

# **APRENDIZAJE PROFUNDO APLICADO A CLASIFICACIÓN DE ESCENAS DE PROPIEDADES**

**Ignacio Gabriel Franco**

Trabajo final de grado presentado al Departamento de Matemática Aplicada de la Universidad Católica de Santiago del Estero para optar al grado académico de Ingeniería en Informática.

Diciembre 2019  
Rafaela, Argentina

**Director: Mariano Ferrero**

# 1 Introducción

## 1.1 Contexto

De la mano del avance tecnológico tanto en materia de hardware como de software, en los últimos años ha sido posible explotar de forma más efectiva y eficiente una rama algo olvidada de la inteligencia artificial: las redes neuronales.

Esta rama ha demostrado en múltiples ocasiones ser capaz de obtener resultados significativos en tareas de detección y localización de objetos, clasificación de escenas, segmentación de imágenes y detección de rostros (entre otras).

Estas prácticas tienen una gran aplicación en la industria como la detección y localización de entidades relacionadas al tráfico para autos que se conducen por si mismos, detección de rostros para controles de ingreso de personas a aeropuertos como también a empresas, detección de elementos aplicado a imágenes médicas (posibles tumores o malformaciones).

En este trabajo se hará frente a uno de los ejes inicialmente mencionados: la clasificación de escenas. Esta asignatura que resulta prácticamente trivial para una persona incluye un conjunto de actividades complejas por si mismas: detección de objetos locales y su disposición dentro de la escena, entorno de fondo, distinción de características entre escenas parecidas, cultura de la que proviene la escena, especificidad a la hora de clasificar, y muchas más. Actualmente, mediante actividades de investigación, competiciones y necesidades del sector privado se han logrado significativos resultados en clasificación de escenas relacionadas a diferentes contextos: distinción entre lugares de una ciudad, clasificación de zonas de la misma a partir de imágenes satelitales, ambientes de una propiedad, escenas relacionadas al tráfico de una ciudad, etc.

A priori, en una fugaz interpretación de la tarea a realizar, se la podría definir como la clasificación de imágenes tradicional, aunque no sea así. En esta actividad se puede destacar tanto detección de objetos como, en algunos casos, la definición de su contorno, sin importar el tamaño del mismo o su posicionamiento dentro de la imagen. La tarea de reconocer escenas acapara varias otras aristas, como son la disposición de los objetos en la imagen, los elementos que se encuentren en la misma, el ambiente en el que se encuentren, el fondo y muchas más. En una escena existen múltiples objetos en diferentes escalas, enfocados desde diferentes ángulos y disposiciones, mientras que en

la clasificación de imágenes se suele tratar con un único objeto centrado. Éstas son, entre muchas otras, algunas de las principales diferencias entre clasificación de escenas e imágenes, dos tareas que pertenecen a un mismo tópico pero que no es posible solucionarlas totalmente utilizando el mismo enfoque para el problema.

Dado que hasta hace no muchos años la cantidad de imágenes a clasificar no tenían la inmensitud actual, queda claro que era posible de abordar la necesidad mediante tareas realizadas por individuos. Siendo que actualmente el flujo de información es mucho mayor, resulta que la automatización del etiquetado imágenes pasó a ser un requerimiento para determinados entornos como empresas de venta y/o alquiler de propiedades, intermediarios dentro de la misma actividad o sitios en los que se suben imágenes de este tipo y se las quiere mantener etiquetadas de forma inmediata.

Dentro de los posibles enfoques a utilizar descriptos en [y otros, 2018a] se encuentran las representaciones esparsas, máquinas de soporte vectorial, redes neuronales artificiales y redes neuronales convolucionales. Dentro de las redes neuronales se suelen usar diferentes arquitecturas en calidad de obtener los mejores resultados, dependiendo de en qué manera se estructure la información. Estas arquitecturas son las Redes Neuronales Convolucionales, las Redes Neuronales Recurrentes y el Aprendizaje por Transferencia.

## 1.2 Motivación

La clasificación de imágenes relacionadas a propiedades inmuebles hace referencia a la capacidad de etiquetar automática y correctamente imágenes de escenas relacionadas a diferentes partes de los mismos de manera tal que luego sea posible consumir la información a cada sector de manera individual por cada bien. Una actividad que resulta altamente atractiva y beneficiosa cuando se trata con cientos o miles de imágenes de propiedades y se quiere explotar esta información para otro tipo de tareas. Dentro de los beneficios más destacables se pueden mencionar la automatización de tarea de etiquetado, las mejoras en sistemas que requieran este tipo de tareas, el ahorro de tiempo para clasificar imágenes de este dominio, entre otras.

En el contexto actual existen empresas que brindan una larga lista de servicios relacionados a las propiedades inmuebles y que la resolución de este problema les sería de gran ayuda tanto en las tareas del día a día como para explotar de mejor manera la información de los inmuebles que ya tienen almacenada internamente. Este tipo de empresas u organizaciones son las

que se dedican a actividades como: la venta y alquiler de bienes propios, la intermediación entre residentes y dueños para alquileres temporales, la valuación y control del estado de las propiedades, entre otras. Dentro de las posibles aplicaciones y ventajas que puede otorgar el adjudicarse con un modelo que se dedique a realizar esta actividad se destaca principalmente la clasificación automática como tal, pero también la verificación de diferentes escenas requeridas para publicar un bien inmobiliario en algún sitio de ventas o alquileres, la extracción de características relevantes de cada tipo de escena para reutilizar en otro tipo de modelo, la recomendación de qué imágenes se deberían subir para lograr un buen posicionamiento de un inmueble en un sitio o sugerencias sobre qué imágenes logran mejores resultados a la hora de vender o alquilar una propiedad, la valoración de propiedades a través de sus imágenes, el refinamiento de valoraciones realizadas mediante datos estructurados gracias al contenido de las imágenes de las propiedades, la agrupación de las mismas según las características de los diferentes sectores con los que cuenta, recomendaciones de propiedades basados en características elegidas por un usuario que luego sean traducidas a información contenida en cada tipo de escena, y muchas otras aplicaciones más que se pueden encontrar o se encontrarán en un futuro próximo en el mercado actual.

Dentro de la previamente mencionada tarea de valuación de inmuebles existen cientos de factores que componen el valor final de los mismos, tales como el terreno total ocupado y el construido, los años desde su construcción, la cantidad de habitaciones de cada tipo, los precios de los inmuebles circundantes y los precios de los inmuebles similares, el estado del inmueble, entre otros. Dada la complejidad de la cantidad de información a tener en cuenta y las diferentes formas en que se presenta (variables numéricas y categóricas, datos no estructurados como imágenes, etc), en conjunto con el alto número de inmuebles que se necesitan tasar en algunos casos se han implementado diferentes enfoques de valoración automática. En [y otros, 2017b] se demuestra que a partir de la información que brindan las imágenes de los inmuebles y mediante aprendizaje profundo es posible mejorar los resultados de los enfoques que no cuentan con esta información y dedican su esfuerzo a realizar las tasaciones sólo con la información tabular y estructurada. Teniendo en cuenta este último punto, queda claro que poder absorber la mayor cantidad de información sobre estas imágenes resulta una tarea de alta significancia. Uno de los enfoques utilizados para hacerlo es basar la estimación del precio únicamente a partir de las imágenes que se tienen de la propiedad, y luego agregar los datos estructurados para mejorar los resultados.

Por otro lado, en actividades como el control o chequeo de imágenes que se suben al hacer publicaciones de propiedades o las sugerencias de escenas a subir dentro de las mismas, reconocer de qué habitación o vista se trata en cada caso resulta la única manera de brindar esta información o solicitud al usuario.

Si de la agrupación de imágenes similares se trata, entonces luego de obtener la relación entre precio con el que se venden las propiedades y las características de las diferentes partes de las mismas sería posible conocer qué particularidades tienen aquellas propiedades que logran mayor margen de ganancia al momento de venderlas, qué se puede hacer con ciertas partes del inmueble para que su valor suba en relación a las cualidades de sus diferentes sectores y otras tareas relacionadas que brindarían un alto valor al momento de postular a la venta una propiedad.

Por último, no está de más mencionar la posibilidad de extraer el conocimiento adquirido por un modelo encargado de hacer estas tareas para hacer frente o refinar resultados de otros modelos, que no necesariamente estén fuertemente relacionados con la clasificación de escenas de propiedades inmuebles.

Como se mostrará a partir de la revisión de antecedentes, la utilización de aprendizaje profundo para clasificar escenas es una opción altamente viable por los resultados que se obtienen. Dependiendo del dataset utilizado y la arquitectura de la red, es posible lograr salidas que logran una performance apta para su implementación en entornos productivos y con altos requerimientos. Tanto es así que, aunque los resultados que se consiguen son buenos, aún es necesario seguir investigando para lograr llegar a un estado en el cual se defina que una arquitectura que bajo ciertos parámetros y a partir de un determinado dataset alcance una exactitud por encima del 99% de todos los casos con los que se encuentre a posteriori. Este trabajó hará foco en esta problemática, a través de la investigación de diferentes métodos de aprendizaje profundo que actualmente alcanzan el estado del arte, utilizando métricas que ya se utilicen en este tópico y conjuntos de datos para que tanto la diversidad como la densidad de las escenas sea la suficiente para brindar conclusiones correctas con respecto a esta tarea.

### 1.3 Estructura del documento

En este trabajo se investigarán diferentes métodos de clasificación de escenas aplicado a propiedades inmuebles. Con el fin de alcanzar el estado del arte en esta tarea y eventualmente intentar mejorar estos resultados, la composición

del trabajo será la siguiente: en el capítulo segundo se realizará una revisión de antecedentes en materia de utilización de técnicas de aprendizaje profundo para clasificar escenas. Por parte del capítulo tercero se definirá el marco teórico y otros conceptos necesarios para entender el funcionamiento de este tipo de técnicas. En el capítulo cuarto se elegirán las limitaciones y mostrarán las hipótesis del proyecto. En los capítulos quinto y sexto de definirán los experimentos a realizar y sus resultados, respectivamente. Para finalizar, en el capítulo séptimo, se declararán las conclusiones del trabajo y posibles futuros trabajos.

## 2 Marco teórico

### 2.1 Aprendizaje automático

Dentro de las diferentes ramas de la inteligencia artificial, los algoritmos utilizados se pueden clasificar en tres grandes ramas según la forma en que son entrenados: algoritmos de aprendizaje supervisado, algoritmos de aprendizaje semi-supervisado y algoritmos de aprendizaje no supervisado.

Los primeros se utilizan cuando para un conjunto de entrenamiento  $X$  se tiene un conjunto de etiquetas  $y$  relacionadas a elemento; de esta manera, los algoritmos encuadrados dentro de esta rama se encargan de estimar una función que mapee los datos del conjunto  $X$  al conjunto  $y$ , con el fin de luego poder realizar inferencias a partir de sólo nuevos casos de entrada.

Los algoritmos de aprendizaje no supervisado se utilizan en situaciones en las que se tiene un conjunto de datos de entrada  $X$  pero no un conjunto de etiquetas asignadas a estos datos, estos algoritmos se emplean para realizar sobre los datos tareas tales como agrupamientos, detección de anomalías, reducción de dimensionalidad, entre otras.

Por último, el aprendizaje semi-supervisado entra en juego en aquellos casos en los que se tiene un pequeño conjunto de datos etiquetado, y un gran conjunto de datos no etiquetados. Existen modelos en los que agregar datos no etiquetados al entrenamiento hace que mejoren sus predicciones, como otros en los que hace que empeoren. Las ideas principales detrás del aprendizaje supervisado son: ayudar a ganar representación general de la población de los datos, y ganar conocimiento sin recurrir a los costos de tiempo y dinero que suponen etiquetar datos que se tienen.

En este trabajo se contará tanto con un conjunto de entrenamiento (que serán bases de datos de escenas de propiedades) como con sus pertenecientes etiquetas, por lo que se intentará mapear mediante técnicas de aprendizaje profundo cada imagen con sus etiquetas, es decir, se trabajará en un problema de aprendizaje supervisado.

Dentro de esta rama existen otros algoritmos que pueden servir para este problema o para otros de dominio diferente como son los árboles de decisión, regresiones lineales o logísticas, ensembles, etc.

## 2.2 Representación de una neurona

En este trabajo, como bien se explicó anteriormente, se abordará la solución mediante técnicas de aprendizaje profundo, por lo que a continuación se dará lugar a las explicaciones pertinentes relacionadas a este tipo de técnica. Para empezar es necesario comentar que el término "profundo" recae en la profundidad de las redes neuronales que se utilizan en esta técnica. Estas redes pueden estar compuestas hasta por millones de neuronas interconectadas entre si; neuronas que se representan como se observa en la imagen 1 donde se muestra un ejemplo con  $n$  entradas, conformada por:

- Entradas: conjunto de datos de entrada  $x_1..x_n$
- Pesos: conjunto de pesos  $w_1..w_n$  correspondientes a cada entrada
- Función de agregación: función que sumariza la multiplicación pesada entre cada entrada  $x$  y su correspondiente peso  $w$ .
- Función de activación: función no lineal responsable de mapear el resultado de la función de agregación en salidas (según que tipo de función sea, los resultados suelen variar entre  $[0, 1]$  o  $[-1, 1]$ ).
- Salida: resultado de la función de activación.

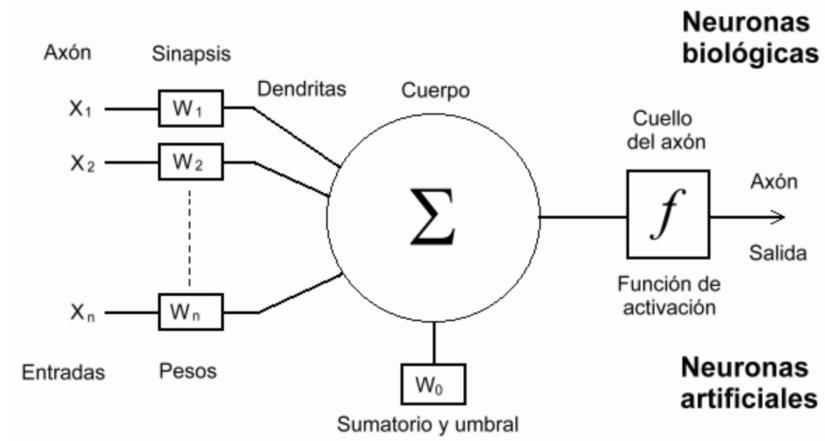


Figure 1: Representación de una neurona

De esta manera, se puede obtener la salida de la neurona  $Y$ , a partir de aplicar la función de activación  $\sigma$  a la suma entre el sesgo  $b$  y la multiplicación matricial del conjunto de entradas  $X$  y los pesos  $W$ .

$$Y = \sigma(W^T X + b) \quad (1)$$

## 2.3 Red Neuronal

Una red neuronal, como su nombre lo enuncia, es una consecución de capas de  $N$  neuronas en cada una, conectadas entre sí como se puede ver en el ejemplo de la figura 2. Es necesario que se cuente con capas de entrada, ocultas y de salida; para el ejemplo enunciado se cuenta con una capa de entrada, dos capas ocultas y una capa de salida.

El fin de las redes neuronales es aprender representaciones del contenido de la información en relación a las salidas esperadas para luego poder hacer inferencias en nuevo contenido, a modo de ejemplo, si se tienen imágenes de perros y gatos, y el fin es detectar si se trata de un perro o un gato la red posiblemente no aprenda las mismas representaciones que si el fin de la misma es detectar presencia o ausencia de animales.

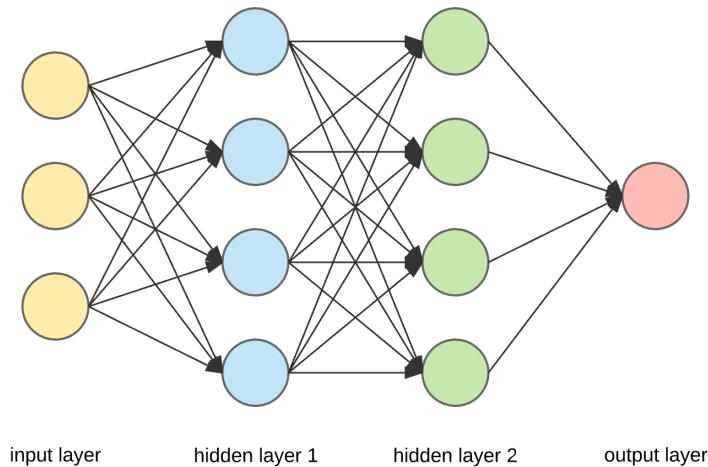


Figure 2: Perceptrón Multi Capa

Para explicar el funcionamiento de las redes neuronales es necesario también introducir algunos conceptos como son la propagación hacia adelante y hacia atrás, función de pérdida y optimización. Durante la fase de entrenamiento,

los pasos que se siguen son: (i) inicialización de pesos y sesgo, (ii) propagación hacia adelante, (iii) evaluación de función de pérdida y (iv) propagación hacia atrás.

Antes de comenzar a entrenar una red es necesario inicializar los pesos que conectan las capas, suele hacerse de manera aleatoria a partir de una distribución normal, aunque existen otros métodos como hacerlo a partir de una distribución uniforme, la inicialización de Xavier, entre otras.

La propagación hacia adelante es la actividad en que se le brindan los datos a la capa de entrada de la red para que se evalúen consecutivamente todas las capas siguientes en esa dirección aplicando  $\sigma$  en cada una, tomando como entrada la capa inmediata anterior. De esta manera, se obtiene como salida de la red sus predicciones.

$$z = W^T X + b \quad (2)$$

Luego, a partir de las salidas  $\hat{y}$  de la red es posible medir la función de pérdida (detallado en 3) que brinda la información de cuán cerca estuvo la red de predecir las salidas correctas y con la que se puede computar la propagación hacia atrás del error, es decir, optimizar la función de pérdida actualizando los pesos desde atrás hacia adelante teniendo en cuenta el error en cada capa. La manera de hacerlo es mediante la derivada parcial de la función de costo con respecto tanto a los pesos  $\omega$  (fórmula 4) como a los sesgos  $b$  (fórmula 5) en cada capa con el fin de obtener un nuevo valor de los pesos y sesgos  $W$  y  $b$  en cada una de éstas.

$$\mathcal{L}(y, \hat{y}) = - \sum_{i=1}^n y_i \log \hat{y}_i \quad (3)$$

$$\omega = \omega - \alpha \frac{\partial J(\omega, b)}{d\omega} \quad (4)$$

$$b = b - \alpha \frac{dJ(\omega, b)}{db} \quad (5)$$

La función de pérdida se define de diferentes maneras según el problema que se esté tratando, ya sea error cuadrático medio, error absoluto medio, proximidad del coseno, etc. La propuesta general para problemas de clasificación es usar *entropía – cruzada* (o *cross – entropy* en inglés), una función derivable que mide qué tan lejos del resultado esperado están las salidas de la red, como se observa en la ecuación 3. Como se mencionó anteriormente, el

objetivo de la red es minimizar la función de pérdida y para esto se requiere de una función de optimización. *Adam*, introducido en [Kingma and Ba, 2014] es un optimizador en que se puede tener ciertas certezas de su correcto funcionamiento ya que su implementación tiene en cuenta las principales mejoras de otros como son el Descenso por el Gradiente con Momentum y la Propagación de la Raíz Cuadrada de la Media de los Cuadrados.

Un problema común al entrenar una red es el sobreentrenamiento, que se da cuando la red sobreajusta sus pesos para el conjunto de entrenamiento y pierde capacidad de generalizar. Una forma fácil de controlarlo es mediante la verificación de la función de pérdida y las métricas elegidas sobre un subconjunto de validación, que son imágenes que no se utilizan para entrenar la red sino que para medir cómo está funcionando la misma (si la red está sobreentrenando, entonces la diferencia entre las métricas de entrenamiento y validación comenzarán a incrementarse, siendo las de entrenamiento mayores a las otras). Para contrarrestar estas situaciones existen capas regularizadoras, es decir, que evitan la concentración de gran porcentaje del conocimiento en un subconjunto de unidades en las capas y por ende el aprendizaje de la representación del contenido sea compartido entre las neuronas de las capas. Aunque existen otras, las más utilizadas son *dropout* y *batch-normalization*. La capa *dropout* se encarga de apagar conexiones aleatoriamente entre capas durante el entrenamiento, esto hace que no se entrene siempre con las mismas unidades dentro de cada capa, y así, que no sean siempre las mismas neuronas las que se activen frente a cada situación. La capa de *batch-normalization* aplica una función de regularización que sale de la media y la varianza de cada sampleo durante el entrenamiento, agregando ruido en las activaciones y evitando que las mismas no sean demasiado altas ni bajas, reduciendo de esta manera el desplazamiento covariante interno (*internal covariance shift*, en inglés), y la manera de hacerlo es mediante la aplicación de las ecuaciones 6, 7, 8, 9, como se explica en el Algoritmo 1 de [Ioffe and Szegedy, 2015].

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad //media\ del\ mini-batch \quad (6)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad //varianza\ del\ mini-batch \quad (7)$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad //normalizacion \quad (8)$$

$$y_i = \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) // \text{escalacion y desplazamiento} \quad (9)$$

Las capas convolutivas y las capas totalmente conectadas requieren de una función de activación para determinar su salida, y aunque existe gran variedad de estas, hay algunas funciones que se sabe que funcionan bien ante determinados problemas. A continuación se detallarán algunas de las funciones de activación que posiblemente se utilicen a en las etapas de experimentación:

- **Sigmoide (*Sigmoid*):** Está definida como 10 y la razón principal por la que se la utiliza es debido a que sólo devuelve valores entre [0, 1], por lo que es muy útil en tareas en las que se necesita predecir la probabilidad de una clase.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

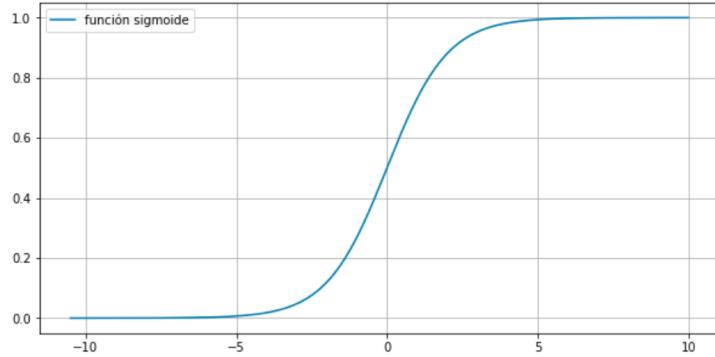


Figure 3: Función Sigmoide

- **Tangente hiperbólica (*TanH*):** Su formulación se explicita en 11. Como se puede observar en la figura 4, mantiene la forma S de la función Sigmoide pero en este caso los resultados posibles van desde [-1, 1].

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (11)$$

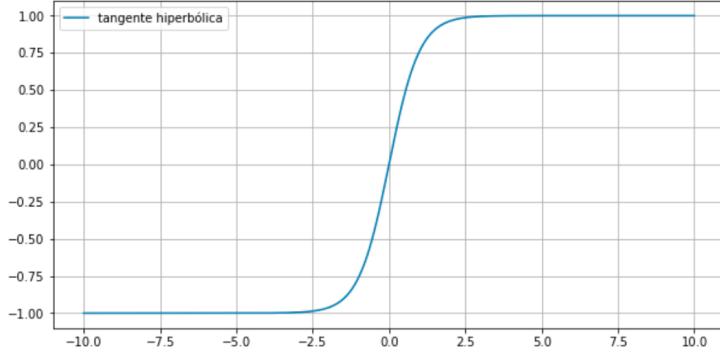


Figure 4: Función Tangente Hiperbólica

- Unidad lineal rectificada (*ReLU*): En la figura 5 se puede observar el comportamiento de esta función (definida en la fórmula 12), en donde para valores negativos el resultado es cero, mientras que para el resto, el resultado es el mismo valor. Por este comportamiento es una de las funciones de activación más utilizadas (junto a sus otras variantes eLu, LeakyReLu, etc) en las capas ocultas de las redes neuronales.

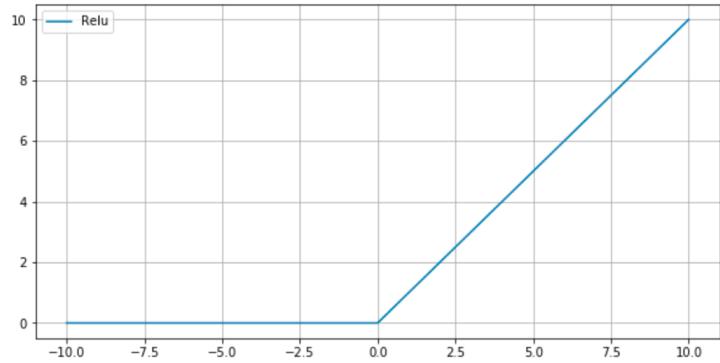


Figure 5: Función ReLu

$$ReLU(x) = \max(0, x) \quad (12)$$

- Exponencial normalizada (*Softmax*): es una variante de la función Sigmoide pero definida para problemas de clasificación multiclase; como se observa en la fórmula 13, normaliza las entradas en una salida que es

en un vector de  $K$  valores de probabilidad (uno por cada clase definida) entre  $[0, 1]$  que sumados dan uno.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{por cada } j = 1, \dots, K \quad (13)$$

Siendo  $K$  la cantidad de clases.

## 2.4 Red Neuronal Convolucional

Las redes neuronales convolucionales son una variante de las redes neuronales en las que la extracción de características se realiza mediante capas convolucionales. Fueron introducidas en [LeCun et al., ] por LeCun y Bengio, investigadores referentes en el mundo del aprendizaje automático y las redes neuronales, y han transformado la forma de resolver problemas de clasificación de imágenes (y otros ) desde entonces.

Como es posible observar en la figura 6, una convolución es una operación matemática. Se puede entender como el resultado de aplicar un filtro (una matriz con forma  $N \times N$ ) en todas las regiones de la imagen con el fin de obtener una nueva.

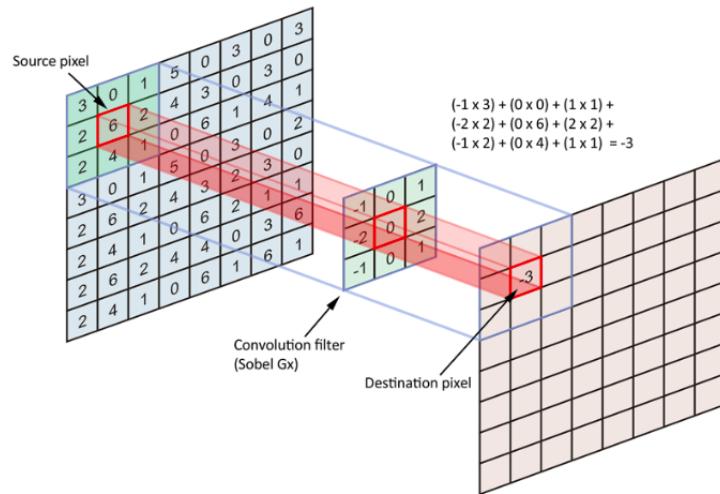


Figure 6: Convolución

$$\begin{aligned}
f = & (-1 * 3) + (0 * 0) + (1 * 1) \\
& + (-2 * 2) + (0 * -6) + (2 * 2) \\
& + (-1 * 2) + (0 * 4) + (1 * 1) = -3
\end{aligned} \tag{14}$$

En el ejemplo se aprecia que el resultado de la aplicación del filtro en la esquina superior izquierda de la imagen se obtiene a partir de multiplicar cada valor del pixel en la imagen de entrada con su par en la misma posición en el filtro, finalmente en el ejemplo se tiene que el valor del nuevo pixel en la imagen de salida se obtiene como se observa en la formulación 14. La operación detallada previamente se aplica a toda la imagen para obtener la imagen resultado, teniendo en cuenta tanto los valores del *stride* como del *padding* definidos para la capa.

El parámetro *stride* define cada cuántos píxeles de la imagen de entrada se aplicará el filtro, mientras que mediante el parámetro *padding* se decide cómo se tratarán los bordes, esto es debido a que existen ocasiones en las que se desea obtener una imagen en la que se le apliquen convoluciones también a los bordes de la misma; las formas más comunes de hacerlo son agregando a los bordes  $N$  píxeles más (dependiendo del *stride*), ya sea imputándolos como ceros o con mismo valor de sus píxeles más cercanos.

Gracias a estas capas las redes convolucionales son capaces de aprender representaciones del contenido invariantes, ya que cada filtro se aplica a todas las subregiones de la imagen haciendo que el aprendizaje obtenido en las capas más profundas se base en las imágenes completas. A continuación se detallará cómo funciona una red convolucional, con el fin de mostrar cómo estas las capas convolutivas se interconectan entre sí, reducen las representaciones computadas y finalmente se conectan con capas totalmente conectadas para realizar inferencia a partir de lo aprendido.

Existen muchas configuraciones de redes neuronales convolucionales, aunque para simplificar la explicación se utilizará una red compuesta de la siguiente manera: capa convolucional, capa de pooling, capa convolucional, capa de pooling, capa totalmente conectada y capa de salida (ilustrada en la figura 7). La capa de entrada en este caso (al igual que en el presente trabajo) serán imágenes en formato RGB normalizadas, es decir, una matriz de  $[alto \times ancho \times n. \ canales]$  por cada imagen; luego esta entrada será convolucionada aplicando la operación matemática previamente explicada y sus salidas serán agrupadas por una capa llamada *pooling*. En esta capa se

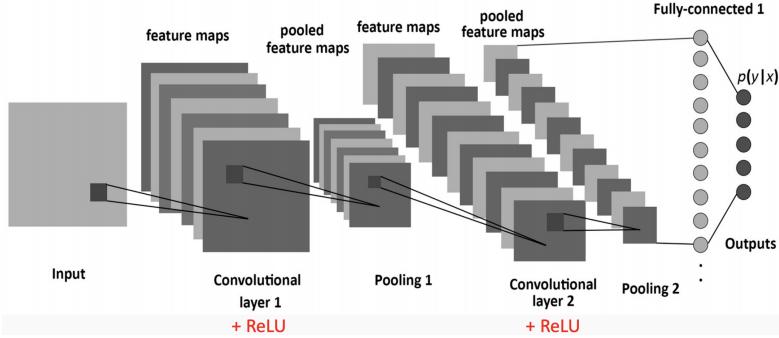


Figure 7: Ejemplo Red Neuronal Convolucional

buscará reducir tanto las representaciones aprendidas como la cantidad de cálculos a realizar: se agruparán mediante la aplicación de un filtro  $f$  y un método  $M$  los resultados de la convolución, siendo  $f$  una matriz (usualmente de  $[2 \times 2]$ ) y  $M$  la utilización de la media o el máximo de los valores de la imagen en cada sector en que se aplique el filtro; al igual que en la convolución, la operación de pooling se aplica en toda la imagen, aunque tiene dos diferencias fundamentales: no tiene parámetros que aprender o ajustar y busca reducir las representaciones a matrices de menor orden. Continuando con la red, se aplica nuevamente una convolución seguida de una capa de pooling que se conecta a una capa totalmente conectada (la misma que se utiliza en el perceptrón multicapa). Finalmente, la capa de totalmente conectada se conecta con una capa de salida que se encargará de computar las probabilidades con las que se predice cada clase (básicamente, otra capa totalmente conectada con función de activación Softmax, mencionada en 2.3).

## 2.5 Redes preentrenadas y Aprendizaje por transferencia

Como entrenar redes neuronales profundas resulta muy costoso computacionalmente tanto por la profundidad de las mismas como por la cantidad de datos con las que se las necesita entrenar, con el pasar del tiempo quienes tienen acceso a capacidad de cómputo comenzaron a liberar redes entrenadas con diferentes bases de datos. Posiblemente, las redes liberadas más conocidas sean las que se entrenaron utilizando ImageNet [Russakovsky et al., 2015], una base de datos de 3.2 millones imágenes centradas en objetos, aunque

también existen otras redes preentrenadas con otras bases de datos de imágenes abiertas como son COCO Dataset [Ronchi and Perona, 2015], Places Dataset [y otros, 2014], etc.

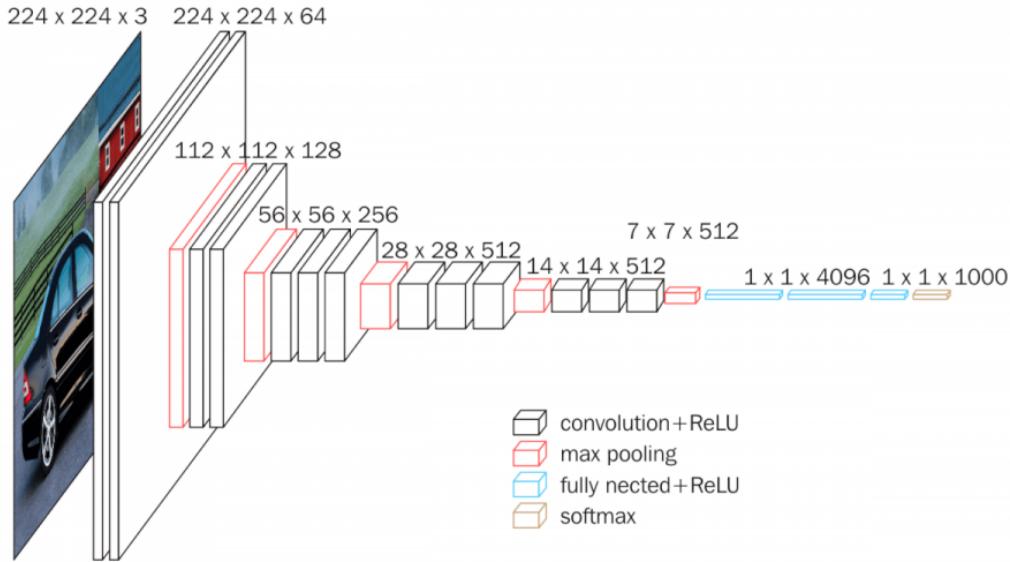


Figure 8: Arquitectura VGG16

En la actualidad existen redes con diferentes arquitecturas preentrenadas utilizando diversos conjuntos de datos como son VGGNet (figura 8), InceptionNet (figura 9), DenseNet, ResNet, Bert, GPT2. Dependiendo el tipo de problema y la base de datos con la que fue entrenada cada red, es posible tomar provecho de las mismas mediante un método llamado *Aprendizaje por transferencia* (o *Transfer Learning* en inglés).

La forma más simple de utilizar estas redes para hacer transferencia de aprendizaje es haciendo inferencia directamente a partir de ellas, es decir, luego de descargarlas, brindarles datos de entrada y realizar una pasada hacia adelante con el fin de obtener las salidas, aunque la única situación para hacerlo es si la red fue preentrenada con el mismo objetivo que el problema en cuestión (no se puede cambiar de un problema de clasificación binaria a uno multiclase directamente). Otra opción es reentrenar a la red con un conjunto de datos propio, de manera de extraer las características aprendidas por la misma y poder refinar sus pesos con un conjunto de entrenamiento que se entiende será más similar a lo que luego se utilizará para realizar predicciones.

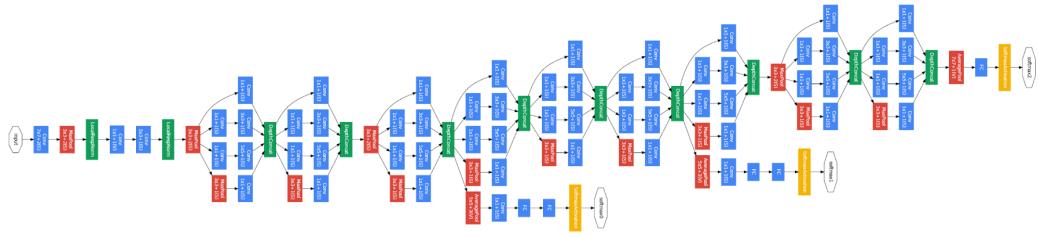


Figure 9: Arquitectura InceptionNet

Por último, otra alternativa es reentrenar estas redes es quitando la capa de salida y agregando capas conectadas a la anteúltima capa de la red. Esto permite tanto agregar capas extra sin entrenar a la red como cambiar las predicciones esperadas utilizando una capa de salida diferente a la inicial.

## 3 Revisión de antecedentes

### 3.1 Clasificación de escenas mediante Redes Neuronales Recurrentes

En [y otros, 2017a] J. H. Bappy y otros introducen un framework para aprender la información de las escenas de manera secuencial. Además, generan y hacen público un dataset de aproximadamente 6000 imágenes etiquetadas pertenecientes a las seis partes que consideran más importantes a clasificar de una casa: frente, patio, dormitorio, baño, living y cocina.

Para empezar con la estructura que proponen, es necesario explicar el algoritmo de preprocesamiento que aplican a cada imagen con el fin de intensificar los contrastes locales de la misma. Se trata de una de las variantes de ecualización de histograma de las imágenes llamada Ecualización de Histograma Adaptativo con Limitación de Contraste (o por sus siglas en inglés C.L.A.H.E., Contrast Limited Adaptive Histogram Equalization). Este algoritmo es la evolución de HE (Histogram Equalization) y de AHE (Adaptive Histogram Equalization). La Ecualización del Histograma de la imagen incrementa el contraste global de las mismas, sobretodo cuando los datos representativos de la imagen están representados por valores de contraste cercanos. Por otro lado, la Ecualización del Histograma Adaptativo computa múltiples histogramas correspondientes a diferentes secciones de la imagen, utilizándolos para redistribuir la limonosidad de la misma. La principal mejora es que obtiene mejores contrastes en sectores locales de la imagen, y define mejor límites dentro de cada región. Este método tiende a amplificar por demás el ruido en regiones relativamente homogéneas de la imagen, la Ecualización del Histograma Adaptativo con Limitación de Contraste se encarga de prevenir esta situación limitando la amplificación, podemos observar un ejemplo de aplicación de esta técnica en la Fig. 10. La propuesta de estos investigadores se basa en aprender la información de la escena secuencialmente, tanto de forma vertical como horizontal. Para hacerlo, crearon dos redes recurrentes LSTM (Long Short Term Memory) con cuatro (4) capas ocultas de ciento veintiocho (128) unidades cada una. Como segundo punto, alimentan a la red con los datos de las imágenes de forma secuencial, es decir, transforman las imágenes a un tamaño de 128x128 y luego alimentan la red con la información secuencial de cada pixel de forma vertical para una de las redes y de forma horizontal para la otra. Para finalizar, introducen una capa densa totalmente conectada (fully connected layer, en inglés) con las

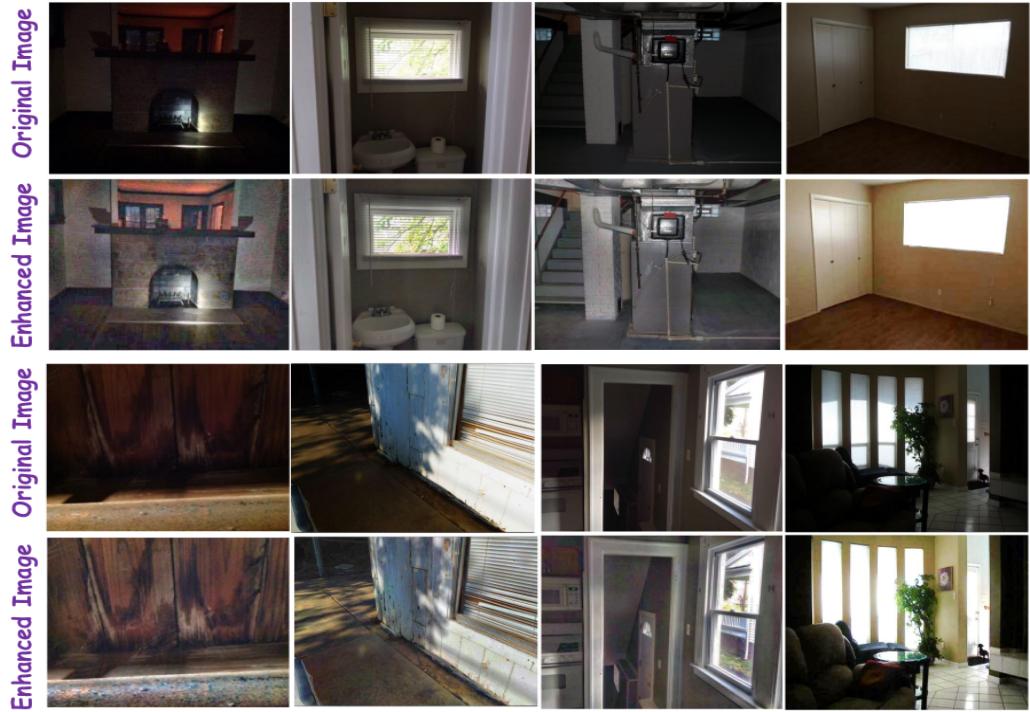


Figure 10: Ejemplo aplicación del algoritmo C.L.A.H.E.

salidas de ambas LSTM, seguida de otra capa densa totalmente conectada que concluye con una capa densa final con activación Softmax.

La arquitectura final propuesta como se puede observar en la Fig. 11, queda de la siguiente manera: dos redes LSTM que reciben los pixeles orientados de forma vertical y horizontal, respectivamente; una capa densa totalmente conectada que recibe las salidas de cada celda de la red LSTM, otra capa densa totalmente conectada que se concatena a la anterior y una capa densa final con activación Softmax que brinda las salidas. Vale mencionar que es necesaria la aplicación del algoritmo CLAHE a cada imagen para el posterior entrenamiento y clasificación mediante la red propuesta.

Las comparaciones realizadas por los investigadores se basan en la métrica Exactitud (Accuracy, en inglés) y se emplean utilizando diferentes configuraciones de esta red, y comparándose con el dataset que ha propuesto (Real Estate Images, en inglés) y el dataset abierto SUN. A modo de remarcarlo, obtienen como mejores valores un 96.92% de exactitud en el dataset REI,

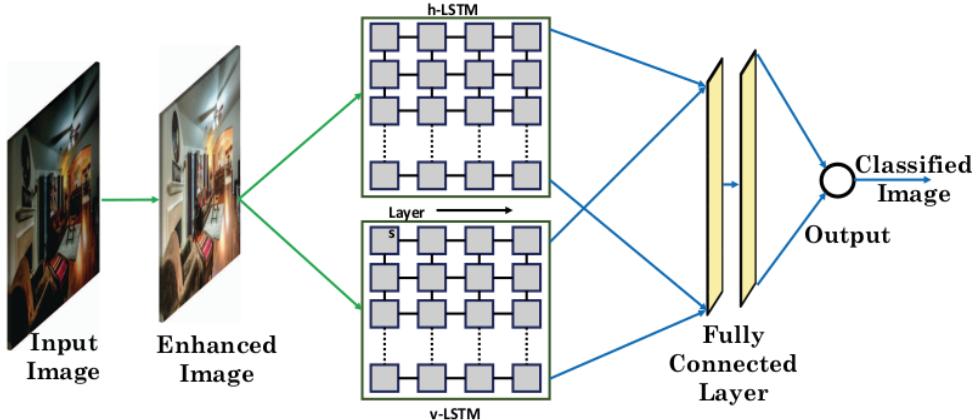


Figure 11: Arquitectura de la red presentada

y un 90.24% en el dataset SUN, utilizando la configuración descripta anteriormente. De esta manera quedan por encima de los resultados de utilizar extracción de características de redes preentrenadas como AlexNet con el dataset ImageNet y VGGNet con el dataset ImageNet.

### 3.2 Clasificación de escenas mediante Redes Neuronales Convolucionales

En [y otros, 2014] se creó un nuevo dataset con siete (7) millones de imágenes de escenas etiquetadas. Zhou y otros utilizaron redes neuronales convolucionales para aprender características profundas sobre las escenas y alcanzar un nuevo estado del arte. Ellos se encargaron de mostrar que las características de más alto nivel aprendidas por redes neuronales profundas en datasets centrados en objetos y a escenas son diferentes: imágenes de objetos no contienen la riqueza y diversidad de información visual que brindan las imágenes de escenas y ambientes para poder identificarlos.

El trabajo comienza por la construcción del dataset: creación de urls a partir de sustantivos y adjetivos relacionados a las escenas, eliminación de los links duplicados y luego de realizada la descarga, la eliminación de imágenes que se encuentren ya en el SUN database. Comparar la calidad del dataset generado en relación a otros de similar magnitud depende no sólo de las imágenes que contengan y sus categorías, sino también de múltiples factores como la variabilidad de las posiciones de la cámara, los estilos de decorado,

la ubicación y el tamaño que los objetos ocupan dentro de la imagen, etc. Es razonable asumir que una buena base de datos de imágenes debería ser densa y diversa. La densidad de una medida de concentración de los datos, y una base de datos de imágenes debe serlo ya que para aprender sobre un elemento (en este caso escenas) es necesario que haya alto grado de concentración de ese elemento. La realidad es que la densidad de un dataset no alcanza, ya que si se cuenta con todas imágenes de la misma habitación tendrá una densidad muy alta pero una diversidad muy escaza. La diversidad es una medida relacionada a la cantidad de clases en un dataset. Un dataset de imágenes debe ser diverso porque es necesario que haya tanto múltiples elementos dentro de la base de datos como también variabilidad de enfoques para sus imágenes. Ambas medidas son difíciles de medir en datasets de imágenes. En este caso, los autores proponen dos medidas: densidad relativa y diversidad relativa. Para la primera los autores asumen que, en el dominio de los datasets de imágenes, una alta densidad es equivalente a que en general las imágenes tienen vecinos similares. Por esta razón para realizar la medición toman una imagen random de un dataset  $A$  (llámese  $a_1$ ) y una imagen random del segundo dataset  $B$  (llámese  $b_1$ ). Si el set  $A$  es más denso que el set  $B$ , entonces es más probable que  $a_1$  tenga menor distancia a su vecino más cercano que  $b_1$ . Con esta definición, se tiene que  $A$  es más denso que  $B$  si y sólo si la densidad de  $A$  dado  $B$  es mayor que la densidad de  $B$  dado  $A$ .

$$Den_B(A) = p(d(a_1, a_2) < d(b_1, b_2)) \quad (15)$$

Esta noción de densidad entre datasets puede aplicarse a múltiples datasets  $A_1, \dots, A_N$ :

$$Den_{A_2, \dots, A_N}(A_1) = p\left(d(a_{11}, a_{12}) < \min_{i=2:N} d(a_{i1}, a_{i2})\right) \quad (16)$$

Si de diversidad se trata, existen varias formas de medirla que mayormente se utilizan en biología para conocer la riqueza de un ecosistema. Para este trabajo los investigadores se basaron en el índice Simpson de diversidad que es una medida de qué tan bien están distribuidos los individuos de las diferentes especies en un ecosistema, y está relacionado a la entropía de la distribución de los mismos. Ellos proponen medir la diversidad relativa de dos datasets basándose en la idea de que si dados dos datasets  $A$  y  $B$ , entonces  $A$  resultará más diverso si al seleccionar aleatoriamente dos imágenes del dataset  $B$  resultan más similares visualmente que seleccionar aleatoriamente dos imágenes del dataset  $A$ . De esta manera, la diversidad de  $A$  con

respecto a  $B$  puede ser definida como:

$$Div_B(A) = 1 - p(d(a_1, a_2) < d(b_1, b_2)) \quad (17)$$

donde  $a_1$  y  $a_2$  pertenecen a  $A$  y  $b_1$  y  $b_2$  pertenecen a  $B$  y fueron todas las imágenes seleccionadas aleatoriamente. De igual manera a la medida anterior, es posible de ser calculada entre más datasets  $A_1, \dots, A_N$ :

$$Div_{A_2, \dots, A_N}(A_1) = 1 - p\left(d(a_{11}, a_{12}) < \min_{i=2:N} d(a_{i1}, a_{i2})\right) \quad (18)$$

siendo  $a_{i1}, a_{i2} \in A_i$  seleccionados aleatoriamente.

En el marco de la experimentación realizada para demostrar que las redes aprenden características diferentes según el tipo de dataset que se utilice para entrenarlas los investigadores se quedaron con aproximadamente 2.48 millones de imágenes correspondientes a 205 categorías con un mínimo de cinco mil y un máximo de quince mil imágenes por cada una como set de entrenamiento (al cual se refieren como "dataset Places205"). El set de validación se seleccionó con cien imágenes por escena y el set de test doscientas, alcanzando un total de cuarenta y un mil imágenes entre estas dos últimas particiones. Finalizado el entrenamiento, los investigadores comparan las respuestas de unidades de varias capas de la red para entender mejor las diferencias entre ImageNet-CNN y Places-CNN, dos redes que tienen idéntica arquitectura pero que fueron entrenadas con sets de datos creados para diferentes propósitos (objetos y escenas, respectivamente). Las diferencias mayores entre las activaciones se dan a partir de la capa de pooling número dos gradualmente hasta la número cinco y también en la capa totalmente conectada número siete, en las que para ImageNet-CNN los campos receptivos se asemejan más a partes de objetos mientras que para Places-CNN en las mismas capas los campos receptivos parecen ser paisajes o estructuras relacionadas a un espacio. En Fig. 12 los investigadores hacen una

	SUN397	MIT Indoor67	Scene15	SUN Attribute
Places-CNN feature	<b>54.32±0.14</b>	<b>68.24</b>	<b>90.19±0.34</b>	<b>91.29</b>
ImageNet-CNN feature	42.61±0.16	56.79	84.23±0.37	89.85
	Caltech101	Caltech256	Action40	Event8
Places-CNN feature	65.18±0.88	45.59±0.31	42.86±0.25	94.12±0.99
ImageNet-CNN feature	<b>87.22±0.92</b>	<b>67.23±0.27</b>	<b>54.92±0.33</b>	<b>94.42±0.76</b>

Figure 12: Métricas por dataset y red

comparación sobre conjuntos de imágenes tanto centradas en objetos como en escenas. La métrica utilizada es exactitud y los sets con que compararon fueron: SUN397, MIT INDOOR67, SCENE15, SUN Attribute, CALTECH101, CALTECH256, ACTION40, EVENT8. En los primeros cuatro, centrados a imágenes, los resultados obtenidos por Places-CNN son mayores a los de ImageNet-CNN. En los segundos cuatro datasets, que son centrados en objetos, ImageNet-CNN es quien comanda en la métrica.

Para finalizar, entrenaron una red híbrida combinando el set de datos de entrenamiento de Places-CNN y de ImageNet-CNN. La llamaron Hybrid-CNN y luego de remover categorías solapadas alcanzó los 3.5 millones de imágenes pertenecientes a 1183 etiquetas diferentes. Esta red logró pequeñas mejoras en algunos de los datasets utilizados en la comparación entre Places-CNN e ImageNet-CNN. Los resultados se muestran en Fig. 13.

SUN397	MIT Indoor67	Scene15	SUN Attribute	Caltech101	Caltech256	Action40	Event8
53.86±0.21	<b>70.80</b>	<b>91.59±0.48</b>	<b>91.56</b>	84.79±0.66	65.06±0.25	<b>55.28±0.64</b>	94.22±0.78

Figure 13: Métricas por dataset - Hybrid-CNN

En [y otros, 2015] demostraron que no es necesario entrenar múltiples redes para realizar las tareas de clasificación de escenas y detección de objetos de una sola vez, ya que los detectores de objetos emergen por sí mismos en redes neuronales convolucionales entrenadas con datasets de escenas. Entender las representaciones aprendidas en las capas intermedias de arquitecturas profundas es un factor importante y del cual se podría sacar más provecho. Como las escenas están, en parte, compuestas por objetos, las redes neuronales convolucionales entrenadas para esta tarea aprenden a identificarlos internamente para definir de qué escena se trata, ergo la clasificación de escenas y la detección de objetos puede ser realizada en un mismo recorrido hacia adelante de la red, sin la necesidad de dar a la misma explícitamente la noción de objetos. La contribución más importante en este trabajo fue demostrar que las redes entrenadas para detección de escenas, internamente aprenden a detectar los objetos relacionados a estas escenas; característica que hace a estas redes explotables para realizar otros propósitos sin la necesidad de tomarse todo el trabajo de crear, entrenar y refinar una red más con la que detectar los objetos que se contienen en estas escenas. En mayor medida, si la red fue entrenada con un dataset centrado en objetos. Para esta tarea los investigadores buscaron simplificar las imágenes de entrada para poder conocer cuáles eran las características de éstas que concentraban la mayor parte

de la información que utilizada por la red, es decir, aquellas en las que luego de ir quitando resto de características de la escena, la exactitud con la que se predecía se mantenía similar. En el primer intento de realizar esta tarea, para cada imagen, ellos crearon una segmentación a partir de los bordes y regiones, removiendo segmentos en diferentes de la siguiente manera: en cada iteración se remueve aquel segmento que produce el menor decrecimiento en la puntuación de clasificación, hasta que la escena sea clasificada incorrectamente. Al finalizar este primer enfoque obtuvieron una representación de la imagen que contiene la información mínima necesaria para que la red clasifique correctamente la escena. En un segundo intento basado en la hipótesis de que para la red Places-CNN existían objetos cruciales en el reconocimiento de escenas, generaron representaciones mínimas de las imágenes anclándose del dataset totalmente anotado SUN Database en cambio de realizar una segmentación automática. Para hacerlo realizaron el mismo procedimiento que en el primer enfoque para obtener estas representaciones con la diferencia que tomaron como verdaderos los segmentos provistos por la base de datos SUN. Vale denotar que para cada escena, son objetos los que usualmente forman parte de la representación mínima necesaria por la red, como es posible observar en la Fig. 14.

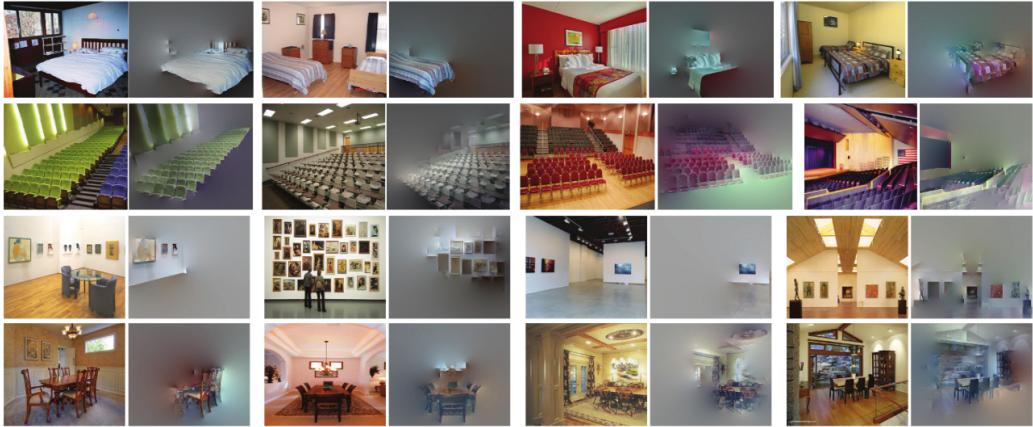


Figure 14: Ejemplos de representaciones mínimas encontradas

Para continuar con el trabajo, realizaron un análisis de los campos receptivos de las unidades y sus patrones de activación. Esto llevó a la observación de que las activaciones de las regiones tienden a tomar en mayor significado semántico a medida que se incrementa en la profundidad de las capas. Final-

mente, los investigadores brindan algunas razones de porqué emergen estos objetos en las tareas de clasificación de escenas: por un lado, que los objetos que emergen son los que más se encontraron en la base de datos SUN, por otro que éstos objetos que emergen son los que permiten discriminar mejor entre diferentes escenas, como es posible de observar en la Fig. 15. Gracias a este trabajo es posible alcanzar el estado del arte en clasificación de escenas utilizando la red Places-CNN, pero también aprovechar las capas intermedias para encargarse de detectar los objetos que aparecen en estas escenas a partir de los campos receptivos de las mismas.

En [y otros, 2018b] Herranz y otros, basándose en la idea de que el reconocimiento de escenas requiere comprender tanto sobre escenas como de objetos en la escena, se dedicaron a la tarea de resolver dos problemas relacionados; por un lado el sesgo inducido por datasets relacionados a objetos de una escala en redes neuronales convolucionales entrenadas con objetos en múltiples escalas y por el otro, cómo combinar eficientemente el conocimiento obtenido de datasets centrados en escenas y de datasets centrados en objetos.

En este trabajo ellos se encargaron de tener en cuenta la escala de los objetos para hacer que la red gane en razón de reconocimiento de las escenas. Para hacerlo eligieron centrarse en dos de los aspectos más importantes de los datasets: escala y densidad. Por el lado de la escala, en el dataset ImageNet cada objeto ocupa casi el total de cada imagen, mientras que por el lado del dataset Sun397 los objetos son mucho más pequeños. Por parte de la densidad, en el dataset centrado en objetos, cada imagen contiene un gran objeto, mientras que en el dataset centrado en escenas cada imagen contiene muchos objetos pequeños.

Gracias a la segmentación de objetos dentro de las escenas que lograron, fueron capaces de crear variaciones de un mismo elemento. De esta manera definieron y generaron los mismos objetos en dos escalas diferentes: escala original (la escala del objeto en la escena original) y escala canónica (se centra el objeto en la imagen y es reescalado para ocupar el tamaño de la imagen, manteniendo aspecto de radio).

La arquitectura propuesta por estos investigadores está enfocada a atender múltiples escalas mediante redes de escalas específicas. Esta gran red es una combinación de varias redes que operan en paralelo sobre sectores extraídos de la versión original de la imagen. Para cada una de las escalas utilizaron una red con una variante de la arquitectura AlexNetCNN llamada Caffe-Net. Para la extracción de los sectores, con el fin de acelerar el procesamiento, utilizaron una red totalmente convolucional (Fully convolutional

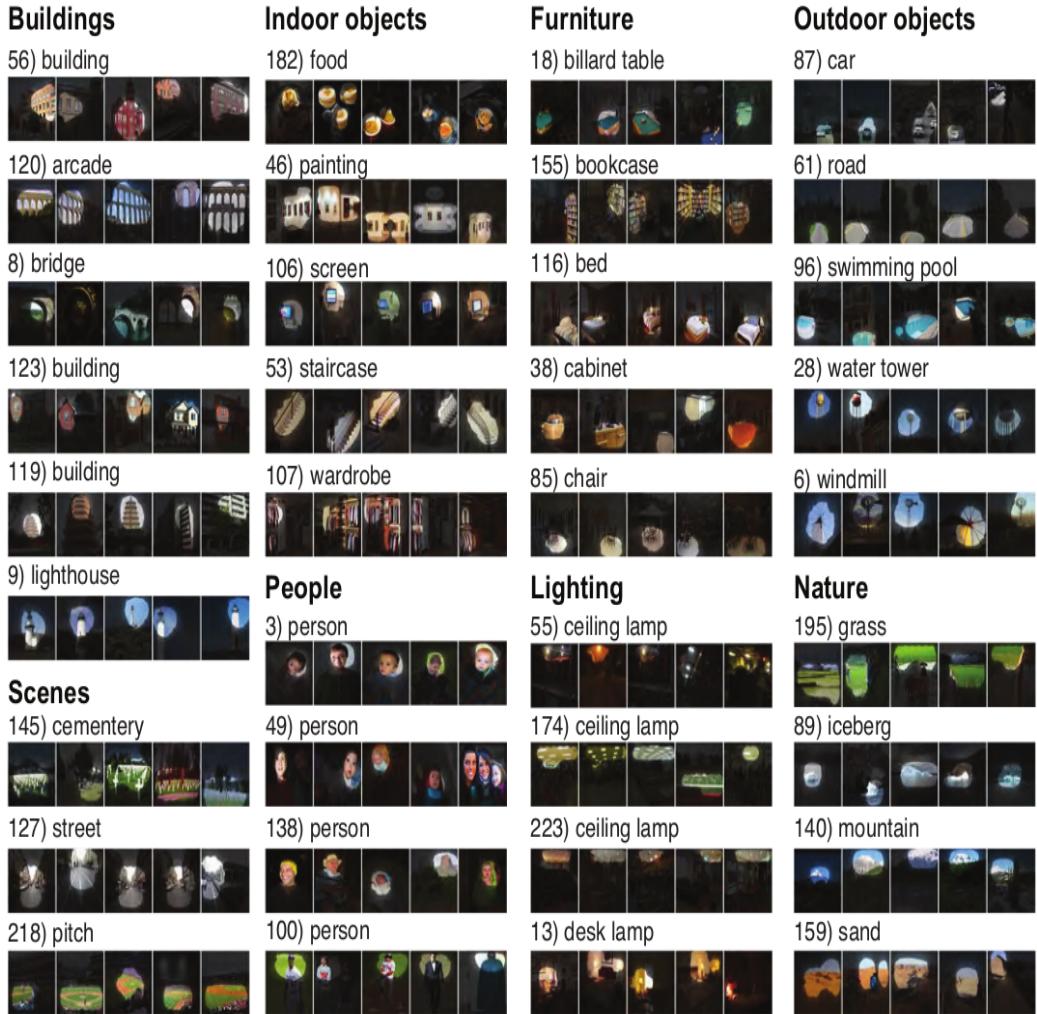


Figure 15: Ejemplos de objetos detectados a partir de la capa de pooling número 5

network), con una capa de agrupamiento mediante máximos para agregar las características de los sectores en características de la imagen en sí.

Ellos detallaron sus dos diferencias principales con la arquitectura híbrida base. Primeramente, ellos utilizan el modelo más adecuado para cada escala de las imágenes. Luego, se encargaron de refinar individualmente los parámetros de cada uno de los modelos generados para adaptarse de la mejor manera a la escala. Además, remarcan que el punto de principal inflexión con

el resto de trabajos similares es que ellos le dan importancia a la escala de la imagen, usando varias redes neuronales convolucionales en cambio de sólo una, alegando que las diferencias de escalas de los objetos entre los datasets ImageNet y Places dan lugar a la principal limitante de performance.

La idea final que otorgan es que la información local obtenida de la capa totalmente conectada número 7 de la red ImageNet-CNN más la información global obtenida de la capa totalmente conectada número 7 de la red Places-CNN funciona mejor que la implementación híbrida base. Esto es así debido a que la red ImageNet-CNN aprende características sobre los objetos en sí (aprendizaje local), mientras que la red Places-CNN aprende características sobre las escenas completas (aprendizaje global).

En [y otros, 2016] Zhang y otros dieron a conocer una nueva estructura para realizar esta tarea, con el cual sobrepasaron el estado del arte en los datasets utilizados. Se trata de Redes Neuronales Convolucionales Aleatorias Potenciadas por el Gradiente (Gradient Boosting Random Convolutional Neural Network, su nomenclatura en inglés), una forma de aprendizaje conjunto (ensemble) que combina varias redes neuronales profundas. Dentro de los aportes más significativos del trabajo se encuentran: la introducción de la red mencionada anteriormente en sí, una nueva función de pérdida multi-clase para poder combinar la potenciación por el gradiente con redes convolucionales y finalmente una variante a la red convolucional llamada red convolucional aleatoria (Random Convolutional Neural Network) que sirve como aprendiz base en tareas de aprendizaje conjunto profundo. La red está diseñada para generar un ensamblado de redes convolucionales aleatorias (RCNet) de manera de poder combinarlas usando una función de pérdida que se ajusta a la red base. La red propuesta como bien se mencionó anteriormente es un ensamblado, es decir, de múltiples redes intentando minimizar la función de costo y mapear datos de entrada con una salida a través de la estimación de una función que sea capaz de realizar este mapeo, en este caso, esta función estará formada por un conjunto de  $M$  funciones agregadas de forma aditiva.

$$\hat{f}(x) = \hat{f}^M(x) = \sum_{t=0}^M \hat{f}_t(x) \quad (19)$$

Los autores proponen tanto una nueva red base como una función de pérdida

para éstas. La función de costo está dada por:

$$\Psi(y, \hat{f}(x)) = - \sum_{k=1}^K y_k \log p_k(x) \quad (20)$$

donde la etiqueta a predecir está representada como 1 de los  $K$  vectores, siendo  $K$  igual al número de clases.  $\hat{f}(x)$  es la estimación general de la función de ensamblado y  $p_k(x)$  es:

$$p_k(x) = \frac{\exp(f_k(x))}{\sum_{l=1}^K \exp(f_l(x))} \quad (21)$$

Basándose en la aleatoriedad de los Bosques Aleatorios, con el fin de evitar el sobreentrenamiento por compartir todas las características entre todas las redes del ensamblado, los autores propusieron una variante a las funciones CNet: RCNet, que aleatoriamente comparte algunos de los parámetros con otras de las redes del ensamblado mediante un banco de filtros.

Durante cada función RCNet se samplea aleatoriamente un set de filtros del banco de filtros compartidos para construir la etapa de extracción de características de la RCNet y simultáneamente actualizar los parámetros de la RCNet y el banco de filtros compartido. El tamaño del filtro de del banco de filtros compartidos es mayor al del filtro en la función RCNet de manera que diferentes redes compartirán algunos parámetros.

Para concluir, los investigadores pusieron a prueba la arquitectura (ver Fig. 16) con dos datasets centrados en imágenes satelitales, comparando los resultados con arquitecturas clásicas de clasificación de imágenes y demostrando no sólo que la red propuesta es apropiada como aprendiz base en arquitecturas de ensamblado, sino que también es posible alcanzar el estado del arte y superarlos en relación a los métodos tradicionales.

### 3.3 Clasificación de escenas mediante Aprendizaje por Transferencia

En [?], un trabajo realizado por Råhlén Oskar y Sjöqvist Sacharias en 2019, se trabajó específicamente con los fines de utilizar el aprendizaje mediante transferencia para clasificar imágenes de propiedades. Resumidamente, ya que luego se abordará debidamente el tópico en el marco teórico, el aprendizaje mediante transferencia se trata de utilizar una red preentrenada  $R$  con

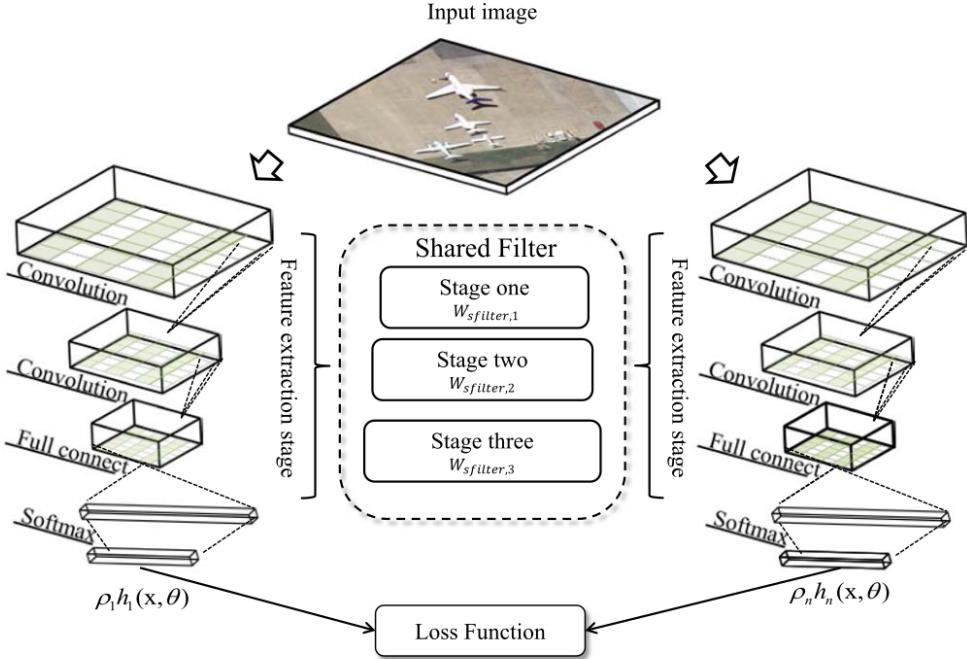


Figure 16: Arquitectura de la red

un conjunto de datos  $A$  para resolver un problema relacionado a un dataset  $B$ , y la manera para hacerlo es reentrenar la red R con los datos del dataset  $B$ . Para este proyecto utilizaron como redes preentrenadas ResNet18, AlexNet, VGG-11, DenseNet-121 e Inception-v3. Los datos con los que cuentan en esta investigación son imágenes extraídas de Google Image search y [hem, ], distribuidos de la siguiente manera para cada etiqueta como se observa en la Fig. 17.

En el trabajo se realizaron tres experimentos, y en cada uno de ellos se testearon todas las redes descriptas previamente, con y sin refinamiento de las mismas. El primero se trata de un clasificador binario para predecir si una imagen contiene o no un balcón; el segundo es igualmente un clasificador binario pero que predice si en la imagen se encuentra un hogar (por ejemplo, hogar a leña), y en el tercer experimento plantean una clasificación multiclase en la que intentan etiquetar las imágenes según si se trata de una cocina, un dormitorio o un baño. Es sobre éste último sobre el que se hará énfasis. Los experimentos con clasificadores binarios para estimar si se encuentra

Category	Training	Validation	Total
balcony	558	139	697
Indoor	486	122	608
Fireplace	382	96	478
no_fireplace	205	51	256
Kitchen	256	64	320
bathroom	208	52	260
bedroom	291	73	364

Figure 17: Distribución de las imágenes por categoría etiquetada

de un balcón o un hogar a leña en la imagen no resultan de gran interés para este problema, aunque es importante remarcar que para el primero se alcanza una exactitud de un 98% luego de refinamientos utilizando la red Inception-v3, mientras que para el segundo el mayor porcentaje se alcanza con la red DenseNet y alcanza un 85.5%. El experimento realizado que resulta de mayor importancia para este proyecto es el tercero, un clasificador de múltiples etiquetas que intenta determinar si la imagen se trata de una cocina, un dormitorio o un baño. Las métricas obtenidas antes de realizar refinamiento de las redes testeadas se dan a conocer en la Fig. 18 Luego

Model	Time	Max. accuracy
RESNET	02m 33s	93.75
Alexnet	02m 18s	93.22
VGG-11	04m 31s	93.75
Densenet	04m 20s	96.87
inception V3	06m 28s	93.75

Figure 18: Métricas usando extracción de características de los modelos preentrenados

de hacer refinamiento de las redes entrenando con las imágenes del dataset generado por los investigadores se alcanzan los resultados expuestos en la Fig 19.

Model	Time	Max. accuracy
RESNET	04m 16s	96.35
Alexnet	02m 34s	94.79
VGG-11	11m 02s	97.91
Densenet	09m 53s	97.91
Inception V3	16m 10s	97.39

Figure 19: Métricas luego de realizar refinamiento de las redes

Como se puede observar, luego de reentrenar los modelos se obtienen mejoras bastante representativas, dado el percentil de exactitud en que se encuentran los resultados. Otro punto a denotar es que en ambos casos es la red DenseNet la que obtiene la mejor performance, aunque toma aproximadamente el doble que el resto de las redes en entrenarse. Los autores concluyen su trabajo explicando que a partir de un bajo número de imágenes para su datasets, y a través de aprendizaje por transferencia, es posible agregar palabras claves a las imágenes, como ser: balcón, hogar, baño, cocina y habitación.

### 3.4 Otros trabajos relacionados a las propiedades inmuebles que toman provecho de las imágenes de los mismos

En [y otros, 2017b] Poursaeed y otros se dedican a la tarea de la estimación de precios de inmuebles basándose en las características visuales de las propiedades. El trabajo incluye una evaluación del impacto visual de las características de una casa en su valor de mercado, la estimación de lujosidad mediante redes neuronales convolucionales, un armazón para la automatización de la valuación utilizando tanto imágenes de las propiedades como metadatos de las mismas y experimentos en los que aplican su trabajo a un nuevo dataset. Para comenzar se encargaron de obtener alrededor de doscientas mil imágenes correspondientes a diferentes ambientes de casas a partir del dataset Places, Houzz (una empresa de alquiler y venta de viviendas) y búsquedas en Google Imágenes. Luego, entrenaron una red con arquitectura DenseNet para la tarea de predecir las etiquetas baño, dormitorio, cocina, living, comedor, in-

terior y exterior. Con este clasificador, alcanzaron un 91% de exactitud en el set de test. A partir de las imágenes etiquetadas, en esta investigación propusieron segmentar cada habitación en ocho niveles de lujosidad utilizando la herramienta de crowdsourcing Amazon Mechanical Turk con el fin de obtener estas etiquetas de cada sector. De esta manera, se hicieron de un embedding de baja dimensión en el cual las imágenes con el mismo nivel de lujosidad se encuentran cercanas entre sí. Mediante el algoritmo t-STE los investigadores obtuvieron un embedding bidimensional de las imágenes, que a partir de visualizaciones de los clusters se determinó que las imágenes con mayor nivel de lujosidad quedan en el centro, mientras que las menos lujosas se ubican alrededor. Para aquellas casas que no presentaban imagen de alguno de los ambientes, lo que hicieron fue imputar el promedio de las otras categorías para representar el nivel de lujosidad de ese ambiente. Una actividad aparte para ellos fue la estimación de precios, para la cual implementaron una regresión que absorve tanto la salida de los modelos que estiman la lujosidad de las habitaciones previamente clasificadas por la DenseNet, como metadatos al respecto de la propiedad (precio de oferta, tamaño, años desde su construcción, cantidad de habitaciones de cada tipo, etc). Vale aclarar que la etiqueta a predecir en cada propiedad es el valor final al que fue adquirida.

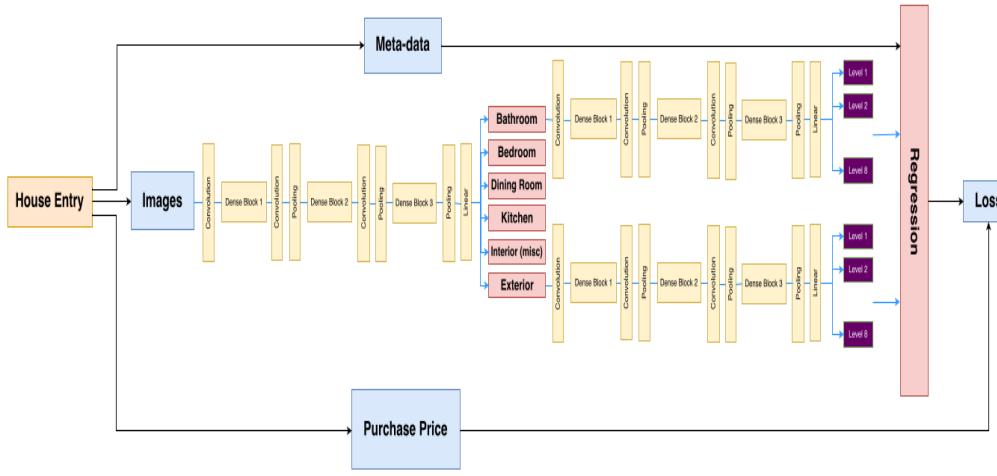


Figure 20: Arquitectura de la red descriptiva

Finalmente, con la arquitectura utilizada (ver Fig. 20) demostraron que es posible mejorar los resultados de estimaciones que actualmente se utilizan

en el mercado (Índice Zestimate de Zillow) disminuyendo la mediana del error de un 7.9% a un 5.6% logrado a partir de la arquitectura presentada.

### **3.5 Antecedentes no académicos**

Dentro de los antecedentes que no califican como investigaciones, nos encontramos con [res, ], un proyecto comenzado en 2016 por Angel Esteban. "Buscando entre docenas de propiedades, él estaba shockeado por la inconsistencia en la calidad de las mismas y la dificultad general para encontrar la casa de ensueños para su familia. Tenía que haber una mejor manera." es lo que muestran en la historia de la empresa, que actualmente cuenta con oficinas en Estados Unidos y Europa. Dentro de las capacidades de visión por computadora que listan relacionadas a las propiedades se encuentran la clasificación de escenas, la detección de características dentro de una imagen, análisis de estado de las habitaciones. Además, gracias a las capacidades mencionadas anteriormente en su sitio, demuestran cómo se aprovecharon del etiquetado de las imágenes para generar sus productos. Éstos son avocados a la experiencia de usuario, modelos de datos y moderación de contenido. En relación a la experiencia de usuario detallan tres productos fundamentales: la experiencia en búsquedas, el análisis comparativo del mercado y la conversión de imagen a discurso. Por el lado de los modelos de datos se encuentran la valoración automática de propiedades, completar información al respecto de una propiedad que pueda no haber sido detallada, es decir, la integridad de datos, las publicidades dirigidas y la analítica de datos relacionados a este contexto. Por parte de la moderación de contenidos se encuentra la detección de marcas de agua, detección de imágenes duplicadas y la detección de información sensible dentro de la imagen (patentes, números telefónicos, rostros). Como se puede ver, algunos de estos productos resultan ya definidos como posibles formas de aprovechar el etiquetado de imágenes en la motivación del proyecto, aunque otros no fueron mencionados y también continúan agregando valor y razón de ser a esta investigación.

## 4 Marco experimental

### 4.1 Asunciones

Para este trabajo se tomarán como ciertos los siguientes puntos:

- 1: Los datasets elegidos se encuentran etiquetados correctamente
- 2: Cada imagen de los datasets con los que se entrenarán los modelos resultan representativas a la escena a la que pertenecen.
- 3: El hardware con el que se llevará a cabo el proyecto funcionará tal y como se establece en sus respectivo manual.
- 4: La red preentrenada a descargar (PlacesCNN) fue entrenada sólo con las imágenes del dataset Places.
- 5: Para ambientes productivos la métrica [bal, ] es la más representativa.

### 4.2 Limitaciones

El proyecto en curso contará con las siguientes limitaciones:

- 1: El análisis se centrará en imágenes de las siguientes escenas de propiedades: cocina, comedor, baño, dormitorio, exterior, living, otros interior.
- 2: Tanto los tiempos de entrenamiento y predicción como el tamaño de las redes, estarán restringidos al hardware con el que se cuenta.
- 3: El trabajo intentará aceptar o refutar las hipótesis enumeradas a continuación, quedando excluídas del alcance del proyecto las posibles investigaciones que surjan a partir del mismo.

### 4.3 Hipótesis

Teniendo en cuenta tanto las limitaciones como las asunciones definidas para el trabajo, se comprobarán las hipótesis declaradas a continuación.

#### **4.3.1 Hipótesis 1**

Como se pudo observar en la revisión de antecedentes, existen múltiples formas de hacer frente al problema. El enfoque más simple podría ser mediante algún método tradicional como son las máquinas de soporte vectorial, pero por lo que se pudo observar, la mayoría de las soluciones utilizadas son redes neuronales convolucionales. Este punto de partida abre la primer hipótesis del trabajo: una red convolucional es capaz de obtener mejores resultados que un perceptrón multicapa en materia de clasificación de escenas.

#### **4.3.2 Hipótesis 2**

Las redes convolucionales tienen la capacidad de aprender características de las imágenes con las que se entrena, aunque a veces resulta muy costoso hacerlo por las diferencias entre imágenes de la misma categoría o bien no se cuenta con la cantidad de imágenes que aporten la densidad y diversidad necesaria para alcanzar el tope máximo en la métrica elegida. En este caso la hipótesis que se plantea está definida como: una red convolucional con una arquitectura  $A$  obtendrá mejores resultados sobre un conjunto de test  $y$  siendo entrenada con conjunto de entrenamiento  $X$  que si es entrenada con subconjuntos de 10% o 50% del conjunto de entrenamiento  $X$ , respectivamente.

#### **4.3.3 Hipótesis 3**

Dado que la cantidad de escenas a clasificar es relativamente baja, sería posible plantear un enfoque en el que se entrene una red convolucional para cada una de las mismas. En este caso se trataría de clasificación binaria en la que la red sea capaz de detectar si se trata de una escena o no, aunque se debería tener en consideración el hecho de no estar cometiendo sobreentrenamiento. La hipótesis a analizar en este punto es:  $N$  redes convolucionales entrenadas como clasificadores binarios (una para cada escena) mejoran los resultados que utilizar una sola red convolucional para clasificar las mismas  $N$  clases, es decir, los resultados obtenidos de contrastar la hipótesis 4.3.2.

#### **4.3.4 Hipótesis 4**

Como se pudo ver en [y otros, 2014], Zhou y otros crearon una red entrenada con millones de imágenes de escenas (Places Dataset) que debería ser

capaz de predecir las imágenes de las escenas con las que fue entrenada. En términos de cantidad de imágenes, esta red está mucho más entrenada que las utilizadas en este trabajo. Lo que se contrastará en este caso es si esta red preentrenada alcanza mejores resultados que una red convolucional entrenada con el dataset del trabajo [y otros, 2017b] (el de mayor cantidad de imágenes que se tiene).

#### 4.3.5 Hipótesis 5

Una de las técnicas revisadas antes de comenzar con el trabajo es Aprendizaje por Transferencia (3.3), en el cual se parte de redes preentrenadas y se las reentrena con el conjunto de imágenes propio para ajustar los pesos al mismo. ¿Haciendo aprendizaje por transferencia a partir de la red PlacesCNN es posible mejorar los resultados obtenidos al contrastar la hipótesis 4.3.4?

#### 4.3.6 Hipótesis 6

En [y otros, 2017a] se demostró que la aplicación de una técnica de ecualización del histograma a las imágenes del dataset (C.L.A.H.E.) mejora los resultados para las redes recurrentes que se utilizaron. Al aplicar este filtro a las imágenes de los tres datasets elegidos y luego haciendo aprendizaje por transferencia tomando como base PlacesCNN, ¿se obtienen mejores resultados que los expuestos al contrastar la hipótesis 4.3.5?

### 4.4 Experimentos

Con el fin de validar las hipótesis descriptas, se realizarán los experimentos que sean suficientes para validar cada una. Los datasets a utilizar serán los generados en los trabajos [y otros, 2017b] y [y otros, 2017a], además del dataset público SCENE15, del cual se seleccionarán las escenas determinadas en las limitaciones del trabajo. En las figuras [AGREGAR FIGURAS] se puede observar la distribución de escenas que contienen de los mismos. La métrica con la cual se compararán los diferentes resultados será Exactitud Balanceada [AGREGAR FÓRMULA] con el fin de obtener los mejores resultados a partir de clasificar correctamente todas las categorías y no sólo algunas; esta métrica se puede explicar como el promedio de la métrica recall para cada clase, y se plantea para poder trabajar con datasets no balanceados (como los que se utilizarán a lo largo de los experimentos).

#### **4.4.1 Experimento 1**

Se entranará tanto una red convolucional como una máquina de soporte vectorial con las imágenes del dataset generado en el trabajo [y otros, 2017b]. A continuación podremos comparar los resultados de ambos modelos de manera que será posible contrastarlos con la hipótesis 4.3.2.

#### **4.4.2 Experimento 2**

Se entrenará la misma red base que en 4.4.1 agregando también los datasets [y otros, 2017a] y SCENE15. A partir de los resultados obtenidos será posible contrastar la hipótesis 4.3.2.

#### **4.4.3 Experimento 3**

Se entrenará una red idéntica a las utilizadas en los experimentos 4.4.1 y 4.4.2 para cada escena descripta en 4.2 como clasificador binario. Luego se compararán los resultados agregados con los obtenidos en los experimentos 4.4.1 y 4.4.2 para determinar si la hipótesis 4.3.3 se cumple.

#### **4.4.4 Experimento 4**

Se descargará la red preentrenada con el dataset Places presentada en [y otros, 2014] llamada PlacesCNN y se clasificarán las imágenes del dataset de testeo con el fin de verificar si se cumple con la hipótesis 4.3.4.

#### **4.4.5 Experimento 5**

Se realizará aprendizaje por transferencia tomando como red base PlacesCNN y entrenando con todos los datasets elegidos. A partir de esta red reentrenada, se podrá constatar la hipótesis 4.3.5.

### **4.5 Experimento 6**

Se aplicará ecualización del histograma a las imágenes de los datasets elegidos, luego se procederá a los mismos pasos que en el experimento 4.4.5. De esta manera, será posible definir la veracidad de la hipótesis 4.3.6.

## References

[bal, ] Balanced accuracy score.

[hem, ] hemnet.se.

[res, ] restb.ai.

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[LeCun et al., ] LeCun, Y., Bengio, Y., et al. Convolutional networks for images, speech, and time series.

[Ronchi and Perona, 2015] Ronchi, M. R. and Perona, P. (2015). Describing common human visual actions in images. In Xianghua Xie, M. W. J. and Tam, G. K. L., editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 52.1–52.12. BMVA Press.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[y otros, 2014] y otros, B. Z. (2014). Learning deep features for scene recognition using places database.

[y otros, 2015] y otros, B. Z. (2015). Object detectors emerge in deep scene cnns.

[y otros, 2016] y otros, F. Z. (2016). Scene classification via a gradient boosting random convolutional network framework.

[y otros, 2018a] y otros, H. P. (2018a). Analysis of machine learning based scene classification algorithms and quantitative evaluation.

[y otros, 2017a] y otros, J. H. B. (2017a). Real estate image classification.

[y otros, 2018b] y otros, L. H. (2018b). Scene recognition with cnns: objects, scales and dataset bias.

[y otros, 2017b] y otros, O. P. (2017b). Vision-based real estate price estimation.