MODELOS DE INTEGRACIÓN DE INFORMACIÓN MULTIMODAL EN CNN PARA EL ANÁLISIS DE IMAGEN SATELITAL EN ENTORNOS URBANOS

Diego Alexander Rueda Plata

Trabajo de Grado para optar al título de Magíster en Matemática Aplicada

Director

Ph.D en Ingeniería Informática, Raúl Ramos Pollán

Co-director

Ph.D en Física, Luis Alberto Núñez

Universidad Industrial de Santander

Facultad de Ciencias

Escuela de Física

Bucaramanga

2020

Dedicatoria

A mis Padres y mi Familia.

Agradecimientos

En primer lugar, agradezco a mi familia por el apoyo económico y moral que tuvieron conmigo durante el desarrollo de mi maestría.

Al profesor Raúl Ramos Pollán de la Universidad de Antioquia por confiar en mis capacidades y guiarme durante el desarrollo de este proyecto.

A los profesores y compañeros del grupo CAGE en la Universidad Industrial de Santander por su asesoría constante y desinteresada, y por facilitar los recursos de computo necesarios para la realización de este proyecto.

A los investigadores Juan Carlos Duque, Ana Beatriz Acevedo y Daniela González de la Universidad EAFIT por facilitar los datos usados en el proyecto y sus valiosos aportes en la construcción de este trabajo.

Tabla de Contenido

Lista de Figuras

Lista de Tablas

Lista de Apéndices

pág.

Glosario

Modalidad es la forma particular en que la información está codificada y percibida por seres humanos.

Sensado remoto es el proceso de detectar y monitorizar las características físicas de un área midiendo la radiación emitida y reflejada a distancia.

Machine Learning son conjuntos de métodos de análisis de datos que automatizan la construcción de modelos analíticos que pueden aprender de los datos a identificar patrones y tomar decisiones con una mínima intervención humana.

Deep Learning es un subconjunto de métodos de machine learning, que no requieren de la intervención humana en la etapa de extracción de características, estas son aprendidas directamente de los datos y usadas para identificar patrones en el conjunto de datos.

Arquitectura de Red En el contexto de deep learning, se refiere a la organización jerárquica de las capas de neuronas artificiales interconectadas.

Dataset colección de elementos de información separados que pueden ser manipulados por un computador.

Epoch es una presentación completa del dataset al modelo de deep learning.

Parámetros son las propiedades del conjunto de datos que son aprendidas durante el proceso de entrenamiento.

Hiper-parámetros son las propiedas que definen el proceso de entrenamiento, y son definidas antes de iniciar este proceso. Controlan el tiempo de entrenamiento, la tasa de aprendizaje o el número de imágenes que la arquitectura de red recibe en cada paso, etc.

Resumen

Título: Modelos de integración de información multimodal en CNN para el análisis de imagen satelital en entornos urbanos. [*]

Autor: Diego Alexander Rueda Plata [**]

Palabras Clave: Redes convolucionales, datos multimodales, deep learning, inteligencia artificial.

Descripción: El aprendizaje multimodal ofrece la posibilidad de capturar correspondencias entre modalidades y obtener una mayor generalización de la problemática analizada. Una modalidad se refiere a la forma en que los datos son percibidos, audio, imágenes o datos estructurados. Este proyecto aborda el análisis de la inyección de datos multimodales en el entrenamiento de arquitecturas pre-entrenadas de redes neuronales para la identificación de tipologías estructurales en edificios residenciales. Se utiliza un conjunto de imágenes obtenidas por exploración remota mediante StreetView en el área urbana de Medellín, cada una de las entradas en el dataset incluye el registro del número de pisos, las coordenadas de latitud y longitud y el estrato del edificio. Usando sólo imágenes se realiza un entrenamiento de diferentes arquitecturas convolucionales para establecer una base de resultados. Seguidamente, un perceptrón multicapa recibe como entrada datos en otra modalidad existentes por cada edificio, y se entrena simultáneamente con las mismas arquitecturas definidas, los vectores finales de características de estas dos redes son concatenados para producir una salida conjunta. Posteriormente, se usan métricas de clasificación para modelos de aprendizaje de máquinas y demostramos el impacto positivo de incluir información bajo otra modalidad, particularmente en clases poco representadas en el dataset, sin embargo se muestran las limitaciones y un tiempo de entrenamiento mayor bajo redes multimodales.

---

[*] Trabajo de grado de Maestría

[**] Facultad de Ciencias. Escuela de Física. Director: Raúl Ramos Pollán, Doctorado en Ingeniería Informática.

Abstract

Title: Multimodal information integration models on CNN for satelital image analysis in urban environments. [*]

Author: Diego Alexander Rueda Plata [**]

Keywords: Convolutional networks, multimodal data, deep learning, machine learning.

Description: Multimodal learning offers the possibility of capturing relations among modalities and obtain a higher generalization of the analyzed problem using machine learning. A modality refers to how data is presented, audio, images or structured data. This project studies the impact of injecting multimodal data on pre-trained neural network architectures to identify structural typologies of residential buildings . An annotated dataset of the metropolitan area of Medellin has been used obtained from StreetView, each one of the buildings includes data on the number of stories, geolocalization (latitude and longitude pairs) and socio-economic stratification of the building. Initially, we train convolutional networks using only the images from the dataset, in order to establish a baseline of results. Following, a multilayer perceptron is created using as input the information from each building, and we train simultaneously with the same convolutional architectures, the final feature vectors are concatenated to produce a single output. We use machine learning classification metrics to measure the positive impact of including data from a different modality, particularly for underrepresented classes in the dataset, while showcasing the limitations and increased training time from multimodal networks.

---

[*] Master Thesis

[**] Faculty of Sciences. School of Physics. Advisor: Raúl Ramos Pollán, Ph.D in Informatics Engineering.

Introducción

Para el 2050, se espera que dos tercios de la población mundial vivan en áreas urbanas, un incremento del 55 % según un informe presentado en el 2018, de igual manera en latino-América donde a la fecha del informe el 80.7 % de la población reside en ciudades, se observa un crecimiento constante en la concentración urbana (United Nations and Affairs.). El rápido crecimiento es un desafío para la planeación urbana y dificulta la administración eficiente de recursos, así como la toma de decisiones. El manejo de movilidad, riesgo estructural, uso de la tierra y densidad urbana, son algunas de las áreas de planeación donde existe dificultad para la obtención de datos que expresan el rápido crecimiento observado en las ciudades.

Las nuevas tecnologías ofrecen soluciones de datos que permiten seguir el paso al crecimiento urbano, mediante estas podemos obtener imágenes satelitales y aéreas de toda la infraestructura vial de la ciudad o podemos obtener imágenes geoespaciales a nivel de calle para analizar la estructura de múltiples edificios. A partir de estas imágenes es posible emplear técnicas de aprendizaje computacional, en particular, redes neuronales convolucionales (CNN) especialmente construidas para aprender características de las imágenes y entrenarlas para obtener información relevante y soportar las decisiones de un experto en las diferentes problemáticas urbanas.

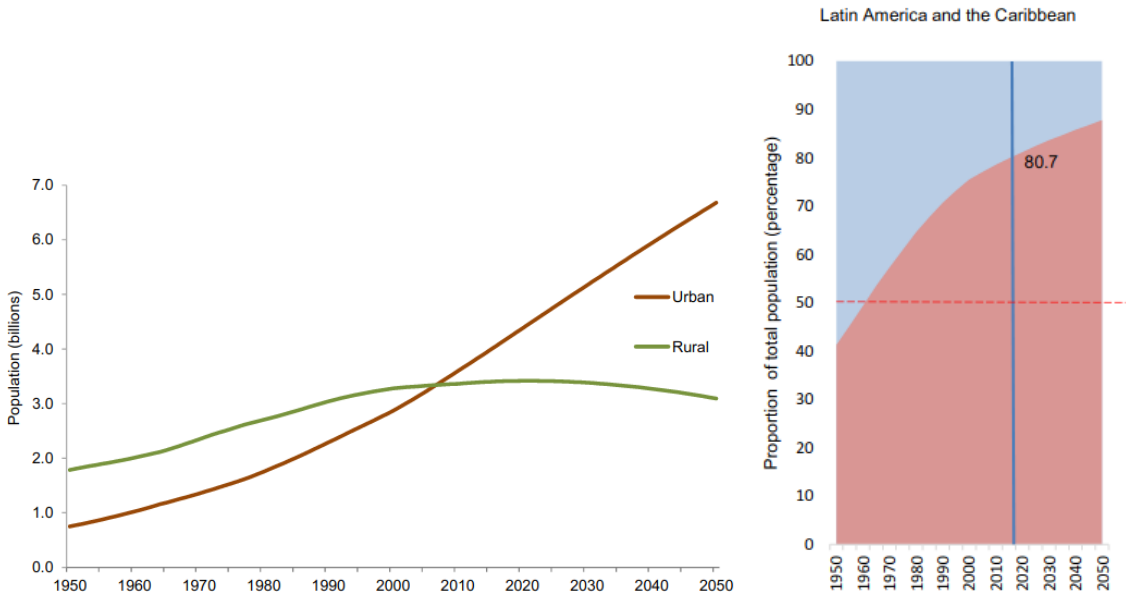


Figura 1. Estimación de crecimiento urbano a nivel mundial y en Latinoamérica. (United Nations and Affairs.)

Adicional a las imágenes obtenidas, dependiendo de la problemática existen diferentes datos que pueden ser adquiridos y utilizados para complementar las interpretaciones de la red neuronal. Estas características son únicas para cada situación, por ejemplo, estadísticas de accidentes registrados en diferentes zonas de la ciudad. El trabajo propuesto busca aprovechar la gran cantidad de datos existentes y combinar los datos con información tabular mediante técnicas de aprendizaje computacional y análisis de datos que permita inferir patrones y crear modelos multimodales que puedan ser usados para analizar problemáticas urbanas.

Definición del problema

Las redes neuronales convolucionales (CNN) han sido aplicadas con éxito en la clasificación de imágenes siendo entrenadas sobre grandes conjuntos de datos, como Imagenet (16). No obstante, estos grandes repositorios no son sencillos de construir, el consumo de tiempo en su etiquetado y obtención de imágenes es costoso. Es importante resaltar que en el entrenamiento de una CNN se considera una sola modalidad (visual), y aunque existan técnicas de transferencia de aprendizaje que permiten mejorar la precisión de una CNN sobre menores conjuntos de datos, no siempre son suficientes para resolver problemáticas complejas y obtener un modelo que presente mejor generalización en el problema de clasificación.

El aprendizaje multimodal ofrece la posibilidad de capturar correspondencias entre modalidades y obtener un mayor entendimiento de la problemática analizada (8). El enfoque en este proyecto está en diseñar e implementar arquitecturas de redes neuronales que permitan combinar imágenes con datos bajo otra modalidad sobre una problemática urbana que nos permita trabajar sobre diferentes modalidades.

1. Objetivos

Objetivo general

Desarrollar modelos de redes neuronales multimodales que permitan el análisis de imágenes para el estudio de entornos urbanos en una problemática definida.

Objetivos específicos

- Definir una problemática urbana a tratar en función de la dificultad y la utilidad de la misma, considerando la disponibilidad de datos multimodales.
- Diseñar e implementar arquitecturas de redes neuronales multimodales para clasificación que permitan facilitar la solución de la problemática urbana seleccionada.
- Evaluar el impacto de la información multimodal en el desempeño y entrenamiento de las redes neuronales convolucionales.

2. Marco Teórico

Para comprender mejor las tecnologías de redes neuronales convolucionales, se presenta un marco teórico enfocado en las técnicas y teoría necesarias para entenderlas.

2.1. Redes neuronales

2.1.1. Neurona artificial.

Las redes neuronales fueron llamadas originalmente redes neuronales artificiales, porque estas fueron construidas para imitar la función neuronal del cerebro humano. Los primeros avances en este campo se remontan a 1957 con el Perceptrón desarrollado por Frank Rosenblatt (49).

Aunque la inspiración biológica es notable, sería erróneo sobre-enfatizar la conexión entre neuronas artificiales y neuronas biológicas. La investigación en redes neuronales artificiales se encuentra guiada por desarrollos en ingeniería y matemática en lugar de biología (25).

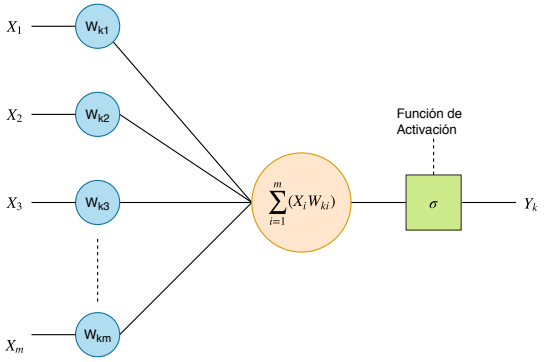


Figura 2. Una neurona artificial.

La neurona k recibe m parámetros x_i. La neurona también tiene m parámetros de peso W_{ki}.

Los parámetros de peso a menudo incluyen un término de sesgo que tiene un valor fijo de entrada de 1. Las entradas y los pesos se combinan linealmente y son sumados. La suma luego es pasada a una función de activación σ que produce una salida y_k de la neurona como se ve en la figura 2.

$$y_k = \sigma\left(\sum_{i=1}^{m} W_{ki} X_i\right) \tag{1}$$

La neurona es entrenada seleccionando los pesos para producir la salida deseada en cada entrada.

2.1.2. Redes neuronales multi-capa.

Una red neuronal es una combinación de neuronas artificiales. Las neuronas están típicamente agrupadas en capas. En una red completamente conectada y multi-capa como la mostrada en la figura 3, cada salida de una capa de neuronas es usada como entrada para cada neurona de la siguiente capa. Por lo tanto, algunas capas procesan los datos originales de entrada, mientras otras usan los datos recibidos de otras neuronas. Cada neurona tiene un número de pesos igual al número de neuronas en la capa anterior.

Una red multi-capa típicamente incluye tres tipos de capas: una capa de entrada, una o más capas ocultas y una capa de salida. La capa de entrada usualmente pasa los datos sin modificarlos y la mayor parte del trabajo ocurre en las capas ocultas. La capa de salida convierte las activaciones de la capa oculta anterior a una salida como una clasificación.

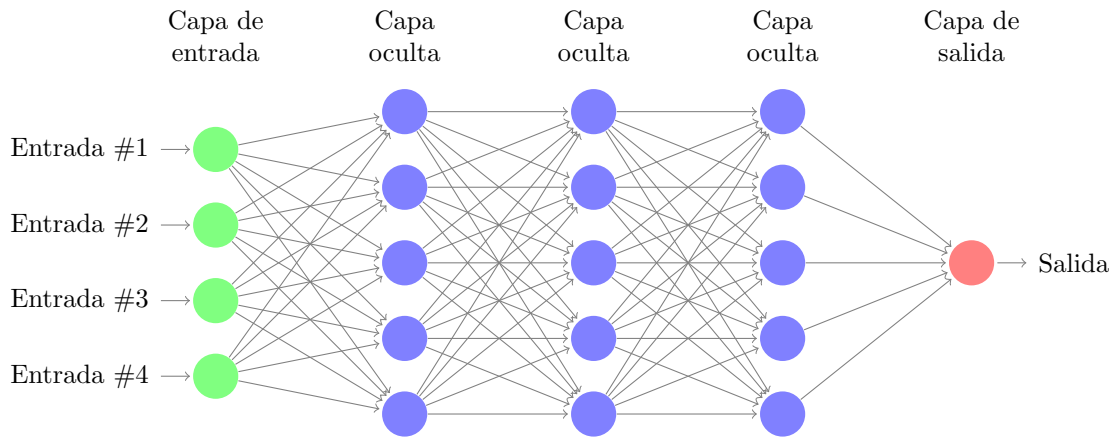


Figura 3. Red neuronal multi-capa

2.1.3. Back-propagation.

El algoritmo de Back-propagation fue uno de los primeros métodos en demostrar que las redes neuronales artificiales pueden aprender buenas representaciones internas (37). Este es un algoritmo para el aprendizaje supervisado de redes neuronales artificiales usando gradiente descendiente. Este algoritmo tiene los siguientes requisitos antes de poder usarse:

1. **Un Dataset** que consista de pares entrada-salida (x_i, y_i), donde x_i es la entrada, y_i es la salida deseada de la red para esta entrada. El conjunto de pares entrada-salida de tamaño N se denota como $X = \{(x_i, y_i), ..., (x_N, y_N)\}$

2. **Una Red Neuronal Multi-capa,** cuyos parámetros en conjunto se denotan como θ. En backpropagation, los parámetros de interés primario son w_{ij}^k, el peso w entre el nodo j en la capa l_k y el nodo i en la capa l_{k-1}, y b_i^k para el sesgo del nodo i en la capa l_k. No hay conexiones entre nodos de la misma capa y las capas están completamente conectadas.

3. **Una Función de Error,** $E(X,\theta)$ la cual define el error entre la salida deseada y_i y la salida obtenida y_i de la red neuronal para una entrada x_i de un conjunto de pares entrada-salida $(x_i, y_i \in X)$ y un valor particular de los parámetros θ.

El algoritmo de backpropagation sigue los siguientes pasos, asumiendo una tasa de aprendizaje adecuada α y una inicializacion aleatoria de los parámetros w_{ij}^k:

1. **Calcular el paso hacia adelante en la red** para cada par de entradas $(\overrightarrow{x_d}, y_d)$ y guardar la salida de la red $\hat{y_d}$, la salida y activación a_j^k y o_j^k para cada nodo j en la capa k, desde la capa de entrada 0 hasta la capa de salida m.

2. **Calcular el paso hacia atrás en la red** para cada par de entradas $(\overrightarrow{x_d}, y_d)$ y guardar los resultados $\frac{\delta E_d}{\delta w_{ij}^k}$ para cada peso w_{ij}^k conectando cada nodo i en la capa $k-1$ al nodo j en la capa k, partiendo desde la capa de salida m hasta la capa de entrada.

   a) Evaluar el término de error para la capa final $\delta_1^m = g_o'(a_1^m)(\hat{y_d} - y_d)$ donde g_o' es la derivada de la función de activación para la capa final y a_1^m es el valor de la salida para la capa final.

   b) Propagar hacia atrás los términos de error para las capas ocultas δ_j^k, calculando desde la última capa oculta $k = m-1$ usando la ecuación $\delta_j^k = g'(a_j^k)\sum_{l=1}^{r^k+1} w_{jl}^{k+1}\delta_l^{k+1}$, donde g' es la derivada de la función de activación para la capa oculta.

   c) Evaluar las derivadas parciales del error individual E_d con respecto a w_{ij}^k de la siguiente manera: $\frac{\delta E_d}{\delta w_j^k} = \delta_j^k o_i^{k-1}$

3. **Combinar los gradientes individuales** para cada par de entrada-salida $\frac{\delta E_d}{\delta w_{ij}^k}$ para obtener el gradiente global $\frac{\delta E(X,\theta)}{\delta w_{ij}^k}$ para el conjunto completo de entradas-salidas $X = (\overrightarrow{x_1}, y_1), ..., (\overrightarrow{x_N}, y_N)$ obteniendo el promedio de los gradientes individuales. $\frac{\delta E(X,\theta)}{\delta w_{ij}^k} = \frac{1}{N} \sum_{d=1}^{N} \frac{\delta}{\delta w_{ij}^k} \left(\frac{1}{2} (\hat{y_d} - y_d)^2 \right) = \frac{1}{N} \sum_{d=1}^{N} \frac{\delta E_d}{\delta w_{ij}^k}$

4. **Actualizar los pesos** de acuerdo a la tasa de aprendizaje α y el gradiente total $\frac{\delta E(X,\theta)}{\delta w_{ij}^k}$ en la dirección opuesta al gradiente $\Delta w_{ij}^k = -\alpha \frac{\delta E(X,\theta)}{\delta w_{ij}^k}$

2.1.4. Funciones de activación.

La función de activación σ determina la salida de cada neurona. Es importante seleccionar la función adecuadamente para crear una red neuronal. Las funciones de activación introducen un comportamiento no lineal que permite a la red aprender funciones complejas (25).

Sigmoide.

El sigmoide puede ser considerado una función de paso suavizado y por ende derivable. Este es útil para convertir cualquier valor a probabilidades y puede ser usado para clasificación binaria. Este mapea una entrada al rango entre 0 y 1.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

La derivada de la función sigmoide es la siguiente:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

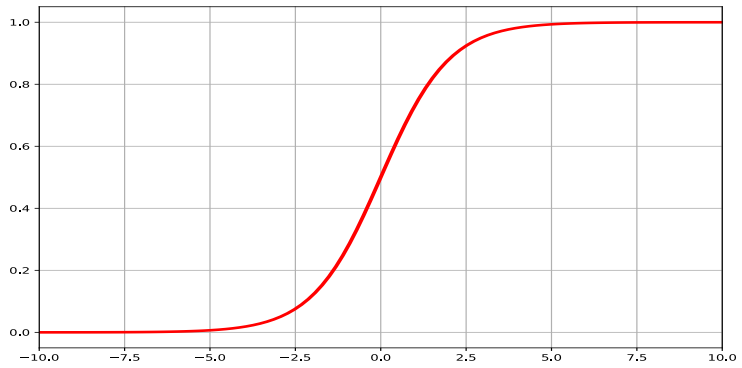


Figura 4. Sigmoide

ReLU.

Unidad lineal rectificada, esta función mapea la entrada x a $max(0,x)$, es decir, los valores negativos serán 0 y los positivos pasaran sin ningún cambio. Esto causa que algunas neuronas se vuelvan inactivas y no se disparen. Debido a que, ReLU no se activa todo el tiempo, estas pueden ser entrenadas con más rapidez. Y al ser una función simple, es computacional-mente menos costosa.

$$relu(x) = max(0,x)$$

La derivada de esta función es:

$$relu'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Y es indefinida en $x = 0$

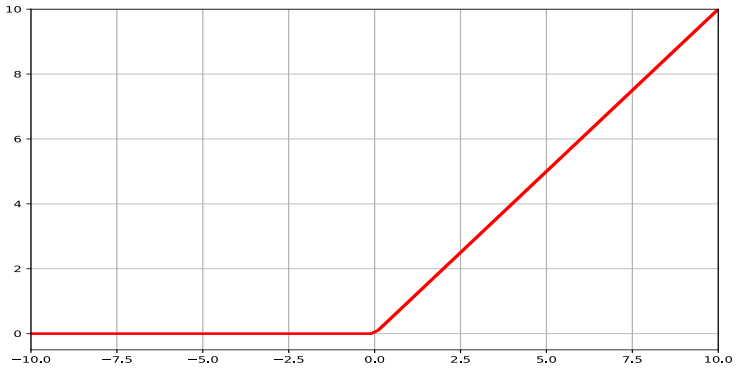


Figura 5. ReLU - Unidad lineal rectificada

Softmax.

Para problemas de clasificación, la función de activación *softmax* es usada en la capa de salida de la red. Esta función de activación recibe un vector de K valores arbitrarios y como salida entrega un vector de K valores en el rango entre 0...1, la suma de este vector es 1. Los valores de salida entregados por esta función pueden ser utilizados como probabilidades de clase.

$$\sigma(s) = \frac{\exp^{s_k}}{\sum_{k=1}^{K} \exp^{s_k}} \tag{2}$$

2.1.5. Deep Learning.

Aunque las redes neuronales multi-capa han existido desde 1980, su desarrollo ha sido limitado por avances tecnológicos. En los últimos 10 años, las redes neuronales han renacido, principalmente por la disponibilidad de equipos de computo más potentes y grandes conjuntos de imágenes. A inicios de los años 2000, fue descubierto que las redes neuronales pueden ser entrenadas eficientemente usando unidades de procesamiento gráfico (GPU). Estas son más eficientes para la tarea que las unidades de procesamiento tradicionales CPU y proporcionan una alternativa relativamente

menos costosa al hardware especializado (54).

Con el *deep learning*, la necesidad de crear soluciones refinadas de aprendizaje computacional refinadas manualmente ha disminuido. Un sistema clásico de detección de patrones, por ejemplo, incluía una fase manual de detección de características antes de una fase de aprendizaje computacional. El equivalente en *deep learning* consiste de solo una red neuronal. Las capas iniciales en la red aprenden a reconocer las características básicas, las cuales son pasadas a capas son las entradas de capas más avanzadas en la red.

2.1.5.1. Redes neuronales convolucionales.

El problema de resolver problemas de visión por computador usando redes neuronales tradicionales está en que una imagen de tamaño mediano contiene una enorme cantidad de información. Una imagen monocromática de 512x512 píxeles, para un total de 262.144 píxeles. Si cada valor de píxel de esta imagen es la entrada a una red neuronal completamente conectada, cada neurona requiere entonces 262.144 pesos. Una imagen de 1920x1080 píxeles requiere 2'073.600 pesos. Si las imágenes son policromáticas, el número de pesos es multiplicado por el número de canales de color (típicamente 3). Por esto, podemos ver que el número total de parámetros libres en la red rápidamente se vuelven extremadamente grandes con grandes imágenes. Un modelo demasiado grande causa un sobre-ajuste a los datos y reduce el rendimiento entregado (10).

Adicionalmente, múltiples tareas de clasificación requieren que la solución sea invariante

en traslación. Es ineficiente entrenar neuronas para reconocer por separado el mismo patrón en la esquina superior izquierda y en la inferior derecha de una imagen. Una red neuronal tradicional falla en tener en cuenta este tipo de estructura (37). La idea básica de una red neuronal convolucional (CNN) fue inspirada por un concepto en biología llamado el campo receptivo. Estos campos son una característica de la corteza visual animal. Estos actúan como detectores que perciben ciertos tipos de estímulos. Esta función biológica puede ser aproximada en computación, usando la operación de convolución. Esta operación entre una imagen f y una matriz filtro g está definida como:

$$h[x,y] = f[x,y] * g[x,y] = \sum_n \sum_m f[n,m]g[x-n,y-m] \tag{3}$$

El producto punto de el filtro g y una sección de la imagen f (con la misma dimensión de g centrada en las coordenadas x,y produce el valor del píxel h en la coordenada x,y. El tamaño del campo receptivo es ajustado por el tamaño de la matriz de filtro. Alineando el filtro sucesivamente con cada sub-imagen de f se produce el valor de la matriz h. En el caso de redes neuronales, la matriz de salida es llamada un *mapa de características* o de activación después de computar la función de activación. Los bordes de la imagen deben ser tratados como un caso especial, si la imagen f no tiene relleno en los bordes el tamaño en la salida de una convolución se reduce en cada paso de convolución. Típicamente, una CNN tiene las siguientes capas:

- Capa de convolución
- Capa de pooling

- Capa fully connected

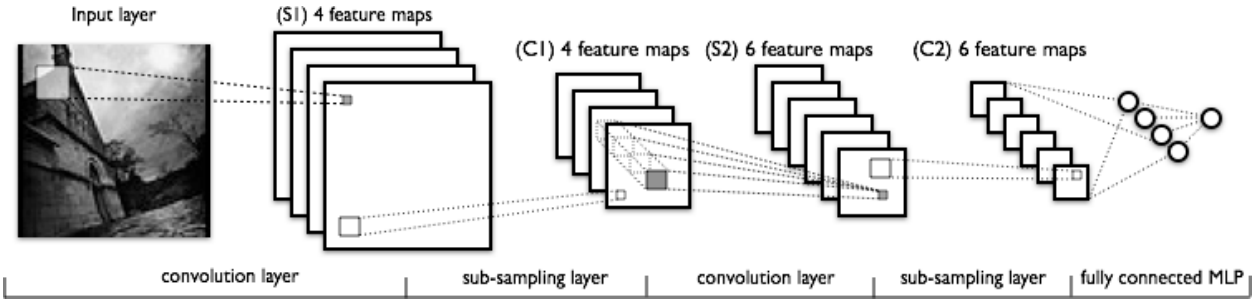


Figura 6. Arquitectura de una red neuronal convolucional.[1]

Un ejemplo de red convolucional es mostrado en la figura 6. El algoritmo de *back-propagation* tambien es aplicable a redes convolucionales. En teoría, las capas cercanas a la entrada deben aprender a reconocer características de bajo nivel, como son bordes y esquinas, y las capas cercanas a la salida aprenden a combinar estas características para reconocer formas más significativas.

2.1.5.2. Capa convolucional.

Un conjunto de filtros de convolución pueden ser combinados para formar una capa convolucional de una red neuronal. Los valores en los filtros de la matriz son tratados como parámetros de neuronas y entrenados usando aprendizaje computacional. La operación de convolución remplaza la operación multiplicación de una capa de red neuronal tradicional. La salida de la capa suele describirse como un volumen. El alto y ancho de este dependen de las dimensiones de el mapa de activación. La profundidad está dada por el número de filtros.

---

[1] `http://deeplearning.net/tutorial/lenet.html`

Dado que los mismos filtros son usados para todas las partes de la imagen, el número de parámetros es reducido drásticamente en comparación con una capa neuronal tradicional. Las neuronas de una capa convolucional comparten los parámetros y sólo están conectadas a una región local de la entrada. Compartir parámetros de convolución asegura la invariación en translación. A continuación se explican mediante ejemplos, las características y el volumen de salida de una capa convolucional.

Conectividad local.

Esta característica consiste en conectar cada neurona en la capa de convolución con una pequeña región en la imagen o volumen de entrada, pero conectada a todos los canales de color en profundidad. Esta región se conoce como **campo receptivo (F)**.

Arreglo espacial.

Consiste de 3 hiper-parámetros que controlan la salida de la capa de convolución.

1. **Profundidad**: controla el número de neuronas en una capa de convolución que están conectadas a la misma región en el volumen de entrada. Nos referimos a esta como la **columna de profundidad**

2. **Paso (S)**: este parámetro especifica la distancia entre columnas de profundidad donde ubicaremos campos receptores.

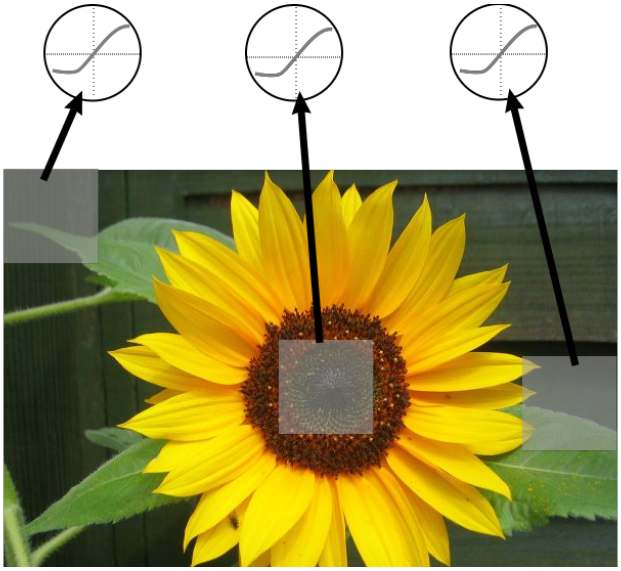


Figura 7. Conectividad local en una capa de convolución

3. **Relleno (P)**: en ocasiones es conveniente rellenar de ceros el borde de el volumen de entrada, normalmente para preservar las dimensiones.

La salida de una capa de convolución se computa de la siguiente manera:

$$W_{in} = \frac{W_{out} - F + 2P}{S} + 1$$

$$H_{in} = \frac{H_{out} - F + 2P}{S} + 1$$

donde W y H se refieren al ancho y alto del volumen respectivamente. Entonces, si tuviéramos la siguiente imagen de entrada con dimensiones $227_W * 227_H * 3_{rgb}$

y definimos los hiper-parámetros:

- Campo receptivo (F) = 11, Profundidad = 96

- Paso (S) = 4, Relleno (P) = 0

Nuestra salida tendrá las siguientes dimensiones $Salida = 55_W * 55_H * 96_{profundidad}$

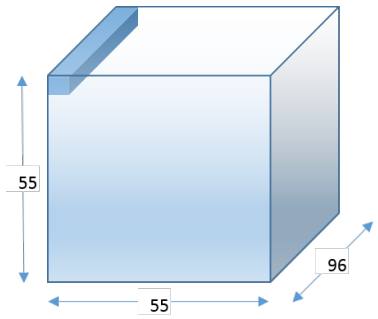


Figura 8. Salida de la capa de convolución

Comparten pesos y sesgo.

Dado que las imágenes tiene propiedades estacionarias, entonces las características aprendidas en una región de la imagen son útiles en una región diferente. Gracias a esta propiedad podemos hacer que todas las neuronas en la misma capa de profundidad compartan los mismos pesos y sesgo, para de esta manera buscar las características aprendidas en diferentes regiones de la imagen simultáneamente.

2.1.5.3. Capa de Pooling.

La función de esta capa consiste en reducir el volumen de entrada de la capa anterior para disminuir el tamaño de esta en memoria y reducir el tiempo de entrenamiento de la red. Al igual que una capa de convolución, la capa de pooling recibe como hiper-parámetros un campo receptivo F y un paso S, normalmente se usa un campo receptivo pequeño, pues un campo muy grande sería muy destructivo para la red y se perdería mucha información.

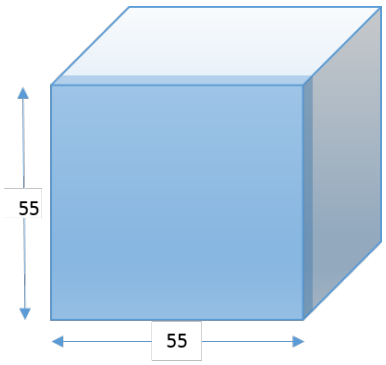


Figura 9. Región en capa de convolución que comparte los mismos pesos y bias

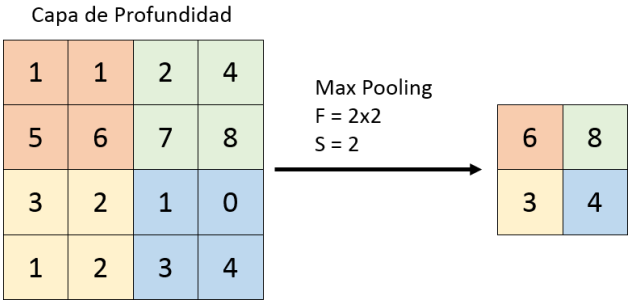


Figura 10. Salida de una Capa de pooling

Pooling tiene el efecto añadido de convertir a la red resultante más invariante a traslación al hacer los detectores menos precisos.

2.1.5.4. Capa Fully Connected.

Normalmente, después de una serie de capas convolucionales y pooling, el razonamiento final es hecho por una o más capas fully connected. Esta conecta todas las neuronas de la capa anterior a todas las neuronas de esta. La última capa fully connected tiene el mismo número de neuronas, así como clases hay en los datos a clasificar. En esencia es la misma red neuronal artificial vista en secciones anteriores.

2.1.6. Aprendizaje multimodal.

En términos generales, una modalidad se refiere a la forma en que algo sucede o es experimentado. El término es usualmente asociado con las modalidades sensoriales que representan nuestros canales primarios de comunicación, como la vista o el tacto. Un problema de investigación o conjunto de datos es considerado *multimodal* cuando este incluye múltiples modalidades de este tipo.

El campo de Aprendizaje computacional multimodal ofrece retos únicos debido a la heterogeneidad de los datos. Aprender de múltiples fuentes de datos respecto a una problemática ofrece la posibilidad de capturar correspondencias entre las modalidades y obtener un mayor entendimiento del fenómeno.

En (8), se propone una taxonomía para el aprendizaje multimodal, para el enfoque en redes neuronales convolucionales de este trabajo sólo se profundizará en la representación conjunta de

modalidades al ser las más relevantes para los problemas candidatos mencionados anteriormente.

2.1.6.1. Representación conjunta.

La representación conjunta proyecta múltiples representaciones unimodales uniendolas en un espacio multimodal, esta es usada en tareas donde los datos multimodales están presentes durante las etapas de entrenamiento e inferencia. Las redes neuronales se han convertido en un método muy popular para representación unimodal, sin embargo, su uso en el dominio multimodal se ha incrementado (41), (43), (60).

Con la finalidad de usar una red neuronal para representar los datos, se entrena la red parar realizar una tarea específica (e.g. Reconocer objetos en imágenes). Debido a la naturaleza multicapa de una red, se teoriza que esta representa los datos de una manera abstracta (9) y por ende, es común usar la última o penúltima capa como una forma de representación. Para construir una representación multimodal usando redes neuronales, cada modalidad es entrenada con varias capas individuales seguido por una capa oculta que proyecta las modalidades a un espacio conjunto. Esta representación es luego pasada a múltiples capas ocultas o usada directamente para predicción. Estos modelos pueden ser entrenados de comienzo a fin, aprendiendo a representar los datos y a realizar una tarea de clasificación.

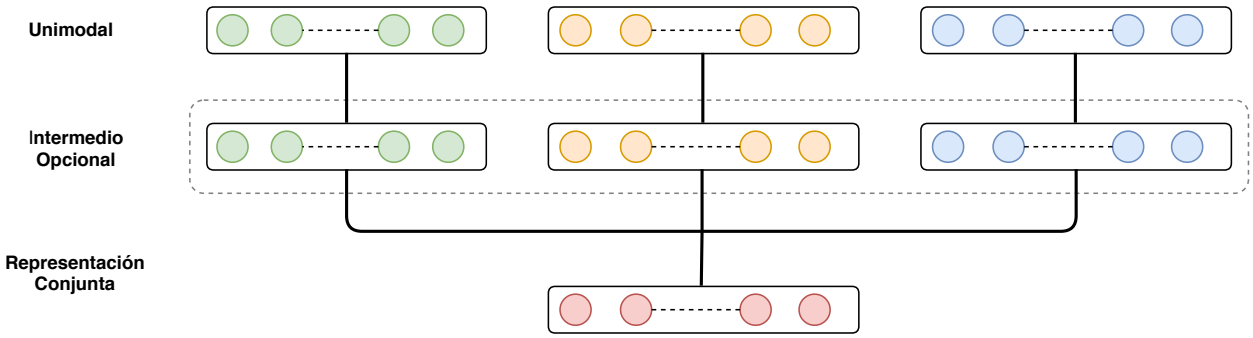


Figura 11. Representación conjunta (8)

2.1.6.2. Representación coordinada.

En esta representación multimodal, en lugar de proyectar las modalidades a un espacio conjunto, aprendemos representaciones separadas para cada modalidad pero coordinadas por una restricción.

Inicialmente, hablamos de **modelos de similitud** que minimizan la distancia entre modalidades en un espacio coordinado. Ejemplos de este acercamiento se pueden ver en (62), donde un espacio coordinado es construido para imágenes y sus anotaciones. Para esto se construye un mapa lineal donde la representación en texto y las imágenes tienen una menor distancia de coseno entre estas y mayor cuando no corresponden.

El uso de redes neuronales ha facilitado la construcción de representaciones coordinadas por su habilidad de aprender representaciones. La ventaja consiste en el hecho que estas pueden aprender conjuntamente representaciones coordinadas, sin usar otras técnicas de aprendizaje computacional diferentes. Ejemplos de estos acercamientos se pueden observar en (18; 59) donde se usan redes neuronales para coordinar representaciones de imágenes y descripciones en texto.

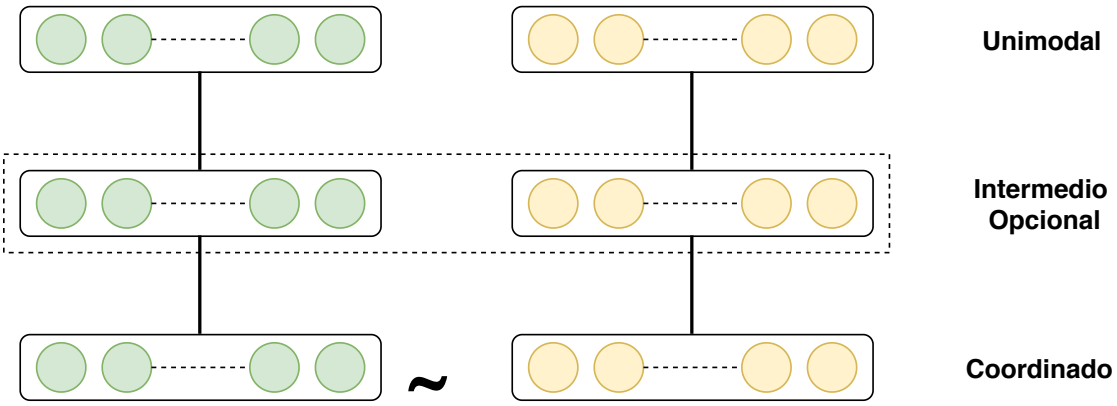


Figura 12. Representación coordinada (8)

2.1.6.3. Traducción.

En el caso de aprendizaje multimodal, traducción se refiere al mapeo de una modalidad a otra. Dada una entidad en una modalidad, la tarea consiste en generar la misma entidad en una modalidad diferente. La traducción multimodal se ha estudiado ampliamente y recientemente el interés en el campo ha sido renovado al combinar esfuerzos de visión por computador y procesamiento de lenguaje natural, así como la creación de grandes datasets multimodales (13) (56).

Es posible categorizar las traducciones en *basados en ejemplos* y *generativos*. Los modelos basados en ejemplos usan un diccionario al traducir entre modalidades. Por otra parte los modelos generativos construyen un modelo que es capaz de producir una traducción entre modalidades. La construcción de modelos generativos ha sido un reto mucho mayor en comparación a los modelos basados en ejemplos, sin embargo, con el uso del aprendizaje profundo ha sido posible la generación de imágenes (46), sonidos (44) y texto (7).

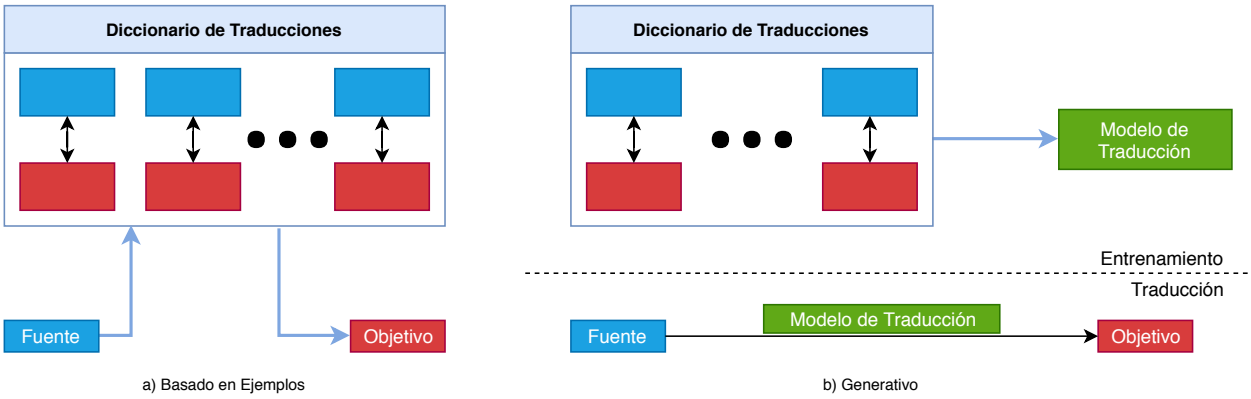


Figura 13. Modelos de traducción basados en *ejemplos* y *generativos*.

2.1.6.4. Alineación.

La alineación multimodal está definida como la búsqueda de correspondencias y relaciones entre componentes de instancias de dos o más modalidades. Estas pueden ser divididas en dos categorías: *implícito* y *explícito*. En alineación *explícita*, el principal objetivo está en alinear subcomponentes de instancias de múltiples modalidades usando una métrica de similitud (34), (73).

Por otra parte, el alineamiento *implícito* es usado como un intermedio para otras tareas. Esto permite un mejor rendimiento en múltiples tareas de reconocimiento de voz, descripción de imágenes y vídeos. Estos modelos no usan ejemplos de alineación supervisados, pero aprenden a alinear los datos durante el entrenamiento.

Estas alineaciones multimodales encuentran numerosas dificultades dado que existen pocos conjuntos de datos explícitamente anotados y es difícil diseñar métricas entre modalidades.

Adicionalmente, existen múltiples posibles alineaciones y no todos los elementos en distintas modalidades tienen correspondencia entre sí.

2.1.6.5. Fusión.

En términos técnicos, la fusión multimodal es el concepto de integrar información de múltiples modalidades con el objetivo de predecir una salida, ya sea una clase o un valor continuo. Es uno de los aspectos más investigados en el aprendizaje multimodal desde hace 25 años (67). El interés por fusión multimodal tiene diferentes beneficios. Primero, tener acceso a múltiples modalidades que observan el mismo fenómeno, permite predicciones más robustas. En segundo lugar, tener múltiples modalidades puede permitir capturar información complementaria, no visible en modalidades individuales (8).

La línea entre representación y fusión multimodal ha sido reducida en especial con modelos basados en redes neuronales profundas, donde la representación del aprendizaje se entrelaza con los objetivos de clasificación o regresión. Estos pueden ser clasificados en dos categorías: *agnósticos* que no dependen directamente del método de machine learning usado, y *basados en modelo* que directamente usan la fusión en la construcción de estos.

Modelos agnósticos.

Estos modelos pueden ser divididos en tres categorías: fusión *temprana*, *tardía* e *híbrida* (6). La fusión *temprana* integra inmediatamente las características después que son extraídas simplemente

concatenando sus representaciones. La fusión *tardía* realiza la integración después que cada modalidad ha tomado una decisión (clasificación o regresión) usualmente promediando o votando entre las resultados de modalidades individuales. Por último, la fusión *híbrida* combina las salidas de la fusión temprana y predictores de modalidades individuales.

Basados en modelos.

Mientras los modelos agnósticos son fáciles de implementar usando métodos de aprendizaje de máquina unimodales, terminan usando técnicas que no son diseñadas para manejar datos multimodales. Esta sección puede ser dividida en tres categorías de fusión: métodos de kernel, modelos gráficos y redes neuronales.

Aunque los métodos basados en kernel y los modelos gráficos han sido ampliamente utilizados en fusión multimodal, (22), (39) en el contexto de esta propuesta, nos interesa el uso de la fusión multimodal en redes neuronales. Estas son usadas en múltiples campos como reconocimiento de gestos, análisis de sentimientos y descripción de vídeos.

2.1.6.6. Co-aprendizaje.

Es particularmente relevante cuando una de las modalidades tiene recursos limitados, ausencia de datos anotados, ruidosos o poco confiables. En esta categoría a menudo una modalidad es usada solo durante entrenamiento y no es usada durante prueba. Se han identificado tres tipos de co-aprendizaje basado en los recursos durante entrenamiento: *paralelo*, *no-paralelo* e *híbrido* (8).

El co-aprendizaje multimodal permite a una modalidad influenciar el entrenamiento de la otra, explotando información complementaria a través de modalidades. Es importante destacar que el co-aprendizaje es una tarea independiente y puede ser usada para crear mejores modelos de fusión, traducción y alineación.

3. Estado del Arte

3.1. Modelos de Exposición e Identificación de Tipología Estructural

El análisis de riesgo sísmico en entornos urbanos se realiza usando una combinación de tres tipos de modelos: de riesgo (probabilidad de que ocurran sismos), de vulnerabilidad (capacidad de las estructuras de sobrellevar estos eventos) y de exposición (que se refiere a los elementos susceptibles de sufrir daño en un evento sísmico) (5). En este contexto, el desarrollo de un modelo de exposición involucra cuantificar estos elementos en entornos urbanos, los cuales pueden ser edificios, poblaciones o actividades socio-económicas. Conocer la tipología estructural de los edificios en una determinada región es una pieza clave en la construcción de este tipo de modelos, dado que permite agrupar los edificios que muestren un rendimiento comparable durante un evento sísmico (65).

La identificación de la tipología estructural, y en especial determinar el sistema de lateral de resistencia de cargas y los materiales de construcción, puede ser realizada con certeza usando los planos de la estructura o por observación directa de expertos. Sin embargo, en la mayoría de casos el acceso a estos planos es restringido, o en construcciones artesanales que no involucran ingeniería, es prácticamente inexistente. Debido a esto, la mejor opción para compilar estos inventarios son estudios in-situ por expertos en ingeniería civil desde diferentes perspectivas del edificio, no obstante, cubrir grandes extensiones urbanas requiere considerables recursos económicos y de tiempo (64), (45), (30), (38).

En la última década, el desarrollo de tecnologías de sensado remoto y exploración, y su alta disponibilidad en entornos urbanos ha permitido la medición de múltiples variables como el área construida, altura de edificios, tipo de techos y edad de las construcciones, donde apoyados en la opinión de un experto ha permitido facilitar de cierta manera el análisis de grandes regiones y apoyar el desarrollo de modelos de exposición (36), (24), aunque los autores consultados resaltan las limitaciones en las perspectivas disponibles de los edificios dificultando el análisis experto.

En el contexto de clasificación de estructuras por expertos, se distinguen dos enfoques: (1) basados en conocimiento, donde se definen un conjunto de reglas, manuales o descripciones que al ser aplicados permiten cierto grado de exactitud, aunque limitada su aplicabilidad a niveles locales (42), (63). (2) basados en datos, al aumentar la complejidad en la clasificación y la cantidad de datos disponibles, se genera una mayor dificultad en la etapa de modelado por un experto, por lo tanto, múltiples autores han basado sus esfuerzos en la aplicación de técnicas de aprendizaje automático a partir de los datos. Estos enfoques serán mostrados en las siguientes secciones del documento.

3.2. Machine Learning en Identificación Estructural

Las limitaciones presentadas anteriormente en la construcción de inventarios urbanos y la disponibilidad de herramientas de exploración remota, ofrecen la oportunidad de aplicar técnicas de Machine Learning en el análisis de estructuras residenciales. Podemos dividir estas técnicas en

dos grandes subconjuntos: supervisado (donde se trabaja con un conjunto de datos, previamente etiquetado en distintas clases por un experto) y no-supervisado (donde los elementos no tienen clases y el algoritmo determina las particiones naturales en los datos). En el enfoque no-supervisado se presentan algunas aplicaciones usando combinaciones de bases de datos oficiales (catastro) y no oficiales (OpenStreetMap), donde las agrupaciones obtenidas coinciden con la opinión de expertos, sin embargo. los autores consultados coinciden en una mayor utilidad de estos métodos cuando la clasificación de grandes agrupaciones de edificios tiene más relevancia que la clasificación individual (61), (21).

Las técnicas de aprendizaje supervisado han tenido una mayor aplicabilidad y reusabilidad en este ámbito, en primer lugar debido a la gran disponibilidad de herramientas de exploración remota, las cuales han permitido la creación de grandes conjuntos de datos donde diversos tipos de estructuras urbanas son identificadas y analizadas por expertos en ingeniería civil, adicionalmente se pueden encontrar registros de mayor complejidad donde las dimensiones de las estructuras son registradas usando polígonos de dos y tres dimensiones (40), (14).

Evaluando la forma, dimensiones y densidad de los edificios, múltiples algoritmos clásicos han sido aplicados a la clasificación de estructuras urbanas. En la literatura se han registrado estudios de aplicaciones de máquinas de soporte vectorial y árboles de decisión (50), (48), (69), (28).

De igual manera, se encuentran estudios de comparación de rendimiento entre diversas téc-

nicas de aprendizaje computacional. Sin embargo, se destaca que la clasificación se hace a un nivel superficial, donde la finalidad del edificio (residencial, comercial, industrial, etc) tiene una mayor importancia y no se ahonda a una clasificación de tipo estructural (53), (27). En (19) se presenta una clasificación de tipos estructurales usando sensado remoto y las características de los edificios encuestados son procesadas y calculadas por los autores. Esta clasificación supervisada se realiza mediante la aplicación de máquinas de soporte vectorial y Random Forest. A conocimiento del autor, este es el único trabajo donde las clases de los edificios tienen una relevancia importante en la construcción de modelos de exposición, dado que considera tanto materiales como sistemas de resistencia de cargas.

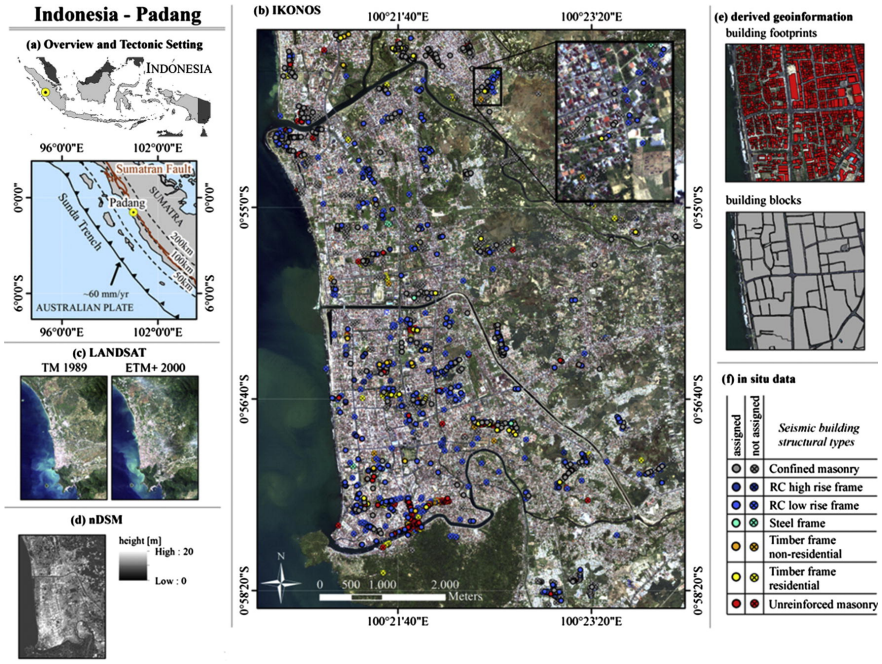


Figura 14. Clasificación de tipo de estructuras usando Random Forest y SVM. (19)

3.3. Deep Learning en Identificación Estructural

Con la emergencia del deep learning y las redes neuronales convolucionales (CNN), se ha desarrollado un importante interés en su aplicación en entornos urbanos, tanto para la identificación de estructuras como en el análisis inteligente de entornos urbanos. Las herramientas de sensado remoto han incrementado exponencialmente la disponibilidad de datos satelitales en todos los aspectos, frecuentemente de origen multimodal y geolocalizados (72).

Las aplicaciones en entornos urbanos son ampliamente diversas, pero se resaltan por su utilidad para la creación de inventarios urbanos las siguientes: en (29) y (23) se presenta un mapeo de cañones urbanos usando exploración remota mediante imágenes de Google Street View. De igual manera, usando imágenes obtenidas por esta tecnología, estudios en la construcción de inventarios de estructuras urbanas se reportan mediante la clasificación de edificios, enfocados principalmente en la finalidad del edificio, mas no a nivel estructural (33), (12).

Adicionalmente, considerables investigaciones han sido presentadas en el análisis de imágenes satelitales para distinguir tanto el uso de la tierra como la finalidad de los edificios, usando como entrada imágenes en alta resolución (70), (17) y una combinación de imágenes multiespectrales a variadas escalas (71) (58). Por último, una comprensiva revisión de recursos e investigaciones de deep learning aplicado a imágenes de sensado remoto puede ser encontrada en (72) donde se destaca la característica sinérgica de estas tecnologías.

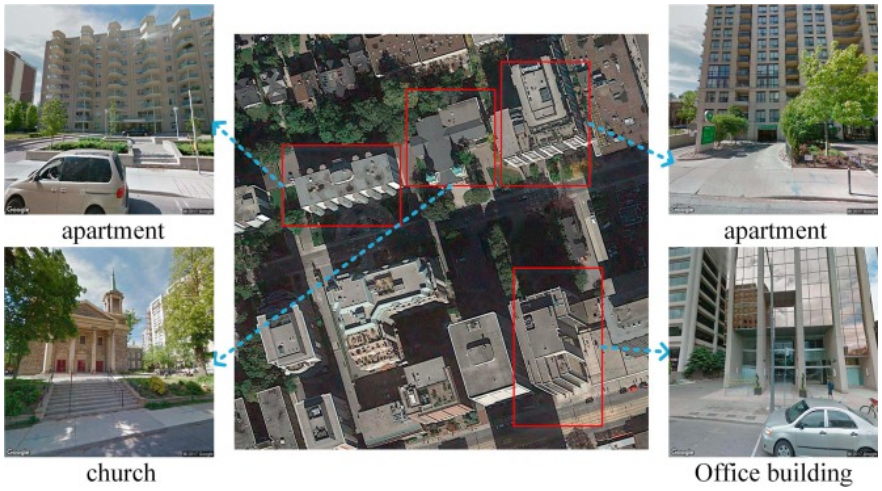


Figura 15. Clasificación de Edificios según su finalidad. (33)

4. Caso de Estudio

4.1. Colaboración

Durante la etapa de definición de una problemática urbana donde, en primer lugar, exista una disponibilidad de datos en múltiples modalidades, y en segundo lugar, tenga una relevancia destacable y aplicabilidad en el análisis de entornos urbanos, fue de suma importancia la colaboración con grupos de investigación de la ciudad de Medellin, el grupo RISE (Research in Spatial Economics) de la Universidad EAFIT y el grupo IN2LAB (Intelligent Information System Lab) de la Universidad de Antioquia.

Figura 16. Grupo IN2LAB y Grupo RISE

4.2. Área de Interés

Medellín al día de hoy es la segunda ciudad más poblada de Colombia con una población estimada en 2,5 millones de habitantes, en un área de 1152 km^2, dividida en 271 barrios agrupados en 16 distritos urbanos. Medellín (Fig. 17) está localizado en un valle intermontañoso a 1460m sobre el nivel del mar, en una zona de riesgo sísmico medio (3). La posibilidad de ocurrencia de temblores y el tamaño de la ciudad a este riesgo en una preocupación de varias décadas (1), (47), (5).

Figura 17. Área de Estudio: Medellín, Colombia

La calidad de la construcción en las edificaciones de Medellín está fuertemente relacionado con el nivel económico y la fecha de construcción. Las mejores prácticas de construcción se encuentran en zonas de ingresos medio-altos y altos. Adicionalmente, sólo a partir de 1984 se reglamentó el diseño sísmico en Colombia, por lo tanto, pocas edificaciones construidas antes de esta fecha puede resistir cargas sísmicas. Por otra parte, un gran porcentaje de las construcciones en Medellín son informales en zonas de bajo ingreso y no cumplen los requisitos y tienen un mal rendimiento al recibir cargas sísmicas.

4.3. Modelo de Exposición

Conocer la tipología de las construcciones en una determinada región es clave para desarrollar un modelo de exposición para determinar el riesgo sísmico (65). El modelo de exposición es una descripción detallada de los bienes en una región incluyendo propiedades, infraestructura, población y actividades económicas (20). Un modelo de exposición, junto con modelos de riesgo sísmico y vulnerabilidad, es usado para estimar la probabilidad de pérdidas en la ocurrencia de un terremoto (4), (51). La evaluación del riesgo sísmico para el conjunto de edificios de un área urbana requiere clasificarlos basados en la tipología estructural, un parámetro que define el comportamiento del edificio cuando es expuesto a cargas sísmicas.

4.3.1. Tipología Estructural de Edificios.

La tipología estructural está dada en función del sistema lateral de resistencia a cargas y sus materiales, la altura del edificio, la fecha de construcción y la forma del edificio, entre otros (2), (31). Siguiendo la taxonomía de edificios desarrollada por Global Earthquake Model (GEM) Foundation (11), en este trabajo se consideran solamente edificios residenciales cuya estructura tipológica está basada en los siguientes tres atributos.

- **Sistema lateral de resistencia de cargas** se refiere a los elementos horizontales y verticales que transfieren las fuerzas sísmicas laterales a las bases del sistema. Estos pueden ser un sistema de muros, vigas y columnas, vigas/columnas/muros, etc.

- **Material del sistema de resistencia** puede ser de concreto reforzado, mampostería, acero,

o piedras, etc.

- **Ductilidad** se refiere a la capacidad del sistema para sufrir deformaciones antes del colapso.

El sistema lateral de resistencia de cargas puede ser identificado a partir de los planos u observación directa por expertos. En la mayoría de los casos, los planos estructurales son limitados y para edificaciones que surgen de procesos de construcción artesanal o iniciativas personales esta información, no existe. Debido a esto, las encuestas realizadas por expertos, se muestra como la mejor opción para generar un inventario de edificios en zona urbanas. Sin embargo, en grandes ciudades no es posible encuestar a todos los bienes presentes, por lo tanto, se asumen varias características para desarrollar un modelo de exposición.

Las opiniones de un experto son necesarias para obtener detalles no incluidos en fuentes de información oficiales como datos de catastro o información de censos. La alta disponibilidad de tecnologías de sensado remoto y el alto grado de penetración en entornos urbanos a nivel mundial, ofrecen la oportunidad de obtener mediciones de área construida, altura de edificios, tipos de tejado, entre otros.

4.3.2. Conjunto de Datos.

El conjunto de datos usado en este trabajo ha sido construido por expertos en ingeniería civil de la Universidad EAFIT usado para la elaboración de un modelo de exposición en la ciudad de Medellín (24). En el marco de colaboración con el Grupo RISE, hemos podido acceder a este conjunto de

datos el cual ha sido construido usando la herramienta Google Street View, obteniendo un total de 11513 imágenes de edificios residenciales; cada imagen fue guardada junto con información básica del edificio como Geolocalización, Estratificación y Número de pisos. Un conjunto de atributos fueron determinados por expertos para determinar el sistema lateral de resistencia de cargas, el tipo del material del sistema y el nivel de ductilidad. Los edificios fueron seleccionados de manera que múltiples niveles socio-económicos, prácticas de construcción y alturas de edificios fueran incluidas.

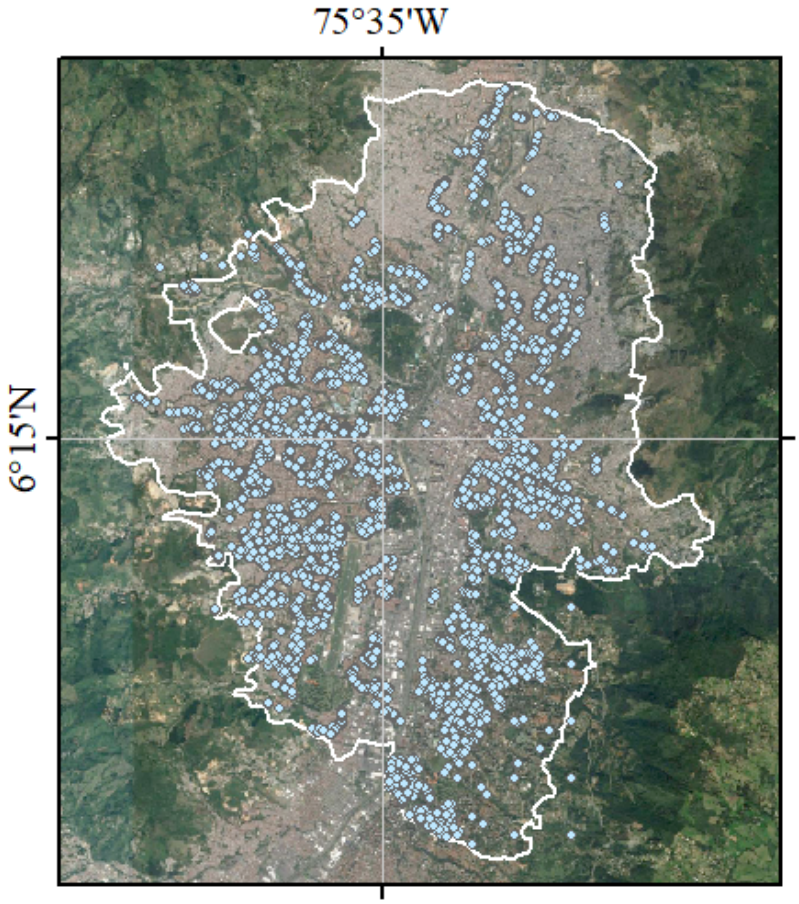


Figura 18. Edificios encuestados en Medellín.

Figura 19. Ejemplos de imágenes atípicas eliminadas del dataset.

Antes de considerar un proceso de entrenamiento de redes neuronales es importante analizar los datos que se usarán y eliminar los valores atípicos que se encuentren. De las 11513 imágenes inicialmente reportadas en el dataset original (24), se eliminaron los edificios idénticos presentes en los datos, por ejemplo estos pertenecientes al mismo complejo residencial. También se eliminaron las imágenes donde las características que permiten identificar el tipo de estructura están obstruidas.

Al terminar este proceso de limpieza de datos, un total de 9989 edificios del dataset original, fueron seleccionados para realizar el proceso de entrenamiento de las diferentes arquitecturas. Cada edificio está identificado por tres parámetros principales relacionados con el sistema de resistencia de cargas laterales: Tipo de Material / Tipo de Sistema de Cargas/ Ductilidad.

Tabla 1
Carácterísticas identificadas sobre los edificios.

Material del Sistema	**Tipo del Sistema**	**Ductilidad**
Mampostería NO Reforzada (MUR)	Muros de Ladrillos (LWAL)	Dúctil (DUC)
Mampostería Reforzada (MR)	Pórticos de Concreto (LINF)	NO Dúctil (DNO)
Mampostería Confinada (MCF)	Sistema Dual (LDUAL)	
Concreto (CR)		

La distribución de los edificios en el dataset para estas características definidas se presentan en la Tabla 2. Se puede observar un desbalance en la cantidad de imágenes de las clases del dataset, por lo que se usaron técnicas de entrenamiento y balanceo de clases durante el entrenamiento de las redes neuronales, dando una mayor importancia a los errores sobre las clases poco representadas.

Tabla 2
Descripción del Dataset

No.	Tipología del Edificio	No. de Edificios	Porcentaje$(\%)$
1	CR/LDUAL/DUC	177	1,77
2	CR/LFINF/DNO	1921	19,23
3	CR/LFINF/DUC	1081	10,82
4	CR/LWAL/DUC	128	1,28
5	MCF/LWAL/DUC	167	1,67
6	MCF/LWAL/DNO	231	2,31
7	MR/LWAL/DUC	195	1,95
8	MUR/LWAL/DNO	6089	60,96

5. Metodología

La metodología presentada en este documento fue pensada para obtener un entendimiento robusto de los métodos de Deep Learning aplicados a esta problemática. Esta metodología consiste en la ejecución del experimento de clasificación usando cinco arquitecturas pre-entrenadas de redes convolucionales de diferentes complejidades y hacer un remuestreo tres veces del dataset original, obteniendo tres distribuciones independientes y dividir cada una en subconjuntos de entrenamiento(train), prueba(test) y validación(val). Por último, obtener un promedio de las métricas de rendimiento sobre las distribuciones para asegurar robustez en la selección del mejor resultado. La Figura 20 representa un esquema de la metodología explicada. Estas secciones serán explicadas a continuación.

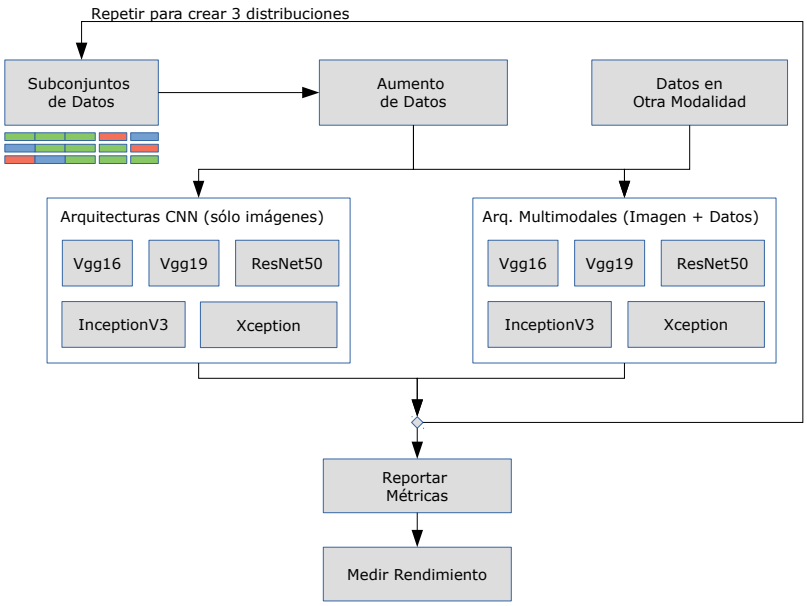


Figura 20. Flujo de trabajo de la Metodología.

5.1. Etapa 1: Método de Validación del Desempeño

En un flujo de trabajo de Machine Learning típico los datos son usados con tres finalidades: (1) entrenamiento del modelo; (2) ajustar los parámetros del modelo; y (3) medir el rendimiento de los modelos seleccionados con datos nunca usados en las etapas anteriores. Esto nos permite obtener un estimado del rendimiento del modelo al ser usado en un ambiente de operación real. Por esta razón, el conjunto de datos es dividido en tres subconjuntos para ser usado en las tres etapas descritas anteriormente y se mantiene la proporcionalidad en las clases del dataset. Para el subconjunto de entrenamiento (train) usamos el 60 % de los datos, para el subconjunto de validación (val) usamos el 20 % y el 20 % restante en prueba (test).

Para reducir el overfitting de las técnicas aplicadas usamos una versión no exhaustiva de cross-validation, para esto creamos tres distribuciones de datos y las métricas reportadas son promediadas sobre estas. Cada una de estas distribuciones consiste en una división 60/20/20 como se explicó previamente. La Figura 21 ilustra estas tres distribuciones y cómo cada una es una configuración de datos diferente.

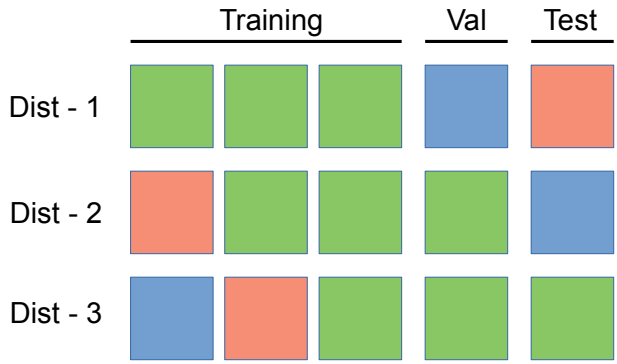


Figura 21. Distribuciones de datos usados en el proyecto.

5.1.1. Aumento de Datos.

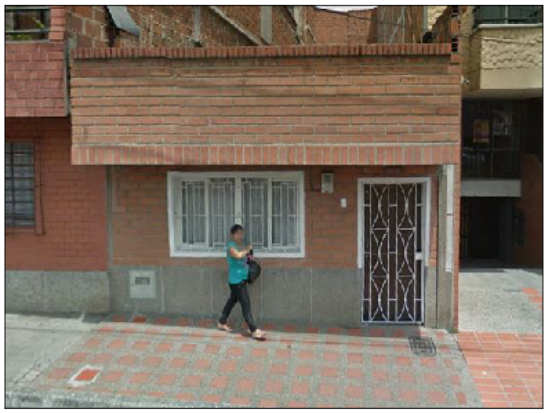
El aumento de datos es un paso fundamental para reducir el sobre-ajuste a los datos de entrenamiento y permite a la red neuronal generalizar mejor sobre datos no conocidos. El método consiste en añadir ruido al subconjunto de entrenamiento a través de transformaciones a las imágenes que permitan virtualmente generar una imagen completamente nueva para la red, pero preservando las características que permitan identificar la imagen. En este proyecto se usó la herramienta **ImgAug** (32) que permite implementar una o múltiples transformaciones de manera aleatoria. A continuación, se muestra un ejemplo de las transformaciones aplicadas al dataset de manera individual.

- **Mirroring:** Reflejar la imagen sobre la horizontal.

(a) Original

(b) Imagen Reflejada

- **GaussianBlur:** Difuminar las imágenes ligeramente usando un kernel gaussiano.



(c) Original

(d) Imagen Difuminada

- **Crop:** Reducir la imagen a un porcentaje aleatorio del tamaño original.

(e) Original

(f) Imagen Recortada

- **ChannelShuffle:** Aleatoriamente cambia los canales de color en la imagen.

(g) Original

(h) Imagen con canales de color cambiados.

- **Rotate:** Rota la imagen un pequeño porcentaje en una dirección aleatoria en la horizontal.

(i) Original

(j) Imagen Rotada

5.1.2. Características numéricas..

El dataset usado en este proyecto incluye datos adicionales bajo otra modalidad que caracterizan cada uno de los edificios como el número de pisos (36 pisos edificio más alto), la estratificación del edificio (baja-media-alta) y las coordenadas de cada uno de estos (latitud y longitud). Dado que incluir información de otras modalidades tiene el potencial de mejorar el rendimiento de los modelos de clasificación, y la obtención de estos datos adicionales requiere un mayor tiempo en la creación del dataset, se hace necesario entender el valor de inyectar la información adicional en una arquitectura de red convolucional.

Para incluir estos datos bajo otra modalidad en el proceso de entrenamiento, primero es necesario transformar las entradas numéricas en vectores usando **one-hot encoding** con el fin de

facilitar la predicción en la red sobre estas características. Los datos se han transformado de la siguiente manera:

- **Número de Pisos:** un vector de **36** posiciones con **1** en el piso reportado para el edificio y **0** para los demás valores.
- **Estrato:** un vector de **3** posiciones con **1** para el estrato reportado y **0** para los demás valores.
- **Coordenadas:** se toman las mínimas y máximas valores de Latitud/Longitud y se crea una rejilla sobre la ciudad de Medellín. De esta rejilla se obtienen un vector para Latitud de 134 posiciones y uno para Longitud de 90 posiciones, ambos con **1** en el par Lat/Lon donde se encuentra el edificio y **0** en las demás posiciones.

Estos datos extra son concatenados en la entrada multimodal creando un vector de entrada de **263 posiciones**.Para que se puedan usar estos datos las arquitecturas convolucionales escogidas deben ser modificadas para recibir como entrada este vector.

5.2. Etapa 2: Diseño e Implementación de Arquitecturas

Para crear una línea base de comparación entre el entrenamiento usando sólo imágenes y el entrenamiento multimodal, se han seleccionado 5 arquitecturas de redes neuronales convolucionales que han mostrado un gran éxito en Imagenet, un desafío de clasificación de imágenes a gran escala con 1 millón de imágenes y 1000 clases (16). Estas arquitecturas y sus respectivos pesos preentrenados sobre Imagenet permiten que usemos una técnica llamada FineTuning en donde las

capas iniciales de la red retienen características visuales aprendidas de Imagenet, y las capas finales aprenden características específicas de nuestro problema de clasificación. Este método ha sido probado que mejora la capacidad de generalización en múltiples tareas de clasificación (66). Las arquitecturas convolucionales escogidas y sus características principales son explicadas a continuación.

5.2.1. Vgg16 / Vgg19.

Estas arquitecturas de 16 y 19 capas respectivamente, fueron diseñadas para analizar el efecto de aumentar la profundidad en una red neuronal en tareas de clasificación. Para la creación de estas redes se usa como base la arquitectura Alexnet (35). Esta arquitectura es modificada reemplazando las capas convolucionales con filtros grandes (11x11, 7x7, 5x5) cada uno por múltiples capas convolucionales con filtros pequeños (3x3, 1x1) que permiten añadir mayor profundidad a la red y mayor no-linealidad lo que le permite aprender funciones más complejas y mantener un número menor de parámetros que la arquitectura base Alexnet. (52) La arquitectura completa de esta red se encuentra en el Apéndice 1.

5.2.2. InceptionV3.

Esta arquitectura es basada en parte en las mejoras obtenidas por las arquitecturas Vgg16 y Vgg19 de factorización de grandes filtros convolucionales en pequeñas pilas de pequeños filtros. En esta arquitectura se introduce el modulo Inception 22, que además de factorizar grandes filtros, también usa filtros de diferente tamaño para aprender características a diferentes escalas. Los módulos Inception son conectados uno tras de otro para generar una arquitectura de una gran profundidad

(55). La arquitectura completa de esta red se encuentra en el Apéndice 2.

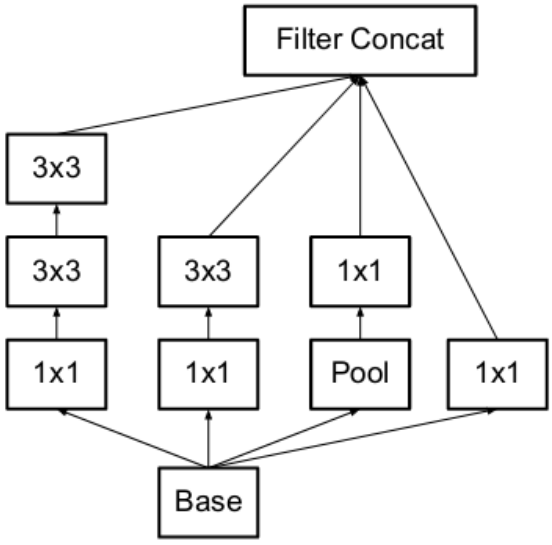


Figura 22. Modulo de InceptionV3. Fuente (55)

5.2.3. Resnet50.

A medida que las redes neuronales crecen en tamaño, con más frecuencia se presenta un problema conocido como 'gradiente desvanecido' (vanishing gradient). Cuando se usa el método de back-propagation, descrito en la sección 2.1.3 durante el entrenamiento de la red neuronal, y se obtienen las derivadas parciales de todas las capas de la red desde la capa final hasta la inicial. Siguiendo la regla de la cadena, estas derivadas son multiplicadas sucesivamente por toda la red para obtener los valores de las derivadas en las capas iniciales. Sin embargo, cuando una gran cantidad de capas ocultas obtienen valores pequeños en estas derivadas y son multiplicadas entre sí, causan que el valor del gradiente disminuya exponencialmente. Un valor muy pequeño del gradiente significa que los pesos de las capas iniciales no serán actualizados efectivamente con cada paso de entrena-

miento.

La solución presentada en Resnet50 (26) a este problema, consiste en usar conexiones residuales entre bloques de capas convolucionales para permitir que la información del gradiente se transmita con más eficiencia en la red. La arquitectura completa de esta red se encuentra en el Apéndice 3.

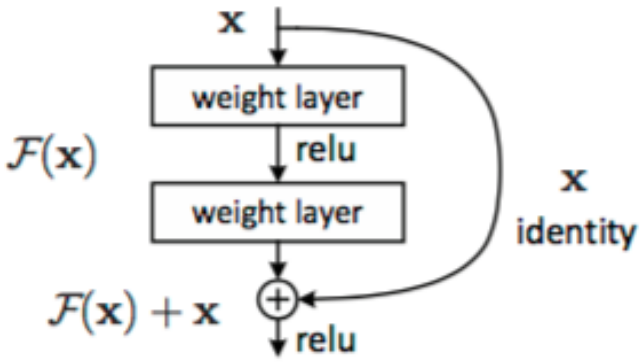


Figura 23. Bloque Residual en ResNet50. Fuente (26)

5.2.4. Xception.

Esta arquitectura construye sobre todas las innovaciones presentadas en las redes anteriores e introduce un bloque de convolución que realiza dos operaciones. (1) Realiza una convolución 1x1 sobre la entrada a la red, (2) realiza una convolución sobre todos los canales de la entrada a la capa. En la práctica esta red ha superado el estado del arte sobre las demás redes mencionadas anteriormente. La arquitectura completa de esta red se encuentra en el Apéndice 4.

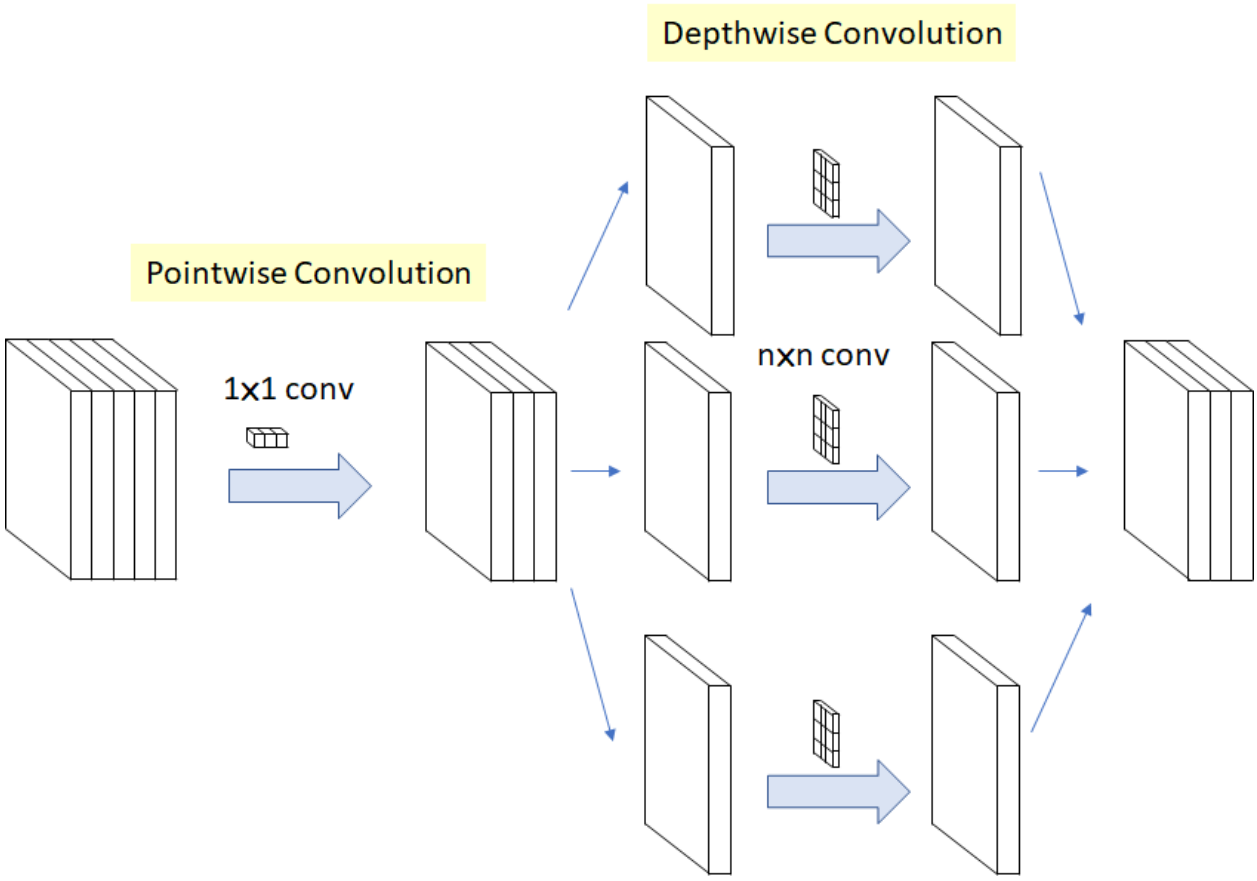


Figura 24. Bloque de Convolución en Xception. Fuente (15)

5.2.5. Arquitecturas multimodales.

Para realizar el segundo experimento, fue necesario usar las arquitecturas convolucionales y adaptarlas las capas finales para aceptar información de otra modalidad. Esto se logró diseñando un perceptrón multicapa de tres capas cuya entrada son los datos adicionales después de aplicar one-hot encoding en los datos numéricos, es decir, un vector con 263 elementos. Durante la fase de entrenamiento se introducen simultáneamente las imágenes en la arquitectura CNN que se está usando y el vector adicional en el perceptrón. Por último, los vectores de características aprendi-

dos en la fase final de estas redes paralelas son concatenados para producir el resultado final de clasificación. En la Figura 25 se muestra una representación de las arquitecturas convolucionales y arquitecturas multimodales.

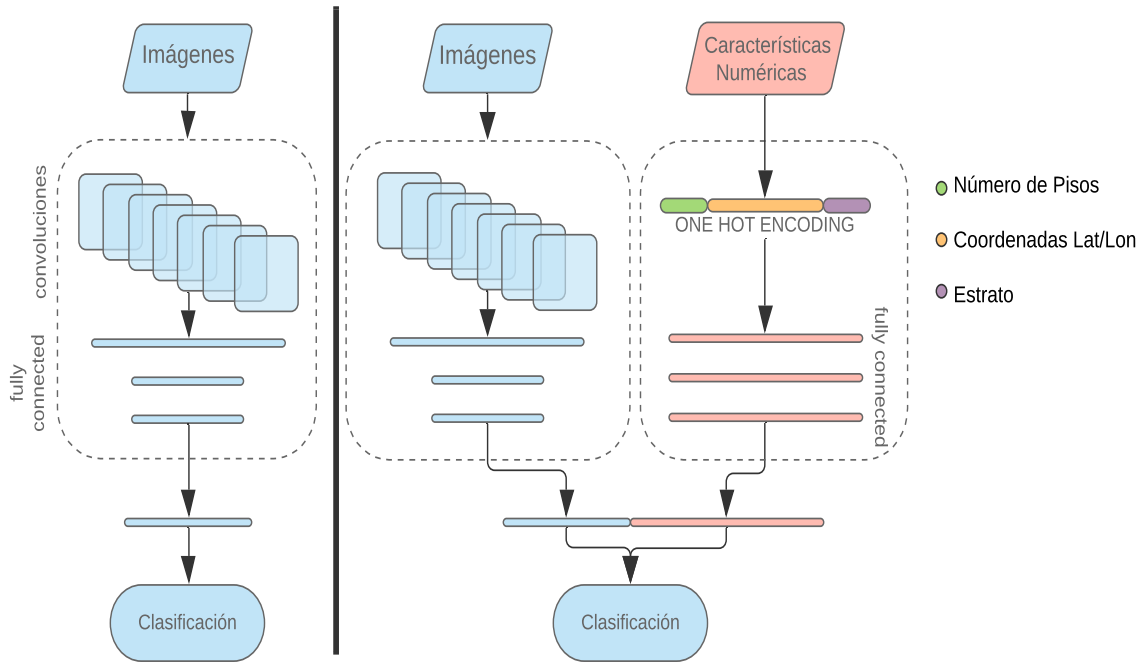


Figura 25. Comparación de arquitecturas con sólo imágenes y multimodal. Fuente propio.

5.3. Etapa 3: Evaluación de Arquitecturas

5.3.1. Fase de Entrenamiento.

En esta sección se describe en detalle las condiciones y el hardware usado para el entrenamiento de las diferentes arquitecturas.

Hardware.

Los entrenamientos fueron realizados en los centros de computación y tarjetas GPU listados a continuación:

- **GUANE** del Centro de Supercomputación y Cálculo Científico de la Universidad Industrial de Santander[2], sobre una GPU Titan X.
- **APOLO** del Centro de Computación Científica de la Universidad EAFIT[3], sobre una GPU Tesla V100.

Hiper-Parámetros de Entrenamiento.

Los parámetros de un entrenamiento se refiere a los pesos W y el sesgo b que se aprenden durante el proceso de optimización. Los hiper-parámetros controlan el proceso de aprendizaje de los parámetros W y b. Estos determinan el aprendizaje de cada una de las redes y permiten la replicabilidad del entrenamiento. El proceso de entrenamiento de una red neuronal es un proceso bastante empírico donde se deben encontrar los mejores hiper-parámetros iterativamente. A continuación se explican los valores usados finalmente y su justificación.

- **Width, Height:** Ancho y Alto de las imágenes **(224x224)**, este tamaño se presenta en común entre todas las redes convolucionales escogidas. Todas las imágenes están en formato JPG.

---

[2] https://www.sc3.uis.edu.co/

[3] http://www.eafit.edu.co/apolo

- **Número de Imágenes por Iteración:** durante el entrenamiento de la red, el conjunto de datos es divido en pequeños lotes. Una iteración de entrenamiento ve uno de estos lotes y actualiza los pesos W y b en la red. Estos lotes son de **48 imágenes** cada uno y está limitado por la cantidad de memoria que tiene la GPU.

- **Epochs:** el entrenamiento se lanza hasta completar un máximo de **600 epochs** o hasta que se cumplan las **condiciones de entrenamiento** explicadas en la siguiente subsección. Una epoch se completa cuando la red ha visto todo el dataset completo (9989 imágenes) dando pasos de iteración de 48 imágenes cada uno.

- **Optimización:** para el proceso de optimización de la red se usa el algoritmo **Adadelta**(68), el cual es una variación del método de Gradiente Descendiente. La ventaja de este algoritmo es su actualización de la tasa de aprendizaje a medida que entrenamos la red.

- **Tasa de Aprendizaje:** este valor inicialmente se fija en *1.0*. Al usar el método de optimización, **Adadelta** la tasa de aprendizaje se ajusta automáticamente durante el entrenamiento.

Condiciones de Entrenamiento.

Estas condiciones definen el proceso de entrenamiento y son necesarios para optimizar el uso de los recursos computacionales y detener el entrenamiento de la red o realizar ajustes cuando el valor de la función de costo no está disminuyendo. Las condiciones son las siguientes:

1. **Detención temprana:** si después de 80 epochs el valor de la función de costo sobre el dataset de validación no ha disminuido un 0.001, el entrenamiento es detenido.

2. **Reducción de Tasa de Aprendizaje:** si después de 40 epochs el valor de la función de costo sobre el dataset de validación no ha disminuido un 0.001, la tasa de aprendizaje se disminuye al 10 %.

3. **Punto de control:** al terminar cada epoch se evalúa la función de costo sobre el dataset de validación. Si este valor es menor al último valor registrado durante el entrenamiento, se guardan los pesos y los hiperparámetros de la red durante ese instante. Este modelo guardado será considerado como el mejor modelo obtenido durante el entrenamiento.

Es importante destacar estas condiciones de entrenamiento dado que afectan directamente el tiempo computacional requerido por las diferentes arquitecturas para alcanzar un punto de equilibrio donde la red no presenta sobreajuste a los datos de entrenamiento de las arquitecturas de sólo imágenes y multimodales. Durante esta fase el tiempo computacional para todos los experimentos ha sido registrado y se realizó una comparación entre arquitecturas con sólo imágenes y multimodales, esta medición es promediada sobre las arquitecturas en las 3 distribuciones.

5.3.2. Fase de Validación y Prueba.

Una vez terminada la fase de entrenamiento sobre todas las arquitecturas (5 de sólo imágenes, 5 multimodales), se obtienen 10 modelos por cada una de las 3 distribuciones para un total de 30 modelos. Para medir y comparar modelos entre las diferentes arquitecturas existen múltiples métricas de clasificación que pueden ser usadas, sin embargo, teniendo en cuenta el des-balance de clases en el conjunto de datos, no todas son significativas.

Para identificar las métricas correctas a utilizar, y al ser este un trabajo interdisciplinario, fue necesario discutir ampliamente con expertos en Ingeniería Civil de la Universidad EAFIT que construyeron y etiquetaron el conjunto de datos usado en este proyecto, y que tienen una vasta experiencia en el desarrollo de modelos de Exposición en la ciudad de Medellín y pueblos aledaños. De esta manera, se distinguen 3 aspectos importantes a medir en los resultados obtenidos por las diferentes redes.

En primer lugar, es necesario considerar la capacidad de la red para distinguir tipologías estructurales frágiles, es decir, las más probables a sufrir daños durante un evento sísmico (No Dúctiles) y evitar clasificarlas como una tipología Dúctil. En segundo lugar, es necesario que los modelos puedan distinguir entre los tipos de materiales usados en las estructuras (Mampostería y Concreto Reforzado). Por último, se debe analizar la clasificación considerando todos los atributos que definen cada edificio identificado en el dataset, es decir, clasificación sobre las 8 clases definidas.

Aunque **Accuracy** (Exactitud, porcentaje de clasificaciones correctas) es posiblemente la métrica más usada en problemas de clasificación, esta es engañosa cuando las clases en el conjunto de datos no están balanceadas o el costo de clasificaciones correctas e incorrectas no es simétrico. De manera que para evaluar el rendimiento de los modelos en nuestro problema se usarán las métricas de **Precision** (Precisión) y **Recall** (Exhaustividad) cuyas definiciones serán explicadas a

continuación. Además se usarán matrices de confusión para observar los cambios de clasificación en las clases del dataset para las distintas modalidades y errores de clasificación, para las 8 clases definidas y los errores de clasificiación de estructuras frágiles (DNO como DUC).

5.3.2.1. Métricas de Clasificación.

Accuracy (Exactitud, porcentaje de clasificaciones correctas) es posiblemente la métrica más usada en problemas de clasificación, sin embargo esta medida es engañosa cuando las clases en el conjunto de datos no están balanceadas o el costo de clasificaciones incorrectas no es simétrico. De esta manera, para evaluar el rendimiento de los modelos obtenidos se usarán otras métricas que tienen en cuenta estas características: **Precision** (Precisión) y **Recall** (Exhaustividad) cuyas definiciones serán explicadas a continuación. Además se usarán matrices de confusión para observar los cambios en la clasificación al nivel de clases.

Tabla 3
Elementos de una matriz de confusión

Resultado.	Real Clase 1	Real Clase 0
Predicción Clase 1	*True Positive*	*False Positive*
Predicción Clase 0	*False Negative*	*True Negative*

Partiendo de la definición de una matriz de confusión como la presentada en la Tabla 3, podemos definir las métricas de **Precision** y **Recall**.

Precision (Precisión)

Esta se define como la proporción de predicciones positivas que son correctas. Un modelo con alta

exactitud tendría pocos **False Positives**. Esta métrica se define de la siguiente manera:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Exhaustividad (Recall)

Esta se define como la proporción de positivos reales que fueron correctamente identificados. Un modelo con alta exhaustividad tendrá una baja cantidad de **False Negatives**. Esta métrica se define de la siguiente manera.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Para calcular la forma multiclase de estas dos métricas, primero se calcula Precision o Recall para cada una de las categorías y se obtiene un promedio ponderado por el número de imágenes en cada clase.

6. Resultados

6.1. Tiempo Computacional

El entrenamiento de una red neuronal convolucional continúa hasta que las condiciones de detención temprana o el número máximo de epochs son alcanzadas. En la Figura 26 podemos observar el tiempo computacional promedio que le toma a las diferentes arquitecturas una sesión de entrenamiento. Se observa que en general la versión multimodal de la arquitectura requiere un mayor tiempo computacional que la versión de sólo imágenes. En el caso de las arquitecturas InceptionV3 y ResNet50 cerca al doble de tiempo para terminar el entrenamiento.

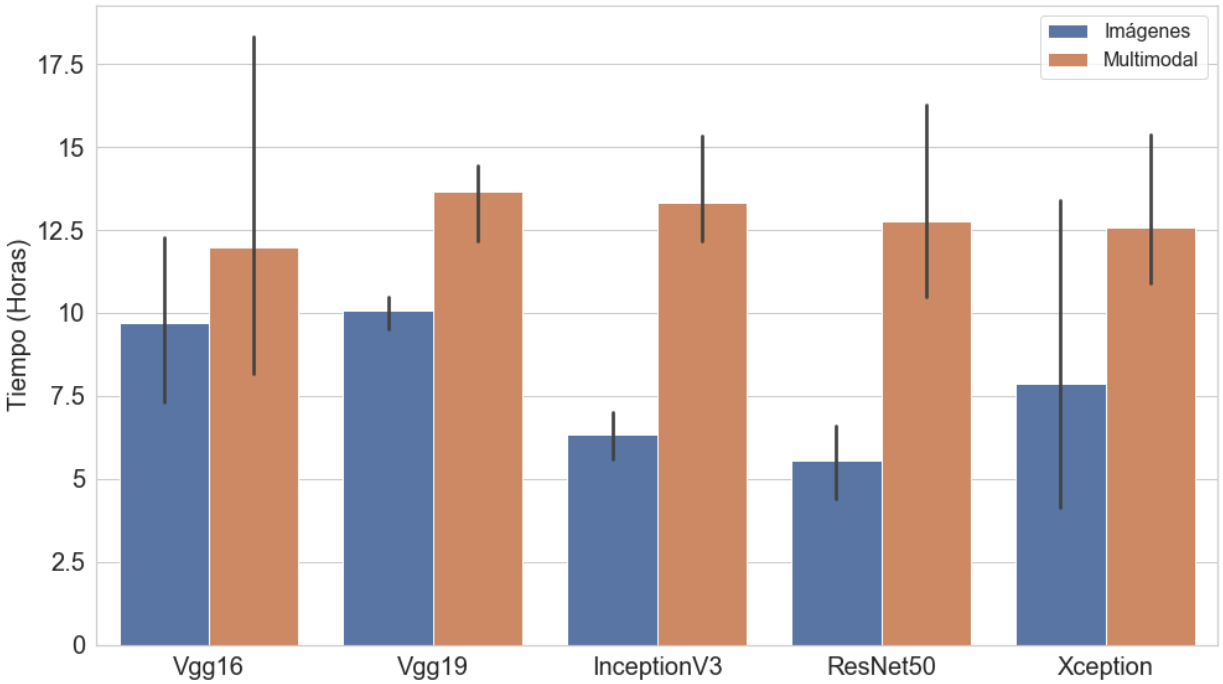


Figura 26. Tiempo computacional promedio por arquitectura en los distintos modos de entrenamiento.

6.2. Análisis de Métricas de Clasificación

Los resultados respecto al subconjunto de Validación se muestran en la Tabla 4 para las dos métricas de interés *recall* y *precision*. Para cada una de las arquitecturas usadas podemos observar una mejora menor en los valores de *precision* para la versión multimodal entre el 5 % y 6 %. Se puede observar de igual manera un ligero aumento en *Recall* para las arquitecturas más simples como **Vgg16** y **Vgg19**.

Tabla 4
Métricas de rendimiento usando Datos de Validación.

	$Recall_{val}$				$Precision_{val}$			
	dist1	dist2	dist3	*promedio*	dist1	dist2	dist3	*promedio*
$Vgg16_i$	0.66	0.60	0.69	0.65	0.71	0.69	0.69	0.70
$Vgg16_m$	0.71	0.64	0.68	0.68	0.76	0.74	0.76	0.75
$Vgg19_i$	0.58	0.61	0.65	0.61	0.72	0.66	0.69	0.69
$Vgg19_m$	0.70	0.70	0.71	0.70	0.76	0.75	0.76	0.75
$InceptionV3_i$	0.73	0.69	0.70	0.70	0.71	0.66	0.69	0.69
$InceptionV3_m$	0.72	0.68	0.68	0.69	0.76	0.75	0.75	0.75
$Xception_i$	0.72	0.71	0.73	0.72	0.71	0.69	0.71	0.70
$Xception_m$	0.71	0.68	0.70	0.70	0.76	0.75	0.75	0.75
$Resnet50_i$	0.71	0.69	0.72	0.71	0.70	0.68	0.69	0.69
$Resnet50_m$	0.73	0.68	0.68	0.70	0.75	0.75	0.75	0.75

i: solo imágenes,
m: multimodal

En la Tabla 5 analizamos el comportamiento de los diferentes modelos en cuanto a la identificación de estructuras frágiles, es decir, analizar las métricas de rendimiento considerando solamente el atributo de Ductilidad: No-Dúctil (CR/LFINF/DNO, MCF/LWAL/DNO, MUR/LWAL/DNO) y Dúctil (CR/LDUAL/DUC, CR/LFINF/DUC, CR/LWAL/DUC, MCF/LWAL/DUC, MR/LWAL/DUC).

Tabla 5
*Métricas de rendimiento **No-Dúctil** usando Datos de Validación.*

	$Recall_{val}$				$Precision_{val}$			
	dist1	dist2	dist3	*promedio*	dist1	dist2	dist3	*promedio*
$Vgg16_i$	0.88	0.86	0.92	0.89	0.94	0.93	0.92	0.93
$Vgg16_m$	0.94	0.88	0.92	0.92	0.95	0.94	0.94	0.94
$Vgg19_i$	0.80	0.89	0.88	0.86	0.94	0.92	0.92	0.93
$Vgg19_m$	0.94	0.93	0.94	0.93	0.95	0.94	0.94	0.94
$InceptionV3_i$	0.94	0.96	0.95	0.95	0.94	0.92	0.92	0.93
$InceptionV3_m$	0.93	0.89	0.91	0.91	0.95	0.94	0.94	0.94
$Xception_i$	0.96	0.97	0.95	0.96	0.94	0.91	0.92	0.92
$Xception_m$	0.93	0.92	0.94	0.93	0.95	0.95	0.94	0.95
$Resnet50_i$	0.97	0.96	0.96	0.96	0.93	0.92	0.92	0.92
$Resnet50_m$	0.95	0.93	0.92	0.93	0.95	0.94	0.94	0.94

i: solo imágenes,
m: multimodal

Podemos observar como las arquitecturas de mayor profundidad (InceptionV3, Xception, Resnet50) consiguen distinguir las estructuras frágiles con facilidad utilizando información solamente de la imagen y la inyección de información multimodal no tiene un efecto positivo en la clasificación. Por otra parte, en las arquitecturas más simples (Vgg16, Vgg19), se observa una mejora importante en las métricas reportadas.

6.3. Análisis de Matrices de Confusión

Basados en las métricas de *Recall* y *Precision* a nivel general y considerando el atributo de Ductilidad, así como tiempo computacional de todas las arquitecturas, se elegió uno de los modelos de mejor rendimiento (**ResNet50**, y seguidamente, se hizo una comparación entre la versión de

sólo imágenes con la respectiva versión multimodal usando matrices de confusión para analizar la clasificación al nivel de cada clase. Las matrices de confusión que se presentan en las Figuras 27 y 29 son el promedio de las tres distribuciones de datos sobre la arquitectura **ResNet50** usando los datos de Prueba (Test).

Para facilidad de lectura, se presenta la tabla de descripción del dataset con la nomenclatura y los índices de cada clase.

Tabla 6
Descripción del Dataset

No.	Tipología del Edificio	No. de Edificios	Porcentaje$(\%)$
1	CR/LDUAL/DUC	177	1,77
2	CR/LFINF/DNO	1921	19,23
3	CR/LFINF/DUC	1081	10,82
4	CR/LWAL/DUC	128	1,28
5	MCF/LWAL/DUC	167	1,67
6	MCF/LWAL/DNO	231	2,31
7	MR/LWAL/DUC	195	1,95
8	MUR/LWAL/DNO	6089	60.96

En la Figura 27 podemos observar dos detalles presentes en la matriz de confusión. Cada clase está numerada como se muestra en la Tabla 6 y muestra el Tipo de Material al que pertenece. Adicionalmente, para distinguir con mayor facilidad los errores en Ductilidad la matriz se divide separando las clases No Dúctiles (2,6,8) y las clases Dúctiles (1,3,4,5,7).

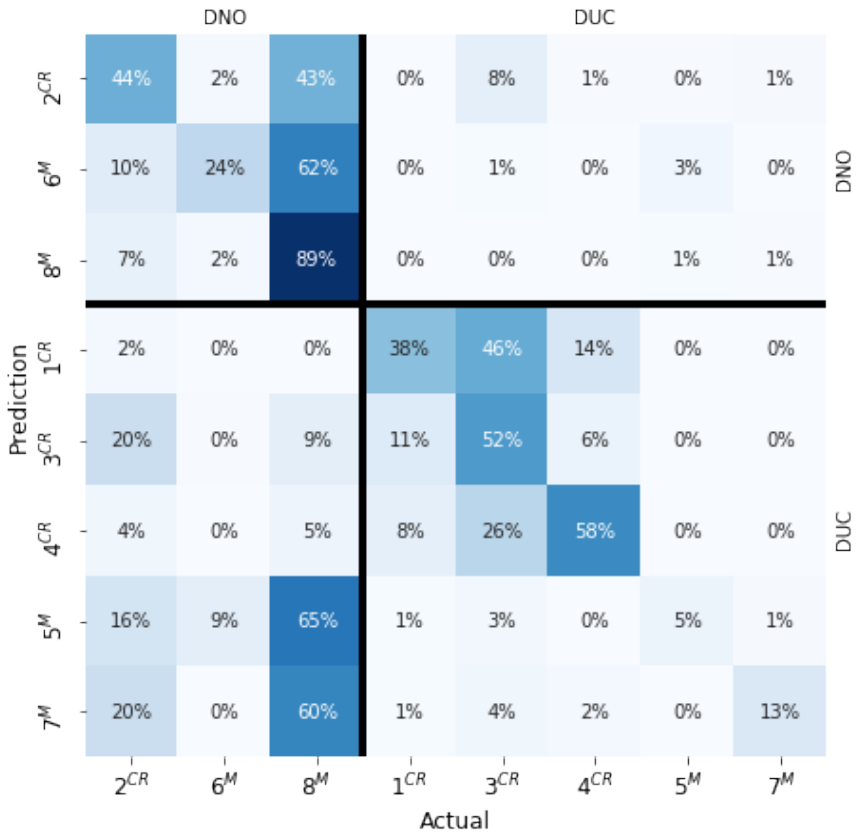


Figura 27. Matriz de confusión promedio para Resnet50 en modalidad de sólo imágenes. Cada fila suma 100 %. El enfoque está en como las predicciones están distribuidas con respecto a los valores reales. (CR: Concreto Reforzado, M: Mampostería) (DNO: No-Dúctil, DUC: Dúctil)

- Considerando los niveles de Ductilidad se puede observar cómo la mayoría de errores en las clases Dúctiles están concentrados en un tipo de clase: confundir las clases 5 (MCF/LWAL/DUC) y 7 (MR/LWAL/DUC) como clase 8 (MUR/LWAL/DNO). El error en este sentido es comprensible y de menor importancia dado que estas clases manejan el mismo tipo de material y sistema de resistencia de cargas. En el sentido contrario, clasificar una estructura No Dúctil como una Dúctil, se destaca porque el error es significativamente menor y las clases DNO son en su gran mayoría confundidas por otras clases No-Dúctiles.

- Cuando consideramos la clasificación según el tipo de material Concreto Reforzado (CR) y Mampostería (M) podemos observar que este modelo tiene un alto nivel de clasificación donde ambas clases tienen un Accuracy igual o superior al 80 %, presentado en la Figura 28.

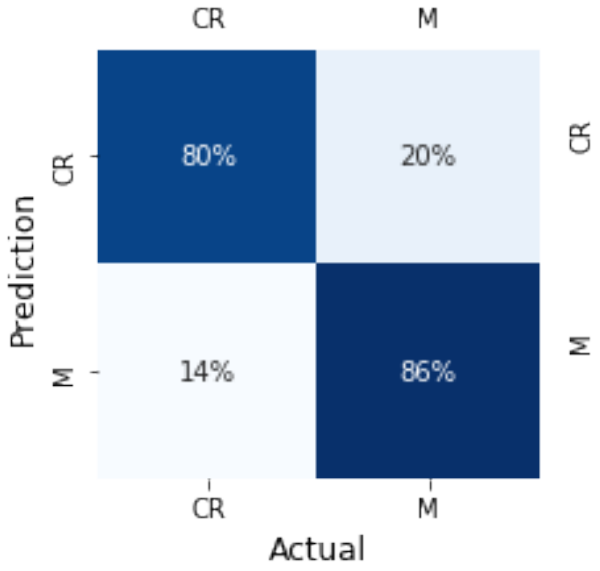


Figura 28. Matriz de confusión promedio para Resnet50 por tipo de Material usando sólo imágenes. Cada fila suma 100 % de los datos en cada subclase. (CR: Concreto Reforzado, M: Mampostería)

- Aunque a nivel de ductilidad y tipo de material el nivel de clasificación es alto. La mayoría de errores ocurren cuando se considera el tipo de sistema de resistencia de cargas. En consulta con ingenieros civiles de la Universidad EAFIT, se destaca que este tipo de sistema es a menudo difícil de distinguir por dos razones: (1) las características que diferencian el tipo de sistema no se pueden analizar desde una sola perspectiva del edificio y (2) a menudo estas características se encuentran ocultas por fachadas o existen dentro del edificio, las cuales no pueden ser obtenidas solo por una imagen.

Cuando analizamos la matriz de confusión obtenida de la arquitectura multimodal de **ResNet50** en la Figura 29 podemos observar una diagonal más marcada (clasificaciones correctas) sobre el dataset.

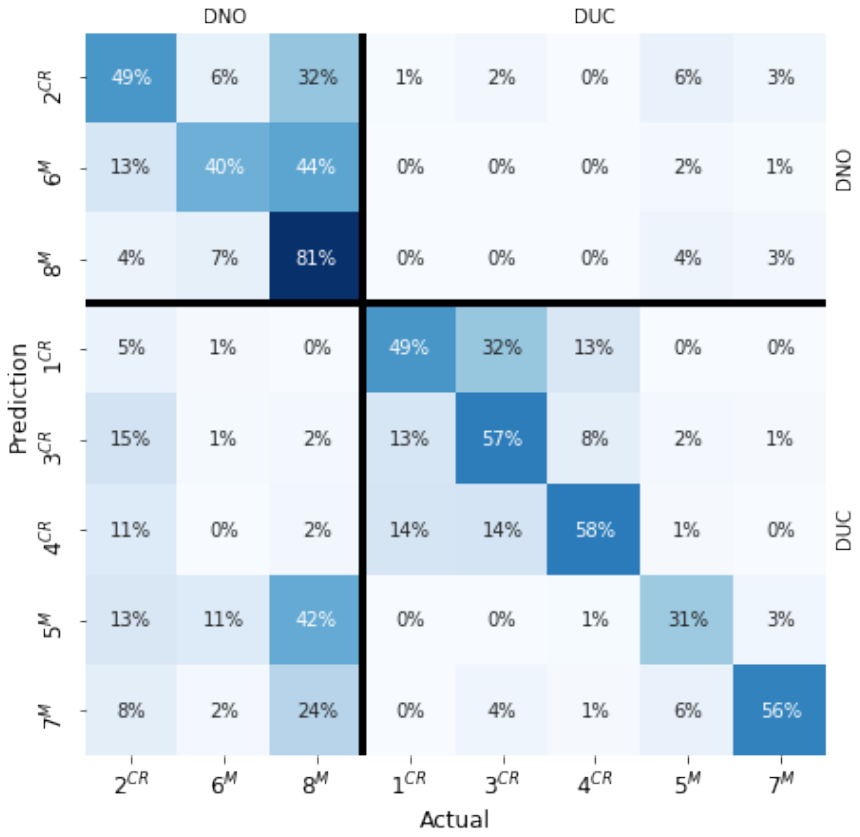


Figura 29. Matriz de confusión promedio para Resnet50 en multimodalidad. Cada fila suma 100 %. El enfoque está en como las predicciones están distribuidas con respecto a los valores reales. (CR: Concreto Reforzado, M: Mampostería) (DNO: No-Dúctil, DUC: Dúctil)

- La identificación de estructuras frágiles a nivel de Ductilidad es importante destacar que el error de mayor importancia (confundir clases No Dúctiles como Dúctiles) se mantiene a un nivel bastante alto, similar al encontrado en la versión de sólo imágenes y, adicionalmente una reducción de errores en el sentido contrario (DUC como DNO) lo cual se evidencia

con una reducción en los falsos positivos donde la clase 5 (-23 %) y la clase 7 (-36 %) eran clasificados como la clase 8.

- A nivel de tipo de material se obtiene una mejora en la clasificación de estructuras en Concreto (CR) del 6 %, y la clasificación de estructuras de Mampostería se mantiene estable.

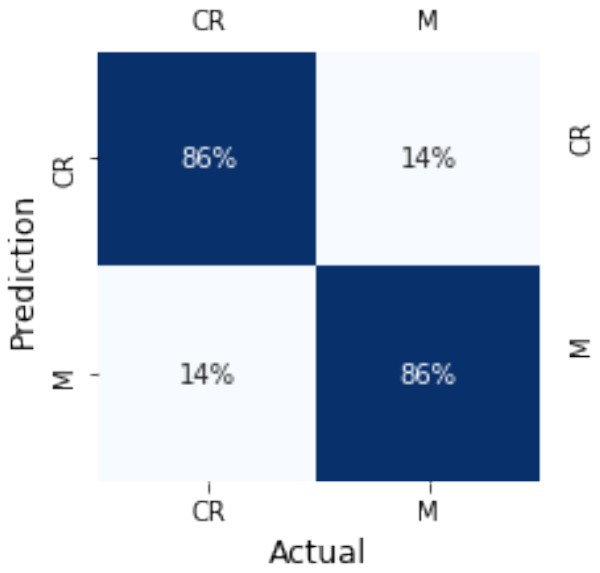


Figura 30. Matriz de confusión promedio para Resnet50 multimodal por tipo de Material en Multimodalidad. Cada fila suma 100 %. El enfoque está en como las predicciones están distribuidas con respecto a los valores reales. (CR: Concreto Reforzado, M: Mampostería)

- Cuando consideramos todos los atributos en la clasificación de las 8 clases del dataset, podemos destacar un incremento bastante importante, particularmente en las clases menos representadas en el dataset. Esto se puede observar con más detalle en la Figura 31 donde se presenta la diferencia entre las matrices de confusión promedio de las dos modalidades de entrenamiento.

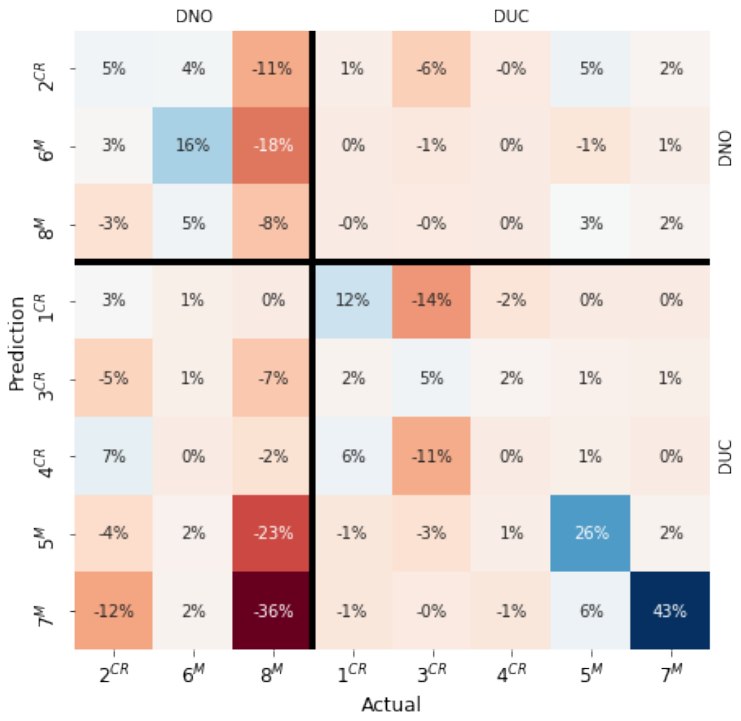


Figura 31. Diferencia entre matrices de confusión promedio de ambas modalidades.

7. Conclusiones y Observaciones Generales

7.1. Conclusiones

En el desarrollo de este proyecto se exploró la fusión de información multimodal en diferentes arquitecturas de redes convolucionales pre-entrenadas como técnica para obtener modelos que presenten una mayor capacidad de generalización bajo métricas de rendimiento de aprendizaje computacional. Se aplicó una metodología que nos permitió abordar el problema de identificación de estructuras a nivel urbano y determinar una confianza estadística sobre las métricas reportadas. Producto de este proyecto se obtienen múltiples modelos de deep learning bajo la modalidad de sólo imágenes y multimodales que permiten clasificar estructuras residenciales con cierto grado de

exactitud.

Considerando lo descrito anteriormente, podemos concluir:

- El tiempo computacional requerido para el entrenamiento de redes convolucionales usando datos multimodales es considerablemente mayor al requerido por su contra-parte de sólo imágenes, aún cuando se usen los mismos hiper-parámetros y condiciones de entrenamiento. Sin embargo, no existe diferencia en el tiempo de inferencia durante la fase de validación.

- Las arquitecturas más simples presentan un mayor impacto positivo por la inclusión de información multimodal. Se mostró cómo la versión multimodal de estas arquitecturas simples presenta métricas de clasificación comparables a arquitecturas de una mayor complejidad. Esto brinda la posibilidad de uso en dispositivos móviles, teniendo en cuenta las restricciones de memoria y procesamiento en los mismos.

- Los modelos de deep learning obtenidos en ambas modalidades permiten la identificación de estructuras residenciales con un grado de precisión y exhaustividad importante. Este rendimiento permitiría reducir ampliamente los costos en la construcción de modelos de exposición y creación de inventarios residenciales, además de permitir a expertos en ingeniería civil analizar regiones urbanas de gran tamaño de manera eficiente.

- Aunque la investigación desarrollada presenta limitaciones en la obtención de los datos (requiere un experto para su identificación) y la calidad de los datos (obstrucciones mayores como vegetación o fotos mal tomadas) se presenta la posibilidad a futuro de implementar

estos modelos en dispositivos móviles donde la toma de imágenes sea realizada por usuarios y los modelos realicen una preidentificación de las estructuras. De esta manera se podría desarrollar un proceso de realimentación y mejora continua en los modelos de deep learning, donde la intervención de un experto en ingeniería civil sería necesario en los casos que los algoritmos presenten un mayor grado de confusión.

7.2. Observaciones

- El trabajo presentado en este libro fue publicado en la revista *Building and Environment*[4] con el título *Automatic detection of building typology using deep learning methods on street level images*.

- Para facilitar la reproducibilidad de este proyecto, todos los datos, el código utilizado y las instrucciones de uso se encuentran en el repositorio *https://github.com/druedaplata/buildings_repo*.

---

[4] https://www.journals.elsevier.com/building-and-environment

Referencias Bibliográficas

[1] (1999). Instrumentación y microzonificación sísmica del área urbana de medellín. Technical report, Sistema Municipal para la Prevención y Atención de Desastres, Medellín, Colombia.

[2] (2003). Hazus–mh mr4 technical manual. Technical report, Federal Emergency Management Agency, Mitigation Division, Washington, DC, USA.

[3] (2009). Estudio general de amenaza sísmica de colombia. Technical report, Asociación Colombiana de Ingeniería Sísmica, Comité AIS–300, Bogotá, Colombia.

[4] (2010). Hazus–mh mr5, multi-hazard loss estimation methodology–earthquake model. Technical report, Department of Homeland Security, Federal Emergency Managment Agency, Mitigation Division, Washington, DC, USA.

[5] Acevedo, A. B., Jaramillo, J. D., Yepes-Estrada, C., Silva, V., Osorio, F. A., and Villar-Vega, M. (2017). Evaluation of the seismic risk of the unreinforced masonry building stock in antioquia, colombia. *Natural Hazards*, 86(1):31–54.

[6] Atrey, P. K., Hossain, M. A., El Saddik, A., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6):345–379.

[7] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[8] Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.

[9] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

[10] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

[11] Brzev, S., Scawthorn, C., Charleson, A. W., Alle, L., Greene, M., Jaiswal, K., and Silva, V. (2013). Gem building taxonomy version 2.0. Technical report, GEM Foundation, Pavia, Italy, Tech. Rep. 2013-02 V1.0.0.

[12] Chen, Q., Wang, L., Wu, Y., Wu, G., Guo, Z., and Waslander, S. L. (2018). Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings. *arXiv preprint arXiv:1807.09532*.

[13] Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

[14] Chen, Y., Hong, T., Luo, X., and Hooper, B. (2019). Development of city buildings dataset for urban building energy modeling. *Energy and Buildings*, 183:252–265.

[15] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

[16] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[17] Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., and Zou, H. (2018). Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, 145:3–22.

[18] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al. (2013). Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129.

[19] Geiß, C., Pelizari, P. A., Marconcini, M., Sengara, W., Edwards, M., Lakes, T., and Taubenböck, H. (2015). Estimation of seismic building structural types using multi-sensor remote sensing and machine learning techniques. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:175–188.

[20] Geiß, C. and Taubenböck, H. (2013). Remote sensing contributing to assess earthquake risk: from a literature review towards a roadmap. *Natural Hazards*, 68(1):7–48.

[21] Geiß, C., Taubenböck, H., Wurm, M., Esch, T., Nast, M., Schillings, C., and Blaschke, T.

(2011). Remote sensing-based characterization of settlement structures for assessing local potential of district heat. *Remote Sensing*, 3(7):1447–1471.

[22] Gönen, M. and Alpaydın, E. (2011). Multiple kernel learning algorithms. *Journal of machine learning research*, 12(Jul):2211–2268.

[23] Gong, F.-Y., Zeng, Z.-C., Zhang, F., Li, X., Ng, E., and Norford, L. K. (2018). Mapping sky, tree, and building view factors of street canyons in a high-density urban environment. *Building and Environment*, 134:155–167.

[24] Gonzalez, D. and Acevedo, A. B. (2017). Actualización del modelo de exposición sísmica para viviendas en medellín (colombia) y su aplicación en la evaluación del riesgo sísmico para viviendas de mampostería no reforzada. In *VIII Congreso Nacional de Ingeniería Sísmica*, Barranquilla, Colombia.

[25] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

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

[27] Hecht, R., Meinel, G., and Buchroithner, M. (2015). Automatic identification of building types based on topographic databases–a comparison of different data sources. *International Journal of Cartography*, 1(1):18–31.

[28] Henn, A., Römer, C., Gröger, G., and Plümer, L. (2012). Automatic classification of building types in 3d city models. *GeoInformatica*, 16(2):281–306.

[29] Hu, C.-B., Zhang, F., Gong, F.-Y., Ratti, C., and Li, X. (2020). Classification and mapping of urban canyon geometry using google street view images and deep multitask learning. *Building and Environment*, 167:106424.

[30] Iannelli, G. C. and Dell'Acqua, F. (2017). Extensive exposure mapping in urban areas through deep analysis of street-level pictures for floor count determination. 1(16).

[31] Jaiswal, K. and Wald, D. J. (2008). Creating a global building inventory for earthquake loss assessment and risk management. Technical report, U.S. Geological Survey Open-File Report 2008-1160.

[32] Jung, A. B., Wada, K., Crall, J., Tanaka, S., Graving, J., Reinders, C., Yadav, S., Banerjee, J., Vecsei, G., Kraft, A., Rui, Z., Borovec, J., Vallentin, C., Zhydenko, S., Pfeiffer, K., Cook, B., Fernández, I., De Rainville, F.-M., Weng, C.-H., Ayala-Acevedo, A., Meudec, R., Laporte, M., et al. (2020). imgaug. `https://github.com/aleju/imgaug`. Online; accessed 01-Feb-2020.

[33] Kang, J., Körner, M., Wang, Y., Taubenböck, H., and Zhu, X. X. (2018). Building instance classification using street view images. *ISPRS journal of photogrammetry and remote sensing*, 145:44–59.

[34] Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image

descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

[35] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[36] Lang, D. H., Kumar, A., Sulaymanov, S., and Meslem, A. (2018). Building typology classification and earthquake vulnerability scale of central and south asian building stock. *Journal of Building Engineering*, 15:261–277.

[37] LeCun, Y., Touresky, D., Hinton, G., and Sejnowski, T. (1988). A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann.

[38] León Torres, J. and Ordaz, M. (2015). Modelo de exposiciÓn top-down a nivel urbano.

[39] Nefian, A. V., Liang, L., Pi, X., Xiaoxiang, L., Mao, C., and Murphy, K. (2002). A coupled hmm for audio-visual speech recognition. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II–2013. IEEE.

[40] Nex, F., Rupnik, E., and Remondino, F. (2013). Building footprints extraction from oblique imagery. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 2:61–66.

[41] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep

learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696.

[42] Orford, S. and Radcliffe, J. (2007). Modelling uk residential dwelling types using os mastermap data: A comparison to the 2001 census. *Computers, Environment and Urban Systems*, 31(2):206–227.

[43] Ouyang, W., Chu, X., and Wang, X. (2014). Multi-source deep learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2329–2336.

[44] Owens, A., Isola, P., McDermott, J., Torralba, A., Adelson, E. H., and Freeman, W. T. (2016). Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413.

[45] Pittore, M. and Wieland, M. (2013). Toward a rapid probabilistic seismic vulnerability assessment using satellite and ground–based remote sensing. 68:115–145.

[46] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.

[47] Restrepo, L. F., Villarraga, M. R., Jaramillo, J. D., Farbiarz, Y., Vélez, A. F., Rendón, D. A., and Ángel, F. P. (2007). *Microzonificación y evaluación del riesgo sísmico del Valle de Aburrá*. Área Metropolitana del Valle de Aburrá.

[48] Riedel, I., Guéguen, P., Dalla Mura, M., Pathier, E., Leduc, T., and Chanussot, J. (2015). Seismic vulnerability assessment of urban environments in moderate-to-low seismic hazard regions using association rule learning and support vector machine methods. *Natural Hazards*, 76(2):1111–1141.

[49] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[50] Sester, M. (2000). Knowledge acquisition for the automatic interpretation of spatial data. *International Journal of Geographical Information Science*, 14(1):1–24.

[51] Silva, V., Crowley, H., Pagani, M., Monelli, D., and Pinho, R. (2014). Development of the openquake engine, the global earthquake model's open-source software for seismic risk assessment. 72.

[52] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[53] Steiniger, S., Lange, T., Burghardt, D., and Weibel, R. (2008). An approach for the classification of urban building structures based on discriminant analysis techniques. *Transactions in GIS*, 12(1):31–59.

[54] Steinkraus, D., Buck, I., and Simard, P. (2005). Using gpus for machine learning algorithms. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 1115–1120. IEEE.

[55] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

[56] Torabi, A., Pal, C., Larochelle, H., and Courville, A. (2015). Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*.

[United Nations and Affairs.] United Nations, D. o. E. and Affairs., S. 2018 revision of world urbanization prospects. `https://www.un.org/development/desa/publications/2018-revision-of-world-urbanization-prospects.html`. Fecha de Consulta: 2018-10-03.

[58] Vakalopoulou, M., Karantzalos, K., Komodakis, N., and Paragios, N. (2015). Building detection in very high resolution multispectral data with deep learning features. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1873–1876. IEEE.

[59] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

[60] Wang, D., Cui, P., Ou, M., and Zhu, W. (2015). Deep multimodal hashing with orthogonal regularization. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[61] Werder, S., Kieler, B., and Sester, M. (2010). Semi-automatic interpretation of buildings

and settlement areas in user-generated spatial data. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 330–339.

[62] Weston, J., Bengio, S., and Usunier, N. (2011). Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

[63] Wurm, M., Taubenbock, H., Roth, A., and Dech, S. (2009). Urban structuring using multisensoral remote sensing data: By the example of the german cities cologne and dresden. In *2009 Joint Urban Remote Sensing Event*, pages 1–8. IEEE.

[64] Yamazaki, F., Mitomi, H., Matsuoka, M., and Honda, K. (2000). Inventory development for natural and built environments-remote sensing technologies for inventory development and risk assessment-characteristics of satellite images in bangkok, thailand. Technical report, NIED, Miki, Hyogo, Japan, Tech. Rep. on The Development of Earthquake and Tsunami Mitigation Technologies and their Integration for the Asia-Pacific Region.

[65] Yepes-Estrada, C., Silva, V., Valcárcel, J., Acevedo, A. B., Tarque, N., Hube, M. A., Coronel, G., and Santa María, H. (2017). Modeling the residential building inventory in south america for seismic risk assessment. *Earthquake Spectra*, 33(1):299–322.

[66] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

[67] Yuhas, B. P., Goldstein, M. H., and Sejnowski, T. J. (1989). Integration of acoustic and visual speech signals using neural networks. *IEEE Communications Magazine*, 27(11):65–71.

[68] Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

[69] Zhang, Y., Burton, H. V., Sun, H., and Shokrabadi, M. (2018). A machine learning framework for assessing post-earthquake structural safety. *Structural Safety*, 72:1–16.

[70] Zhao, W., Du, S., and Emery, W. J. (2017). Object-based convolutional neural network for high-resolution imagery classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(7):3386–3396.

[71] Zhao, W., Guo, Z., Yue, J., Zhang, X., and Luo, L. (2015). On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery. *International Journal of Remote Sensing*, 36(13):3368–3379.

[72] Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., and Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36.

[73] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Apéndices

Apéndice A. Arquitectura de Vgg16 y Vgg19

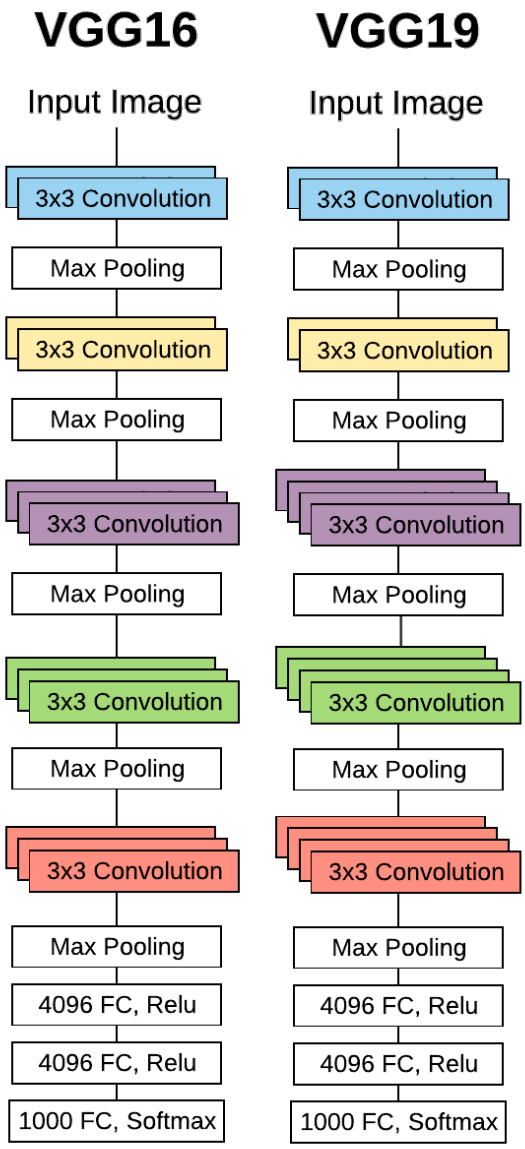


Figura 32. Arquitectura de Vgg16 (Izq) y Vgg19 (Der) (52)

Apéndice B. Arquitectura de InceptionV3

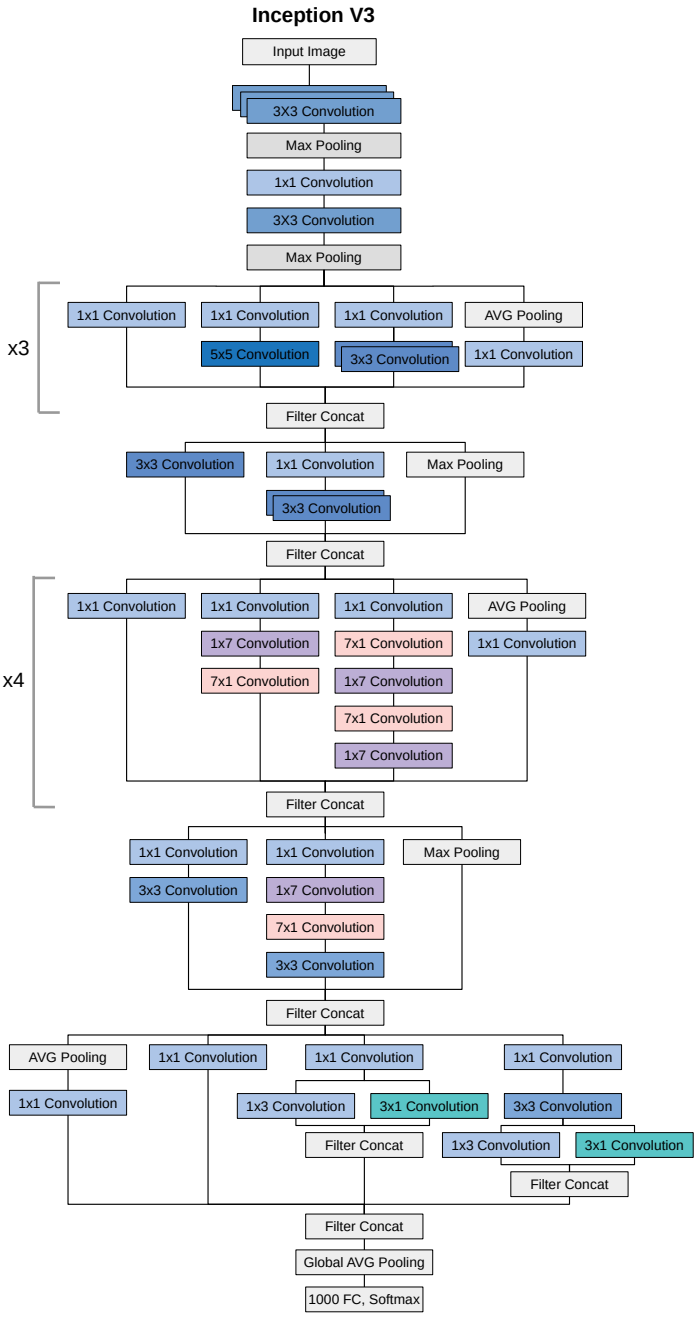


Figura 33. Arquitectura de InceptionV3 (55)

Apéndice C. Arquitectura de ResNet50

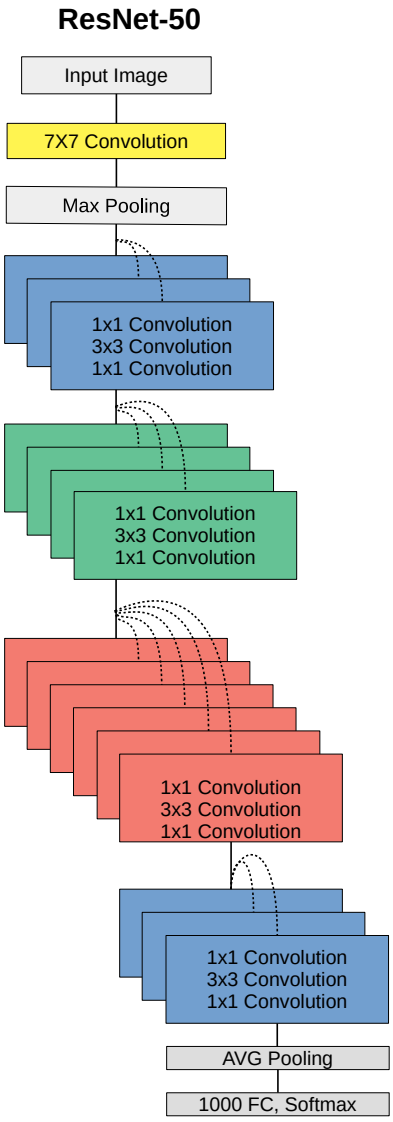


Figura 34. Arquitectura de ResNet50 (26)

Apéndice D. Arquitectura de Xception

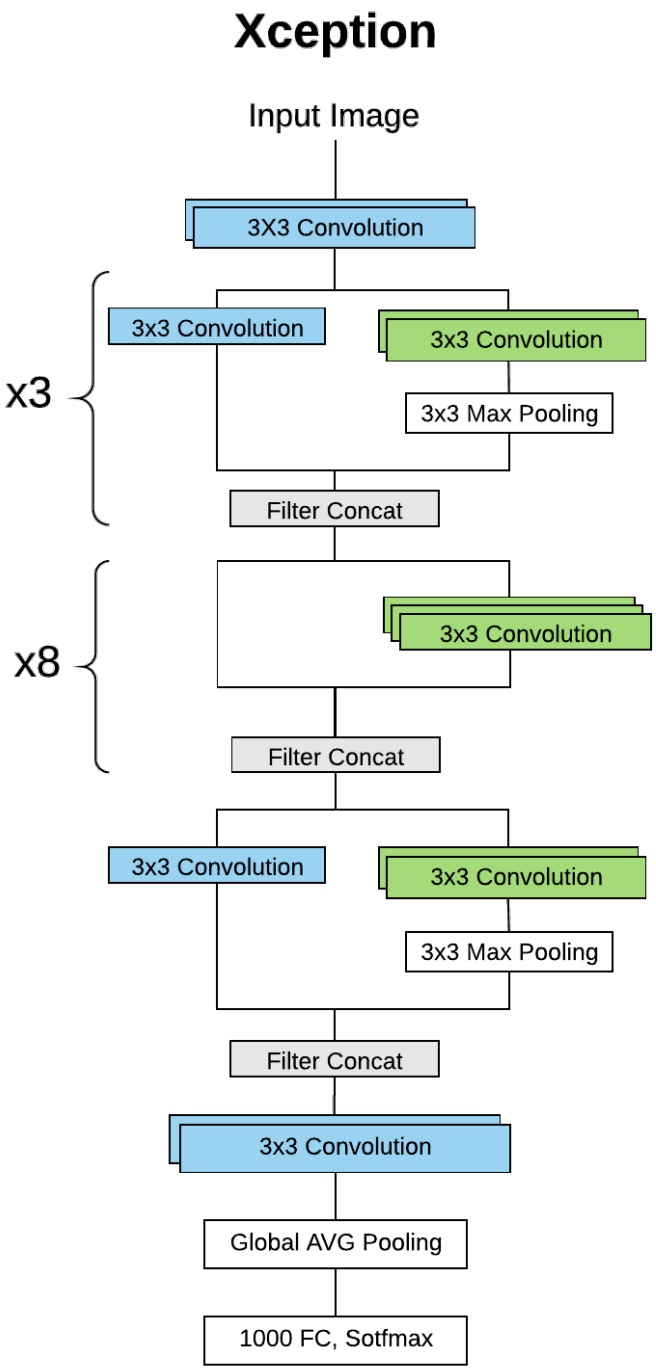


Figura 35. Arquitectura de Xception (15)