Universidad Autónoma de Sinaloa

Facultad de Informática Culiacán - Facultad de Ciencias de la Tierra y el Espacio

Maestría en Ciencias de la Información

Visualización de modelos generados por redes neuronales profundas para la clasificación de imágenes.

Eduardo Díaz Gaxiola

Directores:

Dr. Inés Fernando Vega López

Dr. Arturo Yee Rendón

Culiacán Rosales, Sinaloa. Marzo de 2021

Dedicatoria

Le dedico esta tesis a mi familia que estuvo conmigo a lo largo de la maestría. A mi madre Maria de los Milagros, a mi hermana Cecilia y en especial a mi abuela Ofelia. Desde el cielo te dedico mi trabajo abuelita.

Agradecimientos

Quiero agradecer principalmente a mis asesores, el Dr. Inés Fernando Vega López y el Dr. Arturo Yee Rendón. Gracias a su tiempo, dedicación y conocimientos durante toda la maestría, en especial durante el trabajo de tesis. Gracias a ellos por guiarme en mi formación académica y personal.

A los profesores del posgrado, en especial al M.C. Gerardo Beltrán Gutiérrez y al Dr. Jorge Adalberto Navarro Castillo que influyeron en gran medida en mi formación académica.

A mi familia, mis tíos Gilberto y Nari, mis primos Gilbertito y Gisell, mi abuelo Ismael, que me apoyaron en los momentos difíciles y siempre creyeron en mi.

A mis compañeros Roberto, Elías y Ana, que vivimos muchas experiencias y siempre estuvimos dispuestos a ayudarnos y salir adelante juntos.

A mis amigos Renato, Vlady, Ulises, Uriel, Jorge, Omar y a mis amigas Diana y Cynthia, quienes me animaron y apoyaron con su amistad y sus buenos deseos.

A la M.C Angélica Lizbeth Gómez Rendón que estuvo conmigo al final de mi trabajo, ayudándome a salir adelante y animándome a seguir.

Por último, quiero agradecer a la Universidad Autónoma de Sinaloa, al la Facultad de Informática Culiacán, al Posgrado en Ciencias de la Información, al Parque de Innovación Tecnológica y al Consejo Nacional de Ciencia y Tecnología por darme la oportunidad de recibir una beca con la que me dedique por completo a mis estudios de maestría y en especial al Fondo Sectorial CONACyT INEGI con clave 291772 CONACyT-INEGI 2017 01.

Índice general

Índice de figuras

Índice de tablas

Resumen

Las Redes Neuronales Convolucionales (RNC) se han popularizado recientemente en el área de visión por computadora gracias a su buen desempeño en problemas del clasificación de objetos en imágenes digitales. Estas redes se cuentan con dos procesos, el proceso de caracterización y el proceso de clasificación. Durante el proceso de caracterización, se encuentran las características distintivas de los objetos a clasificar contenidos en las imágenes de entrada. Estas características distintivas son usadas como información de entrada para el proceso de clasificación. El observar cuales fueron las características distintivas encontradas por las RNC durante la caracterización ayudaría a entender las decisiones de los modelos de clasificación. Una manera de conocer el proceso de caracterización es utilizando métodos de visualización, que son funciones matemáticas que generan una representación visual de la caracterización del modelo. En este trabajo, se propone un estudio comparativo entre los métodos de visualización Grad-CAM++, Score-CAM y mapas de saliencia. Además, proponemos como unidad de medida el porcentaje de cobertura de los objetos de interés. Presentamos como casos de estudio la clasificación de cáncer usando un conjunto de datos de imágenes de mamografías y la identificación de especies de plantas usando un conjunto de datos de imágenes de flores.

Capítulo 1

Introducción

La clasificación de objetos en imágenes digitales ha generado gran interés en el área de visión por computadora, esto se debe a la gran cantidad de problemas de clasificación que han surgido hoy en día. Por ejemplo, clasificar distintos modelos de autos que transitan por una calle [34], reconocer los tipos de suelos utilizando imágenes satelitales [32], distinguir entre especies de plantas [28], así como el reconocimiento de enfermedades pulmonares [29]. De esta manera, se han realizado múltiples investigaciones con el objetivo de clasificar de manera correcta los objetos en imágenes digitales.

La clasificación de objetos es un tipo principal de problema de aprendizaje supervisado que es ampliamente estudiado en el área de aprendizaje de máquina. El aprendizaje de máquina (*machine learning*) es una rama de inteligencia artificial que tiene como objetivo construir modelos que capturen el conocimiento aprendido sobre ciertos datos de entrada y que posteriormente infieran conocimiento sobre nuevos datos.

Las características distintivas de objetos son un factor importante en el desarrollo de modelos de aprendizaje de máquina. Las características distintivas son la representación numérica de una muestra de información sin procesar. Esta información sin procesar puede convertirse en datos útiles mediante ingeniería de características (*feature engineering*). La ingeniería de características toma las características y las transforma en formatos compatibles con modelos de aprendizaje de máquina [61]. La ingeniería de características se convierte en una labor de suma importancia ya que el seleccionar y probar las características de manera correcta impactará en el resultado final del modelo de clasificación. Realizar la ingeniería de características de manera

manual puede volverse un proceso complicado debido a la diversa información a procesar y al tiempo que consume [5]. Una alternativa es utilizar el aprendizaje de características (*feature learning*). El aprendizaje de características son un conjunto de técnicas que permiten extraer, a partir de información sin procesar, las características más importantes de los objetos para la elaboración de modelos de aprendizaje, siendo utilizado en problemas como reconocimiento de voz, reconocimiento de objetos y procesamiento de lenguaje natural [8]. Diferentes técnicas de clasificación como máquinas de vectores de soporte [55], árboles de decisión [50] ó redes neuronales artificiales [11] utilizan las características como información de entrada para realizar la clasificación. Por lo tanto, el representar de manera correcta las características de los objetos de interés es una parte fundamental en la solución a los problemas de clasificación.

Dentro de la clasificación de objetos, los problemas de clasificación fina presentan una mayor dificultad. Los problemas de clasificación fina consisten en reconocer múltiples subcategorías dentro de una misma categoría general, contando con gran variación en subcategorías y poca variación entre diferentes subcategorías [33]. Un ejemplo de ello es identificar especies de plantas mediante sus flores [39]. La *Orchidaceae* (comúnmente conocida como Orquídea) es una familia de plantas que, para clasificar entre sus diferentes especies similares, se requiere enfocarse en el color, forma y textura de sus flores [42].

Las técnicas de aprendizaje profundo (*deep learning*), quienes forman parte del aprendizaje de máquina, consiste en aprender las características en forma de múltiples niveles de representación para encontrar información de nivel superior a partir de información de nivel inferior [60]. Las Redes Neuronales Convolucionales (RNC) son técnicas de aprendizaje profundo que realizan el procesamiento de la información utilizando diferentes filtros digitales, encontrando las características distintivas de cada clase a clasificar [25]. Para poder utilizar las RNC es necesario contar con un conjunto de datos, el cual se dividirá en conjunto de entrenamiento y conjunto de validación. Las RNC cuentan con dos procesos: el proceso de caracterización y el proceso de clasificación. Durante el proceso de caracterización (o extracción de características), se recibe como información de entrada el conjunto de entrenamiento, encontrando las características distintivas de los objetos a clasificar mediante el uso de convoluciones. Las convoluciones son la combinación de una matriz de entrada (por lo general una imagen) y otra que se desplaza sobre ella (kernel), aplicando una serie de operaciones en cada una de las posiciones posibles,

para obtener una nueva matriz llamada mapa de características (*feature map*) [13]. Un kernel de convolución es un filtro digital que se aplica a una región de la matriz de entrada (imagen) para extraer las características distintivas que ayuden en el proceso de clasificación. Una vez encontradas las características distintivas, estas son utilizadas como información de entrada para el proceso de clasificación, generando un modelo de clasificación. Un modelo de clasificación es un modelo matemático que es capaz de generar una predicción sobre nuevos datos [24]. Una predicción es la información de salida de un modelo de clasificación que, dada una entrada, regresa una clasificación respecto a la información de entrada. Una vez entrenado el modelo de clasificación, se realiza un proceso de validación utilizando el conjunto de prueba. En este proceso, se valida la precisión del modelo de clasificación.

Dado que los modelos de clasificación entrenados por RNC reciben como entrada un tensor (el cual puede ser una imagen) y regresan un vector de probabilidades, se desconoce visualmente qué características de los objetos contenidos en las imágenes fueron encontradas durante el proceso de caracterización, convirtiendo a este proceso en una "caja negra". El conocer las características encontradas durante el proceso de caracterización ayudaría a entender y explicar las decisiones de los modelos de clasificación. Una manera de esclarecer el proceso de caracterización es utilizando métodos de visualización.

Los métodos de visualización son funciones matemáticas que, dada una imagen de entrada y una matriz de información, generan una representación visual de la caracterización del modelo, lo que permite validar sí las características encontradas por el modelo son las adecuadas para las clases. Las matrices de información se obtienen de valores calculados durante el proceso de caracterización del modelo de clasificación como son los valores de activación y los valores de gradiente. Los mapas de calor son una representación visual del proceso de caracterización, donde las regiones más importantes para el modelo se acercan al color rojo mientras que las regiones menos importantes se acercan al color azul. Los métodos de visualización nos permiten evaluar el proceso de caracterización de los modelos de clasificación.

Dentro del estado del arte, no se encuentra una medida estándar para evaluar el proceso de caracterización de los modelos de clasificación. Se propone tomar como unidad de medida el porcentaje de cobertura de los objetos de interés, calculando este porcentaje a partir de los mapas de calor generados por los métodos de visualización. Se decide utilizar esta unidad de

medida ya que si el modelo de clasificación no se enfocó en los objetos de interés durante el proceso de caracterización, el modelo no clasificará utilizando las características de los objetos de interés.

En este trabajo de tesis, se propone un estudio comparativo entre los métodos de visualización más representativos del estado del arte: Grad-CAM++ [9], Score-CAM [56] y mapas de saliencia [48]. Los mapas de saliencia son métodos de visualización que usan el valor de gradiente del modelo de clasificación para visualizar las características distintivas. Grad-CAM++ es un método de visualización que utiliza tanto el valor de gradiente como el valor de las activaciones del modelo de clasificación. Por último, Score-CAM cambia los valores de la imagen de entrada para calcular las zonas de la imagen que afecten más a la predicción del modelo de clasificación. Al realizar este estudio comparativo, se busca encontrar el método de visualización que evalúe de mejor manera el proceso de caracterización en modelos de clasificación basados en las arquitecturas de RNC, tales como VGG19 [49], InceptionV3 [53], ResNet50 [16], InceptionResNet [52], Xception [10], DenseNet 121, DenseNet 169, DenseNet 201 [19], MobileNet V1 [18] y MobileNet V2 [43]. Los casos de estudio que se presentan en este trabajo son: la clasificación de cáncer usando un conjunto de datos de imágenes de mamografías y la identificación de especies de plantas usando un conjunto de datos de imágenes de flores.

Definición formal del problema

- Sea $X = \{x_1, x_2, x_3, ..., x_n\}$ un conjunto de imágenes de entrenamiento. Donde cada tensor $x_i \in \mathbb{Z}^{h \times w \times d}$ pertenece a los enteros de tres dimensiones alto h, ancho w y profundo d.

- Cada imagen $X_i \in X$ tiene un conjunto de objetos de interés $O_j \in O$ donde O denota el conjunto de objetos de interés de todos los elementos de X.

- Sea o un objeto de interés donde $o \in O$ y $o \in \mathbb{Z}^{h \times w \times d}$ pertenece a los enteros de tres dimensiones alto h, ancho w y profundo d

- Sea $M = \{m_1, m_2, m_3, ..., m_n\}$ un conjunto de matrices de información obtenidas por un modelo de clasificación f denotado como $f : X \to M$. Donde $m_n \in \mathbb{Z}^{h \times w \times d}$ pertenece a los enteros de tres dimensiones alto h, ancho w y profundo d.

- Sea $V = \{v_1, v_2, v_3, ..., v_l\}$ un conjunto de métodos de visualización, tal que, $V : M \rightarrow H$ donde $H = \{h_1, h_2, h_3, ..., h_n\}$ denota el conjunto de mapas de calor de cada elemento del conjunto X y $h_n \in \mathbb{Z}^{h \times w \times d}$ pertenece a los enteros de tres dimensiones alto h, ancho w y profundo d .

Nuestro problema se define formalmente como sigue:

$$\underset{v \in V}{argmax}\ cover : H, O \rightarrow \mathbb{R}$$

Encontrar la función de visualización $v \in V$ que maximice la función de cobertura, que recibe un elemento del conjunto H y O retornando un porcentaje de cobertura.

El resto del documento se organiza de la siguiente forma. En el Capítulo 2 realizamos un estudio del estado del arte sobre las diferentes arquitecturas de RNC, así como de los métodos de visualización utilizados en esta tesis. En el Capítulo 3 describimos nuestras propuestas para evaluar el proceso de caracterización de los modelos entrenados por RNC. En el Capítulo 4 presentamos los resultados de este trabajo, describiendo los conjuntos de datos utilizados, los experimentos realizados y las evaluaciones del proceso de caracterización de cada uno de los modelos. Por último, en el Capítulo 5 presentamos nuestras conclusiones y nuestro trabajo a futuro.

Capítulo 2

Trabajo relacionado

En este capítulo se describe el funcionamiento de las redes neuronales artificiales, que son la base de las redes neuronales convolucionales. Las redes neuronales artificiales son técnicas de aprendizaje de máquina. Por su parte, las redes neuronales convolucionales son técnicas de aprendizaje profundo que utilizan múltiples capas de extracción de características y de aprendizaje. Se presenta, además, un estudio exhaustivo de las diferentes arquitecturas de redes neuronales convolucionales utilizadas en los diferentes retos de clasificación de imágenes, como lo son PlantCLEF e ImageNet. De igual manera, se presenta un estudio comparativo entre los diferentes métodos de visualización propuestos a través de los años en el estado del arte.

2.1. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) son técnicas de aprendizaje de máquina que simulan, mediante representaciones matemáticas, el procesamiento de información del sistema nervioso de los seres vivos. Sus inicios se remontan a estudios realizados en 1943 por Warren McCulloch y Walter Pitts [30]. Ellos proponen un modelo de dos partes conformado por funciones matemáticas. La primera parte del modelo recibía una serie de valores binarios de entrada y los sumaba. Al recibir esta suma, la segunda parte del modelo generaba una respuesta binaria en dependencia al valor de un umbral. Esto se puede apreciar en la siguiente ecuación:

$$y = f(x) = \begin{Bmatrix} 0 \; si \; x < u \\ 1 \; si \; x \geq u \end{Bmatrix} \tag{2.1}$$

donde x es valor de entrada al modelo, u es el valor de umbral y f es la función matemática.

2.1.1. Estructura

Las redes neuronales modelan matemáticamente el funcionamiento de las redes neuronales biológicas. Las células nerviosas se conforman principalmente de dendritas y axones [41]. Las dendritas son las conexiones donde se recibe la información. Los axones son las conexiones donde se envía la señal de salida. A la conexión entre la dendrita de una neurona y el axón de otra neurona se le llama sinapsis. En 1958, Frank Rosenblatt propone un algoritmo de clasificación llamado perceptrón [38]. El perceptrón define la estructura de una neurona que simula el comportamiento de las células nerviosas. Las dendritas son simuladas por las conexiones de entrada a las neuronas, la sinapsis son simuladas por los pesos de la neurona y el axón es simulado por la conexión de salida de la neurona. Adaptando estos conceptos, se puede ver en la Figura 2.1 la estructura un perceptrón simple.

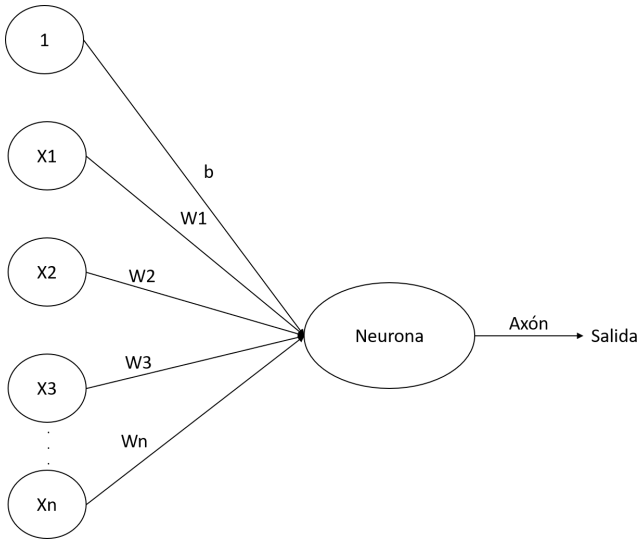


Figura 2.1: Estructura de un perceptrón simple. Esta estructura cuenta con varios valores de entrada. Cada entrada tiene asociada un valor de peso. Los valores de entrada son multiplicados por sus valores de peso para realizar una suma ponderada. Esta suma ponderada es recibida en la función de activación, calculando como resultado un valor de salida.

Neuronas

Las neuronas (también llamadas nodos o unidades) son el componente principal de las redes neuronales artificiales, siendo funciones matemáticas que reciben uno o más valores de entrada y producen una salida. Las neuronas se expresan matemáticamente con la siguiente ecuación:

$$y = f(\sum_{i=1}^{n} x_i w_i + b) \tag{2.2}$$

donde y es la salida de la neurona, i representa el número de entradas que recibe la neurona, f es la función de activación, $\sum_{i=1}^{n} x_i w_i$ es la suma ponderada de los valores de entrada x_i con los pesos w_i y b es el valor de sesgo, utilizado para asegurar una activación en la neurona en caso que todas las entradas sean 0. Las neuronas se encuentran organizadas en la red neuronal en forma de capas de diferentes tipos: [17]:

- Capas de entrada: Son aquellas neuronas que reciben la entrada a la red.
- Capas de salida: Son aquellas neuronas cuya salida se envía directamente a la salida y son las últimas neuronas en procesar información.
- Capas ocultas: Son todas aquellas neuronas que se encuentran entre las neuronas de entrada y las neuronas de salida. Estas neuronas siempre reciben información de otras neuronas y su información de salida es enviada a otras neuronas.

Se puede observar en la Figura 2.2 el ejemplo de una red neuronal con dos neuronas en la capa de entrada, tres neuronas en la capa oculta y una neurona en la capa de salida. Las neuronas en la capa de entrada son neuronas pasivas. Esto quiere decir que solamente reciben la información entrada, sin modificar su valor y enviando la información hacia las capas ocultas. Dentro de las capas ocultas, las neuronas procesan la información y se envía a la capa de salida, donde las neuronas combinan la información de entrada para producir una salida.

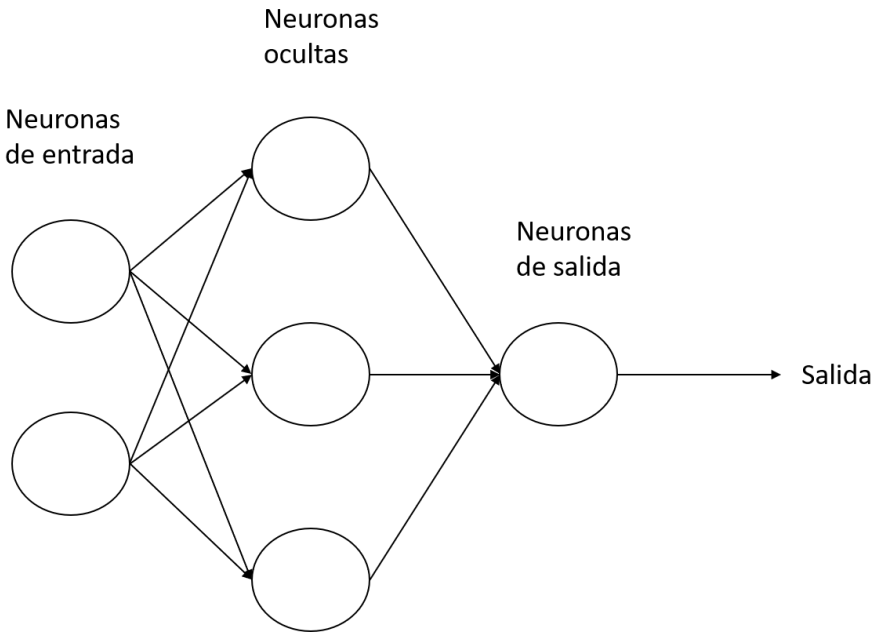


Figura 2.2: Esquema de los tipos de capas en una red neuronal artificial.

Función de activación

La función de activación es una función matemática que determina el valor de salida de la red neuronal. Usualmente todas las neuronas en una misma capa cuentan con la misma función de activación, pudiendo cambiar de función de activación en otras capas. Existen diferentes funciones de activación, siendo algunas de las más utilizadas las siguientes:

- Función de activación identidad: Es la función de activación más básica, donde la función regresa el valor que recibe como entrada. Su definición es $f(x) = x$ donde x es el valor de entrada. En la Figura 2.3 se observa el gráfico de la función identidad.

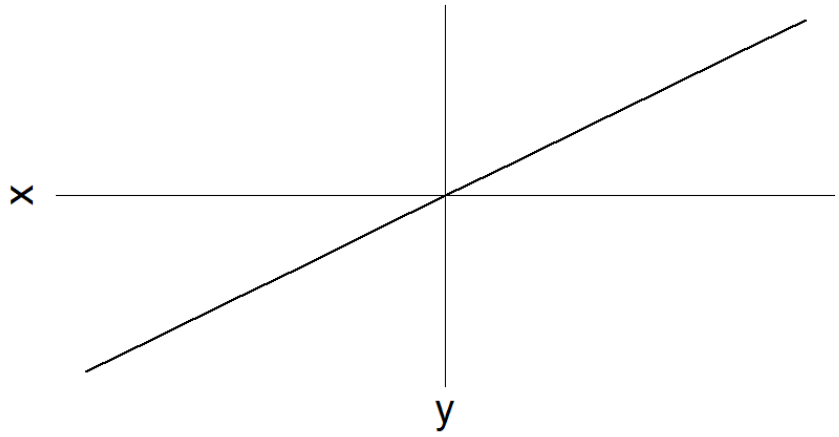


Figura 2.3: Función de activación identidad.

- Función de activación signo: Es una función en la cual, si la salida de la neurona es mayor o igual a 0, la función retorna como valor 1, de lo contrario retorna como valor 0. Su definición es la siguiente:

$$f(x) = \begin{Bmatrix} 0 \; si \; x < 0 \\ 1 \; si \; x \geq 0 \end{Bmatrix}$$

  En la Figura 2.4 se observa el gráfico de la función signo.

- Función de activación Sigmoide: El resultado de esta función se encuentra en el rango de 0 a 1. Esta función es diferenciable, es decir, se puede encontrar la pendiente de la curva en dos puntos cualquiera. Su definición es la siguiente:

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

  En la Figura 2.5 se observa el gráfico de la función Sigmoide.

2.1.2. Entrenamiento

El proceso de entrenamiento de las redes neuronales comienza recibiendo una información de entrada. Esta información es procesada a través de la red generando una salida. Dicha salida se compara con la salida esperada, calculando la diferencia con una función de pérdida.

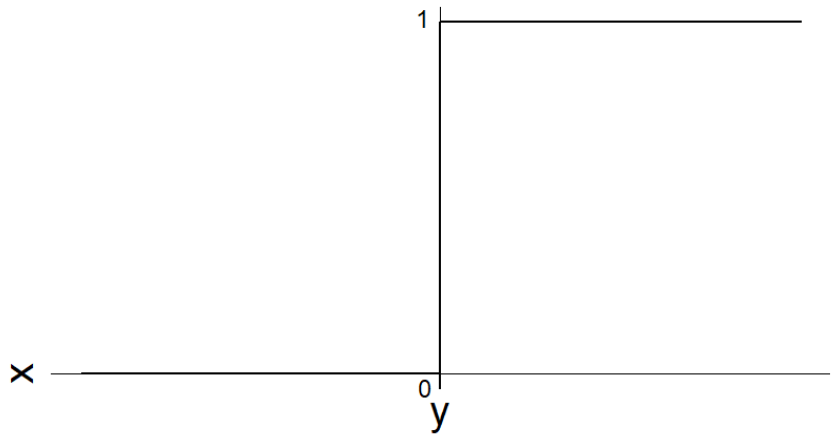


Figura 2.4: Función de activación signo.

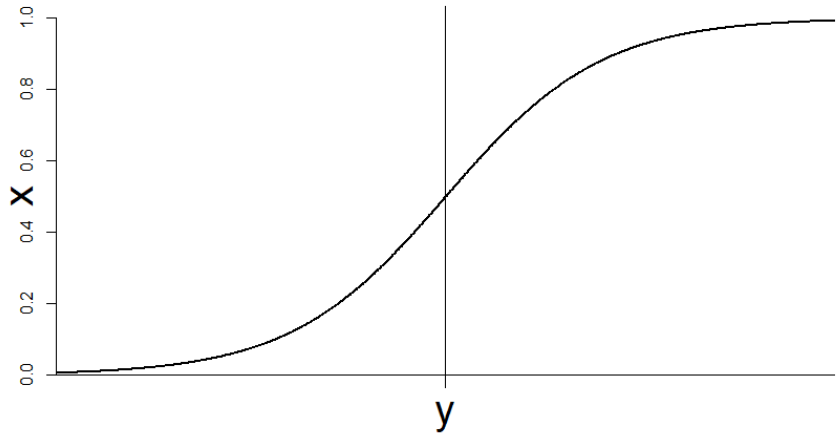


Figura 2.5: Función de activación Sigmoide.

Existen diferentes funciones de pérdida para diferentes necesidades, por ejemplo, la función de entropía binaria cruzada es útil para problemas de clasificación que cuenten con solo dos tipos de respuestas (es decir, clasificación binaria). Así mismo, la función de entropía categórica cruzada se utiliza para problemas de clasificación que cuente con múltiples clases. Por su parte, los algoritmos de optimización son utilizados para actualizar los pesos y el sesgo de la red,

tomando en cuenta el valor de error para esto. Dos de los algoritmos de optimización mayormente utilizados son el descenso de gradiente estocástico (SGD por sus siglas en inglés) [37] y el algoritmo de estimación del momento adaptativo (ADAM por sus siglas en inglés). El algoritmo de descenso de gradiente estocástico calcula el valor de gradiente para la información de entrada y actualiza el valor en dirección opuesta al gradiente. Esto con la finalidad de encontrar un valor mínimo local [23]. En cambio, el algoritmo de estimación del momento adaptativo calcula los porcentajes de incremento en el aprendizaje adaptativo para cada parámetro (pesos y sesgo).
El entrenamiento por regresión lineal es un entrenamiento que es utilizado por una red neuronal que cuente con una sola neurona con función de activación lineal. Otro tipo de entrenamiento es el entrenamiento por propagación hacia atrás. Este tipo de entrenamiento propaga el error calculado de la salida de la red neuronal hacia la entrada de la red neuronal, cambiando los valores de cada neurona respecto al error propagado de la capa anterior.

2.2. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (RNC) son un tipo de redes neuronales, cuyo uso ha sido popularizado para solucionar problemas de visión por computadora. Su origen se encuentra en el trabajo de Kunihiko Fukushima en 1980 [15] en el cual utilizaba dos tipos de neuronas, las neuronas "S" y las neuronas "C". Las neuronas las neuronas "S" ó neuronas simples se encargaban de encontrar características locales (líneas y formas básicas) en el campo receptivo mientras que las neuronas "C" ó neuronas complejas combinaban las respuestas de grupos de células "S". Por su parte, LeCun et ál. [26] propusieron la arquitectura LeNet-5 para clasificar números de códigos postales escritos a mano. Un problema al utilizar redes neuronales artificiales para procesar imágenes es la cantidad de parámetros y operaciones que se necesitan en cada capa, siendo esto costoso computacionalmente [3]. Suponiendo que se utilizan imágenes de tamaño $16 \times 16 = 256$ valores y la primera capa oculta cuenta con 1024 neuronas, la red realizará $256 \times 1024 = 262144$ operaciones solamente en la primera capa. Las RNC cuentan con medidas para reducir el costo computacional de procesar este tipo de información. En lugar de trabajar con redes completamente conectadas, es decir con redes cuyas salidas de las neuronas están conectadas a la entrada de todas las neuronas de la capa siguiente, estas redes trabajan

compartiendo parámetros. Esto quiere decir que solo cierta información es pasada a un número de neuronas en la siguiente capa, reduciendo considerablemente el número de parámetros. En las RNC, las neuronas activan neuronas de la siguiente capa para pasar su información, lo que hace que las neuronas siempre se encuentren conectadas a otras neuronas que compartan una región de interés. Las RNC se encuentran conformadas principalmente de capas convolucionales, capas de agrupación y capas densas.

2.2.1. Capa convolucional

Las RNC abstraen las características distintivas de los objetos a clasificar mediante el uso de convoluciones. Las convoluciones son operaciones matemáticas realizadas a partir de una matriz de operación comúnmente llamada "kernel" o "filtro" aplicadas a una región de la entrada. Los filtros se desplazan por cada espacio disponible de la entrada realizando una multiplicación de matrices, llamando a este movimiento como convolución. El paso (*stride* en inglés) es el valor que determina el número de celdas que avanza la convolución. Al realizar la convolución, se tiene como resultado una matriz de activación que es procesada en la siguiente capa. El tamaño de la matriz de activación puede variar en dependencia del valor de paso y de relleno. El relleno se basa en que, en caso de que un filtro no pueda estar completamente dentro de la entrada, se llenan esos espacios faltantes. El relleno permite que no se pierda información que podría resultar valiosa para capas posteriores. En la Figura 2.6 se observa un ejemplo del

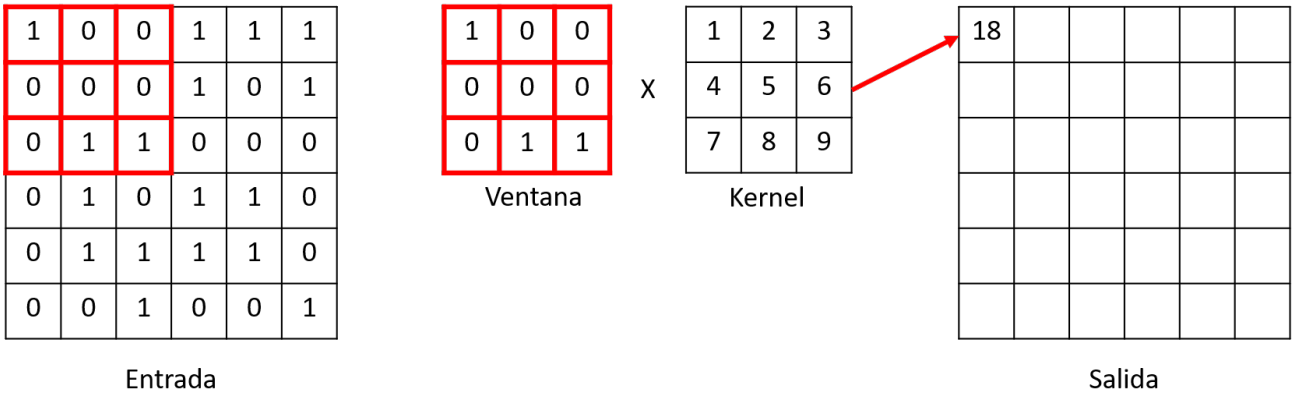


Figura 2.6: Funcionamiento de capa convolucional.

funcionamiento de las capas convolucionales. Se sitúa el kernel en una posición de la entrada

(siguiendo el ejemplo, la esquina superior izquierda) llamando a esta región campo receptivo. Se multiplica cada valor del kernel por cada valor situado en el campo receptivo. Al sumar los resultados de las multiplicaciones se obtiene el valor de una celda de la matriz de activación.

2.2.2. Capa de agrupación

Después de que un tensor es procesado por una capa convolucional, su tamaño es prácticamente el mismo. Para reducir el tamaño de un tensor (reducir el número de dimensiones o la longitud de alguna de sus dimensiones) se utiliza una capa de agrupación. Las capas de agrupación se encargan de agrupar valores de la matriz de entrada y reducir su tamaño, reduciendo el costo computacional. Al igual que las capas convolucionales, las capas de agrupación se desplazan sobre la entrada y aplican su agrupación en regiones. Los tipos de agrupaciones más comunes son la agrupación por valor máximo y la agrupación por promedio. En la agrupación por valor máximo se toma el valor máximo contenido en la matriz de operación, mientras que en la agrupación por promedio se realiza un promedio de los valores contenidos en la matriz de operación, siendo la agrupación por valor máximo la más utilizada en el estado del arte. Su valor de desplazamiento suele ser de 2 en adelante ya que se busca reducir el tamaño de la información. En la Figura 2.7 se observa un ejemplo del funcionamiento de las capas de agru-

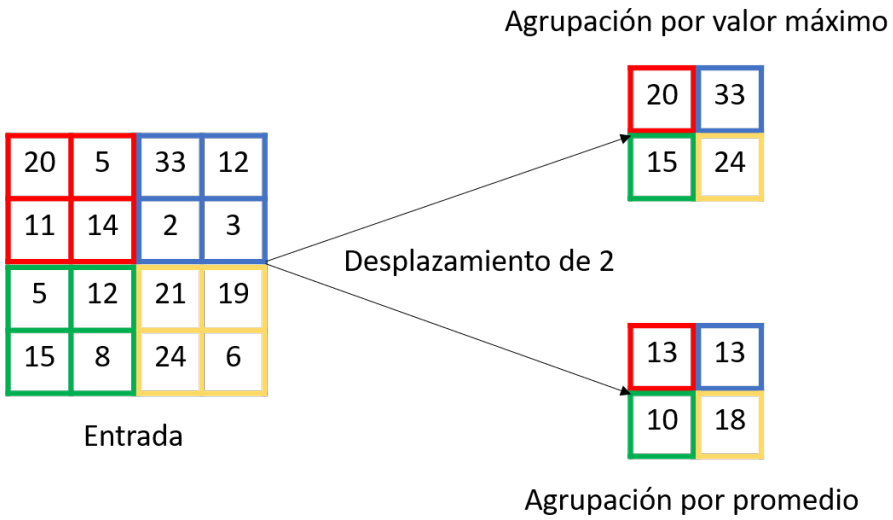


Figura 2.7: Funcionamiento de capa de agrupación.

pación. Se selecciona una región de la entrada utilizando una ventana de tamaño 2×2 y un desplazamiento de 2. Aplicando la agrupación por valor máximo se toma el valor máximo de cada ventana. Por su parte, la agrupación por promedio realiza un promedio de los valores de cada ventana.

2.2.3. Capa densa

Las capas densas son las capas totalmente conectadas tradicionales de las RNA, en donde cada neurona de una capa está conectada a cada neurona de la capa posterior. Estas capas se encuentran normalmente al final de las RNC y se utilizan para realizar el proceso de clasificación. Las capas densas normalmente reciben como información de entrada las características encontradas durante el proceso de caracterización. En algunas arquitecturas de RNC suelen tener más de una capa densa al final de la arquitectura. La última capa densa utiliza la función de activación Softmax que da como resultado un vector de probabilidades, donde cada elemento del vector representa la probabilidad de que la entrada pertenezca a una clase. La función de activación Softmax es una función de activación usualmente utilizada para la capa densa final [35]. La función Sofmax se define como:

$$f(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{n} \exp(z_j)} \tag{2.3}$$

donde z_i es el valor de activación de la i-ésima neurona de salida, n es el número de neuronas de salida y $f(z_i)$ es un vector de probabilidades donde cada valor se encuentra en el rango de $[0, 1]$.

2.3. Arquitecturas de RNC

Una arquitectura de RNC se conforma de una secuencia de capas, usualmente combinando capas convolucionales con capas de agrupación [21]. Para mejorar el rendimiento de las RNC, diversos investigadores proponen arquitecturas que se distinguen por el tamaño de la red, el tipo de conexiones entre capas, los parámetros en las capas convolucionales, entre otras distinciones [22]. Debido a las diversas maneras de conformar una arquitectura de RNC, el encontrar una arquitectura que presenta una precisión alta en diferentes problemas de clasificación sigue siendo

un problema a resolver [46]. A continuación se describen las arquitecturas de RNC utilizadas para este trabajo de tesis.

2.3.1. VGG

En el 2014, Simonyan et ál. [49] propusieron la arquitectura VGG-16 para participar en el reto de ImageNet 2014 logrando un 92.7 % de precisión top-5[40].

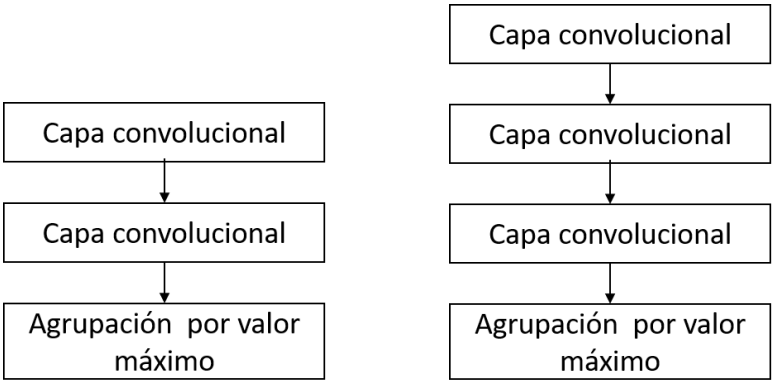


Figura 2.8: A la izquierda un bloque tipo A y a la derecha un bloque tipo B de VGG

La arquitectura de VGG se conforma principalmente de dos tipos de bloques, ilustrados en la Figura 2.8. El bloque A se conforma de dos capas convolucionales seguidas de una capa de agrupación por valor máximo. Por su parte, el bloque B se conforma de tres capas convolucionales seguidas de una capa de agrupación por valor máximo. La información de entrada es una imagen RGB de tamaño 224×224. Esta imagen es enviada al primer bloque convolucional tipo A, que cuenta con capas convolucionales de 64 filtros con tamaño 3×3 y un paso de 1. Al final de estas convoluciones se aplica una capa de agrupación máxima. Esta capa utiliza filtros de tamaño 3×3 para seleccionar el valor máximo del filtro, esto con un paso de 2. Al realizar esto, se reducen las dimensiones a la mitad, generando un tensor de 112×112. Los siguientes bloques cuentan con diferentes tamaños de filtros. Esta arquitectura se basa en cinco bloques: dos bloques A seguidos de tres bloques B, conectados al final a tres capas densas. Al final de la red se cuenta con una capa Softmax con una salida de 1000 valores. Estos valores son el total de clases a predecir con el conjunto de datos de ImageNet.

2.3.2. Inception

En el 2014 el equipo de desarrollo de Google, conformado por Szegedy, propusieron la creación de una nueva arquitectura llamada Inception para participar en el reto de ImageNet 2014 (ILSVRC14) [54]. Con este trabajo, el equipo de desarrollo de Google desarrolló los llamados módulos Inception. Estos módulos cuentan con secciones paralelas de diferentes tamaños de filtros, seguidos de una concatenación. El uso de diferentes tamaños de filtros se debe a la idea de que los objetos contenidos en las imágenes estarán en diferentes escalas de tamaño. Tomando en cuenta esta idea, utilizaron filtros de tamaño 1×1, 3×3 y 5×5 para capturar diferentes características. La estructura del módulo Inception se muestra en la Figura 2.9.

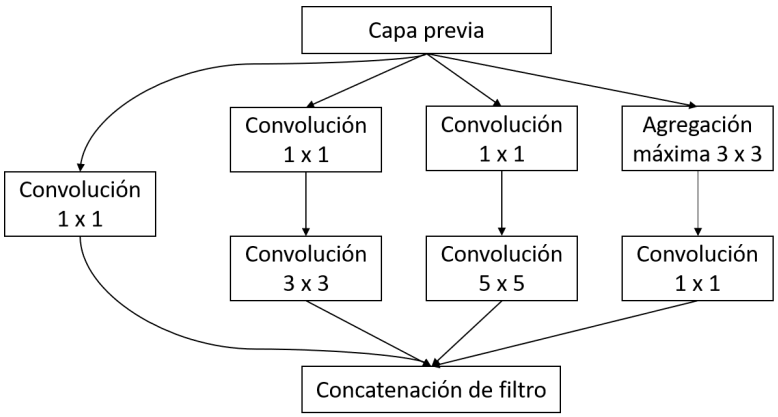


Figura 2.9: Módulo de Inception V1 adaptado del trabajo de Szegedy [54].

Se utilizan capas convoluciones de un filtro de 1×1 para reducir el número de operaciones y eliminar cuellos de botella computacionales. Un ejemplo de esta reducción es utilizar la capa convolucional de un filtro de 1×1 anterior a una capa convolución de un filtro de 5×5. Suponiendo un tensor de entrada de tamaño $56 \times 56 \times 128$ y una capa convolución de un filtro de $5 \times 5 \times 32$, se realizarían $(6 \times 56 \times 128) \times (5 \times 5 \times 32) = 321,126,400$ operaciones. Si entre el tensor de entrada y la capa convolucional de un filtro de $5 \times 5 \times 32$ insertamos, por ejemplo, una capa convolucional de un filtro de $1 \times 1 \times 16$, se realizarían $(56 \times 56 \times 128) \times (1 \times 1 \times 16) + (56 \times 56 \times 16) \times (5 \times 5 \times 32) = 46,563,328$ operaciones, reduciendo un 14.5 % el número de operaciones.

En el 2015, Szegedy y los desarrolladores de Google proponen una nueva versión Inception V3 [53] que vino con la versión 2 implementada en ella. En esta versión se reduce el número de co-

nexiones/parámetros sin afectar la eficiencia de la red. Para esto, hacen uso de la factorización de convoluciones, proponiendo un nuevo módulo A en donde se factorizan las convoluciones con filtros de tamaño 5×5 en dos convoluciones más pequeñas con filtro de tamaño 3×3 (Se puede observar el esquema del módulo A en la Figura 2.10). Esto hace que se reduzca el costo computacional, similar a lo realizado en VGG. Otro módulo propuesto en esta nueva versión

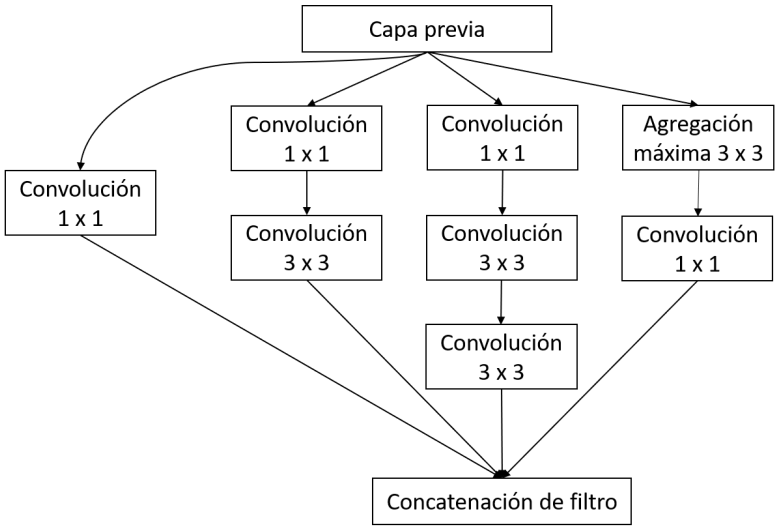


Figura 2.10: Módulo A de Inception V3 adaptado del trabajo de Szegedy [53].

se aprecia en la Figura 2.11 donde se remplazan las convoluciones de tamaño $n \times n$ por dos convoluciones de tamaño $n \times 1$ y $1 \times n$, donde n puede tomar el valor de 3, 5 o 7.

2.3.3. ResNet50

ResNet es una Arquitectura propuesta por He et ál. [16] en el 2015 para participar en el reto de ImageNet 2015. Con ella, se buscaba dar solución al problema de degradación de desempeño en las redes neuronales profundas debido al desvanecimiento de gradiente. Al entrenar una arquitectura de RNC, el flujo del gradiente se obstruye debido a la gran cantidad de operaciones que se realizan entre las capas. Esto quiere decir que al incrementar el número de capas el valor de gradiente tiende a disminuir. Al disminuir el valor de gradiente los pesos de las capas no se modificarán, impidiendo que se realice el proceso de entrenamiento de manera correcta [20]. La solución a este problema fue utilizar conexiones residuales. Las conexiones residuales

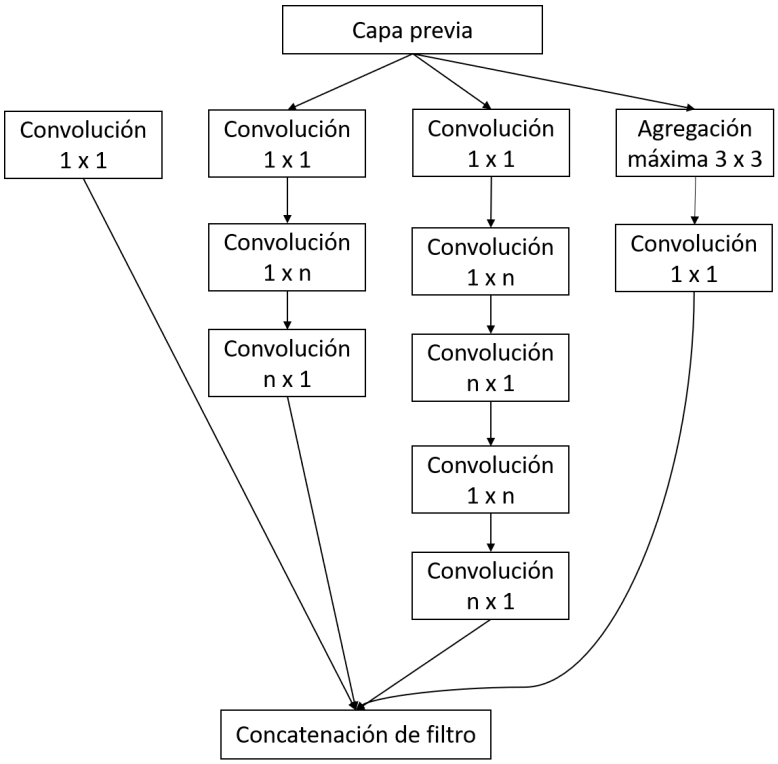


Figura 2.11: Módulo B de Inception V3 adaptado del trabajo de Szegedy [53].

consisten en conectar la entrada de la primera capa con la salida de la última capa mediante una conexión de salto. Esta conexión permite decidir el saltar capas convolucionales, evitando pasar por funciones de activación que reduzcan el valor de gradiente. Se puede observar un diagrama de conexión residual en la Figura 2.12.

Dada una entrada denotada por x, suponen que resultado a obtener mediante el aprendizaje es $F(x)$, que será utilizada como entrada en la función de activación. Utilizando la conexión de salto, la entrada a la función de activación es $F(x) + x$, conservando la información de la entrada x. ResNet se conforma de 152 capas, utilizando conexiones residuales a lo largo de la red. Tomando de ejemplo a VGG, que cada capa se encuentra conectada a su capa anterior, ResNet conecta no solo la capa anterior a la capa actual, sino también una capa detrás de la capa anterior.

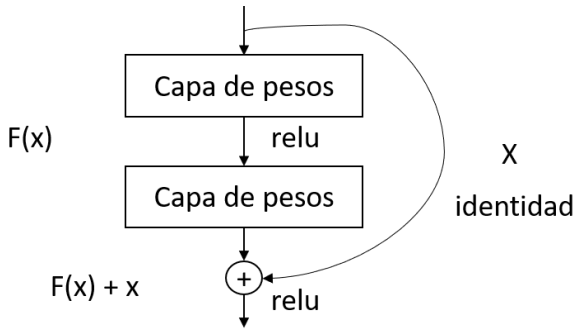


Figura 2.12: Diagrama de conexión residual adaptado del trabajo de He et ál. [16].

2.3.4. InceptionResNet

Para la implementación de esta arquitectura, en el 2016 Szegeady et ál. propusieron la combinación de dos de las ideas más recientes en el estado del arte en su momento. Estas ideas eran las conexiones residuales propuestas por He et ál. [16] y la última versión propuesta de la arquitectura Inception, dando lugar a la arquitectura InceptionResNet [52]. La idea surgió debido a que las conexiones residuales permiten aumentar la profundidad de la red sin generar el problema de desvanecimiento de gradiente [16]. Al ser Inception una arquitectura profunda, se consideró el reemplazar la etapa de concatenación de filtros con conexiones residuales. Esto permite obtener todos los beneficios de la conexión residual, conservando la eficiencia computacional. La arquitectura de InceptionResNet posee tres módulos Inception rediseñados (A, B, C) ilustrados en las Figuras 2.13, 2.14 y 2.15. El módulo InceptionResNet A es una variación del modulo tradicional de Inception V1 que incluye la conexión residual. Por su parte, los módulos de Inception B y C son módulos refactorizados de convolución de un filtro de 7×7 y 3×3 que incluyen una conexión residual. Al incluir la conexión residual, la arquitectura InceptionResNet permite incluir mayor cantidad de módulos A, B y C sin afectar al costo computacional.

2.3.5. Xception

En el 2017, la arquitectura Xception surge como una adaptación de Inception propuesta por Chollet et ál. [10]. El nombre de Xception viene de extremo, término por el que se refieren a los

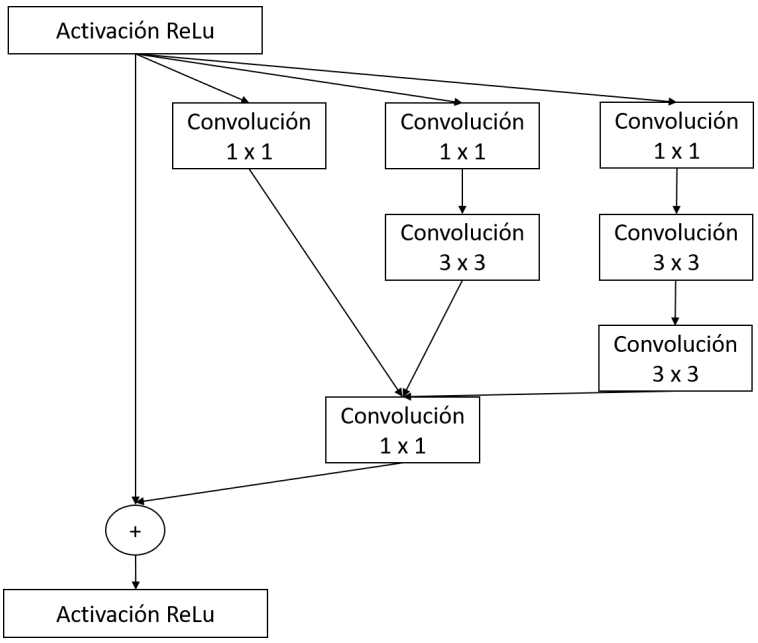


Figura 2.13: Módulo InceptionResNet A basado en un módulo tradicional de Inception V1 adaptado del trabajo de Szegeady et ál. [52].

nuevos módulos que proponen. Esta arquitectura reemplaza los módulos Inception por convoluciones separables en profundidad. Las convoluciones separables en profundidad son diferentes a las convoluciones normales de manera en que, para una entrada de imagen (32, 32, 3) podemos usar cualquier cantidad de filtros convolucionales. Cada uno de estos filtros se ejecutará en los tres canales y la salida será la suma de todos los valores. Sin embargo, en las convoluciones separables en profundidad cada canal tiene un solo kernel de convolución. Por lo tanto, al realizar convoluciones separables en profundidad se puede reducir la complejidad computacional, ya que cada kernel es solo de dos dimensiones y es convolucional solo en un canal. Con esta propuesta la convolución separable en profundidad modificada es la convolución de un filtro de 1×1 seguida por una convolución en profundidad. Existen dos diferencias entre estas convoluciones modificadas respecto a la original. Lo primero es que el orden de la operación cambia. Originalmente se realiza una convolución espacial en un canal seguido de una convolución de un filtro de 1×1, mientras que en la convolución modificada se realiza primero una convolución de un filtro de 1×1 y luego una convolución espacial. Una convolución espacial es una convolución aplicada a un canal solamente. La segunda diferencia existe en la presencia o ausencia de una

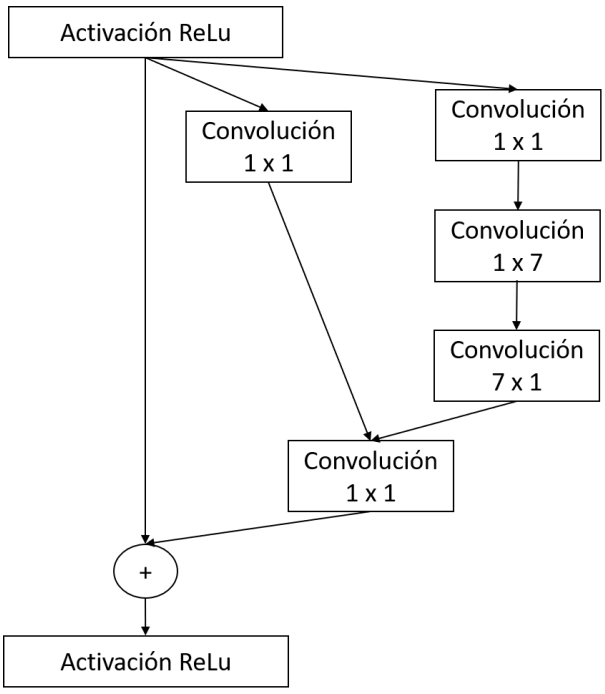


Figura 2.14: Módulo InceptionResNet B basado en un módulo refactorizado de convolución 7×7 adaptado del trabajo de Szegeady et ál. [52].

función de activación. Mientras que en el módulo de Inception se cuenta con una función de activación ReLU, en Xception no se implementa este tipo de función de activación. Podemos observar en la Figura 2.16 la estructura del módulo Xception. El módulo inicia con una entrada, seguida de una convolución de un filtro de 1×1 para cambiar la dimensión de la entrada por el número de canales de salida. Por cada canal de salida se aplica una convolución separable en profundidad, concatenando al final la salida de todas estas convoluciones.

2.3.6. DenseNet

En el 2017, DenseNet fue propuesta por Huang et ál. [19]. DenseNet a diferencia de la arquitectura ResNet que tiene bloques de función identidad donde se fusiona (o suma) una capa anterior con una capa futura, propone una concatenación de resultados de las capas anteriores en lugar de sumarlos. La distinción de esta arquitectura es el desarrollo de los llamados bloques densos. En arquitecturas como VGG y ResNet con capas L tienen conexiones donde la salida de la capa L es la entrada de la capa L+1 y la salida de la capa L+1 será la entrada de la capa

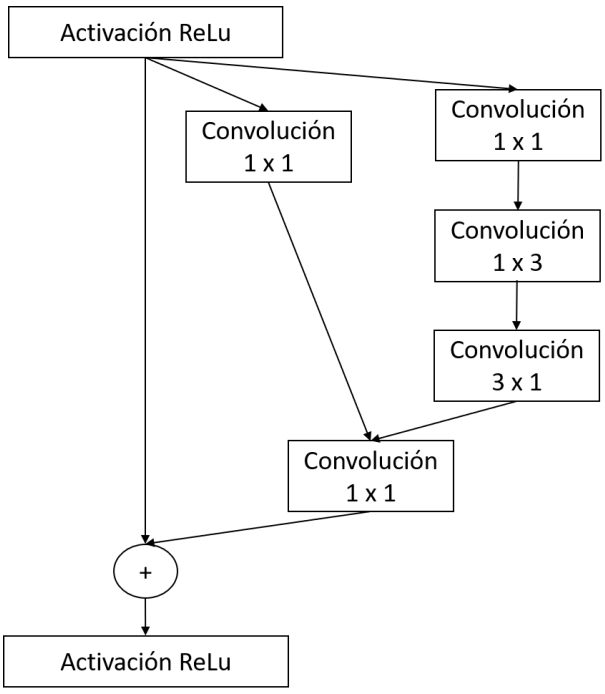


Figura 2.15: Módulo InceptionResNet C basado en un módulo refactorizado de convolución 3×3 adaptado del trabajo de Szegeady et ál. [52].

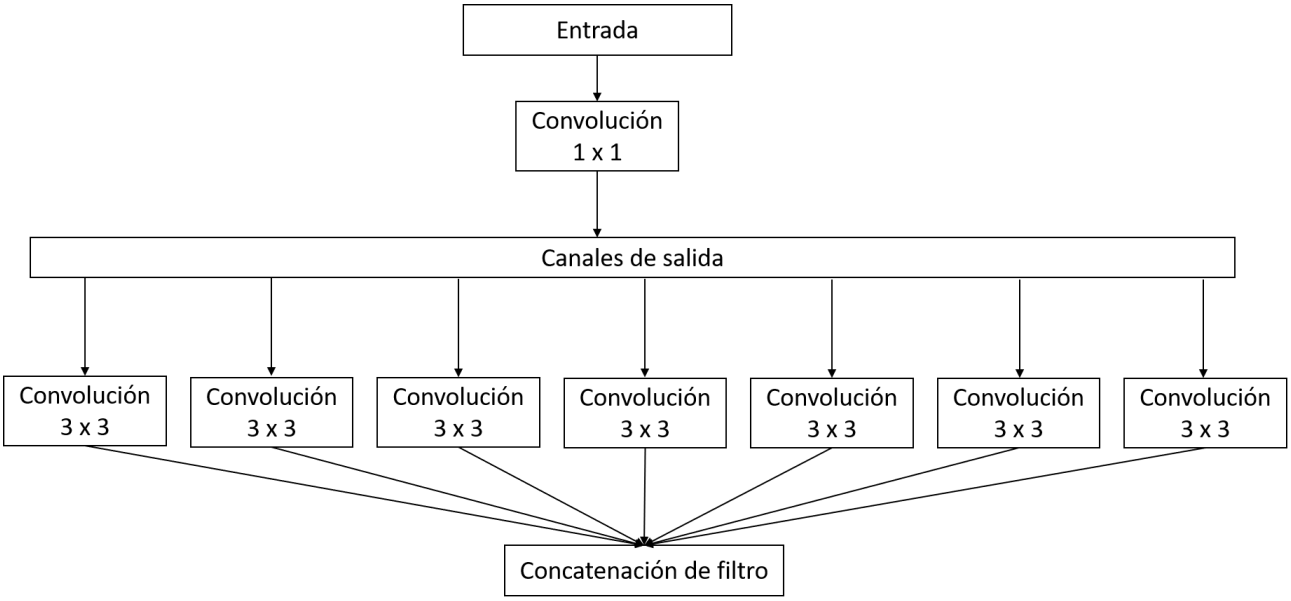


Figura 2.16: Módulo de Xception adaptado del trabajo de Chollet et ál. [10].

L+2, y así sucesivamente.

DenseNet implementa los bloques densos, cuyo esquema se puede apreciar en la Figura 2.17. En los bloques densos la información resultante de la capa L se envía tanto a la capa L+1 como a la capa L+2 hasta la capa L+n contenida dentro del bloque, lo que hace que todas las capas convolucionales dentro de este bloque tengan la información de las capas previas en todo momento, permitiendo reutilizar información útil de capas previas que, de manera tradicional, pudo perderse entre las conexiones.

Al utilizar los bloques densos se elimina el problema de desvanecimiento de gradiente ya que se tiene acceso al valor de gradiente a lo largo de cada función de perdida. Además, DenseNet reduce el número de parámetros. Esto se debe a que, al compartir cada capa la información de todas las capas anteriores, se puede reducir el número de canales de salida.

La arquitectura DenseNet se conforma mayormente de cuatro bloques densos conectados a una capa convolucional y a una capa de agregación, para finalizar con una capa de predicción. El número de capas convolucionales contenidas en los bloques densos es diverso.

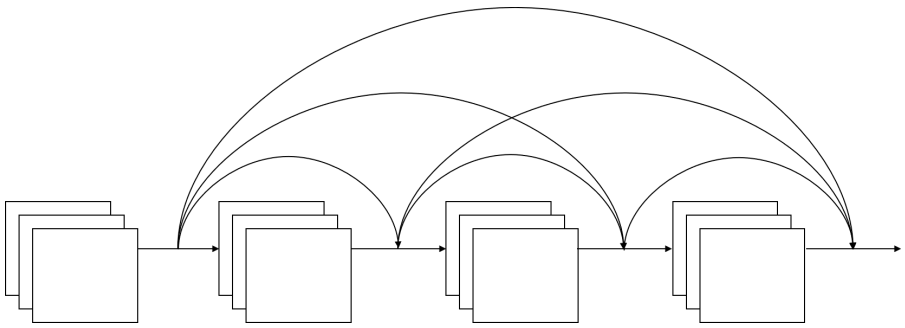


Figura 2.17: Bloque denso de cuatro capas convolucionales adaptado del trabajo de Huang et ál. [19].

2.3.7. MobileNet

Esta arquitectura fue propuesta inicialmente por Howard et ál. en el 2017 [18]. MobileNet al igual que Xception, utiliza convoluciones separables en profundidad para reducir el tamaño y complejidad del modelo. Esto hace a MobileNet una arquitectura recomendable para implementarse en dispositivos móviles [47]. En MobileNet la convolución profunda aplica un solo filtro

a cada canal de entrada. MobileNet propone dos parámetros que pueden ser modificados para reducir el tamaño de la arquitectura: Multiplicador de ancho α y el multiplicador de resolución ρ.

El multiplicador de ancho α es utilizado para controlar el ancho de entrada de una capa, haciendo que el número de canales de entrada M se convierta en αM y el número de canales de salida N se convierta en αN, además de que el costo de convolución separable en profundidad se convierte en:

$$D_k \cdot D_k \cdot \alpha M \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F \tag{2.4}$$

Donde α es un número en el rango de 0 y 1, comúnmente con los valores de 1, 0.75, 0.5 y 0.25. La línea base de MobileNet es $\alpha = 1$ y $\alpha < 1$ reduce el tamaño de la arquitectura. El segundo parámetro es el multiplicador de resolución ρ, que controla la resolución de la imagen de entrada de la red. Usualmente tiene los valores de 224, 192, 160, y 128, siendo 224 el valor por defecto. MobileNet utiliza bloques convolucionales separables en profundidad que realizan el proceso de convolución en dos partes. Se observa el esquema de un bloque convolucional separable en profundidad en la Figura 2.18. La primera parte del bloque utiliza una convolución separable en profundidad y la segunda parte del bloque utiliza una convolución de un filtro de 1×1, combinando los resultados de las convoluciones en un nuevo valor. La estructura de MobileNet se basa en una convolución tradicional de un filtro de 3×3, seguida de trece bloques convolucionales separables en profundidad.

2.4. Métodos de visualización

Una crítica recurrente a las redes neuronales convolucionales ha sido la poca claridad al momento de interpretar los resultados. Esto genera un problema al buscar respuesta sobre el mal comportamiento de nuestro modelo entrenado, o a la baja precisión que presenta. Una manera de interpretar los resultados de estos modelos es utilizando métodos de visualización. Los métodos de visualización son funciones matemáticas con las que es posible obtener información relevante de los modelos entrenados por redes neuronales convolucionales. La información se puede obtener de capas iniciales, intermedias o finales. Además, la información puede llegar a representarse como reconstrucciones basadas en una clase, zonas de interés para la predicción

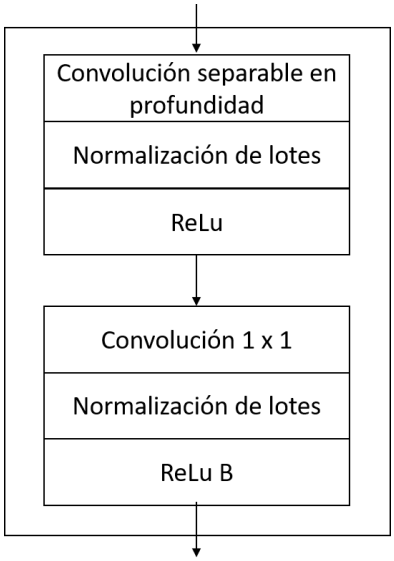


Figura 2.18: Bloque convolucional separable en profundidad de MobileNet adaptado del trabajo de Howard et ál. [18].

del objeto, pixeles puntuales, entre otras representaciones. Dada una imagen de entrada, un modelo entrenado por RNC y una matriz de información, los métodos de visualización generan una representación visual del procesamiento del modelo. Los métodos de visualización pueden ser definidos en cuatro tipos [14], según la matriz de información utilizada para realizar la visualización:

1. Métodos basados en gradientes.
2. Métodos basados en activaciones.
3. Métodos basados en combinación de gradientes y activaciones.
4. Métodos basados en perturbaciones.

Los primeros tres tipos de visualización utilizan información estática para realizar la visualización, es decir, información que no cambia su valor una vez entrenado el modelo, como lo son el valor de gradiente y el valor de las activaciones. En cambio, los métodos basados en perturbaciones utilizan información iterativa para realizar la visualización. Al perturbar nos referimos con cambiar los valores de la imagen de entrada al modelo y realizar su predicción, calculando las

zonas de la imagen que impactan en mayor medida a la predicción del modelo. La perturbación se realiza durante un número determinado de iteraciones.

2.4.1. Métodos basados en gradientes

En estos métodos, se utiliza el gradiente de salida con respecto a la entrada para construir los mapas de visualización. Los métodos basados en gradientes utilizan los valores de gradientes calculados durante la retropropagación para determinar que pixeles de la imagen de entrada cuentan con valor de gradiente mayor, atribuyendo a los valores con mayor valor como más importantes para el modelo [2].

Desconvolución

En el 2013, Zeiler et ál. [59] propusieron una visualización de características de nivel medio como lo son los bordes, líneas paralelas y figuras geométricas, utilizando desconvolución. La desconvolución se basa en crear una red de desconvolución que reconstruirá una imagen tomando como entrada los mapas de características calculados por una RNC. Al igual que las RNC, una red de desconvolución realiza un proceso de entrenamiento, en este caso para reconstruir la imagen. Este proceso de entrenamiento utiliza dos operaciones, la desagrupación y la desconvolución. En una RNC, la agrupación está diseñada para filtrar activaciones ruidosas de una capa inferior, usualmente por el valor máximo o promedio de una convolución. Este tipo de agrupación ayuda a la clasificación reteniendo solo activaciones robustas de las capas superiores, reduciendo la cantidad de datos a procesar en las capas inferiores. Sin embargo, esto hace que se pierda información, lo que afecta en la localización del objeto. Para resolver este problema se utilizan capas de desagrupamiento en la red de desconvolución. Las capas de desagrupamiento realizan la operación inversa de agrupamiento y reconstruyen el tamaño original de las activaciones. Para implementar la operación de desagrupación, se registran las ubicaciones de las activaciones máximas durante la agrupación en variables de conmutación, que se utilizan para volver a colocar cada activación en su ubicación agrupada original. La salida de una capa de desagrupación es un mapa de características ampliado pero escaso, esto se debe a que el mapa resultante de la desagrupación tendrá las dimensiones de la activación original pero con menos

información.

A diferencia de las capas convolucionales, las capas deconvolucionales asocian una activación de entrada única con múltiples salidas. La salida de la capa desconvolucional es un mapa de características ampliado y denso, ya que de un valor de activación se asocian múltiples valores de activación, incrementando el número de valores. Los filtros aprendidos en capas desconvolucionales corresponden a bases para reconstruir la forma de un objeto de entrada. Por lo tanto, similar a la red de convolución, se utiliza una estructura de capas desconvolucionales para capturar diferentes niveles de detalles de forma. Los filtros en las capas inferiores tienden a capturar la forma general de un objeto, mientas que los detalles específicos de la clase se codifican en los filtros superiores. De esta manera, la red tiene en cuenta la información de forma específica de la clase. Podemos observar en la Figura 2.19 el resultado de aplicar desconvolución a una imagen de la especie *Datura stramonium L. (1753)*, donde resaltan los bordes de la parte derecha de la flor, así como el centro de ella.

Figura 2.19: Resultado de la desconvolución a la especie *Datura stramonium L. (1753)*

Retropropagación guiada (Guided backpropagation)

En el 2015, Springenberg et ál. [51] propusieron la combinación de la retropropagación y la desconvolucion para realizar la reconstrucción de una imagen, llamando a este método Retropropagación guiada. En lugar de enmascarar las posiciones de valores negativos encontrados durante el entrenamiento del modelo (retropropagación) o enmascarar los valores negativos encontrados durante la reconstrucción de la imagen (desconvolución), la retropropagación guiada

enmascara los valores que cumplan con alguna de las dos condiciones. Con esto, este método permite cortar el flujo negativo de valores de gradiente, además de cambiar los valores negativos de la capa en reconstrucción actual. La idea de la retropropagación guiada es cortar el flujo de valores negativos debido a que estos valores corresponden a neuronas que inhiben la activación del objeto en capas superiores, por lo que restan importancia a las activaciones del objeto. Al enmascarar nos referimos a cambiar los valores de una imagen a cero, esto con la intención de resaltar el resto de valores en la imagen.
Podemos observar en la Figura 2.20 el resultado de aplicar la retropropagación guiada en la arquitectura Xception para la clase *Melochia pyramidata L. (1753)*.

Figura 2.20: Propagación hacia atrás guiada de la especie *Melochia pyramidata L. (1753)*

2.4.2. Métodos basados en activaciones

Estos métodos utilizan los valores de los mapas de activación para realizar la visualización, usualmente siendo los mapas de activación de la última capa convolucional los que cuentan con mejores características. Esto se debe a que, en la capa convolucional final, se encuentran las características encontradas durante todo el proceso de caracterización [57]. Para visualizar los mapas de activación realizan operaciones como el agrupamiento por promedio (APP ó *global average pooling* por su definición en inglés), maximizar las activaciones de ciertas neuronas o propagar hacia atrás la contribución de cada neurona en la red para cada característica de la entrada.

Maximización de activación

En el 2009, Erhan et ál. [12] propusieron el método de maximización de activación. Dentro de las RNC, cada capa convolucional contiene filtros que maximiza la salida cuando se encuentra un patrón similar al filtro en la imagen de entrada. La idea detrás de la maximización de activación consiste en utilizar como entrada una imagen con valores de pixeles aleatorios. Al momento de maximizar la salida de cada neurona, se modificarán los valores de la imagen de entrada. Como resultado de esta operación, se obtiene una imagen cuyos valores maximizan las activaciones de una neurona de alguna capa convolucional. La implementación de este método se enfocó en las primeras capas de la RNC. Esto con el propósito de encontrar una buena interpretación cualitativa de las características encontradas por el modelo en un alto nivel (como son líneas y texturas).

En la Figura 2.21 podemos observar la activación maximizada del primer filtro en la última capa convolucional en la arquitectura VGG-19. La última capa convolucional detecta las características abstractas del objeto, por lo que el visualizar la activación maximizada de la última capa nos mostrarán características difíciles de interpretar para el ojo humano.

Figura 2.21: Activación maximizada de la primera capa del último bloque convolucional de VGG19 para la especie *Argemone mexicana L. (1753)*

CAM

El mapeo de activación de clases (CAM por sus siglas en inglés) es un método de visualización propuesto por Zhou et ál. en el 2015 [62]. Este método utiliza la agrupación del promedio global (APG) de la RNC para generar un mapa de activación para una clase en particular. Dicho mapa indica las regiones discriminatorias de la imagen utilizadas por la RNC para identificar dicha clase. Para hacer uso de este método, es necesario utilizar una arquitectura de RNC que cuente con un módulo APG antes de la salida de la capa final. Una vez aplicado APG en los mapas de características, estos son usados como entrada para una capa completamente conectada que producirá la salida deseada. Dada esta estructura, se puede identificar la importancia de las regiones de la imagen propagando hacia atrás los pesos de la capa de salida en los mapas de características.

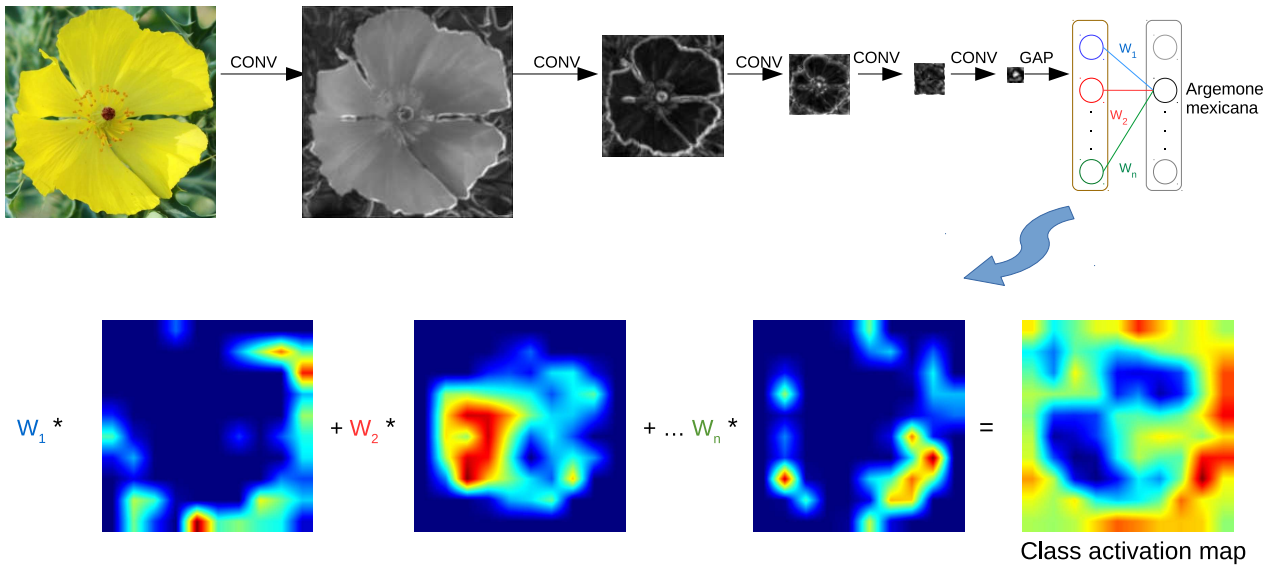


Figura 2.22: Metodología de CAM

Observando la metodología de CAM de la Figura 2.22, se sigue con el procedimiento normal de una RNC, realizando convoluciones hasta llegar al módulo de APG. Este genera el promedio espacial del mapa de características de cada neurona en la última capa convolucional. Se utiliza

una suma ponderada de estos valores para generar el resultado final. Del mismo modo, se calcula una suma ponderada de los mapas de características de la última capa convolucional para obtener los mapas de activación.

Dada una imagen, $f_k(x, y)$ representa la activación de la neurona k en la ubicación espacial (x, y) de la última capa convolucional. Para una neurona k, se define el resultado de realizar la APG como $F^k = \sum_{x,y} f_k(x, y)$. Para una clase c, se define w_k^c como el peso después de realizar la APG para la clase c de la neurona k. Esencialmente, w_k^c indica la importancia de F_k para la clase c.

Se define M_c como el mapa de activación de la clase c:

$$M_c(x, y) = \sum_k w_k^c F_k(x, y) \tag{2.5}$$

$M_c(x, y)$ indica la importancia de la activación en el plano (x, y) guiando a la clasificación de una imagen de la clase c.

2.4.3. Métodos basados en combinación de gradientes y activaciones

Estos métodos, como su nombre lo indica, realizan la visualización utilizando tanto el valor del gradiente como el valor de las activaciones de las neuronas. Al utilizar el valor de gradientes cuentan con la desventaja de que, siendo el valor de gradiente cambiante respecto a alteraciones en la imagen, genera ruido en los mapas de calor resultantes [56]. Para compensar este problema, se utiliza además el valor de activaciones.

Propagación de relevancia por capas (LRP)

Propuesto por Bach et ál. en el 2015 [6] el método de propagación de relevancia por capas (LRP por sus siglas en inglés) determina qué características en un vector de entrada contribuyen en mayor medida a la salida de una red neuronal. LRP define a la magnitud de la contribución de cada píxel para la predicción como R. Este método conserva el valor recibido por una neurona, el cual es distribuido a las capas inferiores de una manera equitativa [31], ejemplificado con la

siguiente ecuación:

$$R_j = \sum_k \frac{Z_j k}{\Sigma_j Z_j k} R_k \tag{2.6}$$

Donde j y k son dos neuronas conectadas en la red neuronal y $Z_j k$ representa el valor en que la neurona j ha contribuido para hacer a la neurona k relevante y propagar los puntajes de relevancia $(R_k)_k$ entre neuronas de una capa inferior. Una vez realizado el paso hacia adelante, no se pierde información relevante entre las neuronas al realizar la propagación de relevancia, manteniendo esta información al momento de generar el mapa de calor, teniendo similitudes con la propagación hacia atrás.

Grad-Cam

El mapeo de activación de clases por gradiente ponderado (Grad-CAM) es un método de visualización basado en el método CAM, propuesto por Selvaraju et ál. en el 2016 [44]. Grad-CAM usa el valor de gradiente de una clase para producir un mapa de calor.
Grad-CAM genera una representación visual para cualquier arquitectura RNC sin necesidad de re-entrenamiento o modificaciones a la arquitectura, contrario a CAM. Para obtener el mapa de calor de la clase a clasificar, se necesita el valor de gradiente para la clase de interés c. Además, es necesaria la salida del modelo para la clase c definida como y^c, con respecto a los mapas de características A^k de una capa convolucional. Se definen los pesos de importancia neuronal como:

$$\alpha_k^c = \frac{1}{Z} \sum_{x,y} \frac{\partial y^c}{\partial A_{x,y}^k} \tag{2.7}$$

donde la primera parte representa la operación de la AGP y la segunda parte representa el valor de gradiente de la salida del modelo y^c con respecto al valor del mapa de características A^k en la posición (x, y) donde x representa el ancho y y representa el alto. α_k^c indica la "importancia" del mapa de características A^k para c. Al realizar una suma ponderada de mapas características y aplicando una función ReLu obtenemos:

$$Grad - CAM = ReLu(\sum_k \alpha_k^c A^k) \tag{2.8}$$

Se aplica una función ReLu debido a que solo se necesitan las características que tienen una influencia positiva en la clase de interés, es decir, píxeles cuya intensidad debe de aumentarse

para aumentar y^c. Los píxeles cuyo valor haya sido negativo es probable que pertenezcan a otra clase en la imagen. De no aplicar la función ReLu, los mapas de calor podrían resaltar áreas que no pertenezcan a la clase deseada.

2.4.4. Métodos basados en perturbaciones

Estos métodos implican perturbar la intensidad de los píxeles de la imagen de entrada a la red neuronal con un ruido mínimo y observar el campo de probabilidad que tiene la predicción de acuerdo al ruido. La idea detrás de esto es que los píxeles que contribuyan en mayor medida a la clasificación acertada del objeto a clasificar, una vez alterado su valor, reducirán el resultado de la clasificación, permitiendo así identificar dichos píxeles mediante mapas de calor [4].

Mapas de oclusión

Zeiler et ál. [58] en el 2013 utilizaron mapas de oclusión para visualizar las características encontradas por modelos predictivos. Esta es una técnica que se basa en ocultar una ventana de un tamaño $W * H$ donde W es un tamaño en ancho y H es un tamaño en alto. Una vez ocultada la región, se procede a realizar la predicción de la imagen y almacenar el resultado de la predicción. Este proceso se repite moviendo la ventana en un tamaño determinado. Con esto, determinamos que áreas de la imagen, en caso de no encontrarse en la imagen, impactaría en la predicción de la clase. El resultado es un mapa donde los valores altos significan regiones de la imagen cuyo valor impacta de mayor manera a la predicción, mientras que valores bajos significa que no son tomados en cuenta para la predicción. Una limitante de este método de visualización es la dependencia del tamaño de la ventana para abarcar una mayor cantidad de características de la imagen, el mapa resultante se enfoca en objetos de interés solamente en tamaños de ventana altos [4].

LIME

Propuesto por Ribeiro et ál. en el 2016 [36], el modelo local interpretable de explicaciones agnósticas (LIME por sus siglas en inglés) es un método utilizado para explicar y evaluar las predicciones de un modelo clasificador. Al ser un método agnóstico, es decir un método el cual

Figura 2.23: Resultado de LIME a la especie *Purshia plicata (D. Don) Henrickson (1986)*

puede ser implementado para diferentes métodos de predicción, no hace uso de valores que otros modelos no puedan contar (como lo es el valor de gradiente). Por lo tanto, utiliza la perturbación de la entrada al modelo y monitorea los cambios a la predicción. Esto beneficia la interpretación al poder cambiar características que sean visualmente más representativas para el ojo humano (letras, figuras). El funcionamiento general de LIME se basa en agrupar pixeles individuales en grupos con valores similares llamados superpíxeles. Una vez agrupados, se comienza a activar y desactivar los grupos, generando imágenes de manera iterativa, calculando predicciones y ponderando la importancia de los grupos para la predicción. Esto permite visualizar que características toma en cuenta el modelo para realizar la predicción. En la Figura 2.23 podemos ver el resultado de implementar el método LIME con 1000 iteraciones, utilizando la arquitectura Xception con la especie *Purshia plicata (D. Don) Henrickson (1986)*, resaltando en la imagen las características mayormente utilizadas por el modelo para realizar la predicción.

A continuación en la Tabla 2.1, se resumen los métodos de visualización descritos a lo largo del capítulo. Los métodos de visualización se clasifican en métodos basados en gradientes, activaciones, combinación de activaciones y gradientes y perturbaciones.

Nombre	Año	Autor	Tipo
Maximización de activación	2009	Erhan et ál.	Activaciones
Desconvolucion	2010	Zeiler et ál.	Gradientes
Mapa de oclusión	2013	Zeiler et ál.	Perturbaciones
Propagación hacia atrás guiada	2015	Springenberg et ál.	Gradientes
LRP	2015	Bach et ál.	Gradientes y activaciones
CAM	2015	Zhou et ál.	Activaciones
Grad-CAM	2016	Selvaraju et ál.	Gradientes y activaciones
LIME	2016	Ribeiro et ál.	Perturbaciones

Tabla 2.1: Resumen de técnicas de visualización consultadas.

Capítulo 3

Evaluación del proceso de caracterización de los modelos de clasificación entrenados por RNC

A lo largo de este capítulo se describen nuestras propuestas de evaluación para el proceso de caracterización de los modelos de clasificación entrenados por RNC. La caracterización es el proceso de las RNC donde se extraen las características distintivas de los objetos contenidos en el conjunto de entrenamiento, que es utilizado como información de entrada para el proceso de clasificación de las RNC.

Conocer la calidad de la caracterización nos permitirá evaluar que tan confiable es la precisión obtenida por nuestros modelos entrenados, además de validar si el entrenamiento de los modelos se realizó tomando en cuenta características relevantes en los objetos de interés. En la Figura 3.1 se observa el flujo de trabajo seguido para evaluar el proceso de caracterización. Se comenzó con un preprocesamiento de las imágenes de los conjuntos de datos. Seguido, se realizó el entrenamiento de los modelos de clasificación. Una vez entrenados los modelos, se visualizaron las características encontradas durante la caracterización utilizando los métodos de visualización, generando mapas de calor. Por último, se aplicaron los índices de evaluación a los mapas de calor, evaluando así el proceso de caracterización de los modelos mediante el porcentaje de cobertura del objeto de interés.

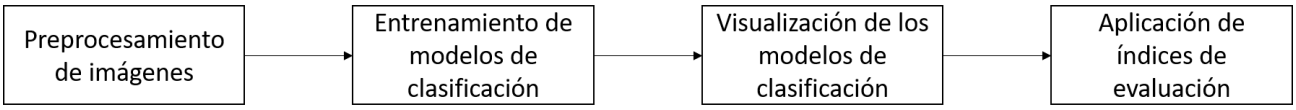
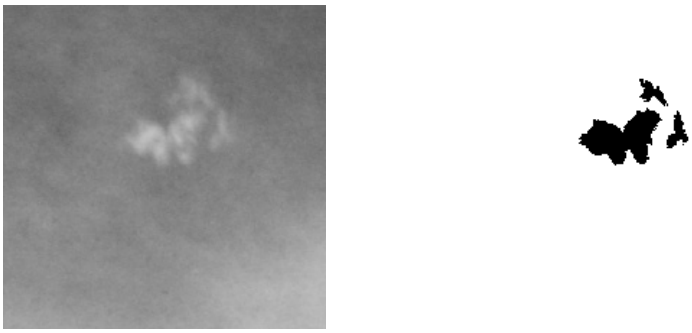


Figura 3.1: Flujo de trabajo seguido para la evaluación del proceso de caracterización.

3.1. Preprocesamiento de imágenes

Se realizó un preprocesamiento de imágenes para los dos conjuntos de datos utilizados. Ya que el conjunto de plantas contaba con múltiples órganos distintivos (flor, hoja, tallo, corteza, entre otros), delimitamos el conjunto a seleccionar la flor como órgano distintivo, tomando en cuenta solamente especies donde la flor fuera un órgano distintivo frecuente en las imágenes. En el caso del conjunto de mamografías, se tomaron en cuenta solamente las zonas señaladas como importantes, realizando recortes a esas zonas.

Se definió un conjunto de imágenes de visualización para los conjuntos de datos, siendo utilizado para aplicar los índices de evaluación del proceso de caracterización. Para poder calcular el porcentaje de cobertura, es necesario tener definido el área del objeto de interés. Por esta razón, para cada imagen del conjunto de visualización se generó una máscara de manera manual. Podemos observar en la Figura 3.3 un ejemplo de la generación de máscaras de objetos de interés tanto para el conjunto de plantas como para el conjunto de mamografías.



3.2. Entrenamiento de modelos de clasificación

Tanto para el conjunto de imágenes de plantas como para el conjunto de imágenes de mamografías se dividieron para formar un conjunto de entrenamiento y un conjunto de prueba. Se

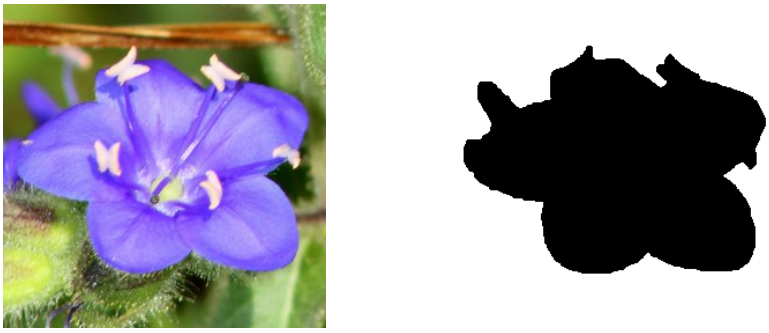


Figura 3.3: Generación de máscaras de los objetos de interés.

utilizaron los conjuntos de entrenamiento para entrenar modelos de clasificación basados en las siguientes arquitecturas de RNC.

- VGG19
- InceptionV3
- ResNet50
- InceptionResNet
- Xception
- DenseNet 121
- DenseNet 169
- DenseNet 201
- MobileNet V1
- MobileNet V2

Una vez entrenados los modelos de clasificación, se utilizaron los conjuntos de pruebas para calcular la precisión de los modelos. Estos modelos de clasificación entrenados fueron utilizados para aplicar los métodos de visualización.

3.3. Visualización de los modelos de clasificación

Se realizó un estudio comparativo entre tres diferentes métodos de visualización, siendo estos mapas de saliencia, Grad-CAM++ y Score-CAM. Mapas de saliencia es un método de visualización basado en los valores de gradiente, Grad-CAM++ es un método de visualización basado tanto en los valores de gradiente como en los valores de activación, mientras que Score-CAM es un método de visualización basado en perturbaciones. Al estar basados en diferentes valores, estos métodos nos permiten analizar qué tipo de método de visualización calcula mayor cobertura en los objetos de interés de los conjuntos de datos utilizados en este trabajo de tesis.

3.3.1. Mapas de saliencia

Este método se deriva del concepto de saliencia en las imágenes. La saliencia se refiere a características distintivas de la imagen en el contexto del procesamiento visual. Propuesto por Symonyan et ál. [48] en el 2013, los mapas de saliencia tienen como propósito representar la visibilidad (o "saliencia") en cada ubicación en la imagen. Los mapas de saliencia generan mapas de calor que se utilizan para diferenciar las características visuales en las imágenes. Los mapas de calor son calculados multiplicando la imagen de entrada con los valores de pesos asociados a la clase a visualizar, calculando los valores de pesos utilizando el valor de gradiente.
Descrito de manera formal, dada una imagen I, una clase c y una clasificación del modelo de RNC con un puntaje para la clase $S_c(I)$, se valora la importancia de los pixeles I con respecto a la influencia en la predicción $S_c(I)$. Calculando los valores de pesos para la clase c como: $w = \frac{\partial S_c}{\partial I}$ siendo w el resultado del valor de gradiente de S_c con respecto a la imagen.
Debe de considerarse el puntaje del modelo para la clase c como:

$$S_c(I) = wI + b \tag{3.1}$$

donde la imagen I es representada en forma de vector (de una dimensión), además de w ser los valores de peso para la clase c y b el valor de sesgo del modelo. La magnitud de los valores en w define la importancia de los píxeles en I para la clase c.

3.3.2. Grad-CAM++

EL mapeo de activación de clases por gradiente ponderado ++ (Grad-CAM++) al igual que Grad-CAM, es un método de visualización basado en CAM, propuesto por Chattopadhyay et ál. en el 2018 [9]. Si bien Grad-CAM es un método de visualización basados en gradientes, cuenta con limitaciones al momento de encontrarse con múltiples objetos. En este tipo de casos su rendimiento tiende a disminuir. Además, en imágenes de un solo objeto, los mapas de calor resultantes no capturan el objeto completo en su totalidad. Para dar solución a estas limitantes, se propuso el método Grad-CAM++.

La localización de un objeto en una imagen es importante para la visualización utilizando Grad-CAM, afectando el hecho de existir múltiples ocurrencias de un objeto con orientaciones o vistas ligeramente diferentes. Grad-CAM++ define α_k^c para capturar la importancia de un mapa de características particular A_k de una clase c. En este método, α_k^c son los coeficientes de ponderación para los gradientes de píxeles para la clase c y para el mapa de características A_k calculados de la siguiente forma:

$$\alpha_k^c = \frac{1}{Z} \sum_{x,y} w_{xy}^{kc} \cdot ReLU(\frac{\partial y^c}{\partial A_{x,y}^k}) \tag{3.2}$$

donde w_{xy}^{kc} son los coeficientes de correlación para el valor de gradiente de los pixeles para la clase c en el mapa de características A_k. Al utilizar w_{xy}^{kc}, se toman en cuenta los valores de múltiples objetos de la clase c contenidos en la imagen, obteniendo un mapa de calor que abarque los múltiples objetos de la siguiente forma:

$$Grad - CAM + +^c = \sum_k \alpha^{kc} \cdot A_{x,y}^k \tag{3.3}$$

3.3.3. Score-CAM

El mapeo de activación de clases ponderado por puntaje (Score-CAM) es un método de visualización desarrollado por Wang et ál. en el 2019 [56]. Este método propone una mejora respecto a otros métodos basados en CAM, tratando de resolver los problemas de ruidos irrelevantes y generar visualizaciones más claras y limpias. Los problemas de ruido se originan por los cambios de valores en el gradiente. Wang et ál. proponen un método basado en perturbaciones que enmáscara una parte de las regiones en la entrada original de la red y analizan cómo va

cambiando el puntaje de la predicción. Esta máscara de activación obtenida es tratada como un tipo de máscara para la entrada de la imagen, haciendo que el modelo prediga sobre la imagen parcialmente enmáscarada. El puntaje en la clase a visualizar se utiliza para representar la importancia en el mapa de características.

Socre-CAM, a diferencia de métodos basados en CAM, no hace uso del valor del gradiente, ya que Wang et ál. teorizaron con la inestabilidad de los valores de gradiente, siendo valores inestables y generadores de ruido aleatorio en los mapas de calor resultantes. En su trabajo los autores realizaron un estudio cuestionando la estabilidad del gradiente mostrando que, alterando un poco la imagen de entrada, aún imperceptible para el ojo humano, eso puede llegar a alterar notablemente al valor del gradiente, alterando a la visualización de los modelos.

A diferencia de Grad-CAM y Grad-CAM++ que utilizan la última capa convolucional para obtener el valor de gradiente y así obtener los mapas de características, Score-CAM utiliza la predicción modelo para obtener los mapas de características. Al obtener los mapas de características de esta manera, se elimina la dependencia al valor del gradiente, haciendo que Score-CAM funcione de manera más general [56].

Dada una imagen I utilizada como entrada al modelo de clasificación, se realiza una normalización de los mapas de características de la última capa convolucional de la siguiente manera:

$$A_{ij}^{k} = \frac{A_{ij}^{k}}{\max A^{k} - \min A^{k}} \tag{3.4}$$

Una vez normalizados los mapas de características, se proyectan las áreas resaltadas de los mapas de características normalizados de tamaño $1 \times W \times H$ multiplicándolos por la imagen I de tamaño $3 \times W \times H$ para obtener una imagen enmáscarada M de tamaño $3 \times W \times H$:

$$M^{k} = A^{k} \cdot I \tag{3.5}$$

Las imágenes enmáscaradas M^{k} son utilizadas como entrada al modelo, obteniendo las predicciones del modelo:

$$S^{k} = Softmax(F(M^{k})) \tag{3.6}$$

donde F es el modelo. Al obtener las predicciones, se extrae el valor de la predicción para la clase a visualizar c:

$$\alpha_{k}^{c} = S_{k}^{c} \tag{3.7}$$

donde α_k^c representa la importancia del mapa k para la clase c. Por último, se calcula el mapa de calor resultante de la siguiente forma:

$$Score - CAM^c = ReLU(\sum_k \alpha_k^c A^k) \tag{3.8}$$

donde se realiza una suma de la multiplicación de la importancia α_k^c para cada mapa de características A^k, aplicando una función ReLU para tomar en cuenta solamente las características positivas para la clase c.

3.4. Aplicación de índices de evaluación

En este trabajo de tesis se propone como medida de evaluación del proceso de caracterización el porcentaje de cobertura del objeto de interés contenido en las imágenes. Siendo así que, mientras mayor porcentaje de cobertura al objeto de interés, mayor calidad en el proceso de caracterización. Sumado a esto, se propone la elaboración de tres diferentes índices de evaluación para el proceso de caracterización.

3.4.1. Porcentaje de cobertura al objeto de interés en condiciones favorables en la imagen

En este índice de evaluación se aplica un método de visualización y se calcula el porcentaje que se cubre del objeto de interés en la imagen. Para esta evaluación, se considera implementar este índice en imágenes con las condiciones óptimas de caracterización, tales como objeto de interés centrado, sin elementos adicionales en la imagen y con una iluminación que no distorsione al objeto de interés. Se toman estas medidas en las imágenes para conocer la calidad de la caracterización en situaciones favorables. Podemos ver en la Figura 3.4 un ejemplo de una imagen de la clase *Mirabilis jalapa L. (1753)* del conjunto de plantas en las condiciones propuestas en este índice de evaluación.

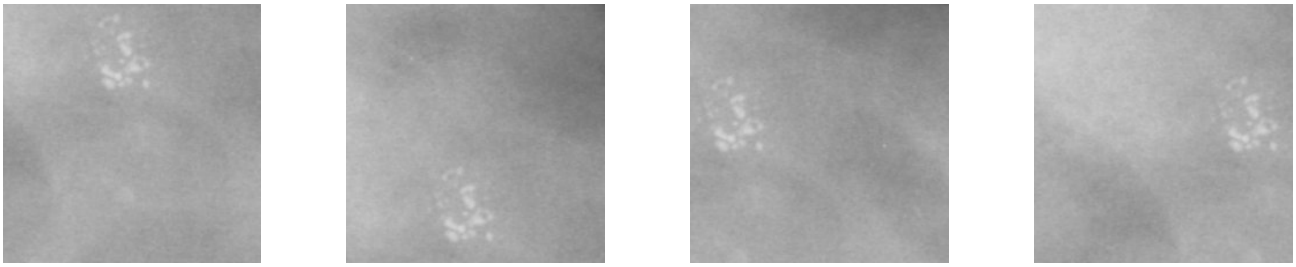
Figura 3.4: Imagen de la clase *Mirabilis jalapa L. (1753)* con las condiciones favorables.

3.4.2. Porcentaje de cobertura al objeto de interés en diferentes posiciones de la imagen

Para este índice de evaluación se propone trasladar el objeto de interés en diferentes posiciones de la imagen:

- En la parte superior de la imagen.
- En la parte inferior de la imagen.
- En la parte derecha de la imagen.
- En la parte izquierda de la imagen.

Con el traslado del objeto de interés en diferentes posiciones de la imagen se considera evaluar si, al aplicar un método de visualización, la caracterización se ve afectada por la posición en que se encuentra el objeto de interés, pudiendo reducirse el porcentaje de cobertura del objeto o cambiar las zonas de cobertura. En la Figura 3.5 podemos observar un ejemplo de la clase maligno del conjunto de datos de mamografías con el objeto de interés posicionado en las cuatro diferentes posiciones de la imagen propuestas en este índice de evaluación.



(a) Superior (b) Inferior (c) Izquierda (d) Derecha

Figura 3.5: Imagen de la clase maligno en las cuatro posiciones propuestas en el índice en diferentes posiciones.

3.4.3. Porcentaje de cobertura al objeto de interés con múltiples objetos en la imagen

En este índice de evaluación se aplica un método de visualización en una imagen donde el objeto de interés no es el único objeto contenido en la imagen. Las imágenes utilizadas para este índice de evaluación cuentan con múltiples objetos contenidos en ella. Esto se realizó con el fin de evaluar la capacidad de caracterizar el objeto de interés ignorando los objetos accesorios en la imagen. Es importante indicar que los objetos accesorios contenidos en las imágenes pueden no pertenecer a la misma clase. Se denomina como objeto accesorio a un objeto perteneciente a una clase que no es el objeto principal para distinguir a la clase. Tomando como ejemplo al conjunto de datos de plantas, donde las hojas o tallos encontrados en las imágenes pueden ser de diferentes clases, resaltando importancia de ignorar estos objetos accesorios durante el proceso de caracterización.

Podemos ver un ejemplo en la Figura 3.6 con una imagen de la clase *Solanum hindsianum Benth. (1844)* del conjunto de datos de plantas, donde existen objetos accesorios a la flor como lo son hojas y tallos.

Figura 3.6: Imagen de la clase *Solanum hindsianum Benth. (1844)* con las condiciones propuestas en el índice con múltiples objetos.

Capítulo 4

Evaluación experimental y resultados

En este capítulo se presenta la evaluación experimental realizada durante este trabajo de tesis. Se contempla la evaluación del proceso de caracterización (utilizando los índices de evaluación propuestos en el Capítulo 3) de los modelos de clasificación entrenados utilizando las diferentes arquitecturas de RNC mencionadas en el Capítulo 3.
Los experimentos fueron ejecutados utilizando una estación de trabajo con sistema operativo Ubuntu 18.04, un procesador Intel Xeon W-2133 de 12 núcleos a 3.6GHz, 32 GB de almacenamiento primario RAM, 2 TB de almacenamiento secundario tipo HDD interfaz SATA y una unidad de procesamiento de gráficos (GPU) NVIDIA GTX 1080 con 8GB de RAM.
Para entrenar los modelos de clasificación, se utilizó un script de Python 3.7.8 que hace uso de la biblioteca Keras 2.2.4 con TensorFlow 1.13.1 como *back-end*. Se utilizaron los siguientes parámetros de entrenamiento.

- 100 épocas.
- Tamaño de lote de 8.
- Descenso Estocástico de Gradiente (DEG) como función de optimización.
- Entropía cruzada binaria como función de pérdida.
- Tasa de aprendizaje de 0.01.

El número de épocas es la cantidad de veces que se recorre el conjunto de entrenamiento durante el entrenamiento. El tamaño de lote es la cantidad de elementos que se utilizan como información

de entrada en la red. La función de optimización es la función que se encarga de optimizar los parámetros de la red para reducir el error. La función de pérdida es la función que calcula el error en la red. La tasa de aprendizaje es el valor que controla el tamaño de las modificaciones a los parámetros en la red.
Para visualizar las características, se utilizaron los métodos de visualización mapas de saliencia (basado en gradiente), GradCam++ (basado en gradiente y activaciones) y ScoreCAM (basado en perturbaciones).

4.1. Conjuntos de datos

Los experimentos reportados en este trabajo se realizaron utilizando dos conjuntos de datos. El primero fue un conjunto de imágenes de plantas y el segundo fue un conjunto de imágenes de mamografías de cáncer de mama. Para realizar los experimentos, ambos conjuntos de datos se dividieron en un conjunto de entrenamiento y un conjunto de prueba, con una relación del 75-25 %. Además, se utilizó un conjunto de datos de visualización para realizar los experimentos de visualización de características. Para el conjunto de imágenes de plantas, se eligieron aleatoriamente 10 especies de plantas, con 10 imágenes por especies. Por su parte, se eligieron las dos clases del conjunto de imágenes de mamografías, con 10 imágenes por clase.

4.1.1. Conjunto de imágenes de plantas

El conjunto de imágenes de plantas cuenta con imágenes correspondientes a 80 especies de plantas nativas de México. La conformación de este conjunto de datos se basa en dos fuentes diferentes. La primera fuente de imágenes es de salidas a campos realizadas por equipos de fotógrafos, ayudados por botánicos expertos en taxonomía de la flora de México. La segunda fuente de imágenes es de descargas del sitio web Naturalista [1], que es un sitio web dedicado a recopilar imágenes e información de diferentes especies de plantas nativas de México, tomando en cuenta para el conjunto de datos solamente especies de plantas que cuenten con flores.
Este conjunto de datos se conforma de 80 clases, siendo cada clase una especie distinta de planta, 50 clases obtenidas de salidas a campo y 30 clases obtenidas del sitio web Naturalista. Cada clase cuenta con 100 imágenes recortadas de flores, contando el conjunto de datos con un

Figura 4.1: Comparación entre dos imágenes diferentes de la especie *Dahlia coccinea Cav. (1791)*

total de 8000 imágenes recortadas diferentes.

Dentro de este conjunto de datos existen clases que cuentan con alta variabilidad intraclase, es decir, son ejemplares de la misma especie que lucen diferentes entre ellas. Podemos ver en la Figura 4.1 un ejemplo de alta variabilidad intraclase entre la especie *Dahlia coccinea Cav. (1791)*, con dos diferentes imágenes de la misma especie que cuentan con diferencias como el color de la flor y forma de los pétalos.

Este conjunto de datos cuenta también con clases con baja variabilidad interclase, es decir, con especies diferentes que cuentan con similitudes considerables entre ellas. Podemos ver en la Figura 4.2 un ejemplo de baja variabilidad interclase entre las especies *Ludwigia octovalvis (Jacq.) P.H. Raven (1962)* y *Ludwigia peploides (Kunth) P.H. Raven (1963)*, contando con similitudes como la forma, color y cantidad de los pétalos.

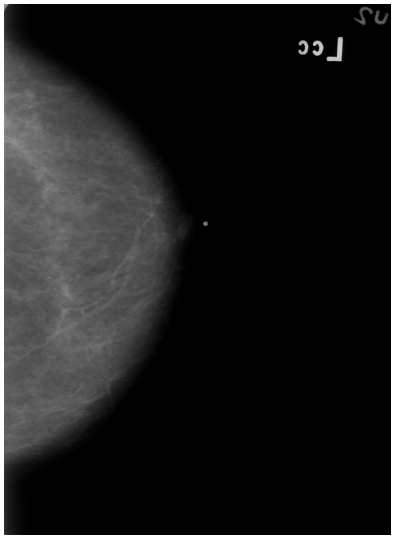
(a) *Ludwigia octovalvis (Jacq.) P.H. Raven (1962)*

(b) *Ludwigia peploides (Kunth) P.H. Raven (1963)*

Figura 4.2: Comparación entre las especies (a) *Ludwigia octovalvis (Jacq.) P.H. Raven (1962)* y (b) *Ludwigia peploides (Kunth) P.H. Raven (1963)*

4.1.2. Conjunto de imágenes de mamografías

Se utilizó el conjunto de imágenes de CBIS-DDSM (subconjunto de datos de imágenes mamarias curadas de la base de datos digital para mamografías proyectadas) [27]. El DDSM es una base de datos de 2620 estudios de mamografía con película escaneada. Este conjunto de datos contiene casos normales, benignos y malignos con información patológica verificada. El conjunto CBIS-DDSM es un subconjunto de los datos DDSM que fueron seleccionados y curados por un mamógrafo capacitado. Cada muestra se compone de una imagen de mamografía y uno o más recortes del área de interés detectada en la mamografía por médicos expertos, que señalaron áreas importantes en la mamografía para clasificarla como benigno o maligno. Para este conjunto de datos, tomamos en cuenta las mamografías diagnosticadas con calcificaciones mamarias. Las calcificaciones mamarias son un tipo de esclerosis que afecta a las arterias mamarias en un tamaño pequeño o mediano [45]. Se han realizado estudios que asocian a las calcificaciones mamarias con el cáncer de mama [7]. Por esta razón, tomamos en cuenta los casos benignos y malignos ya que nos interesa identificar si una calcificación en la mamografía es benigna o maligna, permitiendo así detectar un posible caso de cáncer de mama. En la Figura 4.3 podemos ver un ejemplo de una mamografía con calcificación benigna y una mamografía con calcificación maligna. Al igual que el conjunto de imágenes de plantas, el conjunto de imágenes

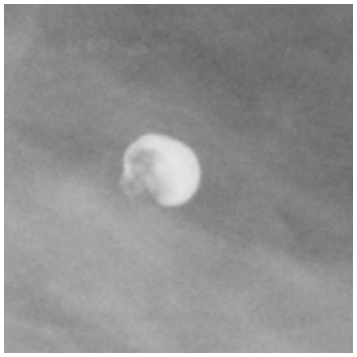


(a) Mamografía con calcificación benigna

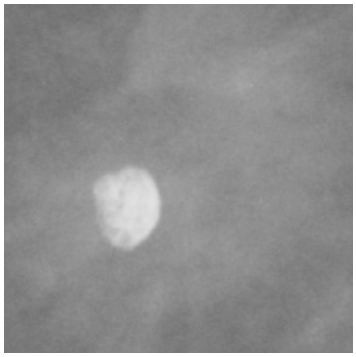
(b) Mamografía con calcificación maligna

Figura 4.3: Ejemplo de una imagen de (a) mamografía con calcificación benigna y una imagen de (b) mamografía con calcificación maligna

de mamografías cuenta con baja variabilidad intraclase. Podemos observar esto en la Figura 4.4, donde tanto la calcificación benigna como maligna cuenta con una forma circular, lo que hace complicado diferenciar entre estos casos.



(a) Ejemplo de calcificación benigna



(b) Ejemplo de calcificación maligna

Figura 4.4: Comparación entre una (a) calcificación benigna y una (b) calcificación maligna

4.2. Exactitud de los modelos

El resumen de los resultados de los modelos de clasificación entrenados con el conjunto de imágenes de plantas se muestran en la Tabla 4.1. Se tomaron como índices de evaluación la exactitud Top-1 y Top-5, es decir, si la clase correcta fue el valor de predicción más alto o si se encontró dentro de los cinco valores de predicción más altos.

Tabla 4.1: Porcentaje de exactitud de los modelos de clasificación entrenados con el conjunto de imágenes de plantas.

Arquitectura de RNC	**% Top-1**	**% Top-5**
DenseNet121	55.05	81.05
DenseNet169	52.45	80.50
DenseNet201	53.10	79.25
InceptionResNet	67.35	87.75
InceptionV3	59.70	83.45
MobileNetV1	59.80	85.50
ResNet50	54.60	80.85
VGG19	39.75	65.85
Xception	71.15	88.75

Los tres modelos de clasificación que tuvieron el mejor desempeño fueron los modelos basados en las arquitecturas Xception, InceptionResNet y MobileNetV1.

- **Xception** con 71.15 % de exactitud Top-1 y 88.75 % de exactitud Top-5,
- **InceptionResNet** con 67.35 % de exactitud Top-1 y 87.75 % de exactitud Top-5, y
- **MobileNetV1** con 59.80 % de exactitud Top-1 y 85.50 % de exactitud Top-5.

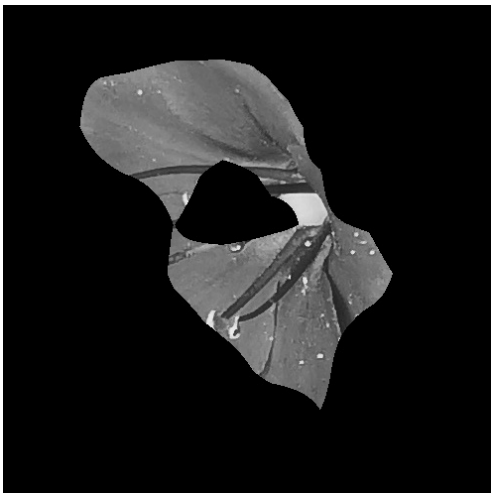
En el caso de los modelos de clasificación entrenados con el conjunto de imágenes de mamografías, se tomó solamente como medida de evaluación la exactitud Top-1 al tener solo dos clases en el conjunto de datos. Los resultados se muestran en la Tabla 4.2. Los tres modelos de

Tabla 4.2: Porcentaje de exactitud de los modelos de clasificación entrenados con el conjunto de imágenes de mamografías.

Arquitectura de RNC	**% Top-1**
DenseNet121	49.15
DenseNet169	49.43
DenseNet201	49.43
InceptionResNet	49.15
InceptionV3	50.56
MobileNetV1	50.28
MobileNetV2	50.00
ResNet50	49.71
VGG19	49.43
Xception	50.00

clasificación que tuvieron el mejor desempeño fueron los modelos de clasificación basados en las arquitecturas InceptionV3, MobileNetV1 y Xception.

- **InceptionV3** con 50.56 % de exactitud,
- **MobileNetV1** con 50.56 % de exactitud y
- **Xception** con 50.00 % de exactitud.



(a) Área del mapa de calor resultante

(b) Área de la máscara

Figura 4.5: Representación gráfica de los elementos necesarios para calcular el porcentaje de cobertura, donde es (a) el área del mapa de calor resultante y es (b) el área de la máscara

4.3. Visualización de los modelos

Se visualizaron los modelos entrenados por las arquitecturas de RNC para ambos conjuntos de datos. Para los tres métodos de visualización se eligió visualizar la última capa convolucional de cada modelo, ya que la última capa convolucional cuenta con todas las características encontradas a lo largo del proceso de caracterización. El resultado de la visualización se representa como un mapa de calor, tomando a este mapa de calor como una máscara, considerando solamente los valores del mapa de calor que superaran un umbral del 45 % del valor máximo posible, determinando este valor una vez realizado pruebas donde porcentajes más altos omitían información y porcentajes más bajos tomaban en cuenta valores con bajas valoraciones. Esta máscara se aplica al área del objeto de interés para calcular el porcentaje de cobertura. En la Figura 4.5 se muestra una representación gráfica de los elementos necesarios para calcular el porcentaje de cobertura de un método de visualización. Sea A el área del mapa de calor resultante, medida como el número de pixeles, sea B el área de la máscara. El índice de cobertura se calcula como la razón de A sobre B y se expresa como un porcentaje.

Se realizó el cálculo de correlación entre el porcentaje de cobertura de cada método de visua-

lización con respecto a la predicción del modelo para la clase de la imagen a visualizar. Por ejemplo, si la imagen a visualizar corresponde a la clase *Cordia dodecandra DC. (1845)* del conjunto de imágenes de plantas, se calcula la correlación para cada método de visualización entre su porcentaje de cobertura con respecto al valor de predicción del modelo para la clase *Cordia dodecandra DC. (1845)*. Al realizar este cálculo de correlación, se busca encontrar una relación entre la exactitud del modelo con el porcentaje de cobertura de los métodos de visualización. La correlación puede tomar un valor entre -1 y 1, mientras más cercano sea el valor a 1 indica una correlación positiva mientras que más se acerque a -1 indica una correlación negativa.

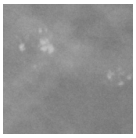
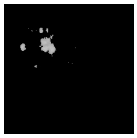
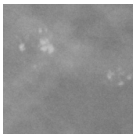
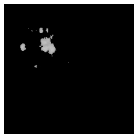
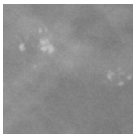
4.3.1. Resultados índice con objetos en condiciones favorables

Se reportan los resultados obtenidos de calcular el porcentaje de cobertura al objeto de interés en la imagen, donde la imagen se encuentra en condiciones favorables para la caracterización tales como objeto de interés centrado, sin elementos adicionales en la imagen y sin distorsión de la imagen por parte de la iluminación. En la Tabla 4.3 podemos observar un ejemplo de los resultados obtenidos, mostrándose el método de visualización, la imagen, el resultado visual y el porcentaje de cobertura. Se observan resultados favorables, ya que el porcentaje de cobertura se encuentra por encima del 85 %, lo que nos indica que el modelo se enfocó en el objeto de interés durante el proceso de caracterización.

Conjunto de imágenes de plantas

El resumen de los resultados por método de visualización se muestra en la Tabla 4.4 para Grad-CAM++, la Tabla 4.5 para Score-CAM y la Tabla 4.6 para mapas de saliencia, contando estas tablas con el porcentaje de cobertura promedio y el valor de correlación entre la predicción del modelo con el porcentaje de cobertura del método.
Analizando los resultados de la Tabla 4.4 se observan como las arquitecturas InceptionV3 y MobileNetV2 obtienen un porcentaje de cobertura de 0 % y como la arquitectura MobileNetV1 obtiene un porcentaje de cobertura de 99.99 % pero con un valor de correlación de -0.06. Al observar los mapas de calor resultantes, notamos como los mapas de calor se encontraban completamente en azul para las arquitecturas InceptionV3 y MobileNetV2, lo que significa que el

Tabla 4.3: Muestra de resultado índice en condiciones favorables, presentando buenos porcentajes de cobertura por encima del 85 %.

Método	**Imagen**	**Resultado visual**	**% Cobertura**
Grad-CAM++			90.11
Score-CAM			89.86
Mapas de saliencia			98.75

modelo no se enfocó en ninguna zona de la imagen para visualizar las características. Por su parte, los mapas de calor resultantes de MobileNetV1 se encontraban completamente en rojo. Esto significa que el modelo se enfocó en toda la imagen durante la caracterización, lo que hace que el modelo no sea capaz de encontrar al objeto de interés dentro de la imagen. Ambos casos nos indica que existe un problema con el valor de gradiente del modelo, por lo que Grad-CAM++ genera resultados atípicos.

Tomando en cuenta los resultados de la Tabla 4.5 se destaca a la arquitectura InceptionResNet con un porcentaje de cobertura de 96.25 % pero con un valor de correlación de -0.14. Esto se debe a que, al obtener porcentajes de cobertura altos, la correlación se verá afectada cuando la predicción del modelo sea baja. Un ejemplo de esto se muestra tomando en cuenta a la Figura 4.6 que es una imagen de la clase *Ipomea purpurea (L.) Roth (1787)* del conjunto de imágenes de plantas, donde el porcentaje de cobertura fue de 100 % pero la predicción del modelo fue de 0.5 % para la clase *Ipomea purpurea (L.) Roth (1787)*. Esto significa que el enfocarse en el objeto de interés no fue suficiente para clasificar correctamente a la imagen.

Otra arquitectura a destacar es Xception, que obtuvo un porcentaje de cobertura de 24.87 % pero con una exactitud de 71.15 % Top-1. Esto nos indica que para Xception, no es necesario

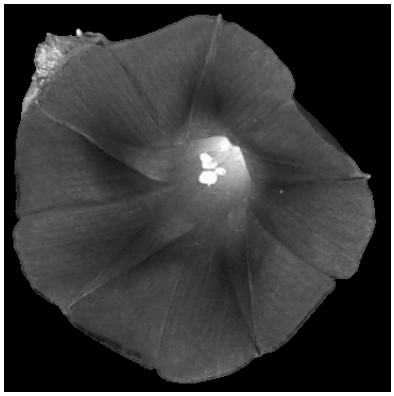


Figura 4.6: Imagen de la clase *Ipomea purpurea (L.) Roth (1787)* con un 100 % de cobertura utilizando Score-CAM en la arquitectura InceptionResNet

Figura 4.7: Imagen de la clase *Ipomea purpurea (L.) Roth (1787)* con un 5.78 % de cobertura utilizando Score-CAM en la arquitectura Xception

enfocarse en todo el objeto de interés para clasificarlo de manera correcta. Tomando de ejemplo a la Figura 4.7 que es una imagen de la clase *Ipomea purpurea (L.) Roth (1787)* del conjunto de imágenes de plantas, donde el porcentaje de cobertura fue de 5.78 % pero la predicción del modelo fue de 99.99 % para la clase *Ipomea purpurea (L.) Roth (1787)*. Esto significa que el modelo encontró que características eran las necesarias para clasificar de manera correcta la imagen, sin necesidad de enfocarse en el objeto de interés por completo.

Podemos observar en la Figura 4.8 como el método Grad-CAM++ obtuvo el mejor porcentaje de cobertura en general. Mientras que Score-CAM obtuvo un porcentaje de cobertura promedio mayor entre las arquitecturas. El elevado desempeño de Grad-CAM++ con un 99.99 % se debió

a un problema con el valor de gradiente, por lo que no se considera un porcentaje confiable. La estrategia de mapas de saliencia obtuvo la peor cobertura en la mayoría de arquitecturas. El desempeño tanto de Grad-CAM++ como de mapas de saliencia se ve afectado por el valor del gradiente. Grad-CAM++ y mapas de saliencia presentaron errores al momento de implementarse en arquitecturas como InceptionV3, MobileNetV1 y MobileNetV2. Se puede intuir que hubo problemas con el valor de gradiente durante el entrenamiento de los modelos, que puede ser generado por el gran número de capas y parámetros de la arquitectura InceptionV3 o debido al número reducido de parámetros combinado con las convoluciones separadas en profundidad de las arquitecturas MobileNetV1 y MobileNetV2. Al no utilizar el valor de gradiente, Score-CAM pudo visualizar el proceso de caracterización de las arquitecturas InceptionV3, MobileNetV1 y MobileNetV2.

En la Figura 4.9 se observa como Score-CAM es el método que obtiene el mayor valor de correlación promedio, mientras que mapas de saliencia obtiene un valor de correlación negativa en la mayoría de arquitecturas, por lo que existe una correlación inversa al porcentaje de cobertura del objeto. En general los valores de correlación no son altos, esto se debe a casos como el presentado para la arquitectura Xception y el método Score-CAM, donde con un porcentaje de cobertura bajo se puede generar una exactitud alta. Este tipo de casos se ven ejemplificados en la mayoría de arquitecturas para mapas de saliencia, que presenta una baja cobertura en promedio, por lo que no existe una fuerte correlación entre la cobertura de mapas de saliencia con la exactitud del modelo.

Tabla 4.4: Resultados índice en condiciones favorables Grad-CAM++ plantas.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**	**% Top-5**
DenseNet121	25.60	0.16	55.05	81.05
DenseNet169	26.30	0.11	52.45	80.50
DenseNet201	28.49	-0.05	53.10	79.25
InceptionResNet	2.30	-0.23	67.35	87.75
InceptionV3	0	NA	59.70	83.45
MobileNetV1	99.99	-0.06	59.80	85.50
MobileNetV2	0	NA	59.80	85.50
ResNet50	6.47	-0.07	54.60	80.85
VGG19	5.27	-0.12	39.75	65.85
Xception	24.87	0.25	71.15	88.75

Tabla 4.5: Resultados índice en condiciones favorables Score-CAM plantas.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**	**% Top-5**
DenseNet121	60.28	0.19	55.05	81.05
DenseNet169	50.24	0.13	52.45	80.50
DenseNet201	48.12	-0.04	53.10	79.25
InceptionResNet	96.25	-0.14	67.35	87.75
InceptionV3	68.59	0.47	59.70	83.45
MobileNetV1	10.00	0.18	59.80	85.50
MobileNetV2	48.42	0.11	59.80	85.50
ResNet50	7.07	0.04	54.60	80.85
VGG19	5.77	0.05	39.75	65.85
Xception	14.50	0.08	71.15	88.75

Tabla 4.6: Resultados índice en condiciones favorables mapa de saliencia plantas.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**	**% Top-5**
DenseNet121	15.12	-0.09	55.05	81.05
DenseNet169	12.70	-0.17	52.45	80.50
DenseNet201	12.18	-0.14	53.10	79.25
InceptionResNet	8.88	-0.21	67.35	87.75
InceptionV3	3.90	-0.09	59.70	83.45
MobileNet	16.19	-0.01	59.80	85.50
MobileNetV2	12.77	0.07	59.80	85.50
ResNet50	7.82	-0.20	54.60	80.85
VGG19	2.07	0.05	39.75	65.85
Xception	10.44	0.05	71.15	88.75

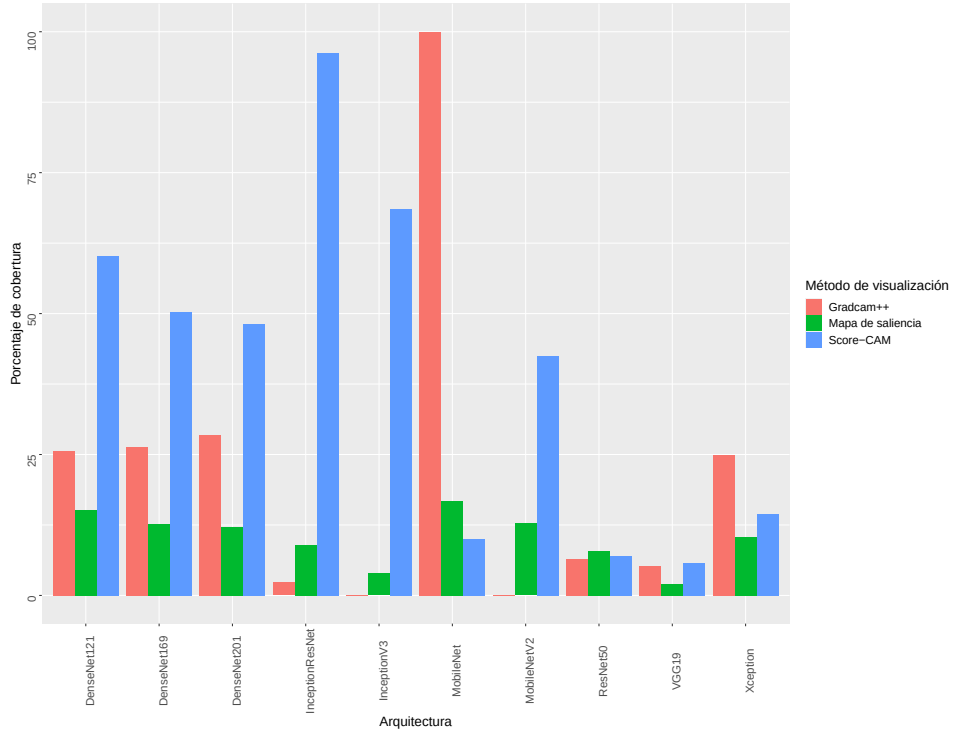


Figura 4.8: Resultados índice en condiciones favorables plantas porcentaje de cobertura.

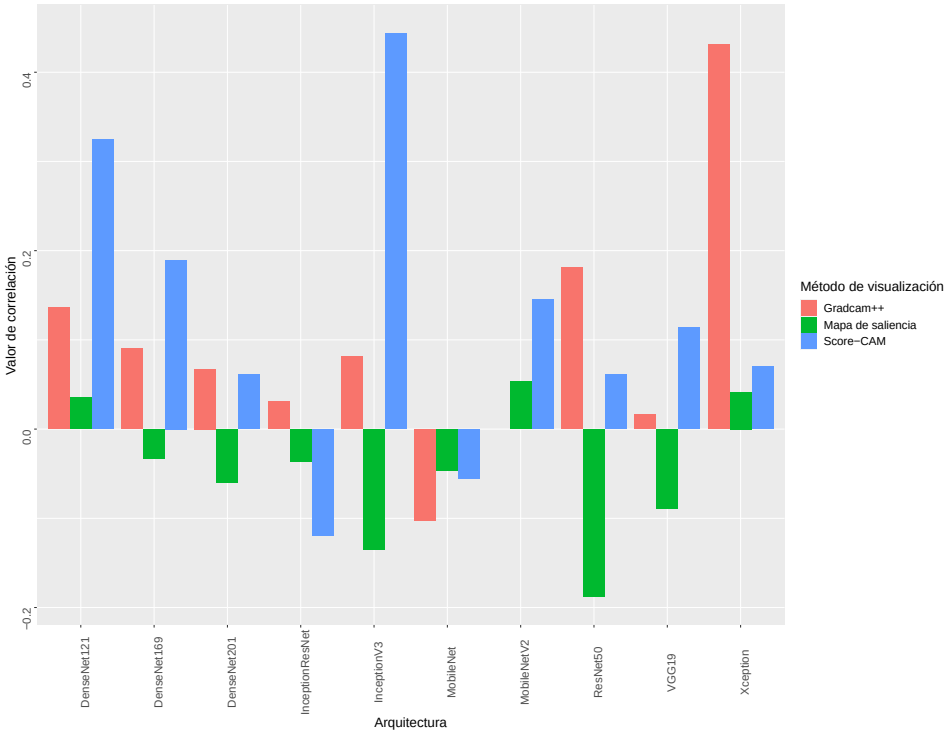


Figura 4.9: Resultados índice en condiciones favorables plantas valor de correlación.

Conjunto de imágenes de mamografías

El resumen de los resultados por método de visualización se muestra en la Tabla 4.7 para Grad-CAM++, la Tabla 4.8 para Score-CAM y la Tabla 4.9 para mapas de saliencia.
Los resultados en las Tablas 4.7 y 4.9 indican un comportamiento similar de bajo desempeño con la mayoría de las arquitecturas utilizando los métodos Grad-CAM++ y mapas de saliencia. Las arquitecturas DenseNet121, DenseNet169, DenseNet201, InceptionResNet y ResNet50 obtuvieron un porcentaje de cobertura de 0 % para ambos métodos, mientras que InceptionV3 y MobileNetV2 obtuvieron 0 % y 5 % utilizando Grad-CAM++ y 0.22 % y 0.26 % utilizando mapas de saliencia. Al contar el conjunto de imágenes de mamografías con solo dos clases y que ninguna de las arquitecturas previamente mencionadas superara el 51 % de exactitud, se pudo concluir que los modelos presentaron un problema de desvanecimiento de gradiente, ya que tanto el método Grad-CAM++ como mapas de saliencia fueron incapaces de visualizar utilizando el valor de gradiente. Solamente las arquitecturas VGG19 y Xception porcentajes de cobertura altos (99.81 % para VGG19 y 94.20 % para Xception).
Se puede apreciar en la Tabla 4.8 como se destaca el porcentaje de cobertura de las arquitecturas DenseNet169, VGG19 y Xception, que superan el 90 %. Al comparar el porcentaje de cobertura de las arquitecturas InceptionResNet y MobileNetV2 con respecto a los métodos Grad-CAM++ y mapas de saliencia se concluyó que estas arquitecturas no fueron capaces de caracterizar de manera correcta y que este problema va más allá a un problema con el gradiente. Podemos atribuir esto a la complejidad de las arquitecturas ya que una arquitectura sencilla como VGG19 fue capaz de caracterizar los objetos. En caso de Xception podemos atribuir su buen desempeño a las convoluciones separables en profundidad, sumado a la forma de realizar estas convoluciones.
Podemos observar en la Figura 4.10 como el método Score-CAM fue el método que obtuvo los mayores porcentajes de cobertura, mientras que Grad-CAM++ y mapas de saliencia solamente obtuvieron resultados por encima del 50 % en las arquitecturas VGG19 y Xception. Con los resultados en la gráfica se puede observar como en la mayoría de las arquitecturas los métodos Grad-CAM++ y mapas de saliencia fueran incapaces de visualizar las características, deduciendo que el valor de gradiente puede llegar a afectar en gran medida la visualización de

Tabla 4.7: Resultados índice en condiciones favorables Grad-CAM++ mamografías.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**
DenseNet121	0	NA	49.15
DenseNet169	0	NA	49.43
DenseNet201	0	NA	49.43
InceptionResNet	0	NA	49.15
InceptionV3	0	NA	50.56
MobileNetV1	5.00	0.01	50.28
MobileNetV2	0	NA	50.00
ResNet50	0	NA	49.71
VGG19	99.87	-0.23	49.43
Xception	93.72	0.19	50.00

características al utilizar estos métodos.

En la Figura 4.11 se observan resultados de correlación bajos en general, pudiendo notar en su mayoria solamente los resultados generados por Score-CAM. Los bajos valores de correlación se deben al obtener porcentajes altos de cobertura que llegan hasta el 90 % pero la exactitud del modelo no supera el 55 % en promedio, haciendo que el porcentaje de cobertura no influya en la predicción del modelo. Al tener una baja exactitud en los modelos pero un alto porcentaje de cobertura podemos concluir que las arquitecturas no son capaces de encontrar las suficientes características para diferenciar las clases entre sí.

Tabla 4.8: Resultados índice en condiciones favorables Score-CAM mamografías.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**
DenseNet121	2.22	0.12	49.15
DenseNet169	96.85	0.13	49.43
DenseNet201	56.66	-0.08	49.43
InceptionResNet	0	NA	49.15
InceptionV3	7.69	0.07	50.56
MobileNetV1	5.00	NA	50.28
MobileNetV2	0	NA	50.00
ResNet50	0.59	0.07	49.71
VGG19	99.81	-0.23	49.43
Xception	94.20	0.17	50.00

Tabla 4.9: Resultados índice en condiciones favorables por mapa de saliencia mamografías.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**
DenseNet121	0	NA	49.15
DenseNet169	0	NA	49.43
DenseNet201	0	NA	49.43
InceptionResNet	0	NA	49.15
InceptionV3	0.22	0.05	50.56
MobileNetV1	0.26	0.03	50.28
MobileNetV2	0.26	0.01	50.00
ResNet50	0	NA	49.71
VGG19	97.48	-0.32	49.43
Xception	97.07	0.43	50.00

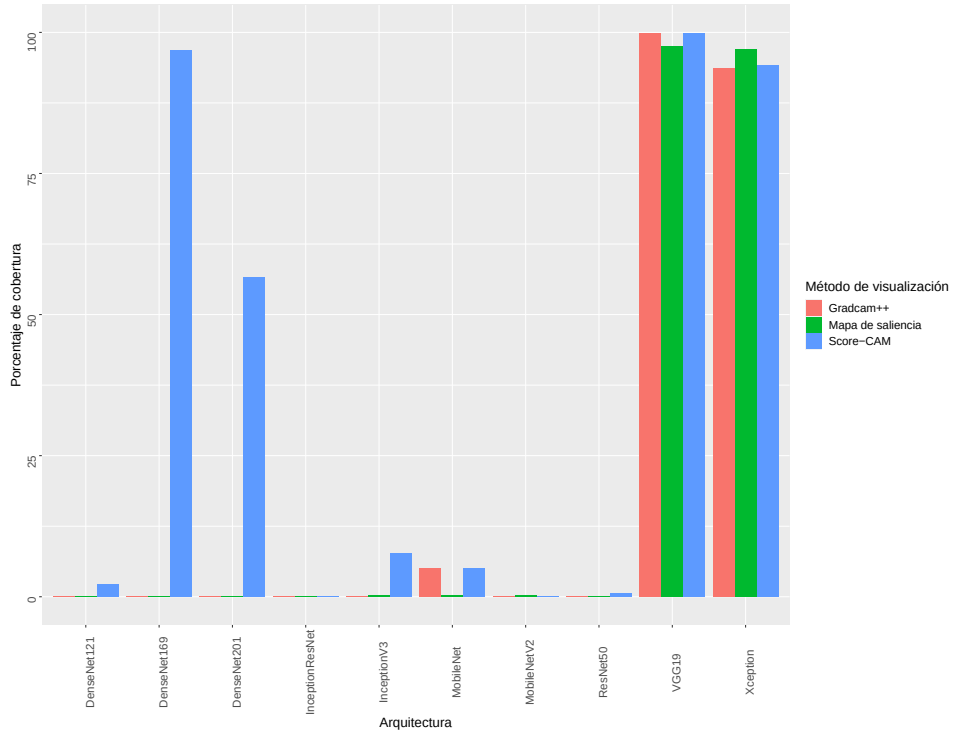


Figura 4.10: Resultados índice en condiciones favorables mamografías porcentaje de cobertura.

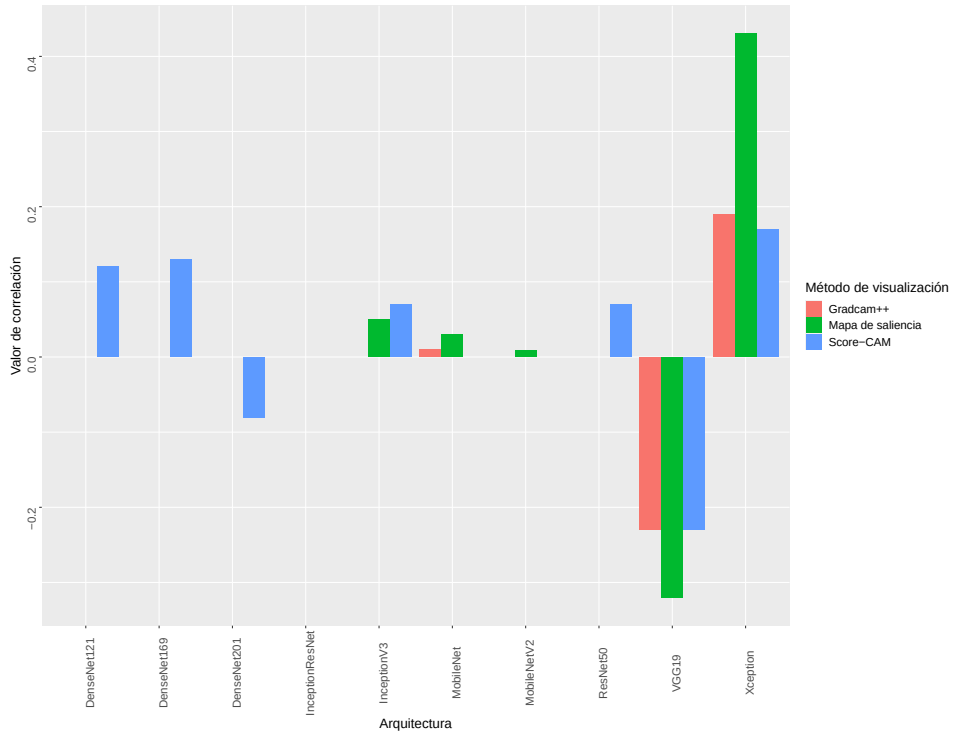
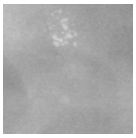
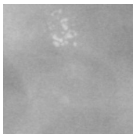
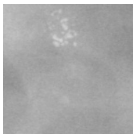


Figura 4.11: Resultados índice en condiciones favorables mamografías valor de correlación.

Tabla 4.10: Muestra de resultado índice en posición superior, presentando porcentajes de cobertura por encima del 90 %.

Método	**Imagen**	**Resultado visual**	**% Cobertura**
Grad-CAM++			100
Score-CAM			100
Mapas de saliencia			100

4.3.2. Resultados índice con objetos en diferentes posiciones

Se reportan los resultados obtenidos de calcular el porcentaje de cobertura al objeto de interés en diferentes posiciones en la imagen, con el propósito de evaluar si para el proceso de caracterización de los modelos la posición de los objetos de interés es relevante. En las Tablas 4.10,4.11,4.12 y 4.13 podemos observar un ejemplo de los resultados obtenidos del índice con diferentes posiciones de evaluación, tomando en cuenta al objeto de interés en la posición superior, inferior, izquierda y derecha de la imagen. En general se observan resultados favorables, con un porcentaje de cobertura por encima del 90 %, lo que nos indica que los modelos caracterizan de igual manera al objeto de interés en diferentes posiciones. Al igual que en los resultados del índice en condiciones favorables, se muestra el método de visualización, la imagen, el resultado visual y el porcentaje de cobertura.

Conjunto de imágenes de plantas

Los resúmenes de los resultados por método de visualización se muestran en las Tablas 4.14, 4.15 para Grad-CAM++, las Tablas 4.16, 4.17 para Score-CAM y las Tablas 4.18, 4.19 para mapas de saliencia.

Tabla 4.11: Muestra de resultado índice en posición inferior, presentando porcentajes de cobertura por encima del 90 %.

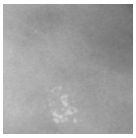
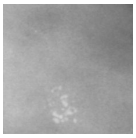
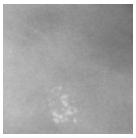
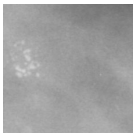
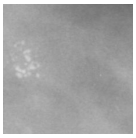
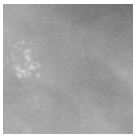
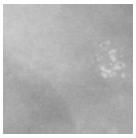
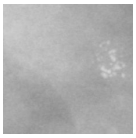
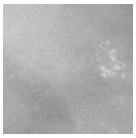
Método	**Imagen**	**Resultado visual**	**% Cobertura**
Grad-CAM++			100
Score-CAM			100
Mapas de saliencia			100

Tabla 4.12: Muestra de resultado índice en posición izquierda, presentando porcentajes de cobertura por encima del 90 %.

Método	**Imagen**	**Resultado visual**	**% Cobertura**
Grad-CAM++			100
Score-CAM			100
Mapas de saliencia			100

En la Tabla 4.14 se observa como los porcentajes de cobertura tuvieron una variación entre posición no mayor al 8 %, siendo DenseNet121 la arquitectura que presenta una mayor variación entre el objeto en la posición superior con respecto al objeto en la posición izquierda. Al igual que en el índice anterior, MobileNetV1 obtuvo una cobertura del 99.99 %, que al observar

Tabla 4.13: Muestra de resultado índice en posición derecha, presentando porcentajes de cobertura por encima del 90 %.

Método	**Imagen**	**Resultado visual**	**% Cobertura**
Grad-CAM++			100.00
Score-CAM			100.00
Mapas de saliencia			93.75

los mapas de calor resultantes, confirmamos que la arquitectura se enfocó de nuevo en toda la imagen durante la caracterización. Se destaca a la posición inferior como la posición con mejor porcentaje de cobertura en promedio. Los porcentajes de cobertura de las arquitecturas InceptionV3 y MobileNetV2 se mantienen en 0 % con respecto al índice anterior. Los resultados de correlación se observan en la Tabla 4.15, manteniendo una correlación baja al igual que los resultados del índice anterior.

En la Tabla 4.16 se observa los porcentajes de cobertura utilizando el método Score-CAM. En comparación a los resultados reportados en el índice anterior, el comportamiento de los modelos se mantiene, siendo la arquitectura InceptionResNet quien obtiene los mejores porcentajes de cobertura, seguida de InceptionV3. En general no existe cambios mayores al 8 % entre posiciones, siendo la posición superior la posición con mejor porcentaje de cobertura en promedio. Los resultados de correlación se observan en la Tabla 4.17, donde se mantienen en general valores de correlación a la baja, debido en su mayoría a valores de cobertura bajos con respecto a precisiones altas.

Como se observa en la Tabla 4.16, los porcentajes de cobertura por parte de mapas de saliencia no superan el 25 % en general, manteniendo valores bajos. Esto se ve reflejado en los valores de correlación de la Tabla 4.17, donde el mayor valor de correlación no supera el 0.25.

Tabla 4.14: Resultados índice con diferentes posiciones cobertura Grad-CAM++ plantas.

Arquitectura	**% Superior**	**% Inferior**	**% Izquierda**	**% Derecha**	**% Top-1**	**% Top-5**
DenseNet121	32.70	34.66	28.42	29.33	55.05	81.05
DenseNet169	36.96	38.25	34.39	35.31	52.45	80.50
DenseNet201	34.28	35.84	37.64	31.99	53.10	79.25
InceptionResNet	3.84	5.74	4.56	5.55	67.35	87.75
InceptionV3	0	0	0.60	0	59.70	83.45
MobileNetV1	99.99	99.99	99.99	99.99	59.80	85.50
MobileNetV2	0	0	0	0	59.80	85.50
ResNet50	14.67	11.78	13.24	11.91	54.60	80.85
VGG19	7.00	8.14	8.04	7.53	39.75	65.85
Xception	24.85	26.65	23.01	25.16	71.15	88.75

Podemos observar en las Figuras 4.12, 4.13, 4.14 y 4.15 4.12 que se mantuvo una constante entre los porcentajes de cobertura de los métodos con respecto a las posiciones del objeto en la imagen. Esto quiere decir que en general la posición del objeto de interés no es de relevancia para las diferentes arquitecturas utilizando el conjunto de imágenes de plantas, siendo capaces de caracterizar el objeto de interés sin importar su posición en la imagen. InceptionResNet es la arquitectura con los valores de cobertura más alto y Score-CAM el método de visualización con el mayor porcentaje de cobertura promedio.

Al igual que los porcentajes de cobertura, los valores de correlación se mantuvieron constantes sin importar la posición del objeto en la imagen, observando esto en las Figuras 4.16, 4.17,4.18 y 4.19.

Tabla 4.15: Resultados índice con diferentes posiciones correlación Grad-CAM++ plantas.

Arquitectura	Superior	Inferior	Izquierda	Derecha	% Top-1	% Top-5
DenseNet121	0.19	0.32	0.14	0.09	55.05	81.05
DenseNet169	0.05	-0.08	0.09	0.10	52.45	80.50
DenseNet201	0.16	0.03	0.07	0.07	53.10	79.25
InceptionResNet	-0.05	-0.06	0.03	-0.10	67.35	87.75
InceptionV3	NA	NA	0.08	NA	59.70	83.45
MobileNetV1	-0.08	-0.05	-0.10	-0.09	59.80	85.50
MobileNetV2	NA	NA	NA	NA	59.80	85.50
ResNet50	0.29	0.26	0.18	0.31	54.60	80.85
VGG19	0.16	-0.07	0.02	0.23	39.75	65.85
Xception	0.44	0.33	0.43	0.44	71.15	88.75

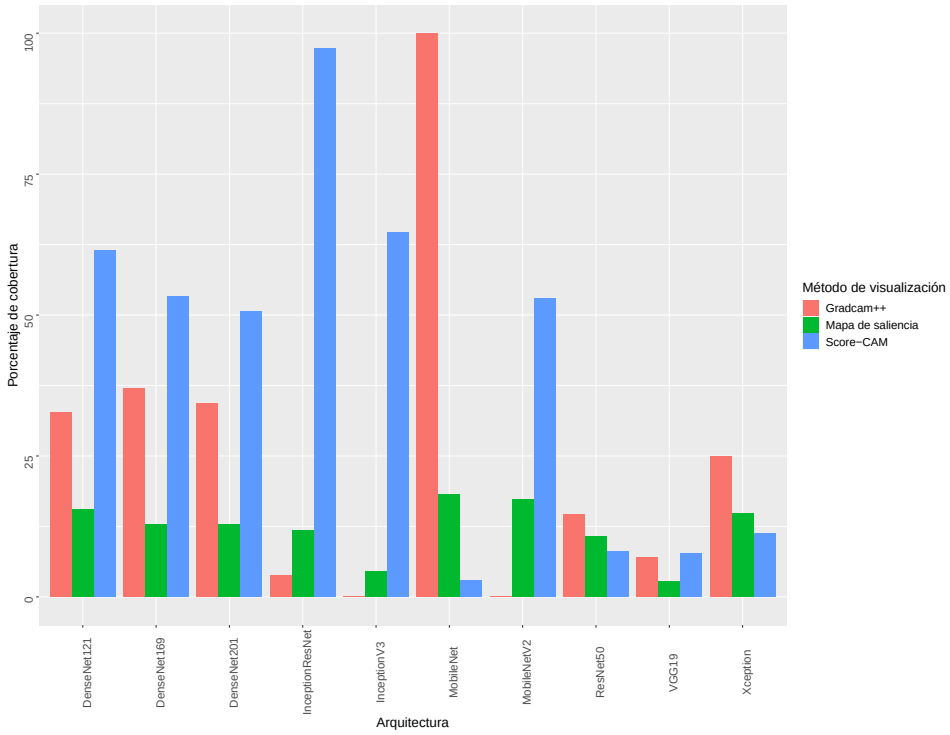


Figura 4.12: Resultados índice en posición superior plantas porcentaje de cobertura.

Tabla 4.16: Resultados índice con diferentes posiciones cobertura Score-CAM plantas.

Arquitectura	**% Superior**	**% Inferior**	**% Izquierda**	**% Derecha**	**% Top-1**	**% Top-5**
DenseNet121	61.51	64.64	60.09	60.68	55.05	81.05
DenseNet169	53.32	59.35	53.79	55.65	52.45	80.50
DenseNet201	50.59	55.73	51.92	51.28	53.10	79.25
InceptionResNet	97.27	97.46	97.84	98.09	67.35	87.75
InceptionV3	64.74	58.41	62.85	63.36	59.70	83.45
MobileNetV1	3.00	2.00	1.00	4.00	59.80	85.50
MobileNetV2	52.87	50.35	49.44	50.52	59.80	85.50
ResNet50	8.10	9.72	8.04	9.16	54.60	80.85
VGG19	7.65	7.75	8.41	7.73	39.75	65.85
Xception	11.30	10.98	11.51	11.63	71.15	88.75

Tabla 4.17: Resultados índice con diferentes posiciones correlación Score-CAM plantas.

Arquitectura	**Superior**	**Inferior**	**Izquierda**	**Derecha**	**% Top-1**	**% Top-5**
DenseNet121	0.28	0.18	0.32	0.29	55.05	81.05
DenseNet169	0.05	0.05	0.19	0.14	52.45	80.50
DenseNet201	0.18	0.07	0.06	0.05	53.10	79.25
InceptionResNet	-0.11	-0.15	-0.12	-0.15	67.35	87.75
InceptionV3	0.22	0.22	0.44	0.29	59.70	83.45
MobileNetV1	-0.01	0.11	-0.05	-0.02	59.80	85.50
MobileNetV2	0.09	0.07	0.14	0.09	59.80	85.50
ResNet50	0.07	0.02	0.06	0.01	54.60	80.85
VGG19	0.35	0.02	0.11	0.29	39.75	65.85
Xception	0.07	0.10	0.07	0.09	71.15	88.75

Tabla 4.18: Resultados índice con diferentes posiciones cobertura mapas de saliencia plantas.

Arquitectura	**% Superior**	**% Inferior**	**% Izquierda**	**% Derecha**	**% Top-1**	**% Top-5**
DenseNet121	15.45	14.91	15.60	14.01	55.05	81.05
DenseNet169	12.82	11.63	12.88	10.53	52.45	80.50
DenseNet201	12.89	12.92	13.64	13.59	53.10	79.25
InceptionResNet	11.72	10.65	12.46	10.62	67.35	87.75
InceptionV3	4.57	4.91	4.85	4.81	59.70	83.45
MobileNetV1	18.15	21.25	17.67	17.41	59.80	85.50
MobileNetV2	17.25	17.84	16.34	18.73	59.80	85.50
ResNet50	10.65	9.72	10.54	9.99	54.60	80.85
VGG19	2.82	2.75	2.60	2.71	39.75	65.85
Xception	14.80	13.19	13.11	15.04	71.15	88.75

Tabla 4.19: Resultados índice con diferentes posiciones correlación mapas de saliencia plantas.

Arquitectura	**Superior**	**Inferior**	**Izquierda**	**Derecha**	**% Top-1**	**% Top-5**
DenseNet121	0.10	-0.04	0.04	0.01	55.05	81.05
DenseNet169	-0.04	-0.03	-0.03	0.01	52.45	80.50
DenseNet201	-0.02	0.05	-0.06	-0.09	53.10	79.25
InceptionResNet	-0.11	-0.06	-0.04	-0.09	67.35	87.75
InceptionV3	-0.05	0.02	-0.14	-0.25	59.70	83.45
MobileNetV1	-0.07	-0.13	-0.05	-0.04	59.80	85.50
MobileNetV2	0.22	0.02	0.05	0.05	59.80	85.50
ResNet50	-0.10	-0.09	-0.19	-0.15	54.60	80.85
VGG19	-0.02	0.06	-0.09	0.09	39.75	65.85
Xception	-0.11	-0.12	0.04	-0.13	71.15	88.75

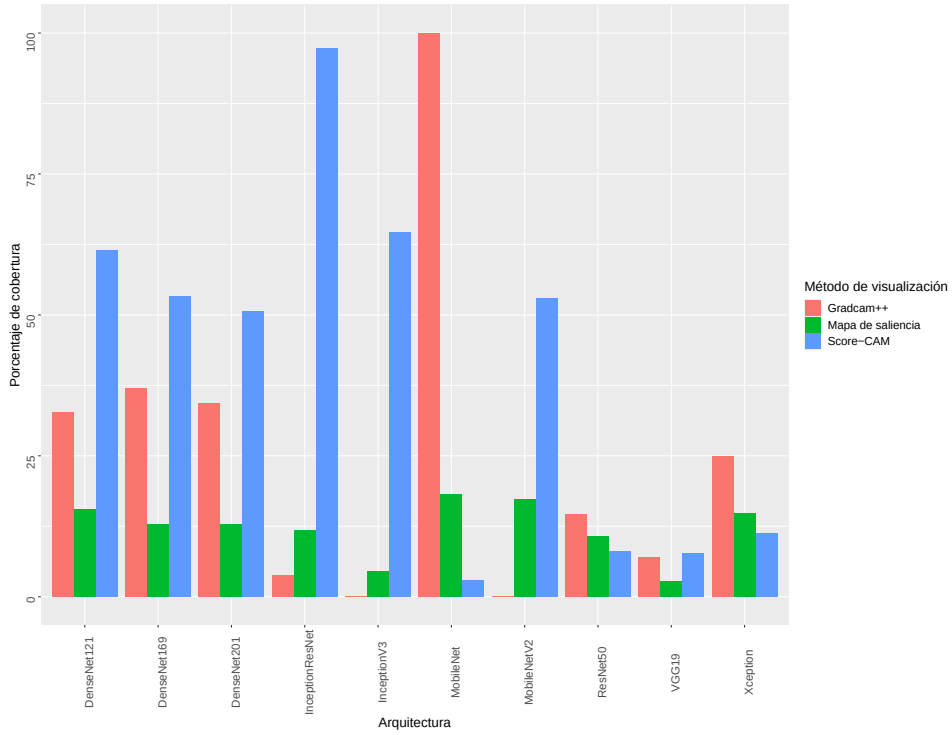


Figura 4.13: Resultados índice en posición inferior plantas porcentaje de cobertura.

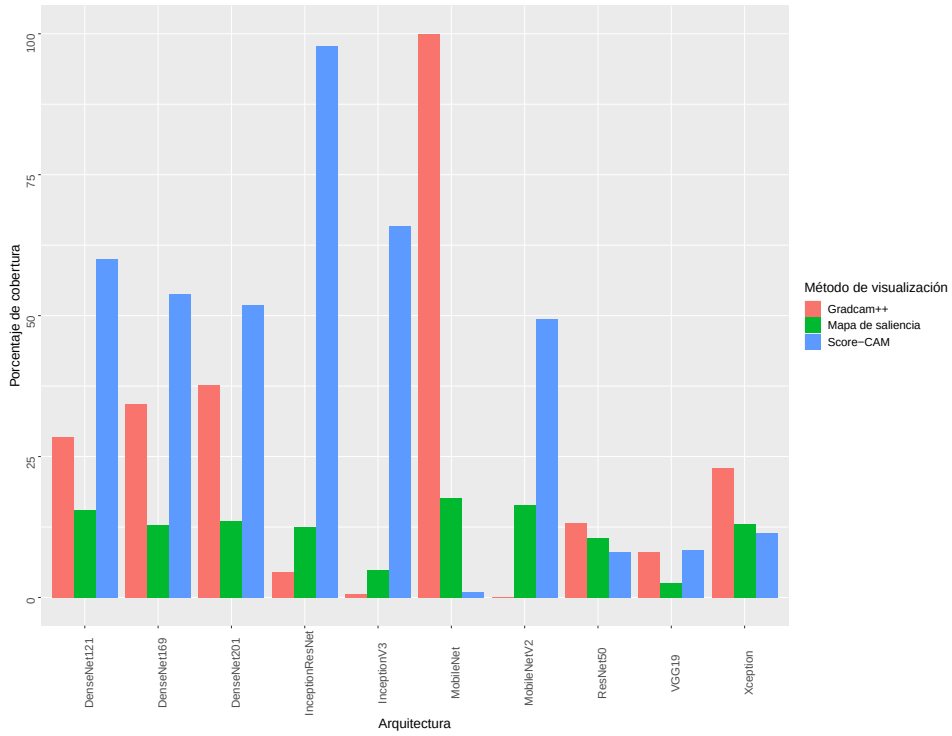


Figura 4.14: Resultados índice en posición izquierda plantas porcentaje de cobertura.

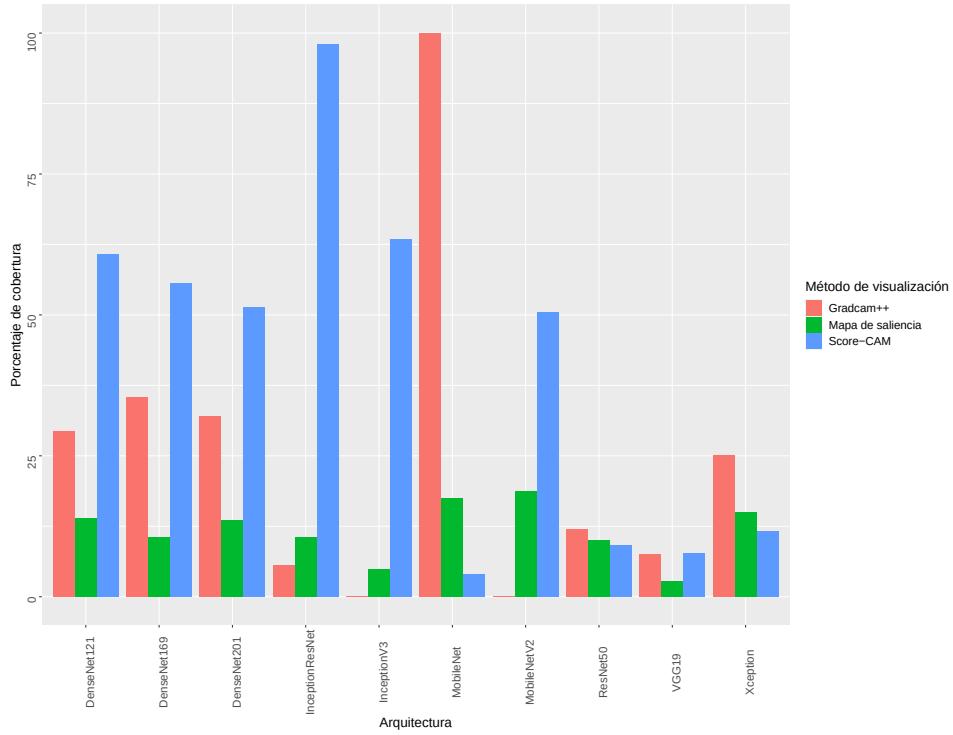


Figura 4.15: Resultados índice en posición derecha plantas porcentaje de cobertura.

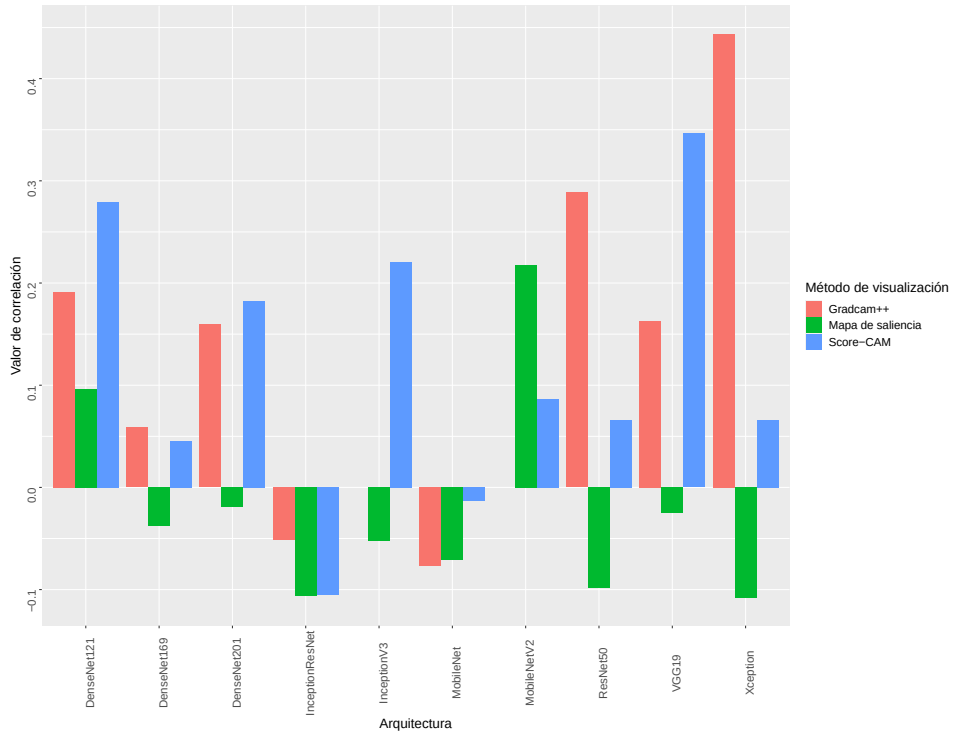


Figura 4.16: Resultados índice en posición superior plantas valor de correlación.

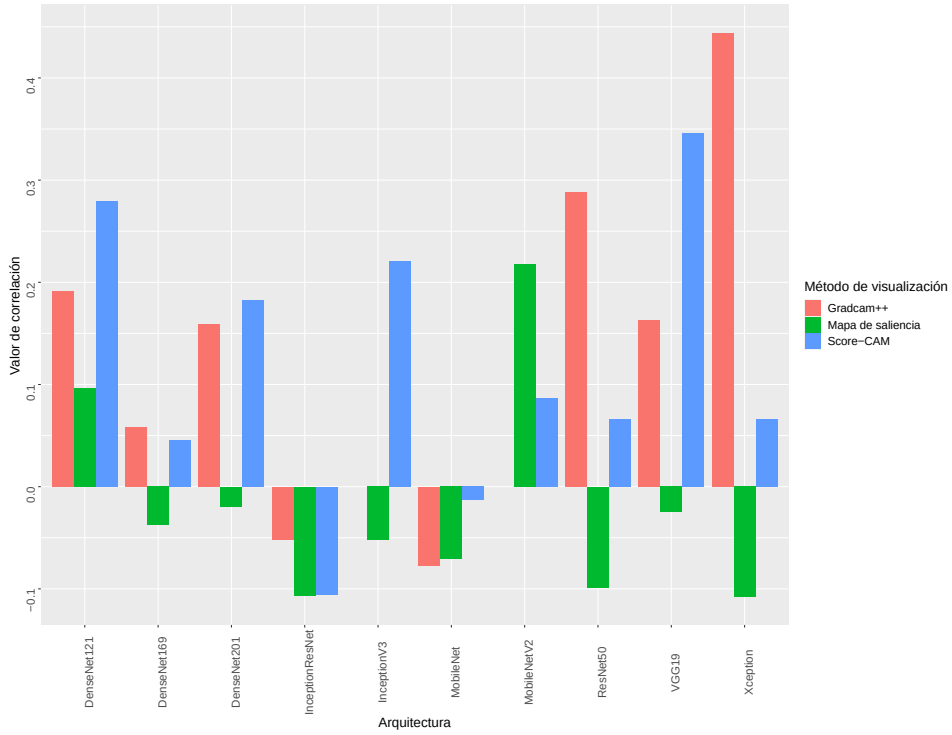


Figura 4.17: Resultados índice en posición inferior plantas valor de correlación.

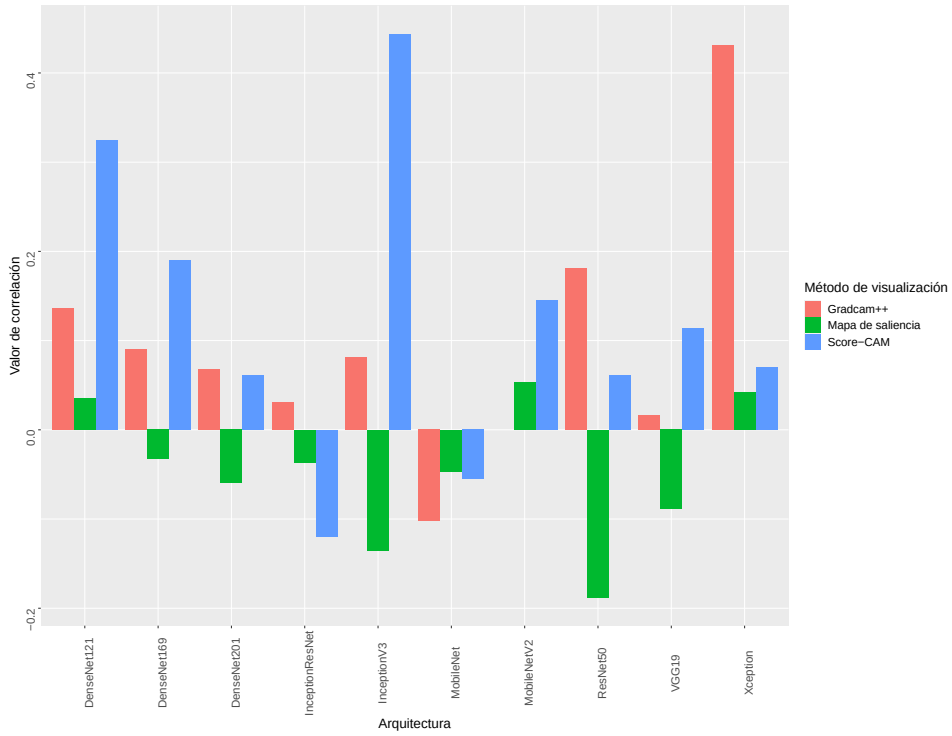


Figura 4.18: Resultados índice en posición izquierda plantas valor de correlación.

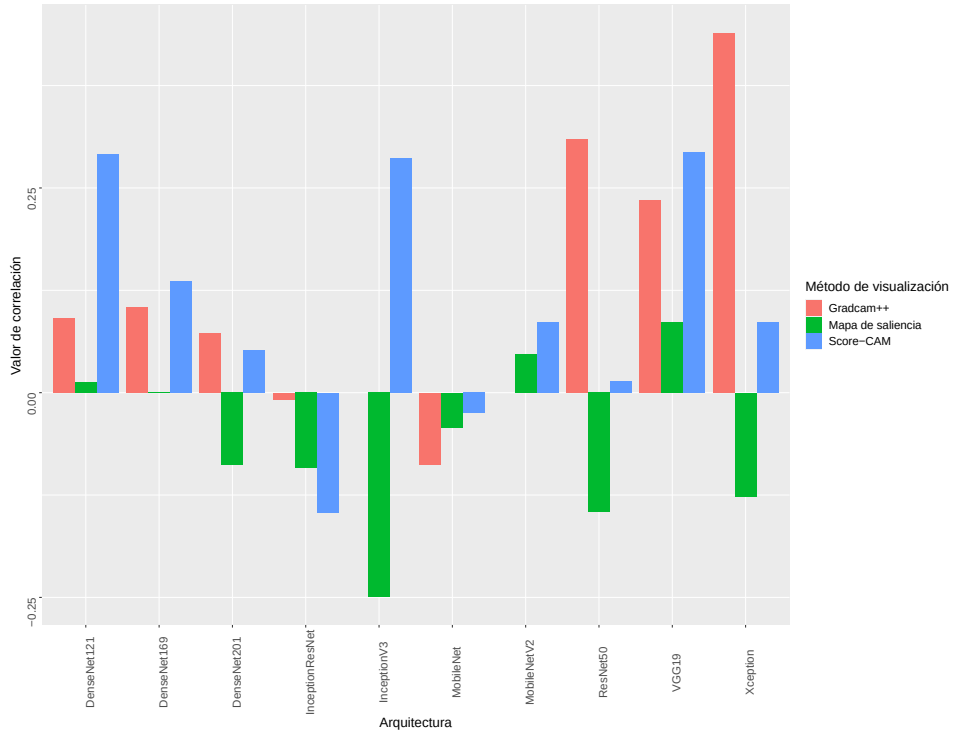


Figura 4.19: Resultados índice en posición derecha plantas valor de correlación.

Conjunto de imágenes de mamografías

El resumen de los resultados por método de visualización se muestra en las Tablas 4.20, 4.21 para Grad-CAM++, las Tablas 4.22, 4.23 para Score-CAM y las Tablas 4.24, 4.25 para mapas de saliencia.

Al igual que en los resultados del índice en condiciones favorables, no fue posible realizar la visualización de las arquitecturas DenseNet121, DenseNet169, DenseNet201, InceptionResNet, InceptionV3 y ResNet50 utilizando Grad-CAM++ y mapas de saliencia debido a problemas con el valor de gradiente. Las arquitecturas VGG19 y Xception fueron las arquitecturas con porcentaje de cobertura mayor al 50 %. En la Tabla 4.20 se muestran el porcentaje de cobertura de Grad-CAM++, donde MobileNetV1, VGG19 y Xception fueron las arquitecturas con algún porcentaje de cobertura. Los valores de correlación de Grad-CAM++ se encuentran en la Tabla 4.21, mostrando valores de correlación negativa principalmente para las arquitecturas MobileNetV1, VGG19 y Xception, debido a los valores de cobertura alto pero con valores de exactitud bajos, generando una correlación negativa. De igual manera, los resultados de cober-

tura y correlación de mapas de saliencia mostrados en las Tablas 4.24 y 4.25 fueron limitados a pocas arquitecturas. VGG19 y Xception obtienen los mejores porcentajes de cobertura superando el 90 % pero obteniendo valores de correlación bajos.

En la Tabla 4.22 se observan los porcentajes de cobertura utilizando Score-CAM. Las arquitecturas VGG19 y Xception se mantienen como las arquitecturas con mayor porcentaje de cobertura. Notamos como la posición de la imagen afecta de manera significativa a las arquitecturas DenseNet121, DenseNet169, DenseNet201, InceptionV3 y ResNet50. La posición izquierda del objeto favorece la cobertura de las arquitecturas DenseNet201 y InceptionV3, la posición inferior del objeto favorece a la arquitectura DenseNet169 y la posición superior del objeto favorece a la ResNet50. Creemos que la posición del objeto influye en arquitecturas que cuenten con variaciones en las conexiones entre capas convolucionales, somo lo son DenseNet y ResNet50. Además, las imágenes de mamografías mantienen una gran similitud en el fondo de la imagen, es decir, zonas de la imagen donde no se encuentre una calcificación, por la posición del objeto cuenta como una característica más para las imágenes de mamografías. En la Tabla 4.23 se observan los valores de correlación que, al igual que los demás experimentos, presentan valores bajos de correlación.

Observando las Figuras 4.20,4.21,4.22 y 4.23 notamos como el comportamiento de las arquitecturas VGG19 y Xception fue constante, cambiando sus valores en menor medida, por lo que la posición de los objetos en las imágenes no es un factor que afecte el proceso de caracterización para estas dos arquitecturas. Por su parte, las arquitecturas DenseNet121, DenseNet169 y DenseNet201 cambiaron de manera notable su porcentaje de cobertura, favoreciendo a DenseNet 121 la posición derecha, a DenseNet169 la posición inferior y a DenseNet201 la posición izquierda.

En las Figuras 4.24, 4.25, 4.26 y 4.27 muestra como existe una correlación negativa en la mayoría. Destaca la correlación de mapas de saliencia con la arquitectura Xception, llegando a alcanzar un valor de 0.72 de correlación con la imagen en la posición inferior utilizando mapas de saliencia. A diferencia de los porcentajes cobertura, los valores de correlación se ven afectados en dependencia a la posición del objeto de interés, teniendo mayores valores de correlación con la imagen en la posición inferior.

Tabla 4.20: Resultados índice con diferentes posiciones cobertura Grad-CAM++ mamografías.

Arquitectura	**% Superior**	**% Inferior**	**% Izquierda**	**% Derecha**	**% Top-1**
DenseNet121	0	0	0	0	49.15
DenseNet169	0	0	0	0	49.43
DenseNet201	0	0	0	0	49.43
InceptionResNet	0	0	0	0	49.15
InceptionV3	0	0	0	0	50.56
MobileNetV1	30.00	30.00	10.00	25.00	50.28
MobileNetV2	0	0	0	0	50.00
ResNet50	0	0	0	0	49.71
VGG19	99.12	99.55	97.52	94.56	49.43
Xception	95.57	94.01	89.94	98.06	50.00

Tabla 4.21: Resultados índice con diferentes posiciones correlación Grad-CAM++ mamografías.

Arquitectura	**Corr Superior**	**Corr Inferior**	**Corr Izquierda**	**Corr Derecha**	**% Top-1**
DenseNet121	NA	NA	NA	NA	49.15
DenseNet169	NA	NA	NA	NA	49.43
DenseNet201	NA	NA	NA	NA	49.43
InceptionResNet	NA	NA	NA	NA	49.15
InceptionV3	NA	NA	NA	NA	50.56
MobileNetV1	-0.03	-0.10	0.36	0.42	50.28
MobileNetV2	NA	NA	NA	NA	50.00
ResNet50	NA	NA	NA	NA	49.71
VGG19	-0.32	0.23	-0.37	0.10	49.43
Xception	-0.07	-0.16	-0.03	-0.19	50.00

Tabla 4.22: Resultados índice con diferentes posiciones cobertura Score-CAM mamografías.

Arquitectura	**% Superior**	**% Inferior**	**% Izquierda**	**% Derecha**	**% Top-1**
DenseNet121	0.56	22.28	0.81	31.01	49.15
DenseNet169	44.96	99.23	46.63	99.10	49.43
DenseNet201	33.63	25.67	92.37	9.19	49.43
InceptionResNet	0	0	0	0	49.15
InceptionV3	12.28	0	16.81	0	50.56
MobileNetV1	20.00	20.00	10.00	10.00	50.28
MobileNetV2	0.08	0.42	0	0	50.00
ResNet50	20.28	0	0.47	0.29	49.71
VGG19	99.02	99.52	97.34	94.06	49.43
Xception	95.71	94.41	91.08	97.98	50.00

Tabla 4.23: Resultados índice con diferentes posiciones correlación Score-CAM mamografías.

Arquitectura	**Corr Superior**	**Corr Inferior**	**Corr Izquierda**	**Corr Derecha**	**% Top-1**
DenseNet121	0.08	-0.10	0.08	0.05	49.15
DenseNet169	0.26	-0.01	-0.27	-0.07	49.43
DenseNet201	0.21	0.19	-0.10	0.10	49.43
InceptionResNet	NA	NA	NA	NA	49.15
InceptionV3	0.09	NA	0.10	NA	50.56
MobileNetV1	-0.30	-0.30	0.36	0.13	50.28
MobileNetV2	0.07	-0.54	NA	NA	50.00
ResNet50	-0.49	NA	0.05	0.05	49.71
VGG19	-0.30	0.23	-0.37	0.10	49.43
Xception	-0.08	-0.17	-0.15	-0.22	50.00

Tabla 4.24: Resultados índice con diferentes posiciones cobertura mapas de saliencia mamografías.

Arquitectura	**% Superior**	**% Inferior**	**% Izquierda**	**% Derecha**	**% Top-1**
DenseNet121	0	0	0	0	49.15
DenseNet169	0	0	0	0	49.43
DenseNet201	0	0	0	0	49.43
InceptionResNet	0	0	0	0	49.15
InceptionV3	0	2.04	2.71	0	50.56
MobileNetV1	0.13	0.26	0.32	0.17	50.28
MobileNetV2	0.31	0.39	0.19	0.06	50.00
ResNet50	0	0	0	0	49.71
VGG19	92.87	93.54	81.40	86.38	49.43
Xception	97.29	94.62	95.45	90.23	50.00

Tabla 4.25: Resultados índice con diferentes posiciones correlación mapas de saliencia mamografías.

Arquitectura	**Corr Superior**	**Corr Inferior**	**Corr Izquierda**	**Corr Derecha**	**% Top-1**
DenseNet121	NA	NA	NA	NA	49.15
DenseNet169	NA	NA	NA	NA	49.43
DenseNet201	NA	NA	NA	NA	49.43
InceptionResNet	NA	NA	NA	NA	49.15
InceptionV3	NA	0.05	-0.40	NA	50.56
MobileNetV1	-0.19	-0.18	-0.20	0.10	50.28
MobileNetV2	0.14	-0.70	-0.03	0.02	50.00
ResNet50	NA	NA	NA	NA	49.71
VGG19	-0.35	-0.14	-0.01	-0.42	49.43
Xception	0.63	0.72	0.54	0.49	50.00

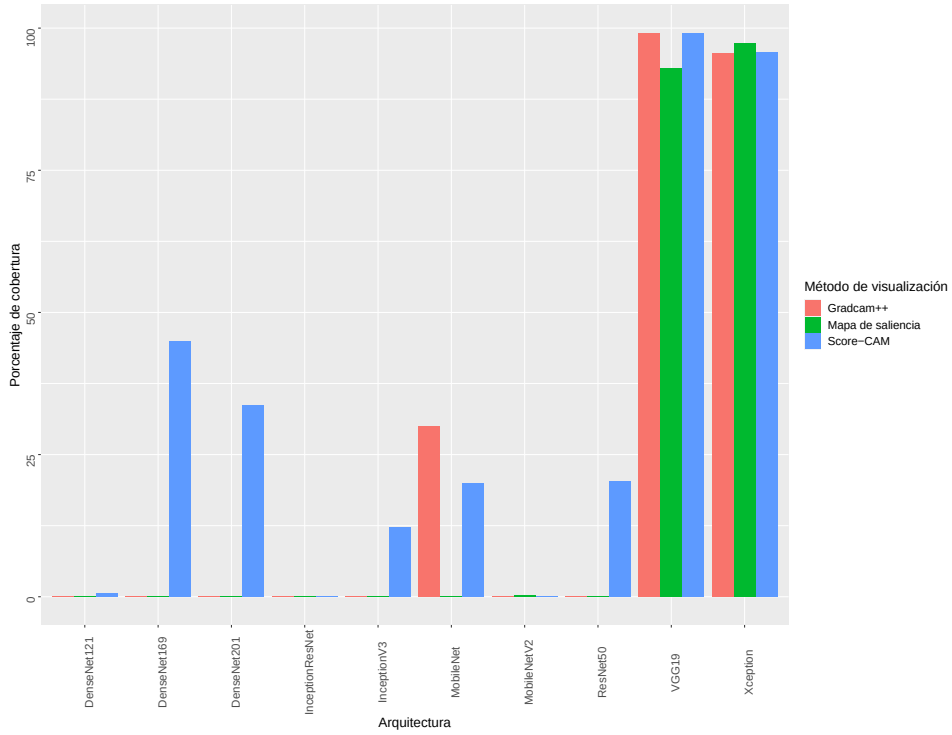


Figura 4.20: Resultados índice en posición superior mamografías porcentaje de cobertura.

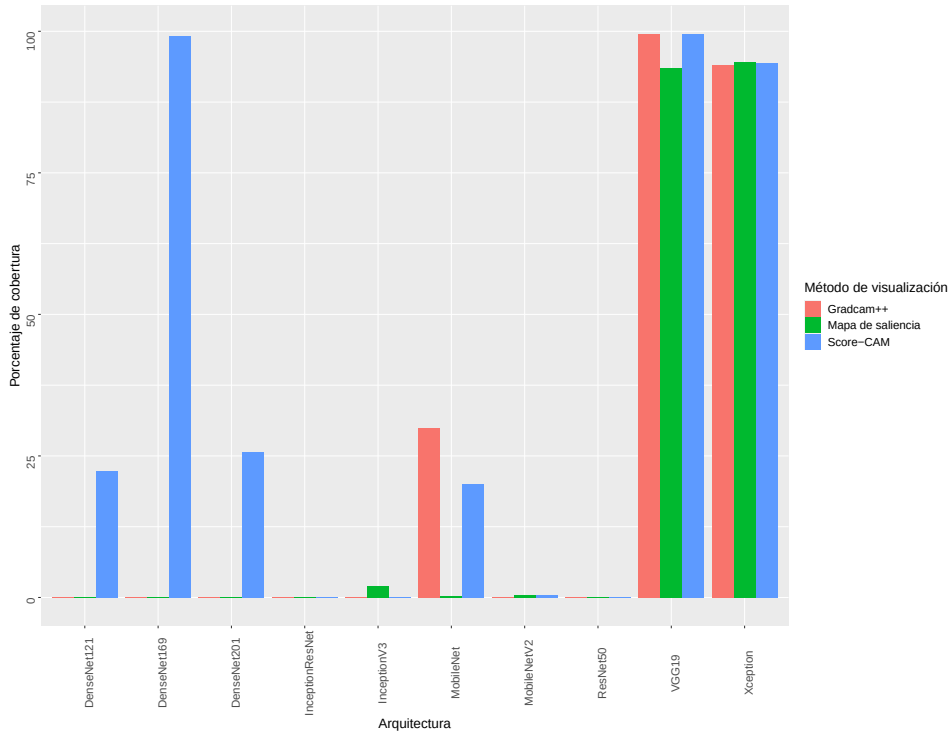


Figura 4.21: Resultados índice en posición inferior mamografías porcentaje de cobertura.

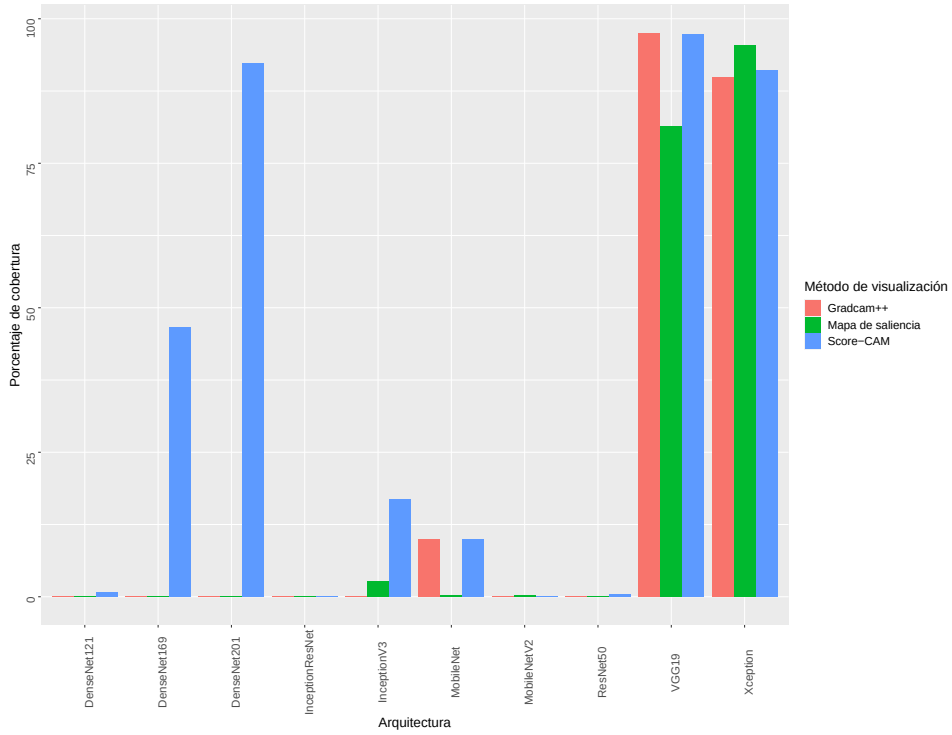


Figura 4.22: Resultados índice en posición izquierda mamografías porcentaje de cobertura.

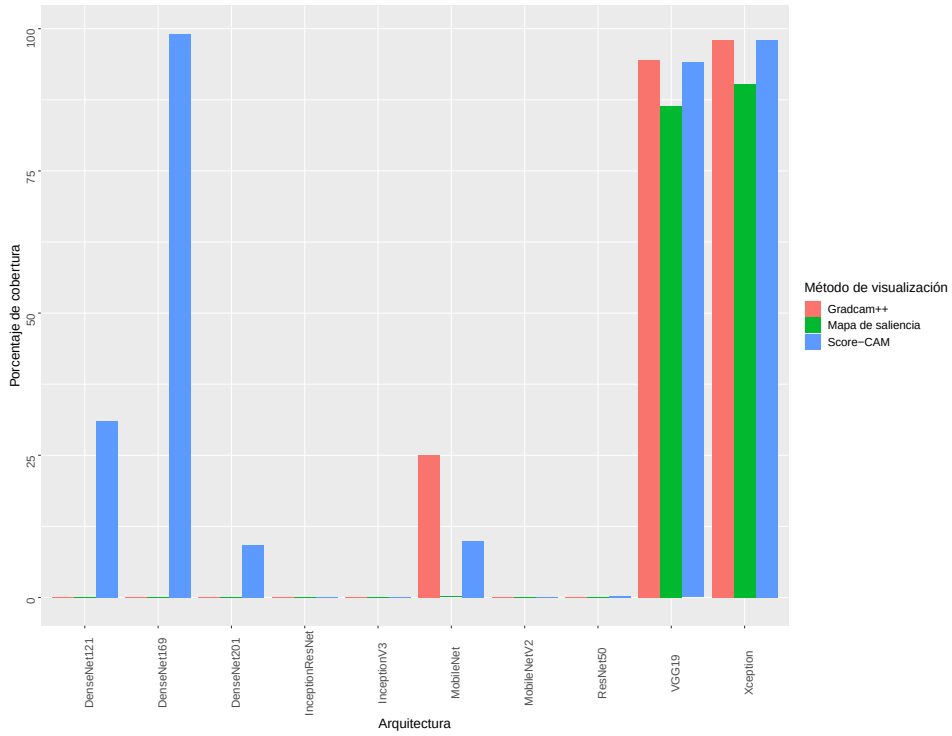


Figura 4.23: Resultados índice en posición derecha mamografías porcentaje de cobertura.

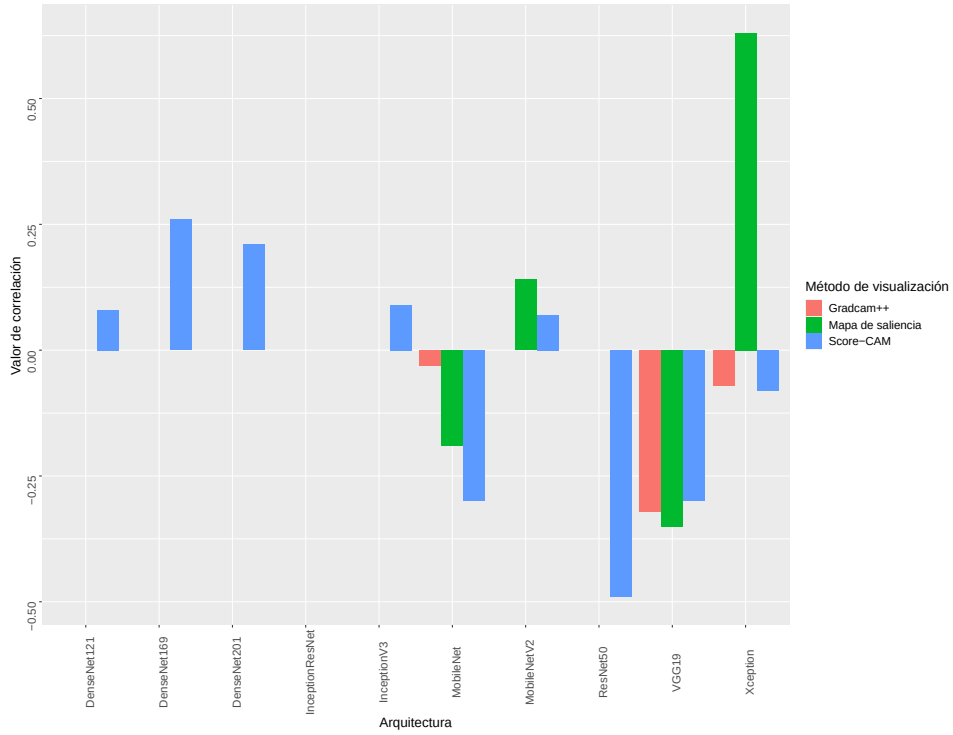


Figura 4.24: Resultados índice en posición superior mamografías valor de correlación.

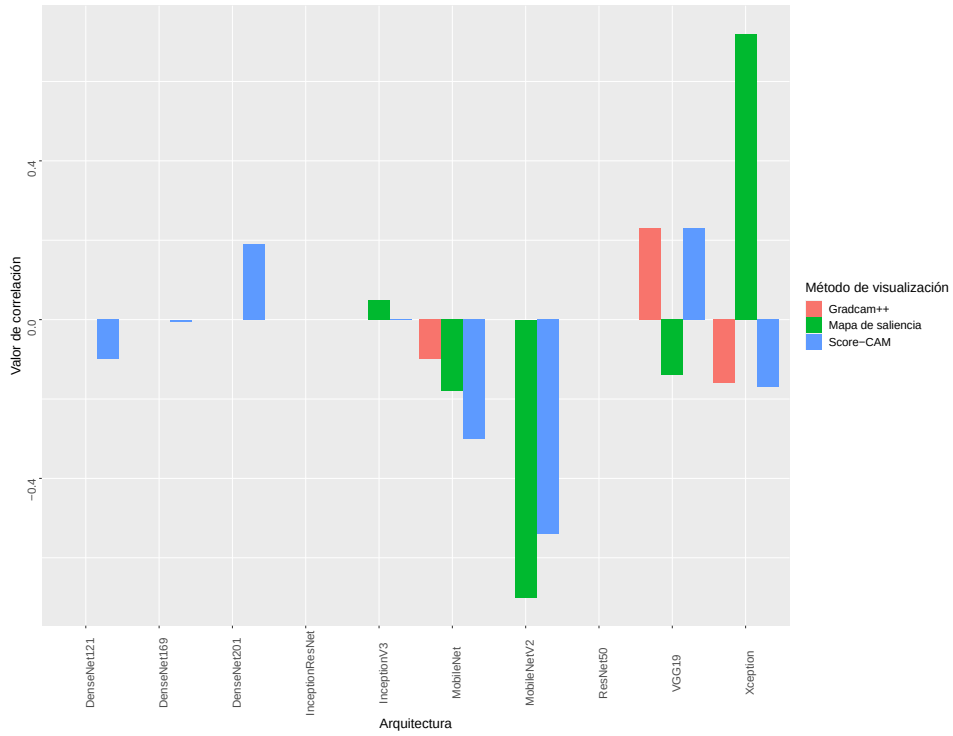


Figura 4.25: Resultados índice en posición inferior mamografías valor de correlación.

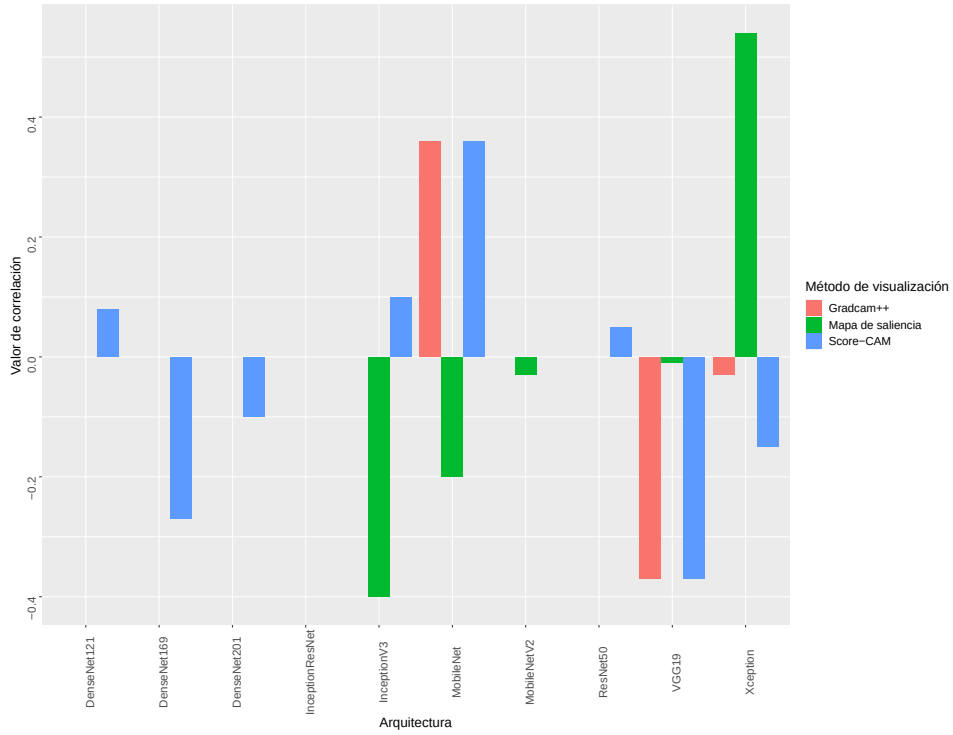


Figura 4.26: Resultados índice en posición izquierda mamografías valor de correlación.

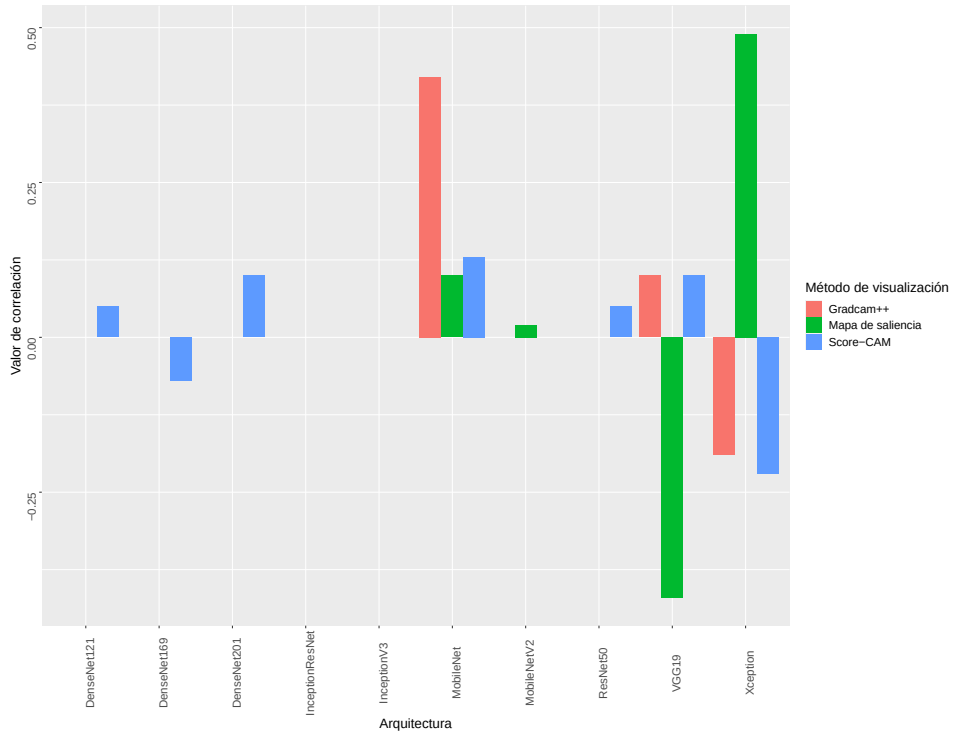


Figura 4.27: Resultados índice en posición derecha mamografías valor de correlación.

Tabla 4.26: Muestra de resultados índice con múltiples objetos, presentando porcentajes de cobertura por encima del 90 %.

Método	**Imagen**	**Resultado visual**	**% Cobertura**
Grad-CAM++			100
Score-CAM			100
Mapas de saliencia			98.86

4.3.3. Resultados índice con múltiples objetos

Se reportan los resultados obtenidos de calcular el porcentaje de cobertura al objeto de interés en imágenes con múltiples objetos contenidos en ella. Se realiza este experimento con el propósito de evaluar si la caracterización de los modelos se ve afectada con la inclusión de objetos accesorios en las imágenes. En la Tabla 4.26 podemos observar un de los resultados obtenidos del índice de evaluación con múltiples objetos, en imágenes donde el objeto de interés no se encuentra aislado, por lo que existe ruido. Este ejemplo presenta porcentajes de cobertura por encima del 90 %, lo que nos indica que el modelo sigue caracterizando enfocándose en el objeto de interés, sin importar objetos accesorios. Al igual que en los resultados de los índices anteriores, se muestra el método de visualización, la imagen, el resultado visual y el porcentaje de cobertura.

Conjunto de imágenes de plantas

El resumen de los resultados por método de visualización se muestra en la Tabla 4.30 para Grad-CAM++, la Tabla 4.31 para Score-CAM y la Tabla 4.32 para mapas de saliencia, contando estas tablas con el porcentaje de cobertura promedio y el valor de correlación.

Tabla 4.27: Resultados índice con múltiples objetos Grad-CAM++ plantas.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**	**% Top-5**
DenseNet121	25.60	0.16	55.05	81.05
DenseNet169	26.30	0.11	52.45	80.50
DenseNet201	28.49	-0.05	53.10	79.25
InceptionResNet	2.30	-0.23	67.35	87.75
InceptionV3	0	NA	59.70	83.45
MobileNetV1	99.99	-0.06	59.80	85.50
MobileNetV2	0	NA	59.80	85.50
ResNet50	6.47	-0.07	54.60	80.85
VGG19	5.27	-0.12	39.75	65.85
Xception	24.87	0.25	71.15	88.75

Comparando los resultados de este índice con respecto a los resultados del índice en condiciones favorables, observamos como los porcentajes de cobertura son idénticos para ambos índices, con lo que deducimos que el proceso de caracterización de las arquitecturas para el conjunto de imágenes de plantas no se ve afectado por objetos accesorios contenidos en las imágenes. Podemos atribuir esto a que los objetos en el conjunto de imágenes de plantas cuentan con características lo suficientemente distintivas como para mantener el enfoque en los objetos de interés, sin importar la presencia de otros objetos.

Al ver la Figura 4.28 observamos como Score-CAM se mantuvo como el método de visualización con mayor porcentaje de cobertura en promedio entre las diferentes arquitecturas, atribuyendo esto al realizar la visualización sin utilizar valores de gradiente.

Notamos en la Figura 4.29 como se mantiene la correlación en valores bajos, sin superar el 0.5, por lo que deducimos que no existe una correlación fuerte entre el porcentaje de cobertura con respecto a la exactitud del modelo.

Tabla 4.28: Resultados índice con múltiples objetos Score-CAM plantas.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**	**% Top-5**
DenseNet121	60.28	0.19	55.05	81.05
DenseNet169	50.24	0.13	52.45	80.50
DenseNet201	48.12	-0.04	53.10	79.25
InceptionResNet	96.25	-0.14	67.35	87.75
InceptionV3	68.59	0.47	59.70	83.45
MobileNetV1	10.00	0.18	59.80	85.50
MobileNetV2	48.42	0.11	59.80	85.50
ResNet50	7.07	0.04	54.60	80.85
VGG19	5.77	0.05	39.75	65.85
Xception	14.50	0.08	71.15	88.75

Tabla 4.29: Resultados índice con múltiples objetos mapa de saliencia plantas.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**	**% Top-5**
DenseNet121	15.12	-0.09	55.05	81.05
DenseNet169	12.70	-0.17	52.45	80.50
DenseNet201	12.18	-0.14	53.10	79.25
InceptionResNet	8.88	-0.21	67.35	87.75
InceptionV3	3.90	-0.09	59.70	83.45
MobileNetV1	16.19	-0.01	59.80	85.50
MobileNetV2	12.77	0.07	59.80	85.50
ResNet50	7.82	-0.20	59.80	85.50
VGG19	2.07	0.05	39.75	65.85
Xception	10.44	0.05	71.15	88.75

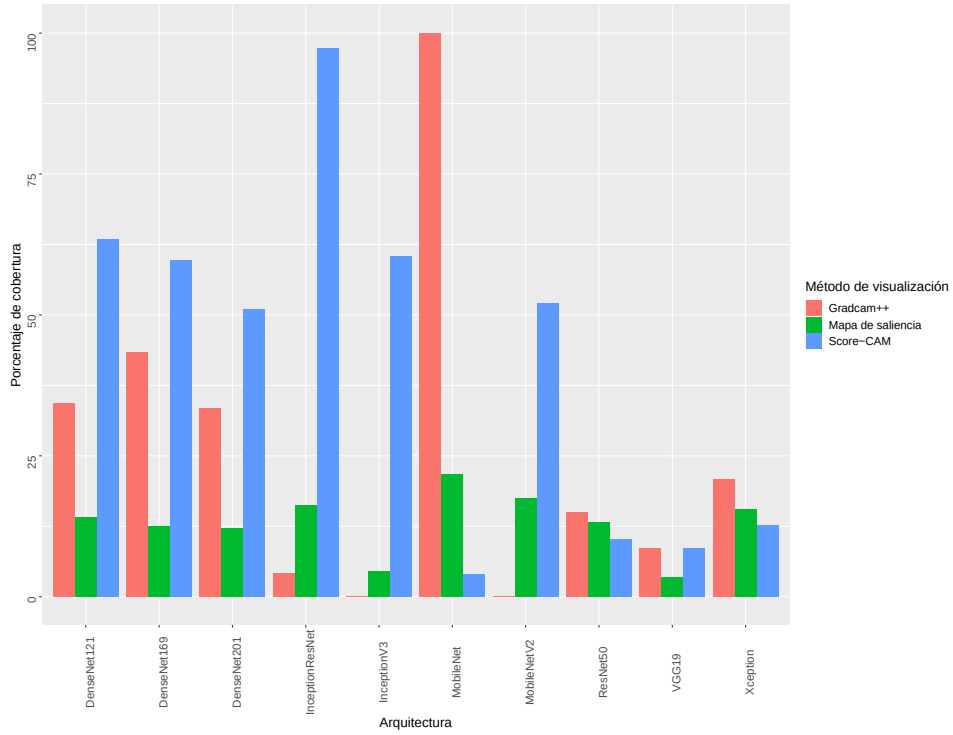


Figura 4.28: Resultados índice con múltiples objetos plantas porcentaje de cobertura.

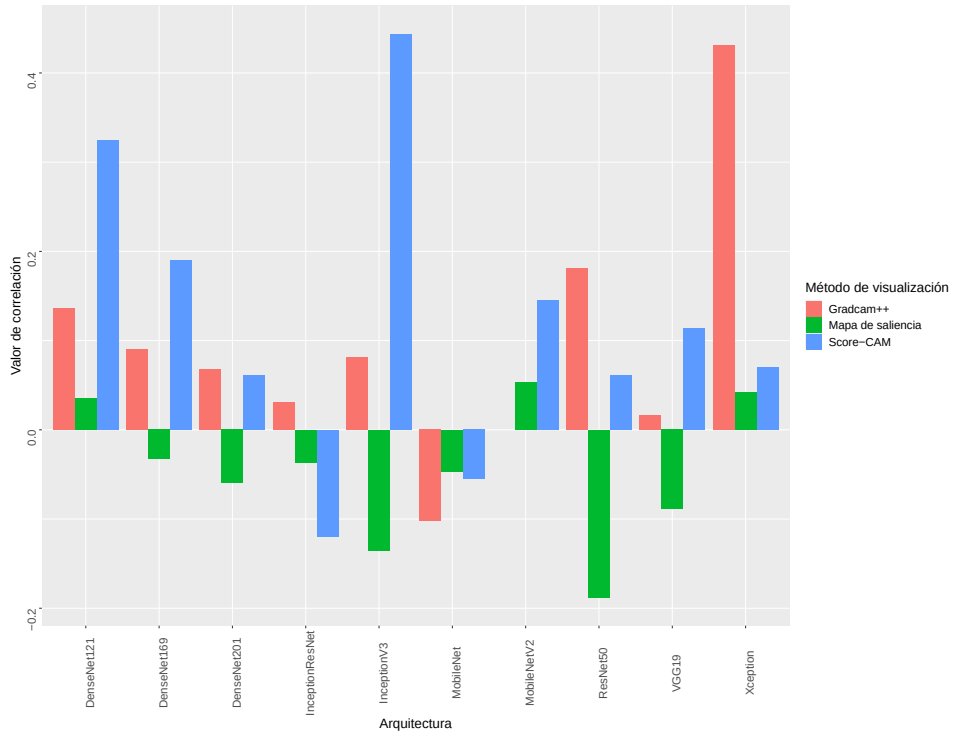


Figura 4.29: Resultados índice con múltiples objetos plantas valor de correlación.

Al realizar los experimentos utilizando los tres índices de evaluación, llegamos a la conclusión de que no existe una correlación fuerte entre la exactitud del modelo con respecto al porcentaje de cobertura abarcado para los modelos de clasificación entrenados utilizando el conjunto de imágenes de plantas. Esto podemos atribuir al problema de clasificación fina, ya que, al existir casos de baja variabilidad interclase, las características distintivas entre clases son pequeñas, por lo que enfocándose en estas pequeñas características nos permite hacer la distinción entre especies, sin la necesidad de enfocarse en el resto del objeto. Un ejemplo de esto es la arquitectura Xception, que obtuvo una exactitud Top-5 de hasta 88.75 % pero porcentajes de cobertura de hasta 11.63 %, lo que nos demuestra que no se necesitó de abarcar los objetos completamente, siempre que identifique de manera correcta las características distintivas.

Conjunto de imágenes de mamografías

El resumen de los resultados por método de visualización se muestra en la Tabla 4.30 para Grad-CAM++, la Tabla 4.31 para Score-CAM y la Tabla 4.32 para mapas de saliencia, contando estas tablas con el porcentaje de cobertura promedio y el valor de correlación.

A diferencia de los resultados utilizando el conjunto de imágenes de plantas, los resultados del índice con múltiples objetos utilizando el conjunto de imágenes de mamografías cuenta con diferencias respecto a los resultados presentados en el índice en condiciones favorables. En la Tabla 4.30 se presentan los resultados de cobertura utilizando Grad-CAM++. Los problemas de visualización con respecto a las arquitecturas con problemas en el valor de gradiente continuó, pero los porcentajes de cobertura de arquitecturas como MobileNetV1 mejoró a un 40 % con respecto al 5 % reportado en el índice en condiciones favorables. Además, VGG19 subió a un 99.96 % comparado al 99.87 % del primero índice. Sin embargo, Xception bajo su cobertura a un 89.20 %. En cuanto a los valores de correlación se mantienen bajos para las arquitecturas Xception y VGG19, subiendo a un 0.53 para la arquitectura MobileNetV1.

En la Tabla 4.31 observamos como los porcentajes de cobertura mejoran en la mayoría de arquitecturas, a excepción de Xception, aun así manteniéndose como la arquitectura con mayor porcentaje de cobertura promedio. De igual manera, los valores de correlación cambian, pero siguen siendo valores de correlación bajos.

Por último, en la Tabla 4.32 se observa como los porcentajes de cobertura para los mapas de

saliencia con respecto a los resultados en el índice en condiciones favorables han bajado para las arquitecturas MobileNetV2, VGG19 y Xception, pero han subido para las arquitecturas InceptionV3 y MobileNetV1.

En la Figura 4.30 se observa cómo, de nueva cuenta, VGG19 y Xception son las arquitecturas que cuentan con un porcentaje de cobertura mayor al 50 % en los tres métodos de visualización, siendo Score-CAM el método de visualización con los valores más altos de cobertura y el mayor promedio de cobertura entre las arquitecturas.

Observando en la Figura 4.31 concluimos como la inclusión de objetos accesorios si afecta al valor de correlación, ya que Grad-CAM++ y mapas de saliencia solamente tienen valores de correlación en tres arquitecturas, comparado a Score-CAM que tiene valores de correlación en nueve arquitecturas.

En general, al contar con tan pocas características distintivas entre las clases, las arquitecturas mejoran su porcentaje de cobertura de objeto al incluir otros objetos en las imágenes. Contrario a lo que se pensaría que, al incluir más objetos se reduciría la cobertura de los objetos de interés, llega a aumentar debido a que se cuentan con más valores en la imagen que contrasten con los objetos de interés. Esto permite distinguir de mejor manera al objeto de interés con el resto de la imagen.

Una vez concluidos los experimentos utilizando los tres índices de evaluación, llegamos a la conclusión de que, en general, las arquitecturas no son capaces de extraer o clasificar de manera correcta el conjunto de imágenes de mamografías. Esto podemos atribuir a que el conjunto de datos no cuenta con las suficientes observaciones y esto impacta directamente a las arquitecturas con gran número de parámetros. Un ejemplo de esto sería la arquitectura InceptionResNet, que es la arquitectura que utiliza un mayor número de parámetros y sus porcentajes de cobertura promedio fueron 0 % en los tres índices de evaluación. Arquitecturas con menor número de parámetros como VGG19 y Xception obtuvieron porcentajes de cobertura promedio mayor al 90 %. Esto nos indica que, para este conjunto de datos, se obtiene mejores resultados en cuanto a la caracterización utilizando arquitecturas menos robustas.

Tabla 4.30: Resultados índice con múltiples objetos Grad-CAM++ mamografías.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**
DenseNet121	0	NA	49.15
DenseNet169	0	NA	49.43
DenseNet201	0	NA	49.43
InceptionResNet	0	NA	49.15
InceptionV3	0	NA	50.56
MobileNetV1	40.00	0.53	50.28
MobileNetV2	0	NA	50.00
ResNet50	0	NA	49.71
VGG19	99.96	0.08	49.43
Xception	89.20	-0.19	50.00

Tabla 4.31: Resultados índice con múltiples objetos Score-CAM mamografías.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**
DenseNet121	4.48	0.10	49.15
DenseNet169	97.13	0.01	49.43
DenseNet201	51.43	0.25	49.43
InceptionResNet	0	NA	49.15
InceptionV3	18.43	0.10	50.56
MobileNetV1	20.00	-0.10	50.28
MobileNetV2	0.03	-0.19	50.00
ResNet50	2.20	0.05	49.71
VGG19	99.96	0.08	49.43
Xception	89.35	-0.19	50.00

Tabla 4.32: Resultados índice con múltiples objetos mapa de saliencia mamografías.

Arquitectura	**% Cobertura**	**Correlación**	**% Top-1**
DenseNet121	0	NA	49.15
DenseNet169	0	NA	49.43
DenseNet201	0	NA	49.43
InceptionResNet	0	NA	49.15
InceptionV3	0.57	0.05	50.56
MobileNetV1	1.94	0.20	50.28
MobileNetV2	0	NA	50.00
ResNet50	0	NA	49.71
VGG19	93.94	0.12	49.43
Xception	96.97	0.30	50.00

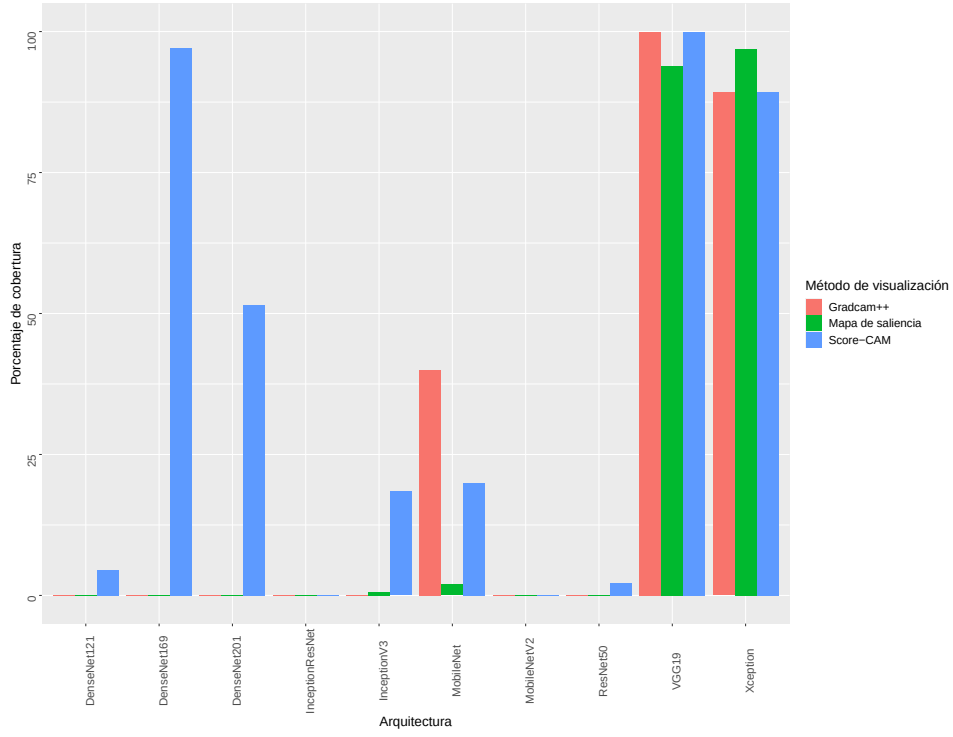


Figura 4.30: Resultados índice con múltiples objetos mamografías porcentaje de cobertura.

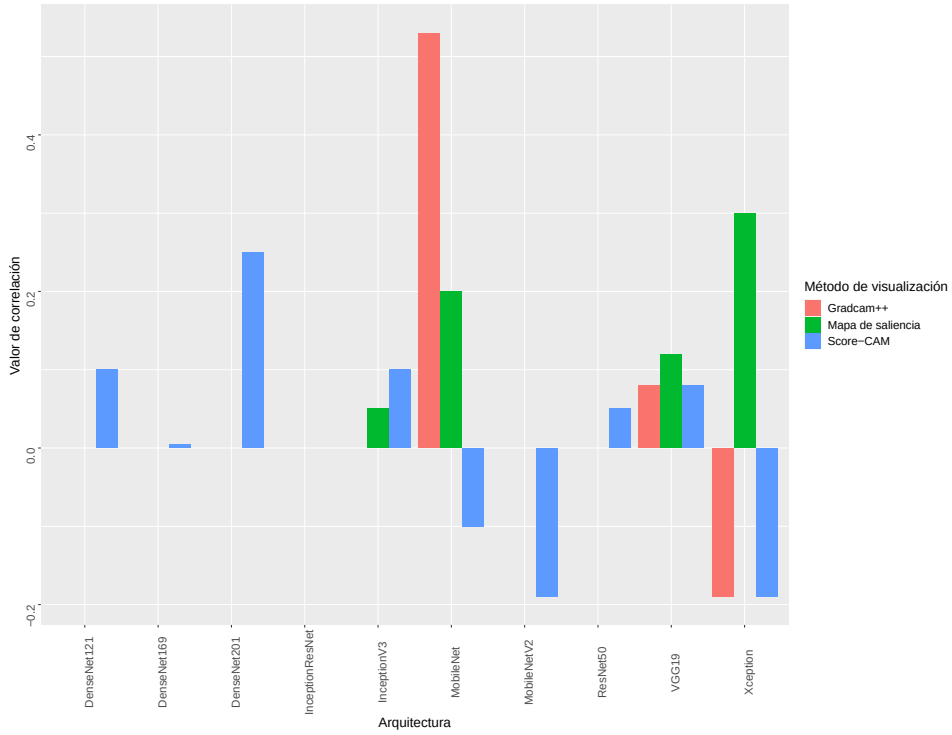


Figura 4.31: Resultados índice con múltiples objetos mamografías valor de correlación.

Capítulo 5

Conclusiones

En este trabajo de tesis se propuso un estudio comparativo entre los métodos de visualización Grad-CAM++, Score-CAM y mapas de saliencia para visualizar el proceso de caracterización de los modelos de clasificación entrenados por RNC. Al realizar este estudio, se abordó el problema de falta de información en el proceso de caracterización de los modelos de clasificación. El observar las características distintivas de los objetos encontradas en el proceso de caracterización nos permite conocer el comportamiento del modelo, y así interpretar los resultados en el proceso de clasificación.

Como casos de estudio, se presentaron el problema de clasificación de imágenes de plantas y el problema de clasificación de imágenes de mamografías. Estos problemas fueron de clasificación fina al contar con baja variabilidad inter clase, es decir, pocas diferencias visuales entre clases, lo que dificulta su clasificación.

Como principal aportación de este trabajo de tesis, se encuentra la elaboración de tres índices de evaluación para el proceso de caracterización de los modelos de clasificación, usando como unidad de medida el porcentaje de cobertura de los objetos de interés. Se propuso esta unidad de medida dado que si el modelo de clasificación no se enfocó en los objetos de interés durante el proceso de caracterización, la clasificación no se realizará utilizando las características distintivas de los objetos.

Una vez entrenados, los modelos de clasificación obtuvieron precisiones de hasta un 88.75 % en Top-5 (Xception) para el conjunto de imágenes de plantas y hasta un 50.56 % de precisión Top-1 (InceptionV3) para el conjunto de imágenes de mamografías.

Aplicando el índice de evaluación con los objetos en condiciones favorables, se observó a MobileNet y a InceptionResNet con el mayor porcentaje de cobertura con el conjunto de imágenes de planta, llegando a obtener un 96.25 % de cobertura utilizando Score-CAM y un 99.99 % de cobertura utilizando Grad-CAM++. En el caso del conjunto de imágenes de mamografías, las arquitecturas VGG19 y Xception fueron quienes obtuvieron un mejor porcentaje de cobertura, superando el 90 % de cobertura utilizando Score-CAM.

Al aplicar el índice de evaluación con los objetos en diferentes posiciones, en donde se trasladaba el objeto de interés a diferentes posiciones de la imagen, se observó que para el conjunto de imágenes de plantas la posición no afecta al proceso de caracterización, ya que no hubo alteraciones significativas en los porcentajes de cobertura. Por su parte, la posición del objeto de interés llegó a afectar al porcentaje de cobertura en los modelos de clasificación entrenados por DenseNet para el conjunto de imágenes de mamografías, lo que hace a la posición del objeto un factor de importancia para el proceso de caracterización de DenseNet utilizando el conjunto de imágenes de mamografías.

Aplicando el índice de evaluación con múltiples objetos, que analiza el comportamiento de los modelos al incluir objetos accesorios al objeto de interés contenido en la imagen, se observa que tanto para el conjunto de plantas como para el conjunto de mamografías, el porcentaje de cobertura no cambió significativamente, manteniendo la cobertura en los objetos de interés. Esto nos indica que las arquitecturas mantienen su enfoque en los objetos de interés.

Tomando en cuenta los porcentajes de cobertura de los objetos para los métodos de visualización, se realizó el cálculo de correlación entre el porcentaje de cobertura con respecto a la precisión de los modelos en la clasificación. Se observa que no existe una fuerte correlación entre el porcentaje de cobertura con respecto a la precisión en la clasificación. El conjunto de imágenes de mamografías llegó a obtener un porcentaje de cobertura de hasta 99 % pero con una exactitud de clasificación que no superaba el 60 %. Concluimos que las arquitecturas no eran capaces de diferenciar las características distintivas entre las dos clases. En el caso del conjunto de imágenes de plantas se llegó a la conclusión que, al ser un problema de clasificación fina, las características distintivas de los objetos podían ser sutiles, por lo que en ocasiones no era necesario enfocarse por completo en el objeto si no solo en estás características sutiles. Tomando como ejemplo a la arquitectura Xception que obtuvo un 88.75 % de exactitud en la clasificación

pero que obtenía hasta un 14.5 % de cobertura utilizando Score-CAM en el primer índice de evaluación, enfocándose entonces en las características distintivas en lugar de todo el objeto de interés.

Como trabajo a futuro, decidimos explorar otras unidades de medición para evaluar el proceso de caracterización de los modelos de clasificación. Esto debido a que el porcentaje de cobertura es un primer indicador que nos permite conocer si se enfocó o no en los objetos de interés, pero el enfocar por completo el objeto no significa que caracteriza de manera adecuada, como fue el caso del conjunto de imágenes de mamografías.

Bibliografía

[1] Naturalista. `https://www.naturalista.mx/`.

[2] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity Checks for Saliency Maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 9525–9536, Montréal, Canada, Dec. 2018.

[3] S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a Convolutional Neural Network. In *Proceedings of the 2017 International Conference on Engineering and Technology*, pages 1–6, Antalya, Turkey, Aug. 2017.

[4] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. A Unified View of Gradient-Based Attribution Methods for Deep Neural Networks. In *Proceedings of the NIPS Workshop on Interpreting, Explaining and Visualizing Deep Learning - Now What?*, California, USA, Dec. 2017.

[5] M. R. Anderson and M. Cafarella. Input Selection for Fast Feature Engineering. In *Proceedings of the 32nd International Conference on Data Engineering*, pages 577–588, Helsinki, Finland, May 2016.

[6] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7):1–46, 07 2015.

[7] R. Baker, K. Rogers, N. Shepherd, and N. Stone. New Relationships Between Breast Microcalcifications and Cancer. *British journal of cancer*, 103:1034–9, 09 2010.

[8] Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35:1798–1828, 08 2013.

[9] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision*, pages 839–847, Nevada, USA, 2018.

[10] F. Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1800–1807, Hawaii, USA, 2017.

[11] B. Dębska and B. Guzowska-Świder. Application of Artificial Neural Network in Food Classification. *Analytica Chimica Acta*, 705(1):283 – 291, 2011. A selection of papers presented at the 12th International Conference on Chemometrics in Analytical Chemistry.

[12] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing Higher-Layer Features of a Deep Network. *Technical Report, Univeristé de Montréal*, 01 2009.

[13] R. Fernández Blanco. *Deep learning para la generación de imágenes histopatológicas realistas mediante aritmética de vectores conceptuales*. Master's thesis, Universitat Oberta de Catalunya, 2019.

[14] R. Fong and A. Vedaldi. *Explanations for Attributing Deep Neural Network Predictions*, pages 149–167. Springer International Publishing, Cham, 2019.

[15] K. Fukushima. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, 36:193–202, 1980.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Nevada, USA, 2016.

[17] J. Heaton. *Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks*. Artificial Intelligence for Humans. Createspace Independent Publishing Platform, 2015.

[18] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 04 2017.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2261–2269, Hawaii,USA, 2017.

[20] F. Informatik, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. *A Field Guide to Dynamical Recurrent Neural Networks*, 03 2003.

[21] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the Best Multi-Stage Architecture for Object Recognition? In *Proceedings of the 12th International Conference on Computer Vision*, pages 2146–2153, Kyoto, Japan, sept 2009.

[22] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artificial Intelligence Review*, pages 1 – 62, 2020.

[23] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 12 2014.

[24] S. Kumar, A. Viinikainen, and T. Hamalainen. Machine Learning Classification Model for Network Based Intrusion Detection System. In *Proceedings of the 11th International Conference for Internet Technology and Secured Transactions*, pages 242–249, Barcelona, Spain, Dec. 2016.

[25] Y. LeCun, Y. Bengio, and G. Hinton. Deep Learning. *Nature*, 521:436–44, 05 2015.

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[27] R. Lee, F. Gimenez, A. Hoogi, K. Miyake, M. Gorovoy, and D. Rubin. A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific Data*, 4:170177, 12 2017.

[28] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino. How Deep Learning Extracts and Learns Leaf Features for Plant Classification. *Pattern Recognition*, 71:1 – 13, 2017.

[29] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen. Medical Image Classification with Convolutional Neural Network. In *Proceedings of the 13th International Conference on Control Automation Robotics Vision*, pages 844–848, Singapore, Singapore, Dec. 2014.

[30] W. S. McCulloch and W. Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity. pages 115–133, 1943.

[31] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. 09 2019.

[32] R. N. Nagashree, N. Aswini, A. Dyana, and C. H. S. Rao. Detection and Classification of Ground Penetrating Radar Image Using Textrual Features. In *Proceedings of the 2014 International Conference on Advances in Electronics Computers and Communications*, pages 1–5, Bangalore, India, Oct. 2014.

[33] Y. Peng, X. He, and J. Zhao. Object-Part Attention Model for Fine-Grained Image Classification. *IEEE Transactions on Image Processing*, 27(3):1487–1500, 2018.

[34] R. Rad and M. Jamzad. Real Time Classification and Tracking of Multiple Vehicles in Highways. *Pattern Recognition Letters*, 26(10):1597 – 1607, 2005.

[35] W. Rawat and Z. Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29:1–98, 06 2017.

[36] M. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, California, USA, 02 2016.

[37] H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[38] F. Rosenblatt. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review*, pages 65–386, 1958.

[39] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer. Early Detection and Classification of Plant Diseases with Support Vector Machines Based on Hyperspectral Reflectance. *Computers and Electronics in Agriculture*, 74(1):91 – 99, 2010.

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

[41] C. Rye, R. Wise, V. Jurukovski, J. DeSaix, J. Choi, and Y. Avissar. Neurons and Glial Cells. In *Biology*. Oct 21, 2016.

[42] N. Sabri, M. F. Kamarudin, R. Hamzah, N. N. A. Manghsor, K. A. F. A. Samah, and N. Hasan. Combination of Color, Shape and Texture Features for Orchid Classification. In *Proceedings of the 9th International Conference on System Engineering and Technology*, pages 315–319, Malaysia, Malaysia, Oct. 2019.

[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Inverted Residuals and Linear Bottlenecks: Mobile Networks forClassification, Detection and Segmentation. 01 2018.

[44] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *Proceedings of the 2017 IEEE International Conference on Computer Vision*, pages 618–626, Venice, Italy, Oct. 2017.

[45] N. Shah, V. Chainani, P. Delafontaine, A. Abdo, J. Lafferty, and N. Abi Rafeh. Mammographically Detectable Breast Arterial Calcification And Atherosclerosis: A Review. *Cardiology in review*, 22, 04 2013.

[46] M. Shaha and M. Pawar. Transfer Learning for Image Classification. In *Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology*, pages 656–660, Coimbatore, India, 2018.

[47] M. F. Shakeel, N. A. Bajwa, A. M. Anwaar, A. Sohail, A. Khan, and H. ur Rashid. Detecting Driver Drowsiness in Real Time Through Deep Learning Based Object Detection. In I. Rojas, G. Joya, and A. Catala, editors, *Proceedings of the 15th International Work-Conference on Artificial Neural Networks*, Gran Canaria, Spain, June 2019.

[48] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *CoRR*, abs/1312.6034, 2014.

[49] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the International Conference on Learning Representations*, California, USA, May 2015.

[50] Y.-Y. Song and Y. Lu. Decision Tree Methods: Applications for Classification and Prediction. *Shanghai archives of psychiatry*, 27:130–5, 04 2015.

[51] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for Simplicity: The All Convolutional Net. 12 2014.

[52] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4278–4284, California, USA, Feb. 2017. AAAI Press.

[53] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the Computer Vision and Pattern Recognition 2016*, Nevada, USA, 06 2016.

[54] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, Massachusetts, USA, Oct. 2015.

[55] A. Tzotsos and D. Argialas. *Support Vector Machine Classification for Object-Based Image Analysis*, pages 663–677. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[56] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu. Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 111–119, Seattle, USA, 06 2020.

[57] W. Yu, K. Yang, Y. Bai, H. Yao, and Y. Rui. Visualizing and Comparing Convolutional Neural Networks. 12 2014.

[58] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Proceedings of the 2014 European Conference on Computer Vision*, pages 818–833, Zurich, Switzerland, Sept. 2014. Springer International Publishing.

[59] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional Networks. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, California, USA, June 2010.

[60] W. J. Zhang, G. Yang, Y. Lin, C. Ji, and M. M. Gupta. On Definition of Deep Learning. In *Proceedings of the 2018 World Automation Congress*, pages 1–5, Washington, USA, June 2018.

[61] A. Zheng and A. Casari. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, Inc., 1st edition, 2018.

[62] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, Nevada, USA, June 2016.