UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

VISIÓN DE MÁQUINA: EJEMPLOS POSITIVOS MÍNIMOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

JAVIER CARRASCO OTÁROLA

PROFESOR GUÍA:
AIDAN HOGAN
PROFESOR GUÍA 2:
JORGE PÉREZ ROJAS

MIEMBROS DE LA COMISIÓN:
PABLO BARCELÓ BAEZA
JUAN MANUEL BARRIOS NUÑEZ

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN
POR: JAVIER CARRASCO OTÁROLA
FECHA: 2019
PROF. GUÍA: AIDAN HOGAN JORGE PÉREZ ROJAS

VISIÓN DE MÁQUINA: EJEMPLOS POSITIVOS MÍNIMOS

Actualmente las redes neuronales están siendo cada vez más importantes en cada aspecto de la vida de las personas, muchas veces complementando y potenciando labores realizadas a diario; otras veces, y cada vez más, toman las labores más repetitivas de la sociedad permitiendo usar el intelecto humano en aquellas cosas donde es realmente necesario.

Uno de los problemas que enfrentan las nuevas tecnologías es determinar cuando las máquinas logran ser lo suficientemente competentes para tomar el relevo; por otro lado se entiende que el desarrollo de mejores redes neuronales conlleva entender, en parte, qué es lo que las hace diferentes de un cerebro biológico y de esta manera poder pensar mecanismos para disminuir la distancia entre ellos.

En esta memoria se presenta, en primera instancia, una forma de caracterizar las redes neuronales por medio de la generación de imágenes positivas mínimas, obtenidas a partir de la disminución de parámetros que afectan la confianza para tomar decisiones de clasificación. Como segunda parte se ofrece un estudio comparativo que evalúa la capacidad de reconocer las imágenes mínimas, generadas en la primera parte, de un grupo de personas para entender las diferencias entre ambos.

Con respecto a los resultados obtenidos en la primera parte, se muestra que dependiendo de qué información se va perdiendo las redes pueden ser más o menos sensible, en particular se observa que son muy poco sensibles a disminuciones en la cantidad de colores disponibles, así como también del contexto donde están insertos los elementos a clasificar. Por otra parte disminuir la resolución de las imágenes tiene un efecto inmediato en la capacidad de reconocer.

Contrastando con los modelos de clasificación, los seres humanos son mucho menos sensibles a la pérdida de información, en especial con la pérdida de colores y resolución, pero tienden a depender de múltiples factores para tomar una decisión, en particular detalles contextuales.

A mis padres.

Agradecimientos

Quisiera agradecer en primer lugar a mi familia, por haberme apoyado durante este largo y sinuoso camino, por haberme dado la libertad de poder equivocarme.

Además, quisiera agradecer a los profesores Aidan Hogan y Jorge Pérez, por la dedicación y consejos que me brindaron durante este proceso, sin ellos esta memoria no hubiese sido posible.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

En la actualidad es cada vez más común ver cómo la visión de máquina está tomando un papel más importante en nuestra vida diaria, desde reconocimiento facial como medio de autenticación, hasta vehículos que son cada vez más independientes de un conductor humano. Todo apunta a que en un futuro la dependencia de este tipo de tecnología va a ser mucho mayor, en especial cuando se considera que actualmente se han desarrollado redes neuronales que son capaces de obtener mejores resultados que los obtenidos por seres humanos expertos, particularmente cuando se trata de clasificar imágenes[5].

Si bien no se han creado máquinas que sean capaces de clasificar correctamente un cien por ciento de las entradas, solo basta que estas estén a la par o sean ligeramente mejores que sus contrapartes humanas para volver a estas últimas virtualmente obsoletas para estas tareas, considerando la gran cantidad de recursos que se ahorran, tanto en tiempo como en dinero, al entrenar una red, pues un humano necesita años de entrenamiento y experiencia para poder llegar a niveles similares de competencia; aun así las máquinas tienden a ser mucho más vulnerables que los seres humanos a intentos deliberados de confundir la clasificación mediante ejemplos adversos[3] (inputs que son clasificados incorrectamente por una red neuronal); esto crea la necesidad de desarrollar mecanismos capaces de crear redes muchos más robustas, sobre todo si estas tecnologías serán utilizadas, potencialmente, en contextos donde las vidas de personas puedan estar en peligro debido a reconocimientos fallidos, como es el caso de aplicaciones en transporte o diagnóstico de enfermedades.

Finalmente, hoy en día, a pesar de todos los avances, las redes neuronales más grandes tienden a ser cajas negras y entender la forma en que trabajan es clave para comprender sus limitaciones y potencialidades para así poder desarrollar mejoras de forma más efectiva; este trabajo busca, de cierta manera, ayudar un poco a esclarecer estas cajas negras al estudiar los ejemplos positivos mínimos, que son los casos bordes en que la red logra identificar correctamente la imagen mostrada al modificar ciertos parámetros que disminuyen la información de la imagen original, tales como colores, resolución etc.

Objetivos

Objetivo General

Este trabajo busca desarrollar mecanismos eficientes para encontrar el punto en el que redes neuronales específicas dejan de clasificar adecuadamente las imágenes mostradas para distintas categorías, de esta manera será posible determinar la sensibilidad de las redes a distintas filtros aplicados a las imágenes.

Lo desarrollado en esta memoria también nos ayudará a proponer nuevos benchmarks para analizar la sensibilidad de redes neuronales ya entrenadas de forma fácil.

Adicionalmente, si bien este trabajo se centrará en imágenes, se espera que los mecanismos mostrados sean extrapolables a otros tipos de datos como texto o sonido.

Objetivos Específicos

1. Desarrollo de programas para encontrar el punto borde en que se deja de clasificar una imagen de forma correcta, y, en consecuencia, encontrar el ejemplo positivo mínimo; estos programas deben ser eficientes pues se deberán procesar una gran cantidad de imágenes.

2. Obtener conclusiones con respecto a la sensibilidad de las redes neuronales utilizadas al enfrentarse a imágenes, de distintas categorías, modificadas con diferentes filtros.

3. Establecer pruebas para comparar la sensibilidad de las redes neuronales con respecto a seres humanos.

Solución

El poder obtener las imágenes positivas mínimas depende de la dimensionalidad de los filtros utilizados; para los casos en que solamente hay una variable en juego basta con aumentar la intensidad de dicho filtro para encontrar el punto en que la red neuronal deja de clasificar correctamente, mientras que, por otro lado, cuando se trata de múltiples dimensiones afectadas, se deberá utilizar un método de optimización eficiente; en este caso se utilizará el método de Powell[11].

Para comparar los seres humanos con las redes neuronales se creará un sistema en línea que presentará las diferentes imágenes mínimas generadas en el punto anterior, las cuales deberán ser clasificadas por las personas. Para poder garantizar la validez de los datos se realizará, en una primera instancia, la experiencia con un grupo controlado de personas para establecer criterios de aceptación para los registros del resto de los usuarios.

Capítulo 2

Marco Teórico

En el presente capitulo se expondrán todos aquellos elementos que son necesarios para entender la presente memoria. Se partirá con una discusión sobre imágenes digitales para luego pasar a explicar conceptos sobre redes neuronales y como estas pueden ser utilizadas para clasificar imágenes, así como introducir aquellas redes que serán utilizadas en el presente trabajo. Finalmente se expondrán los trabajos relacionados que cimientan esta memoria.

2.1. Imágenes

En esta sección se presentarán algunos conceptos relevantes a la representación y compresión de imágenes, en particular, para este trabajo, se tratarán las imágenes mediante el sistema RGB y se almacenarán en formato PNG, que permite compresión sin perdida de información.

2.1.1. RGB

El sistema RGB un sistema de representación de colores de manera aditiva; está compuesta de tres canales, uno rojo (R), uno verde (G) y uno azul (B). Cada canal puede tomar valores discretos entre 0 y 255 donde 0 representa la ausencia total de ese color y 255 la intensidad máxima; de esta manera los colores disponibles en este sistema se generan a partir de la combinación de estos 3 canales, así, por ejemplo, el negro es representado por (0,0,0) y el blanco por (255,255,255).

2.1.2. Resolución

La resolución, en el contexto de imágenes, corresponde al numero de píxeles efectivos que posee una imagen, esto está determinado por su alto y largo (en píxeles). La resolución nos

da una idea de la cantidad de información que posee una imagen.

Si bien las imágenes al ser capturadas tienen una resolución dada, pueden ser reescaladas, aunque no sin inconvenientes, pues al disminuir la resolución se pierde información que no se puede volver a recuperar; por otro lado al aumentar la resolución de manera artificial no se gana nueva información, si no que los píxeles originales ocupan mas espacio en una matriz más grande.

2.1.3. Representación RGB de una Imagen Digital

Una imagen digital, utilizando el sistema de colores RGB, es representada como una matriz de 3 dimensiones, ejemplificada en la Figura 2.1; las columnas representan los píxeles a lo ancho de la imagen; las filas en tanto representan los píxeles a lo largo; la tercera dimensión, por otra parte, esta compuesta por los canales RGB.

En el caso de este trabajo, para poder utilizar las imágenes como entrada para las redes neuronales, es necesario mapear los valores RGB para que estén distribuidos entre los valores flotantes de 0,0 a 1,0.

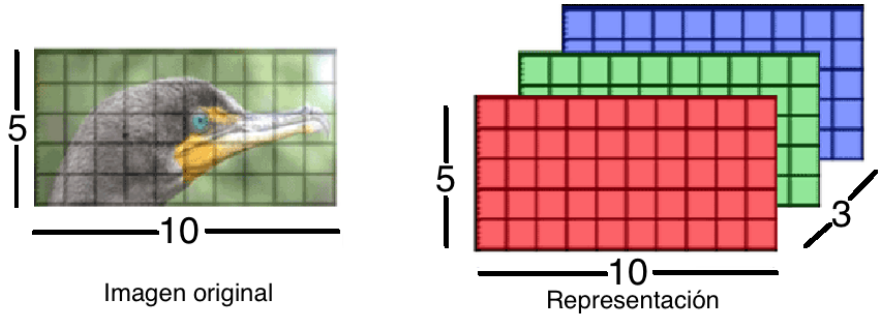

Figura 2.1: Representación matricial de una imagen digital.
Fuente: https://www.researchgate.net

2.1.4. Compresión

Una imagen al ser almacenada digitalmente ocupa un espacio en la memoria; si bien uno esperaría que todas las imágenes de la misma resolución ocuparan el mismo espacio, en la practica no ocurre, esto debido a que las imágenes son guardadas de manera comprimida; esta compresión puede ser tanto *lossy* como *lossless*.

Compresiones *lossy* (con perdida) corresponden a métodos de compresión en que se realizan modificaciones a la imagen de manera que pueda ser representada en un menor espacio; estas modificaciones, si son bien hechas, resultan en imágenes prácticamente indistinguibles, visualmente, de la original pero que pueden ocupar un espacio mucho menor. El formato de imagen más común que utiliza esta compresión es JPEG.

Las compresiones *lossless* (sin perdida), por otro lado, no modifican la imagen original y por lo tanto no hay perdida de información pero logran representar las imágenes en un espacio mas pequeño. Un formato común que ocupa compresión *lossless* es PNG (este es el formato utilizado para las imágenes usadas en este trabajo).

Cuánto una imagen puede ser comprimida depende en gran medida de la distribución de los valores de sus píxeles, es decir, qué tan distintos sean estos, así, por ejemplo, imágenes monocromas son, potencialmente, mucho más compresibles que imágenes que presentan grandes variaciones de color.

2.1.5. Scikit-image

Scikit-image[*] es una librería para Python, pertenciente al ecosistema de librerías para ciencias, matemáticas e ingeniería SciPy[**], que ofrece una colección de algoritmos para el procesamiento de imágenes. Entre sus funcionalidades, que se utilizarán en este trabajo, están el convertir imágenes RGB a su versión de punto flotante, operaciones que permiten obtener información cuantitativa de las imágenes, así como también reescalar y recortar imágenes.

2.2. Redes Neuronales

En esta sección se discutirán aquellos conceptos relacionados con redes neuronales, así como también se mostrarán aquellos modelos utilizados durante el desarrollo de esta memoria

2.2.1. Red Neuronal Artificial

Una red neuronal artificial, o simplemente red neuronal, es una abstracción simplificada del modelo cerebral; en su versión más simple está compuesta por una capa de entrada que recibe información, por medio de un vector, que luego es procesada y cuyos resultados son entregados a una capa de salida que entrega un resultado. Estas capas están compuestas a su vez por módulos que emulan neuronas. Versiones más complejas de redes neuronales, como la mostrada en la Figura 2.2, poseen una o varias capas ocultas; estas capas están conectadas una después de otra recibiendo la información procesada por la capa anterior, procesándola nuevamente y pasándosela a la siguiente capa, así hasta llegar a la capa de salida; este tipo de red neuronal son, hoy en día, más utilizadas debido a que pueden resolver una cantidad mayor de problemas.

Las redes neuronales son especialmente competentes en la tarea de clasificar; para esto el método más común utilizado es el softmax: en primera instancia se le asigna a cada neurona de la capa de salida una etiqueta o categoría (debiesen haber tantas neuronas en esta capa

[*] `https://scikit-image.org`
[**] `https://www.scipy.org`

como categorías se requieran), por otra parte, cada una de las neuronas en la capa final entrega un valor entre 0 y 1, dependiendo de la información que se le entregó a la red, entonces, se toma la categoría con la neurona de salida que entrega el mayor valor como la clasificación que hizo la red neuronal para la entrada.

Una parte fundamental para que las redes neuronales puedan realizar sus tareas es el entrenamiento; para este trabajo los modelos de redes neuronales utilizados fueron entrenadas con un método denominado aprendizaje supervisado, en el cual se tienen varios ejemplos previamente etiquetadas con su categoría respectiva; estos ejemplos son entonces entregados a la red para que los clasifique. Aquí pueden ocurrir dos casos: el primer caso corresponde a que la etiqueta del ejemplo sea igual a la que entrega la red, en cuyo caso la red está bien y no debe ser ajustada; el segundo caso que puede ocurrir es que la red dé una categoría distinta a la etiqueta, cuando pasa esto se deben ajustar automáticamente los valores de las operaciones que se realizan, en las capas no finales, para lograr que cuando nuevamente se le entregue a la red una entrada similar a la mal clasificada, esté más cerca de la respuesta correcta.

Cabe mencionar que las redes aprenden solamente cuando están en modo de entrenamiento y cuando no lo están solamente entregan el resultado, sea este correcto o no, sin variar sus valores posteriormente.

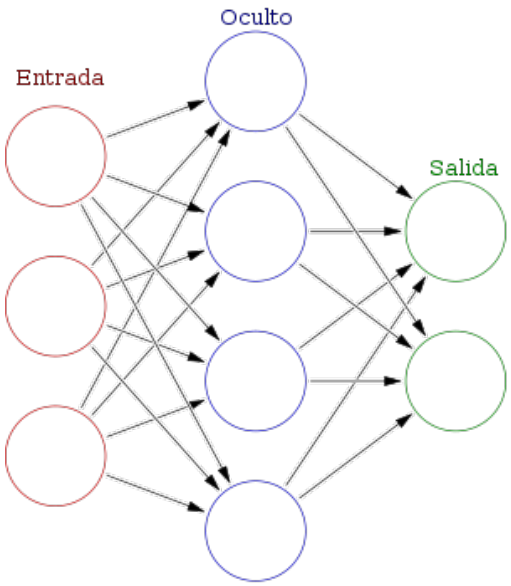

Figura 2.2: Red neuronal con una capa oculta.
Fuente: https://www.wikimedia.org/

2.2.2. Red Neuronal Convolucional

Las redes neuronales convolucionales son un caso de redes neuronales que han demostrado ser particularmente competentes a la hora de clasificar imágenes[9], siendo estas las más usadas hoy en día para estas tareas.

Como puede verse en la Figura 2.3, la forma más simple de una red convolucional está conformada por múltiples de capas de convolución, cada una seguida por una capa de muestreo, hasta llegar a una estructura de red neuronal parecida a la de la Figura 2.2.

Con respecto a la entrada de estas redes es clave entender la representación matricial de las imágenes, pues las operaciones de estas redes operan sobre matrices. Si bien en este trabajo se utilizarán imágenes en el sistema RGB, las redes neuronales convoluciones pueden también funcionar con otros, o varios, sistemas, como por ejemplo, escalas de grises; es necesario entonces conocer de antemano cuál debe usarse para el caso particular del modelo utilizado.

La entrada luego es pasada a una capa de convolución; estas se encargan de extraer características de la imagen mediante el uso de filtros. Estos filtros corresponden a matrices que operan sobre submatrices de la imagen y cuyos resultado componen el mapa de características para ese filtro de la imagen. Los filtros pueden ser de varios tipos y tamaños dependiendo de los elementos que se quieran rescatar, los cuales pueden ser texturas, lineas rectas, cambios bruscos de colores, etc. Las matrices obtenidas al utilizar cada filtro sobre la matriz original se denominan mapas de características; estas matrices son entonces entregadas a la siguiente capa, denominada capa de muestreo.

La capa de muestreo básicamente reduce las dimensiones de cada matriz manteniendo la información más relevante. Para esto hay muchas alternativas pero de manera simple se traducen a tomar una matriz que sirve de ventana para un grupo de píxeles contiguos a los cuales se le aplica una función, como calcular el promedio o tomar el máximo valor; este nuevo valor será el representante de esta ventana que formará parte de una matriz mas pequeña. Una vez que todos los mapas de características hayan sido evaluados se puede repetir el proceso de convolución o pasar directamente a la red multicapa.

Finalmente la red multicapa toma aquellas características de más alto nivel de la imagen para poder clasificarlas; esto permite abstraer toda aquella información que no es relevante para la red; el resultado entonces se obtiene mediante la aplicación de softmax sobre la capa de salida.

Con respecto al entrenamiento de redes neuronales convolucionales este es análogo al caso de las redes neuronales tradicionales, aunque es necesario entender que la información que recibe la red multicapa está dada por los pasos de convolución y muestreo y que estos pueden aplicar filtros que entregan características no relevantes.

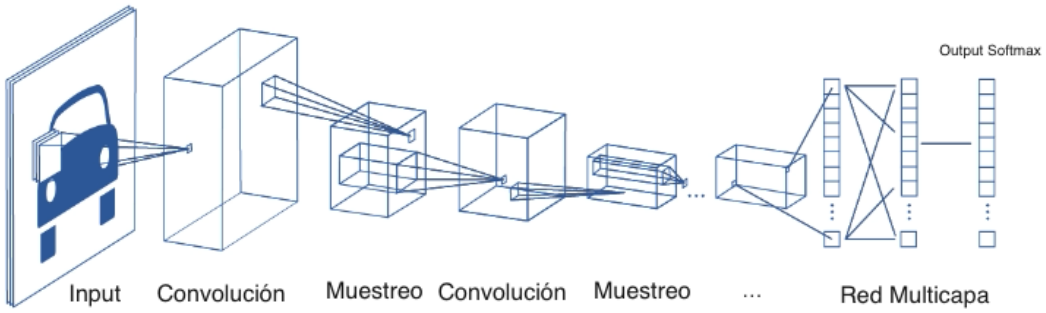

Figura 2.3: Red neuronal convolucional.
Fuente: https://www.mathworks.com

2.2.3. GPU

Una unidad de procesamiento gráfico o GPU (graphics processing unit) es una unidad de procesamiento al igual que una CPU pero orientada a operaciones de punto flotante, las cuales se encuentran principalmente en procesamiento de gráficos. La principal diferencia radica en que una GPU posee cientos de núcleos capaces de manejar miles de hilos, favoreciendo tareas repetitivas que requieren un alto grado de paralelismo, como es el caso de manejo de matrices. Su paralelismo hace que las GPUs sean especialmente eficientes para manejar redes neuronales convoluciones en donde gran parte del procesamiento se hace por medio de operaciones matriciales.

Hoy en día frameworks y librerías, como Caffe 2 y TensorFlow, utilizan funcionalidades de las GPU para lograr reducir los tiempos de entrenamiento e inferencia de las redes neuronales, siendo el entrenamiento una de las etapas que toma más tiempo, llegando a demorar días e incluso semanas para el caso de las redes más complejas.

Para lograr integrar las GPU se utilizan plataformas como CUDA u OpenCL; la primera es propietaria y desarrollada por Nvidia, por tanto, solamente puede ser utilizadas por las GPU de esta compañia; OpenCL, por otro lado, es de código abierto. En el caso de este trabajo se trabajará con la API de CUDA.

2.2.4. Caffe 2

Caffe 2 es un framework, desarrollado por Facebook, para redes neuronales con varias capas basado en Caffe [8]; este framework agrega mayor modularidad y nuevas funcionalidades para el manejo y despliegue de redes neuronales.

Entre las ventajas de Caffe 2 se encuentra que puede procesar un gran volumen de datos por hora y está especialmente optimizado para hacer uso de GPU para las labores de clasificación y entrenamiento. Adicionalmente soporta varios modelos especializados en clasificación como lo son las redes neuronales convolucionales. También existe gran disponibilidad de redes neuronales ya entrenadas para este framework, las cuales pueden ser utilizados fácilmente.

Actualmente se encuentra en proceso de formar parte de PyTorch, una librería de aprendizaje de máquinas orientada a Python, por lo que se recomienda utilizar la API de PyTorch a futuro.

2.2.5. GoogLeNet

GoogLeNet[14] es una red neuronal convolucional, creada por un equipo de Google, ganadora del Large Scale Visual Recognition Challenge (ILSVRC) obteniendo un porcentaje de clasificaciones incorrectas para el top-5 (en una capa softmax, la categoría correcta debe estar entre los 5 valores mas altos) del 6.66 %; esto es relevante pues corresponde a la primera red neuronal capaz de obtener resultados comparables con el ser humano que, para el caso de un experto [***], obtiene errores cercanos al 5.1 %.

Esta red neuronal está compuesta por 22 capas y alrededor de 4 millones de parámetros; su principal diferencia está radicada en las denominadas capas *inception* (Figura 2.4); 9 de las 22 capas son de este tipo. Los módulos *inception* proveen operaciones de convolución en paralelo con distintos tamaños, permitiendo, de esta manera, extraer características con distintos grados de detalle, en particular se utilizan convoluciones de 1×1, 3×3 y 5×5; finalmente todos los mapas de característica son concatenados para entregarlos a la siguiente capa.

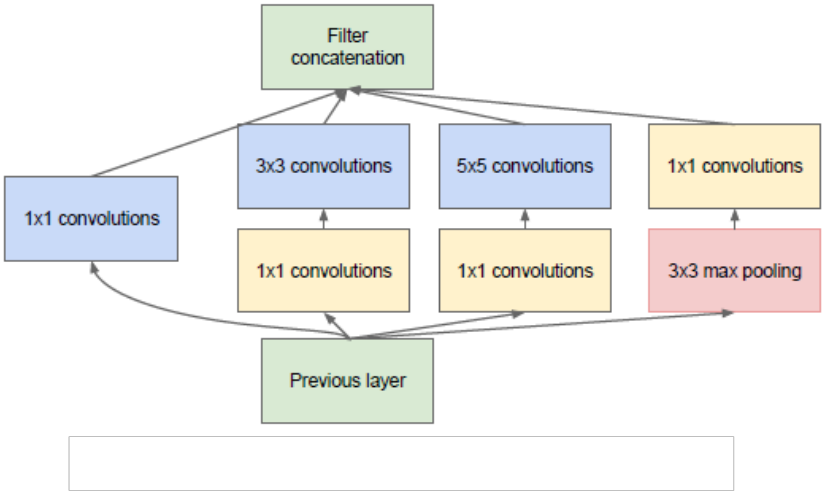

Figura 2.4: Módulo *inception* de GoogLeNet.
Fuente: Going Deeper with Convolutions[14]

.

[***] Andrej Karpathy `http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/`

2.2.6. SqueezeNet

SqueezeNet[7] es una red neuronal convolucional desarrollada por miembros de DeepScale, UC Berkeley y Stanford que provee niveles de exactitud similares a AlexNet[9] (80,3 % para top-5) pero con un 2 % de los parámetros, esto hace que sea una red relativamente pequeña y fácil de desplegar, en especial en dispositivos con escasa memoria disponible.

El menor tamaño y similar exactitud se consiguen básicamente utilizando 3 estrategias:

- Privilegiar filtros de tamaño 1×1 en vez de tamaños mas grandes.
- Disminuir el número de canales de input a los filtros mediante la utilización de módulos denominados *fire*.
- Retrasando el uso de las capas de muestreo para que las capas de convolución tengan mapas de activación (cuan grande es la señal que proveen las neuronas en la capa) más grandes, es decir, se procesan varias capas de convolución antes de pasar por una capa de muestreo.

Los módulos *fire* mostrados en la Figura 2.5 están conformado por dos partes. La primera parte, denominada squeeze, comprime el número de canales de los inputs de la capa mediante uso de filtros de tamaño 1×1, estos después alimentan la segunda parte del módulo, denominada expand, que expande el input entregado por el módulo anterior mediante el uso de filtros de tamaño 1×1 y 3×3.

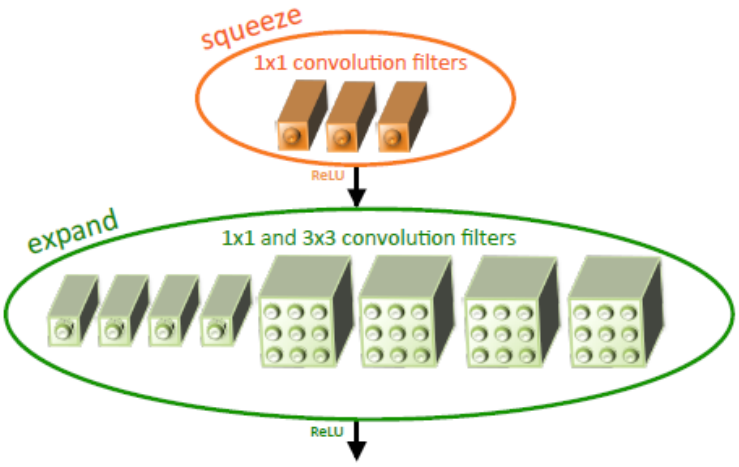

Figura 2.5: Módulo *fire* de SqueezeNet.
Fuente: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size [7]

.

2.3. Clasificación de Imágenes

En esta sección se presentarán conceptos relacionados con la clasificación de imágenes por parte de las redes neuronales.

2.3.1. Ejemplo Adverso

Los ejemplos adversos son entradas que son clasificadas de manera incorrecta por las redes neuronales; estas pueden ser entradas originales o creadas mediante la aplicación de ciertos filtros, como ruidos, a entradas que en principio eran clasificadas correctamente, filtros que podrían ser indetectables, visualmente, por el ser humano.

Los ejemplos adversos son a menudo utilizados en la etapa de entrenamiento de las redes neuronales para mejorar su robustez; estos ejemplos también pueden ser generados para crear entradas falsas para que sean clasificadas de cierta manera, como por ejemplo, al querer traspasar sistemas de reconocimiento facial.

2.3.2. Ejemplo Positivo Mínimo

Los ejemplos positivos mínimos (en el contexto del trabajo también se denominarán imágenes mínimas) corresponden a imágenes que contienen la mínima información necesaria para poder ser identificadas por un clasificador, humano o máquina; esto se basa en el principio que los clasificadores tienden a centrarse en ciertas características de las imagines para lograr identificarlas. Estas características pueden ser: formas, texturas, ciertos contextos, etc.

En el caso de las redes neuronales, estas características dependen de elementos como la arquitectura de las redes o las imágenes con las que fue entrenada, así, fundamentalmente, es posible que las imágenes mínimas de una red neuronal no presenten ningún problema para otra y viceversa.

Las ejemplos positivos mínimos pueden ser obtenidos mediante la utilización de filtros que disminuyan la prominencia de ciertas características como pueden ser la disminución de colores disponibles, recortes de imagen, etc.

2.3.3. ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

La ILSVRC es un desafió anual, comenzado por ImageNet[****], que busca evaluar algoritmos para la detección y clasificación de imágenes; la principal motivación de esta competencia es fomentar y comparar el progreso en el área de visión de máquinas. En esta competencia, las redes neuronales presentadas deben ser entrenadas por un mismo conjunto de imágenes, provistas por ImageNet, para así poder comparar las distintas arquitecturas de manera correcta, esto pues el desempeño de una red neuronal está muy ligado a las imágenes utilizadas en la fase de entrenamiento.

[****] http://www.image-net.org

2.4. Trabajo relacionado

En esta sección se presentarán los trabajos relacionados que cimientan la presente memoria, además de una discusión sobre el algoritmo utilizado para obtener los ejemplos positivos mínimos.

2.4.1. Atoms of Recognition in Human and Computer Vision[15]

En este trabajo se presenta el concepto de *imágenes mínimas reconocibles*; estas corresponden a las mínimas zonas dentro de una imagen que son reconocidas por seres humanos; estas zonas mínimas son locales y se centran en ciertas características de la figura que se quiere reconocer; mostrando así aquellos distintos elementos claves, o átomos de reconocimiento, para la tarea de clasificación humana.

El principal aporte de este trabajo para esta memoria es mostrar que existen características que, si bien son utilizadas por los seres humanos, no están siendo aprovechadas por las redes neuronales actuales; esto indica una diferencia fundamental entre la forma en que las personas y las redes neuronales atacan el problema de reconocimiento. Por otra parte se mostró que pequeños cambios en las imágenes mínimas reconocibles reducen drásticamente la capacidad de los seres humanos para clasificarlas.

2.4.2. Minimal Images in Deep Neural Networks: Fragile Object Recognition In Natural Images[12]

Este estudio extiende el trabajo mostrado la subsección 1.4.1, mostrando que en las redes neuronales también ocurre el fenómeno de disminución drástica en la exactitud de la clasificación, esto al tratar de clasificar zonas de una imagen que solo diferían por pocos píxeles; estas zonas borde son denominadas *imágenes de reconocimiento frágil*. Estas imágenes serían, en promedio, de mayor tamaño y habrían muchas más para las redes neuronales que para los seres humanos.

Se puede extraer entonces que para el caso de disminuir la región, quitando pistas contextuales, los seres humanos y redes neuronales convolucionales comparten el mismo problema (en menor y mayor grado).

2.4.3. Benchmarking Neural Network Robustness to Common Cocrruptions and Perturbations[6]

En este trabajo se busca establecer un benchmark para determinar la robustez de una red neuronal convolucional; para ello se seleccionan perturbaciones comunes, tales como ruido gaussiano, pixelación o compresión JPEG, para luego ser aplicados, en diferentes intensidades,

a un conjunto de imágenes del conjunto de validación de ImageNet; luego este nuevo conjunto es clasificado, desprendiéndose la robustez deseada tomando en cuenta la intensidad de las perturbaciones. Entre las conclusiones de este trabajo se desprende que las redes neuronales más grandes presentan una mayor robustez ante la presencia de ruidos.

2.4.4. ImageNet-trained CNNs are Biased Towards Texture; Increasing Shape Bias Improves Accuracy and Robustness[4]

En este trabajo se demuestra que las redes neuronales convolucionales actuales, entrenadas con el conjunto de imágenes de ImageNet (que son las usadas por las redes neuronales ocupadas en la memoria), están, por defecto, sesgadas hacia la utilización de texturas, en contraste con las formas, como principal característica para determinar la clase de una imagen; esta es una importante diferencia pues los seres humanos se centran más en las formas; de esta forma, si se quiere que las redes neuronales clasifiquen de manera similar a las personas, es necesario revertir el sesgo. Una posible solución a esto es estilizar las imágenes del conjunto de entrenamiento para que las texturas sean menos prominentes. Un resultado muy relevante para esta memoria es que aquellas redes neuronales convolucionales que han sido entrenadas de esta manera presentan una mayor robustez frente a ruidos.

2.4.5. Algoritmo de Powell

El algoritmo de Powell [11] es un método de optimización multivariable que no requiere de las derivadas de las funciones a minimizar.

Considerando x=[x_0...x_n] como solución de partida, p=[p_0...p_n] las direcciones de movimiento en las distintas dimensiones, TIMES las veces que se quiera repetir y MIN(i) una función que avanza x al mínimo en la dirección p[i], en seudo código el algoritmo es:

```
for i in range(0,TIMES):
    y = x
    for i in range(1,n):
        MIN(i)
    for i in range(1,n-1):
        p[i] = p[i+1]
    p[n] = y-x
    MIN(n)

return x
```

Capítulo 3

Problema

Actualmente los ejemplos adversos son utilizados principalmente durante la fase de entrenamiento para mejorar la robustez de las redes neuronales mediante modificaciones a los conjuntos de entrenamiento originales[13]. Este acercamiento tiene la desventaja que mejora la robustez en un espacio de acción similar a las distorsiones utilizadas en el proceso de entrenamiento. Otras formas de combatir las imágenes distorsionadas, tanto por errores como maliciosamente modificadas, se centran en la detección de estas modificaciones antes de entrar a la red[10] para luego ser pre procesadas mediante filtros para revertir o minimizar los efectos que condicionan una mala clasificación[16].

Lamentablemente todos estos métodos pueden ser sobrepasados al construir nuevas funciones de perdida que alteren las imágenes, siendo estas potencialmente infinitas[2]. De esta manera los métodos de detección de ejemplos adversos como los referenciados anteriormente tendrían una vida útil bastante corta. Esto significaría un gran peligro para sistemas que implementen visión computacional para tareas críticas que puedan ser objetivos de sabotaje, por lo que seguir por ese camino sería bastante limitante aunque por ahora es suficiente.

Por otro lado, obtener un mayor entendimiento de las redes, así como de las características de las que depende una correcta clasificación nos podría llevar a desarrollar modelos mucho mejores en lo que a seguridad respecta, ayudando así a la masificación de las tecnologías basadas en aprendizaje de máquinas.

Si bien en el último tiempo se han hecho grandes avances en el área de clasificación por medio de redes neuronales, incluso se han desarrollados modelos de redes que son capaces de superar a humanos expertos[1], también han salido a la luz casos donde se han desplegado tecnologías de alto impacto social sin que estén lo suficientemente maduras, obteniéndose pésimos resultados(*)(**), este tipo de situaciones no hacen más que generar desconfianza en la población, la cual es necesaria si se quiere que las máquinas cada vez apoyen más nuestras labores.

*https://www.technologyreview.com/f/613922/london-polices-face-recognition-system-gets-it-wrong-81-of-the-time/

**https://www.pcmag.com/news/369398/human-help-wanted-why-ai-is-terrible-at-content-moderation

En primera instancia es necesario identificar como se comportan distintas redes ante la perdida de información en las imágenes que se les presentan, esto pues en un contexto de uso real las imágenes pueden no ser de la mejor calidad o pueden tener distorsiones que afecten el rendimiento de las redes; por ejemplo, una red neuronal, que recibe información de una cámara de seguridad exterior, está expuesta a que las imágenes que tiene que analizar estén distorsionadas debido a cambios súbitos de luz, cambios de colores debido al paso del día, objetos que no pueden ser identificados debido a que están muy lejos o muy cerca. Es clave entonces saber de antemano aquellas situaciones que podrían afectar el desempeño de las redes neuronales para poder subsanar, si se puede, cualquier factor que dificulte su labor.

Finalmente, tomando en cuenta que las redes neuronales buscan reemplazar el trabajo realizado por un humano, es justo compararlas con estos para así entender aquellas variables y limitantes que hacen a uno otro más adecuado para ciertas labores y a la vez permitir el desarrollo de modelos cada vez mejores.

Capítulo 4

Solución

En esta sección se discutirá la solución presentada para abordar los problemas de caracterizar las redes neuronales ante la pérdida de información, así como también, el poder comparar el desempeño humano con el de las redes neuronales.

Para efectos de la realización de este trabajo se separó la solución de acuerdo a los dos problemas a estudiar.

4.1. Caracterización de las Redes Neuronales al Enfrentarse a Pérdida de Información de las Entradas

Esta sección se centrará en comparar redes neuronales entre sí, a fin de establecer las diferencias o similitudes en el proceso de clasificación de imágenes de diferentes categorías disminuyendo ciertos parámetros que hacen a una imagen identificable.

4.1.1. Diseño General

Para poder caracterizar las redes de acuerdo a cómo se ve afectado su desempeño al disminuir la información disponible de las imágenes que se le muestran para clasificar, se hará uso de filtros que permitan disminuir, de forma monótona, el tamaño, en bytes, de las imágenes; para esto asumiremos que este parámetro representará la entropía de los datos que conforman la imagen, esto pues mediante algoritmos de compresión sin pérdida de información es posible representar la imagen en un espacio cada vez menor cuando se aplican los filtros con cada vez más intensidad.

Por otra parte, entendiendo que existe un límite para el cual no es posible para la red neuronal, con la información disponible, continuar clasificando correctamente las imágenes, es que se podrán establecer los parámetros de los filtros disponibles que generarán las imá-

genes más pequeñas posibles, en bytes, que aún son capaces de ser reconocidas por las redes neuronales, obteniendo, de esta manera, las imágenes positivas mínimas para cada clase por cada filtro; esto permitirá no solo cuantificar las características de estas, sino que también obtener un registro de imágenes mediante el cual se podrá realizar un análisis cualitativo de aquella información que se va perdiendo, pudiendo así compararlas con seres humanos.

4.1.2. Redes Neuronales Utilizadas

En principio se deben seleccionar las redes neuronales a utilizar; para esto se deben considerar aquellas arquitecturas que han mostrado un mejor desempeño en los últimos tiempos, como por ejemplo los ganadores de la ImageNet Large Scale Visual Recognition Competition (ILSVRC). Para este trabajo en particular se utilizaron las redes GoogLeNet y SqueezeNet disponibles en `https://github.com/caffe2/models`; la primera se seleccionó puesto que es la primera en obtener resultados similares al ser humano en la tarea de clasificar imágenes mientras que la segunda es una red mucho más pequeña, por lo que es más fácil de desplegar en distintos contextos de uso real.

Para este trabajo no se experimentará con un número mayor de redes neuronales debido a que la segunda parte conlleva una comparación con humanos, razón por la cual se privilegió el obtener más resultados para los ejemplos positivos mínimos de las dos redes mencionadas en el párrafo anterior que obtener pocas comparaciones para muchas redes, esto pues que el recurso humano es escaso.

4.1.3. Categorías Utilizadas

De las mil categorías disponibles, el trabajo se centró en las 20 categorías y macro categorías para la que se entrenó la versión de SqeezeNet que se usó y para el caso de GoogLeNet se hizo un mapeo de las categorías que están dentro de las categorías más grandes para el caso que fuese necesario.

El número reducido de categorías se definió debido a que se requería poder replicar la tarea de clasificación con seres humanos y se consideró que si se hubiesen presentado muchas opciones resultaría una tarea muy complicada, complejidad debida a que, en primera instancia, los voluntarios debían familiarizarse con las etiquetas disponibles para evitar que escogieran una equivocadamente por ignorar que existía una más adecuada, más aún, es necesario poder presentar todas las categorías disponibles sin utilizar elementos como desplazamiento de páginas o sub menús de etiquetas; en particular 20 categorías caben cómodamente en una interfaz siendo los suficientemente grandes como para que puedan ser leídas sin problemas por los usuarios.

A continuación se presentarán las categorías, indicadas con color rojo las que engloban un conjunto de las etiquetas disponibles originalmente y en negro aquellas que ya estaban disponibles por si solas en el esquema de categorías.

Categorías:

- Araña
- Ave
- Flor
- Fruta
- Gato
- Hipopótamo
- Hongo
- Vegetal/Hortaliza
- Insecto
- León
- Lobo
- Mono
- Oso
- Perro
- Pez
- Reptil
- Tiburón
- Tigre
- Vehículo
- Zorro

4.1.4. Imágenes Utilizadas

Para los experimentos se emplearon las imágenes utilizadas para la ILSVRC, disponibles en `https://image-net.org`.

Debido a que ambas redes neuronales son distintas y pueden haber diferencias entre las imágenes que detectan correctamente cada una, se utilizarán aquellas pertenecientes a la intersección de los conjuntos de imágenes que son clasificados de manera correcta por ambas redes neuronales, esto pues lo que se quiere analizar son las características que utiliza cada red para tomar una decisión correcta y no simplemente cual red es mejor clasificando. Además, las imágenes utilizadas pertenecen a un conjunto de validación que no ha sido utilizado por ninguna de las redes en el proceso de entrenamiento; tomando en cuenta todo lo anterior, por cada categoría se seleccionaron 100 imágenes de manera aleatoria que fuesen clasificadas correctamente por ambas redes obteniéndose una muestra de 2000 imágenes de tamaño 224×224 píxeles; este tamaño es necesario para que las imágenes fuesen entradas válidas para las redes neuronales utilizadas (este número puede variar según las características de las redes neuronales, pudiendo ser números fijos o rangos entre un tamaño mínimo y máximo).

4.1.5. Filtros Utilizados

Para establecer aquellos mecanismos de pérdida o falta de información más comunes a los que se puede enfrentar una red en un contexto real es que se eligieron los siguientes filtros, los que se aplicarán, de manera individual o conjunta, a las imágenes antes de ser presentadas a la red neuronal:

- **Disminución de la resolución real de la imagen:** Esto constará de 2 fases: durante la primera fase se disminuirá la resolución de la imagen por un factor dado por medio

de la función *resize* de scikit-image, posteriormente, en la segunda parte, se vuelve a la resolución original pero manteniendo las características de la imagen de menor resolución; esto se logra utilizando la misma función *resize* pero desactivando el *anti-aliasing*. La segunda parte es necesaria para mantener la invariante del tamaño (en pixeles) de la imágenes utilizadas, de esta manera se asegura que las entradas puedan ser procesadas por las redes neuronales.

Un ejemplo de este filtro puede verse en la Figura 4.1, en donde la información de una imagen de la categoría león es reducida a la encontrada en una imagen con el 10 % de la resolución real de la imagen original.

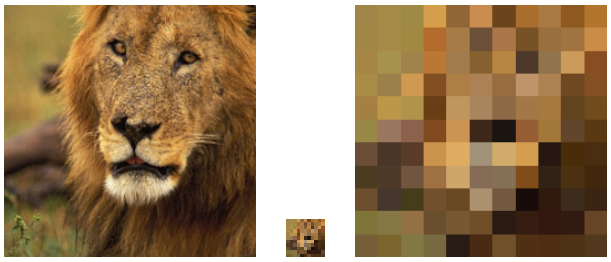

Figura 4.1: Aplicación del filtro disminución de resolución real con valor 0.1, imagen inicial (izquierda), imagen después de la primera fase (centro), imagen final (derecha).

- **Recorte:** Este filtro posee 4 sub-filtros recorte desde arriba, desde abajo, derecho e izquierdo; cada uno transforma las líneas horizontales (caso de las dos primeras) o verticales (caso dos últimas) en líneas blancas, reemplazando los píxeles originales de la imagen por píxeles con valor (1.0, 1.0, 1.0) (equivalente en la escala RGB a (255, 255, 255)), desde el borde y en la dirección del filtro; se hace esto, en vez de simplemente recortar, pues se requiere que la zona no afectada mantenga la resolución pero por otro lado es necesario que la imagen tenga un mismo tamaño (en píxeles).

  Un ejemplo del filtro recorte es mostrado en la Figura 4.2 en donde se aplica el filtro a una imagen de la categoría perro con valores recorte desde arriba del 20 %, recorte desde abajo del 50 %, recorte desde la izquierda del 20 % y recorte desde la derecha del 50 %; esto da como resultado una imagen que enmascara todos los píxeles recortados con píxeles blancos dejando solamente el cuadrado correspondiente a una sección del hocico del perro.

Figura 4.2: Aplicación del filtro recorte para arriba=0.2, abajo=0.5, derecha=0.5, izquierda=0.2. Imagen inicial a la izquierda e imagen final a la derecha.

- **Disminución de la cantidad de colores:** Este filtro establece la cantidad de valores que puede tomar cada píxel dentro de la escala RGB. Inicialmente cada píxel puede tomar valores desde 0 a 255 en incrementos de 1; al aplicar el filtro entonces el tamaño de ese incremento se hace mas grande perdiéndose todos los valores intermedios, así, por ejemplo, si inicialmente se tienen 255^3 colores disponibles, aplicar el filtro dejando el 10 % de los colores (25 valores por canal), deja un total disponible de 25^3. Para realizar este filtro simplemente se inicializa una lista de valores posibles, equidistantes, dado la cantidad de valores que puede tomar cada canal; posteriormente, cada valor de los canales de cada píxel de la imagen se aproxima al valor más cercano de lista de valores posibles.

  La aplicación de este filtro se ilustra en la Figura 4.3 donde se reduce la cantidad de colores disponibles al 1 % para una imagen de la categoría fruta.

Figura 4.3: Aplicación del filtro disminución de colores al 1 % de colores disponibles. Imagen inicial a la izquierda e imagen final a la derecha.

- **Agregación de todos los filtros:** este ultimo filtro simplemente aplica todos los filtros al mismo tiempo para lograr determinar cómo se comporta frente a este caso extremo.

Con el fin de que el estudio de cada filtro por separado pueda ser comparable es necesario establecer que cada uno tendrá una intensidad que va desde 0 a 1 donde 0 implica que el filtro no sea aplicado y 1 que el filtro ha sido aplicado completamente.

Los filtros mostrados en esta sección, incluyendo la combinación, han sido seleccionados debido a que son comúnmente encontrados en contextos reales en los cuales las redes neuronales se pueden desempeñar.

El caso del filtro que disminuye la resolución real de la imagen se presenta en casos, como por ejemplo, cuando la cámara que captura las imágenes tiene un sensor de baja resolución o también el caso cuando el objeto que se quiere identificar se encuentra muy lejano por lo que la cantidad de píxeles que lo representan es baja.

Para el caso del filtro recorte, es posible encontrarlo en imágenes obtenidas donde falta información contextual, como pueden ser instancias donde hay elementos que tapen parte del objeto que se quiere identificar, así como también casos donde la imagen fue obtenida de muy cerca.

El filtro que disminuye el espacio de colores disponibles puede ser encontrado en situaciones

donde las imágenes hayan pasado por un proceso de compresión *lossy*, condiciones extremas de luz tanto baja como alta, así como también imágenes tomadas por cámaras poco sensibles.

Es necesario destacar que, si bien gran parte de los problemas pueden ser solucionados con mejores cámaras, en el contexto actual, se tiene que considerar que no toda la población tiene acceso a estas y la idea es que estas tecnologías estén disponibles para la mayor cantidad posible de personas.

4.1.6. Obtención de Ejemplos Positivos Mínimos

Con el fin de obtener los ejemplos positivos mínimos hay que dividir el análisis en los casos unidimensionales y multidimensionales; el primero corresponde a aquellos filtros para los cuales, para minimizar el tamaño (en bytes) de las imágenes, basta con aumentar la intensidad del filtro; el segundo caso corresponde a aquellos filtros que se conforman por múltiples filtros y que, si bien aumentar la intensidad de un filtro disminuye la confianza que tiene la red neuronal para cierta imagen, el ejemplo positivo mínimo es obtenido por una combinación de todos los filtros y por tanto pueden haber muchos mínimos locales que son mayores al global, que es el que se busca.

- Caso unidimensional: Este corresponde a los filtros que disminuyen la calidad de imagen y la cantidad de colores disponibles. Para obtener las imágenes mínimas basta con aumentar paulatinamente el filtro hasta que ya no se logre detectar correctamente la imagen; esto ocurre cuando la categoría con mayor confianza (valor de salida de la capa softmax de la red neuronal para cada categoría) pasa a ser otra distinta a la original, esto debido a que se han perdido las características que la red neuronal buscaba para identificar la categoría original. Finalmente, el ejemplo positivo mínimo corresponde a la última imagen que fue detectada correctamente.
- Caso multidimensional: Para esto es necesario utilizar algoritmos de optimización, dado que un algoritmo de fuerza bruta puede ser muy costos, en particular, dado que las funciones se comportan de manera bastante monótona, se decidió utilizar el algoritmo de Powell. Para garantizar que este método convergiese al mínimo, se comparó con un algoritmo de fuerza bruta, resultando que para el caso en que el algoritmo de Powell itera 6 veces, se llegó al resultado esperado, provisto por el algoritmo de fuerza bruta, con un margen de error menor al 5 %, es decir, la diferencia entre los tamaños, en bytes, de los ejemplos positivos mínimos obtenidos por ambos métodos no supera el 5 %.

Para definir el ejemplo positivo mínimo, es necesario establecer una función objetivo que corresponda una métrica de orden total, es decir, que una imagen sea menor a otra de manera inequívoca; para esto es necesario establecer una noción de entropía en la imagen, que para este caso, se considerará como la cantidad de información contenida en la imagen. En particular, la entropía se estimará en base al tamaño de la imagen, en bytes, luego de aplicar un algoritmo de compresión *lossless*, que en este caso se aplica al guardar la imagen en formato PNG; de esta forma la función que se quiere minimizar corresponde al tamaño de la representación, en formato PNG, de la imagen.

4.2. Comparación de las Redes Neuronales con Humanos

Esta sección busca establecer las diferencias entre el ser humano y las redes neuronales al ser expuestas a la misma pérdida de información, en particular se busca comparar el desempeño de los seres humanos frente a las imágenes mínimas generadas en la sección anterior para comprender aquellas variaciones a las que son más sensibles.

4.2.1. Diseño General

Para poder comparar el desempeño de las redes neuronales con el de un ser humano es necesario diseñar un framework que simule las condiciones en que las redes neuronales clasifican; esto es un proceso simple, dado que basta con que se le presente una imagen para que luego decida a qué clase pertenece la imagen mostrada. Aunque el framework sea sencillo hay que establecer ciertas condiciones para evitar ciertas ventajas inherentes a cada modelo:

- No se puede volver a mostrar una entrada previamente clasificada.
- Debe haber un número bajo de categorías, tal que el ser humano sea capaz de guardarlas todas en su memoria a corto plazo.

El primer caso se desprende de que los seres humanos aprenden activamente, es decir, al clasificar de manera errónea son capaces de internalizar el resultado inmediatamente, lo que se traduce en que si la misma imagen es mostrada una segunda vez probablemente logre clasificarla de manera correcta.

La segunda condición es necesaria para estandarizar la salida de la red neuronal con la del ser humano. Para lograr esto se barajaron varias ideas, tales como sacar una muestra de las opciones disponibles para ser presentadas a las personas, lo cual significaría que la red neuronal tuviese más posibilidades de equivocarse al tener más opciones de donde elegir (1 entre 1000), o implementar un sistema de búsqueda donde el voluntario escribiese la opción que cree correcta, pero esto podría llevar a casos donde se buscaran etiquetas no disponibles, al no poseer la habilidad de recordar las 1000 categorías o que buscase otros elementos presentes en la imagen, como por ejemplo pasto. Finalmente, se decantó por utilizar una red neuronal que fuese entrenada para identificar menos categorías.

Para desarrollar los experimentos se decidió implementar una aplicación web que pudiese poner a prueba a los distintos usuarios voluntarios frente a las imágenes mínimas generadas, esto pues dada la gran cantidad de imágenes disponibles era necesario obtener un gran número de voluntarios para que todas las variantes de las imágenes fuesen clasificadas.

Entendiendo el riesgo que un experimento en línea conlleva, pues se entiende que no todos participarán de buena fe, fue necesario dividir el experimento en dos etapas:

En la primera fase se realizará el experimento con un grupo de 20 voluntarios donde cada uno clasificará 125 imágenes únicas, esto con el fin de establecer un desempeño base. Para la segunda fase se liberará la aplicación a la web para que cualquiera pueda realizar el

experimento y de esta manera obtener resultados más generalizables.
Una vez completada la segunda fase se tomarán como válidos todos aquellos resultados obtenidos por personas que estén dentro de dos desviaciones estándar del promedio de la precisión base de la fase 1.

Para poder establecer que el desempeño de los usuarios se deba efectivamente a la pérdida de información, y no a la inhabilidad para clasificar ciertas imágenes en particular, es necesario establecer una línea de base para todas las categorías, por lo que también se les deberá mostrar imágenes que no presenten pérdida de información.

4.2.2. Imágenes Utilizadas

Se utilizó un subconjunto de las imágenes mínimas obtenidas en el punto anterior caracterizado de la siguiente manera: Por cada una de las 20 categorías a estudiar se tomaron 20 imágenes al azar obteniendo un total de 400 imágenes únicas; para cada una de estas imágenes se agregaron sus versiones mínimas para cada filtro más la imagen mínima compuesta de todos los filtros, esto para ambas redes neuronales, obteniéndose un conjunto a utilizar de 3600 imágenes posibles para este experimento (considerando también las 400 imágenes originales).

4.2.3. Información a Recopilar

Para poder evaluar las respuestas de los humanos es necesario, para cada participante, obtener los valores obtenidos para cada instancia de clasificación:

- Filtro utilizado: Esta variable corresponde al filtro utilizado para minimizar la imagen el cual puede ser *original*, para el caso que la la imagen no tenga filtro, *resolución*, *colores disponibles*, *recorte* y finalmente *mínima* cuando se utilizan los tres filtros; esto servirá para determinar aquellos filtros para los que el ser humano es mas sensible.
- Red que minimiza: En caso que aplique (salvo donde se muestra la imagen original), esta entrada permitirá establecer cómo se compara cada red (GoogLeNet y Squeezenet) de manera individual con la persona.
- Clasificación real: Esta corresponde a la etiqueta que tiene la imagen original.
- Clasificación humano: Esta variable es la clase que la persona determino como la correspondiente a la imagen que se le mostró.

4.2.4. Diseño de Interfaz

La interfaz del experimento cuenta con dos vistas. La primera, como se puede ver en la Figura 4.4, muestra una imagen a clasificar y como leyenda *Adivina a qué categoría pertenece la imagen mostrada*, de esta manera, el participante puede identificar rápidamente lo que debe realizar en el experimento. En la columna del lado derecho se encuentran las distintas

categorías disponibles de las cuales el usuario puede seleccionar aquella a la que cree pertenece la imagen. La idea principal es mantener una interfaz simple e intuitiva para así evitar que los usuarios tengan que leer instrucciones muy complicadas y así maximizar la cantidad de personas dispuestas a participar del estudio.

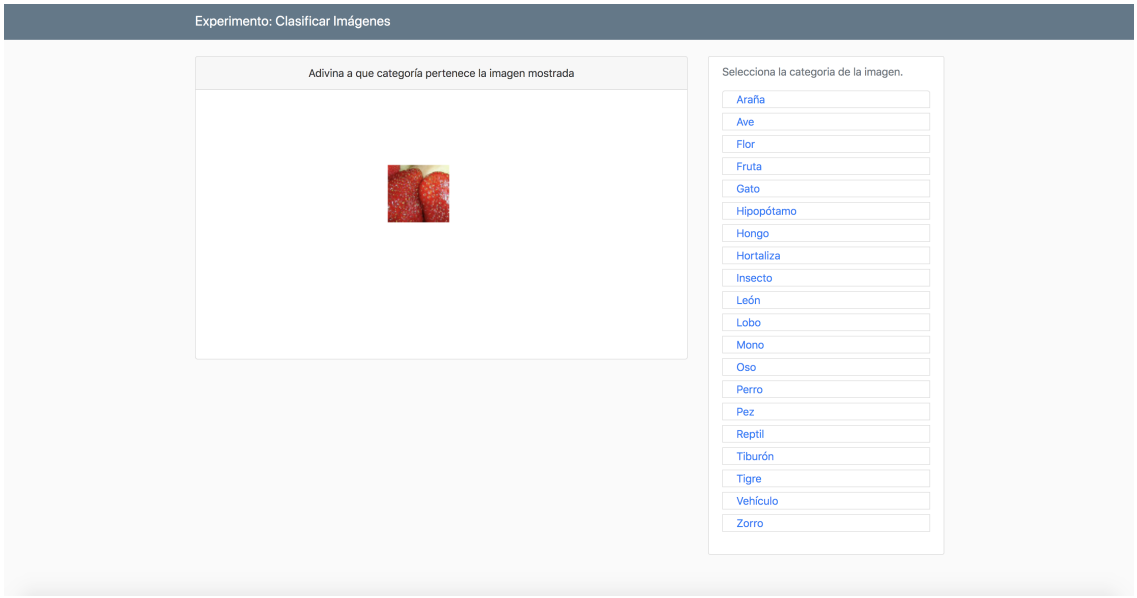

Figura 4.4: Interfaz de clasificación de imagen

Una vez seleccionada la categoría deseada, se le entrega al usuario retroalimentación, en la segunda vista, en base a si la categoría seleccionada fue correcta (Figura 4.5) o no (Figura 4.6), mostrándole la imagen original sin filtros, así como también un mensaje con la categoría correcta con fondo verde en caso correcto o rojo en caso incorrecto, además de darle la opción de continuar con otra imagen o simplemente finalizar el experimento. Adicionalmente se muestran los registros de la sesión, mostrándole su desempeño en forma de respuestas correctas e incorrectas y también el porcentaje de correctas versus las respuestas totales. Si bien darle feedaback al usuario no es necesario para los objetivos del experimento, la ludificación de la experiencia ayuda a despertar el interés por seguir, así como promueve que la persona trate de responder correctamente.

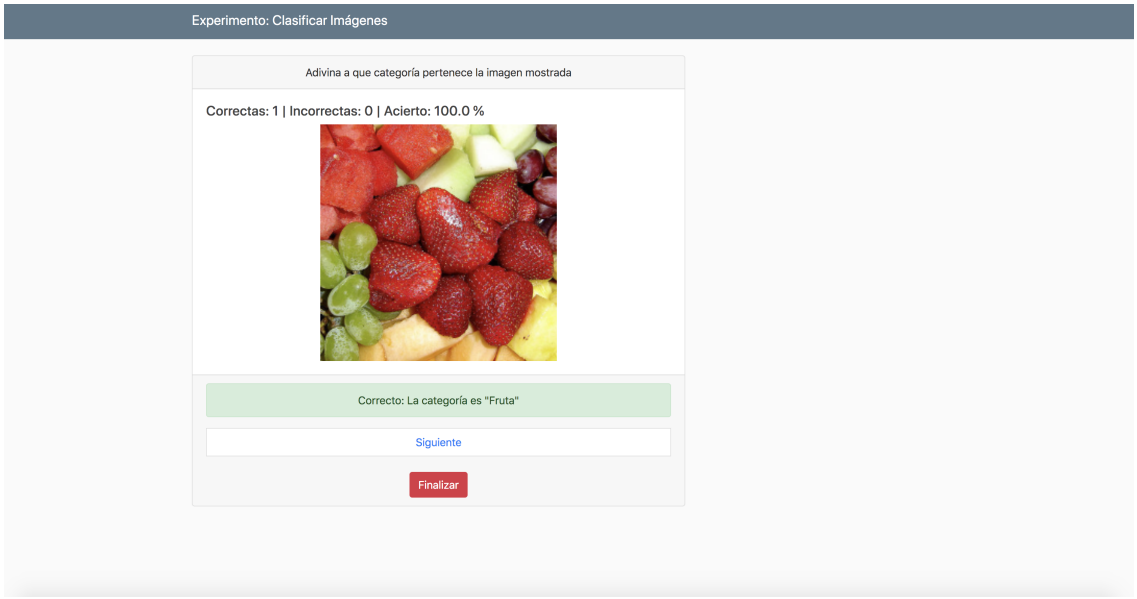

Figura 4.5: Interfaz respuesta correcta

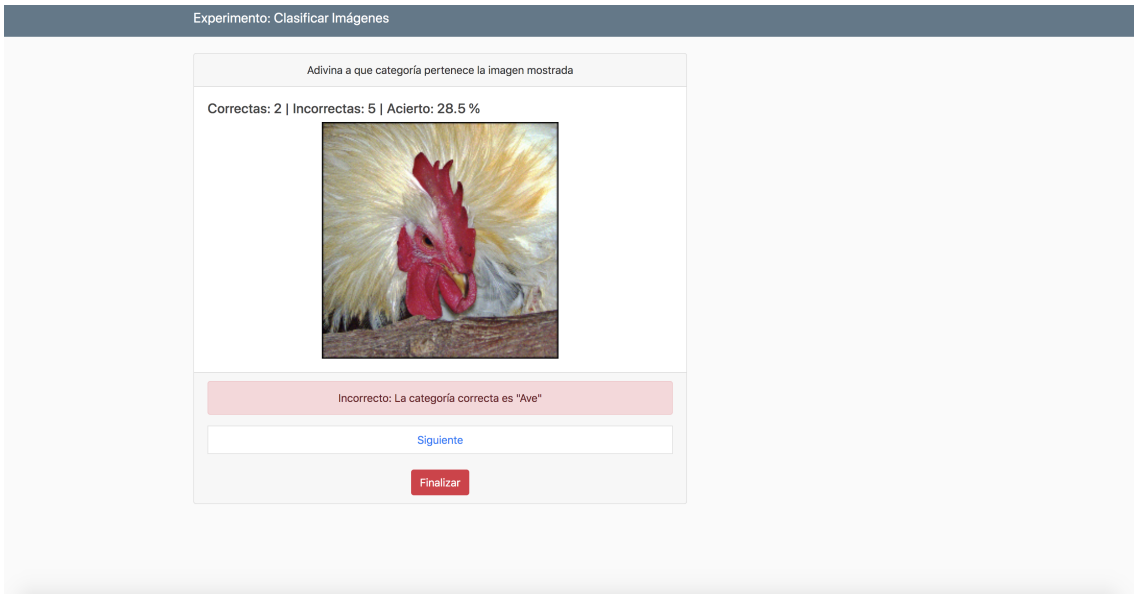

Figura 4.6: Interfaz respuesta incorrecta

4.2.5. Implementación

Para la implementación de la aplicación se decidió utilizar el framework Flask debido a su facilidad de implementación así como la librería Bootstrap para manejar el aspecto de la página. Flask [*] es un microframwork de desarrollo web para Python, entre sus características se encuentra su modularidad y extensibilidad; es ideal si se requiere hacer proyectos sencillos

[*] `http://flask.pocoo.org`

o proyectos unipersonales como es el caso de este trabajo. Bootstrap [**], en tanto, es uno de los framework open-source más populares para HTML, CSS y JavaScript, el cual provee herramientas de diseño para el desarrollo web.

Con el fin de poder manejar la información de usuario se utilizaron sesiones para mantener los registros del usuarios cuando estos navega entre páginas; esto es necesario dado que cada usuario puede identificar múltiples imágenes en una sola sesión y se necesita almacenar cada respuesta que el usuario registre bajo un mismo nombre. De esta forma cuando el usuario se conecta por primera a la aplicación se genera, de manera aleatoria, un id único para dicho usuario el cual servirá de nombre para el archivo donde se guardaran las respuestas obtenidas. Las respuestas se guardarán automáticamente en el archivo a medida que se vayan generando de manera que no se pierdan si por alguna razón el usuario se desconecta.

Para determinar qué imágenes se entregarán para clasificar, se armará una lista con todas las imágenes originales únicas que hay disponibles, posteriormente, al momento de seleccionar una imagen para se identificada en la aplicación, se eligiera una de la lista de imágenes disponibles al azar, así como también, de manera aleatoria, la versión de dicha imagen, de las 9 disponibles (cuatro filtros por cada red mas la versión original), que se mostrará. Posteriormente, la imagen previamente seleccionada sale de la lista de imágenes disponibles para esta sesión; esto permite que la misma imagen no se mostrada una segunda vez en cualquiera de sus versiones para evitar, de esta manera, que la persona la clasifique correctamente en esa segunda instancia.

Con respecto a la información obtenida del usuario, esta se va almacenando en un archivo JSON creado por cada usuario que participa, esto pues permite una mayor facilidad y flexibilidad para el manejo de los datos a obtener. JSON (*JavaScript Object Notation*) es un formato de texto liviano que permite guardar registros, así como parsearlos , de manera sencilla . En Python se tratan como diccionarios.

A continuación se muestra un ejemplo del archivo JSON que contiene la información de cada iteración de la prueba para un usuario en particular:


```
{"id": "2d61eef2-dd56-403f-8ced-a67fd70dd683",
"resultados": [
    {"tipo": "Tibur\u00f3n", "adivinanza": "Tibur\u00f3n",
    "imagen": "ILSVRC2012_val_00030156.png", "efecto": "colores",
    "red": "GoogLeNet"},
    {"tipo": "Tigre", "adivinanza": "Tigre",
    "imagen": "ILSVRC2012_val_00038124.png", "efecto": "minimo",
    "red": "Squeezenet"},
    {"tipo": "Oso", "adivinanza": "Oso",
    "imagen": "ILSVRC2012_val_00010281.png", "efecto": "resolucion",
    "red": "Squeezenet"}
]}
```


[**] http://getbootstrap.com

En el ejemplo de archivo mostrado se almacena, en primera instancia, el id único de la sesión, a modo de identificador del voluntario, luego, por cada imagen clasificada, se almacena la información descrita en la subsección *Información a Recopilar*; este ejemplo el archivo corresponde a un usuario que realizó tres clasificaciones. A modo de ejemplo, para comprender como se guarda la información, en la primera clasificación le fue presentada al usuario una imagen del tipo *Tiburón*, correspondiente a la generada por el filtro de disminución de colores al enfrentarse a la red GoogLeNet y cuya imagen original es ILSVRC2012_val_00030156.png; Ante esta situación el usuario selecciona correctamente la categoría *Tiburón* .

Finalmente para poder desplegar la página, se utilizarán los servidores de *pythonanywhere*, que corresponde a un webhosting optimizado para aplicaciones web desarrolladas en Python.

Capítulo 5

Resultados y Análisis

En este capitulo se presentarán y analizarán los resultados de las soluciones descritas en el capitulo anterior; en primera instancia, se revisarán los datos obtenidos de la generación de los ejemplos positivos mínimos para las redes SqueezeNet y GoogLeNet. Posteriormente se discutirán los resultados obtenidos de la comparación entre las redes neuronales y los humanos.

5.1. Caracterización de las Redes Neuronales al Enfrentarse a la Pérdida de Información de las Entradas

Con el fin de ejemplificar las imágenes positivas mínimas obtenidas se presentarán todas las imágenes obtenidas a partir de una imagen en particular, es decir por cada red neuronal y por cada filtro:

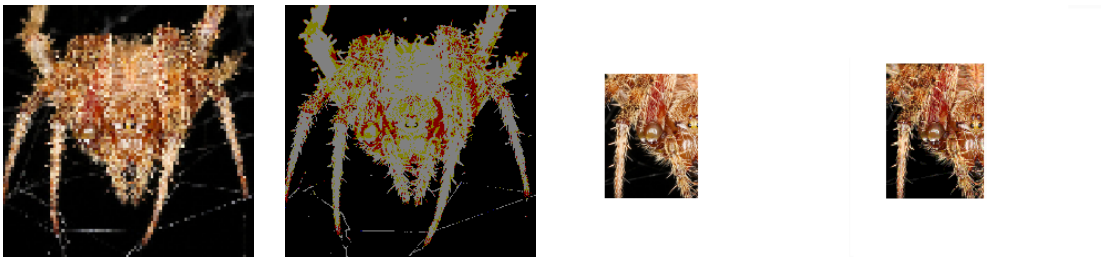

Figura 5.1: Ejemplos positivos mínimos para GoogLeNet para los filtros (de derecha a izquierda) disminución de la resolución real, disminución de colores, recorte y filtros agregados.

Figura 5.2: Ejemplos positivos mínimos para SqueezeNet para los filtros (de derecha a izquierda) disminución de la resolución real, disminución de colores, recorte y filtros agregados.

Como puede verse, en las Figuras 5.1 y 5.2, GoogLeNet genera ejemplos positivos mínimos con una mayor intensidad de los filtros,es decir, con menor tamaño, que SqueezeNet; de igual forma es esperable que los ejemplos positivos mínimos obtenidos de la agregación de los filtros tengan una zona recortada levemente mayor, esto pues también se ponderan los filtros que disminuyen resolución real y colores disponibles, por lo que las redes neuronales deben depender de otras características para tomar una decisión correcta.

Con respecto a la disminución de confianza, de la categoría correcta, los casos de filtros unidimensionales se ejemplifican en la Figura 5.3, (la totalidad de los gráficos por cada clase están en el apéndice pero todos presentan características similares a la figura mostrada); puede notarse que, para la disminución de colores, la confianza es prácticamente inelástica hasta encontrarse con valores menores al 10 % de colores disponibles, donde disminuye drásticamente; por otra parte la disminución de la resolución real de las imágenes provoca un efecto inmediato en la confianza de la red disminuyendo paulatinamente hasta llegar a valores cercanos a cero. Estas diferencias se pueden dar, en parte, debido a que las redes neuronales entrenadas por ImageNet son parciales hacia las texturas [4], por tanto estas no se ven afectadas con la disminución de colores sino hasta que la cantidad de colores disponibles sea muy poca; en cambio la disminución de calidad tiene un efecto directo sobre las texturas al retornar imágenes cada vez más pixeleadas.

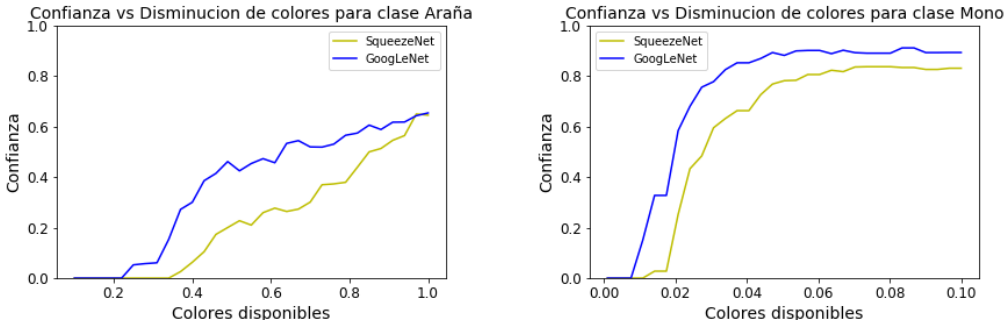

Figura 5.3: Confianza promedio por clase para filtro disminución de colores (derecha) y disminución de la resolución real (izquierda).

Con respecto a los filtros multidimensionales; para el agregado de las clases mostrados en la Figura 5.4, es posible notar que GoogLeNet nuevamente permite minimizar mucho más las imágenes antes que dejen de ser clasificadas correctamente; también es importante notar,

en el gráfico del filtros agregados, que para el filtro de disminución de color, los valores de intensidad máximos parecen estar concentrados en torno a los mismos valores; esto se condice con lo establecido anteriormente que la confianza disminuye de manera abrupta pasado un cierto umbral.

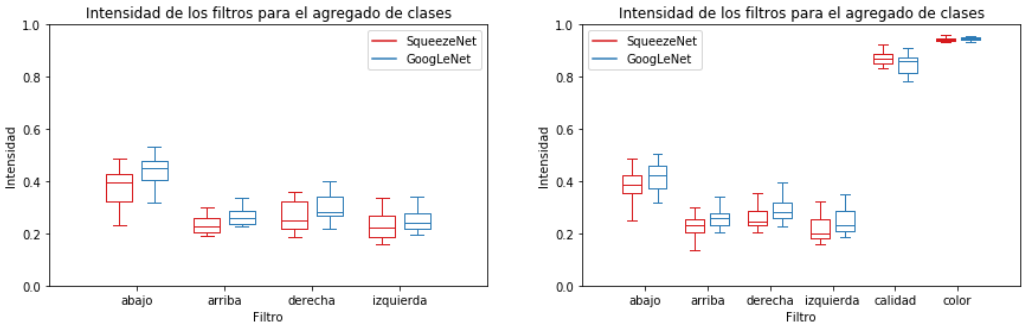

Figura 5.4: Intensidad promedio agregada por filtro para recorte (izquierda) y todos los filtros juntos (derecha).

Con respecto a la minimización del tamaño de las imágenes, en el caso del filtro agregado, mostrado en la Figura 5.5, se sigue la tónica mostrada hasta ahora que indica que GoogLeNet logra mejores resultados, en particular para las clases tiburón y araña, siendo la única excepción la clase flor para la cual SqueezeNet se desempeña un poco mejor.

Con respecto a cuánto pueden ser disminuidas las imágenes, usando el filtro agregado, se tienen clases que pueden disminuir hasta un 3 % del tamaño original, en bytes, como el caso de perro y zorro, es decir, la información necesaria para su clasificación es muy poca.

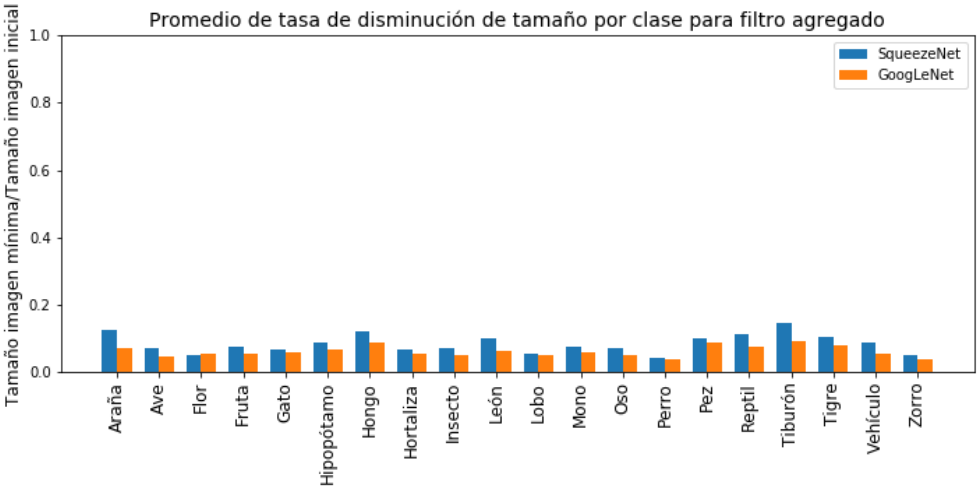

Figura 5.5: Tamaño promedio imágenes mínimas por clase para filtro agregado.

Si observamos los casos de minimización de tamaño por clase según los filtros de colores, resolución y recorte, observados en las Figuras 5.6, 5.7 y 5.8, se puede notar, en primera instancia, que todas las clases obtienen tamaños comprimidos promedio mayores que para

el caso de filtros agregados de la Figura 5.5; esto es esperable dado que el filtro agregado puede disminuir la información de las imágenes en varias dimensiones. Con respecto al filtro recorte puede verse claramente que es el que presenta una mayor diferencia entre las clases que más pueden comprimirse y las que menos, en particular, para la red SqueezeNet; de manera similar el filtro recorte también presenta la mayor diferencia entre la compresión de las redes utilizadas, como es caso extremo de la clase araña donde GoogLeNet casi dobla la tasa de compresión presentada por SqueezeNet. La Figura 5.6 muestra que el filtro de colores es el que, en promedio, presenta una mejor compresión para las dos redes utilizadas, siendo también la que menos diferencia presenta entre ambas. Finalmente el filtro resolución es el que menos compresión, en promedio, permite.

Con respecto a la compresión por clases, mostradas en las Figuras 5.6, 5.7 y 5.8, se puede notar que, dado un filtro, tanto GoogLeNet como SqueezeNet presentan un comportamiento similar, es decir, aquellas clases que más se pueden comprimir son las mismas para ambas redes, así como también lo son las que menos; esto puede deberse, principalmente, a que las redes buscan los mismos elementos y solo difieren en su capacidad para encontrarlos. Por otro lado se puede observar que para los distintos filtros, las clases más y menos comprimidas varían, incluso dándose el caso extremo de la clase zorro, la cual para el filtro recorte es la clase que más se puede comprimir y para el filtro SqueezeNet es una de las que menos; esto puede deberse a que las distintas características relevantes para la correcta clasificación de las imágenes son más o menos propensas a perderse dependiendo del filtro; esto también se condice con el hecho que las redes neuronales usan distintas características de las imágenes para clasificar.

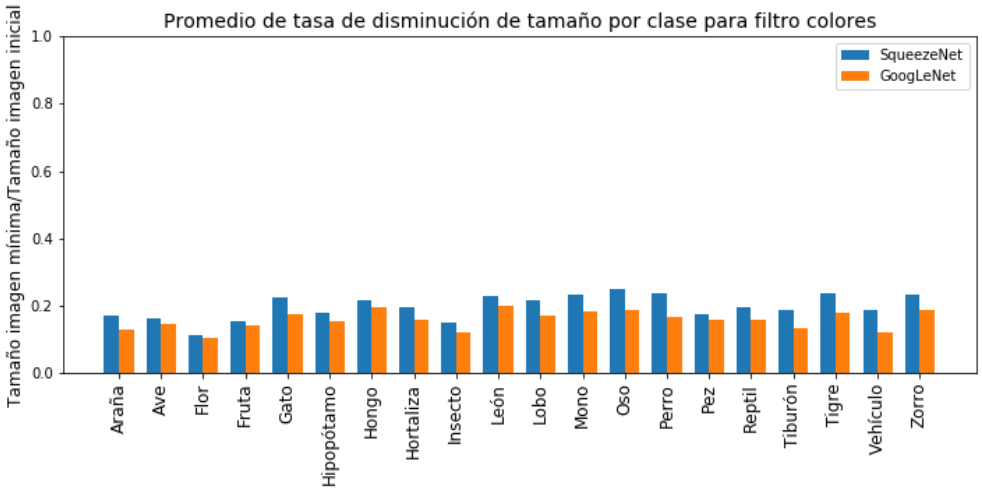

Figura 5.6: Tamaño promedio imágenes mínimas por clase para filtro color.

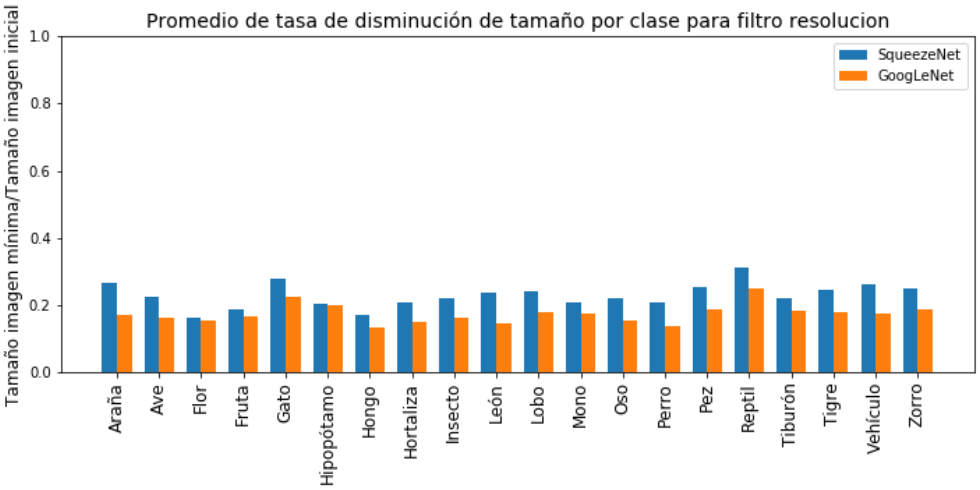

Figura 5.7: Tamaño promedio imágenes mínimas por clase para filtro resolución.

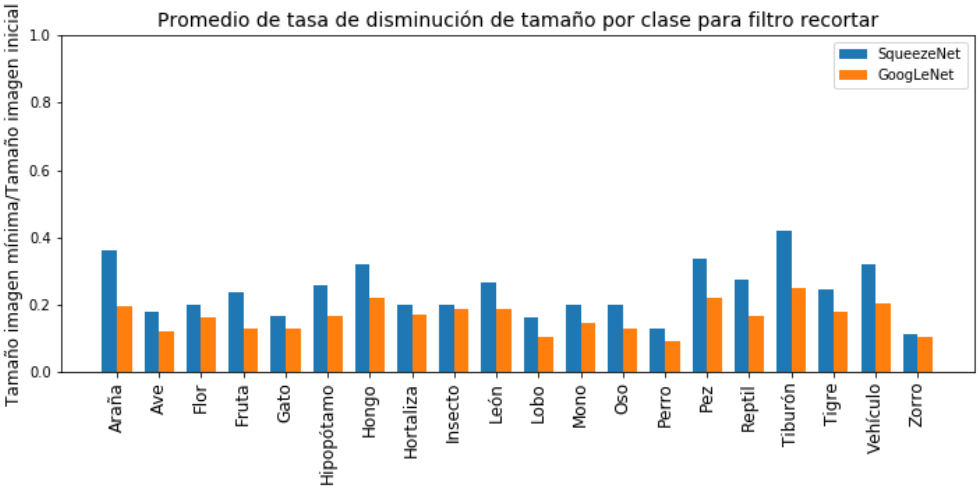

Figura 5.8: Tamaño promedio imágenes mínimas por clase para filtro recorte.

De esta sección se desprende que una de las mejores formas de reducir el tamaño de las imágenes, sin perder confianza, es bajar la cantidad de colores disponibles; esto es particularmente importante si las entradas de la red neuronal son recibidas de manera remota y el tamaño de los archivos es relevante. Por otro lado es necesario comprender bien el contexto de uso de las redes neuronales, dado que puede ocurrir que una red inicialmente sea capaz de reconocer muy bien una clase pero al ser usada en la vida real esta sujeta a qué tan bien haya sido entrenada para enfrentar los distintos ruidos que se puedan presentar y para esto es clave poder entender qué tan robusta es la red ante los distintas formas de estos ruidos.

5.2. Comparación de redes neuronales con humanos

En esta sección se presentarán los resultados de la prueba de clasificación por humanos de las imágenes mínimas generadas en la etapa anterior.

5.2.1. Datos obtenidos

En la primera fase, donde se realizo la prueba en un espacio controlado, se obtuvo un promedio de respuestas correctas del 0.875 con una desviación estándar del 0.061, lo que significa que para la segunda fase, cuando se realizaron pruebas en línea, se consideraron solo aquellas personas que tuviesen una tasa de respuestas correctas mínima de 0.753, esta se obtiene de calcular una distancia de dos desviaciones estándar del promedio del grupo de control, para garantizar la calidad de los registros, quedando caracterizados los datos finales de la siguiente manera:

- Personas totales : 423
- Personas con resultados válidos : 352
- Total de clasificaciones válidas : 8637
- Clasificaciones promedio por persona : 24.5
- Máximo de clasificaciones por persona : 175
- Mínimo de clasificaciones por persona : 1

5.2.2. Comparación entre imágenes originales y modificadas según clase.

Según puede verse en el la Figura 5.9, tal como se podría esperar, el desempeño de las personas disminuye al verse enfrentado a los ejemplos positivos mínimos de las redes neuronales, en comparación con las imágenes originales, sin embargo existe una gran diferencia según la categoría que se observa; por un lado se tienen categorías que prácticamente no varían, como lo son el caso de vehículo, tigre e hipopótamo que además cumplen con la particularidad que son aquellas categorías que en un principio tuvieron una mayor tasa de clasificaciones correctas; de igual forma las categorías flor, hongo, zorro y lobo son las que más disminuyeron su tasa de respuestas correctas; esto es esperable dado que justamente estas son las categorías para las que hay otras que a simple vista pueden parecer similares; de igual forma este argumento se puede trasladar al caso de las categorías mejor clasificadas, salvo para tigre para el cual hay otras clases de felinos.

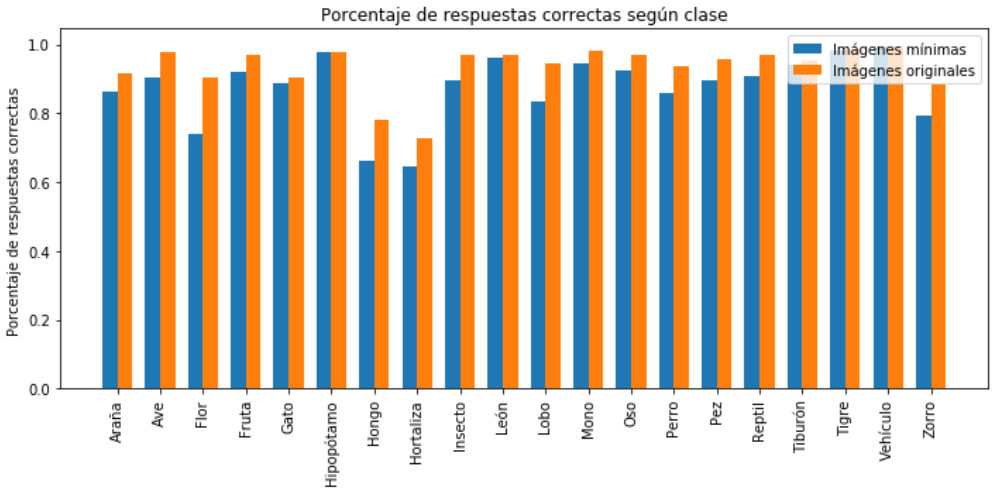

Figura 5.9: Comparación del desempeño humano para imágenes sin filtro y con filtro.

5.2.3. Desempeño humano ante los distintos filtros

	Correctas	Incorrectas	Porcentaje Correctas
Sin Filtro	880	78	91.8 %
Disminución de Colores	1745	159	91.6 %
Recorte	1612	302	84.2 %
Disminución de Resolución Real	1741	160	91.5 %
Agregado de Filtros	1623	337	82.8 %

Tabla 5.1: Desempeño de las personas frente a imágenes positivas mínimas según filtro

Según puede verse en la Tabla 5.1, pareciera no haber diferencia en la capacidad de las personas para clasificar ante la falta de filtros y los filtros que disminuyen calidad de la imagen y la cantidad de colores disponibles; esto significaría que los seres humanos serían mucho menos sensibles que las redes neuronales actuales frente a estos cambios.

Con respecto al filtro recorte y el agregado de filtros, se muestra en la misma tabla que afectan la capacidad de las personas para evaluar correctamente las distintas imágenes; esto indicaría que, para estos filtros, los humanos son más sensibles, aunque de igual forma siguen siendo menos sensibles que las redes neuronales.

Con respecto al filtro recorte, también presente en el agregado de filtros, disminuye de manera muy distinta la información a los filtros del punto anterior, dado que quita información del contexto de la imagen y en muchos casos hace que la imagen presente una parte específica del objeto muy distinta a las características que los seres humanos asocian a tal clase; este punto se ve aún más confirmado al notarse que el agregado de filtros, tal como se podría esperar, es el que presenta mayores dificultades para las personas, lo cual lleva a pensar que la perdida de información por medio de los filtros de colores y calidad no son efectivos mientras existan

otros elementos de la imagen que puedan suplir dicha falta pero que de igual manera se afectan ante un caso extremo como en el caso de las imágenes mostradas en la Figura 5.10, que corresponden a las imágenes que fueron clasificadas incorrectamente una mayor cantidad de veces por humanos, mas no por las redes neuronales. Es necesario también establecer que el filtro recorte, y en consecuencia el filtro agregado, es el que quita más información relevante a la forma de los objetos que se quieren clasificar, versus color y resolución que más afectan la textura de la imagen, entonces, el hecho que los humanos tengan precisamente mayores problemas con aquellos filtros que afectan la información de las formas apoya la hipótesis que indica que los humanos son mas sensibles a la perdida de esta información.

Figura 5.10: Imágenes clasificadas incorrectamente más veces. A la izquierda una imagen de la categoría hortaliza/vegetal y a la derecha una imagen de la categoría hongo

Es también necesario notar que si bien los distintos filtros podrían afectar el desempeño humano, en ningún caso existe una supremacía por parte de la red neuronal, esto pues es común que en un grupo de personas, hayan distintos grados de familiaridad con ciertas categorías y el objetivo es lograr reemplazar clasificadores humanos expertos; para que esto ocurriera debiese haber un porcentaje cercano al 10 % de la población que pueda contestar correctamente (considerando que por azar debiese haber al menos un 5 % de respuestas correctas).

5.2.4. Desempeño humano ante las imágenes mínimas de distintas redes neuronales

	Correctas	Incorrectas	Porcentaje Correctas
GoogLeNet	3412	420	82.8 %
SqueezeNet	3313	534	89.0 %

Tabla 5.2: Desempeño de las personas frente a imágenes positivas mínimas de cada red neuronal

Según puede verse en la Tabla 5.2 la diferencia entre los porcentajes de respuestas correctas es significativa , considerando una deviacion estándar de 0.096 para el caso SqueezeNet y 0.104 para GoogLenet; se obtiene un p-valor menor a 0.001, lo que indica que existe una diferencia real entre la habilidad de una persona para clasificar ejemplos positivos mínimos de una red versus la otra; esto se condice con lo esperado, considerando que GoogLeNet es una red que genera ejemplos positivas mínimos de menor tamaño que SqueezeNet.

5.2.5. Caracterización de los errores humanos

Con respecto a las clases por la que son confundidas las imágenes mostradas, que aparecen en el mapa de calor de la Figura 5.11, puede notarse claramente que aquellas clases que poseen similitudes tienden a ser confundidas entre si como lo son el caso de una araña y un insecto o una hortaliza al ser confundida por una fruta (aunque a veces sea por diferencias en la definición personal de cada una).

Es interesante notar que muchas veces aquellas confusiones no ocurren hacia ambos lados con la misma ocurrencia como el caso del perro con el lobo; imágenes de este último tienden a ser clasificadas como el primero en mayor medida que la reciproca; esto puede deberse a la mayor familiaridad con los primeros o tal vez que haya menos tipos de perro que tienen características de lobo que tipos de lobo que tienen características de perro.

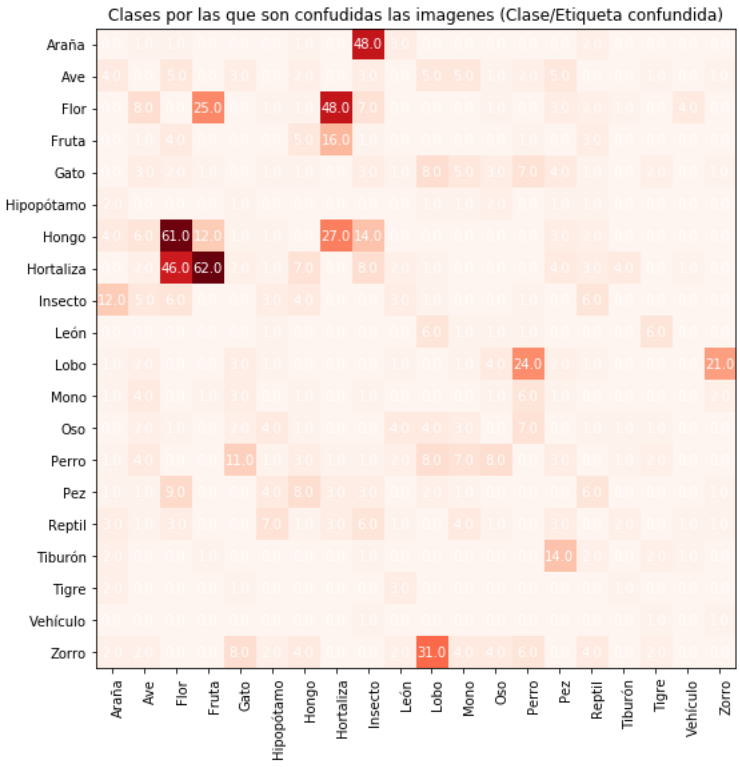

Figura 5.11: Clases por las que la clase correcta es confundida por los humanos

Capítulo 6

Conclusión

En esta memoria se han presentado una manera de generar imágenes mínimas, para distintos tipos de filtros, que disminuyen la información de las imágenes de entrada para redes neuronales, y como estos métodos son válido para lograr caracterizar los elementos mas relevantes que consideran las redes neuronales al momento de poder clasificar correctamente las distintas categorías para las que fueron entrenadas.

Se pudo mostrar que las redes neuronales utilizadas no son sensibles a variables como la cantidad de colores disponibles hasta que se llega a un umbral tras el cual la confianza disminuye drásticamente. Por otro lado también se mostró que las redes neuronales pueden clasificar correctamente aún cuando se aísla a los sujetos a clasificar del contexto; esto es extremadamente importante si se quieren utilizar en diferentes situaciones. Otros filtros, como la disminución de resolución de las imágenes, afectan de forma inmediata la confianza de las redes; esto es muy importante a tener en cuenta puesto que cuando se diseñen sistemas de reconocimiento de imágenes, como el de un vehículo autónomo, hay que sopesar el trade-off que presenta utilizar imágenes de mayor resolución; estas pueden ser mejores para detectar objetos lejanos, pues son representados por más píxeles, pero a su vez es mas costoso el procesar, transmitir y almacenar una imagen de mayor resolución.

Con respecto al experimento con personas, se desprende que los humanos son menos sensibles a los filtros utilizados en este experimento, en parte porque compensan la falta de distinta información con otra información disponible de mejor manera que las redes neuronales. A su vez pareciera ser que el contexto juega un rol fundamental para la clasificación para el caso humano; con contexto no solo se refiere al fondo de la imagen si no que también, por ejemplo, cuando se presenta una oreja de perro sin perro.

Uno de los principales miedos que había con realizar este experimento en línea era que la calidad de los resultados fuera deficiente; esto resultó ser infundado, pues gran parte los participantes respondieron a conciencia, obteniéndose resultados muy similares al grupo de control, dado por supuesto que existan parámetros de aceptación para las respuestas. De esta forma, el extender los experimentos a un formato en línea, permitió obtener muchos más participantes que si se hubiesen hecho solamente sesiones presenciales, además de ser mucho

menos costoso en recursos considerando toda la logística que se requiere para las sesiones controladas.

Con respecto al trabajo futuro sería bueno replicar la caracterización en sistemas de clasificación binarios, como puede ser redes neuronales que identifican si hay presencia de tumores o no; de esta manera se podría identificar si hay diferencias considerables en la forma en que una red con múltiples categorías toman decisiones. Otro trabajo interesante sería generar imágenes mínimas de personas para poder comparar directamente si la red es más o menos sensible que los humanos, y no solo al revés como se hizo en este trabajo; generar los ejemplos positivos mínimos para humanos también sería mucho más replicable en trabajos futuros, pues solo se genera una vez para cada filtro y estos pueden ser probados fácilmente en nuevas redes.

Bibliografía

[1] Diego Ardila, Atilla P. Kiraly, Sujeeth Bharadwaj, Bokyung Choi, Joshua J. Reicher, Lily Peng, Daniel Tse, Mozziyar Etemadi, Wenxing Ye, Greg Corrado, David P. Naidich, and Shravya Shetty. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature Medicine*, 25:1, 06 2019.

[2] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pages 3–14, New York, NY, USA, 2017. ACM.

[3] Samuel F. Dodge and Lina J. Karam. A study and comparison of human and deep learning recognition performance under visual distortions. *CoRR*, abs/1705.02498, 2017.

[4] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *CoRR*, abs/1811.12231, 2018.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[6] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *CoRR*, abs/1903.12261, 2019.

[7] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016.

[8] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014.

[9] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[10] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[11] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155, 1964.

[12] Sanjana Srivastava, Guy Ben-Yosef, and Xavier Boix. Minimal images in deep neural networks: Fragile object recognition in natural images. *CoRR*, abs/1902.03227, 2019.

[13] Sining Sun, Ching-Feng Yeh, Mari Ostendorf, Mei-Yuh Hwang, and Lei Xie. Training augmentation with adversarial examples for robust speech recognition. *Interspeech 2018*, Sep 2018.

[14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[15] Shimon Ullman, Liav Assif, Ethan Fetaya, and Daniel Harari. Atoms of recognition in human and computer vision. *Proceedings of the National Academy of Sciences*, 113(10):2744–2749, 2016.

[16] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *CoRR*, abs/1704.01155, 2017.

Apéndice

A.1. Gráficos de Confianza Promedio de Clasificación por Clase ante la Aplicación del Filtro de Disminución de Color

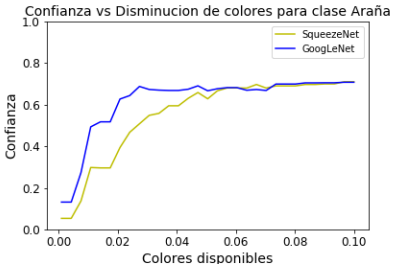

Figura A.1: Confianza para la clase Araña para filtro de color.

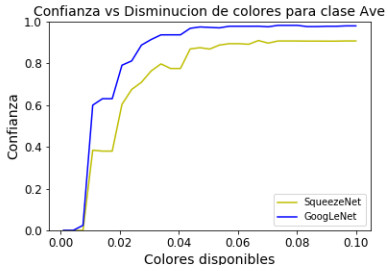

Figura A.2: Confianza para la clase Ave para filtro de color.

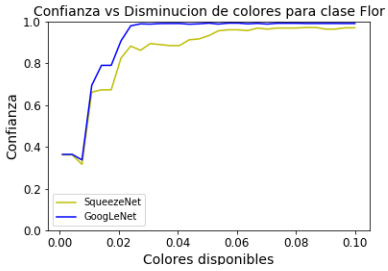

Figura A.3: Confianza para la clase Flor para filtro de color.

Figura A.4: Confianza para la clase Fruta para filtro de color.

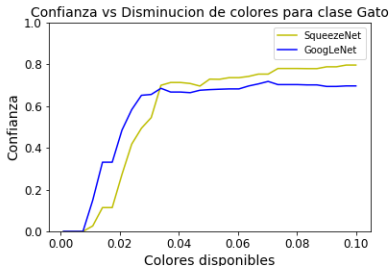

Figura A.5: Confianza para la clase Gato para filtro de color.

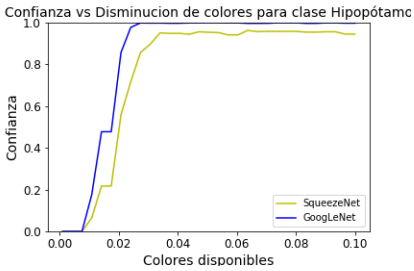

Figura A.6: Confianza para la clase Hipopótamo para filtro de color.

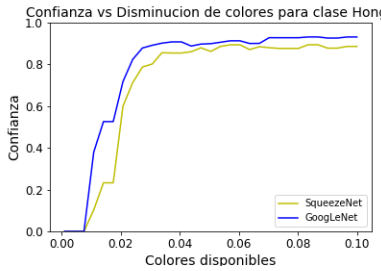

Figura A.7: Confianza para la clase Hongo para filtro de color.

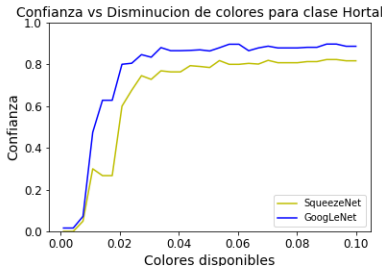

Figura A.8: Confianza para la clase Hortaliza/Vegetal para filtro de color.

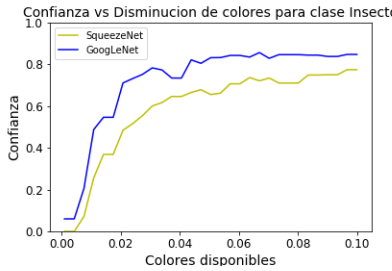

Figura A.9: Confianza para la clase Insecto para filtro de color.

Figura A.10: Confianza para la clase León para filtro de color.

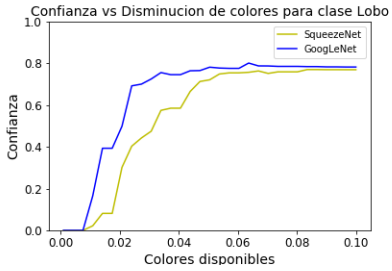

Figura A.11: Confianza para la clase Lobo para filtro de color.

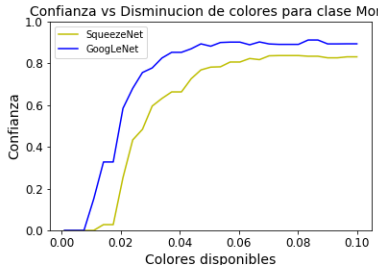

Figura A.12: Confianza para la clase Mono para filtro de color.

Figura A.13: Confianza para la clase Oso para filtro de color.

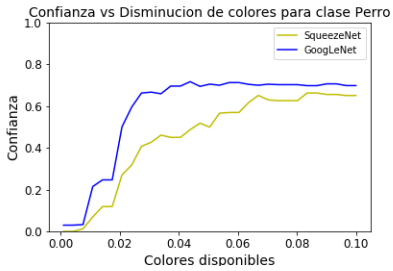

Figura A.14: Confianza para la clase Perro para filtro de color.

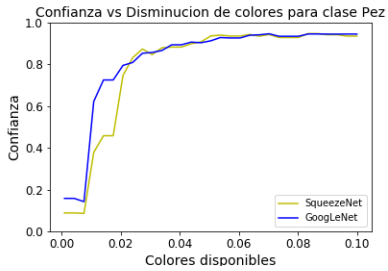

Figura A.15: Confianza para la clase Pez para filtro de color.

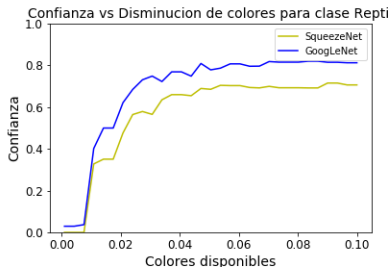

Figura A.16: Confianza para la clase Reptil para filtro de color.

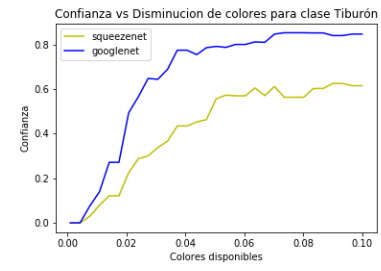

Figura A.17: Confianza para la clase Tiburón para filtro de color.

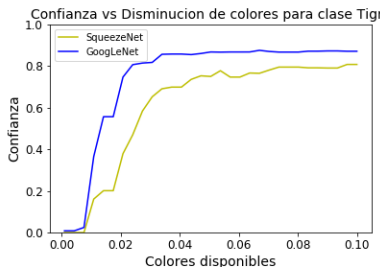

Figura A.18: Confianza para la clase Tigre para filtro de color.

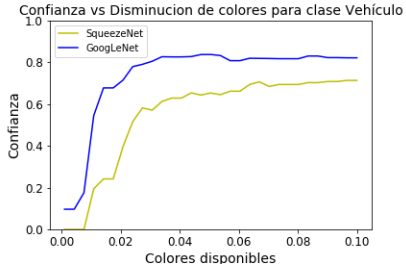

Figura A.19: Confianza para la clase Vehículo para filtro de color.

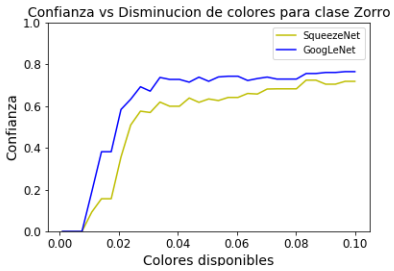

Figura A.20: Confianza para la clase Zorro para filtro de color.

A.2. Gráficos de Confianza Promedio de Clasificación por Clase ante la Aplicación del Filtro de Disminución de Resolución

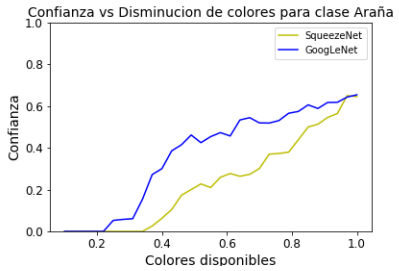

Figura A.21: Confianza para la clase Araña para filtro de resolución.

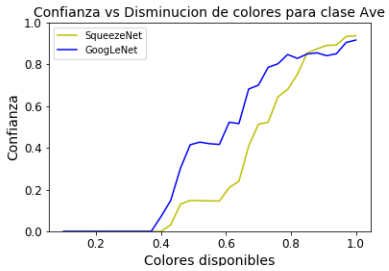

Figura A.22: Confianza para la clase Ave para filtro de resolución.

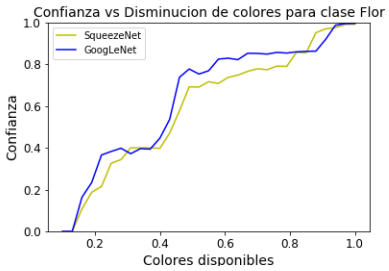

Figura A.23: Confianza para la clase Flor para filtro de resolución.

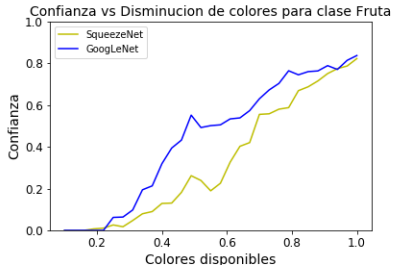

Figura A.24: Confianza para la clase Fruta para filtro de resolución.

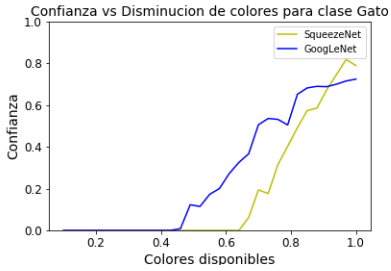

Figura A.25: Confianza para la clase Gato para filtro de resolución.

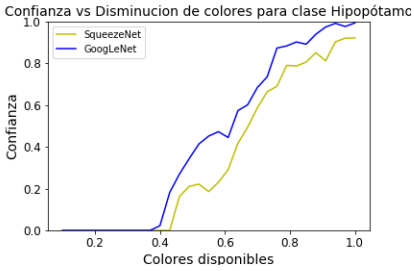

Figura A.26: Confianza para la clase Hipopótamo para filtro de resolución.

Figura A.27: Confianza para la clase Hongo para filtro de resolución.

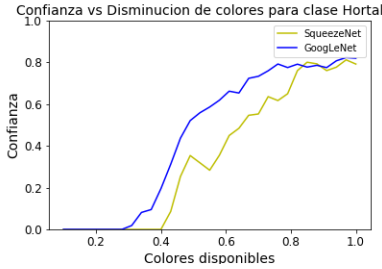

Figura A.28: Confianza para la clase Hortaliza/Vegetal para filtro de resolución.

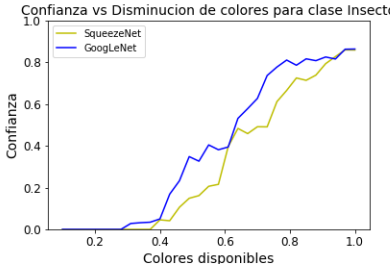

Figura A.29: Confianza para la clase Insecto para filtro de resolución.

Figura A.30: Confianza para la clase León para filtro de resolución.

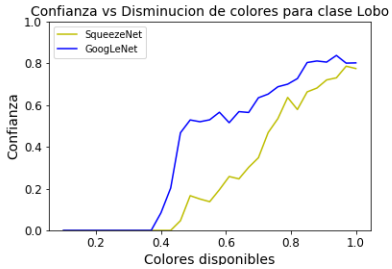

Figura A.31: Confianza para la clase Lobo para filtro de resolución.

Figura A.32: Confianza para la clase Mono para filtro de resolución.

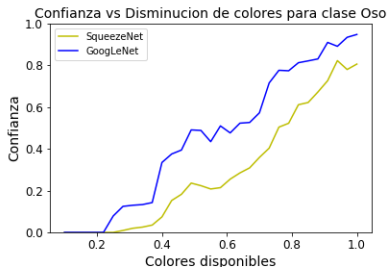

Figura A.33: Confianza para la clase Oso para filtro de resolución.

Figura A.34: Confianza para la clase Perro para filtro de resolución.

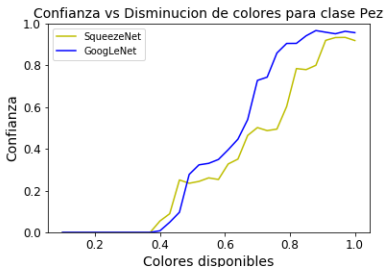

Figura A.35: Confianza para la clase Pez para filtro de resolución.

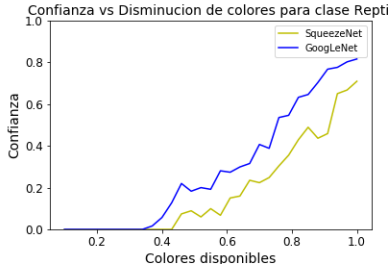

Figura A.36: Confianza para la clase Reptil para filtro de resolución.

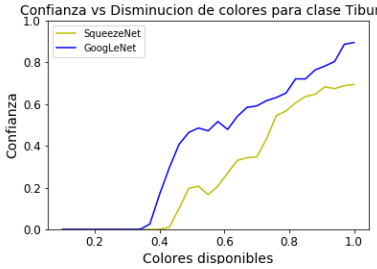

Figura A.37: Confianza para la clase Tiburón para filtro de resolución.

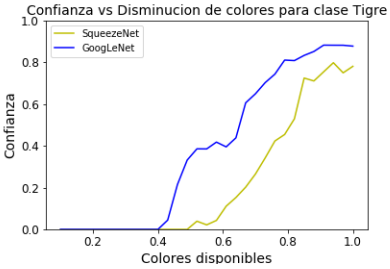

Figura A.38: Confianza para la clase Tigre para filtro de resolución.

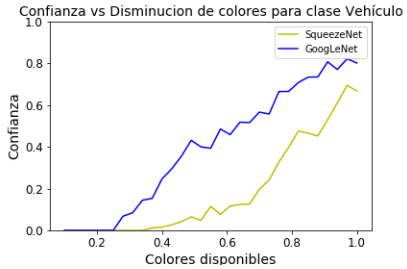

Figura A.39: Confianza para la clase Vehículo para filtro de resolución.

Figura A.40: Confianza para la clase Zorro para filtro de resolución.