerido algunas otras alternativas (como la normalización de picos, mínimos cuadrados ponderados y mínimos cuadrados asimétricos) y han demostrado su eficacia [12].

El propósito del paso de preprocesamiento puede variar, como mejorar el análisis exploratorio, aplicar la ley de Lambert-Beers para modelos de calibración bilineales o mejorar modelos posteriores de clasificación.

Sin embargo, es esencial seleccionar las técnicas de preprocesamiento cuidadosamente para evitar el sobreajuste o la eliminación de información valiosa. No se recomienda aplicar pasos de preprocesamiento, (sin embargo, por la naturaleza de los problemas que enfrentamos es posible que a veces nos resulte más satisfactorio el uso de varias técnicas aplicadas en el preprocesamiento) y la complejidad del modelo debe gestionarse

cuidadosamente durante el preprocesamiento para asegurar la robustez en predicciones futuras.

En general, la elección de técnicas de preprocesamiento en la espectroscopia NIR es fundamental para obtener resultados precisos y confiables en el análisis de datos y tareas de modelado posteriores.

Ahora vamos a nombrar algunas de las técnicas más comunes utilizadas en el preprocesamiento de imágenes hiperespectrales.

2.2.1 NORMALIZACIÓN DE REFERENCIA

La normalización de referencia es una técnica de transformación de datos que se utiliza para escalar los datos para que se puedan representar en un rango más pequeño. Por ejemplo, si se tienen datos que varían entre 0 y 1000, la normalización de referencia puede escalarlos para que varíen entre 0 y 1. Esto es útil para evitar que los datos con valores más grandes dominen los datos con valores más pequeños [37].

El objetivo principal de esta técnica es asociar formas similares a los mismos datos en una única forma de datos. Como, por ejemplo, podemos coger datos específicos como “Número”, “Num.”, “Nro.”, “Nº” o “#” y normalizarlos a “Número” en todos los casos.

2.2.2 FILTRO SAVIZKY-GOLAY

Un filtro Savitzky-Golay es un filtro digital que se aplica a un conjunto de puntos de datos digitales con el objetivo de suavizar los datos, es decir, aumentar la precisión de los datos sin distorsionar la tendencia de la señal.

Para aplicar el filtro de Savitzky-Golay, se deben seguir los siguientes pasos:

- Seleccionar el tamaño de la ventana del filtro: esto se refiere al número de puntos de datos que se utilizan para calcular el nuevo valor de cada punto. Si se selecciona

un tamaño de ventana más grande, el filtro será más suave, pero también será menos sensible a los cambios en la señal.

- Seleccionar el grado del polinomio: esto se refiere al grado del polinomio que se utiliza para capturar características importantes de la señal original, como los máximos y mínimos relativos, así como el ancho de los picos [38].

2.2.3 NORMALIZACIÓN MÁXIMA

La Normalización Máxima es una variante específica de la normalización de datos. Esta técnica se centra en ajustar la escala de las características numéricas para que sean comparables entre sí, generalmente transformando los datos para que estén en un rango entre 0 y 1. La Normalización Máxima es especialmente útil cuando las características tienen diferentes rangos y deseamos estandarizarlas a una escala común.

Para normalizar un conjunto de datos utilizando la normalización máxima, sigue los siguientes pasos:

- Encuentra el valor máximo en el conjunto de datos.
- Divide cada valor en el conjunto de datos por el valor máximo.
- El resultado es el valor normalizado.

A diferencia de la Normalización de Referencia, que se puede utilizar para estandarizar diferentes formas de representación de datos textuales, la normalización máxima se aplica exclusivamente a características numéricas para eliminar cualquier sesgo introducido por la magnitud y escala de las características [39].

2.2.4 (SNV) NORMALIZACIÓN ESTÁNDAR VARIANTE

La Normalización Estándar Variante (SNV) es una técnica de preprocesamiento que se utiliza en espectroscopía para corregir los efectos de escala o ganancia en las variables medidas para una muestra. La SNV es una normalización ponderada que calcula la desviación estándar de todas las variables agrupadas para la muestra dada. La SNV es útil

para corregir los efectos de escala que se deben a propiedades de interés (como la concentración) y para separar los efectos sistemáticos interferentes. La SNV también ayuda a dar a todas las muestras un impacto igual en el modelo [40].

Para aplicar la SNV, primero se calcula la media y la desviación estándar de todas las variables para la muestra dada. Luego, se resta la media de cada variable y se divide por la desviación estándar. Con ello normalizamos las variables para que tengan una media de cero y una desviación estándar de uno. La SNV se puede aplicar a una sola muestra o a un conjunto de muestras.

2.2.5 (MSC) MULTIPLICATIVE SCATTER CORRECTION

La Multiplicative Scatter Correction (MSC) es una técnica de normalización que se utiliza en espectroscopía para corregir los efectos de la dispersión de la luz y los cambios en la longitud de trayecto óptico en las mediciones de una muestra. La MSC es una técnica de normalización clásica utilizada para corregir los espectros de forma que sean similares a un espectro de referencia. La MSC es útil para corregir los efectos de la dispersión de la luz y los cambios en la longitud de trayecto óptico en las mediciones de una muestra.

Para aplicar la MSC, primero se debe seleccionar un espectro de referencia. Luego, se calcula la media de los espectros de la muestra y se divide por el espectro de referencia. Esto normaliza los espectros para que sean lo más similares posible al espectro de referencia. La MSC se puede aplicar a una sola muestra o a un conjunto de muestras [41].

2.2.6 (EMSC) ELABORATED MULTIPLICATIVE SCATTER CORRECTION

El objetivo de la Elaborated Multiplicative Scatter Correction (EMSC) es corregir los efectos de la dispersión de la luz y los cambios en la longitud de trayecto óptico en las mediciones de una muestra. La EMSC es una técnica de normalización avanzada que busca ajustar los espectros para que sean lo más similares posible a un espectro de referencia. A diferencia de la técnica de MSC, la EMSC no solo corrige los efectos de la

dispersión de la luz y los cambios en la longitud de trayecto óptico, sino que también aborda los efectos de la absorción de la luz y la reflexión especular.

Para calcular la EMSC, primero se debe seleccionar un espectro de referencia. Luego, se calcula la media de los espectros de la muestra y se divide por el espectro de referencia. Esto normaliza los espectros para que sean lo más similares posible al espectro de referencia. La EMSC se puede aplicar a una sola muestra o a un conjunto de muestras.

2.2.7 (ASLS) ASYMMETRIC LEAST SQUARES

La técnica Asymmetric Least Squares (AsLS) se utiliza para estimar los coeficientes de modelos de regresión lineal, donde los coeficientes se basan en medidas de ubicación definidas por una función de criterio de cuadrados mínimos asimétricos. Los estimadores de cuadrados mínimos asimétricos tienen propiedades análogas a los estimadores de cuantiles de regresión, pero son mucho más simples de calcular, al igual que las estadísticas de prueba correspondientes.

Los estimadores de coeficientes se pueden utilizar para construir estadísticas de prueba para la homocedasticidad y la simetría condicional de la distribución del error. Estas pruebas se comparan favorablemente con otras pruebas comúnmente utilizadas de estas hipótesis nulas en términos de eficiencia relativa local. En consecuencia, la estimación de cuadrados mínimos asimétricos proporciona un método conveniente y eficiente para resumir la distribución condicional de una variable dependiente dada los regresores, y para probar si un modelo lineal es una caracterización adecuada del “valor típico” para esta distribución condicional.

Para calcular los coeficientes de un modelo de regresión lineal utilizando la técnica Asymmetric Least Squares (AsLS), se utilizan medidas de ubicación definidas por una función de criterio de cuadrados mínimos asimétricos. Los estimadores de cuadrados mínimos asimétricos tienen propiedades análogas a los estimadores de cuantiles de regresión, pero son mucho más simples de calcular, al igual que las estadísticas de prueba correspondientes. Los estimadores de coeficientes pueden usarse para construir estadísticas de prueba para la homocedasticidad y la simetría condicional de la

distribución del error, y encontramos que estas pruebas se comparan favorablemente con otras pruebas utilizadas de estas hipótesis nulas en cuanto a eficiencia relativa local. En consecuencia, la estimación de cuadrados mínimos asimétricos proporciona un método conveniente y relativamente eficiente para resumir la distribución condicional de una variable dependiente dada los regresores, y un medio para probar si un modelo lineal es una caracterización adecuada del “valor típico” para esta distribución condicional.

2.2.8 (WLS) WEIGHED LEAST SQUARES

Los mínimos cuadrados ponderados son una técnica de regresión que se usa cuando los datos no cumplen el supuesto de homocedasticidad, es decir, que la varianza de la variable dependiente no es constante. También se usa cuando se quiere dar más peso a ciertas observaciones que a otras, por ejemplo, por su precisión o relevancia. La idea es minimizar la suma de los cuadrados de los residuos multiplicados por unos pesos no negativos.

Los mínimos cuadrados ponderados se usan para ajustar una función a un conjunto de datos cuando la varianza de la variable dependiente no es constante o cuando se quiere dar más peso a algunas observaciones que a otras. Para usar este método, se necesita lo siguiente:

- Una familia de funciones candidatas a ajustar, por ejemplo, una línea recta, una parábola, una exponencial, etc.
- Unos pesos no negativos que reflejen la precisión o relevancia de cada observación, por ejemplo, el inverso de la varianza, la frecuencia, el costo, etc.
- Un software estadístico que pueda calcular los coeficientes de la función elegida que minimicen la suma ponderada de los cuadrados de los residuos, es decir, las diferencias entre los valores observados y los valores predichos por la función.

2.2.9 (MC) MEAN CENTERING

El principal objetivo del Mean Centering (MC) es transformar los datos para que tengan una media de cero. Esto puede tener varias ventajas, como mejorar la interpretación de los coeficientes de regresión, reducir la multicolinealidad, o facilitar el cálculo de algunas medidas de distancia o similitud.

El MC es una forma de cambiar los números para que sean más fáciles de comparar. Supongamos que tenemos una caja con diferentes frutas: manzanas, naranjas, plátanos y uvas. Cada fruta tiene un peso diferente, por ejemplo, una manzana pesa 150 gramos, una naranja pesa 200 gramos, un plátano pesa 120 gramos y una uva pesa 10 gramos. Si quieres saber cuál es la fruta más pesada, puedes comparar los pesos directamente. Pero si quieres saber cuál es la fruta más grande, los pesos no te sirven, porque dependen de la densidad de la fruta. Entonces, lo que puedes hacer es restar el peso medio de todas las frutas a cada peso. El peso medio de las frutas se calcula sumando todos los pesos y dividiendo el resultado por el número total de frutas. En este caso, el peso medio es $(150+200+120+10)/4=120$ gramos. Al restar el peso medio a cada peso individual, obtenemos valores que indican cuánto se desvía cada fruta del peso medio. Este proceso se llama centramiento de la media.

2.2.10 (TSC) TWO DIMENSION SAVINZKY-GOLAY

El objetivo de Two Dimensions Savinzky Golay (TSC) es aplicar un filtro de suavizado y diferenciación a datos bidimensionales, como imágenes o superficies. Este filtro se basa en ajustar polinomios de bajo grado a subconjuntos de datos adyacentes, usando el método de mínimos cuadrados. El filtro de TSC permite preservar las características de los datos originales, como los máximos, mínimos y anchos de los picos, sin introducir distorsiones en la tendencia de la señal.

Para realizar el filtro de TSC, se necesita lo siguiente:

- Un conjunto de datos bidimensionales, como una imagen o una superficie, que se quiere suavizar o diferenciar.
- Un tamaño de ventana, que es el número de filas y columnas de la submatriz de datos que se va a ajustar con un polinomio.
- Un orden del polinomio, que es el grado máximo de los términos de x e y que se van a usar en el ajuste.
- Unos coeficientes de convolución, que son los valores que se multiplican por cada dato de la submatriz para obtener el valor suavizado o diferenciado en el centro de la ventana.

El proceso es el siguiente:

- Se recorre la matriz de datos con la ventana, de forma que cada dato se sitúe en el centro de la ventana al menos una vez.
- Se aplica el filtro de convolución a cada submatriz de datos, es decir, se multiplica cada dato por su coeficiente correspondiente y se suma el resultado.
- Se guarda el valor obtenido en una nueva matriz, que tendrá el mismo tamaño que la original, pero con los datos suavizados o diferenciados.

Los coeficientes de convolución se pueden calcular de forma analítica o numérica, usando el método de mínimos cuadrados.

2.2.11 IMAGE SEGMENTATION BY SPECTRAL ANGLE

El objetivo de la image segmentation by spectral angle es dividir una imagen en regiones que tengan un ángulo espectral similar. El ángulo espectral es una medida de la similitud entre dos vectores que representan el espectro de cada píxel. Esta técnica se usa principalmente para el análisis de imágenes hiperespectrales, que son imágenes que tienen cientos de bandas espectrales, y que se usan para la detección de materiales, la clasificación de terrenos, la identificación de objetos, etc.

Para calcular la image segmentation by spectral angle, se necesita lo siguiente:

- Una imagen hiperespectral, que es una imagen que tiene muchas bandas espectrales, y que se usa para ver cosas que no se ven a simple vista.

- Un ángulo espectral, que es una medida de la similitud entre dos píxeles basada en sus espectros. Cuanto más pequeño sea el ángulo, más parecidos son los píxeles.
- Un número de regiones, que es el número de grupos que se quiere formar con los píxeles de la imagen.
- Un algoritmo de clustering, que es un método que agrupa los píxeles según su ángulo espectral.

El proceso es el siguiente:

- Se convierte la imagen hiperespectral en una matriz de datos, donde cada fila representa un píxel y cada columna representa una banda espectral.
- Se calcula el ángulo espectral entre cada par de píxeles, usando la fórmula:

$$\theta_{ij} = \arccos\left(\frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}\right)$$

donde x_i y x_j son los vectores que representan los espectros de los píxeles i y j , y \cdot es el producto escalar.

- Se aplica el algoritmo de clustering a la matriz de ángulos espectrales, asignando cada píxel a una de las regiones definidas.
- Se obtiene la imagen segmentada, donde cada región tiene un color diferente.

2.2.12 DERIVADAS DE NORRIS WILLIS

Las derivadas de Norris-Williams son una técnica utilizada en el preprocesamiento de datos espectrales para mejorar la calidad de los datos antes del análisis. Esta técnica se aplica para eliminar el ruido y mejorar la señal, lo que facilita la identificación de características relevantes en los datos espectrales.

El proceso de preprocesamiento mediante derivadas de Norris-Williams incluye los siguientes pasos:

- Aplicación de la derivada para resaltar cambios rápidos en la señal.

- Eliminación de tendencias y variaciones lentas en los datos que pueden ocultar características importantes.
- Mejora de la resolución espectral para una mejor identificación de picos y valles en los datos.

Por ejemplo, al aplicar derivadas de Norris-Williams a un conjunto de datos espectrales de una muestra, se pueden resaltar las diferencias entre picos de absorción y eliminación de fondo, lo que mejora la precisión de los análisis subsecuentes.

Esta técnica es particularmente útil en la espectroscopía y otras aplicaciones de imágenes hiperespectrales, donde la calidad de los datos es crucial para obtener resultados precisos y confiables.

2.2.13 HYPERESPECTRAL IMAGE CORRECTION WITH DOUBLE STANDARD REFLECTANCE

El objetivo de Hyperspectral Image Correction With Double Standard Reflectance Plates es corregir las distorsiones ópticas y atmosféricas que afectan a las imágenes hiperespectrales, usando dos placas de reflectancia estándar con reflectancias del 2 y del 98%. Estas placas se usan como información espectral previa para la corrección de las muestras con información espectral verdadera conocida. El método se basa en el algoritmo de mínimos cuadrados ponderados y el modelo de transferencia radiativa.

El proceso para realizar la corrección de imágenes hiperespectrales con dos placas de reflectancia estándar es el siguiente:

- Se coloca una placa de reflectancia del 2% y otra del 98% en la escena que se va a capturar con el sensor hiperespectral, de forma que se puedan ver en la imagen.
- Se adquiere la imagen hiperespectral y se extraen los espectros de las dos placas, que se usan como referencia para la corrección.
- Se aplica el algoritmo de mínimos cuadrados ponderados para estimar los parámetros del modelo de transferencia radiativa, que relaciona la reflectancia de la superficie con la radiación medida por el sensor.

- Se usa el modelo de transferencia radiativa para corregir los efectos ópticos y atmosféricos que distorsionan la imagen hiperespectral, obteniendo una imagen corregida con los valores de reflectancia verdadera de la superficie.

2.2.14 DERIVADA LOGARÍTMICA

El objetivo del derivado logarítmico en imágenes hiperespectrales es realzar las características espectrales de los píxeles, reduciendo el efecto de la iluminación y el ruido.

El derivado logarítmico se calcula aplicando el logaritmo natural a cada banda espectral y luego derivando el resultado. Este método se usa para mejorar la clasificación y la detección de materiales en las imágenes hiperespectrales.

Supongamos que se desea comparar dos imágenes hiperespectrales de la misma escena, pero tomadas en momentos diferentes. Por ejemplo, se pretende analizar cómo ha cambiado el color de las hojas de los árboles en otoño. Para llevar a cabo esta comparación, es necesario medir la diferencia entre las bandas espectrales de cada imagen. Una técnica para lograrlo es utilizando la derivada logarítmica, que es una operación matemática que indica cómo cambia una banda espectral en respuesta a cambios en el color. La derivada logarítmica se calcula de la siguiente manera:

- Primero, aplicas una función llamada logaritmo natural a cada banda espectral. Esta función te ayuda a reducir los números grandes y a aumentar los números pequeños, para que sean más fáciles de comparar. Por ejemplo, el logaritmo natural de 100 es 4.6, y el de 0.01 es -4.6. Así, los números que están más cerca de 1 se quedan más o menos igual, y los que están más lejos se acercan más.
- Segundo, aplicas una función llamada derivada a cada banda logarítmica. Esta función te dice cómo cambia la banda logarítmica cuando cambia el color. Por ejemplo, si la banda logarítmica aumenta mucho cuando el color pasa de rojo a naranja, la derivada será un número positivo grande. Si la banda logarítmica disminuye mucho cuando el color pasa de verde a azul, la derivada será un número negativo grande. Si la banda logarítmica no cambia mucho cuando el color cambia, la derivada será un número cercano a cero.

- Tercero, obtienes el derivado logarítmico como el resultado de la derivada de cada banda logarítmica. Este resultado te indica cómo cambia la banda espectral cuando cambia el color, pero de una forma más fácil de comparar.

El derivado logarítmico te ayuda a ver las diferencias entre las imágenes hiperespectrales, sin que te confundan la iluminación o el ruido. Así, puedes clasificar o detectar los objetos que aparecen en las imágenes según sus propiedades espectrales.

2.2.15 HYPER-HUE SATURATION INTENSITY (HHSI)

El algoritmo Hyper-Hue Saturation Intensity (HHSI) es una técnica de procesamiento de imágenes que se utiliza para mejorar la calidad de las imágenes en color. El modelo HHSI es una extensión del modelo de color HSI (Hue, Saturation, Intensity) que se utiliza para representar los colores en una imagen. El modelo HSI se basa en la idea de que los colores se pueden describir en términos de tres componentes: matiz, saturación e intensidad. El matiz se refiere al color en sí mismo, la saturación se refiere a la pureza del color y la intensidad se refiere a la cantidad de luz que contiene el color.

El algoritmo HHSI se aplica de la siguiente manera:

- Se transforma la imagen hiperespectral en una imagen de tres canales, usando el modelo de color HSI (Hue, Saturation, Intensity), que representa el tono, la saturación y la intensidad de cada píxel.
- Se calcula el canal H (Hue) como el ángulo entre el vector espectral de cada píxel y un vector de referencia, que puede ser el promedio de todos los vectores espectrales o el vector de un material de interés.
- Se calcula el canal S (Saturation) como la distancia entre el vector espectral de cada píxel y el eje de intensidad, que es el vector que tiene la misma magnitud en todas las bandas espectrales.
- Se calcula el canal I (Intensity) como la media de todas las bandas espectrales de cada píxel.
- Se obtiene la imagen HHSI como el resultado de combinar los tres canales H, S e I.

El algoritmo HHSI permite resaltar las diferencias espectrales entre los píxeles, sin que se vean afectadas por la iluminación o el ruido. Así, se puede visualizar o clasificar mejor los objetos o materiales que aparecen en la imagen hiperespectral.

2.2.16 BACKGROUND BALANCING

El objetivo del background balancing es eliminar el efecto del fondo o del ruido en las imágenes hiperespectrales, que son imágenes que tienen muchos colores o espectros. El background balancing se basa en estimar el espectro del fondo y restarlo al espectro de cada píxel, para obtener el espectro verdadero de la superficie. Este método se usa para mejorar la clasificación y la detección de materiales en las imágenes hiperespectrales.

Para calcular el background balancing, se siguen estos pasos:

- Se estima el espectro del fondo, que es el espectro que tiene el píxel más oscuro de la imagen hiperespectral, que se supone que no contiene ningún material de interés.
- Se resta el espectro del fondo al espectro de cada píxel de la imagen, obteniendo el espectro corregido de la superficie.
- Se normaliza el espectro corregido de cada píxel, dividiéndolo por su suma total, para obtener el espectro relativo de la superficie.

2.3 TÉCNICAS DE ANÁLISIS HIPERESPECTRAL PARA LA CLASIFICACIÓN DE MATERIALES

El trabajo presentado se centra en la clasificación de materiales analizando la información hiperespectral obtenida mediante una cámara hiperespectral, que será descrita en la sección 3.1. En la literatura existen diversos métodos para clasificar materiales mediante el análisis de imágenes hiperespectrales.

Cuando se trabaja con imágenes RGB, las cuales constan únicamente de tres canales espectrales, la clasificación se realiza basándose en la información espacial. Pero hay que

tener en cuenta que una imagen hiperespectral consta de varios cientos de canales espectrales, por lo que en este tipo de imágenes es común trabajar a nivel de pixel.

A continuación, analizaremos algunos métodos utilizados para la clasificación de imágenes espectrales. Estos métodos pueden dividirse en dos grupos, un primer grupo que consta de algoritmos determinísticos y un segundo grupo que hace uso de técnicas de inteligencia artificial.

2.3.1 APRENDIZAJE AUTOMÁTICO

Para extraer información útil de un conjunto de datos, se pueden utilizar modelos estadísticos. Otra forma de extraer la información es mediante el uso de algoritmos que pueden aprender de los datos. Esto se conoce como aprendizaje automático, que es más comúnmente utilizado cuando se trabaja con conjuntos de datos de mayor dimensionalidad.

Cuando se trabaja con datos en forma de imágenes, el tipo de problema más común a resolver es la clasificación. Dado que una imagen RGB solo consta de tres canales espectrales, tiene sentido realizar la clasificación basándose en la información espacial, y un modelo de aprendizaje automático ampliamente utilizado para eso es las redes neuronales convolucionales (CNN, por sus siglas en inglés).

Sin embargo, para imágenes hiperespectrales, que pueden tener varios cientos de dimensiones espectrales, resulta más interesante centrarse en las dimensiones espectrales en lugar de las espaciales. En cambio, un algoritmo de aprendizaje automático aprendería a clasificar los píxeles de la imagen de forma independiente. Un problema crítico a tener en cuenta, sin embargo, es la gran cantidad de datos debido a la elevada dimensionalidad espectral. Como ejemplo, una imagen de tamaño 1024*1024 píxeles agregarían 1 048 576 muestras al conjunto de datos. Utilizando 448 canales espectrales, los datos en bruto recibidos de la cámara ya serían de alrededor de 1 GB para una sola imagen. Después del preprocesamiento, el tamaño podría duplicarse o incluso reducirse a la cuarta parte dependiendo de la precisión aritmética.

2.3.2 ARTIFICIAL NEURAL NETWORK

Una red neuronal es un modelo de inteligencia artificial que imita el funcionamiento del cerebro humano, mediante capas de nodos o neuronas artificiales que se conectan entre sí y procesan información en forma de señales eléctricas y químicas.

El uso de Redes Neuronales Artificiales (ANN, por sus siglas en inglés) ha aumentado considerablemente en popularidad desde que el aprendizaje profundo comenzó a mostrar su utilidad, especialmente en visión por computadora. Por ejemplo, aplicado a las imágenes hiperespectrales, ANNs se han sido utilizadas en aplicaciones alimentarias como el control de calidad del cereal, o el cuidado de terrenos agrícolas.

En una red neuronal artificial, se crean neuronas (a menudo llamadas nodos) y se conectan entre sí a través de conexiones ponderadas. Se calcula un valor basado en la entrada a una neurona, que a su vez se convierte en la entrada de otras neuronas. Simulando así las conexiones sinápticas del cerebro humano [13]. Un ejemplo del mismo se puede ver en la figura 3.

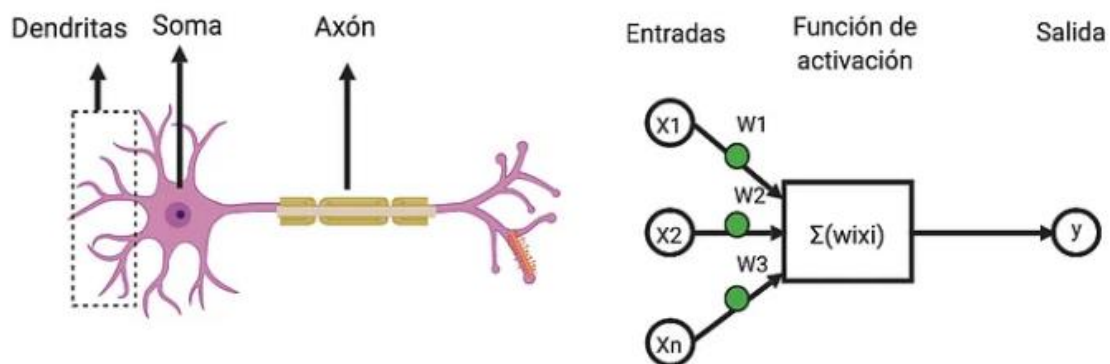


Figura 3 comparación de una neurona biométrica vs neurona artificial

Estas conexiones sinápticas en nuestro modelo las simulamos con una función de activación. Una función de activación es una función que transmite la información generada por la combinación lineal de los pesos y las entradas de una neurona artificial.

La función de activación puede ser lineal o no lineal, y determina la salida de la neurona y su grado de activación. Algunas funciones de activación comunes son:

- **Sigmoide:** Es una función que tiene forma de S y que mapea cualquier valor real a un rango entre 0 y 1. Se usa para problemas de clasificación binaria o para modelar probabilidades. En la figura 4 se puede apreciar una gráfica de activación sigmoide.

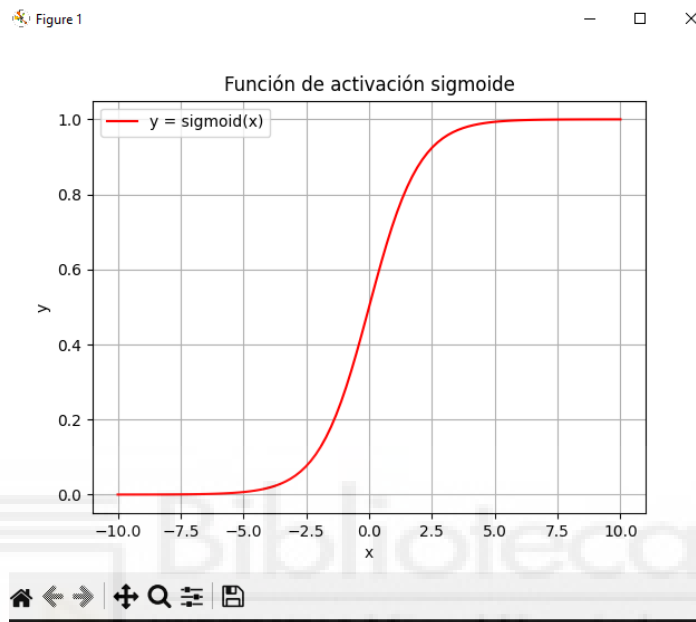


Figura 4 Gráfica función de activación sigmoide

- **Tangente hiperbólica:** Es una función que tiene forma de S y que mapea cualquier valor real a un rango entre -1 y 1. Se usa para problemas de clasificación multiclase o para modelar valores simétricos. En la figura 5 se puede apreciar una gráfica de activación hiperbólica.

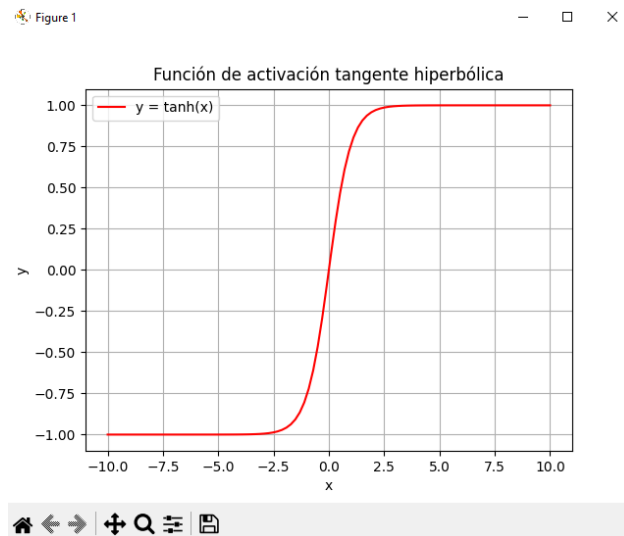


Figura 5 Gráfica función de activación hiperbólica

- **ReLU (Rectified Linear Unit):** Es una función que tiene forma de escalera y que mapea cualquier valor negativo a 0 y cualquier valor positivo a sí mismo. Se usa para problemas de regresión o para introducir no linealidad y esparsidad en la red. En la figura 6 se puede apreciar una gráfica de activación RELU.

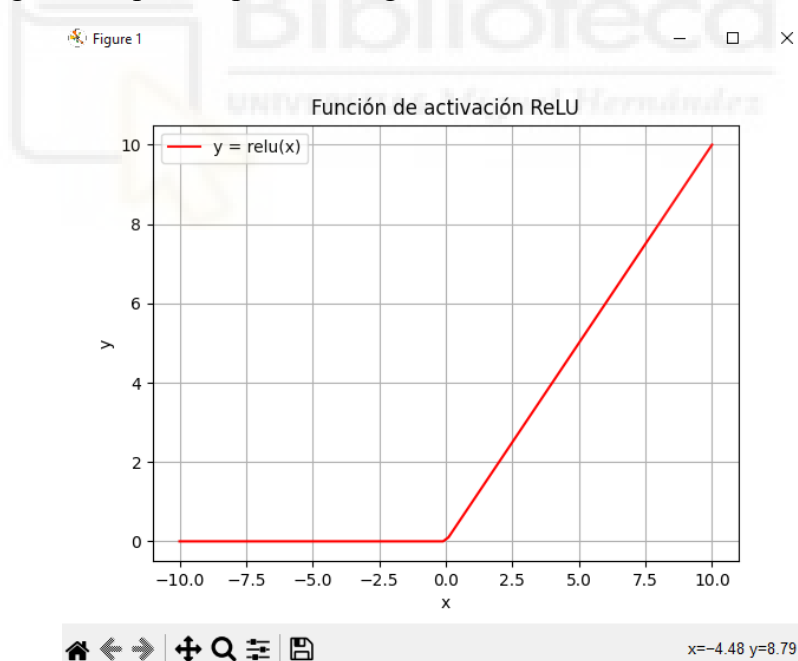


Figura 6 Gráfica función de activación RELU

- **Softmax:** Es una función que normaliza un vector de valores reales a un vector de probabilidades que suman 1. Se usa para problemas de clasificación multiclase o para modelar distribuciones de probabilidad. En la figura 7 se puede apreciar una gráfica de activación softmax-.

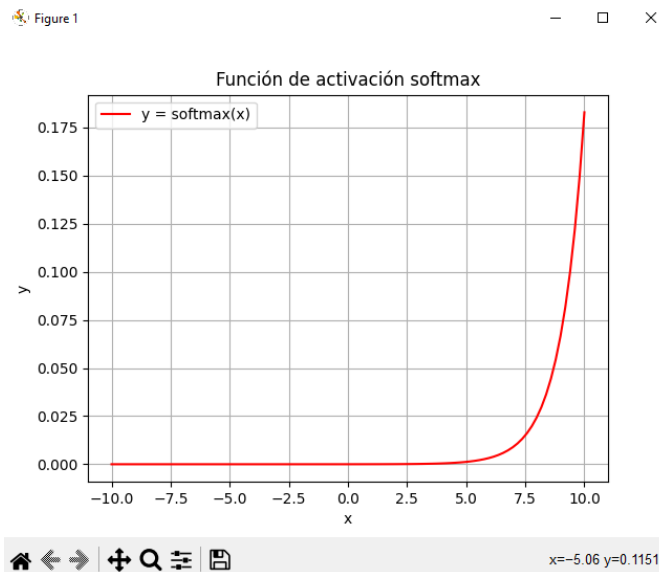


Figura 7 Gráfica función de activación softmax

2.3.3 PERCEPTRÓN MULTICAPA

Un perceptrón multicapa es una red neuronal artificial formada por múltiples capas de neuronas, para resolver problemas no linealmente separables, la principal limitación del perceptrón simple. El perceptrón multicapa puede estar total o localmente conectado, es decir, cada neurona de una capa puede recibir las entradas de todas o algunas de las neuronas de la capa anterior.

Las capas se pueden clasificar en tres tipos: capa de entrada, capas ocultas y capa de salida. La capa de entrada no realiza ningún procesamiento, solo introduce los datos en la red. Las capas ocultas aplican funciones de activación derivables (como la sigmoide o la ReLU) a las entradas y producen salidas que pasan a las capas posteriores. La capa de salida devuelve los valores finales de la red, que pueden ser uno o más, dependiendo del tipo de problema que se quiera resolver (regresión, clasificación binaria o multiclase).

El entrenamiento de un perceptrón multicapa se realiza mediante un algoritmo llamado propagación hacia atrás (o retropropagación del error), que consiste en ajustar los pesos de las conexiones entre las neuronas para minimizar una función de error que mide la diferencia entre las salidas deseadas y las obtenidas por la red. Este algoritmo se basa en

el cálculo del gradiente descendente y en la aplicación de la regla de la cadena para propagar los errores desde la capa de salida hasta la capa de entrada.

El perceptrón multicapa es un aproximador universal, es decir, puede aproximar relaciones no lineales entre los datos de entrada y salida. Sin embargo, también tiene algunas limitaciones, como la dificultad para extrapolar bien fuera del rango de los datos de entrenamiento, o la existencia de mínimos locales en la función de error que pueden detener el aprendizaje antes de alcanzar una solución óptima [42][43][44].

Un ejemplo de perceptrón multicapa lo encontramos en la siguiente imagen (figura 8), donde disponemos de 2 capas ocultas con 5 neuronas cada una, las cuales nos ayudan a clasificar los diferentes puntos dados:

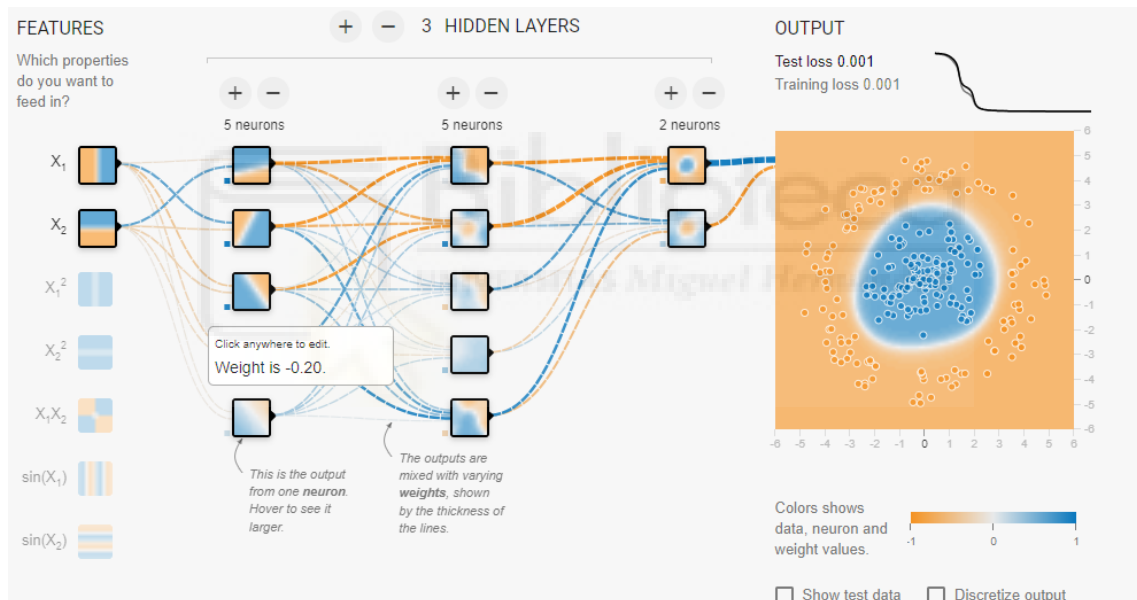


Figura 8 Ejemplo perceptrón multicapa en playground tensorflow

2.3.4 SUPPORTED VECTOR MACHINE

La Máquina de Vectores de Soporte (SVM, por sus siglas en inglés) es un método de aprendizaje automático antiguo y popular. No es muy diferente de una Red Neuronal Artificial (ANN) en su forma más simple, pero para problemas más sencillos a menudo resulta tener un mejor rendimiento.

Las SVM, o Máquinas de Soporte Vectorial (Support Vector Machines en inglés), son un conjunto de algoritmos de aprendizaje supervisado utilizados entre otros objetivos para la clasificación .

Las SVM buscan un conjunto de parámetros que separen las clases a identificar, es decir que permitan la clasificación. En un espacio multidimensional, como son las imágenes hiperespectrales, esto se conoce como hiperplano. Los parámetros que definen el hiperplano se encuentran al maximizar un margen alrededor de él, de modo que la mayor cantidad posible de muestras estén lejos del hiperplano. La forma en que una SVM hace esto se reduce a maximizar los cuadrados de las distancias .

En este tipo de algoritmo el objetivo principal es encontrar un hiperplano que separe de la mejor forma posible dos clases de datos, maximizando el margen entre ellas. Por ejemplo, si tenemos un espacio lleno de diversos grupos de puntos, nuestro objetivo sería encontrar la línea que separa ambos grupos con el objetivo de clasificar los distintos objetos. Un ejemplo sencillo en 2 dimensiones sería el siguiente: (Figura 9)

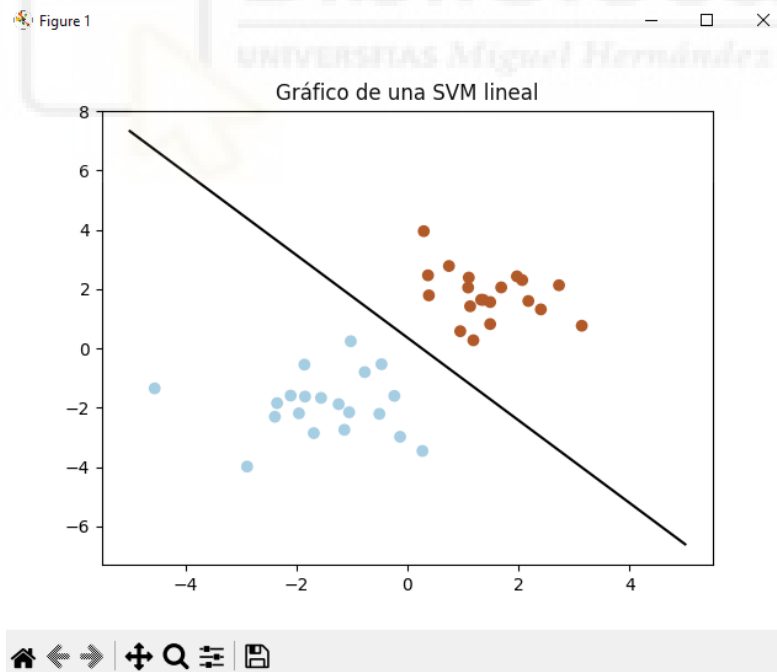


Figura 9 Gráfico ejemplificativo de una SVM lineal

2.3.5 (CNN) CONVOLUTIONAL NEURAL NETWORK

El modelo CNN (Convolutional Neural Network) o Red Neuronal Convolutiva es un tipo de red neuronal artificial que se inspira en el córtex visual del ojo humano para procesar imágenes y reconocer patrones. Es un algoritmo de aprendizaje profundo que se utiliza para problemas de visión por computador, como la clasificación, la segmentación, la detección o la generación de imágenes [45][46].

El funcionamiento del algoritmo CNN se puede resumir en los siguientes pasos [47][48]:

- Se recibe una imagen de entrada en forma de matriz de píxeles, que puede tener una o varias dimensiones según el número de canales de color (gris, RGB, etc.).
- Se aplica una operación de convolución sobre la imagen, que consiste en deslizar un filtro o kernel sobre la matriz y realizar un producto escalar entre el filtro y la región de la imagen que cubre. Esto produce una nueva matriz llamada mapa de características o mapa de activación, que resalta las zonas de la imagen donde el filtro ha detectado algún patrón.
- Se aplica una función de activación no lineal sobre el mapa de características, como la función ReLU, para introducir no linealidad en el modelo y eliminar los valores negativos o cercanos a cero.
- Se aplica una operación de agrupación o pooling sobre el mapa de características, que consiste en reducir su tamaño mediante alguna función como el máximo, el mínimo o el promedio. Esto permite disminuir la complejidad computacional, la memoria requerida y el sobreajuste del modelo, así como aumentar la invariancia a pequeñas transformaciones de la imagen.
- Se repiten los pasos anteriores varias veces, usando diferentes filtros y tamaños de agrupación, para obtener múltiples mapas de características que representan distintos niveles de abstracción y complejidad de la imagen. Las primeras capas suelen detectar bordes, formas y colores simples, mientras que las últimas capas suelen detectar objetos, rostros y escenas complejas.
- Se aplanan el último mapa de características obtenido en un vector unidimensional, que contiene las características más relevantes de la imagen. Este vector se denomina código CNN y se usa como entrada para la siguiente parte del modelo.
- Se aplica una parte de clasificación sobre el código CNN, que consiste en una o varias capas totalmente conectadas o densas, similares a las de una red neuronal multicapa (MLP). Estas capas combinan las características del código CNN mediante pesos y funciones de activación para producir una salida final, que puede

ser una etiqueta de clase (para problemas de clasificación) o un valor numérico (para problemas de regresión).

La convolución es una operación que consiste en aplicar un filtro o kernel sobre una región de la imagen, multiplicando cada valor del filtro por el valor correspondiente de la imagen y sumando los resultados. La convolución permite detectar bordes, formas, colores y otras características de la imagen. La agrupación es una operación que consiste en reducir el tamaño de la imagen, tomando el valor máximo, mínimo, medio o promedio de una región de la imagen. La agrupación permite disminuir la complejidad de la red y hacerla más invariante a las transformaciones de la imagen, como la rotación, el traslado o el escalado. La activación es una operación que consiste en aplicar una función no lineal sobre la salida de la convolución o la agrupación, para introducir no linealidad en la red y permitir que aprenda funciones más complejas.

A continuación, se muestran algunos ejemplos que ilustran el funcionamiento de una CNN

- Capa convolucional: El filtro se desliza sobre la imagen, realizando la operación de convolución en cada posición. El valor de la salida se calcula como la suma de los productos entre el filtro y la región de la imagen. En la figura 10 podemos encontrar un ejemplo de capa convolucional.

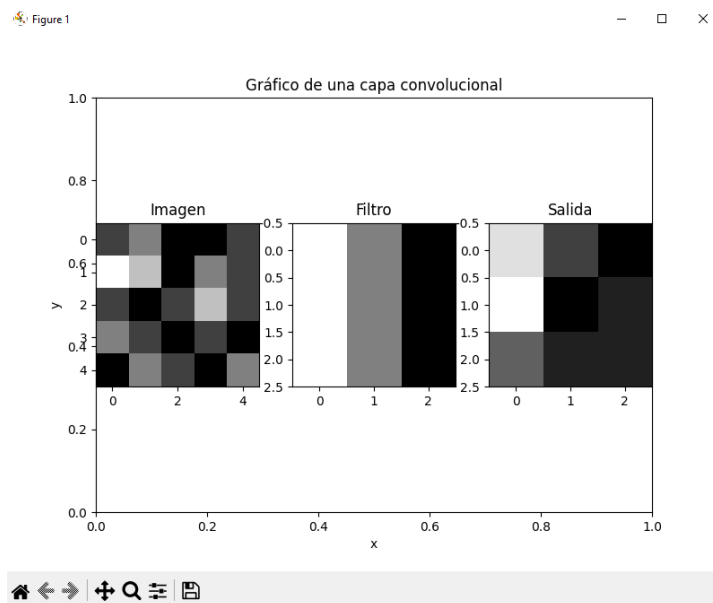


Figura 10 Gráfico de una capa convolucional

- **Capa de agrupación:** La operación de agrupación de máximo consiste en tomar el valor más alto de cada región de la imagen, reduciendo así su tamaño. En la figura 11 podemos encontrar un ejemplo de capa de agrupación.

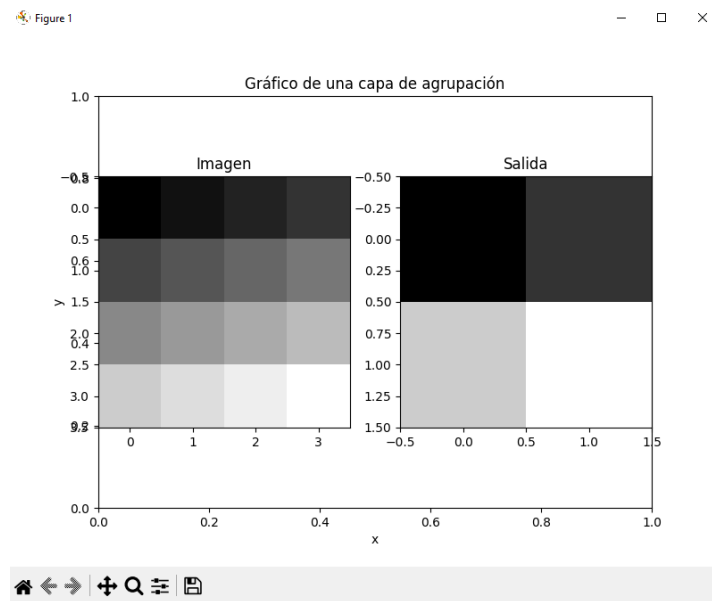


Figura 11 Gráfico de una capa de agrupación

- **Capa de activación:** La función de activación ReLU consiste en tomar el máximo entre el valor de entrada y cero, introduciendo así no linealidad en la red. En la figura 12 podemos encontrar un ejemplo de capa de activación.

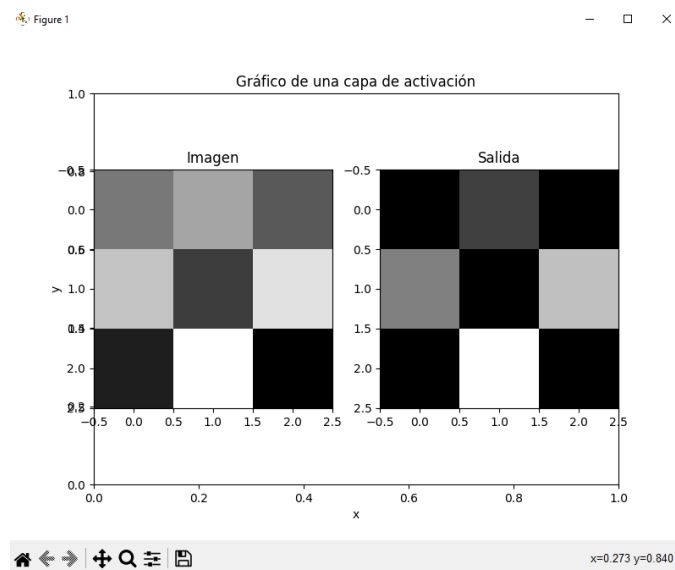


Figura 12 Gráfico de una capa de activación

2.3.6 (KNN) K-NEAREST NEIGHBOURGS

El modelo KNN (K Vecinos más Cercanos) es un algoritmo de aprendizaje supervisado que puede ser utilizado para resolver problemas de clasificación y de regresión. Se basa en la idea de que los puntos con características similares tienden a tener la misma etiqueta o valor de salida. El algoritmo no aprende explícitamente un modelo, sino que memoriza las instancias de entrenamiento y las utiliza para predecir la etiqueta o el valor de una nueva instancia basándose en la distancia a sus vecinos más cercanos. El clasificador KNN comprueba la distancia entre algunos ejemplos de prueba y los valores conocidos de algunos ejemplos de entrenamiento.

El funcionamiento del algoritmo KNN se puede resumir en los siguientes pasos [49][50]:

- Seleccionar el número de K vecinos a considerar.
- Calcular la distancia entre la nueva instancia y las instancias de entrenamiento, usando alguna medida como la distancia euclidiana, manhattan o minkowski.
- Tomar las K instancias más cercanas según la distancia calculada.
- Entre las K instancias, contar el número de puntos en cada categoría (para problemas de clasificación) o calcular el promedio de los valores (para problemas de regresión).
- Atribuir a la nueva instancia la categoría más frecuente o el valor promedio entre las K instancias.

El algoritmo KNN tiene algunas ventajas, como su simplicidad, su flexibilidad y su robustez ante datos no lineales o ruidosos. Sin embargo, también tiene algunas desventajas, como su alto coste computacional y de memoria, su sensibilidad a la elección del valor de K y de la medida de distancia, y su dificultad para manejar datos con muchas dimensiones o con atributos categóricos [51][52].

Para aplicar el algoritmo KNN a redes neuronales, se puede utilizar el concepto de vecindad para definir una función de activación que dependa de los valores de las unidades cercanas. Esto puede mejorar el rendimiento y la capacidad de generalización de las redes neuronales, especialmente para problemas complejos como el reconocimiento de patrones o la minería de datos.

2.3.7 (PLS-DA) PARTIAL LEAST SQUARES DISCRIMINANT ANALYSIS

El objetivo del proceso análisis discriminante con mínimos cuadrados parciales (PLS-DA) es explicar y predecir la pertenencia de observaciones a varias clases usando variables explicativas cuantitativas o cualitativas. Este método es una adaptación de la regresión mínimos cuadrados parciales (PLS) que tiene ventajas sobre el análisis discriminante clásico, como mayor robustez, flexibilidad y capacidad para manejar datos con multicolinealidad, valores faltantes o alta dimensionalidad.

El proceso PLS-DA utiliza el algoritmo PLS aplicado sobre la tabla disyuntiva completa obtenida de la variable dependiente cualitativo. Se obtienen tantos modelos como categorías de la variable dependiente y se asocia una observación a la categoría que tenga un valor más alto en la ecuación correspondiente. Puede manejar tanto variables explicativas cuantitativas como cualitativas. En esencia, busca una combinación lineal de las variables originales que optimice la separación entre clases.

El método PLS-DA presenta varias fortalezas notables. En primer lugar, es un excelente selector de características, lo que lo convierte en una herramienta valiosa para identificar las variables más relevantes en un conjunto de datos. Además, resulta especialmente útil para la clasificación en problemas con etiquetas de clase, facilitando la diferenciación precisa entre distintas categorías basadas en las características observadas.

No obstante, PLS-DA también tiene ciertas debilidades que deben ser consideradas. Es propenso al sobreajuste, por lo que es fundamental emplear técnicas de validación cruzada para asegurar la generalización de los resultados. Asimismo, este método requiere un número suficiente de muestras para funcionar correctamente, lo cual puede ser una limitación en estudios con datos escasos. A pesar de estas limitaciones, PLS-DA encuentra aplicaciones en campos como la metabolómica y el análisis omics, siendo especialmente útil en contextos con grandes volúmenes de datos, muchas características y pocas muestras.

2.3.8 CLASIFICACIÓN BAYESIANA

La Clasificación Bayesiana es un método de aprendizaje supervisado que se basa en el teorema de Bayes para estimar la probabilidad de que una imagen o una región de una imagen pertenezca a una determinada clase o categoría. El teorema de Bayes dice que la probabilidad posterior de una clase, dada la imagen, depende de la probabilidad previa de la clase, la probabilidad de la imagen dada la clase y la probabilidad marginal de la imagen. La clase con la mayor probabilidad posterior se asigna a la imagen o la región.

El proceso bayesiano se puede resumir en los siguientes pasos:

- Definir las hipótesis que se quieren contrastar y asignarles una probabilidad a priori, que refleje el grado de creencia que se tiene sobre ellas antes de observar los datos.
- Definir el modelo estadístico que relaciona los datos con las hipótesis, y calcular la probabilidad de los datos dada cada hipótesis, que se llama verosimilitud.
- Aplicar el teorema de Bayes para obtener la probabilidad a posteriori de cada hipótesis, que refleje el grado de creencia que se tiene sobre ellas después de observar los datos.
- Comparar las probabilidades a posteriori de las hipótesis y elegir la que tenga el valor más alto, o calcular el factor de Bayes, que mide la evidencia a favor o en contra de una hipótesis respecto a otra.

Gráfica de las probabilidades a priori: (Figura 13)

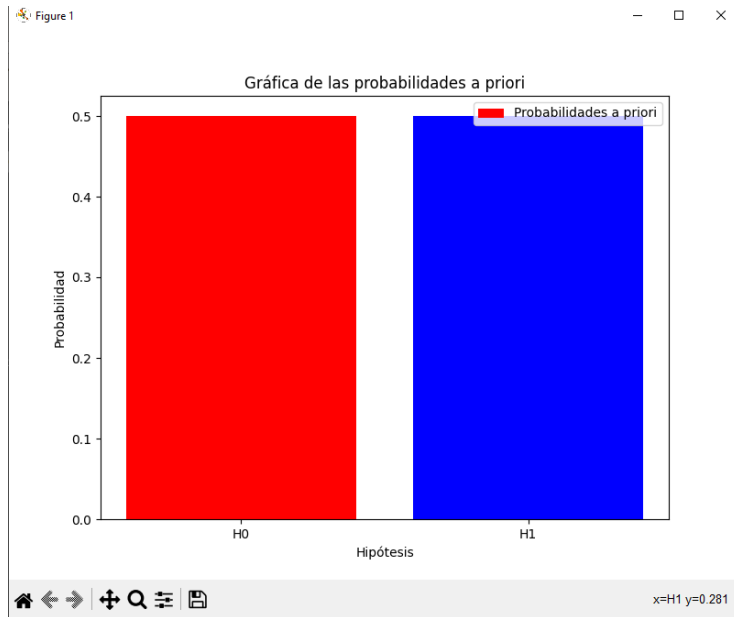


Figura 13 Gráfica de probabilidades a priori

- Gráfica de la verosimilitud: (Figura 14)

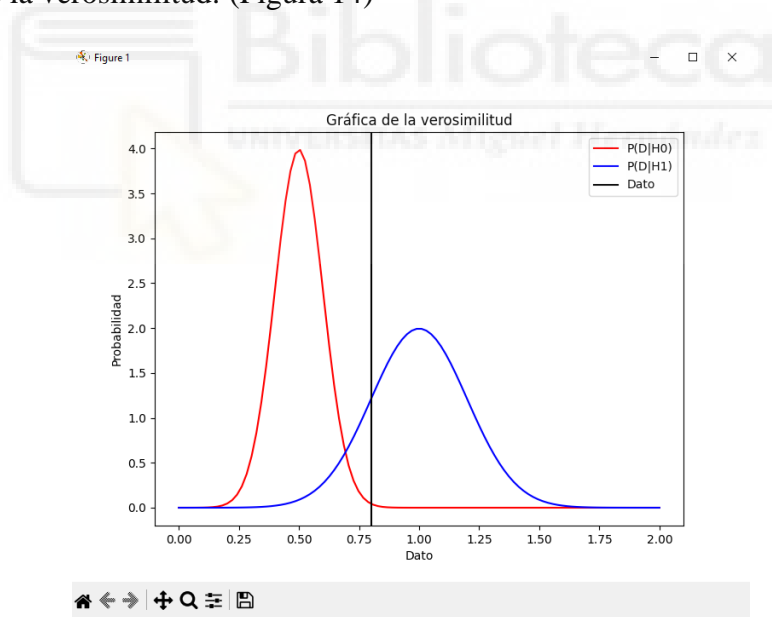


Figura 14 Gráfica de la verosimilitud

- Gráfica de las probabilidades a posteriori: (Figura 15)

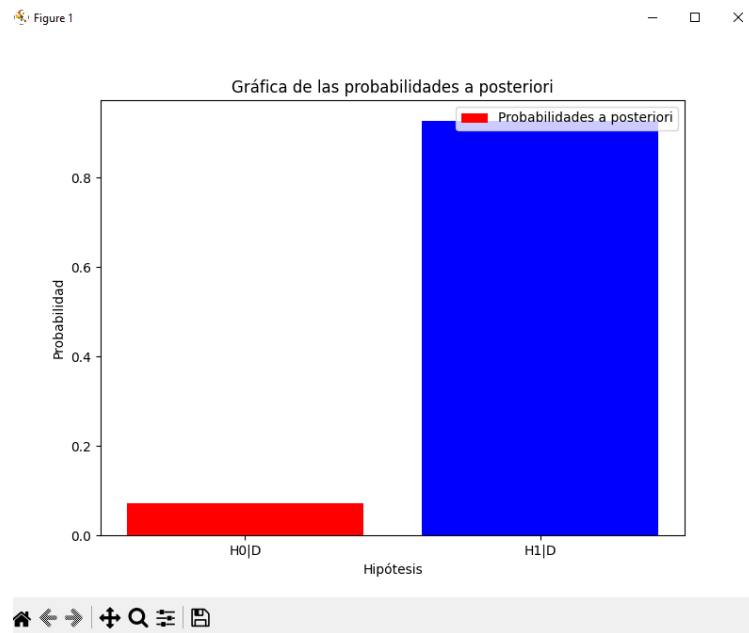


Figura 15 Gráfica de las probabilidades a posteriori

- Gráfica del factor de Bayes: (Figura 16)

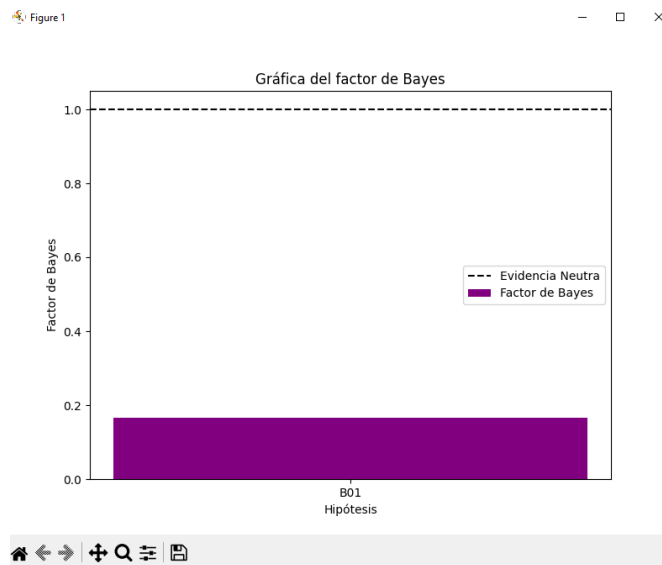


Figura 16 Gráfica del factor de Bayes

2.3.9 ARBOLES DE DECISION

Un árbol de decisión está compuesto por dos tipos de nodos: los nodos de decisión y los nodos resultado. Los nodos de decisión, que son los nodos intermedios, guían la evaluación y tienen una lista de nodos hijo y una propiedad asociada para tomar decisiones. Los nodos resultados, que son las hojas del árbol, contienen el resultado final de la clasificación o regresión.

La evaluación de un árbol implica comparar las propiedades de un objeto con los criterios de los nodos a lo largo del árbol, comenzando desde el nodo raíz y avanzando hacia los nodos hijo según el resultado de las comparaciones. Este proceso continúa hasta que se alcanza un nodo resultado, que determina la clase a la que pertenece el objeto evaluado. En la figura 17 podemos ver un ejemplo de árbol de decisión

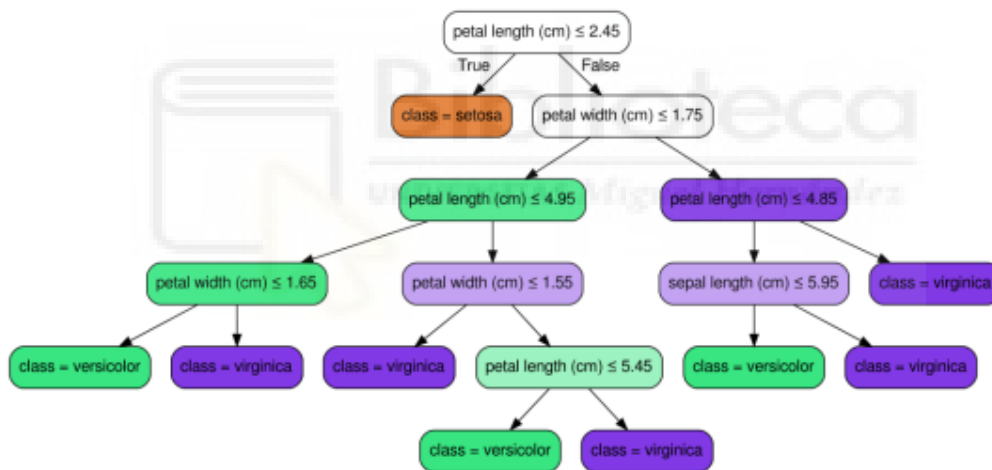


Figura 17 Ejemplo de árbol de decisión

2.3.10 RANDOM FOREST

El método de Random Forest es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio muestreado independientemente y con la misma distribución para todos los árboles en el bosque. El error de generalización para los bosques converge casi seguramente a un límite cuando el número de árboles en el bosque se hace grande. El error de generalización de un bosque de clasificadores de

árboles depende de la fuerza de los árboles individuales en el bosque y la correlación entre ellos. Usando una selección aleatoria de características para dividir cada nodo se obtienen tasas de error que se comparan favorablemente con Adaboost [53], pero son más robustos con respecto al ruido¹.

El método de Random Forest se puede aplicar tanto a problemas de clasificación como de regresión, y se puede usar para estimar cantidades de interés heterogéneas, como efectos causales o cuantiles condicionales². Además, se pueden obtener medidas de importancia de las variables basadas en el decrecimiento promedio de la impureza o en la reducción promedio del error

2.3.11 (PCA) PRINCIPAL COMPONENT ANALYSIS

El Análisis de Componentes Principales (PCA) tiene como objetivo simplificar la complejidad de espacios muestrales con muchas dimensiones mientras conserva su información. A continuación, se destacan los aspectos clave:

1. Reducción de Dimensionalidad:

- PCA reduce la cantidad de variables originales en un conjunto de datos.
- Encuentra un número menor de componentes principales que explican aproximadamente la misma variabilidad que las variables originales.
- Esto facilita la visualización y el análisis de datos en espacios de menor dimensión.

2. Identificación de Variables Relevantes:

- PCA permite identificar las variables más importantes que contribuyen a la variabilidad observada en los datos.
- Ayuda a descartar las variables menos relevantes o redundantes.

3. Exploración de Datos:

- PCA se utiliza para visualizar la estructura subyacente de un conjunto de datos multidimensional.

- Representa gráficamente cómo las observaciones se distribuyen en los componentes principales.
4. **Preservación de Variabilidad:**
- Aunque se reduzca la dimensionalidad, PCA intenta conservar la mayor cantidad posible de variabilidad presente en los datos originales.
 - Los componentes principales capturan la información más relevante.

2.3.12 HIERARCHICAL MODEL

El modelo jerárquico tiene como objetivo organizar y estructurar datos en niveles sucesivos, donde cada nivel contiene subconjuntos de unidades de análisis. Aquí están los aspectos clave:

1. **Reducción de la Complejidad:**
 - El modelo jerárquico simplifica la representación de datos al agruparlos en niveles jerárquicos.
 - Cada nivel contiene subgrupos de unidades relacionadas.
 - Facilita la visualización y el análisis de datos al dividirlos en componentes más manejables.
2. **Organización Estructurada:**
 - Permite **organizar un modelo** de manera estructurada.
 - Los submodelos se agrupan bajo actividades o categorías principales.
 - Es útil para representar sistemas complejos con múltiples componentes interrelacionados.
3. **Reutilización de Submodelos:**
 - Los submodelos pueden ser reutilizados en diferentes partes del mismo modelo o en otros modelos.
 - Esto ahorra tiempo y esfuerzo al no tener que recrear los mismos componentes una y otra vez.
4. **Colaboración y Documentación:**
 - Varios modeladores pueden trabajar simultáneamente en diferentes submodelos sin interferir entre sí.

- La estructura modular permite pruebas parciales y facilita la documentación.

En resumen, el modelo jerárquico es una herramienta para organizar, simplificar y representar datos complejos en niveles estructurados. Puede aplicarse en diversos campos, desde simulaciones hasta análisis estadísticos [54].

2.3.13 HRNet+OCR (HIGH RESOLUTION NET + OBJECT CONTEXTUAL REPRESENTATION)

La estructura HRNet+OCR es una red neuronal basada en una lista de clasificación en el sitio web 'paperswithcode.com', que se trata del rendimiento de la segmentación semántica en el conjunto de datos de prueba 'Cityscapes', basado en el valor de IOU. La estructura HRNet+OCR tiene el valor más alto de IOU, que es del 84.5%. HRNet+OCR es una columna vertebral derivada basada en HRNet.

El nombre completo de HRNet es High-Resolution Network (Red de alta resolución). HRNet tiene una característica que agregará las representaciones de salida en cuatro resoluciones diferentes, y luego utilizará convoluciones 1X1 para fusionar estas representaciones. La representación de salida se alimentará al clasificador.

El nombre completo de OCR es Object Contextual Representations (Representaciones contextuales de objetos). Es un modelo de función que se puede integrar en HRNet o ResNet. El propósito es que la etiqueta de clase asignada a un píxel es la categoría del objeto al que pertenece el píxel. En la red neuronal normal, la etiqueta de clase se asignará al píxel si este ha cumplido con la característica que la red ha aprendido sobre esta clase. Este hecho puede causar que algunos píxeles que no pertenecen a un objeto, como un punto en el fondo, se etiqueten como un objeto.

En el código original de HRNet, los desarrolladores solo proporcionan la opción de SGD para el entrenamiento y prueba en el conjunto de datos Cityscapes utilizado en su proyecto. En el artículo [8] se menciona que para el conjunto de datos Cityscapes, el optimizador SGD mejora el rendimiento de la red en un 0.6% en calidad panóptica en comparación con el optimizador Adam. También es necesario mencionar que, en investigaciones y proyectos anteriores, el optimizador Adam proporciona un mejor

rendimiento en datos hiperespectrales para la columna vertebral de la red Deep Res U-Net y U Net++.

Existen otras estructuras de redes neuronales y artículos relacionados, y se pueden encontrar códigos a través de 'paperwithcode.com', como DCNAS, Panoptic-DeepLab. También hay algunas redes neuronales especialmente diseñadas para aprender datos hiperespectrales, pero utilizan la biblioteca TensorFlow en lugar de PyTorch, como HypbridSN y Recurrent 3D-CNN [55] [56] [57].

2.3.14 (RBNN) RADIAL BASIS NEURAL NETWORK

Una red neuronal de funciones de base radial (RBNN) es una red neuronal artificial que utiliza funciones de base radial como funciones de activación. El objetivo principal de esta red es aproximar funciones complejas y resolver problemas de clasificación y regresión. A continuación, se detalla su funcionamiento:

1. Funciones de Base Radial:

- Las RBF utilizan funciones de base radial como su función de activación.
- Estas funciones son radiales en torno a un centro y dependen solo de la distancia desde ese centro.
- Ejemplos comunes de funciones de base radial son las funciones Gaussianas.

2. Arquitectura de la Red:

- Las RBF generalmente tienen tres capas:
 - **Capa de entrada:** Recibe el vector de entrada.
 - **Capa oculta:** Utiliza funciones de base radial como función de activación.
 - **Capa de salida:** Proporciona la combinación lineal de las salidas de las funciones de base radial.
- La arquitectura modular permite una representación eficiente y una rápida convergencia durante el entrenamiento.

3. Universalidad Aproximada:

- Bajo ciertas condiciones, las RBF son aproximadores **universales** en un conjunto compacto de datos.
- Esto significa que, con suficientes neuronas ocultas, una RBF puede aproximar cualquier función continua en un conjunto cerrado y acotado con precisión arbitraria.

4. **Ventajas:**

- **Velocidad de aprendizaje rápida:** Las RBF pueden converger rápidamente durante el entrenamiento.
- **Generalización efectiva:** Pueden aprender y generalizar patrones de datos de manera eficiente.

En resumen, las redes RBF son herramientas poderosas para aproximar funciones no lineales y resolver problemas complejos. Su estructura modular y su capacidad de aproximación universal las hacen valiosas en diversas aplicaciones como la clasificación de imágenes hiperespectrales [58][59] [60].

2.4 MATERIALES

En cuanto a los materiales analizados en este trabajo, se ha dispuesto de datasets de datos correspondientes a plásticos ABS, PET, PP_PE, y PS-Yogures, como se muestra en la tabla 1.

El HSI con diferentes tipos de plástico se utiliza para demostrar la funcionalidad básica del enfoque y recibe más atención debido a la existencia de información de referencia. Como parte de una aplicación de clasificación de residuos, la demostración también incluye experimentos con residuos reales que consisten en plásticos y residuos. Por otro lado, el conjunto de datos basado en residuos se utiliza para demostrar el potencial y la flexibilidad del enfoque en diferentes tipos de aplicaciones.

Tabla 1 Materiales estudiados


Plástico	Descripción
ABS	El acrilonitrilo butadieno estireno (ABS) es un plástico resistente y rígido ampliamente utilizado en la fabricación de piezas de automóviles, electrodomésticos y juguetes. Ofrece una buena resistencia al impacto y propiedades mecánicas versátiles.
PET	El tereftalato de polietileno (PET) es un plástico transparente y resistente utilizado comúnmente en envases de alimentos y bebidas. Es reciclable y presenta una excelente barrera contra la humedad y el oxígeno, lo que lo hace adecuado para el envasado de productos sensibles.
PP_PE	El polipropileno (PP) y el polietileno de baja densidad (PE) son plásticos versátiles y resistentes al impacto. El PP se utiliza en una amplia variedad de aplicaciones, desde envases hasta componentes automotrices. El PE es comúnmente utilizado en bolsas de plástico y envases flexibles.
PS- Yogures	El poliestireno (PS) es un plástico transparente y rígido utilizado en la fabricación de vasos y envases desechables, como los utilizados para yogures y postres. Es ligero y proporciona una buena protección contra la humedad y la contaminación externa.

CAPÍTULO 3

SISTEMA DE

ADQUISICIÓN DE

DATOS



En este capítulo se explica cómo se han obtenido los datos, explicando cual es el sistema de adquisición de datos y la forma de estos. También se realiza una explicación de cómo son los hipercubos que se han utilizado.

3.1 INFRAESTRUCTURA HARDWARE

El sistema completo consta de cuatro partes básicas:

- Sistema de transporte del material a clasificar.
- Sistema de adquisición de HSI.
- Sistema de recuperación y clasificación del material.
- Sistema informático, incluidas las redes de comunicación.

El aspecto más importante para este proyecto es la cámara y el sistema de iluminación. La cámara utilizada es una cámara hiperespectral, que captura imágenes en múltiples bandas del espectro electromagnético. Estas imágenes proporcionan información detallada sobre la composición química de los materiales.

Especificaciones de la cámara:

- Resolución espectral: 5 nm
- Rango espectral: 400-1000 nm
- Resolución espacial: 1024x1024 píxeles
- Velocidad de captura: 60 cuadros por segundo

El sistema de iluminación está compuesto por focos halógenos que cubren todo el espectro a utilizar. Estos focos proporcionan una iluminación uniforme y constante, crucial para obtener datos espectrales precisos y reproducibles.

3.2 ESTRUCTURA DE DATOS OBTENIDOS

Como ya se ha explicado anteriormente, las imágenes hiperespectrales difieren de las imágenes comunes al utilizar una mayor resolución en la dimensión espectral y un rango de longitudes de onda más amplio. La mayor resolución espectral y las longitudes de onda invisibles para el ojo humano permiten que una computadora extraiga mucha más información del objeto capturado. Para capturar imágenes hiperespectrales, se utiliza un espectrógrafo hiperespectral, que es un tipo de instrumento de teledetección que cuantifica la radiación electromagnética de una escena. La radiación de la escena proviene de la energía solar, energía térmica o una combinación de ambas. Los diferentes tipos de espectrógrafos hiperespectrales pueden detectar y cuantificar diferentes longitudes de

onda de la radiación electromagnética. Las longitudes de onda que el espectrógrafo hiperespectral puede detectar afectan el tipo de información que se puede obtener de la escena. Las propiedades físicas del objeto capturado afectan cómo refleja, absorbe y emite las diferentes longitudes de onda de la radiación, lo que también afecta la información recibida. Un ejemplo del mismo se puede visualizar en la figura 18.

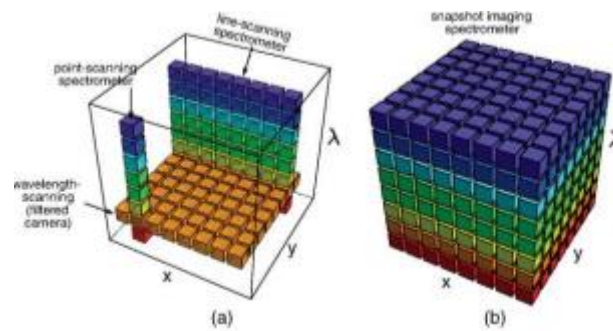


Figura 18 Representación de una imagen hiperespectral

En nuestro caso, se recibe un fichero .bin con toda la información de la captura de imágenes. Este fichero contiene toda la información de la imagen hiperespectral, con 224 bandas y 640 píxeles. Antes de ser procesadas, las imágenes presentan una serie de ruidos que pueden ser eliminados. Debido a que las primeras 10 bandas y las últimas 10 no aportan información relevante, se ha decidido recortar estos espectros, eliminando datos que harían el procesamiento más pesado y lento. Además, en nuestro fichero .bin también tenemos píxeles en los que se está leyendo parte de la cinta transportadora, estática que no contiene ningún tipo de plástico que vayamos a clasificar. Por lo tanto, recortamos espacialmente la imagen para quitarle 90 píxeles por cada lado. De esta forma, obtenemos un hipercubo más reducido, con los datos esenciales para el proceso de clasificación.

Tras explicar el recorte espacial y espectral, se pueden añadir gráficos como el siguiente: (Figura 19)

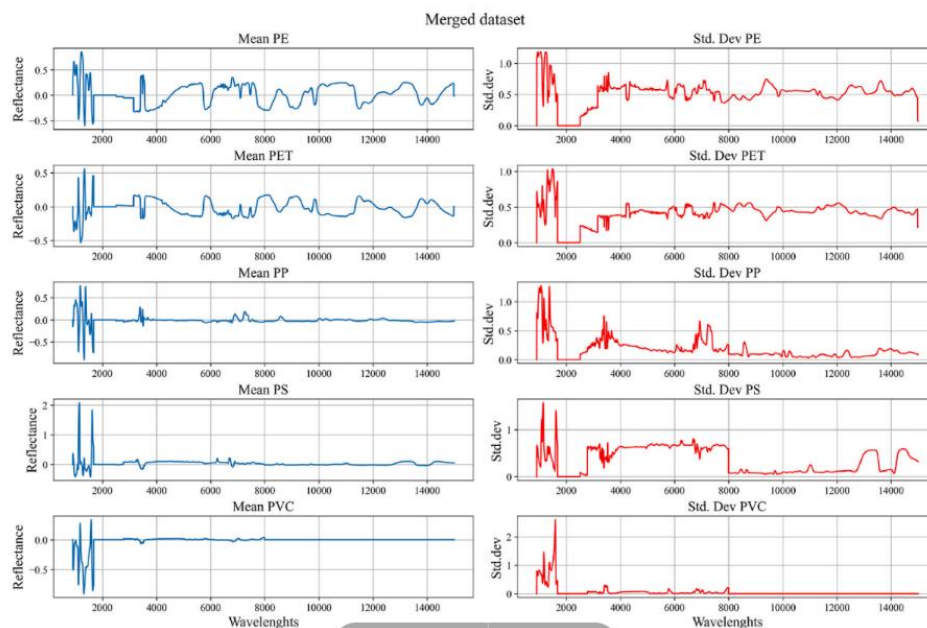


Figura 19 Ejemplo de representación de las bandas en diversos materiales

Una vez nos hemos deshecho de los datos redundantes, hemos de preparar nuestros propios datos para poder entrenar a nuestros distintos clasificadores. Aquí nos encontramos con otro problema, ya que hay espectros de los datos en los que es difícil distinguir entre la cinta y lo que es un plástico a clasificar. Para solucionar este problema lo que hacemos es recorrer todos los hipercubos de cada tipo de datos, los cuales están recogidos en un mismo directorio, y se calcula el máximo y el mínimo de la mediana de todas sus bandas. Con ello conseguimos una idea de cómo están compuestos los datos de nuestros hipercubos, ya que los valores más bajos corresponderán siempre a la cinta y los altos a los objetos que queremos clasificar; además, podemos asignar un grado de permisividad para posteriormente procesar con exactitud los datos que sobrepasen ese umbral, actuando de esa forma sobre la cantidad más eficiente de los datos posibles. Estos valores se guardarán en un fichero al que se asignarán todos los datos de todos los materiales a los que nos referimos.

Tras ellos vamos a segmentar los datos, para, con ello separar manualmente los datos que se deben clasificar, debido a que no podemos visualizar todas las bandas a la vez, para este proceso de segmentación se hará la media de todas las bandas halladas en en

hipercubo y mediante OPENCV seremos capaces de visualizar una imagen en blanco y negro de los cubos.

Una vez visualizadas nuestras imágenes procedemos a segmentarlas manualmente, para ello es posible seleccionar cada punto de los que se compone una imagen. Como es un proceso demasiado costoso manualmente, se desarrolló un sistema en el que, con el modelo SAM, de Meta se segmentan los cubos seleccionando uno o varios puntos de cada imagen seleccionada.

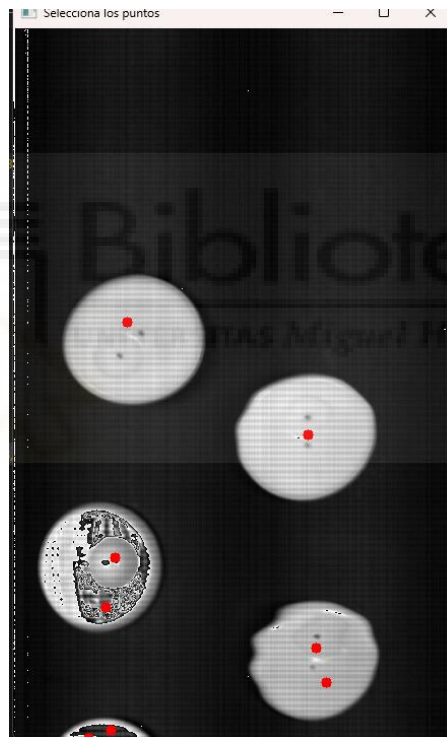


Figura 20 Puntos seleccionados para hacer la segmentación

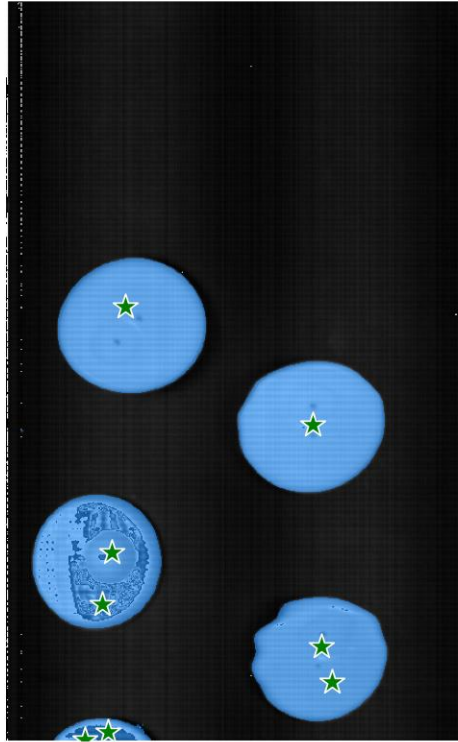
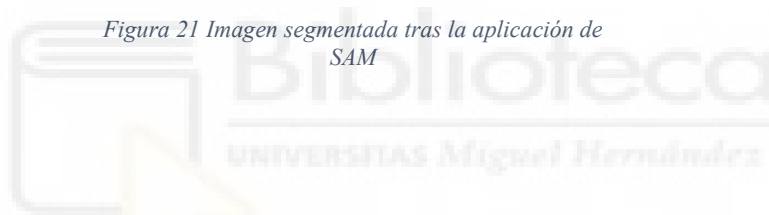


Figura 21 Imagen segmentada tras la aplicación de SAM



Como se muestra en la figura 20, los puntos rojos son los que se deben de seleccionar por nosotros manualmente, y tras la actuación del modelo SAM se segmenta la imagen como se puede ver en la figura 21.

Tras ello obtenemos una máscara donde se señala en que coordenadas de la imagen hiperspectral se encuentran los datos a clasificar.

En resumen, para el proceso de etiquetado y preparación de datos para la red neuronal, primero descargamos los datos desde un servidor en la nube. Luego, preprocesamos los datos extrayendo valores importantes como la media, el valor máximo y el valor mínimo de cada una de las imágenes. Esta información nos permite descartar gran parte del ruido en la imagen, optimizando así el tiempo de procesado de los datos.

Una vez hemos descartado los datos redundantes, y como anteriormente se ha mencionado realizando también un crop espectral y espacial de la imagen podemos

empezar a anotar los datos, para ello se han desarrollado 2 tipos de anotación, el primero el explicado anteriormente y el segundo el cual seleccionas la sección rectangular en la cual se encuentra el objeto deseado. Por otro lado, y como se muestra en la figura 22 tras probar diferentes formas de etiquetado de imágenes también se puede usar una herramienta comercial.

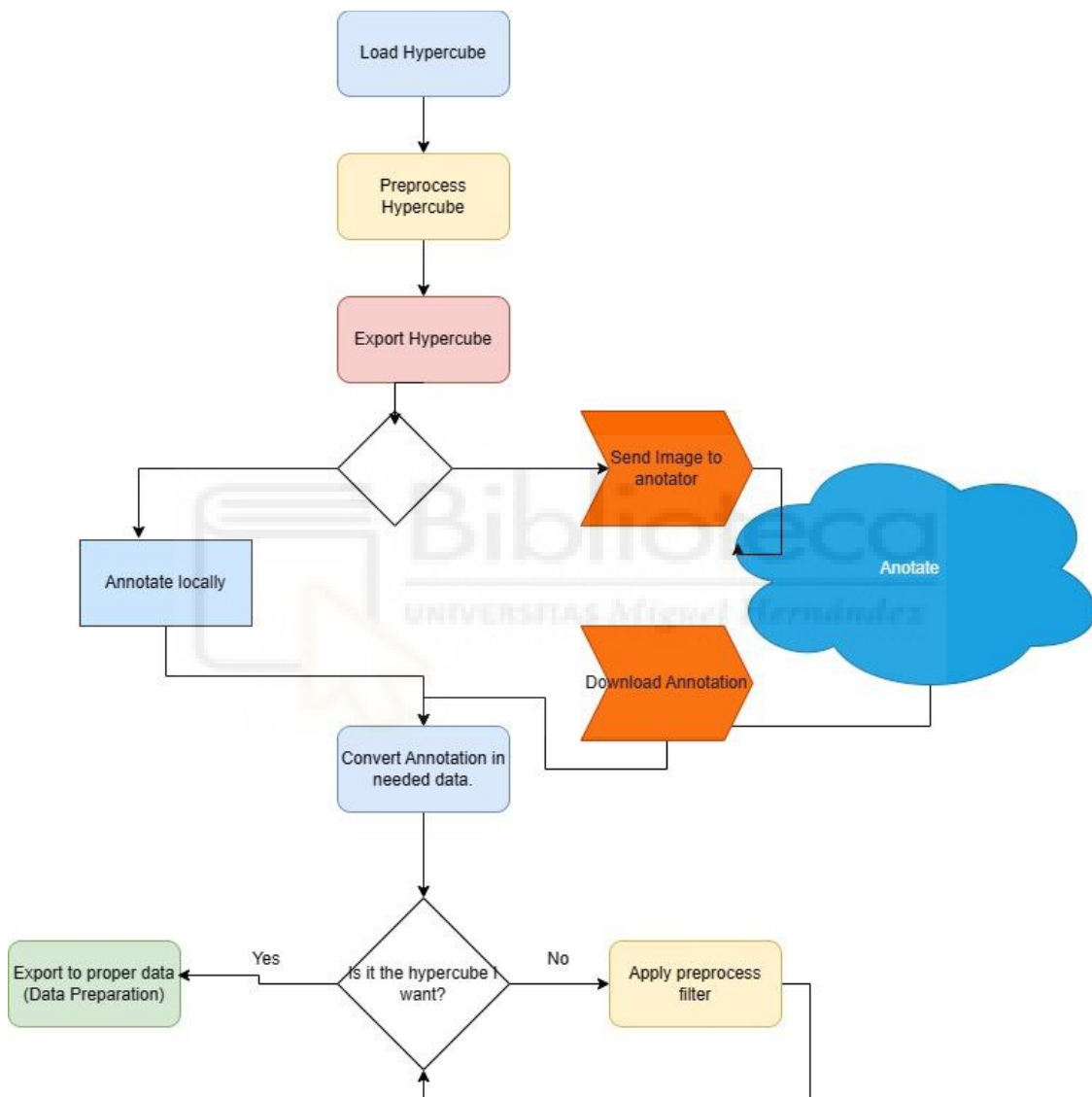


Figura 22 Sistema de manejo de HSI

La idea es utilizar aplicaciones externas disponibles en el mercado, aprovechando el software como servicio (SaaS) que ya poseemos para etiquetar imágenes, debido a que, para una comprensión más clara del problema, lo que realmente se etiquetan son las

imágenes resultantes del preprocesado, consiguiendo así una percepción. Lo que vamos a hacer aquí es:

- Cargar los archivos
 - Preprocesar las imágenes
 - Recorte espacial
 - Promedio
 - Algunos preprocesos para mejorar la visualización de la imagen para el ser humano
- Exportar el cubo a un formato de imagen.

Una vez que hemos preprocesado la imagen para ser procesada por un humano, lo que tenemos que hacer es enviar la imagen a nuestra máquina de etiquetado. Tenemos una variedad de software de etiquetado:

- AWS
- Microsoft Azure
- Roboflow
- CVAT
- Label Box
- Scale AI
- Make Sense
- VGG Image Annotator
- Labelme
- Dashdoodler
- Label Studio

Una vez que tenemos nuestras imágenes etiquetadas, dependiendo del formato en el que estén etiquetadas, tenemos que crear un convertidor para que coincida esta información con nuestros datos, según lo que realmente queremos.

Tras haber acabado el proceso de etiquetado es hora de entrenar el modelo. Para ello y siguiendo las pautas marcadas por tensorflow (figura 23) procederemos al proceso de hiperparametrización.

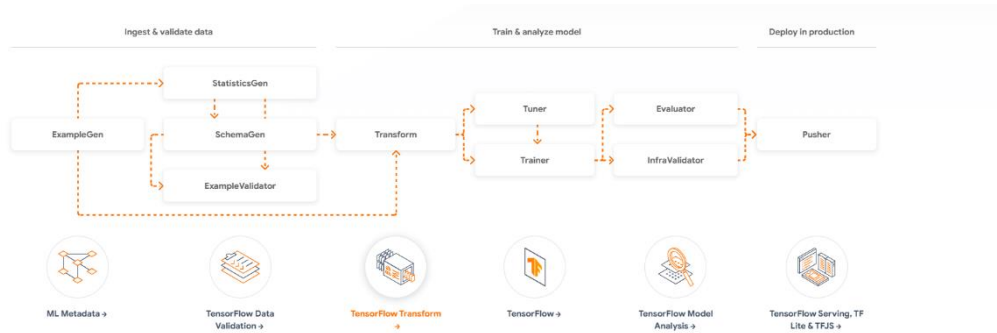


Figura 23 ML Pipeline por TensorFlow

Es importante remarcar que, para ayudar con dicho proceso, como se muestra en la figura 24, se ha utilizado la herramienta de Tensorboard, la cual nos permite tener un seguimiento más exhaustivo y detallado del proceso de hiperparametrización con una visualización de datos detallada.

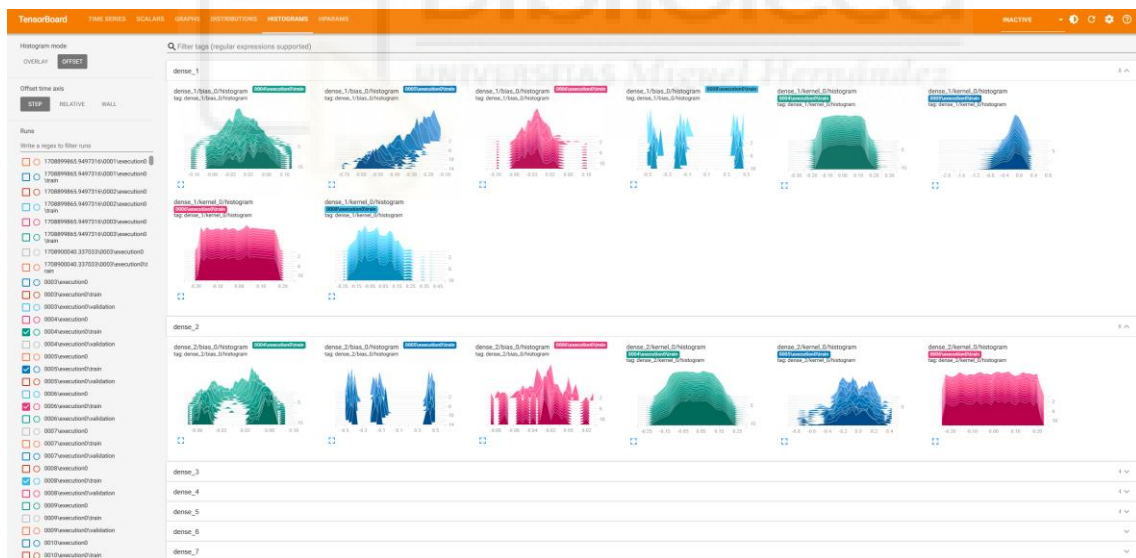


Figura 24 Visualización del entrenamiento con tensorboard

Una vez determinamos los hiperparámetros ideales, en línea con las directrices recomendadas por TensorFlow en su plataforma, procedemos a reentrenar la red neuronal utilizando esta configuración. Además, para facilitar una implementación robusta y escalable del modelo, hemos integrado TensorFlow Serving mediante las herramientas

TFX. Esto nos permite encapsular el modelo y gestionar su utilización de manera eficiente.



CAPÍTULO 4

TÉCNICAS DE

CLASIFICACIÓN

En este capítulo se analiza la técnica de clasificación utilizada, concretamente el perceptrón multicapa.



4.1 INTRODUCCIÓN

Alentados por los resultados de los artículos analizados, hemos decidido probar en nuestro propio sistema el resultado de aplicar un modelo, el perceptrón multicapa, este ofrece una clasificación píxel a píxel muy precisa, con la cual esperamos una posible mejora significativa en la precisión del sistema.

4.2 PERCEPTRÓN MULTICAPA

El clasificador MLP es un conjunto completamente conectado de nodos no lineales inspirados en el funcionamiento de las neuronas en el cerebro. Las neuronas reciben señales, las procesan y producen una salida específica.

Las neuronas aprenden a responder a diferentes tareas considerando diferentes ejemplos del conjunto de entrenamiento. El MLP más simple tiene una capa de entrada, una capa oculta y una capa de salida. En este contexto, el vector de características de entrada incluye valores como la longitud de onda, mientras que la matriz de pesos y el vector de sesgo influyen en los cálculos del vector oculto. La variable estimada se obtiene a través de una función no lineal, como la tangente hiperbólica, que transforma los valores del vector oculto en la salida final.

Un clasificador MLP con dos o más capas ocultas y neuronas completamente interconectadas funciona de manera similar. En este caso, se utilizan múltiples capas con diferentes vectores de activaciones para cada capa. La capa de entrada se puede representar como el vector inicial de características. En la figura 25 tenemos un ejemplo de cómo funcionaría un perceptrón multicapa.

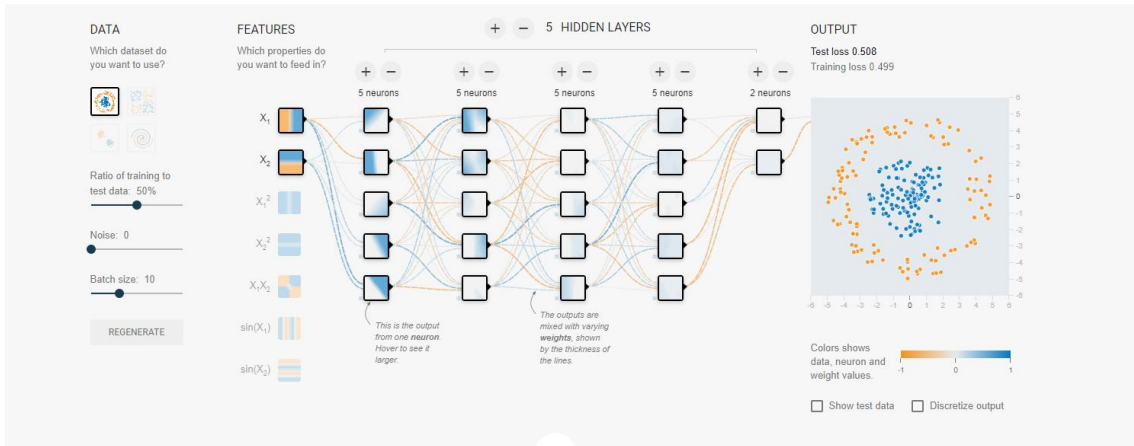


Figura 25 Representación del perceptrón multicapa en play TensorFlow



CAPÍTULO 5

ANÁLISIS DE

TÉCNICAS DE

CLASIFICACIÓN

En este apartado se realiza una explicación concreta sobre el proceso que se ha seguido para el desarrollo del modelo. También se explica cómo se han utilizado diversas herramientas como Tensorboard o Keras Tunner.

5.1 PERCEPTRÓN MULTICAPA

5.1.1 FUENTE DE DATOS

Tras el primer proceso de preprocesamiento de datos y obteniendo una máscara como se comentaba en el capítulo 3 de este documento, para procesar los datos del perceptrón multicapa se deben obtener cada píxel, con los 204 posibles valores etiquetados en un CSV aparte.

Una vez se han obtenido todos los pixeles requeridos etiquetados se almacenarán en distintos CSV dependiendo del tipo de plástico que exista, para este paso se ha etiquetado también la cinta obteniendo un total de 5 clases a etiquetar:

- 1-> ABS
- 2-> PET
- 3-> PP
- 4-> PE
- 5-> CINTA

Tras este proceso de etiquetado, y con 1 fichero CSV para cada uno de los valores se deben de mezclar los datos para entrenar a nuestro modelo con algo lo más similar a la realidad posible. Esto presenta otro problema ya que por el gran volumen de datos no se puede cargar todos los datos en memoria a la vez.

Para solucionar este problema, se ha desarrollado un script que permite mezclar los datos originales y crear diversos ficheros más reducidos, con los cuales es más sencillo trabajar. En este caso, se han creado 100 ficheros, de los cuales el 20% se ha dedicado a la prueba (test) y el 80% restante se ha dedicado al entrenamiento de nuestra red. Una vez obtenidos los hiperparámetros necesarios, se ha usado la totalidad de los datos iniciales para el entrenamiento de la red, obteniendo buenos resultados.

5.1.2 CONFIGURACIÓN DE RED

Para este propósito usaremos varios tipos de capas, respetando siempre la capa de salida en la cual debemos de usar la función softmax, como ya se ha mencionado anteriormente. Para obtener la configuración más precisa de la red se ha desarrollado un optimizador sobre el cual se realizarán las pruebas para determinar que configuración da el mejor resultado.

Como capa de entrada o Input Layer nuestro perceptrón multicapa aceptará 204 valores distintos los cuales equivalen a cada una de las bandas proporcionadas por nuestra cámara hiperespectral. Como salida y tras efectuar el softmax obtenemos una probabilidad de que sea una de las 5 clases a estudiar. Para las otras capas le hemos aportado al optimizador:

- Dropout. La capa de Dropout establece aleatoriamente las unidades de entrada en 0 con una frecuencia de "rate" en cada paso durante el entrenamiento, lo que ayuda a prevenir el sobreajuste. Las entradas que no se establecen en 0 se escalan por $1/(1 - \text{rate})$ para que la suma de todas las entradas no cambie [61].
- Dense. La capa Dense implementa la operación: $\text{salida} = \text{activación}(\text{punto_producto}(\text{entrada}, \text{kernel}) + \text{sesgo})$, donde activación es la función de activación aplicada elemento a elemento, pasada como argumento "activation", kernel es la matriz de pesos creada por la capa y sesgo es el vector de sesgo creado por la capa (solo aplicable si use_bias es True). Estos son todos atributos de Dense [62].

5.1.3 OPTIMIZADOR

El éxito de un proyecto de aprendizaje automático depende crucialmente de la elección de buenos hiperparámetros. A medida que el aprendizaje automático continúa madurando como campo, depender únicamente de prueba y error para encontrar buenos valores para estos parámetros (también conocido como "grad student descent") simplemente no es escalable. De hecho, muchos de los resultados de vanguardia en la actualidad, como EfficientNet, se descubrieron mediante algoritmos sofisticados de optimización de hiperparámetros.

Keras Tuner es un marco de optimización de hiperparámetros fácil de usar y distribuido que resuelve los puntos problemáticos de realizar una búsqueda de hiperparámetros. Keras Tuner facilita la definición de un espacio de búsqueda y aprovecha los algoritmos incluidos para encontrar los mejores valores de hiperparámetros. Keras Tuner viene con los algoritmos de Optimización Bayesiana, Hyperband y Búsqueda Aleatoria integrados, y también está diseñado para que los investigadores puedan ampliarlo fácilmente y experimentar con nuevos algoritmos de búsqueda [63].

En concreto para realizar pruebas con nuestros datos se han realizado pruebas para obtener los hiperparámetros necesarios con 2 optimizadores diferentes. Ambos han servido para determinar desde los pesos de la red hasta el número de capas y la función de activación óptima en nuestro caso. Estos han sido la Optimización Bayesiana y la Hyperband [64] [65].

5.1.4 RESULTADOS

Como resultado obtenemos un modelo capaz de realizar la clasificación pixel a pixel mediante la entrada de las 204 bandas con una precisión del 99,6 %, obteniendo así una imagen como la figura 26, en la cual se puede apreciar en color verde los píxeles pertenecientes a la cinta y en rojo los píxeles pertenecientes al ABS.

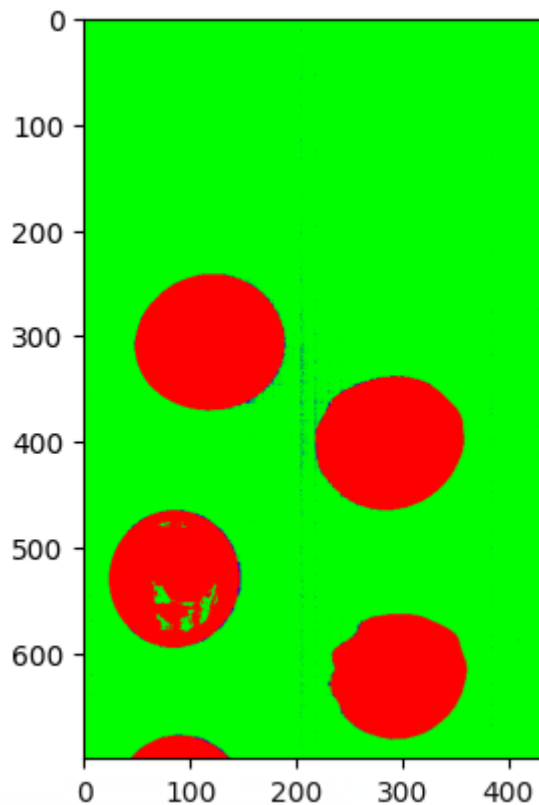


Figura 26 Imagen segmentada

Sin embargo, a pesar del alto grado de acierto en la predicción y debido a la naturaleza del perceptrón multicapa el tiempo de ejecución es muy elevado, llegando a ser de 45 segundos para una imagen de 2000 líneas como la de la figura 27, si bien es cierto que no es un tiempo de procesado óptimo, sí que es posible optimizar dicho tiempo con el uso de aceleración por GPU, paralelizando así todo el proceso y utilizando hardware más dedicado como una JetsonNano la cual está especializada en este tipo de proyectos.

A pesar de no servir para una aplicación donde se analiza el hipercubo con una imagen concreta, sí que es posible que se adapte a un HSI lineal en el cual, línea a línea se vaya procesando toda la imagen. Esto sería útil para detección de material no deseado, como por ejemplo si existe algún objeto de cartón o para prevención de riesgo evitando así material orgánico en la línea de procesado.

CAPÍTULO 6

CONCLUSIONES



Basándonos en los resultados obtenidos en este proyecto, es evidente que hemos logrado desarrollar un modelo altamente preciso para la clasificación pixel a pixel de imágenes hiperespectrales, alcanzando una impresionante precisión del 99,6%. Sin embargo, es importante destacar que, debido a la complejidad del perceptrón multicapa, el tiempo de ejecución es significativamente alto, lo que podría afectar la eficiencia del proceso. No obstante, esta limitación puede ser mitigada mediante la utilización de aceleración por GPU y hardware especializado como la JetsonNano, lo que permitiría optimizar el tiempo de procesamiento.

A partir de la experiencia adquirida en este proyecto, el siguiente paso lógico sería explorar y probar diferentes tipos de modelos utilizando los datos ya etiquetados. Entre los modelos más prometedores se encuentran el uso de una red neuronal convolucional (CNN) y el HSI, este último mencionado en varios rankings como una opción destacada para problemas de clasificación de imágenes hiperespectrales. Estos modelos podrían ofrecer un equilibrio entre precisión y eficiencia en el tiempo de procesamiento, lo que los convierte en candidatos ideales para aplicaciones prácticas.

Además, aunque el modelo actual puede no ser adecuado para aplicaciones que requieran un análisis detallado de hipercubos con imágenes específicas, sí podría adaptarse para su uso en HSI lineales, donde se procesa la imagen línea por línea. Esta capacidad sería especialmente útil para la detección de materiales no deseados o para la prevención de riesgos al evitar la presencia de materia orgánica en la línea de procesamiento. En resumen, este proyecto nos ha proporcionado una sólida base para continuar explorando y mejorando nuestras capacidades en el ámbito de la clasificación de imágenes hiperespectrales

BIBLIOGRAFÍA

- [1] F. Gruber, W. Grählert, P. Wollmann, y S. Kaskel, "Classification of black plastics waste using fluorescence imaging and machine learning," *Recycling*, vol. 4, no. 4, p. 40, 2019.
- [2] R. Geyer, J. R. Jambeck, y K. L. Law, "Production, use, and fate of all plastics ever made," *Sci. Adv.*, vol. 3, no. 7, p. e1700782, 2017.
- [3] C.-Q. Wang, H. Wang, J.-G. Fu, y Y.-N. Liu, "Flotation separation of waste plastics for recycling-A review," *Waste Manag.*, vol. 41, pp. 28-38, 2015.
- [4] G. L. Hearn y J. R. Ballard, "The use of electrostatic techniques for the identification and sorting of waste packaging materials," *Resour. Conserv. Recycl.*, vol. 44, no. 1, pp. 91-98, 2005.
- [5] A. Kuleke, C. Gurschler, G. Spöck, R. Leitner, y M. Kraft, "On-line classification of synthetic polymers using near infrared spectral imaging," *J. Near Infrared Spectrosc.*, vol. 11, no. 1, pp. 71-81, 2003.
- [6] S. Serranti, A. Gargiulo, y G. Bonifazi, "Hyperspectral imaging for process and quality control in recycling plants of polyolefin flakes," *J. Near Infrared Spectrosc.*, vol. 20, no. 5, pp. 573-581, 2012.
- [7] B. Riise, M. Biddle, y M. Fisher, "X-Ray Fluorescence Spectroscopy in plastics recycling," 2000.
- [8] S. Polat, A. Tremeau, y F. Boochs, "Rule-based classification of hyperspectral imaging data," arXiv [cs.CV], 2021.
- [9] Y.-H. Li et al., "Research and application of several key techniques in hyperspectral image preprocessing," *Front. Plant Sci.*, vol. 12, p. 627865, 2021.
- [10] A. Faltynkova, G. Johnsen, y M. Wagner, "Hyperspectral imaging as an emerging tool to analyze microplastics: A systematic review and recommendations for future development," *Microplast. Nanoplast.*, vol. 1, no. 1, 2021.
- [11] A. Signoroni, M. Savardi, A. Baronio, y S. Benini, "Deep learning meets hyperspectral image analysis: A multidisciplinary review," *J. Imaging*, vol. 5, no. 5, p. 52, 2019.
- [12] J. M. Amigo, "Practical issues of hyperspectral imaging analysis of solid dosage forms," *Anal. Bioanal. Chem.*, vol. 398, no. 1, pp. 93-109, 2010.

- [13] J. Nyman, "Pixel classification of hyperspectral images."
- [14] J. Nalepa, M. Antoniak, M. Myller, P. Ribalta Lorenzo, y M. Marcinkiewicz, "Towards resource-frugal deep convolutional neural networks for hyperspectral image segmentation," *Microprocess. Microsyst.*, vol. 73, art. no. 102994, 2020.
- [15] B. Tu, C. Zhou, D. He, S. Huang, y A. Plaza, "Hyperspectral classification with noisy label detection via superpixel-to-pixel weighting distance," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4116-4131, 2020.
- [16] J. Jiang, X. Qiao, y R. He, "Use of near-infrared hyperspectral images to identify moldy peanuts," *J. Food Eng.*, vol. 169, pp. 284-290, 2016. doi: 10.1016/j.jfoodeng.2015.09.013
- [17] E. Moliner y L. S. Romero, "Verónica vilaplana. Weakly supervised semantic segmentation for remote sensing hyperspectral imaging," en *Proceedings of the ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Piscataway, NJ: IEEE, 2020. doi: 10.1117/1.jei.28.3.033012
- [18] M. Zhang, Y. Jiang, C. Li, y F. Yang, "Fully convolutional networks for blueberry bruising and calyx segmentation using hyperspectral transmittance imaging," *Biosyst. Eng.*, vol. 192, pp. 159-175, 2020.
- [19] A. S. Shirokanev, N. Y. Ilyasova, y N. S. Demin, "Analysis of convolutional neural network for fundus image segmentation," *J. Phys. Conf. Ser.*, vol. 1438, no. 1, art. no. 012016, 2020.
- [20] S. Trajanovski, C. Shan, P. J. C. Weijtmans, S. G. B. de Koning, y T. J. M. Ruers, "Tongue tumor detection in hyperspectral images using deep learning semantic segmentation," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 4, pp. 1330-1340, 2021.
- [21] M. Couceiro y P. Ghamisi, "Case study III: Swarm robotics," en *Fractional Order Darwinian Particle Swarm Optimization*, Cham: Springer International Publishing, 2016, pp. 41-71.
- [22] L. Pang, J. Xiao, J. Ma, y L. Yan, "Hyperspectral imaging technology to detect the vigor of thermal-damaged *Quercus variabilis* seeds," *J. For. Res.*, vol. 32, no. 2, pp. 461-469, 2021.
- [23] K. V. Saravana y N. E. Rengasari, "A survey of hyperspectral image segmentation techniques for multiband reduction," *Aust. J. Basic Appl. Sci.*, vol. 15, pp. 446-451, 2019.
- [24] P. D. Dao, K. Mantripragada, Y. He, y F. Z. Qureshi, "Improving hyperspectral image segmentation by applying inverse noise weighting and outlier removal for optimal scale selection," *ISPRS J. Photogramm. Remote Sens.*, vol. 171, pp. 348-366, 2021.
- [25] S. N. Esfahani, V. Muthukumar, E. E. Regentova, K. Taghva, y M. Trabia, "Complex Food Recognition using Hyper-Spectral Imagery," en *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020.

- [26] Z. Xin, S. Jun, T. Yan, C. Quansheng, W. Xiaohong, y H. Yingying, "A deep learning based regression method on hyperspectral data for rapid prediction of cadmium residue in lettuce leaves," *Chemometr. Intell. Lab. Syst.*, vol. 200, art. no. 103996, 2020.
- [27] Q. Wang, Y. Liu, Q. Xu, J. Feng, y H. Yu, "Identification of mildew degrees in honeysuckle using hyperspectral imaging combined with variable selection," *J. Food Meas. Charact.*, vol. 13, no. 3, pp. 2157-2166, 2019.
- [28] X. Qiu, G. Jia, H. Zhao, y C. Zhang, "Antinoise estimation of temperature and emissivity for FTIR spectrometer data using spectral polishing filters: Design and comparison," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3292-3308, 2021.
- [29] A. L. Machidon, F. Del Frate, M. Picchiani, O. M. Machidon, y P. L. Ogrutan, "Geometrical approximated principal Component Analysis for hyperspectral image analysis," *Remote Sens. (Basel)*, vol. 12, no. 11, art. no. 1698, 2020.
- [30] M. P. Uddin, M. A. Mamun, M. I. Afjal, y M. A. Hossain, "Information-theoretic feature selection with segmentation-based folded principal component analysis (PCA) for hyperspectral image classification," *Int. J. Remote Sens.*, vol. 42, no. 1, pp. 286-321, 2021.
- [31] A. Datta y A. Chakravorty, "Hyperspectral image segmentation using multi-dimensional histogram over principal component images," en *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2018.
- [32] G. Turra, S. Arrigoni, y A. Signoroni, "CNN-based identification of hyperspectral bacterial signatures for digital microbiology," en *Image Analysis and Processing - ICIAP 2017*, Cham: Springer International Publishing, 2017, pp. 500-510.
- [33] B. Arad y O. Ben-Shahar, "Filter Selection for Hyperspectral Estimation," en *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [34] Y. Fu, T. Zhang, Y. Zheng, D. Zhang, y H. Huang, "Joint camera spectral sensitivity selection and hyperspectral image recovery," en *Computer Vision – ECCV 2018*, Cham: Springer International Publishing, 2018, pp. 812-828.
- [35] S. Galliani, C. Lanaras, D. Marmanis, E. Baltsavias, y K. Schindler, "Learned Spectral Super-Resolution," arXiv [cs.CV], 2017.
- [36] S. A. Robila, "Independent Component Analysis," en *Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 109-132.
- [37] A. Zabalza, J. Ren, J. Zheng, C. Q. Huimin Zhao, Z. Yang, y S. Marshall, "Novel Segmented Stacked Auto Encoder for Effective Dimensionality Reduction and Feature Extraction in Hyperspectral Imaging," *Neurocomputing*, vol. 185, pp. 1-10, 2016.

- [38] X. Cao et al., "Computational Snapshot Multispectral Cameras: Toward dynamic capture of the spectral world," *IEEE Signal Process. Mag.*, vol. 33, no. 5, pp. 95-108, 2016.
- [39] R. Pu, *Hyperspectral remote sensing: Fundamentals and practices*. Londres, Inglaterra: CRC Press, 2017.
- [40] W. Hu, Y. Huang, L. Wei, F. Zhang, y H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sens.*, vol. 2015, art. no. 1-12, 2015.
- [41] L. Mou, P. Ghamisi, y X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639-3655, 2017.
- [42] "El Perceptrón Multicapa," *Interactivechaos.com*. [En línea]. Disponible en: <https://interactivechaos.com/es/manual/tutorial-de-deep-learning/el-perceptron-multicapa>. [Consultado: 23-jul-2023].
- [43] J. Pereira, "Perceptrón Multicapa y Algoritmo de Retropropagación," *Mql5.com*, 03-mar-2021. [En línea]. Disponible en: <https://www.mql5.com/es/articles/8908>. [Consultado: 23-jul-2023].
- [44] "IBM Documentation," *Ibm.com*, 22-mar-2021. [En línea]. Disponible en: <https://www.ibm.com/docs/es/spss-statistics/25.0.0?topic=networks-multilayer-perceptron>. [Consultado: 23-jul-2023].
- [45] "Convolutional Neural Network: definición y funcionamiento," *Formation Data Science / Datascientest.com*, 16-dic-2021. [En línea]. Disponible en: <https://datascientest.com/es/convolutional-neural-network-es>. [Consultado: 23-jul-2023].
- [46] Na, "¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador," *Aprende Machine Learning*, 29-nov-2018. [En línea]. Disponible en: <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>. [Consultado: 23-jul-2023].
- [47] "Cuándo usar las redes neuronales MLP, CNN y RNN," *TopBigData*, 18-ago-2022. [En línea]. Disponible en: <https://topbigdata.es/cuando-usar-las-redes-neuronales-mlp-cnn-y-rnn/>. [Consultado: 23-jul-2023].
- [48] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, y R. Yang, "Salient object detection in the deep learning era: An in-depth survey," arXiv [cs.CV], 2019.
- [49] L. Gonzalez, "K Vecinos más Cercanos - Teoría - Aprende IA," *Aprende IA*, 19-jul-2019. [En línea]. Disponible en: <https://aprendeia.com/algoritmo-k-vecinos-mas-cercanos-teoria-machine-learning/>. [Consultado: 23-jul-2023].
- [50] R. Díaz, "Algoritmo KNN - cómo funciona y ejemplos en Python," *The Machine Learners*, 12-may-2020. [En línea]. Disponible en: <https://www.themachinelearners.com/algoritmo-knn/>. [Consultado: 23-jul-2023].

- [51] "Descubra el algoritmo KNN: un algoritmo de aprendizaje supervisado," *Formation Data Science | Datascientest.com*, 28-dic-2021. [En línea]. Disponible en: <https://datascientest.com/es/que-es-el-algoritmo-knn>. [Consultado: 23-jul-2023].
- [52] "El algoritmo K-NN y su importancia en el modelado de datos," *Merkle*. [En línea]. Disponible en: <https://www.merkle.com/es/es/blog/algoritmo-knn-modelado-datos>. [Consultado: 23-jul-2023].
- [53] Y. Freund y R. Schapire, "Machine Learning: Proceedings of the Thirteenth International conference," pp. 148-156.
- [54] C. M., *Pattern Recognition and Machine Learning*, 1a ed. New York, NY: Springer, 2006.
- [55] T. Liu, *Polymer Semantic Segmentation with Hyperspectral Imaging and Artificial Neural Networks*. NHLStenden Centre of Expertise in Computer Vision & Data Science.
- [56] R. Bro y A. K. Smilde, "Principal component analysis," *Anal. Methods*, vol. 6, no. 9, pp. 2812-2831, 2014.
- [57] B. Carrera, V. L. Piñol, J. B. Mata, y K. Kim, "A machine learning based classification models for plastic recycling using different wavelength range spectrums," *J. Clean. Prod.*, vol. 374, art. no. 133883, 2022.
- [58] S. B. Concari, "Las teorías y modelos en la explicación científica: implicancias para la enseñanza de las ciencias," *Ciênc. Educ. (Bauru)*, vol. 7, no. 1, pp. 85-94, 2001.
- [59] "Radial Basis Function Networks," *DeepAI*, 17-may-2019. [En línea]. Disponible en: <https://deepai.org/machine-learning-glossary-and-terms/radial-basis-function-network>. [Consultado: 23-jul-2023].
- [60] "Radial basis networks and custom keras layers," *Kaggle.com*, 03-feb-2019. [En línea]. Disponible en: <https://www.kaggle.com/code/residentmario/radial-basis-networks-and-custom-keras-layers>. [Consultado: 23-jul-2023].
- [61] "Tf.Keras.Layers.Dropout," *TensorFlow*. [En línea]. Disponible en: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout. [Accedido: 24-jul-2023].
- [62] "Tf.keras.layers.dense," *TensorFlow*. [En línea]. Disponible en: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense. [Accedido: 24-jul-2023].
- [63] "Hyperparameter tuning with Keras Tuner," *Tensorflow.org*. [En línea]. Disponible en: <https://blog.tensorflow.org/2020/01/hyperparameter-tuning-with-keras-tuner.html>. [Accedido: 24-jul-2023].
- [64] E. A. Smirnov, D. M. Timoshenko, y S. N. Andrianov, "Comparison of regularization methods for ImageNet classification with deep convolutional neural networks," *AASRI Procedia*, vol. 6, pp. 89-94, 2014.

[65] C. Yuan y S. S. Aghaian, "A comprehensive review of Binary Neural Network," arXiv [cs.NE], 2021.



ÍNDICE DE FIGURAS

Figura 1 Conjunto de datos unificado	16
Figura 2 Conjunto de datos separados en capas	16
Figura 3 comparación de una neurona biométrica vs neurona artificial.....	31
Figura 4 Gráfica función de activación sigmoide.....	32
Figura 5 Gráfica función de activación hiperbólica	33
Figura 6 Gráfica función de activación RELU	33
Figura 7 Gráfica función de activación softmax	34
Figura 8 Ejemplo perceptrón multicapa en playground tensorflow	35
Figura 9 Gráfico ejemplificativo de una SVM lineal	36
Figura 10 Gráfico de una capa convolucional	38
Figura 11 Gráfico de una capa de agrupación	39
Figura 12 Gráfico de una capa de activación	39
Figura 13 Gráfica de probabilidades a priori.....	43
Figura 14 Gráfica de la verosimilitud.....	43
Figura 15 Gráfica de las probabilidades a posteriori	44
Figura 16 Gráfica del factor de Bayes	44
Figura 17 Ejemplo de árbol de decisión	45
Figura 18 Representación de una imagen hiperespectral	544
Figura 19 Ejemplo de representación de las bandas en diversos materiales	545
Figura 20 Puntos seleccionados para hacer la segmentación	546
Figura 21 Imagen segmentada tras la aplicación de SAM	547
Figura 22 Sistema de manejo de HSI	58
Figura 23 ML Pipeline por TensorFlow	60
Figura 24 Visualización del entrenamiento con tensorboard.....	60
Figura 25 Representación del perceptrón multicapa en play TensorFlow.....	64
Figura 26 Imagen segmentada.....	69

ÍNDICE DE TABLAS

Tabla 1 Materiales estudiados.....	51
------------------------------------	----



ANEXO I: CÓDIGO EMPLEADO PARA LA REALIZACIÓN DEL TRABAJO.



Junto a la entrega de la memoria de este proyecto se hace entrega del código empleado a la hora de realizar el mismo, éste consta del fichero general (TFG_IA) dentro del cual están las carpetas `Classifiers`, `dataPreparation` y `Preprocess`.

- TFG_IA
 - **Classifiers:** Aquí se encuentra el perceptrón multicapa y todos los archivos necesarios para trabajar con él.
 - **dataPreparation:** En esta carpeta se encuentran los archivos relacionados con el manejo de los hipercubos, con ella podrás cargar hipercubos y segmentarlos manualmente.
 - **Preprocess:** Aquí están todas las herramientas necesarias para aplicar preprocesamiento a nuestros hipercubos.