



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

CLASIFICACIÓN EN IMÁGENES SATELITALES: SUPERFICIE CONSTRUIDA Y USO  
DE TIERRA

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL MATEMÁTICO

RODRIGO FELIPE LARA MOLINA

PROFESOR GUÍA:  
FELIPE TOBAR HENRÍQUEZ

MIEMBROS DE LA COMISIÓN:  
JAIME ORTEGA PALMA  
PABLO ESTÉVEZ VALENCIA

Este trabajo ha sido parcialmente financiado por CMM - Conicyt PIA AFB170001

SANTIAGO DE CHILE  
2018

RESUMEN DE LA MEMORIA PARA OPTAR  
AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO  
POR: RODRIGO FELIPE LARA MOLINA  
FECHA: 2018  
PROF. GUÍA: FELIPE TOBAR HENRÍQUEZ

## CLASIFICACIÓN EN IMÁGENES SATELITALES: SUPERFICIE CONSTRUIDA Y USO DE TIERRA

El propósito de esta memoria es aportar elementos para el cálculo de dos indicadores que son parte de los Objetivos de Desarrollo Sustentable (ODS), lanzados por la ONU en 2015. Ambos se inscriben dentro del objetivo 11, “*Lograr que las ciudades y los asentamientos humanos sean inclusivos, seguros, resilientes y sostenibles*”.

El primer caso se enfoca en el indicador ODS 11.3.1, “*cociente entre la tasa de consumo de tierras y la tasa de crecimiento de la población*”. Se utilizan imágenes satelitales de acceso público (*Landsat 8*) y de resolución media (30 m por píxel), y etiquetas provenientes del Atlas de Expansión Urbana, proyecto de *New York University*. Se busca identificar a nivel de píxel la superficie construida presente en las imágenes. Esto se realiza mediante diferentes algoritmos de clasificación: regresión logística, máquinas de soporte vectorial, *gradient tree boosting* y redes neuronales. Se muestra que la solución con máquinas de soporte vectorial posee una ventaja significativa en su rendimiento.

El segundo caso se centra en el indicador ODS 11.7.1 “*cuota promedio del área edificada de las ciudades que es espacio abierto para uso público para todos [...]*”. Se utilizan imágenes de alta resolución (0.5 m por píxel) y sus metadatos asociados, provenientes de un desafío en línea, donde se busca clasificar entre 62 clases de edificaciones y usos de tierra como por ejemplo, aeropuertos, escuelas, cultivos, minas, etc. Se emplea un modelo a nivel del estado del arte en clasificación de imágenes, *Inception-V3*, preentrenado en ImageNet 2012, que es capaz de combinar la información de la imagen con los metadatos correspondientes. El modelo obtenido se posiciona dentro del tercio superior de la competencia.

Los resultados obtenidos en ambos experimentos permiten suponer que este trabajo presenta propuestas metodológicas factibles, que en el primer caso facilitan cálculo del indicador, y en el segundo, vislumbran la conveniencia de una solución basada en imágenes satelitales.

En el Capítulo 1 se introduce el trabajo de memoria y se definen los objetivos del proyecto. En el Capítulo 2 se ahonda en la estructura de las imágenes satelitales, se define el problema de clasificación, se describen los algoritmos de clasificación que son empleados y se analiza brevemente el estado del arte para ambas problemáticas. En los capítulos 3 y 4, se profundiza en los experimentos de clasificación de superficie construida y de edificaciones respectivamente. Finalmente, en el Capítulo 5 se desarrollan algunas recomendaciones y se sintetizan las contribuciones de este trabajo.

Se deja de manifiesto que una parte de este trabajo fue presentado en el concurso de *posters* de la Escuela de Verano Interamericana de Inteligencia Computacional (EVIC) 2017, realizada en Valparaíso, Chile y otra, en la *Machine Learning Summer School* (MLSS) 2018 realizada en Buenos Aires, Argentina.



*A mis padres, Pamela, Mauricio y Álvaro por apoyarme siempre.*



# Agradecimientos

En primer lugar, quiero agradecer a mi familia, por estar a mi lado a lo largo de todo mi proceso formativo, por inspirarme y por constituir gran parte de lo que soy. Principalmente agradecer a mi madre Pamela y a mis padres Mauricio y Álvaro. También hacer una mención honrosa a mis abuelos, Sonia, Enrique, Silvia, Eva y Pedro; a mis tíos Mabel, Ricardo, Marco, Lidia y Patricio; a mis hermanos Roberto, Benjamín, Montserrat, Álvaro, Sofía y Cristóbal; a mis primas Trinidad, Sol, Violeta y Julieta; a mi madrina Silvia y a mi madrastra Cecilia.

A todos los amigos que encontré durante mi paso por la universidad, un proceso de aprendizaje intenso y arduo que finalizo, en gran parte, gracias al apoyo que encontré en ellos. Piero, Sergio, Guido, Camila, Matías, José, Andrés, Diana, Víctor, Guillaume y Pierre, gracias por estar ahí cuando necesitaba una palabra de aliento, un consejo, una carcajada o simplemente la compañía de personas excepcionales como ustedes lo son.

Deseo además agradecer a mi familia chilena en Francia: Joaquín, Alonso, Esteban, Op-hélie, Pablo, Gary, Cécile, Lucas, Joaquín y Ricardo. El desafío de pasar dos años de mi formación fuera de Chile, habría sido imposible sin el apoyo de ustedes. Agradezco mucho haber tenido la oportunidad de encontrarlos en este viaje. Mencionar también a mis amigos del *Club Raid*, en especial a Guillaume, Paul, Solène, Pierre-Victor, Maxime, Bruno, Pierre, Fabrice, François y Pierrick, por ser quienes me hicieron descubrir la cultura francesa desde dentro y me permitieron reencontrar en su compañía, mi encanto por la naturaleza y el deporte.

No puedo dejar de mencionar a mis amigos del colegio, que han estado en mi vida por más de la mitad de mi historia: Ignacio, Javier, Paulo, Axel, Karla, Felipe, Carlos, David, Branko, Esteban, Andrea, Javiera, Karla. Gracias por darme la oportunidad de crecer con y junto a ustedes. Por brindarme un espacio para tomar la vida con humor y alegría, pero también con profundidad, en momentos en los que lo ameritan.

Por último, mi gratitud a los miembros de la comisión y a GAMES, especialmente a mis compañeros Charles, Mauricio, Alejandro y a mi guía Felipe. Con su apoyo hoy cierro mi memoria, lleno de satisfacción y orgullo, por todo el esfuerzo y el crecimiento que este proceso significó para mi.



# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto general . . . . .	1
1.2. Definición del problema . . . . .	2
1.2.1. Objetivos . . . . .	3
1.2.2. Alcances . . . . .	3
<b>2. Preliminares</b>	<b>4</b>
2.1. Imagen satelital . . . . .	4
2.2. Clasificación . . . . .	6
2.2.1. Conjuntos de entrenamiento, validación y prueba . . . . .	6
2.2.2. Algoritmos de clasificación . . . . .	7
2.2.3. Medidas de desempeño en clasificación . . . . .	21
2.3. Estado del arte . . . . .	22
2.3.1. Clasificación de superficie construida . . . . .	22
2.3.2. Clasificación de edificaciones y usos de suelo . . . . .	23
<b>3. Detección de superficie urbana construida</b>	<b>25</b>
3.1. Motivación . . . . .	25
3.2. Datos . . . . .	26
3.3. Metodología . . . . .	28
3.3.1. Construcción de la base de imágenes . . . . .	28
3.3.2. Configuración del experimento . . . . .	29
3.3.3. Preprocesamiento . . . . .	32
3.3.4. Selección de modelos . . . . .	33
3.4. Resultados . . . . .	34
3.5. Análisis y discusión . . . . .	37
<b>4. Clasificación de edificaciones y usos de suelo</b>	<b>39</b>
4.1. Motivación . . . . .	39
4.2. Datos . . . . .	40
4.3. Metodología . . . . .	43
4.3.1. Arquitectura del modelo . . . . .	43
4.3.2. Procedimientos de entrenamiento . . . . .	43
4.3.3. Preprocesamiento . . . . .	45
4.3.4. Métrica de desempeño . . . . .	47
4.4. Resultados . . . . .	48

4.4.1.	Comparación de los procedimientos de entrenamiento . . . . .	48
4.4.2.	Tabla de posiciones FMoW . . . . .	49
4.4.3.	Clasificación de espacios abiertos . . . . .	49
4.5.	Análisis y discusión . . . . .	52
4.5.1.	Impacto del desequilibrio de clases . . . . .	53
4.5.2.	Comparación con los concursantes de FMoW . . . . .	54
4.5.3.	Factibilidad de la detección automática de espacios abiertos . . . . .	54
<b>5.</b>	<b>Conclusiones y recomendaciones</b>	<b>56</b>
5.1.	Conclusiones . . . . .	56
5.2.	Recomendaciones . . . . .	56
	<b>Bibliografía</b>	<b>58</b>

# Índice de Tablas

2.1.	Comparación de resolución de imágenes de los problemas CSC y CEUS. . . .	5
2.2.	Ejemplos de <i>kernels</i> para MSV. . . . .	10
2.3.	Modelo <i>Inception-V2</i> con estructura adaptada para inferencia en el desafío ILSVRC 2012 [28] . . . . .	19
3.1.	Detalle de resolución espectral y espacial de <i>Landsat 8</i> . . . . .	27
3.2.	Variación de hiperparámetros para cada algoritmo. . . . .	34
3.3.	Número de neuronas por capa para las diferentes arquitecturas de RN. . . .	34
3.4.	Hiperparámetros óptimos para cada algoritmo. . . . .	35
3.5.	P-valor del Test de Wilcoxon en el f1-score obtenido en el conjunto de prueba.	35
4.1.	División de los conjuntos de entrenamiento, validación y prueba. . . . .	44
4.2.	Metadatos utilizados como entrada del modelo. . . . .	47
4.3.	Tiempo promedio de entrenamiento por procedimiento de entrenamiento. . .	48
4.4.	Desempeño promedio en la clasificación de espacios abiertos. . . . .	51

# Índice de Ilustraciones

2.1. Diagrama explicativo de una imagen multi-espectral. . . . .	5
2.2. Regresión logística para probabilidad de no pago en función del balance de la cuenta bancaria de los clientes. . . . .	8
2.3. Ejemplo de separación de dos clases a través de un plano lineal. . . . .	9
2.4. Diagrama de un árbol de regresión para la predicción del logaritmo del salario de un jugador de béisbol. . . . .	11
2.5. Estructura básica de la red neuronal totalmente conectada, con una capa oculta.	13
2.6. Red convolucional de 2 dimensiones. . . . .	15
2.7. Operación de convolución. . . . .	15
2.8. Otros parámetros que determinan la operación de convolución. . . . .	16
2.9. Ilustración del procedimiento del operador <i>max-pooling</i> de tamaño $2 \times 2$ y <i>stride</i> 2. . . . .	16
2.10. Arquitectura del modelo <i>Inception-V1</i> . . . . .	17
2.11. Diferentes versiones del bloque <i>Inception</i> . . . . .	18
2.12. Estructura del bloque auxiliar para los diferentes modelos <i>Inception</i> . . . . .	19
2.13. Bloque raíz del modelo <i>Inception-V1</i> . . . . .	20
2.14. Clasificador de salida del modelo <i>Inception-V1</i> . . . . .	20
2.15. Matriz de confusión para un problema de clasificación binaria. $C_1$ representa la clase positiva y $C_2$ la negativa. . . . .	21
2.16. Matriz de confusión para un problema de clasificación de $K$ clases. . . . .	21
3.1. Emisión atmosférica y posicionamiento de las bandas de captura de satélite <i>Landsat 8</i> . . . . .	27
3.2. Consolidación de imagen satelital y etiquetas del Atlas de Expansión Urbana [16]. . . . .	28
3.3. Distribución de clases aglomeradas en el problema CSC. . . . .	29
3.4. Distribución de clases en las ciudades del estudio en el problema CSC. . . . .	30
3.5. <i>Scatterplot Matrix</i> de una muestra de 10 millones de píxeles etiquetados. . . . .	31
3.6. Representación t-SNE de una muestra de 100 mil píxeles . . . . .	32
3.7. Diagrama de caja de <i>f1-score</i> para los clasificadores seleccionados. . . . .	35
3.8. Diagrama de caja de precisión, exhaustividad y <i>f1-score</i> para los clasificadores seleccionados en cada una de las clases . . . . .	36
3.9. Resultados para Caracas: (a) Comparación entre etiquetas verdaderas y predichas. (b) Matriz de confusión . . . . .	37
3.10. Caracas - Venezuela, Landsat 8 - RGB, 2014. . . . .	38

4.1.	Muestra aleatoria de una imagen por clase del conjunto de datos FMoW. . .	41
4.2.	Distribución del número de imágenes por POI que corresponden a observaciones en diferentes instantes de tiempo en el conjunto de datos FMoW. . . . .	42
4.3.	División del conjunto de datos FMoW. . . . .	42
4.4.	Arquitectura del modelo de referencia FMoW. . . . .	43
4.5.	Soporte de cada una de las clases a nivel de POI. . . . .	45
4.6.	Procedimiento de aumentación de tamaño de los <i>bounding box</i> . . . . .	46
4.7.	Rotación de imágenes en el conjunto de entrenamiento. . . . .	46
4.8.	<i>F-score</i> en el conjunto de prueba por procedimiento de entrenamiento. . . . .	48
4.9.	Variación de métricas durante el proceso de entrenamiento. . . . .	49
4.10.	Matriz de confusión por POI normalizada del modelo NN-1. . . . .	50
4.11.	<i>F1-score</i> por clase a nivel de POI. . . . .	51
4.13.	Matriz de confusión para el modelo con mejor <i>f1-score</i> en la clasificación de espacios abiertos. . . . .	51
4.12.	Posicionamiento del mejor procedimiento de entrenamiento entre los concursantes del desafío FMoW. . . . .	52
4.14.	<i>F1-score</i> y soporte por clase a nivel de POI. . . . .	53
4.15.	Ejemplos cualitativos de mala clasificación en las clases con <i>f1-score</i> más bajo. . . . .	54



# Capítulo 1

## Introducción

A continuación, se expone el marco contextual de la memoria. Primero, se presentan los Objetivos de Desarrollo Sostenible (ODS), dentro de los que se enmarcan los indicadores que motivan la realización de este trabajo. Cada uno de los indicadores serán definidos, en la medida en que sea necesario, en los capítulos dedicados al problema que estos motivan. En segundo lugar, se describen los objetivos generales y específicos de la memoria, para finalizar con la exposición de los alcances de esta.

### 1.1. Contexto general

Por primera vez en el año 2000, la Organización de las Naciones Unidas (ONU) y los países miembros definieron metas mundiales a realizar. Dichas metas, llamadas Objetivos de Desarrollo del Milenio (ODM) fueron un ambicioso proyecto que buscaba fijar un horizonte común y generar sinergias entre diversas organizaciones e instituciones tanto del desarrollo, como de la academia, de la sociedad civil, y del sector público y privado [10]. Estos objetivos caducaron en el año 2015 dejando éxitos, fracasos y aprendizajes, pero al mismo tiempo dando paso a una nueva generación de objetivos mundiales, mucho más amplia y ambiciosa que los ODM. Se trata de los Objetivos de Desarrollo Sostenible (ODS).

Los ODS, están conformados por 17 objetivos, que agrupan en 169 metas cuantificadas a través de 232 indicadores que abarcan temáticas, tanto sociales, como ambientales, políticas o económicas [11]. Cada objetivo se divide en metas y cada meta presenta indicadores que miden el grado de avance hacia el logro de dichas metas. El objetivo 11, *“lograr que las ciudades y los asentamientos humanos sean inclusivos, seguros, resilientes y sostenibles”* es el que motiva la presente memoria. El uso de imágenes satelitales y de algoritmos de clasificación, para responder a problemáticas de medición que levanta este objetivo, es el eje principal de este trabajo.

La meta 11.3 *“aumentar la urbanización inclusiva y sostenible y la capacidad para la planificación y la gestión participativas, integradas y sostenibles de los asentamientos humanos [...]”* y más específicamente su indicador 11.3.1, *“cociente entre la Tasa de Consumo de Tie-*

rras (TCT) y la Tasa de Crecimiento de Población (TCP)” suscita la primera parte de este trabajo (Capítulo 3) en donde se utilizan imágenes satelitales de libre acceso y de resolución media (100-15m por píxel<sup>1</sup>) provenientes de *Landsat 8* (cf. Sección 3.2). Esta elección se justifica en la pretensión de avanzar hacia una solución de bajo costo para el cálculo del indicador 11.3.1 de los ODS, que sea accesible para diferentes entidades de estadística pertenecientes a países de diversos niveles de ingreso. Por esta misma razón, se trabajará con herramientas informáticas de código abierto, más específicamente en el lenguaje de programación Python con sus librerías dedicadas de código abierto.

Por otro lado, la meta 11.7 “proporcionar acceso universal a zonas verdes y espacios públicos seguros, inclusivos y accesibles [...]” y concretamente, su indicador 11.7.1 “cuota promedio del área edificada de las ciudades que es espacio abierto para uso público para todos, desagregadas por sexo, edad y personas con discapacidades” motiva el problema de detección de construcciones y uso de suelo que es desarrollado en el Capítulo 4. Dada la falta de datos etiquetados para dicha tarea, se decide abordar el desafío de *Mapa Funcional del Mundo* [13] (FMoW, por sus siglas en inglés) que contiene 62 clases de edificaciones y uso de suelos, entre las cuales algunas pueden representar espacio abierto. Por otro lado las imágenes de este conjunto de datos (alrededor de 500 mil imágenes de 130 mil puntos de interés) presentan una resolución de 0.5 m. por píxel, lo que es una resolución dos órdenes de magnitud inferior a la de las imágenes de libre acceso que es de 10 m. por píxel<sup>2</sup>. Si bien dichas imágenes provienen de satélites de acceso pagado, se decide incursionar en el desafío FMoW como una primera aproximación metodológica, para constatar la factibilidad del reconocimiento automático de espacios abiertos en imágenes satelitales.

Es importante destacar que existe una categorización de los indicadores ODS que los divide en 3 grupos según el estado de consolidación de su metodología de cálculo y de la producción de datos existentes para su cómputo (*tier 1*, *tier 2* y *tier 3*). Uno de los indicadores *tier 2*, que tienen una metodología establecida internacionalmente pero donde los datos disponibles para los países no son suficientes, es el indicador 11.3.1. Por su parte el indicador 11.7.1 forma parte de los indicadores *tier 3*, que no poseen ni la metodología, ni los datos para poder calcularse [14]. Este trabajo busca aportar elementos a la metodología de cálculo de ambos indicadores y es este hecho, el que le da relevancia.

## 1.2. Definición del problema

La presente memoria abordará dos problemáticas de clasificación en imágenes satelitales. Una a nivel local (píxel), la detección de superficie urbana construida (de ahora en adelante, CSC) y otra a nivel de vecindario (o sección de imagen), el reconocimiento de edificaciones y de uso de suelo (CEUS).

---

<sup>1</sup>Resolución de  $k$  metros por píxel: cada píxel representa en promedio un área de  $k^2$  metros cuadrados.

<sup>2</sup>Actualmente la resolución espacial más alta en un satélite de libre acceso la posee el satélite de Agencia Espacial Europea (ESA, por sus siglas en inglés) Sentinel 2.

### 1.2.1. Objetivos

El objetivo general es aportar avances en la metodología de cálculo de indicadores ODS en el ámbito de las estadísticas urbanas que presentan desafíos metodológicos por resolver.

El trabajo se centrará en los dos indicadores mencionados anteriormente y los objetivos específicos para cada uno de ellos son:

1. Para CSC, aportar elementos a la metodología de clasificación de superficie urbana construida, mediante el uso de imágenes satelitales de libre acceso y, la construcción y evaluación de diferentes modelos de clasificación de Aprendizaje de Máquinas (AM) supervisado (cf. Sección 2.2). De esta manera, se busca generar recomendaciones que permitan hacer más eficiente el cálculo del indicador 11.3.1 de los ODS (cf. Sección 3.1).
2. Para CEUS, constatar la factibilidad del uso de imágenes satelitales de alta resolución y modelos de AM supervisado para la clasificación de espacios abiertos urbanos. Así, se busca aportar elementos que faciliten la definición de metodologías de cálculo para el indicador 11.7.1 de los ODS. Todo lo anterior dentro del marco del desafío FMoW que es el hilo conductor en esta parte del trabajo.

Los objetivos específicos son afrontados separadamente en los capítulos 3 y 4 de la presente memoria.

### 1.2.2. Alcances

Si bien el indicador ODS 11.3.1 motiva el problema CSC, no se procede al cálculo de este indicador. Su cómputo implicaría medir la superficie urbana construida en diferentes años para así obtener la tasa de consumo de tierras. Aunque esta problemática es parcialmente atacada en el experimento CSC, presente en el Capítulo 3, el trabajo se enfoca en encontrar el modelo más adecuado para el reconocimiento de la superficie construida y no incursiona en el cálculo de la tasa de crecimiento urbano, ya que sólo son analizadas imágenes *circa* 2014, ni tampoco en el cálculo del cociente mencionado. Además dicho ejercicio sólo se centra en 26 ciudades de América Latina, para las que se poseen etiquetas provenientes del *Atlas de Expansión Urbana* [15] e imágenes capturadas por el satélite *Landsat 8* [2].

Análogamente, aunque el experimento CEUS es motivado por el indicador ODS 11.7.1, la presente memoria se enfoca en construir un modelo con buen desempeño para el desafío FMoW (relativo a los resultados obtenidos por los concursantes de este desafío), generando así como subproducto, un modelo capaz de reconocer espacios abiertos urbanos. Lo anterior se debe a que entre las 62 clases de edificaciones y usos de suelo del desafío, dos de ellas entran en la categoría de espacio abierto urbano. Cabe la posibilidad de que posteriormente dicho modelo pueda ser utilizado como elemento base de la detección automática de espacios abiertos en áreas edificadas, pero no se profundiza en el diseño de una solución acabada, ya que esto escapa a los alcances de este trabajo.

# Capítulo 2

## Preliminares

En este capítulo se tratan los elementos y conceptos fundamentales necesarios para la comprensión de esta investigación. En primer lugar, se explica la imagen satelital, pues es el objeto mediante el cual se estudiarán los dos problemas abordados en este trabajo. Luego, se explica el concepto de clasificación como tarea del aprendizaje de máquinas supervisado y se describen tanto los algoritmos que serán utilizados como las medidas de desempeño con las que son evaluados los modelos generados. Finalmente, se analiza el estado del arte de ambos problemas abordados en la memoria: Clasificación de Superficie Construida (CSC) y Clasificación de Edificaciones y Uso de Suelo (CEUS).

### 2.1. Imagen satelital

Las imágenes satelitales con las que se trabaja en esta investigación son multiespectrales, esto es, codifican la información de cada píxel en diferentes bandas de longitudes de onda, permitiendo así la identificación de características de los materiales fotografiados que están presentes más allá del espectro visible. Además, cada imagen contiene metadatos que permiten identificar cuáles son las coordenadas exactas representadas en cada píxel y por consiguiente, la superficie cubierta por cada uno de sus píxeles. Adicionalmente, los metadatos incluyen información que permite realizar una serie de correcciones asociadas a los demás factores que influyen en la captura de la imagen, como la posición del sol o ángulo de captura, entre otros. Para cada píxel de la imagen multiespectral se captura la intensidad de la luz (o la cantidad de fotones) proveniente de la zona espacial que representa dicho píxel, para cada una de las bandas de longitudes de onda que la cámara es capaz de adquirir (cf. Figura 2.1). En el caso particular de la imagen multiespectral satelital, se distinguen 4 tipos de resolución [8]:

- **Resolución espacial:** tamaño equivalente promedio de la arista del píxel en la superficie de la tierra.
- **Resolución temporal:** frecuencia temporal con la que el satélite sobrevuela y captura información de un área determinada.

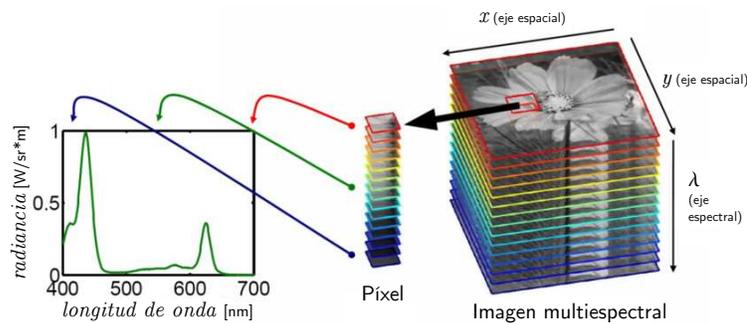


Figura 2.1: Diagrama explicativo de una imagen multi-espectral.  
Fuente: Adaptado de [38]

- **Resolución espectral:** rango de longitudes de onda capturadas y número de bandas.
- **Resolución radiométrica:** nivel de diferenciación de diferentes niveles de intensidad de radiación para cada una de las bandas.

Dentro de estos 4 tipos de resoluciones, el único que no será considerado para los fines de esta memoria es la resolución temporal, pues en CSC sólo se analizará una única imagen por cada ciudad. En CEUS los instantes de tiempo son definidos por quienes confeccionaron el conjunto de datos FMoW. En la Tabla 2.1 se describen las resoluciones relevantes para los problemas abordados:

Resolución	CSC	CEUS
<b>Espacial</b> (número de bandas)	15(1), 30(8) y 100(1)[m/píxel]	0.5(1), 2(3 y 7) [m/píxel]
<b>Espectral</b>	11 bandas	8 y 4 bandas
<b>Radiométrica</b>	12 bits	16 bits

Tabla 2.1: Comparación de resolución de imágenes de los problemas CSC y CEUS.

Las imágenes registran la intensidad de la luz recibida en cada una de las bandas en un valor digital ( $VD$ ), que mediante una transformación lineal puede ser transformado en radiancia. La radiancia es una magnitud física que se mide en  $\frac{W}{sr \times m^2}$ , es decir captura la intensidad de la luz en el tiempo (Watts) emitida por unidad de superficie (metro cuadrado) en una cierta dirección angular sólida (esterorradián).

Otra magnitud física que es relevante para el estudio de las imágenes satelitales es la irradiancia. Al igual que la radiancia, esta mide la intensidad de la luz en el tiempo, pero no emitida o reflejada por una superficie, sino que la que es recibida en una superficie. Por esto, su unidad de medida es  $\frac{W}{m^2}$  [8].

Para el problema CSC es necesario hacer comparables imágenes que fueron capturadas bajo condiciones de captura y de radiación solar diferentes. Para ello se utiliza la corrección *Top of the Atmosphere* (ToA), que permite transformar los valores de los píxeles en cantidades físicas de reflectancia a nivel de la parte superior de la atmósfera. Se profundiza en esta corrección en la Sección 3.3.3.

En cuanto al problema CEUS, no se efectúa ninguna corrección atmosférica, aún si las condiciones de captura son igualmente variables para las imágenes tratadas. En lugar de ello los metadatos de las imágenes satelitales son entregadas al modelo como características que permiten efectuar la decisión. Más detalles sobre el preprocesamiento y la utilización de los metadatos son entregados en la Sección 4.3.2.

## 2.2. Clasificación

El aprendizaje de máquinas (AM) supervisado, busca crear modelos que se construyan a partir de un conjunto de datos u observaciones, que consisten en un par (*características, etiquetas*). Dichos modelos son construidos con la finalidad de que sean capaces de extraer información que permita inferir las *etiquetas* a partir de las *características* de datos no constituyentes del modelo [40]. El conjunto de observaciones que constituyen el modelo se denomina *conjunto de entrenamiento* y la palabra supervisado proviene de la existencia de un supervisor (muchas veces humano) que es quien asigna las etiquetas a cada observación en el conjunto de entrenamiento.

En otras palabras, dado un conjunto de entrenamiento conformado por  $m$  pares observaciones  $(x_j, y_j)$ ,  $j \in \{1, \dots, m\}$  (donde  $x$  representa las características e  $y$  representa las etiquetas), se busca encontrar una función  $f(\cdot)$  capaz de inferir  $y$  a partir de las características representadas en  $x$ . Esto se realiza de acuerdo a un criterio determinado de rendimiento, que permite cuantificar cuan bueno es el modelo. Cuando las etiquetas pertenecen a un conjunto de discreto no ordenado, digamos  $S = \{c_1, \dots, c_m\}$  (i.e. conformado por  $m$  clases), se trata de un problema de clasificación. En cambio cuando el conjunto de las etiquetas toma valores en  $\mathbb{R}$  (o más generalmente en un espacio vectorial  $\mathbb{E}$ ), se habla de un problema de regresión.

### 2.2.1. Conjuntos de entrenamiento, validación y prueba

Como fue mencionado en la introducción de esta sección en un problema de AM supervisado, se tiene un conjunto de pares  $\{(x_i, y_i)\}_{i=1}^N$ . Una práctica usual para abordar este tipo de problema es subdividir el conjunto de observaciones en tres subconjuntos: entrenamiento, validación y prueba.

Un algoritmo de AM permite crear un modelo matemático que se ajusta a un conjunto de observaciones. De acuerdo a lo descrito previamente, dicho conjunto se llama conjunto de entrenamiento, ya que el modelo se aprende en o se adapta a él.

Usualmente, existe también una serie de parámetros de alto nivel, y generalmente fijos antes de que se inicie el proceso de entrenamiento, que pueden expresar por ejemplo la complejidad del modelo o la velocidad con la que el modelo se ajusta a los datos. Estos son denominados hiperparámetros y no pueden ser aprendidos mediante el proceso de entrenamiento. Esta es la razón de la existencia de un conjunto de validación, que es utilizado para aprender los hiperparámetros. El procedimiento consiste en entrenar varios modelos para

diferentes hiperparámetros y luego seleccionar el modelo que tiene mejor desempeño en el conjunto de validación.

En cuanto al conjunto de prueba, este se utiliza para medir la capacidad de generalización del modelo entrenado, es decir, la capacidad de predecir correctamente la etiqueta  $y_i$  para un  $x_i$  que no fue observado ni en el conjunto de validación, ni en el conjunto de entrenamiento.

## 2.2.2. Algoritmos de clasificación

Los dos problemas abordados en la memoria son problemas de clasificación, por lo que a continuación se detallan los diferentes algoritmos que son utilizados: regresión logística, máquinas de soporte vectorial, *Gradient Tree Boosting* y redes neuronales. Se profundiza especialmente en el modelo de redes neuronales de convolución llamado *Inception-V3* [43] que es utilizado en el problema CEUS.

### Regresión logística

Es un modelo estándar para abordar problemas de clasificación. En este contexto la regresión logística (RL) se utiliza como un punto de referencia, ya que es el menos flexible y el más sencillo de ajustar, de los métodos empleados. Resulta más sencillo describir la RL para un problema de clasificación de dos clases (binario) en el que las clases (o etiquetas) se representan por 0 o 1, es decir  $y_i \in \{0, 1\}$ . Bajo esta configuración, el modelo de RL determina la clase a la que pertenece una observación mediante el cómputo de la probabilidad de pertenencia a la clase 1, a través de:

$$f(x) = \mathbb{I}(p(x) > \frac{1}{2}), \quad p(x) = \frac{\exp^{\langle \vec{x}_i, \vec{\beta} \rangle + \beta_0}}{1 + \exp^{\langle \vec{x}_i, \vec{\beta} \rangle + \beta_0}}$$

donde  $f(x)$  es la función que determina la clase a la que pertenece  $x$  (función de decisión) y  $p(x)$ , la probabilidad mencionada.

A modo de ejemplo, en la Figura 2.2 se muestra un modelo de RL para un problema en el que  $x \in \mathbb{R}$  es el balance de cuenta bancaria e  $y \in \{0, 1\}$  representa si el cliente pagó sus deudas a tiempo (0) o no (1). La función en azul es la probabilidad de no pago en función del balance bancario obtenido mediante una regresión logística.

Este modelo tiene por finalidad encontrar un hiperplano (en el espacio de características) separador de las clases positiva y negativa, que maximice algún criterio de rendimiento. Es común que para ajustar este tipo de modelo se maximice la función de log-verosimilitud que se expresa:

$$\begin{aligned} L(\vec{\beta}, \beta_0) &= \sum_{i=1}^N (y_i \log(p(x_i; \vec{\beta}, \beta_0)) + (1 - y_i) \log(1 - p(x_i; \vec{\beta}, \beta_0))) \\ &= \sum_{i=1}^N y_i (\langle \vec{x}_i, \vec{\beta} \rangle + \beta_0) - \log(1 + \exp^{\langle \vec{x}_i, \vec{\beta} \rangle + \beta_0}) \end{aligned} \quad (2.1)$$

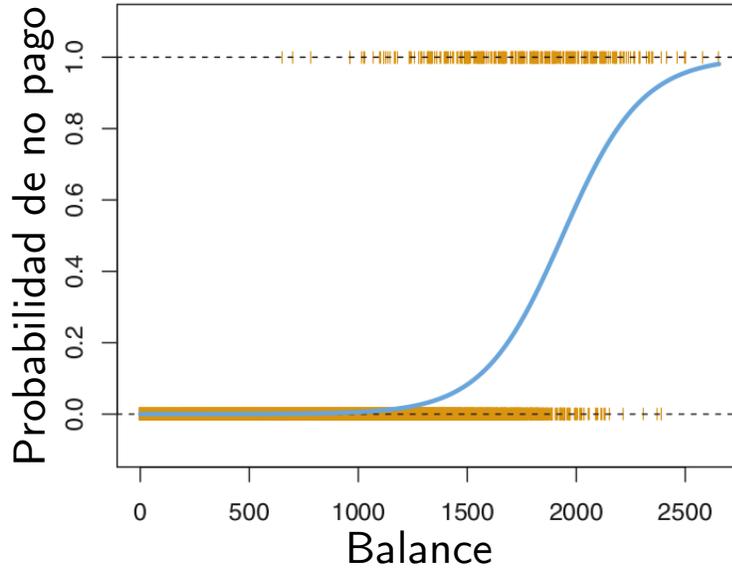


Figura 2.2: Regresión logística para probabilidad de no pago en función del balance de la cuenta bancaria de los clientes.

Adaptado de [26]

Las derivadas de primer y segundo orden de 2.1 tienen forma explícita y son utilizadas por diferentes métodos de optimización para encontrar el máximo y así los parámetros  $\vec{\beta}$  y  $\beta_0$  óptimos.

Para abordar un problema de clasificación multiclase de  $K$  clases mediante RL se puede utilizar la función de entropía cruzada  $L = -\sum_{i=1}^N \sum_{j=1}^K \log(p_{\mathbb{I}(y_i=C_j)}(x_i))$ , que también posee derivadas de primer y segundo orden explícitas. Esta misma noción de rendimiento es también utilizada para el ajuste de redes neuronales, ya que como se describe en la sección dedicada a estas, la RL puede ser vista como la unidad base de un modelo de red neuronal.

Otra posible estrategia para extender este problema a una regresión multiclase, es la de crear varios clasificadores binarios. Los métodos más comunes para esto son conocidos como *one versus one* (OVO) y *one versus rest* (OVR). Para un problema de  $K$  clases, el método OVO construye  $K$  clasificadores  $\{f_k\}_{k=1}^K$  en donde la clase positiva corresponde a la clase  $k$  y la negativa a todas las demás. Así, la función de decisión asigna:

$$f(x) = \arg \max_k f_k(x)$$

Por otro lado, el método OVO construye  $K(K-1)$  clasificadores  $\{f_{i,j}\}_{i,j=1}^K$  con  $i \neq j$ , en donde se toma como clase positiva a la clase  $i$  y negativa a la clase  $j$ , produciendo así una función de decisión de la forma:

$$f(x) = \arg \max_k \sum_{j=1, j \neq k}^K f_{k,j}(x)$$

## Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial (MSV) son un algoritmo de AM supervisado que puede ser utilizado tanto en regresión como en clasificación. Cuando se trata de un problema de clasificación, básicamente consiste en separar un conjunto de datos que pertenecen a diferentes clases mediante la frontera de máximo margen.

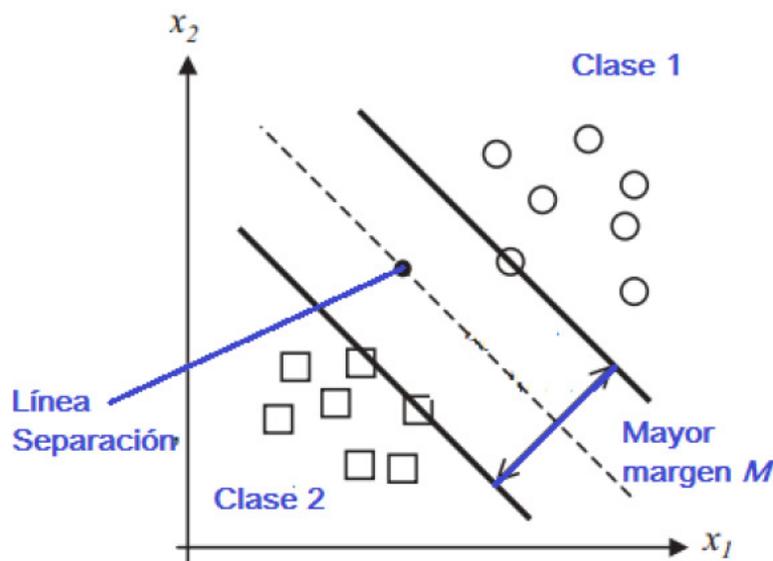


Figura 2.3: Ejemplo de separación de dos clases a través de un plano lineal.

Fuente: Adaptado de [44]

Para un problema de dos clases, como el de la Figura 2.3 que es linealmente separable, la frontera de máximo margen es aquella línea punteada que presenta una distancia mayor a los elementos más cercanos de cada clase, los que son denominados vectores soporte y corresponden a las observaciones que tocan ambas líneas en negrita [26]. Dicho problema se puede formular como:

$$\max_{\vec{\beta}, \beta_0, \|\vec{\beta}\|=1} M \quad \text{s.a.} \quad y_i(\langle \vec{x}_i, \vec{\beta} \rangle + \beta_0) \geq M \quad (2.2)$$

donde  $M$  representa el margen entre la frontera y los vectores más cercanos a esta (vectores soporte),  $y_i \in \{-1, 1\}$ ,  $x_i \in \mathbb{R}^d$ ,  $\forall i \in \{1, N\}$ , con  $N$  el número de observaciones. Por su parte  $\beta$  corresponde al vector unitario, normal al hiperplano separador.

Mediante el cambio de variable  $M = \frac{1}{\|\vec{\beta}\|}$ , un poco de álgebra y el método de los multiplicadores de Lagrange, el problema (2.2) tiene solución de la forma:  $\hat{\vec{\beta}} = \sum_{i=1}^N \hat{\alpha}_i y_i \vec{x}_i$ ,  $\hat{\alpha}_i \geq 0 \forall i \in \{1, \dots, N\}$ , es decir,  $\hat{\vec{\beta}}$  es una combinación lineal de las observaciones. En consecuencia, el problema 2.2 se reduce a encontrar los  $\hat{\alpha}_i$  óptimos. Son aquellos  $\vec{x}_i$  que tienen asociado un  $\hat{\alpha}_i > 0$  los denominados los vectores soporte.

En consecuencia, la función de decisión queda entonces definida mediante:

$$f(\vec{x}) = \text{signo}(\langle \vec{x}, \hat{\vec{\beta}} \rangle + \beta_0) \quad (2.3)$$

Si las clases son separables, pero no linealmente, las MSV se pueden extender mediante una función que expresa una noción de similitud. Dicha función denominada *kernel* es la que permite construir fronteras no lineales entre las clases. El *kernel* toma el lugar del producto interno ( $\langle \cdot, \cdot \rangle$ ) en las ecuaciones 2.2 y 2.3. El *kernel* debe ser una función de la forma  $K(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  que es simétrica<sup>1</sup> y semidefinida positiva<sup>2</sup> para ser válida. A modo de ejemplo, en la Tabla 2.2 se muestran algunos de los *kernels* más utilizados en la literatura.

Función kernel	Tipo de clasificador
$K(x_i, x_j) = (x_i^T x_j)$	Linear, producto punto
$K(x_i, x_j) = [(x_i^T x_j) + 1]^d$	Polinomio de grado d
$K(x_i, x_j) = \exp(-\frac{\ x_i - x_j\ ^2}{2\sigma^2})$	RBF Gausseano
$K(x_i, x_j) = \tanh(\beta x_i^T x_j + b)$	Perceptrón de multicapa
$K(x_i, x_j) = \frac{1}{\sqrt{\ x_i - x_j\ ^2 + \beta}}$	Función multicuadrática inversa

Tabla 2.2: Ejemplos de *kernels* para MSV.  
Fuente: Adaptado de [44]

Si las clases están solapadas este modelo se extiende utilizando variables de holgura  $\{\chi_i\}_{i=1}^N$  que permitan a algunas de las observaciones estar del lado equivocado de la frontera. Es decir, la restricción en (2.2) se modifica a:

$$y_i(\langle \vec{x}_i, \vec{\beta} \rangle + \beta_0) \geq M(1 - \chi_i), \quad \chi_i \geq 0, \quad \forall i \in \{1, \dots, N\}$$

$$\sum_{i=1}^N \chi_i \leq L \quad (2.4)$$

Permitiendo así, mediante el cambio de variable  $M = \frac{1}{\|\vec{\beta}\|}$  y la inclusión de la restricción 2.4 en la función objetivo, la reformulación problema 2.2 en:

$$\text{mín } \|\vec{\beta}\| + C \sum_{i=1}^N \chi_i \quad \text{s.a. } \chi_i \geq 0, \quad y_i(\langle \vec{x}_i, \vec{\beta} \rangle + \beta_0) \geq 1 - \chi_i \forall i \in \{1, \dots, N\} \quad (2.5)$$

donde C representa la noción de penalización para cada observación que se ubica del lado equivocado de la frontera. Dicho hiperparámetro puede ser encontrado mediante un conjunto de validación (cf. Sección 2.2.1).

Las estrategias más comunes para extender un modelo de MSV a una regresión multiclase ya fueron abordadas en la descripción de RL, estas consisten en crear varios clasificadores según los procedimientos OVO y OVR.

<sup>1</sup> $G(\cdot, \cdot) : E \times E \rightarrow \mathbb{R}$  se dice simétrica si  $\forall x, y \in E, G(x, y) = G(y, x)$ .

<sup>2</sup> $G(\cdot, \cdot) : E \times E \rightarrow \mathbb{R}$  se dice semidefinida positiva si  $\forall \{x_i\}_{i=1}^n \subset E$ , la matriz  $\mathbf{G} = (G(x_i, x_j))_{i,j}$  es semidefinida positiva, esto es,  $\lambda_i \geq 0$  con  $\lambda_i$  valor propio de la matriz  $\mathbf{G}, \forall i \in \{1, \dots, n\}$ .

## Gradient Tree Boosting

Para entender el algoritmo *Gradient Tree Boosting* (GTB) para clasificación es necesario abordar dos conceptos preliminares. El primero es el de árbol de decisión para el problema de regresión. Este puede ser representado mediante un diagrama como el de la Figura 2.4. Dicho diagrama, representa un problema en el cual se trata de predecir el logaritmo del salario de un conjunto de jugadores de béisbol según la cantidad de años que llevan jugando en la primera división y según el número de bateos que realizaron durante el año anterior. Dicho árbol tiene dos nodos y tres hojas. Cada nodo está definido por un criterio que separa los datos: hacia la izquierda se dirigen las observaciones que cumplen con el criterio y a la derecha las que no. La función de decisión  $f(x)$  en cada una de las hojas está dada por el promedio de las etiquetas de las observaciones que corresponden a dicha hoja [19].

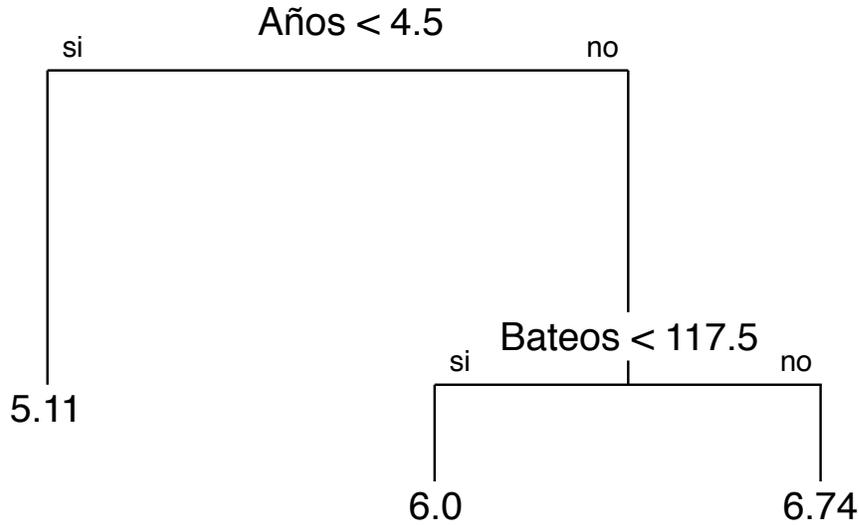


Figura 2.4: Diagrama de un árbol de regresión para la predicción del logaritmo del salario de un jugador de béisbol.

Adaptado de [19]

Así, el árbol de decisión divide el espacio de características en  $J$  regiones, donde  $J$  es el número de hojas y, para el problema de regresión, busca reducir el criterio de rendimiento denominado error cuadrático medio (ECM), correspondiente a  $\sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$ . Más aún, el largo del segmento que separa el nodo padre de sus hijos es proporcional a la reducción del ECM inducida por el nodo padre. En consecuencia, es posible inferir que de acuerdo al árbol de la Figura 2.4, la cantidad de años en primera división es una característica mucho más importante que la cantidad de bateos en el año anterior, para inferir el salario de un jugador.

El segundo concepto por abordar es el de *boosting* de árboles, el cual está basado en la idea de que a partir de varios clasificadores débiles es posible construir un clasificador robusto. Así, un *boosting* de árboles es un modelo de la forma:

$$f_m(x) = \sum_{i=1}^m \alpha_i T(x, \Theta_i)$$

y por esto, forma parte de la familia de algoritmos llamados *stagewise additive modeling*. Esto significa que es construido de forma iterativa mediante un procedimiento en que en base al modelo con  $m - 1$  clasificadores débiles se eligen  $\alpha_m$  y  $\Theta_m$  según criterios de optimización específicos.

El GTB para un problema de clasificación de  $K$  clases,  $\{C_1, \dots, C_k\}$ , es una función que entrega el vector de probabilidad de pertenencia a cada una de las clases:

$$\vec{f}(\vec{x}) = (p_k(\vec{x}))_{k=1}^K$$

Esta es la razón por la que el concepto de árbol de decisión para el problema de regresión es introducido, ya que si bien se trata de un modelo para clasificación, para ajustar el GTB es necesario ajustar un modelo de regresión que toma valores en  $[0, 1]^K$ .

La función de costo que permite ajustar el modelo puede ser diferente. Se utiliza nuevamente el ejemplo de la entropía cruzada:

$$J(y, \vec{p}(x)) = - \sum_{j=1}^k \log(p_{\mathbb{I}(y=C_j)}(x))$$

El modelo se construye mediante el ajuste de múltiples árboles de regresión que aproximan al inverso aditivo gradiente de la función de costo evaluada en todos los puntos de entrenamiento, esto es  $-\nabla_{f(x)} J(f(x)) \Big|_{f(x)}$ . Dichos árboles tienen restricciones de tamaño fijas, por ejemplo  $J_m$  hojas, lo que permite evitar el sobreajuste. Una vez el árbol  $T(\Theta_m)$  está ajustado, se determina:

$$\alpha_{j_m} = \arg \min_{\alpha} \sum_{x_i \in R_{j_m}} J(y_i, f_{m-1}(x_i) + \alpha)$$

para cada una de las regiones  $J_m$  de dicho árbol y se actualiza el modelo  $f_m(x) = f_{m-1}(x) + \sum_{j_m=1}^{J_m} \alpha_{j_m} \mathbb{I}(x \in R_{j_m})$

La idea subyacente de aproximar  $-\nabla_{f(x)} J(f(x)) \Big|_{f(x)}$  se fundamenta en el hecho de que es la dirección de descenso más pronunciada. Así, en cada una de las iteraciones del modelo, se mejora el rendimiento del modelo anterior, en términos de la función de costo  $J$ [19].

## Redes neuronales

Las redes neuronales (RN) han demostrado ser la técnica por excelencia en visión computacional y específicamente, las redes neuronales de convolución (RNC). El trabajo [28] es conocido como uno de los primeros que reunió un conjunto de resultados de vanguardia logrados por este tipo de algoritmos de clasificación. Las redes neuronales pueden utilizarse además para otras tareas, como regresión o reducción de dimensionalidad, pero en esta sección nos centraremos en su uso para clasificación.

Una red neuronal profunda para el problema de clasificación, está compuesta por una capa de entrada con tantas neuronas como características poseen las observaciones y una capa de

salida con tantas neuronas como clases tenga el problema, que entrega la probabilidad de pertenencia a cada una de dichas clases. Adicionalmente, contiene capas ocultas que son las que comunican dichas capas de entrada y de salida. La cantidad de capas ocultas y el tamaño de estas (número de neuronas), permite controlar el nivel de complejidad o flexibilidad de la red. En la Figura 2.5 se muestra un modelo de RN totalmente conectada con una sola capa oculta de tamaño 4 (con 4 neuronas).

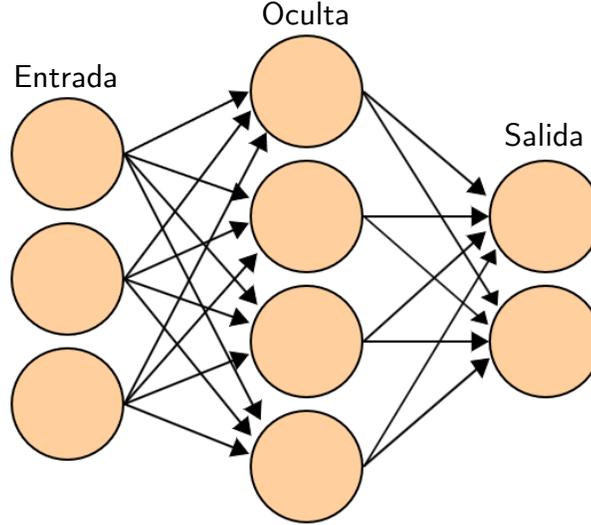


Figura 2.5: Estructura básica de la red neuronal totalmente conectada, con una capa oculta. Fuente: Adaptado de [32]

La estructura básica de la red es la neurona. La neurona  $i$  de la capa  $l$ ,  $n_i^l$ , está definida por un vector de pesos  $\vec{w}_i^l$  y una función de activación  $f^l$  (compartida por todas las neuronas de la capa  $l$ ). Denotando por  $\vec{Z}_i^{l-1}$  el vector de entradas provenientes de neuronas de la capa  $l-1$  que están conectadas con la neurona  $N_i^l$ , la salida de dicha neurona se calcula de la forma:

$$z_i^l = f^l(\vec{a}_i^l) = f^l(\langle \vec{Z}_i^{l-1}, \vec{w}_i^l \rangle + w_{i,0}^l) \quad (2.6)$$

Cabe notar que la expresión anterior tiene la misma forma a la Ecuación (2.1) del modelo RL, salvo por la función de activación  $f^l$  que para el caso de la RL es la sigmoide  $f(x) = \frac{\exp^x}{1+\exp^x}$ . Así el modelo de red neuronal puede ser vista como una composición de múltiples RL donde la sigmoide puede ser reemplazada por otras funciones diferenciables que no necesariamente están restringidas a tomar valores en  $[0, 1]$ . Sin embargo, de acuerdo a lo descrito anteriormente, la capa de salida debe tomar valores en  $[0, 1]^K$ , más aún debe entregar una distribución de probabilidad. Para ello, generalmente se utiliza una capa *softmax* o función exponencial normalizada, esto es:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \forall j \in \{1, \dots, N^{h_L}\} \quad (2.7)$$

donde  $N^{h_L}$  es la cantidad de neuronas de la última capa oculta. Así, el modelo es capaz de entregar la probabilidad de pertenencia a cada clase. Esto último permite que exista otra

similitud entre la RL, GTB y RN, que es el uso de la entropía cruzada como funcional de costo para la clasificación multiclase.

Se debe destacar que la diferenciabilidad de la función de activación es de suma importancia, ya que el entrenamiento de una RN se hace mediante el algoritmo de retropropagación del error, el cual necesita la diferenciabilidad de las funciones de activación. Este algoritmo consiste en utilizar el método de descenso de gradiente del funcional de costo para ajustar los pesos de la red, es decir, para una tasa de aprendizaje en la iteración  $r$ ,  $\gamma_r$  y denotando  $\vec{\alpha}_m \doteq \vec{w}_m^{l-1}$  y  $\vec{\beta}_k \doteq \vec{w}_k^l$ , los vectores de peso correspondientes a neuronas  $n_m^{l-1}$ ,  $n_k^l$  conectadas de capas consecutivas, se tiene:

$$\begin{aligned}\beta_{k_m}^{(r+1)} &= \beta_{k_m}^{(r)} - \gamma_r \sum_{i=1}^N \frac{\partial J_i}{\partial \beta_{k_m}^{(r)}}, \\ \alpha_{m_p}^{(r+1)} &= \alpha_{m_p}^{(r)} - \gamma_r \sum_{i=1}^N \frac{\partial J_i}{\partial \alpha_{m_p}^{(r)}}\end{aligned}$$

Mediante la regla de la cadena es posible encontrar la recurrencia dada por las ecuaciones de retropropagación:

$$\frac{\partial J_i}{\partial \alpha_{m_p}^{(r)}} = z_{p_i}^{l-1} f'_{l-1}(\langle \alpha_m, z_{\cdot i}^{l-1} \rangle) \sum_{k=1}^{N^{h_l}} \beta_{k_m} \delta_{k_i}^{(r)}$$

donde  $\delta_{k_i}^{(r)} = \frac{1}{z_{k_i}} \frac{\partial J_i}{\partial \beta_{k_m}^{(r)}}$ . De esta forma, como la derivada  $\frac{\partial J_i}{\partial \beta_{k_m}^{(r)}}$  puede calcularse en forma explícita para la capa de salida, es posible expresar todas las derivadas parciales de la red en función de los errores en la salida, las entradas y las salidas de cada una de las capas, los pesos en la iteración  $r$  y las derivadas de las funciones de activación[19].

**Redes neuronales de convolución (RNC).** Como fue mencionado antes, una familia de RN especialmente útil en el campo de la visión computacional es el modelo de RNC. Por su parte las redes neuronales de convolución presentan la misma estructura básica de capas de entrada, de salida y ocultas, pero poseen una arquitectura particular en cuanto a las conexiones entre las capas. Tal como se muestra en la Figura 2.6, las neuronas de las capas ocultas de las redes de convolución están conectadas sólo a una vecindad de la capa precedente. Además, las letras  $A$  y  $B$  que se muestra en cada una de las neuronas de las dos primeras capas ocultas, representan el hecho de que los pesos son compartidos por cada una de esas neuronas de la capa. Dichos pesos se conocen como filtros de convolución. Esto último reduce drásticamente la cantidad de parámetros en comparación a una RN totalmente conectada con el mismo número de neuronas.

La unidad básica de las RNC es el filtro de convolución. En la Figura 2.7 se ilustra el procedimiento de convolución de matrices cuadradas con filtros cuadrados. En ella se simboliza en color amarillo un filtro de convolución de  $3 \times 3$ , que es aplicado sobre la imagen (matriz blanca de la izquierda). El procedimiento consiste en multiplicar punto a punto el filtro de convolución con las submatrices de tamaño idéntico al filtro y sumar todos los

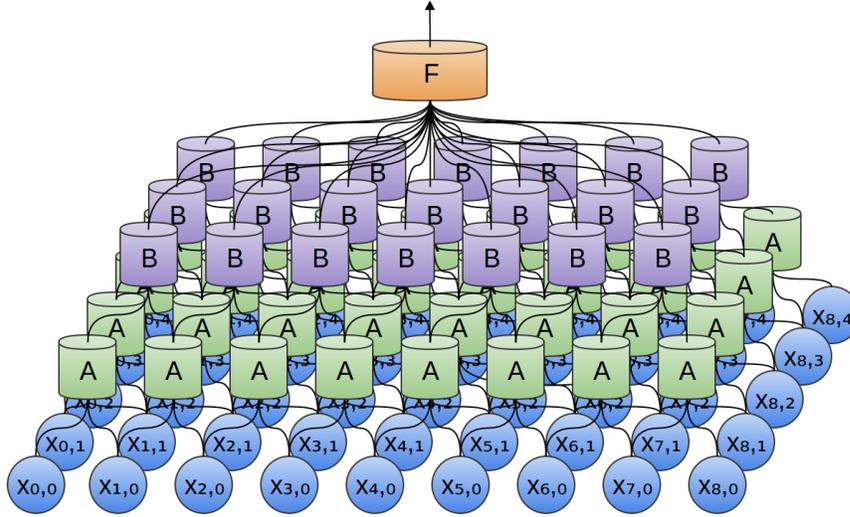


Figura 2.6: Red convolucional de 2 dimensiones.  
Fuente: extraído de [32]

componentes de dicha matriz resultante. El orden en la matriz resultante, se ilustra con los colores rojo y verde de la Figura 2.7. Esto se generaliza fácilmente a imágenes de 3 (RGB) o más (imágenes multispectrales) canales, ya que el filtro de convolución siempre tiene el mismo número de canales (bandas) que el tensor que es convolucionado. Vale decir, los filtros de convolución para un tensor de tamaño  $n \times n \times c$ , son siempre de la forma  $f \times f \times c$ .

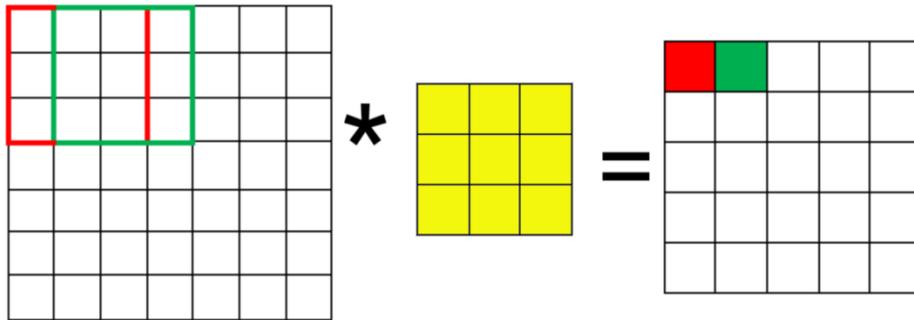


Figura 2.7: Operación de convolución.

Además del tamaño del filtro de convolución existen dos parámetros que también caracterizan la convolución. Se trata del *stride* y el *padding*. El *stride* determina la distancia mínima que hay entre los centros de las submatrices convolucionadas de la imagen. Por su parte, el *padding* determina la cantidad de elementos que serán agregados a cada lado de la matriz. En la Figura 2.8 se ilustra cómo influyen el tamaño y en el cómputo de la matriz de resultante. Cuando el *stride* es 1 y el *padding* permite que el tamaño de la entrada sea igual al de la salida, se denomina convolución *same*(S). Por el contrario, cuando no hay *padding*, se habla de una convolución válida (V). Así, para una operación de convolución de tamaño  $f \times f$ , un *stride*  $s$  y un *padding*  $p$ , que es aplicada a una matriz de entrada de  $n \times n$  tendrá tamaño  $m \times m$  donde:

$$m = \frac{n + 2p - f}{s} + 1 \quad (2.8)$$

Cabe mencionar que cuando se opera con filtros no cuadrados, la misma ecuación es respetada para los ejes de ancho y alto por separado.

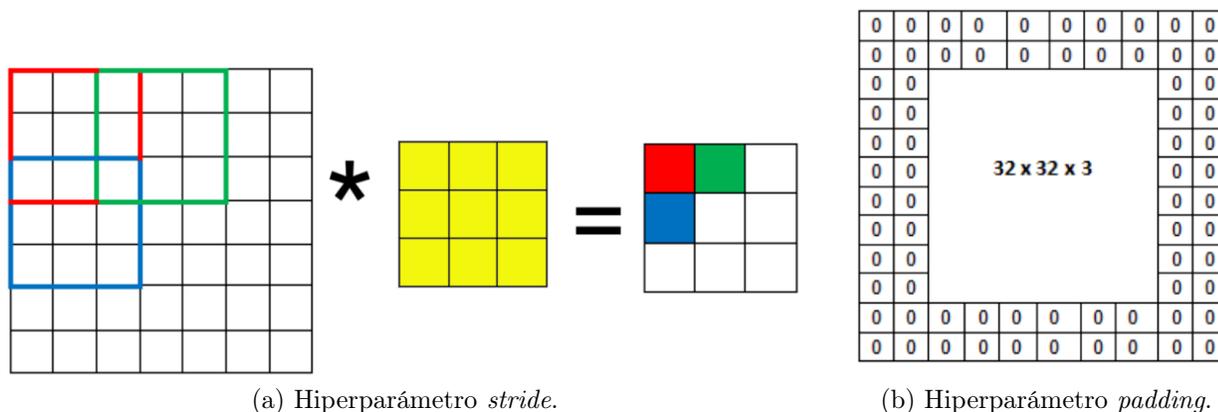


Figura 2.8: Otros parámetros que determinan la operación de convolución.

Adicionalmente existe otra operación básica utilizada en las redes de convolución. Se trata del *pooling*. Los tipos de *pooling* más conocidos son el del máximo (*max-pooling*) y el de la media (*average-pooling*). Respectivamente, estos computan el máximo y la media de la región donde se aplican. Para ilustrar el funcionamiento de esta operación se adjunta la Figura 2.9. Es importante recalcar que a diferencia de la operación de convolución, donde este se aplica a lo largo de todos los canales del tensor de entrada, el *pooling* se aplica a nivel de cada canal. Así, si un tensor de entrada a un operador de *pooling* posee  $c$  canales, la salida seguirá teniendo  $c$  canales.

Este tipo de estructura es adecuada para la clasificación de imágenes, ya que permite extraer características de alto nivel que se deducen de conjuntos de píxeles aledaños. Además, el carácter compartido de los parámetros de las capas de convolución, permiten hacer la detección de características invariante a traslación.

Como se muestra en la sección siguiente, en general, en los modelos de RNC, cada capa

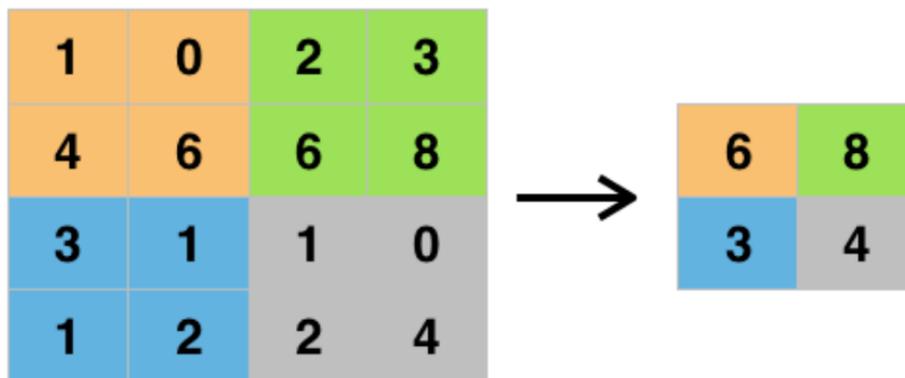


Figura 2.9: Ilustración del procedimiento del operador *max-pooling* de tamaño  $2 \times 2$  y *stride* 2.

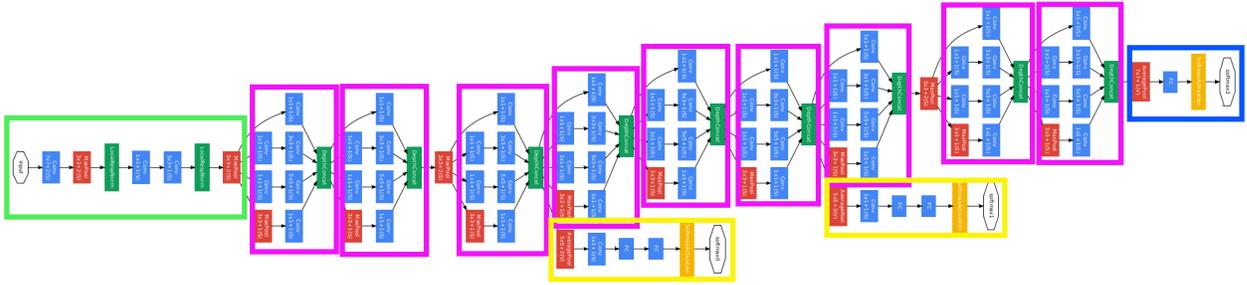


Figura 2.10: Arquitectura del modelo *Inception-V1*.  
Adaptado de [42]

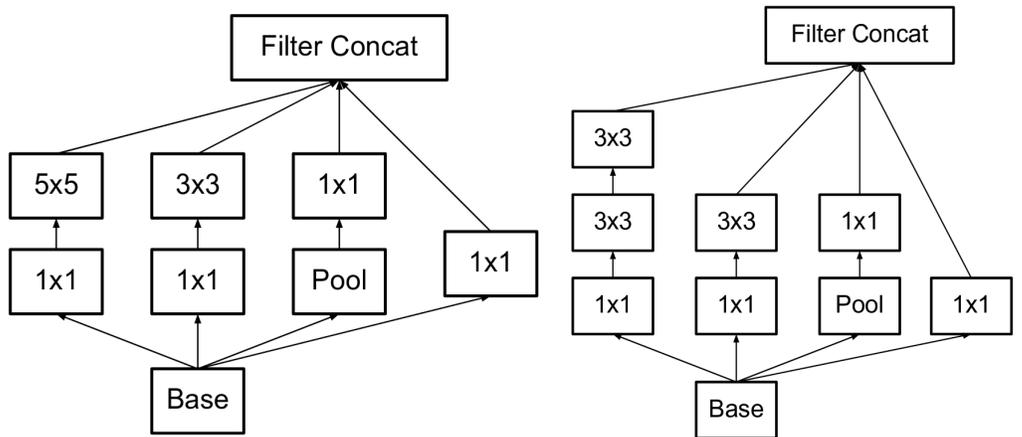
convolucional está compuesta por varios filtros de convolución, cuya salida es el tensor de las matrices resultantes al aplicar cada uno de esos filtros, concatenadas en el eje de los canales. Así, la salida de una capa convolucional de  $F$  filtros de convolución que recibe una entrada de  $c$  canales, entrega en la salida un tensor de tamaño  $m \times m \times F$ , con  $m$  determinado por la Ecuación (2.8), es decir, el tamaño del filtro, el *stride* y el *padding*.

## Modelos *Inception*

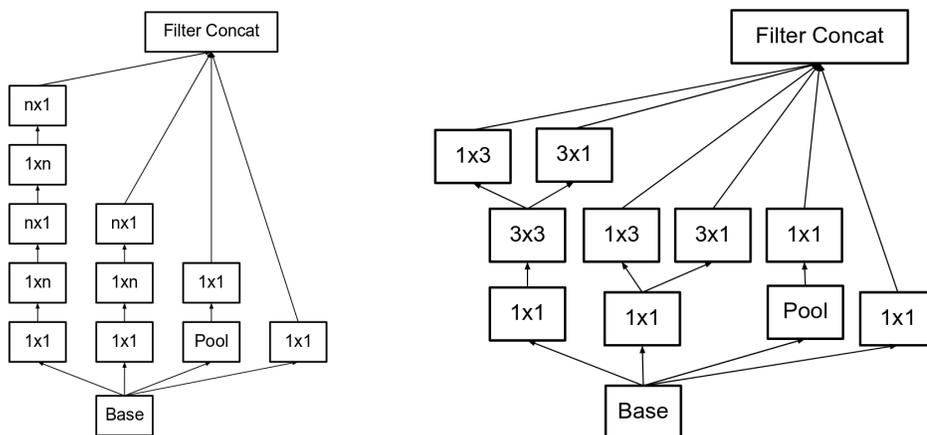
El modelo *Inception-V1* u originalmente GoogLeNet [42], fue el modelo ganador del desafío ILSVRC 2014, más conocido como ImageNet. Dicho desafío consistía en la clasificación de 1,2 millones de imágenes y 22 mil categorías. El modelo Inception V1 logró el error de prueba *top 5* más bajo, 6.7% []

Para explicar la arquitectura del modelo *Inception-V3* se utilizará como base la arquitectura de la primera versión de la red (*Inception-V1*). Esta última se puede observar en la Figura 2.10 y está compuesta por 4 bloques constituyentes que se describen separada y comparadamente en los casos que corresponda.

Destacados por rectángulos color violeta, se observan 9 bloques *Inception*, que son los bloques más relevantes y característicos del modelo. En la imagen 2.11 se pueden observar el bloque *Inception* original y tres de las versiones de este, que están presentes tanto en *Inception-V2* como en *Inception-V3*. La característica distintiva e innovadora introducida por el modelo *Inception-V1*, consiste en combinar características extraídas mediante filtros de diferentes tamaños. Además, esta combinación de diferentes tamaños de filtro se hace de manera computacionalmente eficiente mediante las convoluciones de  $1 \times 1$  que preceden a los filtros de tamaño mayor y que siguen a los *max-pooling* de  $3 \times 3$  [42]. Centrado también en la eficiencia computacional, se diseñan las 3 versiones del bloque *Inception* en donde a través de filtros más pequeños y asimétricos (en consecuencia con menos parámetros), se puede lograr la misma, o incluso mejor expresividad que en el modelo original. Esta técnica se denomina factorización de convoluciones. Se diseñan 3 modelos del bloque *Inception* diferentes, ya que se comprobó empíricamente que en la medida que el tamaño de la matriz de características se iba reduciendo, se hacía más conveniente la utilización de convoluciones de mayor tamaño [43]. De acuerdo a esto, el modelo Inception V2 queda definido como se muestra en la Tabla 2.3.



(a) Bloque *Inception* original (*Inception-V1*). (b) Bloque *Inception* con factorización - tipo 1 (*Inception-V2* y *V3*).



(c) Bloque *Inception* con factorización - tipo 2 (*Inception-V2* y *V3*).

(d) Bloque *Inception* con factorización - tipo 3 (*Inception-V2* y *V3*).

Figura 2.11: Diferentes versiones del bloque *Inception*.  
Extraídos de [42, 43]

Otra parte interesante de los modelos *Inception* son los llamados bloques auxiliares que corresponden a las secciones rodeadas de rectángulos amarillos en la Figura 2.10. Ellas fueron concebidas para hacer frente a la problemática del desvanecimiento de gradiente. Este problema aparece usualmente en redes en las que dada la profundidad de estas, la retropropagación del error tiende a cero para algunas capas de la red.

Durante el entrenamiento, el error que se obtiene en dichas capas (cf. Figura 2.12), es multiplicado por  $1/3$  con el fin de privilegiar la clasificación del tronco principal del modelo. En lo que respecta a la fase de predicción, dichos bloques auxiliares son descartados.

Si bien el objetivo inicial de la introducción de aquellos bloques auxiliares era la de aumentar el gradiente en las capas medias de la red, en [43] se probó que eliminando el primero de aquellos bloques auxiliares, los resultados eran idénticos durante las primeras épocas de ajuste de parámetros. Así, se postula que en realidad dichos bloques auxiliares serían otra

Tipo	Tamaño del filtro / stride	Tamaño de la entrada
conv	3×3 / 2	299×299×3
conv	3×1 / 1	149×149×32
conv padded	3×3 / 1	147×147×32
pool	3×3 / 2	147×147×64
conv	3×3 / 1	73×73×64
conv	3×3 / 2	71×71×80
conv	3×3 / 1	35×35×192
3×Inception	Como en la Figura 2.11b	35×35×288
5×Inception	Como en la Figura 2.11c	17×17×768
2×Inception	Como en la Figura 2.11d	8×8×1280
pool	8×8	8×8×2048
linear	logits	1×1×2048
softmax	clasificador	1×1×1000

Tabla 2.3: Modelo *Inception-V2* con estructura adaptada para inferencia en el desafío ILSVRC 2012 [28]

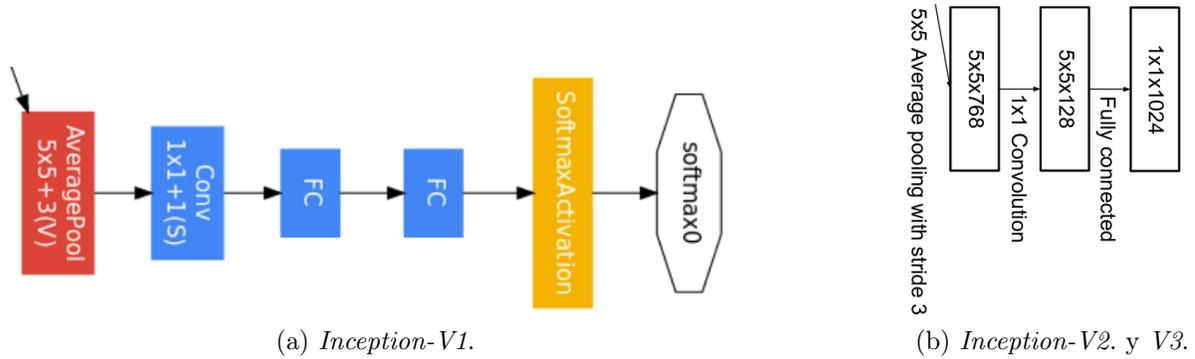


Figura 2.12: Estructura del bloque auxiliar para los diferentes modelos *Inception*.  
Extraído de [42, 43]

forma de regularización. En coherencia con este hallazgo, en el modelo *Inception-V2*, existe un único bloque auxiliar que se ubica en la salida del último de los bloques *Inception* tipo 2 (cf. Figura 2.11c).

Otra de las características que diferencia a los modelos *Inception*, es el uso de una técnica llamada normalización por *batches* [25]. Como fue descrito en la sección dedicada a RN, estas se entrenan mediante descenso de gradiente. Un *batch* es simplemente un conjunto de observaciones que corresponden a una iteración del descenso de gradiente, es decir, el gradiente se computa para el costo acumulado de los elementos del *batch*. Así, dado un *batch*  $\mathcal{B} = \{x_i\}_{i=1}^m$ , si se aplica normalización por *batches* a la activación  $A$ , computando la media del *batch*  $\mu_{\mathcal{B}} = \frac{1}{m} \sum_{i=1}^m A_i$  y la varianza del *batch*  $\sigma_{\mathcal{B}}^2 = \frac{1}{m} \sum_{i=1}^m (A_i - \mu_{\mathcal{B}})^2$ , se tiene:

$$\text{BN}_{\gamma, \beta}(A_i) = \gamma \hat{A}_i + \beta \quad , \quad \text{con } \hat{A}_i = \frac{A_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2}}$$

Cabe notar que la función BN es diferenciable y así los parámetros pueden ser aprendidos mediante descenso de gradiente. La idea subyacente detrás de esta transformación de la acti-

vacación es hacer a la capa que recibe la activación normalizada por *batches*, más independiente de las variaciones en las capas que lo preceden. Dicho método ha mostrado ser efectivo para acelerar los procesos de entrenamiento y posee además otras cualidades [25].

Si bien la normalización por *batches* se utiliza en *Inception-V2* e *Inception-V3*, en la primera, sólo se usa para las capas de convolución, y en la segunda, además de aquello, también se emplea en el bloque auxiliar.

La sección rodeada por un rectángulo de color verde en la Figura 2.10 corresponde a las primeras capas de la red. Dicha sección se conoce como el bloque raíz del modelo *Inception*. En la Figura 2.13 se muestra en detalle este componente para el modelo *Inception-V1*. La principal diferencia con los otros dos modelos radica en el reemplazo de la capa de normalización de respuesta local [28], por la normalización por *batches*. Adicionalmente, *Inception-V3* reemplaza la primera convolución de  $7 \times 7$  por una factorización de tres convoluciones de  $3 \times 3$  con *stride* de 1.

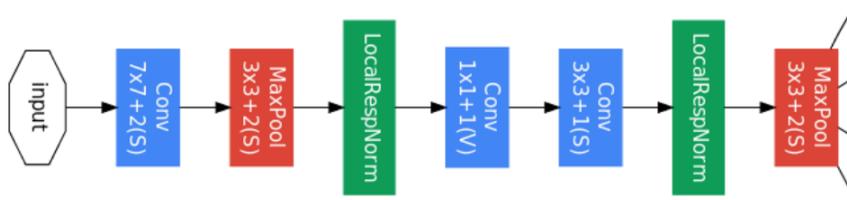


Figura 2.13: Bloque raíz del modelo *Inception-V1*.  
Extraído de [42]

Finalmente, la sección destacada por el rectángulo de color azul es lo que constituye el clasificador de salida. Como es mencionado en la sección de redes neuronales, la función que permite entregar las probabilidades de pertenencia a cada clase es generalmente la función *softmax* (Ecuación (2.7)). Para el caso del problema CEUS en donde se utilizará el modelo *Inception-V3* como parte de la solución propuesta, esta región es eliminada, ya que siempre debe adaptarse al problema que se está resolviendo. En el caso de la Figura 2.14, el modelo está adaptado para resolver el problema de ILSVRC 2012, por lo que posee mil neuronas de salida.

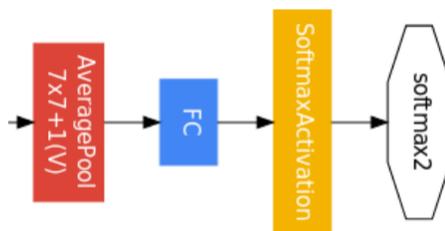


Figura 2.14: Clasificador de salida del modelo *Inception-V1*.  
Extraído de [42]

Para el problema CEUS, se hará uso de forma indirecta del clasificador de salida original del modelo *Inception-V3* ya que el modelo se inicializa con los pesos entrenados en ImageNet [28]. Sin embargo, durante la fase de entrenamiento y predicción, esta capa es reemplazada por otra estructura (cf. 4.3.1).

### 2.2.3. Medidas de desempeño en clasificación

Para medir el desempeño de la función  $f$ , existen varios indicadores. Los más comunes derivan de la matriz de confusión, que se presenta para el caso binario en la Figura 2.15.

- **Verdaderos Positivos (VP)**: número de observaciones de clase positiva que fueron clasificadas correctamente como clase positiva.
- **Falsos Negativos (FN)**: número de observaciones de clase positiva que fueron erróneamente clasificados como clase negativa.
- **Falsos Positivos (FP)**: cantidad de observaciones de clase negativa que fueron erróneamente clasificados como positivos.
- **Verdaderos Negativos (VN)**: cantidad de observaciones de clase negativa que fueron correctamente clasificados como negativos

Una matriz de confusión para 3 o más clases es la generalización del caso binario y las cantidades  $VP_i$ ,  $FN_i$ ,  $FP_i$  y  $VN_i$  pueden ser obtenidas para cada una de las clases  $c_i \in \{c_1, \dots, c_K\}$ , aglomerando las demás clases según muestra la Figura 2.16.

		Clase predicha	
		$C_1$	$C_2$
Clase real	$C_1$	VP	FN
	$C_2$	FP	TN

Figura 2.15: Matriz de confusión para un problema de clasificación binaria.  $C_1$  representa la clase positiva y  $C_2$  la negativa.

		Clase predicha				
		$C_1$	$C_2$	...	$C_K$	
Clase real	$C_1$	$m_{1,1}$	$m_{1,2}$	...	$m_{1,K}$	■ $VP_2$
	$C_2$	$m_{2,1}$	$m_{2,2}$	...	$m_{2,K}$	■ $FN_2$
	⋮	⋮	⋮	⋮	⋮	■ $FP_2$
	$C_K$	$m_{K,1}$	$m_{K,2}$	...	$m_{K,K}$	■ $VN_2$

Figura 2.16: Matriz de confusión para un problema de clasificación de  $K$  clases.

Los rectángulos indican la forma de agregar los valores para obtener  $VP$ ,  $FN$ ,  $FP$  y  $VN$  para la clase  $C_2$ .

La medida de desempeño utilizada para ambos problemas de este trabajo es el  $f1$ -score, que es el promedio armónico entre la precisión y la exhaustividad, es decir:

$$f1 = 2 \cdot \frac{\text{precisión} \cdot \text{exhaustividad}}{\text{precisión} + \text{exhaustividad}}$$

$$\text{precisión} = \frac{VP}{VP + FP} \quad , \quad \text{exhaustividad} = \frac{VP}{VP + FN}$$

Se usa el promedio armónico entre ambas cantidades ya que este tipo de promedio es adecuado para promediar tasas.

En un problema de  $K$  clases, se puede obtener un  $f1$ -score para cada una de las clases, definiendo tal como se muestra en la Figura 2.16. Para el caso del experimento CSC, se utiliza el promedio aritmético del  $f1$ -score obtenido en cada clase. En cambio, en CEUS se utiliza un promedio ponderado de este, definido por el desafío *Functional Map of the World* (FMoW).

## 2.3. Estado del arte

### 2.3.1. Clasificación de superficie construida

Una gran dificultad que presenta la detección de superficie construida consiste en que las áreas urbanas observadas por los sensores satelitales se caracterizan por la presencia de diferentes entidades que poseen diversas características espectrales. La variabilidad del espectro se debe a dos factores principales: en primer lugar, a la relación entre la resolución espacial del sensor y la escala característica de las entidades presentes, y en segundo lugar, a la distribución espacial de diferentes materiales en las áreas construidas y la mezcla de estos. Otro factor que también influye son las condiciones de iluminación, junto con los diferentes tamaños de edificios que causan sombras variables que afectan en gran medida el comportamiento espectral observado [18].

Además de esos factores provenientes de la naturaleza intrínseca del problema, existen otros factores que hacen más dificultosa esta tarea, como el hecho de que no existe una base de datos confiable y de gran escala de superficie urbana construida [20, 31]. A continuación se presenta un proyecto que intentó afrontar esta carencia mediante la utilización de diversas fuentes de información.

**Capa global de asentamientos humanos.** Proyecto desarrollado en el *Joint Research Institute of the European Commission*, que identifica superficie urbana construida mediante la segmentación de imágenes satelitales de *Landsat 8* para las épocas 1975, 1990, 2000 y 2015. Diversas instituciones colaboraron en su implementación, tanto de carácter internacional (ONU, Banco Mundial), como académicas (Universidad de Columbia, Universidad Católica de Lovaina, entre otras).

El algoritmo de clasificación está basado en reglas de asociación que permiten relacionar las características de las imágenes satelitales con datos provenientes de múltiples fuentes, como *Open Street Map*, datos espaciales de densidad poblacional y datos de extensión urbana de resolución más baja y más fina.

Los resultados están disponibles en línea para consulta pública. Además de un mapa mundial de superficie construida, han publicado una grilla que cuantifica el grado de confianza de la clasificación para cada uno de los píxeles. Los resultados obtenidos en un conjunto de prueba, presentan un promedio aritmético de exhaustividad de 0,77, lo que lo posiciona por

encima de otras iniciativas similares [18]. Dicho conjunto de prueba fue construido mediante muestras de campo por la Oficina Europea de Estadística (más conocida como Eurostat) y contiene 3 clases: cultivo, bosques y superficie construida [17].

### 2.3.2. Clasificación de edificaciones y usos de suelo

La clasificación de edificaciones y de usos de suelo, o más generalmente, la extracción de información semántica en imágenes satelitales, es mucho más reciente que la extracción de información contenida a nivel de píxel, como es el caso del problema CSC. Esto se debe principalmente a que la resolución espacial captada por los satélites modernos es mayor, permitiendo representar los objetos de interés con un mayor número de píxeles.

Sin embargo, aún si la resolución espacial de las imágenes hace posible una caracterización más rica de los objetos de interés, la cantidad de conjuntos de datos que permitan entrenar y evaluar modelos, es escasa.

#### Conjuntos de datos similares

Algunos de los esfuerzos más significativos que presentan similitudes con el conjunto de datos utilizado en esta memoria (FMoW[7]) se describen a continuación:

**UC Merced Land-Use Dataset.** Construido en 2010 y compuesto por 2.100 imágenes de escenas aéreas divididas en 21 clases desde 100 imágenes RGB de  $256 \times 256$  píxeles, con una resolución espacial de 0,3 m por píxel [46]. Algunas de las clases de usos de tierra y edificaciones son: autopista, bosque, campo de cultivo, estacionamiento, intersección vial, playa, puerto, residencial de alta densidad, residencial de densidad media, residencial de densidad baja y río. Este conjunto de datos fue extraído de ortoimágenes<sup>3</sup> aéreas que son parte del Mapa Geológico Nacional de los Estados Unidos, disponible en el sitio del *US Geological Survey* (USGS). Cabe mencionar que presenta clases difíciles de distinguir, como las tres clases de zonas residenciales, que difieren principalmente en la densidad de las estructuras. Este es el más conocido conjunto de datos para la tarea de la clasificación de usos de suelo y ha sido ampliamente utilizado [5].

**NWPU-RESISC45.** El conjunto de datos *Northwestern Polytechnical University* consiste en 31.500 imágenes, que cubren 45 clases de 700 imágenes por clase, con tamaños variables y fue confeccionado en 2017. Las clases de edificaciones y usos de suelo abarcan las ya mencionadas en el conjunto de datos anterior, más algunas como: aeropuerto, área comercial, área industrial, autopista, central térmica, humedal, desierto, estación de tren, iglesia, isla, lago y puente. Todas las imágenes fueron descargadas desde la plataforma *Google Earth*.

---

<sup>3</sup>Una ortoimagen es una fotografía aérea geométricamente corregida de modo que la escala es uniforme.

Algunas de las diferencias principales que presentan estos conjuntos de datos con el utilizado en CEUS son las siguientes: En primer lugar, la cardinalidad de las imágenes por clase es al menos un orden de magnitud superior para el conjunto FMoW. En segundo lugar, ambos conjuntos de datos mencionados sólo utilizan información proveniente de la imagen y no de los metadatos registrados por el satélite durante la captura. Por último, no existe posibilidad de explotar la información temporal, ya que sólo existe una observación por cada una de las imágenes.

## Extracción de características

Otro aspecto fundamental de la clasificación de edificaciones y usos de suelo en imágenes satelitales, es la extracción de características que permiten distinguir entre las clases. A continuación, describimos tres tipos diferentes de características que han ido variando en popularidad en los últimos años.

**Ingeniería de características.** En la literatura, fue el enfoque inicial que se utilizó para extraer características. Entre dichas características se distinguen dos subgrupos. Aquellas que extraen información a nivel de la imagen completa y otras que lo hacen a nivel local. Parte del primer grupo son: histogramas de colores, patrones de locales binarios y *GIST* [33], que miden color, textura, escala y orientación respectivamente. Para medir estos mismos aspectos, pero a nivel local, existen las características *SIFT* [29] y *HOG* [9]. Si bien este tipo de características es interpretable, los mejores resultados no son obtenidos con estas.

**Obtenidas con enfoque de AM no supervisado.** Son creadas con el fin de mejorar el desempeño de la predicción sin escatimar en la pérdida de interpretabilidad. Se pueden obtener mediante métodos provenientes de dos enfoques diferentes del AM no supervisado<sup>4</sup>, la reducción de dimensionalidad y el agrupamiento. Entre los métodos de reducción de dimensionalidad más populares empleados en este ámbito están: Análisis por Componentes principales, *sparse coding* [34] y *autoencoders* [22]. Por otro lado, los métodos de agrupamiento se utilizan para seleccionar observaciones que son representativas del conjunto de datos completo. Un método clásico de agrupamiento que es ampliamente utilizado es k-medias.

**Obtenidas con enfoque de AM supervisado.** Son el tipo de características que ha tomado mayor importancia en los últimos años [5]. Esto se debe a que a diferencia de las características provenientes de un enfoque no supervisado, el enfoque supervisado permite aprovechar el potencial de las etiquetas, para así extraer características que hacen más distinguibles observaciones de diferentes clases. El método más ampliamente utilizado es el de las redes neuronales convolucionales, que ya fueron abordadas en la Sección 2.2.2.

---

<sup>4</sup>A diferencia del AM supervisado, en el AM no supervisado se busca encontrar una estructura escondida en los datos, sin necesidad de una etiqueta dada por un supervisor.

# Capítulo 3

## Detección de superficie urbana construida

En este capítulo se describe cómo se aborda el problema de clasificación de superficie construida (CSC). En primer lugar, se describe parcialmente el cálculo del indicador ODS 11.3.1 (desde el cual se deriva este problema) y se describen las diferentes fuentes de datos que se utilizan: las imágenes satelitales *Landsat 8* y las etiquetas extraídas del Atlas de Expansión Urbana de NYU. A continuación se describe como se llevó a cabo el experimento, ahondando en el preprocesamiento de las imágenes y en la metodología para la selección de modelos. En seguida se detallan los resultados obtenidos, se interpretan dichos resultados comentando también las posibles fuentes de error de los modelos empleados y finalmente se concluye que el modelo obtenido presenta un buen desempeño al compararse con la solución descrita en la Sección 2.3.1.

### 3.1. Motivación

El indicador 11.3.1 (cf. Sección 1.1) está clasificado *tier 2* [14] ya que hay una metodología existente pero la producción estadística de los países no es suficiente. Esto da especial importancia a la realización de este trabajo.

Para entender el indicador es necesario definir dos conceptos:

- **Tasa de crecimiento de población (TCP):** mide el aumento de población en un territorio definido durante un periodo de tiempo (usualmente un año). La fórmula para su cálculo es:

$$\text{TCP} = \frac{\log\left(\frac{P_{t+y}}{P_t}\right)}{y}$$

donde  $P_t$  y  $P_{t+y}$  son la población al inicio y al final del periodo respectivamente, e  $y$  es la cantidad de años entre ambas mediciones.

- **Tasa de consumo de tierra (TCT):** es el aumento de superficie construida durante

un periodo de tiempo. Esta tasa se calcula según la fórmula:

$$\text{TCT} = \frac{\log\left(\frac{S_{t+y}}{S_t}\right)}{y}$$

donde  $S_t$  y  $S_{t+y}$  son la superficie construida al inicio y al final de periodo respectivamente, e  $y$  es la cantidad de años entre ambas mediciones.

Así, para obtener el indicador 11.3.1, se necesita tomar el cociente entre ambas tasas, esto es:

$$\text{Ind}_{11.3.1} = \frac{\text{TCT}}{\text{TCP}}$$

La relevancia de este indicador viene dada por la tendencia de las ciudades a expandir su zona periférica. Un estudio global sobre 120 ciudades, reveló que en promedio la TCT sobrepasa en promedio 3 veces la TCP [12]. En general este crecimiento acelerado corresponde a un uso ineficiente de la tierra, incrementando la segregación espacial, aumentando los costos de transporte y elevando los niveles de emisión de dióxido de carbono, entre otros.

## 3.2. Datos

### *Landsat 8*

El satélite fue lanzado en febrero del 2013 [2] y forma parte de la misión *Landsat* de la NASA que ya posee más de 45 años ininterrumpidos de captura de imágenes que están a libre disposición desde 2008. El método de captura se clasifica dentro de la teledetección pasiva, pues detecta radiación natural emitida o reflejada por el área que está siendo observada.

Las imágenes capturadas por *Landsat 8* poseen 11 bandas que se muestran en la Figura 3.1. Cada banda posee resoluciones espaciales diferentes, que varían entre 15 a 100 m. por píxel (Figura 3.1). Cabe mencionar que las bandas de 100 m. son adaptadas a resolución de 30 m. para así generar una descripción más simple del área cubierta por cada píxel. El satélite captura la imagen de un área en cuestión cada 16 días (resolución temporal) y posee una resolución radiométrica de 12 bits, que es escalada a 16 bits para los productos entregados al público.

Las imágenes fueron descargadas desde el sitio de USGS, seleccionando imágenes con la menor cantidad de nubes y que pertenecieran al año 2014, obteniendo así imágenes cercanas a las utilizadas por el Atlas de Expansión Urbana [16]. Se procede de esta forma ya que no se publicaron las fechas exactas de las imágenes utilizadas en dicha investigación.

### Atlas de Expansión Urbana

El Programa de expansión urbana del Instituto Marron de Gestión Urbana y la Escuela de Negocios Stern, ambos de la NYU, en colaboración con ONU-Hábitat y el Instituto Lincoln

Banda	Longitud de onda ( $\mu\text{m}$ )	Resolución espacial (m)
Banda 1 - Ultra Azul (coastal/aerosol)	0.43 - 0.45	30
Banda 2 - Azul	0.45 - 0.51	30
Banda 3 - Verde	0.53 - 0.59	30
Banda 4 - Rojo	0.64 - 0.67	30
Banda 5 - Infrarrojo cercano (NIR)	0.85 - 0.88	30
Banda 6 - Infrarrojo de onda corta (SWIR) 1	1.57 - 1.67	30
Banda 7 - Infrarrojo de onda corta (SWIR) 2	2.11 - 2.29	30
Banda 8 - Pancromático	0.50 - 0.68	15
Banda 9 - Cirrus	1.36 - 1.38	30
Banda 10 - Infrarrojo Térmico (TIRS) 1	10.60 - 11.19	100 * (30)
Banda 11 - Infrarrojo Térmico (TIRS) 2	11.50 - 12.51	100 * (30)

Tabla 3.1: Detalle de resolución espectral y espacial de *Landsat 8*

(\*) Las bandas 10 y 11 son adaptadas a resolución de 30 m. en los productos distribuidos al público.  
Adaptado desde [2]

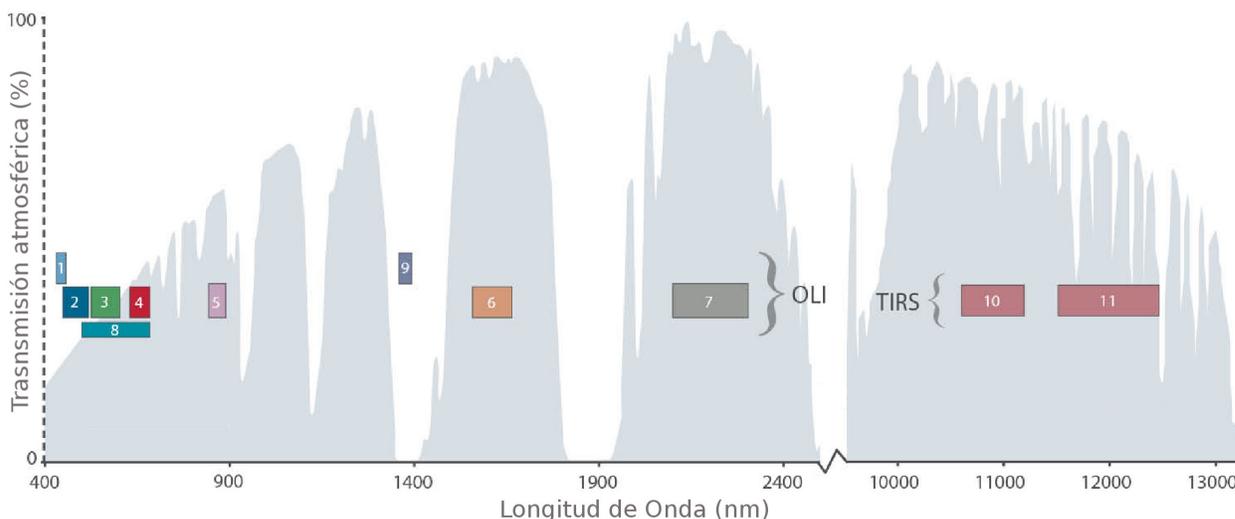


Figura 3.1: Emisión atmosférica y posicionamiento de las bandas de captura de satélite *Landsat 8*.

Adaptado desde [2]

de Política de Tierras, ha iniciado un esfuerzo de investigación de la expansión urbana global. Este esfuerzo dio lugar al Atlas de Expansión Urbana 2016 [16], que es una continuación de un proyecto anterior llamado Atlas de Expansión Urbana 2012 [3]. El programa de monitoreo estudia una muestra global representativa de 200 ciudades del mundo. Los resultados se encuentran en línea a disposición del público [15].

En el Atlas de Expansión Urbana se clasifican mediante un algoritmo con asistencia humana las zonas urbanas en tres categorías: zona construida, *espacio abierto* y *agua*. Cada imagen tiene en promedio 6,200 x 6,200 píxeles en tres años diferentes: 1990, 2000 y 2014. Los resultados de la clasificación son evaluados con un dataset de 10,000 imágenes de alta resolución disponibles en la plataforma *Google Earth Engine* y analizados por expertos en 2010, como parte de un trabajo previo [39]. Se logra una exactitud cercana al 90 %.

El Atlas constituye el etiquetado para el entrenamiento del algoritmo de clasificación que se elabora. La muestra que se utiliza como insumo para este estudio comprende los resultados sobre las 26 ciudades de América Latina y el Caribe seleccionadas por el Atlas. Los resultados se presentan en formato de imagen georreferenciada, por lo que su integración con las imágenes satelitales es sencilla (cf. Figura 3.2).

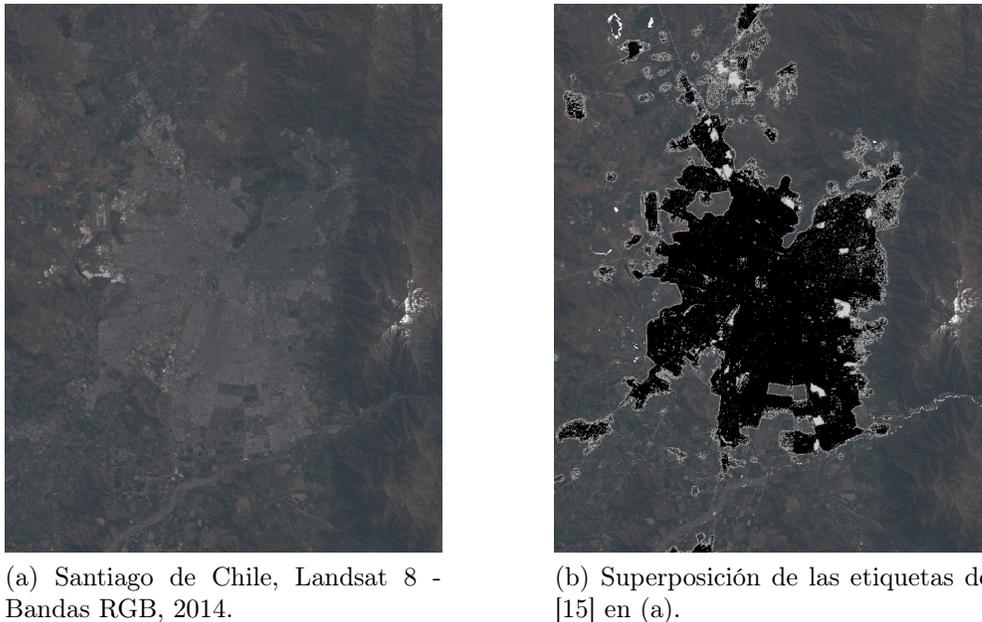


Figura 3.2: Consolidación de imagen satelital y etiquetas del Atlas de Expansión Urbana [16]. En b) las etiquetas se presentan en escala de grises, ignorando la etiqueta *espacio abierto* rural, que cubre todas las zonas que siguen visibles en RGB

### 3.3. Metodología

#### 3.3.1. Construcción de la base de imágenes

Ya que las imágenes de *Landsat 8* están compuestas por 11 bandas, cada píxel está representado por 11 características. Por otro lado, de [15] se obtienen 7 clases: urbano *construido*, suburbano *construido*, rural *construido*, *espacio abierto* urbano, *espacio abierto* suburbano, *espacio abierto* rural y *agua*. Para simplificar el problema estas clases se aglomeran en 3: *construido*, *espacio abierto* y *agua*. La distribución de clases se grafica en la Figura 3.3.

Si bien se trata de 26 ciudades, la totalidad de la superficie etiquetada para ellas se captura a través de 34 imágenes de *Landsat 8*. Dado que hay superficie etiquetada que se repite en imágenes que capturan zonas colindantes de la misma ciudad, es necesario elegir cuál de las imágenes es la que presentará dicha área. En la Figura 3.4 se muestra la totalidad de la superficie etiquetada, en sus respectivas imágenes.

Como análisis exploratorio, con el fin de analizar la separabilidad de las clases, se muestran en la Figura 3.5 los gráficos de nube de puntos para cada par de bandas. En el triángulo

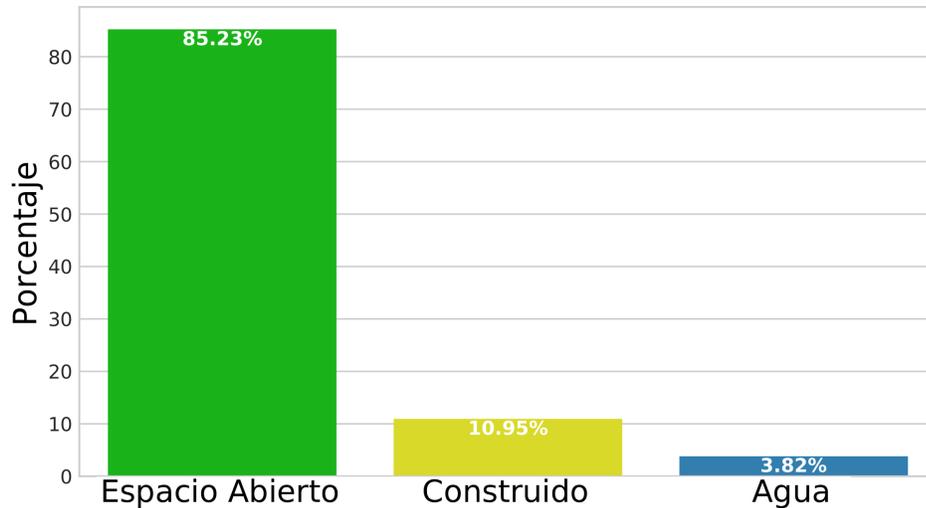


Figura 3.3: Distribución de clases aglomeradas en el problema CSC.

superior derecho se grafican por sobre las demás clases los píxeles de superficie construida, y en el triángulo inferior se grafican aquellos de la clase *espacio abierto* por sobre las demás. En la diagonal se grafican la distribución en la banda correspondiente, para cada una de las clases.

En la misma línea, para explorar la separabilidad de las clases se realiza un *t-distributed stochastic neighbor embedding* (t-SNE) [30]. Es un método probabilístico de reducción de dimensionalidad que permite aseverar si las clases son separables, pero no, si estas no lo son. Esto último se debe a que dos puntos serán cercanos en la proyección (Figura 3.6) con alta probabilidad si estos son cercanos en el espacio de alta dimensión.

### 3.3.2. Configuración del experimento

La cantidad total de píxeles etiquetados es de aproximadamente 115 millones. Los recursos computacionales y limitaciones de tiempo incitan a tomar muestras aleatorias de los datos para entrenar, evaluar y seleccionar los modelos. Cada muestra consistió en 3 subconjuntos: entrenamiento, validación y prueba. La división de conjuntos está hecha a nivel de ciudad, es decir, el conjunto de entrenamiento está constituido de 100 mil píxeles provenientes de 13 ciudades escogidas aleatoriamente. Por su parte, el conjunto de validación y prueba, son constituidos por 10 millones de píxeles provenientes de 6 y 7 ciudades respectivamente.

El conjunto de entrenamiento se diseña con igual cantidad de muestras de cada clase, dado el carácter desequilibrado de la distribución de clases del problema (cf. Figura 3.3). Esto último se fundamenta en que a comienzos de la investigación, los modelos entrenados en conjuntos de entrenamiento no balanceados, muestran comportamientos no satisfactorios, como por ejemplo, clasificando todos los píxeles en la clase mayoritaria.

Por otro lado, los conjuntos de validación y prueba son diseñados con una proporción de píxeles de cada clase acorde a la distribución de clases de la totalidad del conjunto de

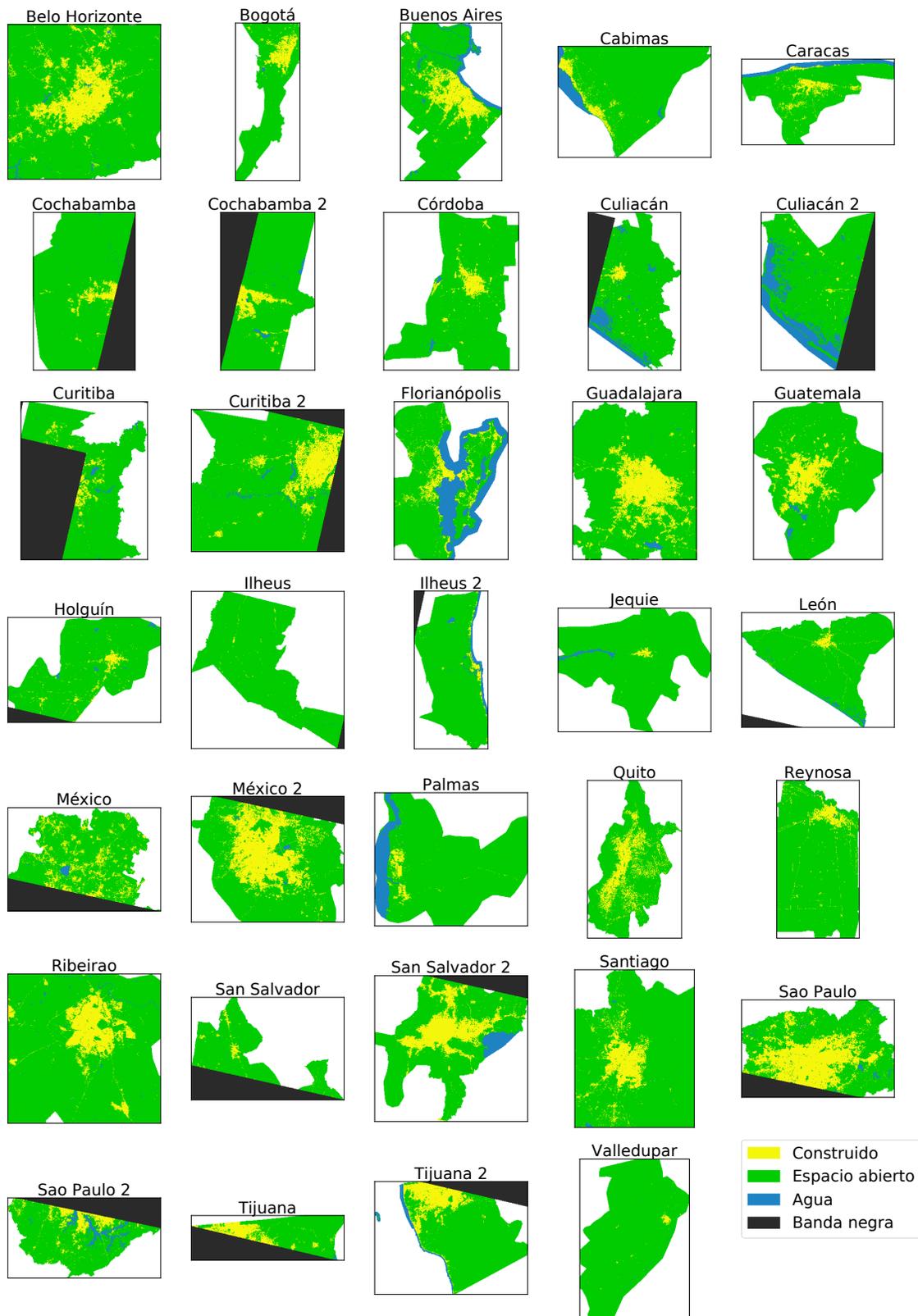


Figura 3.4: Distribución de clases en las ciudades del estudio en el problema CSC.  
Banda negra corresponde a un área de la imagen satelital que no presenta información.

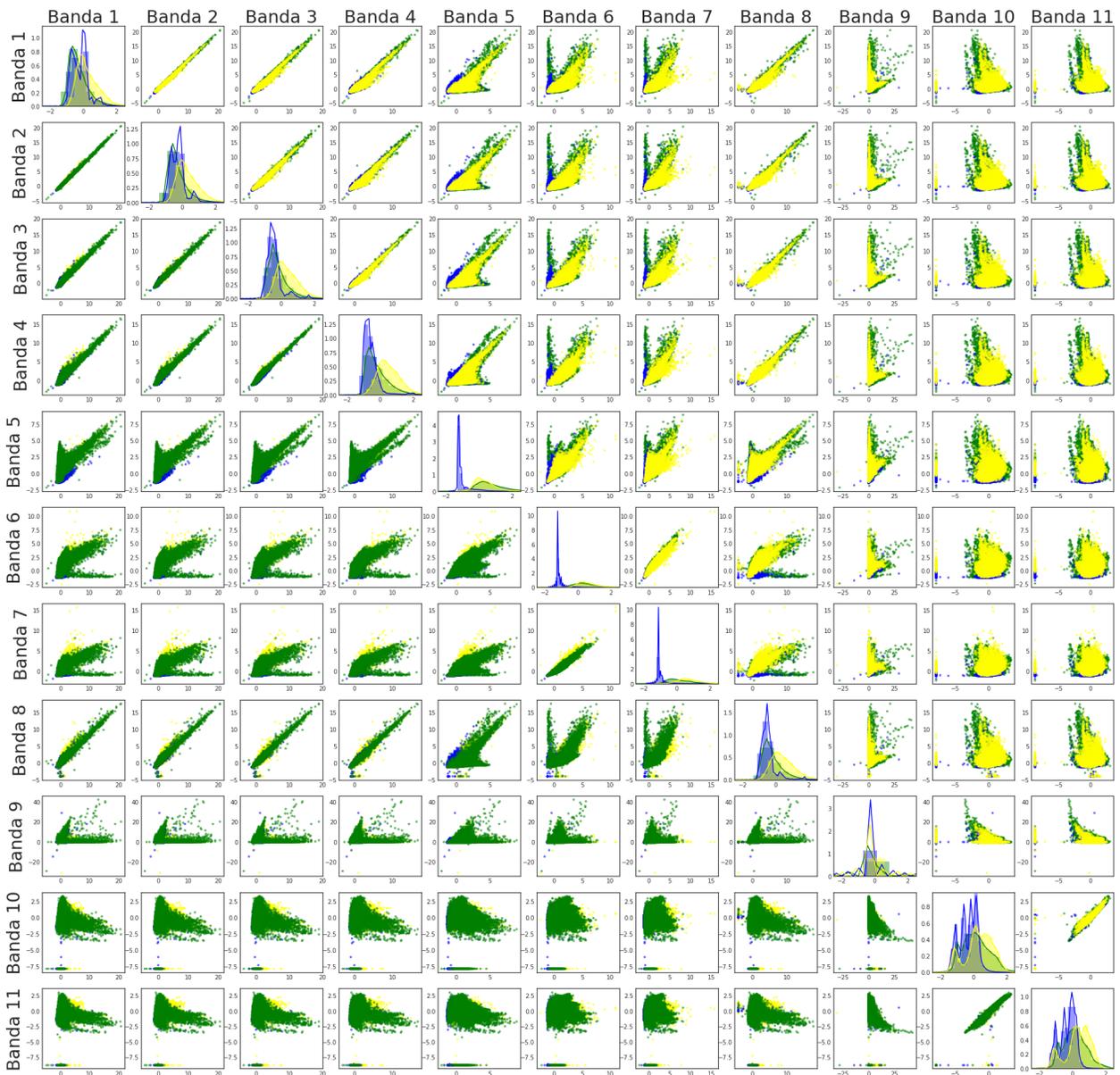


Figura 3.5: *Scatterplot Matrix* de una muestra de 10 millones de píxeles etiquetados.

datos, es decir, con un 85%, 11% y 4% para las clases espacio abierto, *construido* y *agua* respectivamente. Esta elección se hace en base al supuesto de que el problema que se quiere solucionar es desbalanceado, por lo que los modelos generados deben ser evaluados en esa configuración. Otra razón que motiva a fijar las proporciones de cada clase es que al generar 30 muestras en las que para cada uno de los conjuntos (entrenamiento, validación y prueba), las ciudades pertenecientes a estos varían según la muestra, la proporción de clases puede variar mucho en cada uno de los conjuntos, ya que existen ciudades en las que la presencia de algunas clases es escasa y la de otras muy grande. Por lo tanto, si no se fija una proporción de clases en los conjuntos de validación y prueba, se puede generar un sesgo en la medida de desempeño de los clasificadores.

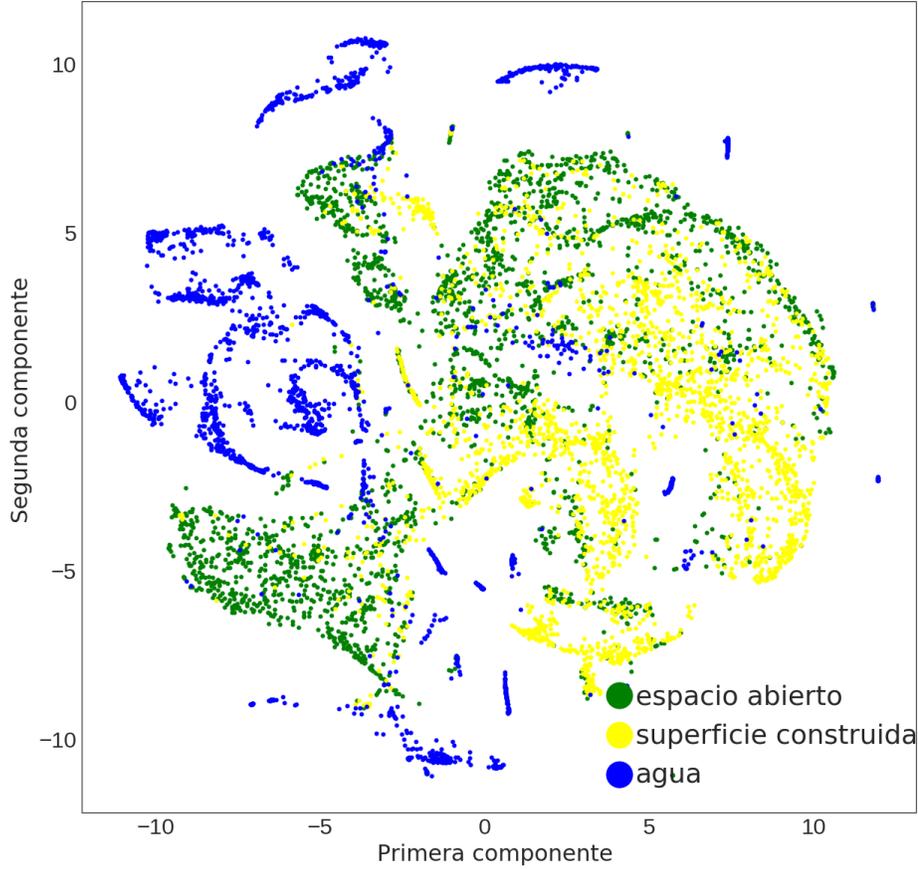


Figura 3.6: Representación t-SNE de una muestra de 100 mil píxeles

### 3.3.3. Preprocesamiento

Dentro de los procedimientos usuales para procesar imágenes satelitales está la llamada calibración radiométrica, que consiste en convertir el valor digital del píxel en radiancia, que es la cantidad física que captura el satélite originalmente (cf. Sección 2.1). Un segundo nivel en el procesamiento consiste en transformar la radiancia en otra cantidad física que depende del material observado. En el caso de las bandas del espectro solar (bandas 1 hasta 9 para *Landsat 8*) dicha cantidad es la reflectancia. En cambio para las bandas de longitud de onda térmicas (bandas TIRS), la cantidad asociada es la temperatura.

Para obtener la radiancia a partir del valor digital de la banda  $k$ , se aplica la transformación lineal 3.2.

$$L_k = a_k^0 + a_k^1 \text{VD}_k \left[ \frac{W}{\text{sr} \times \text{m}} \right] \quad (3.1)$$

donde  $a_k^0 = \frac{L_k^{\max} - L_k^{\min}}{R_{\text{VD}}}$  con  $R_{\text{VD}}$  el rango de valores capturados por la banda  $k$  y  $L_k^{\max}$ ,  $L_k^{\min}$  son el máximo y el mínimo valor de radiancia capturado por la banda  $k$ , respectivamente. Por su parte, la constante  $a_k^0$  induce un sesgo adecuado para los bajos niveles de radiancia. Dado que este cálculo se realiza con constantes  $a_k^0$ ,  $a_k^1$  que aproximan la magnitud de la radiancia en la parte superior de la atmósfera, este valor se denomina *Top-of-the-Atmosphere* (ToA) radiancia.

La ToA reflectancia se define como la energía reflejada desde una superficie en la dirección angular del sensor y normalizada por la energía incidente en dicha superficie [8]. Para obtenerla a partir de la radiancia de la banda  $k$ , la fórmula es la siguiente:

$$\rho_k = \frac{D\pi L_k}{E_k^0 \cos(\theta_i)} \quad (3.2)$$

donde  $D$  es un factor de corrección que toma en cuenta las variaciones de distancia entre la tierra y el sol en función del día del año,  $L_k$  el valor de radiancia (ecuación 3.1),  $E_k^0$  es la irradiancia solar en la parte superior de la atmósfera y  $\theta_i$  es el ángulo de incidencia en la superficie observada con respecto a la vertical.

Para obtener la ToA temperatura a partir de las bandas de onda térmica mediante los valores de radiancia, la ecuación es la siguiente:

$$T(L_k) = \frac{c_2\nu_k}{\ln\left(\frac{c_1\nu_k^3}{L_k}\right)} [\text{K}] \quad (3.3)$$

donde  $c_1$  y  $c_2$  son constantes derivadas del inverso de la ecuación de Plank para cuerpos negros [37] y  $\nu_k$  es la longitud de onda central de la banda  $k$ .

Cabe mencionar que todos las constantes (o una aglomeración de ellas) mencionadas en esta sección (ecuaciones (3.1) - (3.3)), vienen incorporadas en los metadatos de las imágenes *Landsat 8*.

Una vez realizada la conversión de valores digitales a magnitudes de ToA reflectancia y temperatura, y antes de comenzar la fase de entrenamiento, todas las bandas son centradas y escaladas (media 0 y varianza 1), de acuerdo a los estadísticos de cada uno de los conjuntos de entrenamiento.

### 3.3.4. Selección de modelos

Para cada una de las 30 muestras, se entrenan 9 modelos MSV, 6 de RL y 4 GTB y 10 NN, luego se evalúan en su conjunto de validación respectivo y se seleccionan los mejores hiperparámetros para cada familia de algoritmos. Finalmente se reentrenan los modelos con los hiperparámetros seleccionados en el conjunto de entrenamiento y un subconjunto del conjunto de validación, para ser evaluados en el conjunto de prueba.

Los modelos MSV fueron implementados en la librería *scikit-learn* [36]. Todos ellos son de kernel gaussiano, ya que en pruebas iniciales se constató un mejor rendimiento utilizando dicho *kernel*. Los hiperparámetros que se seleccionan mediante los conjuntos de validación son el costo asociado a una clasificación errada ( $C$ ) y el ancho del *kernel* gaussiano ( $\gamma$ ).

Por su parte, en los modelos de RL, los hiperparámetros que se hacen variar son el coeficiente de regularización<sup>1</sup> y la estrategia para generalizar a un problema multiclase (cf. Sección 2.2.2). Al igual que los MSV, son implementados en *scikit-learn*.

---

<sup>1</sup>En general, cuando se habla de un coeficiente de regularización en AM, es una cantidad que penaliza la magnitud de algunos parámetros del modelo que están relacionados con el sobreajuste.

algoritmo	desde	hasta	#
parámetros			
<b>MSV</b>			9
Costo observaciones mal clasificadas ( $C$ )	0,1	10	3
Ancho del kernel ( $\gamma$ )	0,001	0,1	3
<b>RL</b>			6
Coefficiente de regularización	0,1	10	3
Esquema multiclase	OVR	entropía cruzada	2
<b>GBT</b>			4
Tasa de aprendizaje	0,001	1	4
<b>RN</b>			10
Dropout	0,2	0,4	2
Arquitecturas	-	-	5

Tabla 3.2: Variación de hiperparámetros para cada algoritmo.

modelo	C.O.1	C.O.2	C.O.3
A	44	22	22
B	22	44	22
C	22	22	44
D	44	44	-
E	22	22	22

Tabla 3.3: Número de neuronas por capa para las diferentes arquitecturas de RN.

En lo que concierne al algoritmo GTB, son implementaciones del modelo XGBoost [4] para clasificación, de la librería *xgboost* [4]. El hiperparámetro que se ajusta en el conjunto de validación es la tasa de aprendizaje. En los términos de la Sección 2.2.2, se multiplica la magnitud del paso óptimo ( $\gamma$ ) encontrado en cada paso por este coeficiente, de tal forma que el ajuste de parámetros se haga con menor velocidad (dicho parámetro debe tomar valores entre 0 y 1).

Por último, los diferentes modelos de RN totalmente conectada se implementan en Keras [6] con *backend* de Tensorflow [1]. Los hiperparámetros ajustados son el *dropout* y la arquitectura del modelo (como se muestra en la Tabla 3.3). Al igual que la normalización por *batches* 2.2.2, el *dropout* es una técnica de regularización de RN. Una capa con un *dropout* de  $p$  (con  $p \in [0, 1]$ ) significa que con probabilidad  $p$  cada una de las neuronas no se activará, y las restantes amplificarán su salida por  $\frac{1}{p}$  [41].

### 3.4. Resultados

Lo que corresponde a la selección de hiperparámetros para cada algoritmo, se muestra en la Tabla 3.4.

Algoritmo	parámetro
parámetros	óptimo
<b>MSV</b>	
Costo observaciones mal clasificadas ( $C$ )	10
Ancho del kernel ( $\gamma$ )	0,1
<b>RL</b>	
Coefficiente de regularización	10
Esquema multiclase	OVR
<b>GBT</b>	
Tasa de aprendizaje	1
<b>RN</b>	
Dropout	0,2
Arquitecturas	A

Tabla 3.4: Hiperparámetros óptimos para cada algoritmo.

En la Figura 3.7 se muestra un diagrama de caja<sup>2,3</sup> con el desempeño en términos del  $f1$ -score (ver Sección 2.2.3) obtenido por los clasificadores seleccionados a través de las 30 muestras.

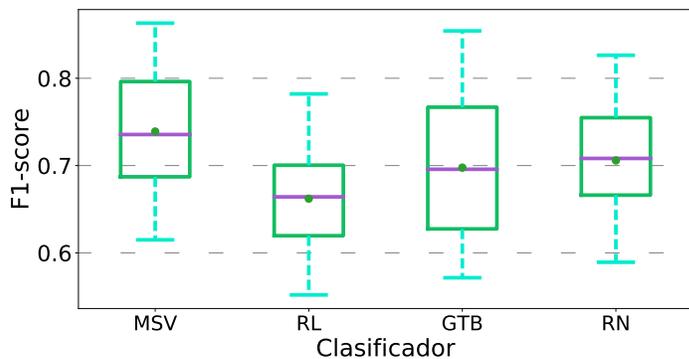


Figura 3.7: Diagrama de caja de  $f1$ -score para los clasificadores seleccionados.

	RL	GTB	RN
MSV	1,7e-6	2,8e-4	1,2e-3
RL		2,4e-3	3,1e-4
GTB			5,4e-1

Tabla 3.5: P-valor del Test de Wilcoxon en el  $f1$ -score obtenido en el conjunto de prueba.

Se observa que MSV exhibe el mejor rendimiento con una diferencia estadísticamente significativa según el Test de Wilcoxon [45], que permite cuantificar la probabilidad de que la diferencia entre medias de dos mediciones diferentes haya ocurrido por azar. Como se observa en la tabla, hay una probabilidad siempre menor al 0.3 % de que la diferencia entre la media

<sup>2</sup>En la Figura 3.7, los bordes inferior y superior del recuadro verde son el primer y tercer cuartil, respectivamente. La línea morada en el medio es la mediana y el punto verde es la media. Los brazos terminan en las observaciones más pequeñas y más grandes que están a lo más a distancia 1.5 RI del primer y tercer cuartil, respectivamente. Finalmente, los círculos blancos son considerados *outliers* para este tipo de gráfico.

<sup>3</sup>Rango Intercuartil (RI): diferencia entre el primer y tercer cuartil de una distribución.

obtenida por el modelo de MSV y cualquiera de los otros modelos sea producto del azar.

Para observar el desempeño de los clasificadores en cada una de las clases, se adjunta la Figura 3.8. Se constata que los modelos presentan una mayor diferencia entre clases para la precisión, donde se obtienen resultados altos (media sobre 0.96) para la clase *espacio abierto* y bajos para *construido* y *agua* (media inferior al 0.5 y 0.6 respectivamente).

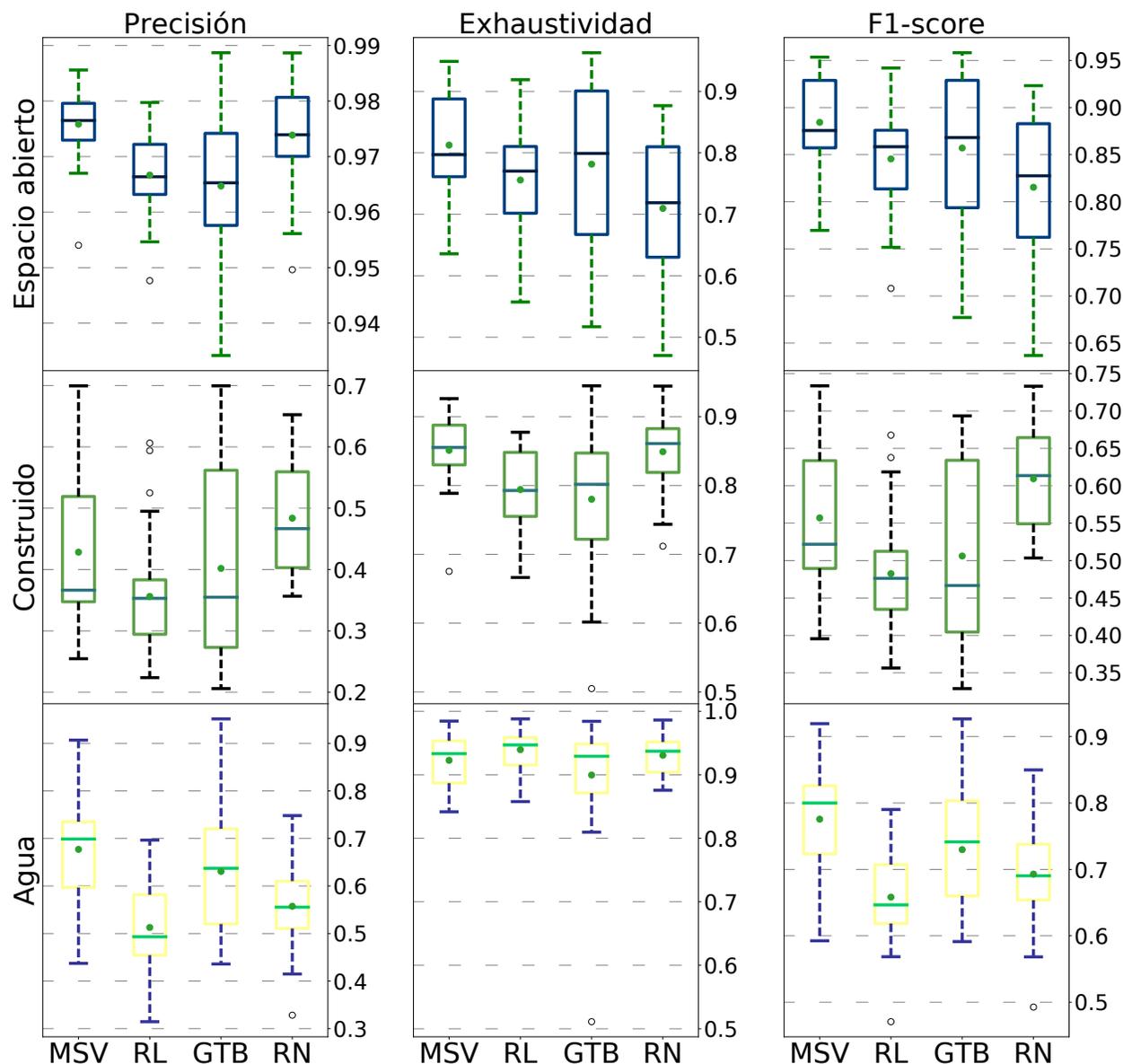


Figura 3.8: Diagrama de caja de precisión, exhaustividad y  $f1$ -score para los clasificadores seleccionados en cada una de las clases

Para ilustrar cómo el mejor clasificador (aquel que presenta el mejor  $f1$ -score en la Figura 3.7) se desempeña en una imagen del conjunto de prueba, se muestra su predicción para Caracas, Venezuela, en la Figura 3.9a. Su matriz de confusión se puede ver en la Figura 3.9b.

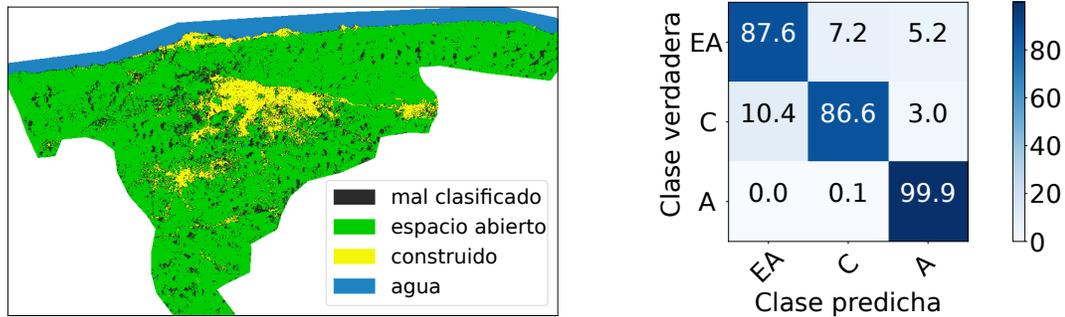


Figura 3.9: Resultados para Caracas: (a) Comparación entre etiquetas verdaderas y predichas. (b) Matriz de confusión

### 3.5. Análisis y discusión

El bajo desempeño en términos de precisión obtenida en las clases *construido* y *agua* en comparación a la de *espacio abierto*, es de esperar dado el desbalance de clases en el conjunto de prueba. Dado que la proporción de dichas clases es 11 %, 4 % y 85 % respectivamente, es mucho más probable que haya una cantidad importante de falsos positivos en comparación a los verdaderos positivos para las clases minoritarias (*construido* y *agua*) y análogamente, que la cantidad de falsos positivos sea muy baja en comparación a los verdaderos positivos de la clase *espacio abierto*. Por otro lado, en términos de exhaustividad se constata un rendimiento similar a nivel de las clases, lo que se condice con el hecho de que la exhaustividad es invariante al desbalance de clases.

El hecho de utilizar una métrica que es sensible al balance de clases como lo es la precisión y por tanto el *f1-score*, podría ser visto como algo incorrecto. Sin embargo, se cree que el sólo centrarse en una métrica invariante al balance de clases, como es la exhaustividad, no permite cuantificar una noción clave del desempeño de un clasificador, como es la precisión, que estima la probabilidad con la que un elemento clasificado de una clase, pertenece realmente a esta. Para el caso del indicador 11.3.1, la precisión es clave para poder evaluar la confiabilidad de la estimación del área cubierta superficie construida.

Otro aspecto interesante que se puede observar en la Figura 3.8, es que existe una dispersión importante tanto en los resultados en cuanto a precisión como a exhaustividad a lo largo de las clases. Una explicación plausible puede ser que la cardinalidad del conjunto de entrenamiento (menos de un 0,1 % de los datos etiquetados), no sea suficiente para representar correctamente a las clases del problema, sobre todo en la clase *espacio abierto* donde los 33.3 mil píxeles de dicha clase representan menos de un 0.03 % del total de datos etiquetados de la clase.

Al comparar los resultados en la Figura 3.9a obtenidas por el clasificador MSV con el mayor *f1-score*, con la imagen en RGB (Figura 3.10), se puede notar que el algoritmo presenta problemas con la presencia de nubes. Sin embargo, al realizarse pruebas con un algoritmo capaz de identificar las nubes, llamado FMask [47] y agregar la clase nubes a nuestra base de datos, se obtienen resultados menos satisfactorios.



Figura 3.10: Caracas - Venezuela, Landsat 8 - RGB, 2014.

4

Si bien no es posible comparar los resultados obtenidos en este ejercicio, con los citados en la Sección 2.3.1, ya que no se usa el mismo conjunto de datos de prueba, cabe mencionar que el promedio aritmético de la exhaustividad obtenida entre las clases *construido* y *espacio abierto* (que son la totalidad de las clases en dicha investigación) es de 0.83 y por lo tanto es superior a 0.77 que es obtenido en [18].

# Capítulo 4

## Clasificación de edificaciones y usos de suelo

En este capítulo se describe como se aborda el problema de clasificación de edificaciones y uso de suelo (CEUS). Al comienzo se describe parcialmente el cálculo del indicador ODS 11.3.7, el cual motiva este experimento. En segundo lugar, se describen los datos utilizados: FMoW, un conjunto de más de 500 mil imágenes satelitales de alta resolución, etiquetadas en 63 clases de usos de suelo. Dichos datos son parte del desafío en línea homónimo, en el marco del cual se desarrolla todo el trabajo descrito en el capítulo. Una vez descrito este contexto, se describe la arquitectura de los modelos empleados y los diferentes procedimientos mediante los cuales estos son entrenados. En seguida, se explica el preprocesamiento de los datos abordando el procedimiento de aumentación de datos y otros procesos relacionados con la imagen y los metadatos. Se detalla además la métrica con la que se evalúa el desempeño de los modelos obtenidos, que es la misma utilizada para posicionar a los concursantes del desafío. Finalmente, se describen los resultados obtenidos, que permiten concluir cuál es el procedimiento de entrenamiento con mejor desempeño, se comenta sobre las posibles fuentes de error de los modelos construidos y se concluye que el uso de imágenes satelitales de alta resolución en conjunto con modelos de RNC pueden representar una alternativa factible para el cálculo del indicador mencionado.

### 4.1. Motivación

El indicador 11.7.1 (cf. Sección 1.1) es parte de los indicadores *tier 3* dado que ni la metodología está definida, ni los datos producidos por los países son suficientes para la producción del indicador [14]. Para su cálculo son necesarios tres componentes:

- a) identificación de la superficie urbana como superficie construida,
- b) estimación de la superficie total de espacios abiertos y
- c) análisis de la parte de la superficie construida correspondiente a calles.

El componente *a)* es analizado en el Capítulo anterior ya que es un subproducto del cálculo del indicador 11.3.1. El componente *b)* es el que incita el desarrollo del trabajo descrito en este capítulo, ya que se propone utilizar imágenes satelitales de alta resolución espacial, para facilitar el cálculo de dicho componente. Por último el componente *c)* escapa del alcance del presente trabajo.

Una vez que aquellos tres componentes son obtenidos, el cómputo del indicador 11.3.7 está dado por la siguiente fórmula:

$$\frac{\text{Superficie total de espacios abiertos} + \text{Superficie total correspondiente a calles}}{\text{Superficie total construida}} \quad (4.1)$$

Si bien, es reconocido por ONU-Hábitat que el uso de imágenes satelitales no es suficiente insumo para el cómputo del componente *b)* [12], ya que este requiere diferenciar los espacios abiertos de uso público de los de uso privado, se decide explorar el uso de imágenes satelitales para esta tarea. Se cree que puede ser de utilidad generar sistemas de clasificación automática de espacios abiertos, que combinados con otras fuentes de información podrían ser parte de una metodología de bajo costo para el cálculo de dicho componente. Sin embargo, la concepción de una solución de este tipo, va más allá del alcance de esta memoria. Como fue mencionado en la Sección 1.2.1, sólo se busca aportar elementos que faciliten la definición de metodologías de cálculo para el indicador en cuestión.

## 4.2. Datos

El conjunto de datos utilizado para el problema CEUS es denominado FMoW. Fue construido por *John Hopkins University* y la empresa norteamericana *Digital Globe*. Fue lanzado como parte del desafío homónimo que fue alojado por *topcoder.org* [24] entre julio 2017 y febrero 2018. Los datos consisten en alrededor de 500 mil imágenes satelitales de 130 mil puntos de interés (POI, por sus siglas en inglés), etiquetadas dentro de 63 clases diferentes, en donde 62 corresponden a edificaciones y usos de suelo, y la restante a detección falsa (cf. Figura 4.1).

Existen dos presentaciones del conjunto de datos: **fmow-full** y **fmow-rgb**. En dichas presentaciones, las imágenes están en formatos TIF y JPG respectivamente. Aquellas en formato TIF consisten en imágenes multiespectrales de 4 u 8 bandas. En cuanto a las imágenes en JPEG, sólo se conservan las 3 bandas RGB de dichas imágenes multiespectrales. Sin embargo, a ambas presentaciones (**fmow-full** y **fmow-rgb**) se les añade una proyección de la totalidad de las bandas en RGB mediante el algoritmo *WorldView-2 Pan-sharpening* [35], que además de comprimir la información de todas las bandas en sólo 3, permite transferir la resolución espacial de la banda de alta frecuencia a aquellas 3 bandas. Dicho de otra forma, esto permite llevar la resolución espacial de las bandas no pancromáticas (2 m. por píxel en promedio) a la de la banda pancromática (0.5 m. por píxel en promedio). Para el experimento que se efectúa en esta memoria, sólo se utilizan las imágenes proyectadas en formato JPG. En cuanto a la información temporal capturada en el conjunto de datos FMoW, para cada POI se puede presentar más de una imagen. Esta multiplicidad corresponde a capturas realizadas en



Figura 4.1: Muestra aleatoria de una imagen por clase del conjunto de datos FMoW.

diferentes instantes de tiempo. En la Figura 4.2 se muestra la distribución de dicho número de observaciones temporales por POI para el conjunto de datos.

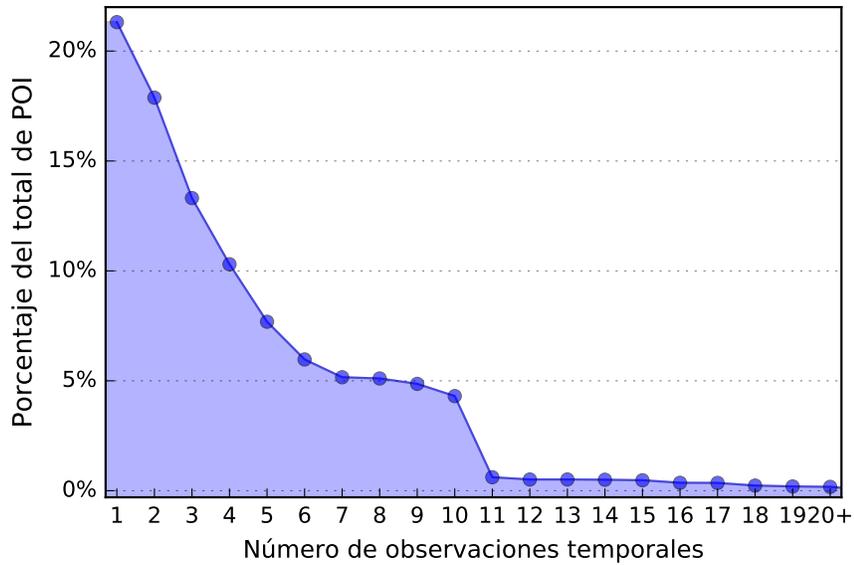


Figura 4.2: Distribución del número de imágenes por POI que corresponden a observaciones en diferentes instantes de tiempo en el conjunto de datos FMoW.

Dado que este conjunto de datos fue construido para un desafío en línea, no todas las imágenes del conjunto de datos fueron liberadas, ni tampoco todas las etiquetas de los diferentes POI. El conjunto de datos FMoW, está dividido en cuatro subconjuntos que se muestran en la Figura 4.3. Los subconjuntos de entrenamiento y validación fueron puestos a disposición de los competidores, incluyendo las imágenes y las etiquetas. Por su parte, el conjunto de prueba fue utilizado para seleccionar las diez soluciones con mejores resultados en la primera fase del concurso. Por último, los organizadores utilizaron al conjunto secuestrado para elegir al ganador entre las diez soluciones ya mencionadas.

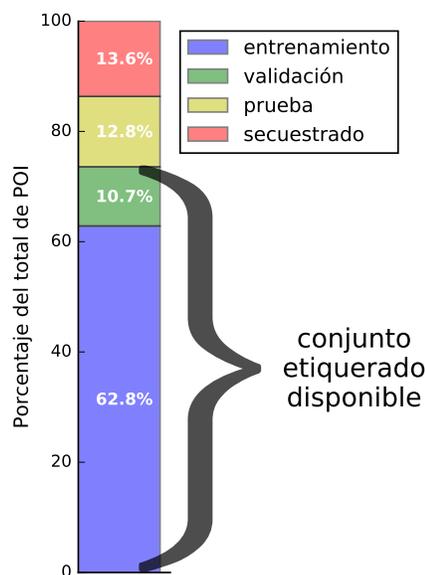


Figura 4.3: División del conjunto de datos FMoW.

## 4.3. Metodología

### 4.3.1. Arquitectura del modelo

El modelo de referencia propuesto por los organizadores de la competencia FMoW, consiste en una red convolucional *DenseNet 161* [23], en cuya salida se concatenan los metadatos normalizados (cf. Sección 4.3.3). Este vector concatenado se utiliza como entrada para una red neuronal completamente conectada de dos capas de 4096 neuronas y una capa de 63 neuronas (lo que corresponde al número de clases) cuya función de activación es *softmax* [7].

El modelo propuesto en esta memoria intercambia la red convolucional *huang2017densely 161* por una red *Inception-V3* (cf. Figura 4.4). Las demás estructuras se mantienen inalteradas, lo que permite evaluar comparativamente el desempeño de la arquitectura *Inception V3* para este problema en específico.

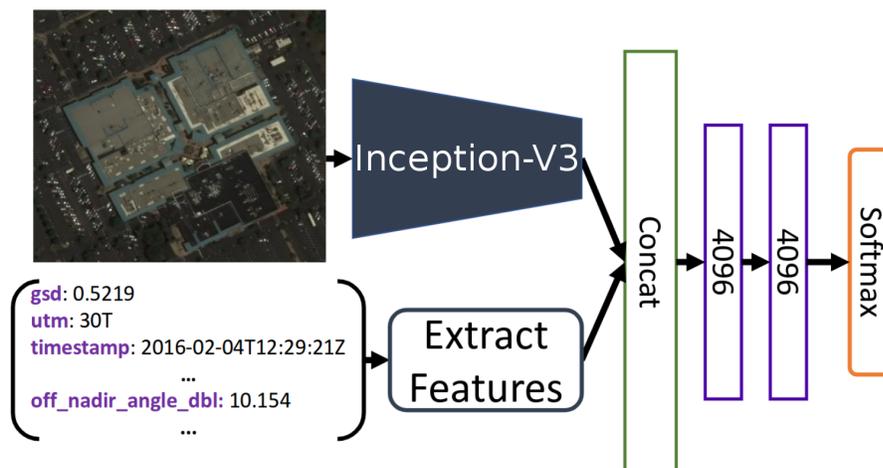


Figura 4.4: Arquitectura del modelo de referencia FMoW.  
Adaptado de [7]

### 4.3.2. Procedimientos de entrenamiento

Como se muestra en la Figura 4.3 la cantidad de datos etiquetados disponibles para el experimento es menor que la utilizada en el desafío FMoW. Con el fin de hacer lo más análogo posible esta experiencia a la primera fase del concurso (cf. Sección 4.2), se replica la proporción de POI pertenecientes a los conjuntos de entrenamiento, validación y test. Dicho de otra forma, como para la primera fase del desafío el subconjunto *secuestrado* no era parte del conjunto de datos a analizar, se mantuvieron las proporciones de POI de entrenamiento, validación y prueba, sin considerar el subconjunto *secuestrado*, dividiendo solamente el *conjunto etiquetado disponible* (cf. Tabla 4.1). Esta división se realizó 4 veces, aspirado lograr más robustez estadística en los resultados obtenidos.

Conjunto	Proporción en segunda fase	Proporción en primera fase	Cantidad de POI en el experimento
entrenamiento	62,8 %	72,8 %	71.091
validación	10,7 %	12,4 %	12.109
prueba	12,8 %	14,8 %	14.452

Tabla 4.1: División de los conjuntos de entrenamiento, validación y prueba.

Al igual que en el modelo de referencia, la red convolucional es inicializada con los pesos de *ImageNet* 2012 [28]. Como función de pérdida se utiliza la entropía cruzada (cf. Sección 2.2.2) y como algoritmo optimizador *Adam* [27] con un tasa de aprendizaje inicial de  $10^{-4}$ .

Con el fin de explorar el impacto de la distribución de clases desequilibrada (cf. Figura 4.5) se crearon cuatro procedimientos de entrenamiento:

- **No balanceado, no ponderado (NN)**: no hay sobremuestreo de ninguna de las clases, ni tampoco se le agrega peso a las muestras.
- **No balanceado, ponderado (NP)**: no se aplica sobremuestreo de ninguna clase, pero se asigna peso inversamente proporcional al soporte de las clases.
- **Balanceado, no ponderado (BN)**: dada la cantidad de muestras máxima por clase, la cual es alcanzada por la clase *campo de cultivo* (cf. Figura 4.5), se sobremuestran todas las demás, hasta alcanzar aquella cantidad de muestras para todas las clases. No se da peso a las muestras.
- **Balanceado, ponderado (BP)**: se aplica sobremuestreo hasta alcanzar el balance de clases (análogo a BN) y además se asigna peso proporcional a la importancia de las clases en el cálculo del promedio ponderado de la competición FMoW (ver la Sección 4.3.4).

Adicionalmente, todos los modelos fueron entrenados con *Early Stopping*<sup>1</sup>, monitoreando el criterio del desafío FMoW, es decir el promedio ponderado del *f1-score* (cf. Sección 2.2.3) de las clases. En la Sección 4.3.4 se profundiza en la ponderación de cada una de las clases.

La *paciencia*<sup>1</sup> para los modelos NN y NP es tres, en cambio para los modelos BN y BP es uno. Esta discrepancia se justifica en la diferencia de tamaño del conjunto de entrenamiento para los procedimientos balanceados y no balanceados, ya que el tamaño de este, para el procedimiento balanceado, es aproximadamente 5 veces el del conjunto de entrenamiento del procedimiento no balanceado.

---

<sup>1</sup>Heurística de regularización ampliamente usada en el entrenamiento de RN en la cual cada cierto número de iteraciones de descenso de gradiente, se verifica el progreso de alguna métrica sobre el conjunto de validación. Si el progreso de la métrica se estanca o empeora, durante  $p$  verificaciones, se detiene el proceso de entrenamiento.  $p$  se denomina *paciencia*.

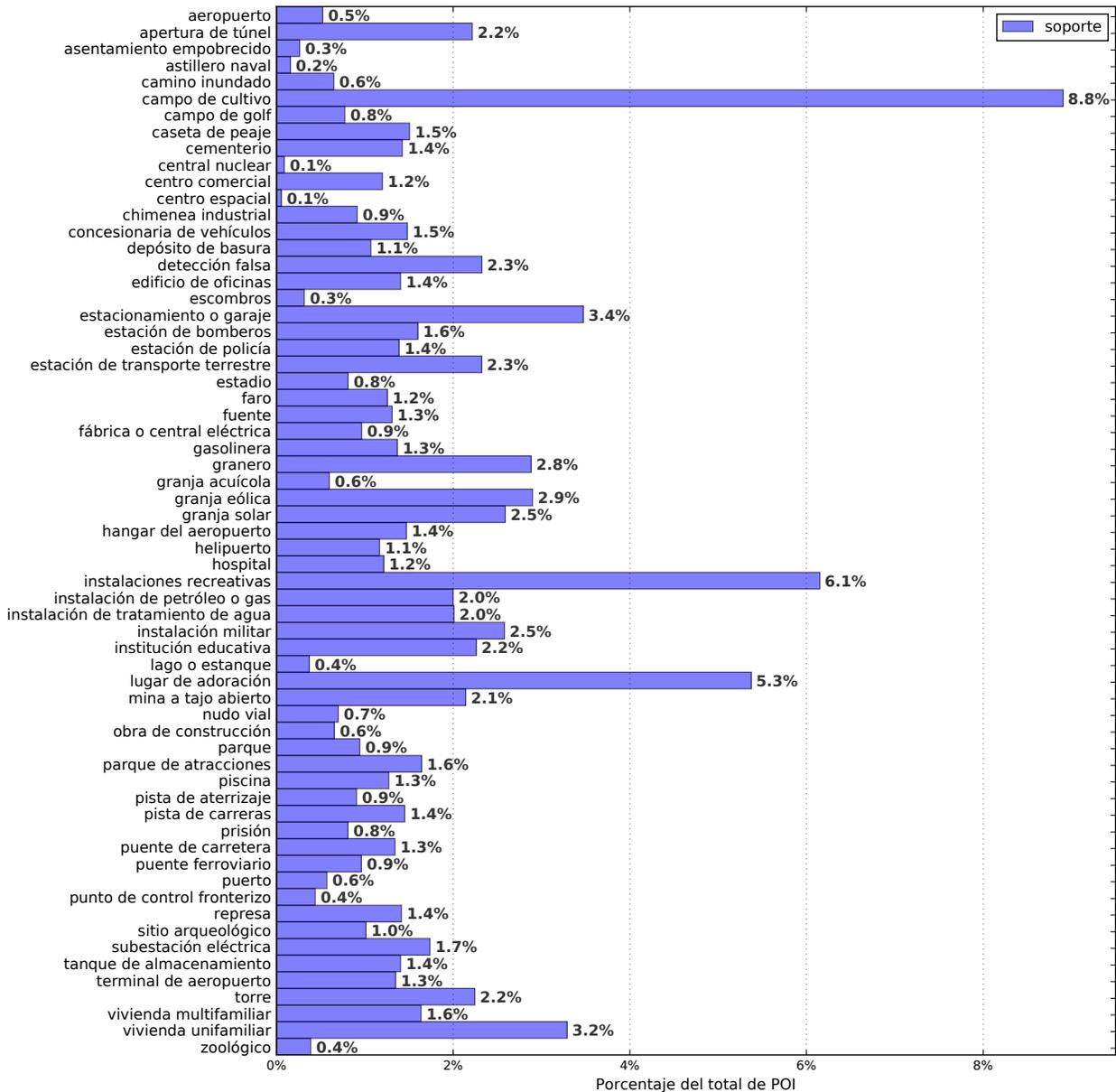
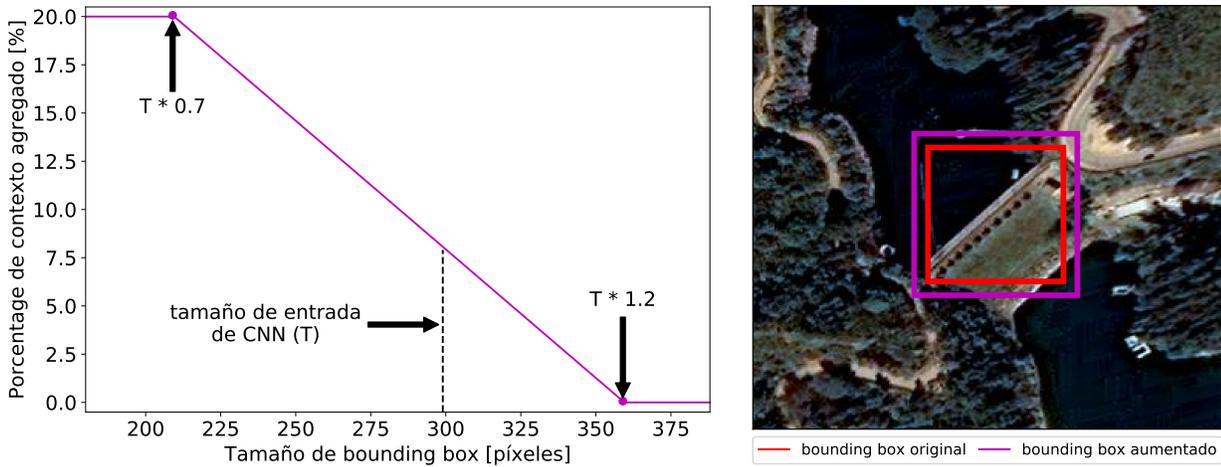


Figura 4.5: Soporte de cada una de las clases a nivel de POI.

### 4.3.3. Preprocesamiento

En lo que concierne a las imágenes, siguiendo lineamientos dados en [7], se emplea un procedimiento para agregar más contexto a los *bounding box* de tamaño pequeño (relativo al tamaño de la entrada de la red convolucional, que es de  $299 \times 299$  píxeles). Este consiste en agregar un 20% de margen a los *bounding box* de tamaño menor a 225 píxeles y no agregar margen alguno a los *bounding box* de tamaño mayor a 360 píxeles. Para los *bounding box* de tamaño entre esas dos cotas, interpolando linealmente, se agrega un margen proporcional al tamaño. En la Figura 4.6 se ilustra el comportamiento de esta función. Con las nuevas coordenadas del *bounding box* enriquecido, se recortan las imágenes y se redimensionan a  $299 \times 299$ .



(a) Función de enriquecimiento de *bounding box*.

(b) Ilustración de *bounding box*. aumentado

Figura 4.6: Procedimiento de aumentación de tamaño de los *bounding box*.

Con el propósito de hacer aumentación de datos<sup>2</sup>, las imágenes de entrenamiento son rotadas aleatoriamente en  $180^\circ$  para los ejes x e y, y entre  $0^\circ$  y  $360^\circ$  para el eje z (cf. Figura 4.7).

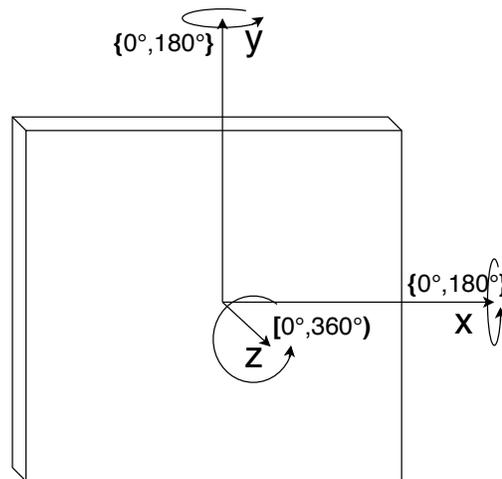


Figura 4.7: Rotación de imágenes en el conjunto de entrenamiento.

Con el fin de acelerar los tiempos de entrenamiento y generar un balance en la utilización de recursos computacionales (CPU y GPU) se decide hacer el procedimiento de recorte y redimensionamiento *offline*, es decir, las imágenes transformadas son guardadas en el disco duro. Por otro lado, las rotaciones aleatorias son efectuadas *online*, en la medida en que la red va entrenándose.

Además de la imagen, el modelo toma como entrada los metadatos correspondientes a esta. En la Tabla 4.2 se enumeran los metadatos que son utilizados por el modelo. Todos son

<sup>2</sup>La aumentación de datos es un procedimiento usual en el entrenamiento de redes neuronales, que trata de generar nuevas muestras a partir de un conjunto de datos.

normalizados y escalados, restando el valor mínimo y dividiendo por el máximo (escalamiento *min-max*).

Metadatos	Detalles
<b>Tamaño de <i>bounding box</i> e imagen</b>	Ancho y alto del <i>bounding box</i> y la imagen. Fracción del alto y ancho de la imagen utilizada por el <i>bounding box</i> .
<b>Coordenadas UTM</b>	Latitud y longitud en el sistema de coordenadas UTM. Se normalizan utilizando los posibles valores: 20 de latitud, 60 de longitud.
<b>Timestamp</b>	Año, mes, día, hora, minuto, segundo y día de la semana. Todos se encuentran en sistema UTC.
<b>Tamaño equivalente del píxel (GSD)</b>	Distancia promedio entre los centros de los píxeles de la imagen a nivel de suelo.
<b>Ángulos</b>	<i>Off-nadir</i> : entre el centro de la imagen y el punto terrestre justo bajo el satélite, desde el satélite. Azimuta objetivo: entre el norte de la imagen y el norte geográfico, desde el centro de la imagen. Azimuta solar: entre el sol y el norte geográfico, desde el centro de la imagen. Elevación solar: entre el plano de la imagen y el sol, desde el centro de la imagen.

Tabla 4.2: Metadatos utilizados como entrada del modelo.

#### 4.3.4. Métrica de desempeño

La clasificación efectuada por el modelo es entrenada a nivel de imágenes. Sin embargo esta es evaluada a nivel de POI, esto es: dado un conjunto de imágenes que corresponden a un mismo POI, se suman las probabilidades de pertenencia a cada una de las 63 clases y se asigna la clase que presente la mayor suma de probabilidades.

En lo que respecta al cálculo de la puntuación obtenida en el desafío FMoW, esta corresponde al promedio ponderado de los *f1-score* de cada clase, multiplicado por 1.000.000. De aquí en adelante, esta se denota ***F-score***. La ponderación asignada a cada clase es determinada por su pertenencia a los siguientes conjuntos [13]:

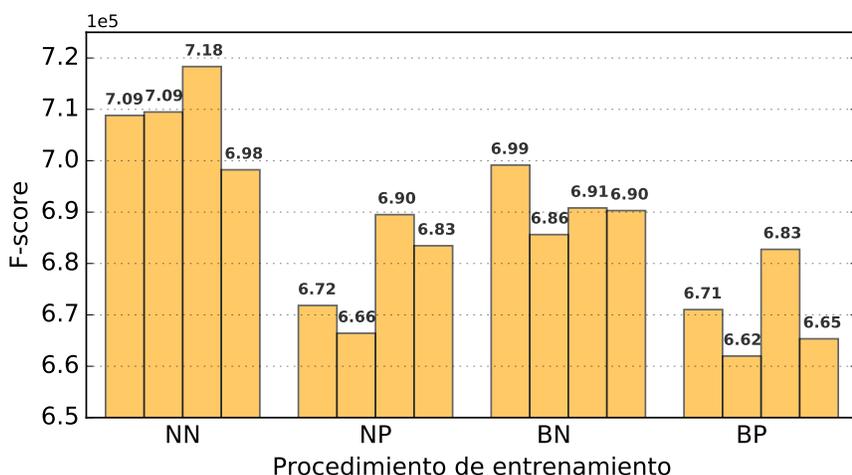
- **Ignorado (peso 0.0)**: *detección falsa*.
- **Bajo impacto (peso 0.6)**: *parque eólico, apertura de túnel, granja solar, central nuclear, instalación militar, campo de cultivo, aeropuerto, camino inundado, escombros, vivienda unifamiliar*.
- **Alto impacto (peso 1.4)**: *punto de control fronterizo, sitio en construcción, institución educativa, fábrica o central eléctrica, estación de bomberos, estación de policía, estación de servicio, chimenea, torre, puente de carretera*.
- **Impacto medio (peso 1.0)**: todas las categorías restantes.

Cabe mencionar que si bien la clase de *detección falsa* tiene peso nulo, el impacto de clasificar incorrectamente una imagen de esta clase se ve reflejado en un menor rendimiento en la clase a la cual dicha imagen es asignada.

## 4.4. Resultados

### 4.4.1. Comparación de los procedimientos de entrenamiento

En la Figura 4.8 se muestran los resultados de los modelos para cada uno de los procedimientos de entrenamiento (cf. Sección 4.3.2) a nivel de POI. Cada una de las barras corresponde a un modelo entrenado para una división diferente de los subconjuntos de entrenamiento, validación y prueba. Se observa que el procedimiento que obtuvo mejores resultados es el no balanceado, no ponderado (NN). Se adjunta además la Tabla 4.3, en donde se muestran los tiempos de entrenamiento para cada uno de los modelos entrenados.



modelo	tiempo [horas]
NN	30,6
NP	32,7
BN	28,8
BP	25,7

Tabla 4.3: Tiempo promedio de entrenamiento por procedimiento de entrenamiento.

Figura 4.8:  $F$ -score en el conjunto de prueba por procedimiento de entrenamiento.

Con el fin de apreciar el progreso de diferentes métricas durante el proceso de entrenamiento se adjunta la Figura 4.9. En esta, se muestra solamente el proceso de entrenamiento para un modelo del procedimiento NN, ya que el comportamiento de los demás modelos y procedimientos es análogo. Se grafica con una línea punteada vertical la época en la que se alcanza el máximo  $F$ -score del entrenamiento, lo que corresponde al modelo seleccionado. Las tres épocas posteriores corresponden a las épocas en que se siguió el entrenamiento en búsqueda de un mejor  $F$ -score, que finalmente no es alcanzado.

La matriz de confusión normalizada obtenida por el modelo con el más alto  $F$ -score (NN-1) se muestra en la Figura 4.10. Si bien la diagonal concentra la mayor cantidad de los casos, existen algunas clases en las que el valor en la diagonal es bastante cercano o incluso igual al alcanzado en alguna de las coordenadas extradiagonales. Para identificar aquellas clases, en

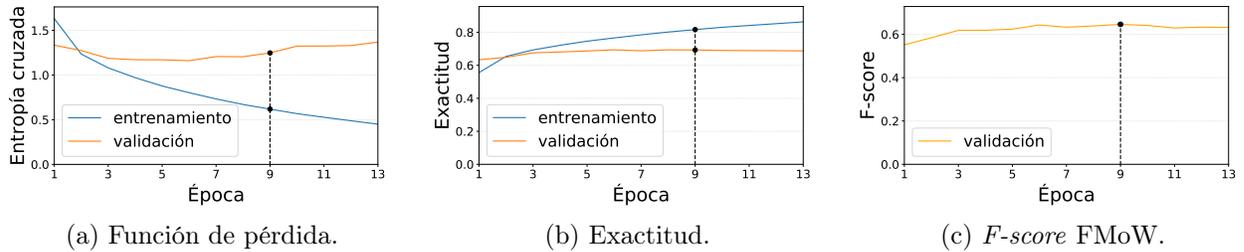


Figura 4.9: Variación de métricas durante el proceso de entrenamiento.

la Figura 4.11 se muestra el  $f1$ -score obtenido en cada clase, ordenadas de menor a mayor.

#### 4.4.2. Tabla de posiciones FMoW

Para evaluar el rendimiento del mejor de los procedimientos de entrenamiento (NN), los resultados de este son comparados con aquellos obtenidos por los concursantes del desafío FMoW. En la Figura 4.12, se observa que el mejor modelo obtenido en esta memoria estaría posicionado lugar veinteaño. Tomando en cuenta sólo los concursantes que obtuvieron un resultado mayor que cero (58), esta solución se encuentra en el tercio superior de los modelos presentados en la competencia.

Es necesario reiterar que si bien el experimento se configuró de manera que las condiciones fueran lo más similares a las del desafío FMoW, las comparaciones son aproximadas ya que los conjuntos de entrenamiento, validación y test no son los mismos (ver Sección 4.2).

#### 4.4.3. Clasificación de espacios abiertos

Las clases que caben dentro de la clasificación de espacios abiertos de potencial uso público son: parques e instalaciones recreativas. Los diferentes procedimientos de entrenamiento son comparados en la Tabla 4.4. La matriz de confusión del modelo que obtuvo el  $f1$ -score más elevado se muestra en la Figura 4.13.



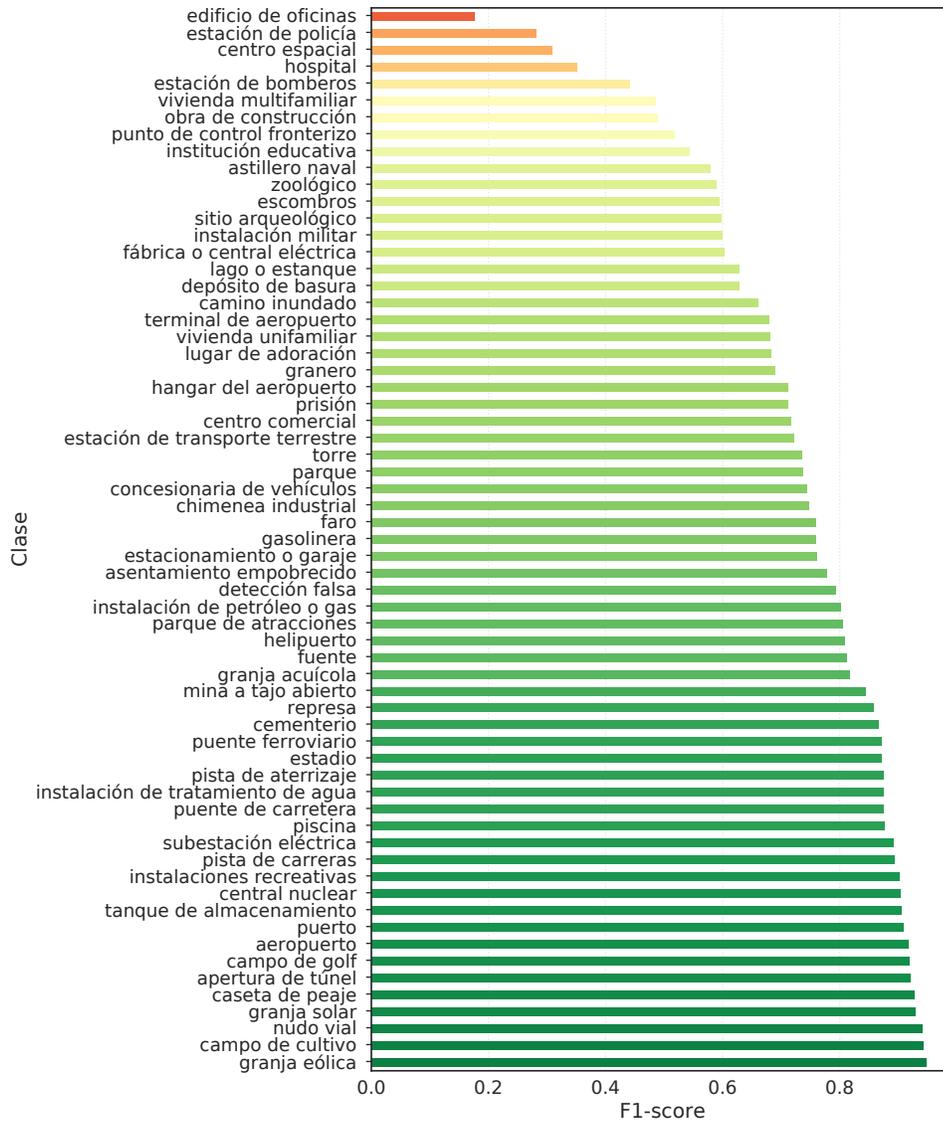


Figura 4.11:  $F1$ -score por clase a nivel de POI.

modelo	precisión	exhaustividad	$f1$ -score
NN	<b>0,93</b>	0,88	<b>0,90</b>
NP	0,82	<b>0,92</b>	0,87
BN	0,85	0,91	0,88
BP	0,86	0,88	0,87

Tabla 4.4: Desempeño promedio en la clasificación de espacios abiertos.

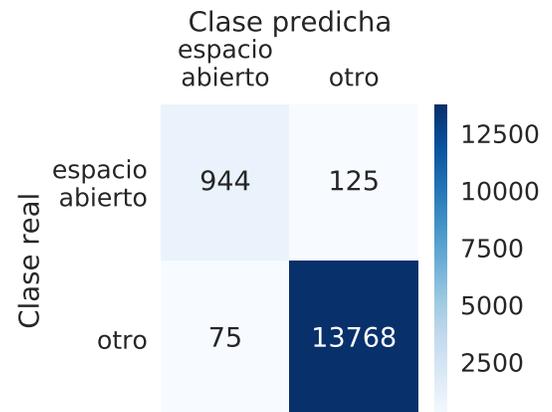


Figura 4.13: Matriz de confusión para el modelo con mejor  $f1$ -score en la clasificación de espacios abiertos.

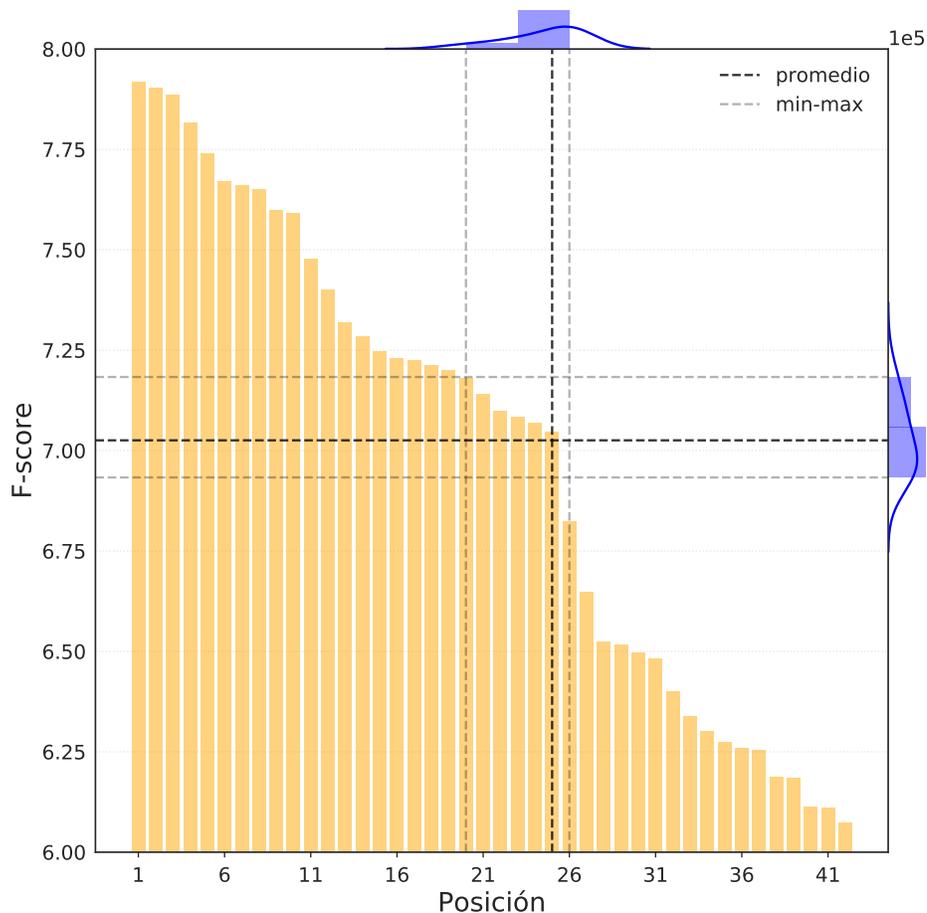


Figura 4.12: Posicionamiento del mejor procedimiento de entrenamiento entre los concursantes del desafío FMoW.

## 4.5. Análisis y discusión

Como se puede observar en la matriz de confusión (Figura 4.10), existen algunas clases en las que ciertos componentes extradiagonales alcanzan un valor cercano o igual a los de la diagonal. Entre ellos se puede distinguir un grupo en que la similitud de las clases es alta. Este es el caso de la clase *astillero naval* que en el 45% de las oportunidades es clasificada como *puerto*, o de la *vivienda multifamiliar* clasificada un 20% de las veces como *vivienda unifamiliar*. En dichos casos se puede apelar a una baja separabilidad de las clases en cuestión. Por otro lado, existe también un grupo de clases en las que la similitud es menos evidente. Por ejemplo la clase *centro espacial* con un 22% de sus imágenes clasificadas como *parque de atracciones*, u *hospital* con un 20% como *institución educativa*. En estos casos, se sugiere que el error está más relacionado con el modelo que con la separabilidad de las clases.

### 4.5.1. Impacto del desequilibrio de clases

La comparación entre los diferentes procedimientos de entrenamiento arroja resultados que sugieren que el desequilibrio de clases no puede ser enfrentado con el sobremuestreo ni la ponderación de muestras, ya que los mejores resultados son obtenidos para el modelo sin ponderación de muestras y sin balance de clases.

Si se analizan el *f1-score* obtenido por clase y el soporte de cada clase en el conjunto de entrenamiento (cf. Figura 4.14), se puede observar que las clases con menor soporte no son necesariamente las que presentan los peores resultados.

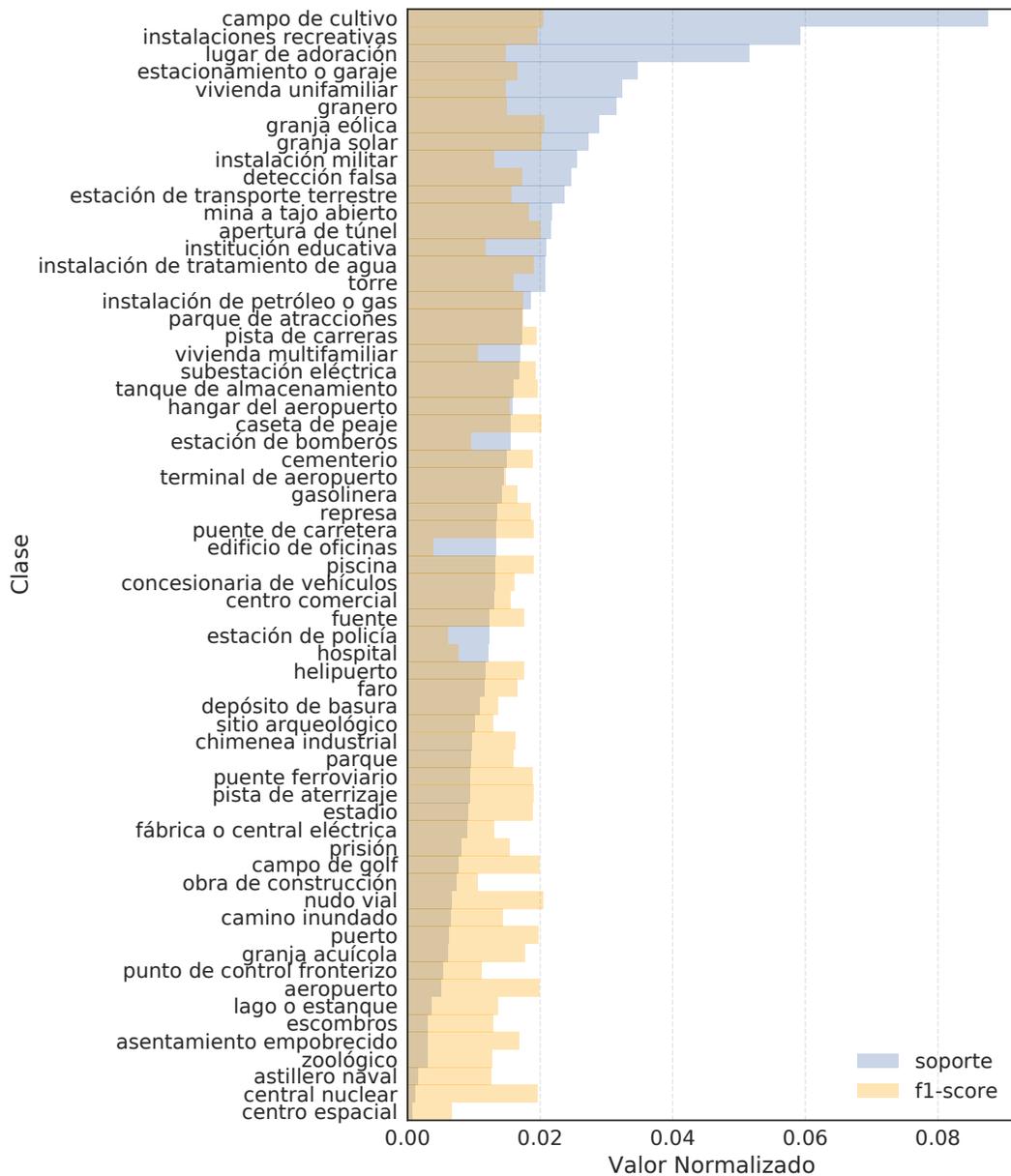


Figura 4.14: *F1-score* y soporte por clase a nivel de POI.

Con el fin de analizar cualitativamente aquellas clases en las que se obtiene el peor desem-

peño, en la Figura 4.15 se muestran dos casos en los que el POI fue mal clasificado. La clase asignada en ambos casos corresponde a la clase que presenta la mayor proporción de los falsos negativos (cf. 2.2.3) de la clase en cuestión.



Figura 4.15: Ejemplos cualitativos de mala clasificación en las clases con *f1-score* más bajo.

#### 4.5.2. Comparación con los concursantes de FMoW

Aún si los resultados alcanzados por el modelo obtenido en este trabajo están un 10-12% por debajo de los resultados ganadores del desafío, cabe destacar que los tres primeros lugares corresponden a *ensembles*<sup>3</sup> de redes neuronales, por lo que la diferencia en desempeño es razonable, pues se está comparando el desempeño de una red neuronal con el obtenido por un conjunto de estas. Si bien esto no implica que el modelo propuesto sea más interpretable (ya que las redes empleadas son de igual complejidad), la solución propuesta es más eficiente computacionalmente, ya que se trata de un solo modelo de RNC.

Por otro lado, el haber obtenido un modelo que se posicione en el 35% superior de la tabla de posiciones, podría ser explicado por el enriquecimiento de los *bounding box* y las técnicas de aumentación de datos empleados (cf. Sección 4.3.3), ya que los otros aspectos de la solución concebida no deberían ser muy diferentes de los empleados por gran parte de los demás concursantes. En otras palabras, aquellos elementos son los que se creen diferenciadores de la solución propuesta.

#### 4.5.3. Factibilidad de la detección automática de espacios abiertos

El desempeño obtenido en las clases que pueden ser categorizadas como parte de los espacios abiertos urbanos, es alto en comparación a aquellos obtenidos en las otras clases. Si

<sup>3</sup>En AM se habla de un *ensemble* de modelos cuando se utiliza un conjunto de modelos para una tarea específica.

bien, esto no permite asegurar que es factible generar un modelo de detección<sup>4</sup> automática de espacios abiertos urbanos, el tener un modelo con buen desempeño en la clasificación del objeto que se busca detectar, es un pilar fundamental para la construcción de aquel.

Cabe destacar que si se analizan los resultados obtenidos por las clases que forman la superclase de espacio abierto en el conjunto FMoW por separado, los resultados obtenidos en la clase parque distan bastante de los obtenidos por la superclase. Esto proviene del desbalance de clases entre *instalaciones recreativas* (6.1%) y *parques* (0.9%).

Finalmente, otro elemento a destacar es la superioridad del modelo NP por sobre el NN en términos de exhaustividad. Si bien, cuatro realizaciones del modelo son probablemente insuficientes para poder aseverar la significancia estadística de esta diferencia, hay elementos para creer que este procedimiento puede ser el más indicado si se quiere maximizar la exhaustividad del modelo en la superclase de espacio abierto. Por un lado, en términos de la función de costo, la clase parque que es la más deficiente gana importancia, lo que debería elevar el desempeño del modelo sobre esta. Por otro lado la clase instalaciones recreativas, pareciera ser fácilmente separable de las demás, ya que está en el cuarto superior de las clases con mejor desempeño *f1-score* (cf. Figura 4.11), por lo que incluso si pierde influencia sobre el funcional de costo, podría mantener su buen desempeño.

---

<sup>4</sup>En visión computacional, se distingue la clasificación de imágenes de la detección de objetos, ya que en el primero se entrega una etiqueta por cada imagen y en el segundo, se busca la etiqueta y además la localización del objeto de interés en la imagen. Esta localización generalmente se entrega como un *bounding box*.

# Capítulo 5

## Conclusiones y recomendaciones

### 5.1. Conclusiones

Este trabajo aporta avances en la metodología de los indicadores ODS que aborda, ya sea a nivel de costo de las soluciones o a nivel de evidencia empírica de la factibilidad del uso de imágenes satelitales para el cálculo de estos.

En lo que concierne a la problemática CSC, se diseña una solución de código libre, con imágenes satelitales y etiquetas de libre acceso, evidenciando así la posibilidad de estimar el componente de superficie construida del indicador 11.3.1, sin importar el nivel de ingreso del país que busque calcularlo. Adicionalmente, se presenta un modelo de MSV que es significativamente mejor que los demás, lo que permite recomendar el uso de dicho algoritmo para enfrentar esta problemática.

En cuanto al problema CEUS, si se supone que los resultados obtenidos en la configuración del experimento son extrapolables a los del desafío (cf. Sección 4.4.2), este trabajo presenta un modelo de red convolucional con resultados satisfactorios en comparación a los demás concursantes del desafío FMoW, al ubicarse en el tercio superior de las soluciones presentadas. Además, se constata que ni el balanceo de clases, ni la ponderación de muestras, tiene una influencia positiva sobre los resultados de la clasificación para el modelo propuesto en términos de desempeño general, pero la ponderación de muestras si puede tener un impacto positivo cuando se trata de la superclase de espacios abiertos. Finalmente, se muestra también que al menos a nivel de las imágenes presentes en el conjunto de datos FMoW, la clasificación de imágenes dentro de la superclase espacio abierto, definida en la memoria, es factible y presenta buenos resultados (*f1-score* de 0.9).

### 5.2. Recomendaciones

Como es mencionado en la Sección 3.5, uno de los ejes de mejoramiento más evidentes de la solución presentada al problema CSC, es la búsqueda de un tamaño óptimo del conjunto

de entrenamiento, pues puede que algunas de las clases estén subrepresentadas en el conjunto de entrenamiento de los modelos propuestos.

Adicionalmente, para dicha problemática, con el fin de encontrar el equilibrio óptimo entre precisión y exhaustividad en cada una de las clases, puede ser interesante buscar la cota óptima mediante el análisis de la curva precisión-exhaustividad.

Por último, para el mismo problema, puede ser interesante validar los resultados obtenidos por los modelos propuestos en el conjunto de datos [17] y así poder evaluarlos comparativamente con respecto a los modelos presentes en la comparación efectuada en [18].

En lo que concierne al problema CEUS, una posible extensión del trabajo realizado es la utilización de estructuras que exploten las relaciones temporales de las imágenes. Dado que en múltiples ocasiones existen varias imágenes que describen a un mismo POI, es posible extender el modelo presentado con algún algoritmo que explote la dimensión temporal. En el trabajo [7], se propone una red neuronal recurrente, denominada *Long short-term memory* (LSTM) que usa las características extraídas por el modelo convolucional y es capaz de aumentar en un 2% el desempeño promedio en términos de *f1-score*. Un modelo que se recomienda probar es el de LSTM bidireccional [21], ya que permite extraer información en ambas direcciones temporales.

# Bibliografía

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, and Yuan Yu y Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software disponible en tensorflow.org.
- [2] National Aeronautics and Space Administration (NASA). Landsat 8. [en línea] <<https://landsat.usgs.gov/landsat-8>>, 2008. [consulta: 24 junio 2017].
- [3] Shlomo Angel, Alejandro M. Blei, Daniel L. Civco, and Jason Parent. *Atlas of urban expansion*. Lincoln Institute of Land Policy Cambridge, MA, 2012.
- [4] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.
- [5] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.
- [6] François Chollet et al. Keras. [en línea] <<https://keras.io>>, 2015. [consulta: 3 noviembre 2017].
- [7] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. *arXiv preprint arXiv:1711.07846*, 2017.
- [8] Emilio Chuvieco. *Fundamentals of satellite remote sensing*. CRC press, Florida, USA, 2nd edition, 2009.
- [9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [10] Organización de las Naciones Unidas. The millennium development goals report 2015.

- 2015.
- [11] Organización de las Naciones Unidas. Transforming our world: The 2030 agenda for sustainable development. 2015.
  - [12] Programa de Naciones Unidas para los Asentamientos Humanos. *SDG Goal 11 Monitoring Framework*, volume 1. Programa de Naciones Unidas para los Asentamientos Humanos, 2016.
  - [13] Intelligence Advanced Research Projects Activity (IARPA) Oficina del Director de Inteligencia Nacional de Estados Unidos. Functional map of the world. [en línea] <<https://www.iarpa.gov/challenges/fmow.html>>, 2017. [consulta: 20 octubre 2017].
  - [14] Naciones Unidas División de Estadísticas, Departamento de Asuntos Económicos y Sociales. IAEG-SDGs — SDG indicators. [en línea] <<https://unstats.un.org/sdgs/iaeg-sdgs/tier-classification/>>, 2016. [consulta: 25 junio 2017].
  - [15] Angel et al. Atlas of urban expansion. [en línea] <<http://www.atlasofurbanexpansion.org/>>, 2016. [consulta: 27 mayo 2017].
  - [16] Angel et al. *Atlas of Urban Expansion—2016 Edition, Volume 1: Areas and Densities*. New York University, UN-Habitat and Lincoln Institute of Land Policy, 2016.
  - [17] Eurostat. *LUCAS the EU’s land use and land cover survey : 2017 edition*. Publications Office, Luxembourg, 2017.
  - [18] Aneta Florczyk. *Operating procedure for the production of the global human settlement layer from Landsat data of the epochs 1975, 1990, 2000, and 2014*. Publications Office, Luxembourg, 2016.
  - [19] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
  - [20] Ran Goldblatt, Wei You, Gordon Hanson, and Amit K. Khandelwal. Detecting the boundaries of urban areas in india: A dataset for pixel-based image classification in google earth engine. *Remote Sensing*, 8(8), 2016.
  - [21] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.
  - [22] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
  - [23] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens vanderMaaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.
  - [24] Jack Hughes. Topcoder. [en línea] <<https://www.topcoder.com>>, 2001. [consulta: 22

octubre 2017].

- [25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [26] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, New York, NY, 2013.
- [27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [29] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9:2579–2605, 2008.
- [31] Juraj Murcko. Object-based classification for estimation of built-up density within urban environment. Master’s thesis, Dresden University of Technology, 2017.
- [32] Christopher Olah. Colah’s blog. [en línea] <<http://colah.github.io/>>, 2013. [consulta: 13 diciembre 2017].
- [33] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.
- [34] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.
- [35] Chris Padwick, Michael Deskevich, Fabio Pacifici, and Scott Smallwood. Worldview-2 pan-sharpening. In *Proceedings of the ASPRS 2010 Annual Conference, San Diego, CA, USA*, volume 2630, 2010.
- [36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [37] Max Planck. On the law of distribution of energy in the normal spectrum. *Annalen der physik*, 4(553):1, 1901.
- [38] Gerrit Polder, Erik Pekkeriet, and Marco Snickers. A spectral imaging system for detection of botrytis in greenhouses. In *EFITA-WCCA-CIGR Conference on Sustainable Agriculture through ICT Innovation*, 07 2013.

- [39] David Potere, Annemarie Schneider, Shlomo Angel, and Daniel L. Civco. Mapping urban areas on a global scale: which of the eight maps now available is more accurate? *International Journal of Remote Sensing*, 30(24):6531–6558, 2009.
- [40] Christian Robert. Machine learning, a probabilistic perspective, 2014.
- [41] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
- [42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
- [44] Lipo Wang. *Support vector machines: theory and applications*, volume 177. Springer Science & Business Media, 2005.
- [45] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.
- [46] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279. ACM, 2010.
- [47] Zhe Zhu, Shixiong Wang, and Curtis E Woodcock. Improvement and expansion of the fmask algorithm: cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images. *Remote Sensing of Environment*, 159:269–277, 2015.