

RESNET50 CON MÓDULOS DE ATENCIÓN SE – CBAM

Universidad Autónoma de Occidente
Especialización en Inteligencia Artificial

Deep Learning Avanzado
Docente: PhD. Juan Carlos Perafan

Alba Ramirez, 2216260 – Carlos Arbey Mejia, 2210549 - Andres Felipe Guerra, 2211058 – Milton Guarin, 2210702

Abstract-- Attention modules are used in Deep Learning, to make CNNs learn and focus on the most important information of a feature map, with the aim of improving the classification of a class in an image, in object detection applications.

In the present work, two attention mechanisms will be implemented (Squeeze-Excitation and CBAM) modifying a ResNet50 architecture and validating the performance of the CIFAR10 image dataset. The result and analysis of each model will be presented to validate the one with the best performance.

Keywords: Deep Learning, CNN (Convolutional Neural Networks), SE(Squeeze-Excitation), CBAM (Convolutional Attention Module), RESNET50, CIFRARIO

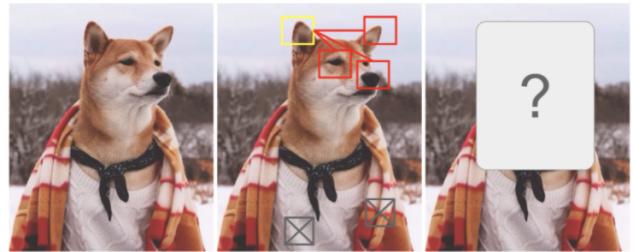


Fig. 1. A Shiba Inu in a men's outfit.

La atención visual humana nos permite enfocarnos en una determinada región (p.e. mirar la oreja puntiaguda del perro en el recuadro amarillo de la fig.1) mientras percibimos el resto de imagen en baja resolución.

En visión computacional, cuando entrenamos un modelo, queremos que el modelo pueda enfocarse en partes importantes de la imagen y una forma de lograrlo es utilizar los mecanismos de atención.

Al introducir estos mecanismos de atención en modelos con redes neuronales convolucionales (CNN's) tales como RESNET50, VGGNet, etc. se ha logrado mejorar considerablemente el desempeño de los modelos en tareas de clasificación y detección de objetos en imágenes, tal como lo demostraron los autores de los artículos: “Learn to Pay Attention” [1], “Squeeze-and-Excitation Network” [2] y “CBAM: Convolutional Block Attention Module” [3], los cuales se utilizarán en el presente trabajo realizando un comparativo de su desempeño.

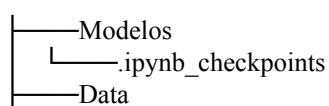
I. Introducción

Los mecanismos de atención surgieron como una mejora para el procesamiento de lenguaje natural (NLP) en redes neuronales. Posteriormente, estos mecanismos o sus variantes se empezaron a utilizar en otras aplicaciones, incluida la visión por computadora, el procesamiento del habla, etc. para mejorar el desempeño de las CNN's en tareas de clasificación de gran escala.

Estos mecanismos permiten a la red a poner atención dinámicamente a secciones de la entrada que son relevantes para la salida, similar a como los humanos nos concentramos en palabras específicas de una oración u objetos en una imagen.

II. Implementación

Para llevar a la implementación del proyecto, inicialmente se creó un repositorio en Google Drive donde se almacenó la siguiente información:



```

└── .ipynb_checkpoints
    └── img

```

Data: Se almacenaron los archivos Json con los valores de resultados de los diferentes entrenamientos de los modelos, aquí se almacenó la información de **loss** y **accuracy** de las últimas 10 épocas tanto para el dataset de entrenamiento como el de validación.

Modelos: Se almacenaron los archivos h5 de los de los modelos entrenados, almacenando la arquitectura completa con sus pesos en un archivo y en otro solo los pesos por si son necesarios.

img: Se almacena los archivos png y jpg de las imágenes de validación final de los modelos y sus diferentes resultados con la aplicación del heatmap respectivo, para así poder visualizar la atención de la última capa de activación de cada modelo.

En un cuaderno de Google Colab (*ResNet50_SE_y_CBAM.ipynb*) se implementaron los tres modelos a comparar:

- ResNet50 Original (Sin módulo de atención)
- ResNet50_SE con (Squeeze-and-Excitation)
- ResNet50_CBAM con (Convolutional Block Attention Module)

2.1 ResNet50 Original

ResNet son las siglas de Residual Network. Es una red neuronal innovadora que fue presentada por primera vez por Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun en su artículo de investigación de visión por computadora de 2015 titulado “Aprendizaje Residual Profundo para el Reconocimiento de Imágenes”.

Este modelo fue inmensamente exitoso, como se puede comprobar por el hecho de que su conjunto ganó la primera posición en la competencia de clasificación ILSVRC 2015 con un error de solo 3.57%. Además, también ocupó el primer lugar en la detección de ImageNet, la localización de ImageNet, la detección de COCO y la segmentación de COCO en las competencias ILSVRC y COCO de 2015 [5].

ResNet50 se utiliza para indicar la variante que puede funcionar con 50 capas de red neuronal (ver fig 2).

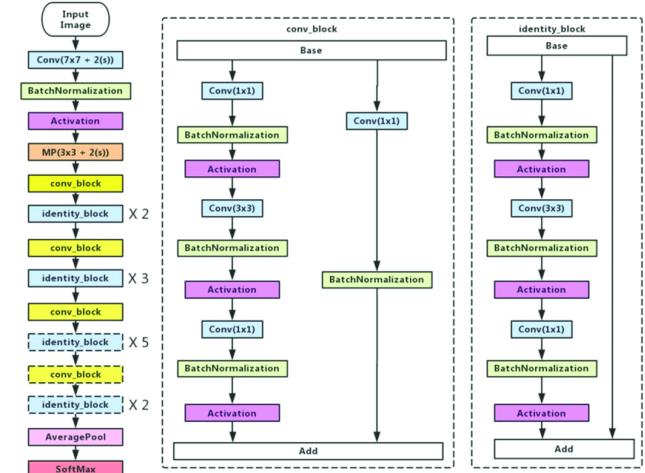


Fig. 2. Arquitectura ResNet50

Para el trabajo se implementó la arquitectura ResNet50 original en el cuaderno de Google Colab y se utilizaron los siguientes parámetros de entrenamiento, como se observa en la tabla 1:

Modelo	ResNet50
Input	32x32x3
Parametros	23.581.642
Epochs	20
Loss	categorical_crossentropy
Optimizer	Adam

Tabla 1. Parámetros de entrenamiento ResNet50

2.2 ResNet50 con módulo de atención SE (Squeeze-and-Excitation)

El mecanismo de atención Squeeze-and-Excitation fue introducido en el año 2018 por Hu et al. en su artículo “Squeeze-and-Excitation Networks” en CVPR 2018 con una versión de revista en TPAMI [7].

Se utilizó en la competencia ImageNet de este año y ayudaron a mejorar el resultado del año pasado en un 25%. Además de este enorme aumento de rendimiento, se pueden agregar fácilmente a las arquitecturas existentes (ver fig 3.)

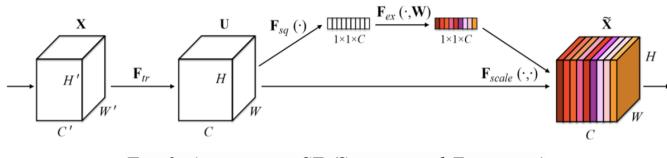


Fig. 3. Arquitectura SE (Squeeze-and-Excitation)

Es un mecanismo de atención que se basa en introducir un bloque de construcción para las CNN que mejora la interdependencias de canales casi sin costo computacional.

En la arquitectura ResNet50 se introdujeron los bloques de Squeeze y Excitation:

La primera es la operación Squeeze, que realiza la compresión de características a lo largo de la dimensión espacial convirtiendo cada canal de características bidimensionales en un número real.

Este número real tiene un campo receptivo global hasta cierto punto, y la dimensión de salida está en línea con el número de canal de características de entrada. Partido. Caracteriza la distribución global de respuestas en el canal de características y hace que las capas cercanas a la entrada también obtengan un campo receptivo global, que es muy útil en muchas tareas.

La segunda es la operación Excitation, que es un mecanismo similar a las puertas en las redes neuronales recurrentes. Los pesos se generan para cada canal de características a través de un parámetro w, donde el parámetro w se aprende a modelar explícitamente la correlación entre los canales de características.

Finalmente, hay una operación de Repeso. El peso de la salida de Excitación se considera la importancia de cada canal de características después de la selección de características, luego se pondera a la característica anterior por canal por multiplicación para completar el original en la dimensión de canal. Recalibración de características.

Para el entrenamiento del modelo ResNet50 con SE, se establecieron los siguientes parámetros, como se observa en la tabla 2:

Modelo	ResNet50_SE
Input	32x32x3
Parametros	23.598.026
Epocas	40
Loss	categorical_crossentropy
Optimizer	Adam

Tabla 2. Parámetros de entrenamiento ResNet50_SE

Además, se entrenó el modelo con SE variando el optimizador por un Adam ajustado, lo que permitió mayor rapidez en el entrenamiento como se observa en la tabla 3:

Modelo	ResNet50_SE Adam
Input	32x32x3
Parametros	23.598.026
Epocas	40
Loss	categorical_crossentropy
Optimizer	Adam
lr	0,0001
beta_1	0,9
beta_2	0,999
epsilon	None
decay	1,00E-06
amsgrad	False

Tabla 3. Parámetros de entrenamiento ResNet50_SE con Adam ajustado

2.3 ResNet50 con módulo de atención CBAM

CBAM – Convolutional Block Attention Module es un mecanismo de atención que toma como entrada un mapa de características de un bloque convolucional intermedio e integra secuencialmente dos submódulos: Channel attention module y Spatial attention module. Este mapa de atención se multiplica por el mapa de características para obtener finalmente una salida más refinada, como se observa de forma generalizada en la fig 4.

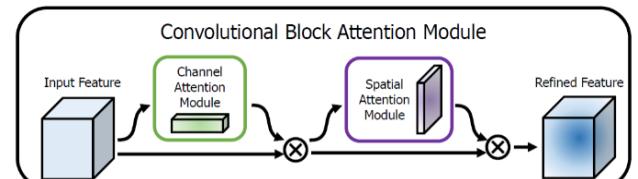


Fig. 4. Arquitectura general de CBAM

Para la implementación, se crearon dos funciones: ch_attention y sp_attention siguiendo la arquitectura de la figura 5 y 6 respectivamente.

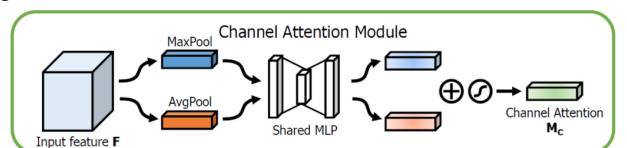


Fig. 5. Channel Attention Module

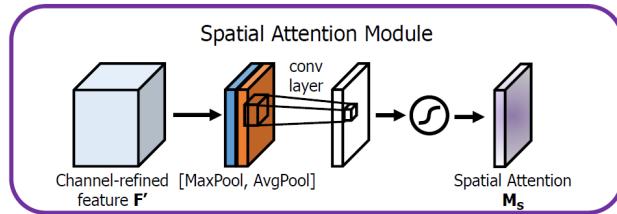


Fig. 6. Spatial Attention Module

Para el entrenamiento del modelo ResNet50 con CBAM, se establecieron los siguientes parámetros:

Modelo	ResNet50_CBAM
Input	32x32x3
Parametros	27.232.738
Epochas	40
Loss	categorical_crossentropy
Optimizer	Adam

Tabla 4. Parámetros de entrenamiento ResNet50_CBAM

III. Resultados

Implementamos dos mecanismos de atención Squeeze-Excitation (SE) y Módulo de Atención de Bloque Convolucionales (CBAM), aplicado a una arquitectura básica ResNet50 buscando mejorar su poder representación intentando replicar los resultados obtenidos por los autores y creadores de los respectivos módulos de atención.

Para el análisis presentamos los resultados de desempeño de los modelos propuestos RESNET50, RESNET50-SE y RESNET50-CBAM aplicados a la base de datos CIFAR10, como se ve en la tabla 5.

	Modelo	Acc_Trainig	Acc_Val	Loss_Trainig	Loss_Val
0	ResNet50	87.85	75.85	44.89	85.86
1	ResNet50_SE	97.52	81.42	12.13	96.22
2	ResNet50_CBAM	98.30	82.89	10.04	88.20
3	ResNet50_SE_ADAM	88.51	73.83	43.10	101.43

Tabla 5. Comparativo de desempeño de los modelos

Se puede observar que los modelos con módulo de atención ResNet50_SE y ResNet50_CBAM, superan al modelo básico con un accuracy en el set de validación de 81.42% y 82.89% respectivamente, lo que en teoría comprueba mejora la clasificación de las imágenes.

Se presentan también los gráficos comparativos de accuracy del entrenamiento y validación a través del entrenamiento en la figura 7 y 8 respectivamente.

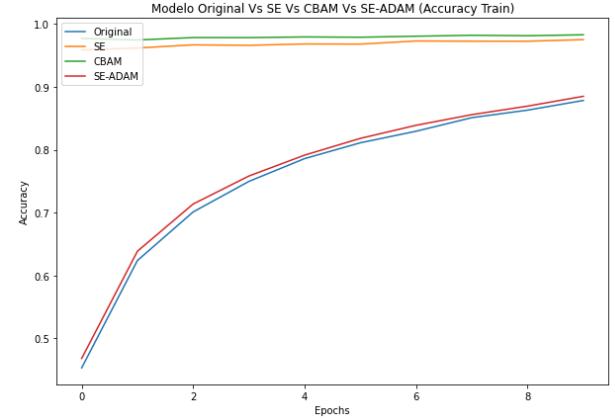


Fig.7 Gráfico comparativo de Accuracy de Entrenamiento

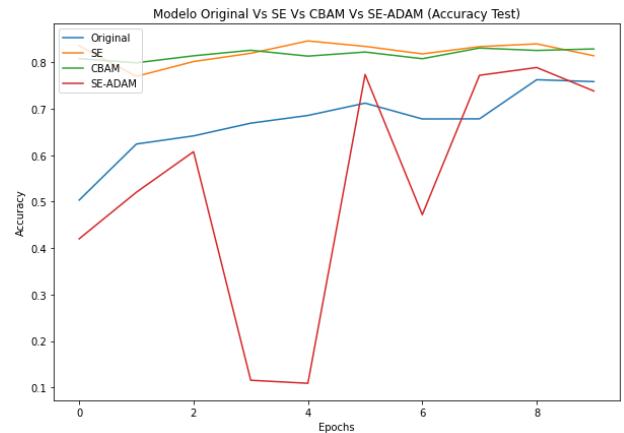
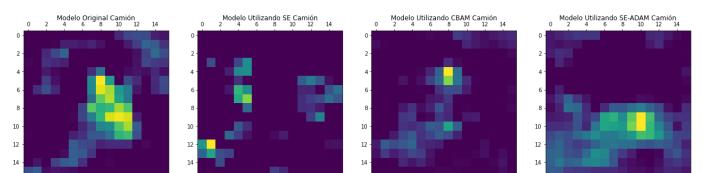
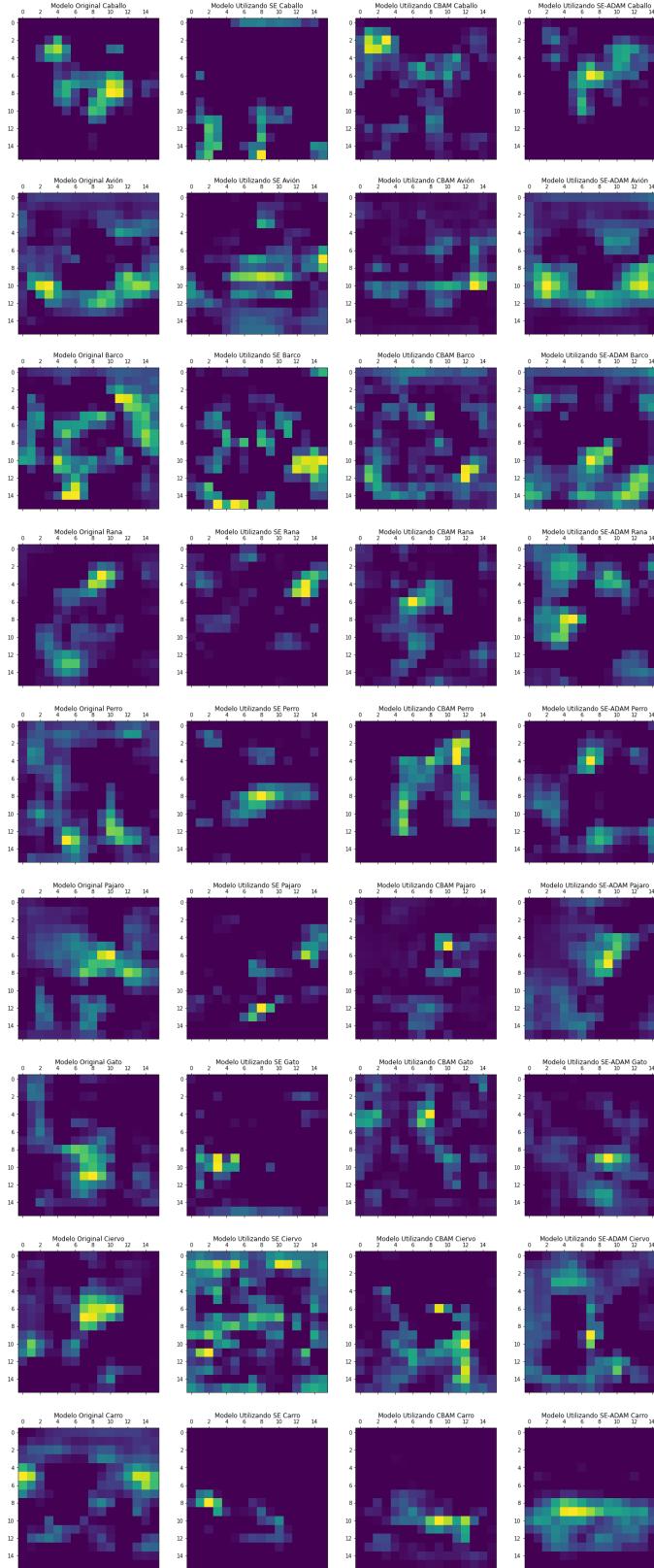


Fig.8 Gráfico comparativo de Accuracy de Validación

Para validar el foco de atención de cada modelo respecto a las imágenes, utilizamos Grad-CAM como método de visualización que utiliza gradientes para calcular la importancia de las ubicaciones espaciales en capas convolucionales.

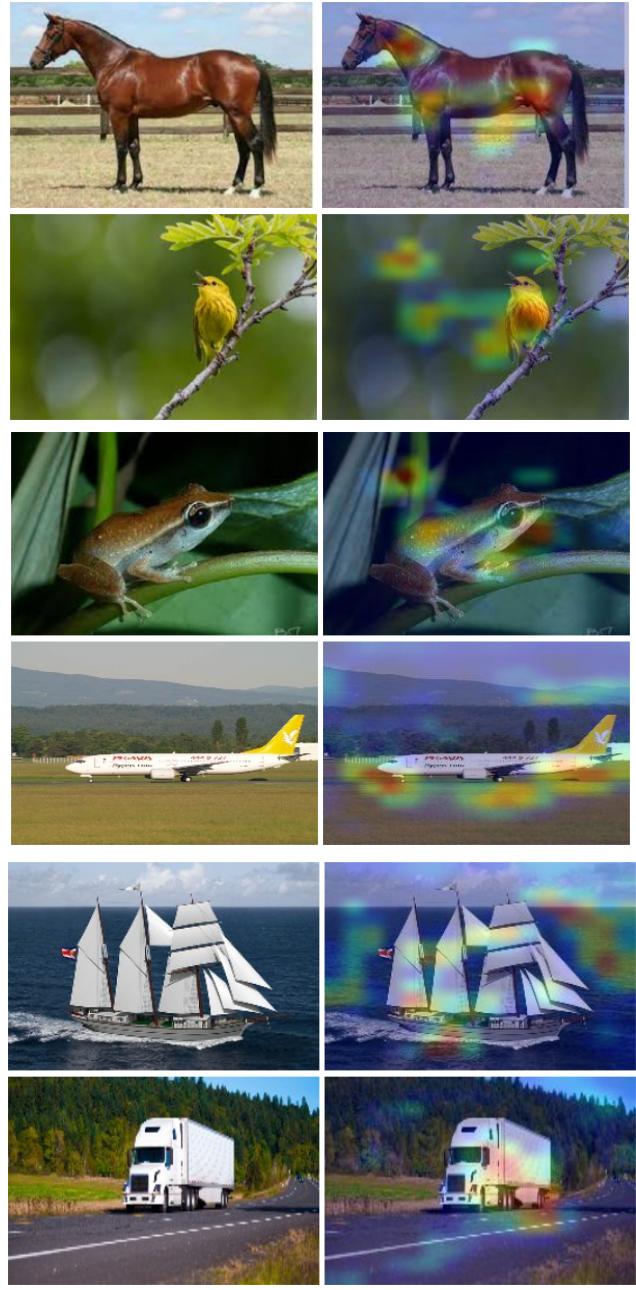
Observando el heatmap de cada imagen y modelo, vemos que la atención la realiza mejor el modelo RestNet_SE con Adam que tiene parámetros ajustados.

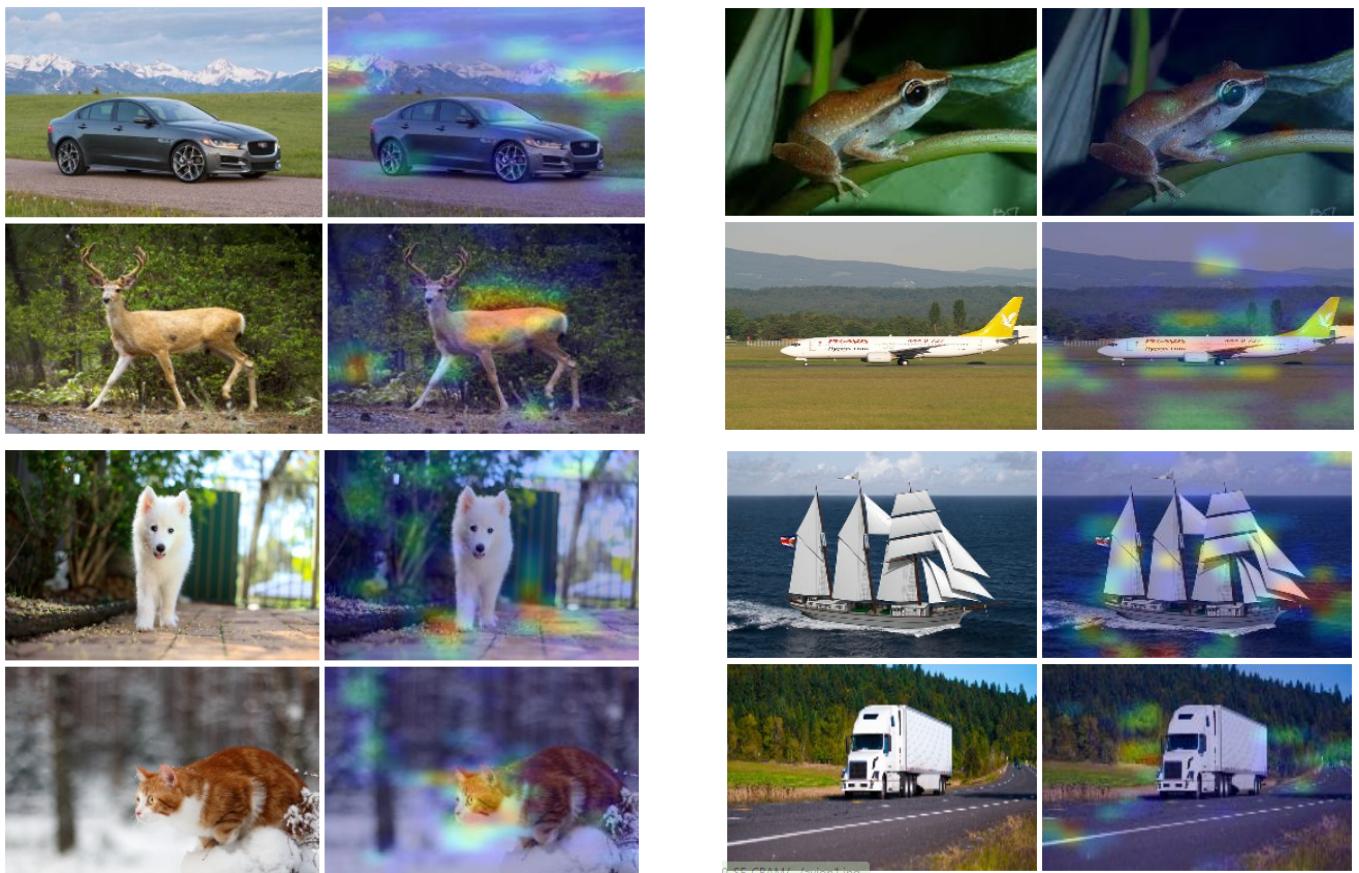




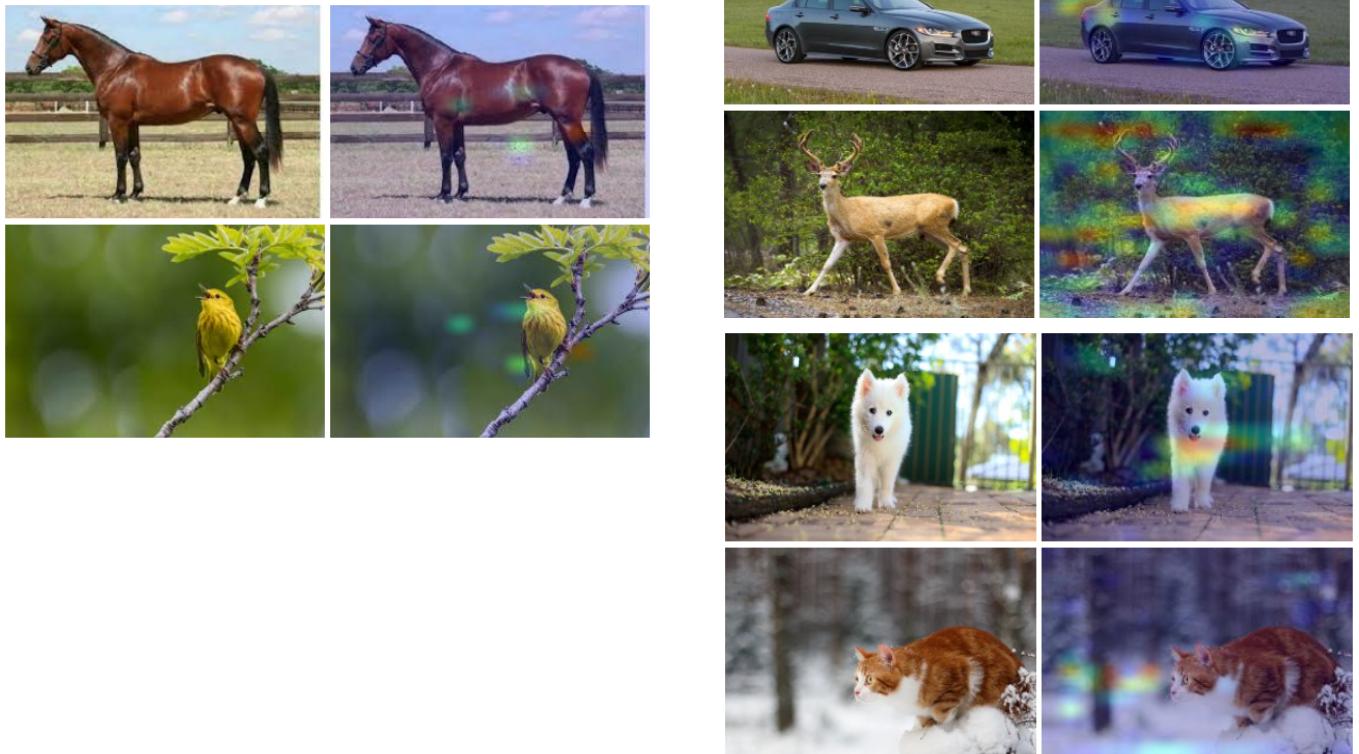
A continuación se muestra la validación de las arquitecturas en 10 diferentes imágenes de validación visualizando la atención con GradCam:

1. Arquitectura original

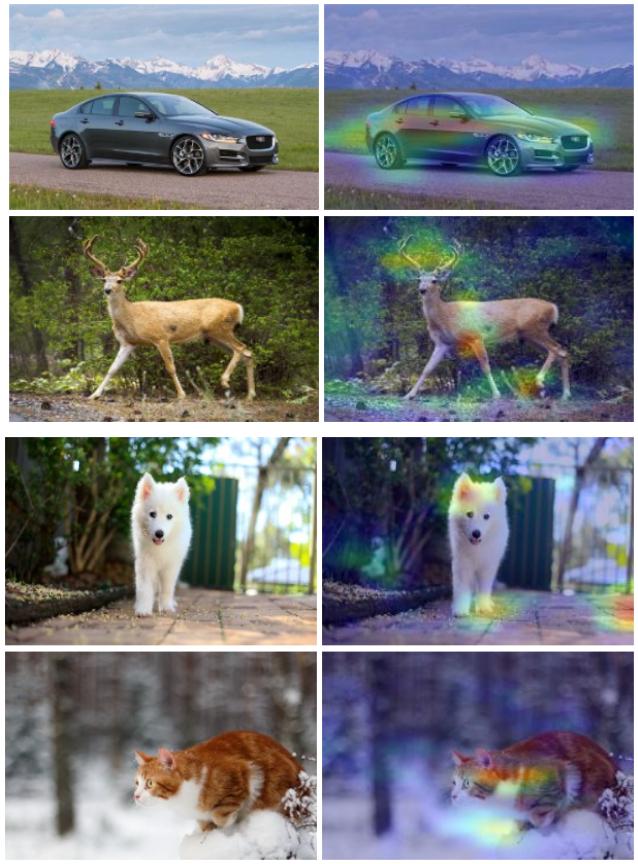
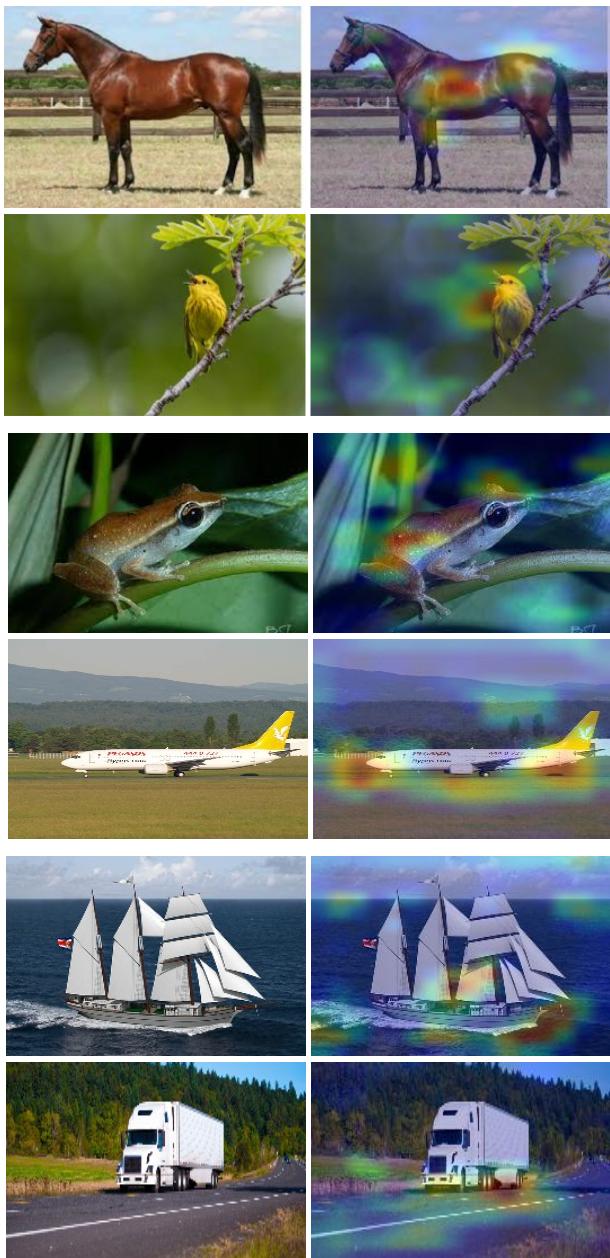




2. Squeeze Excitation

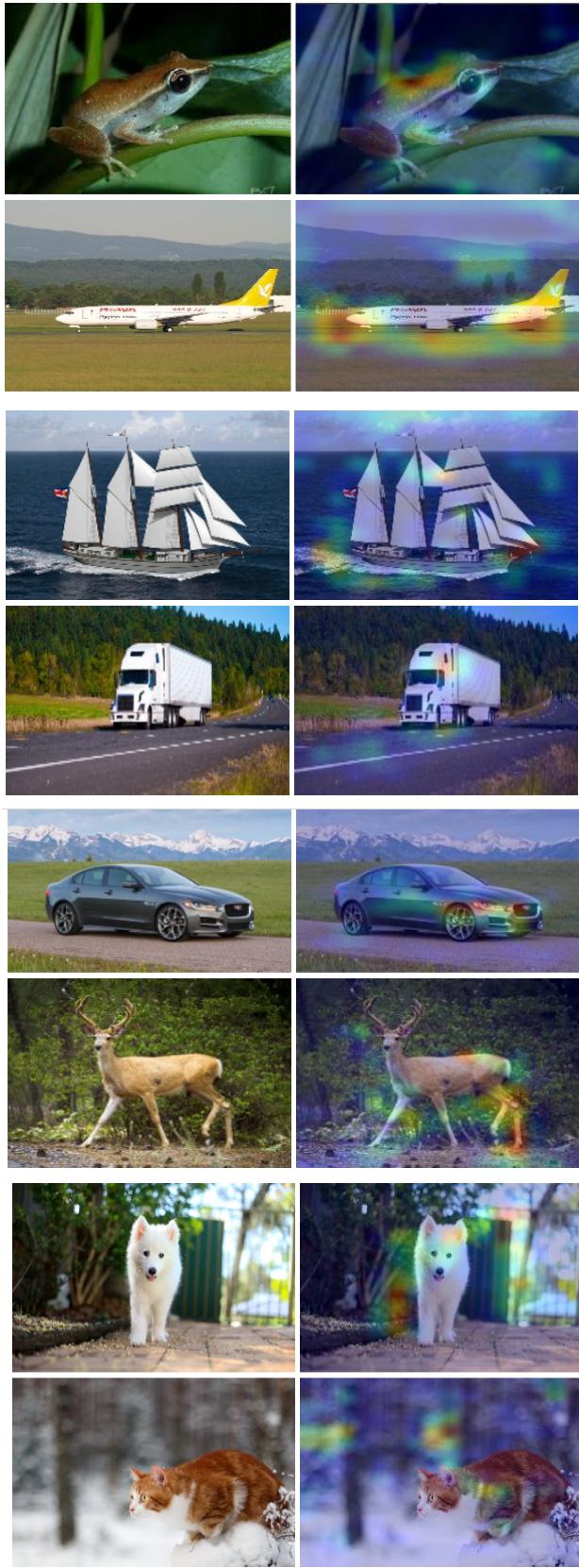


3. Squeeze Excitation Adam ajustado



4. CBAM





Con el heatmap mostrado anteriormente se superpone las 10 imágenes de pruebas a este y se observa que ResNet 50 original, se centra en algunos rasgos deseados de las imágenes que parecen significativos a la hora de predecir la clasificación aún con un accuracy obtenido en el dataset de validación del 72.2%. Por otro lado el modelo ResNet50_CBAM, logra centrarse en la mayoría de rasgos significativos de las imágenes a la hora de predecir la clasificación, lo que en teoría comprueba una mejora a la clasificación de las imágenes.

Conclusiones

A través de la implementación realizada y descrita anteriormente, se pudo validar que los mecanismos o modelos de atención definitivamente mejoran el poder de foco de las redes neuronales convolucionales y su representación. El uso de cada mecanismo de atención dependerá de la tarea que se quiera realizar, el dataset seleccionado y por supuesto, el tiempo de entrenamiento de cada modelo.

En este caso, si bien tuvimos un aceptable desempeño en cada uno de los modelos implementados, se esperaba obtener una mejor distribución de los heatmaps en cada imagen que realmente enfocara el objetivo, sin embargo, tuvimos problemas con la herramienta utilizada para el entrenamiento (Google Colab), ya que cada ciertas épocas se reiniciaba y bloqueaba las cuentas utilizadas.

En ese sentido, si bien la implementación de los módulos de atención en la arquitectura de red (ResNet50) fue exitosa, concluimos que el entrenamiento requiere de muchas más épocas para obtener resultados más acordes que coincidan con los resultados presentados en los artículos de los autores de los modelos.

Referencias

- [1] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, Philip H.S. Torr. “Learn To Pay Attention”, <https://arxiv.org/abs/1804.02391>, Abril 2018
- [2] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu, “Squeeze-and-Excitation Networks”, <https://arxiv.org/abs/1709.01507>, Septiembre 2017
- [3] Sanghyun Woo, Jongchan Park, Joon-Young Lee, In So Kweon. “CBAM:Convolutional Block Attention Modules”, <https://arxiv.org/abs/1807.06521> , Julio 2018

- [4] Draelos Ballantyne, Rachel Lea. “Learn to pay attention! Visual Attention in CNN’s”
<https://towardsdatascience.com/learn-to-pay-attention-trainable-visual-attention-in-cnns-87e2869f89f1>, Agosto 2019.
- [5] Weng, Lilian, “Attention! Attention”
<https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html>, Junio 2018.
- [6] Boesch, Gaudenz. “Deep Residual Networks (ResNet, ResNet50) – Guide 2021”.
<https://viso.ai/deep-learning/resnet-residual-neural-network/>, Agosto 2021
- [7] Prove, Paul-Louis. “Squeeze-and-Excitation Networks”,
<https://towardsdatascience.com/squeeze-and-excitation-networks-9ef5e71eacd7>, Octubre 2017.
- [8] Tsang, Sik-Ho. “Reading CBAM-Convolutional Block Attention Module (Image Classification)”,
<https://sh-tsang.medium.com/reading-cbam-convolutional-block-attention-module-image-classification-ddbaf10f7430>, Octubre 2018.

