

# Breast Cancer Diagnosis Based on CNNs Using Thermal Imaging

1<sup>st</sup> Miguel Pimiento

*Escuela de Ingeniería de Sistemas e Informática*  
*Universidad Industrial de Santander*  
Bucaramanga, Colombia  
mgefrad@gmail.com

2<sup>nd</sup> Guillermo Pinto

*Escuela de Ingeniería de Sistemas e Informática*  
*Universidad Industrial de Santander*  
Bucaramanga, Colombia  
camachopinto\_@hotmail.com

3<sup>rd</sup> Oscar Torrens

*Escuela de Ingeniería de Sistemas e Informática*  
*Universidad Industrial de Santander*  
Bucaramanga, Colombia  
oscarivan\_torrens@outlook.com

4<sup>rd</sup> Cristian Hernández

*Escuela de Ingeniería de Sistemas e Informática*  
*Universidad Industrial de Santander*  
Bucaramanga, Colombia  
cristianmanuel2304@hotmail.com

## I. RESUMEN

Este estudio investiga el rendimiento de diferentes redes neuronales convolucionales (CNNs) del estado del arte para clasificar imágenes térmicas con el fin de poder detectar el cáncer de mama. Se usaron CNNs como Xception, ResNet56, VGG y AlexNet, variando los hiperparámetros para buscar la mejor configuración para la clasificación. El código fuente está disponible en <https://github.com/gpintoruiz/Thermal-Imaging-Breast-Cancer-Detection>

En este trabajo se utilizó un conjunto de datos de imágenes térmicas llamado DMR-IR [5], el cual contiene imágenes de 56 pacientes, de los cuales 19 están sanos y 37 presentan una mama maligna. Las imágenes fueron tomadas con una cámara FLIR SC620, la cual posee un error de 2 °C o el 2% de la lectura y es capaz de tomar imágenes entre -50 °C y 400 °C. Cabe resaltar que los diagnósticos de los pacientes fueron confirmados previamente por personal experto a través de mamografías y biopsias [6].

## II. INTRODUCCIÓN

El cáncer de seno es un tipo de cáncer que se desarrolla en los conductos galactóforos o en los lobulillos (producción de leche materna) [1]. Este tipo de cáncer se caracteriza por desarrollarse mayormente en mujeres y multiplicarse sin control, produciendo así tumores con células de mama alteradas [2]. Justamente, al formar tumores con dichas células, el calor no se disipa de la manera convencional, provocando así una zona que estará más caliente que sus alrededores, esto debido al problema de disipación. Es aquí cuando las imágenes térmicas toman fuerza como una forma alternativa no invasiva para ayudar en la detección temprana del cáncer de seno [3].

La cuestión radica en hacer uso de una cámara térmica para obtener una imagen (termografía) en la cual dichos bultos sean distinguibles del resto de tejidos, permitiendo así clasificar las imágenes. Para el proceso de clasificación de las imágenes se utilizaron diversas CNNs [4] con el objetivo de encontrar características en las imágenes que posteriormente serán usadas para determinar si estas pertenecen a un paciente enfermo o no. Cabe aclarar que, si bien para este trabajo se sabe que las imágenes de pacientes enfermos corresponden a pacientes con cáncer de mama, esto no implica que las CNNs utilizadas detecten dicha enfermedad [3]. Esto último debido a que al final lo único que se mide y procesa son datos de temperatura.

## III. MÉTODO PROPUESTO

### A. Pre-processing and data augmentation

Primero cargamos todas las matrices de temperatura y las imágenes de máscara en escala de grises en Python, en 1 se puede observar un ejemplo. Luego, se ajusta la imagen usando la matriz y se aplica una segmentación binaria. Cabe mencionar que dependiendo del estado de salud de los pacientes, se tiene en cuenta una mama (enferma) o ambas (sanas) para un total de 1522 imágenes.

Se siguieron metodologías referenciadas [6], como el recorte, redimensionamiento y normalización, dividiendo por la temperatura máxima de todas las imágenes (solo si está habilitado) de cada imagen térmica de mama. El producto de este proceso es una imagen térmica con un tamaño de 250 × 300 puntos de temperatura; en consecuencia, redujimos en una cuarta parte el coste computacional.

El paso de aumento de datos, aplicado solo a las imágenes usadas en entrenamiento, conlleva cinco tipos de generación de datos de imagen como se puede observar en 2, todas con una probabilidad del 50%: (i) volteo horizontal y (ii) vertical; (iii) rotación entre 0 y 45 grados; (iv) desenfoque gaussiano con kernel size igual a 5 y; (v) transformación afín aleatoria (hasta el 5% del tamaño de la imagen).

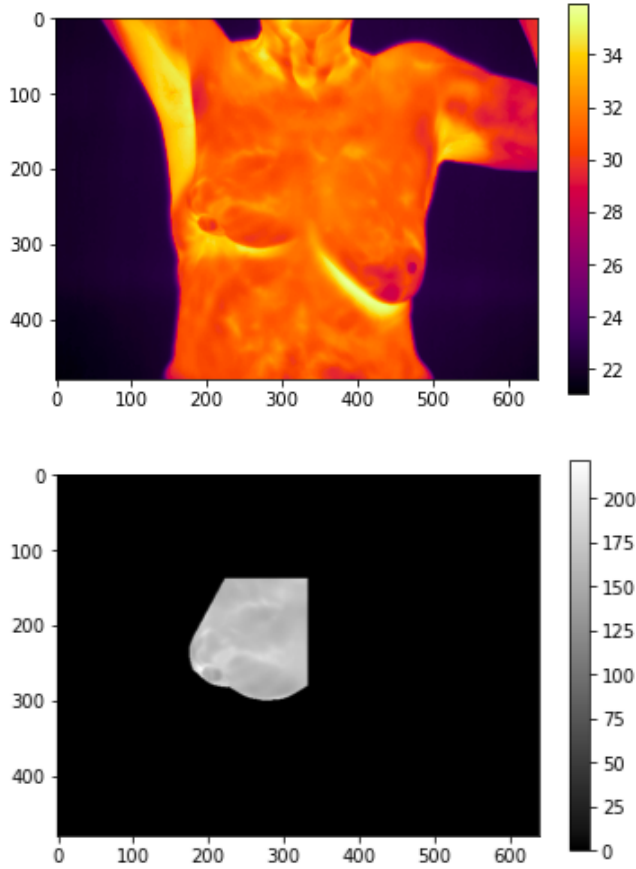


Fig. 1. Ejemplos de imágenes por paciente en la base de datos DMR-IR.

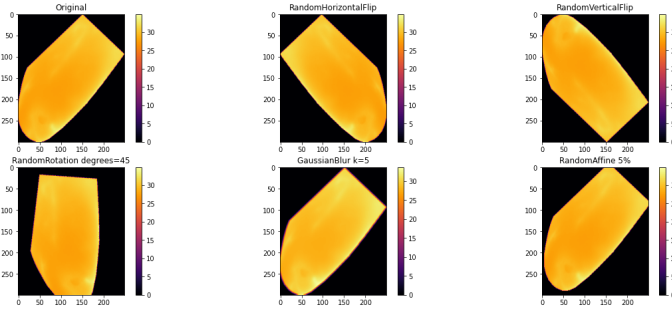


Fig. 2. Aumento de datos aplicado a las imágenes.

### B. Metodología de división de la base de datos

En lugar de dividir inmediatamente toda la base de datos, realizamos una división equilibrada por pacientes: 48 pacientes para el conjunto de entrenamiento y 8 para el conjunto de prueba como se muestra en 3. Además, para obtener una estimación más precisa del rendimiento de los modelos, realizamos una validación cruzada, que se explica en IV.

Bajo ese marco de entrenamiento propuesto, probamos varias arquitecturas CNN del estado del arte: ResNet [7], VGG16 [8], AlexNet [9] y Xception [10], todas las archi-

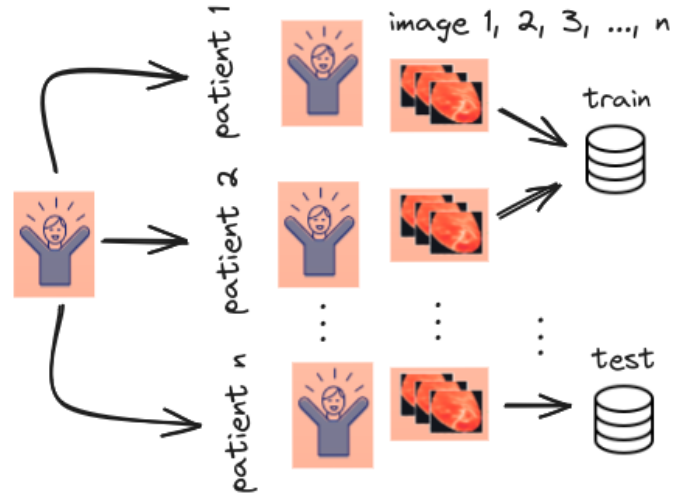


Fig. 3. División de la base de datos por paciente.

tecturas se implementaron en Pytorch siguiendo las pautas establecidas en los documentos originales con las siguientes dos limitaciones: (i) la arquitectura debe recibir una imagen de un canal [1, 250, 300] y; (ii) la arquitectura debe entregar una sola neurona de salida. Además, para computar la pérdida en el problema de clasificación binaria (maligno, sano) usamos BCEWithLogitsLoss [11].

### C. Arquitectura Alexnet

Creada por Alex Krizhevsky, Ilya Sutskever y Geoffrey E. Hinton Alexnet [9] fué una red neuronal que revolucionó el enfoque desde el cual se realizaban este tipo de proceso. Fué la primera CNN en entrenar con GPU y la CNN que revolucionó el mundo del deep learning. La red cuenta con mas de 60 millones de parametros y cuenta originalmente con una entrada de  $227 \times 227 \times 3$  (en el caso de este informe estos parametros de entrada deben ser otros).

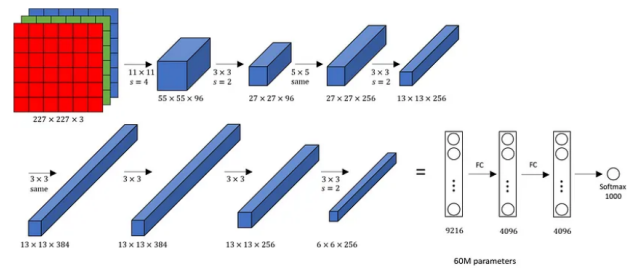


Fig. 4. La arquitectura Alexnet.

Composición de la Alexnet: Dejando de lado la ya mencionada entrada, la alexnet cuenta con 5 capas de convolución, 2 capas FC, 2 capas normalizadas, 3 capas de max pooling y una capa softmax (salida). Las capas de convolución utilizan RELU como función de activación, esta decisión no es tomada

a la ligera dado que el uso de RELU es determinante para obtener los mejores resultados. Referente al max pooling, en Alexnet se usan ventanas 3x3.

Alexnet fué muy importante para el deep learning al punto de que varios de sus patrones constituirían las bases de las redes actuales [12].

#### D. Arquitectura VGG16

Es una red neuronal convolucional presentada en el desafío ImageNet del 2014, por Karen Simonyan y Andrew Zisserman del Visual Geometry Group de la Universidad de Oxford [13].

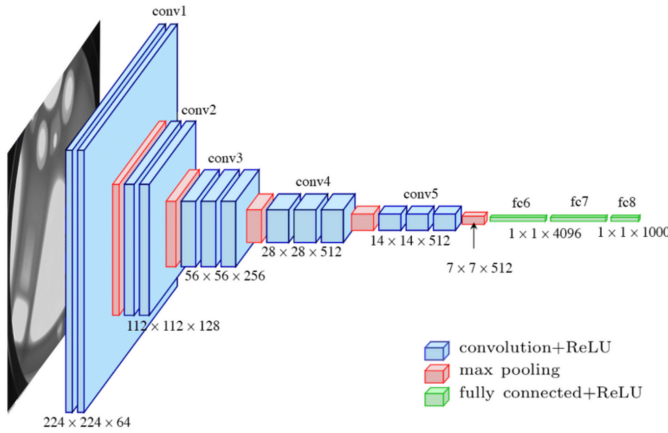


Fig. 5. La arquitectura VGG16.

Consta de 16 capas de profundidad 5, las cuales incluyen 13 capas convolucionales organizadas en bloques. Cada bloque presenta de 2 a 3 capas convolucionales que usan filtros pequeños de  $3 \times 3$ , con un paso de 1 píxel y un padding de 1 píxel, permitiendo capturar características complejas manteniendo la simplicidad de la red.

Al final de cada bloque se aplica un max-pooling con un filtro de  $2 \times 2$  y un paso de 2 píxeles, reduciendo las dimensiones espaciales de la imagen.

Por último, tiene 3 capas completamente conectadas. La primera y la segunda cuentan con 4096 unidades cada una, con activación ReLU, y la última capa tiene hasta 1000 unidades, la cual corresponde al número de clases del problema que se esté trabajando [8].

La VGG16 es uno de los algoritmos más populares para la clasificación de imágenes, además de su facilidad de uso. Asimismo, ha sido base para investigaciones de visión por computadora y ha influenciado el diseño de nuevas arquitecturas como la ResNet.

#### E. Arquitectura Xception

Xception [10] es una arquitectura muy conocida, cuya intuición o hipótesis lleva al extremo la idea propuesta por Inception [14] (de ahí el nombre Xception, con X de *extreme*). La hipótesis fundamental es: *¿Sería razonable hacer una suposición mucho más fuerte que la hipótesis de Inception, y asumir que las correlaciones entre canales y las correlaciones espaciales pueden mapearse completamente por separado?*

Para implementar esta idea, Xception utiliza una variación de las convoluciones  $1 \times 1$  propuestas en Network in Network [15] y las convoluciones separables en profundidad (*depthwise separable convolutions*). Una convolución separable en profundidad, consiste en una convolución en profundidad, es decir, una convolución espacial realizada de forma independiente en cada canal de una entrada, seguida de una convolución puntual, es decir, una convolución  $1 \times 1$ , proyectando los canales de salida de la convolución en profundidad en un nuevo espacio de canales.

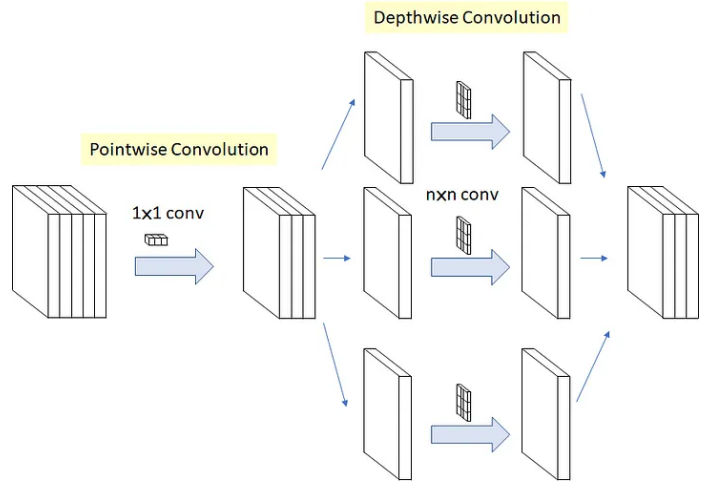


Fig. 6. Ilustración de una convolución separable en profundidad [16].

La arquitectura Xception: los datos primero pasan por el *entry flow*, luego por el *middle flow* que se repite ocho veces y finalmente por el *exit flow*. Todas las capas de Convolution y SeparableConvolution son seguidas por normalización por lotes (batch normalization) (no incluida en el diagrama). Todas las capas de SeparableConvolution utilizan un multiplicador de profundidad de 1 (sin expansión de profundidad). Una descripción completa de las especificaciones de la red se muestra en la Figura 7. Xception tiene 36 capas de convolución que forman la base de extracción de características de la red. Las 36 capas de convolución están estructuradas en 14 módulos, todos los cuales tienen conexiones residuales lineales a su alrededor, excepto los primeros y últimos módulos. En resumen, Xception es una pila lineal de capas de convoluciones separables en profundidad con conexiones residuales.

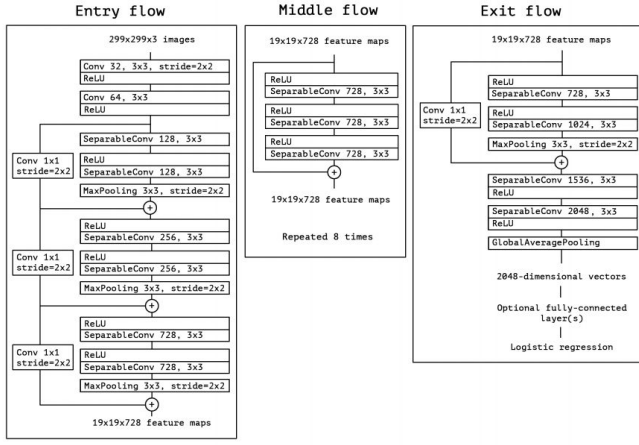


Fig. 7. La arquitectura Xception.

### F. Arquitectura ResNet56

La ResNet (Red Neuronal Residual), introducida por He et al. en 2015, es una arquitectura de red neuronal profunda que ganó el concurso de ImageNet en el año 2015[17]. Resnet surgió para resolver el problema de la degradación, es decir, la precisión de la red caía rápidamente durante el entrenamiento entre mas capas esta poseía, lo que implica que las CNNs menos profundas tenían mejor desempeño que sus homologas mas profundas, lo cual es algo inconcebible por el hecho de que una red mas profunda puede ser construida a traves de su homologa menos profunda con capas extras que apliquen la operación de identidad.

Gran parte del problema de la degradación se debe al desvanecimiento del gradiente, es decir el gradiente cae exponencialmente durante la retropropagación haciendo cada vez mas difícil el entrenamiento de redes muy profundas, es por esto que ResNet esta construido a partir de bloques residuales, los cuales al hacer uso de las skip connections, permiten una mejor retropropagación del gradiente, debido a que este no debe pasar por tantas capas intermedias, tal y como se muestra en la Figura 8.

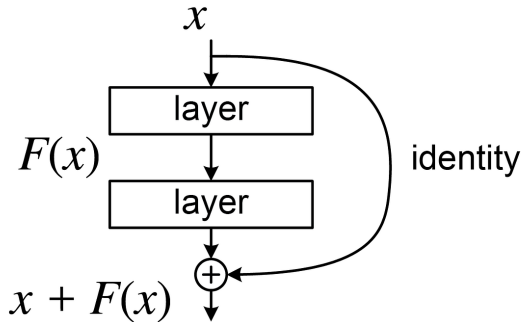


Fig. 8. Ilustración de un bloque residual con su respectiva skip connection [17].

Para este problema en especifico se implemento Resnet56,

la cual cuenta con una capa inicial de convolución con 16 filtros de 3x3, para entrar en 3 bloques intermedios, los cuales poseen cada uno 9 bloques residuales, estos últimos poseen cada uno 2 capas convolucionales cada una de 16,32,64 respectivamente y su respectiva skip connection, y una capa final Fully Connected. Ver Figura 9.

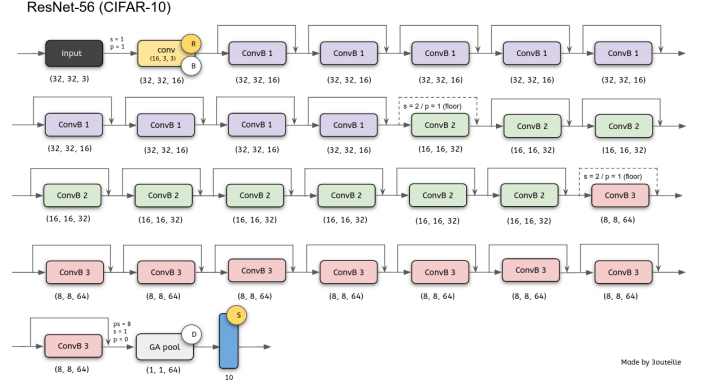


Fig. 9. Ilustración de la ResNet56 [17].

## IV. RESULTADOS

Se realizaron pruebas con cada una de las CNNs previamente presentadas. Primero, se determinaron los mejores hiperparámetros para cada modelo, utilizando optimización bayesiana y probando en un fold al azar en cada iteración, los resultados se muestran en la tabla I.

Nuestro conjunto de datos se dividió en 7 grupos, cada uno compuesto por 8 pacientes, representando cada grupo un "FOLD" 10.

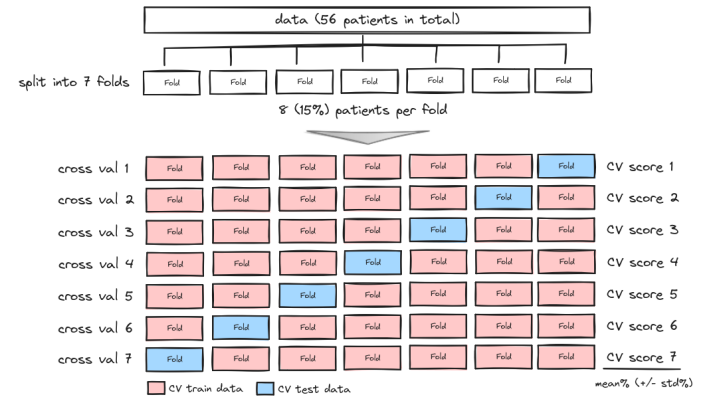


Fig. 10. Validación cruzada por grupos.

Utilizando los mejores hiperparámetros de cada modelo, se entrenó cada uno en todos los FOLD y se obtuvieron sus métricas, como se aprecia en la tabla II.

Finalmente, se calculó el promedio de cada métrica y su desviación estándar para obtener los resultados finales de cada modelo, que se muestran en la tabla III.

TABLE I  
MEJORES HIPERPARÁMETROS PARA CADA MODELO

Modelo	Augmented	Batch Size	Learning Rate	Normalize	Optimizer
AlexNet	false	8	0.000015528833190894192	true	adam
VGG16	false	8	0.00001552883319089419	true	adam
Xception	true	64	0.0006177338121893195	false	adam
ResNet56	true	32	0.00001219638732981837	true	sgd

TABLE II  
MÉTRICAS DE EVALUACIÓN PARA CADA MODELO Y FOLD

Modelo	Fold	Accuracy	Recall	Precision	F1-score
AlexNet	1	0.7885	0.6785	0.9038	0.7551
	2	0.9152	0.9259	0.9327	0.9211
	3	0.9017	0.8401	0.8714	0.8360
	4	0.9062	0.7990	1.0000	0.8759
	5	0.9910	1.0000	0.9761	0.9857
	6	0.8214	0.6363	0.9642	0.7307
	7	0.8169	0.7861	0.9005	0.8293
VGG16	1	0.3942	0.0000	0.0000	0.0000
	2	0.7946	0.5382	0.8214	0.6346
	3	0.8839	0.7169	0.9286	0.7927
	4	0.6205	0.9202	0.5477	0.6740
	5	0.9554	1.0000	0.9137	0.9494
	6	0.8214	1.0000	0.7255	0.8253
	7	0.8705	0.8082	1.0000	0.8842
Xception	1	0.8969	0.8354	1.0000	0.9096
	2	0.9118	0.8640	1.0000	0.9266
	3	0.9883	0.9753	1.0000	0.9874
	4	0.9872	0.9853	0.9922	0.9885
	5	0.8265	0.6267	1.0000	0.7677
	6	0.8449	0.7787	0.8756	0.8208
	7	0.8186	0.5719	1.0000	0.7228
ResNet56	1	0.9330	0.8947	1.0000	0.9430
	2	0.8202	0.6078	1.0000	0.7474
	3	0.9911	0.9827	1.0000	0.9910
	4	0.9031	0.7903	1.0000	0.8774
	5	1.0000	1.0000	1.0000	1.0000
	6	0.9037	1.0000	0.8293	0.9052
	7	0.8827	0.8206	1.0000	0.8975

## V. CONCLUSIONES

Analizando el rendimiento de las CNNs, se puede observar cómo la red que mejor rendimiento tuvo para este estudio fue la ResNet56 y la red con menor rendimiento general fue la VGG16. Por otra parte, se hace hincapié en el buen rendimiento de la AlexNet (no probada en el paper original), dado que muestra resultados cercanos a la mejor red para este estudio y puede ser una alternativa menos profunda a redes como Xception o ResNet56. De esto se puede destacar a las CNNs como herramientas clave para este tipo de escenarios.

Hemos revisado varios estudios, sus técnicas y metodologías hacia bases de datos de imágenes térmicas para el diagnóstico del cáncer de mama; sin embargo, es importante mencionar algunas limitaciones. En primer lugar, existe una escasez de información (imágenes térmicas), ya que actualmente solo se dispone de una base de datos conocida (DMR-IR), lo cual restringe la capacidad de generalización que las CNNs podrían alcanzar. Se requiere más investigación e inversión

para desarrollar nuevas bases de datos públicas.

Respecto a los inconvenientes para este estudio: Al usar Google Colaboratory se cuenta con recursos limitados, por lo cual hay limitaciones que se ven reflejadas en los hiperparámetros. Batch sizes grandes (64) provocan un desbordamiento de la memoria, por lo cual no se pueden usar, y fue necesario manejar hiperparámetros que nos dieran resultados en menos tiempo dado que se usó la versión gratuita de la GPU de Google Colaboratory, lo cual limita el tiempo máximo en el que se pueden realizar las pruebas. Con base en lo anterior, se recomienda contar con más recursos para evitar las limitaciones de memoria y tiempo. El preprocesamiento de las imágenes es esencial. Elegir un resize adecuado para las imágenes no solo influye en los resultados, sino también en el tiempo de cómputo, lo cual es crucial en respuestas médicas.

En cuanto a la data augmentation, inicialmente se probaron transformaciones de intensidad a nivel de píxel, lo que resultó en peores resultados. Sin embargo, esto no es lógico, dado que el propósito de la data augmentation es mejorar el rendimiento con menos datos. Al eliminar esas transformaciones, se obtuvieron mejores resultados en las redes Xception y ResNet56, lo que indica que conservar en la mayor medida posible la información de temperatura original es más efectivo para este problema.

Por último, es importante mencionar el potencial de las imágenes térmicas con los resultados obtenidos. Si bien las imágenes térmicas no sirven para detectar el cáncer de mama por sí mismas, ayudan a detectar bultos y anomalías, las cuales podrían contribuir a un proceso mayor para un futuro diagnóstico, razón por la cual deberían ser tomadas en cuenta.

TABLE III  
MÉTRICAS DE EVALUACIÓN PROMEDIO PARA CADA MODELO

Modelo	Accuracy	F1-score	Precision	Recall
AlexNet	0.8773 (+/- 0.0660)	0.8477 (+/- 0.0828)	0.9356 (+/- 0.0431)	0.8477 (+/- 0.0828)
VGG16	0.7630 (+/- 0.1788)	0.6800 (+/- 0.2958)	0.7053 (+/- 0.3199)	0.7119 (+/- 0.3286)
Xception	0.8963 (+/- 0.0660)	0.8748 (+/- 0.0978)	<b>0.9811 (+/- 0.0431)</b>	0.8053 (+/- 0.1477)
ResNet56	<b>0.9191 (+/- 0.0580)</b>	<b>0.9088 (+/- 0.0788)</b>	0.9756 (+/- 0.0597)	<b>0.8709 (+/- 0.1334)</b>

#### REFERENCES

- [1] “What is breast cancer? its types, signs, causes & treatments — onco,” Onco About Cancer. (), [Online]. Available: <https://onco.com/about-cancer/cancer-types/breast-cancer/> (visited on 07/01/2024).
- [2] World Health Organization. “Breast cancer.” Accessed: 2024-07-04. (2024), [Online]. Available: <https://www.who.int/es/news-room/fact-sheets/detail/breast-cancer> (visited on 07/04/2024).
- [3] N. M. u. din, R. A. Dar, M. Rasool, and A. Assad, “Breast cancer detection using deep learning: Datasets, methods, and challenges ahead,” *Computers in Biology and Medicine*, vol. 149, p. 106073, Oct. 1, 2022, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2022.106073. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010482522007818> (visited on 07/01/2024).
- [4] G. Litjens, T. Kooi, B. E. Bejnordi, *et al.*, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, Dec. 1, 2017, ISSN: 1361-8415. DOI: 10.1016/j.media.2017.07.005. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1361841517301135> (visited on 07/01/2024).
- [5] “Visual lab - a methodology for breast disease computer-aided diagnosis.” (), [Online]. Available: <https://visual.ic.uff.br/en/proeng/thiagoelias/> (visited on 07/01/2024).
- [6] J. Zuluaga-Gomez, Z. Al Masry, K. Benagoune, S. Meraghni, and N. Zerhouni, “A CNN-based methodology for breast cancer diagnosis using thermal images,” *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 9, no. 2, pp. 131–145, Mar. 4, 2021, Publisher: Taylor & Francis \_eprint: <https://doi.org/10.1080/21681163.2020.1824685>, ISSN: 2168-1163. DOI: 10.1080/21681163.2020.1824685. [Online]. Available: <https://doi.org/10.1080/21681163.2020.1824685> (visited on 07/01/2024).
- [7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778, ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90. [Online]. Available: <http://ieeexplore.ieee.org/document/7780459/> (visited on 06/30/2024).
- [8] K. Simonyan and A. Zisserman, “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION,” 2015.
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 24, 2017, ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3065386. [Online]. Available: <https://dl.acm.org/doi/10.1145/3065386> (visited on 06/30/2024).
- [10] F. Chollet, *Xception: Deep learning with depthwise separable convolutions*, Apr. 4, 2017. arXiv: 1610.02357[cs]. [Online]. Available: <http://arxiv.org/abs/1610.02357> (visited on 06/30/2024).
- [11] “Bcewithlogitsloss — pytorch 2.3 documentation.” (), [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html>.
- [12] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 28, 2015, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14539. [Online]. Available: <https://www.nature.com/articles/nature14539> (visited on 06/30/2024).
- [13] R. G, *Everything you need to know about vgg16*, Great Learning, Accessed: 2024-07-05, 2021. [Online]. Available: <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>.
- [14] C. Szegedy, W. Liu, Y. Jia, *et al.*, *Going deeper with convolutions*, Sep. 16, 2014. arXiv: 1409.4842[cs]. [Online]. Available: <http://arxiv.org/abs/1409.4842> (visited on 07/04/2024).
- [15] M. Lin, Q. Chen, and S. Yan, *Network in network*, Mar. 4, 2014. arXiv: 1312.4400[cs]. [Online]. Available: <http://arxiv.org/abs/1312.4400> (visited on 07/04/2024).
- [16] S. Tsang, “Review: Xception — with depthwise separable convolution, better than inception-v3 (image classification),” Mar. 20, 2019. [Online]. Available: <https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568>.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015. [Online]. Available: <https://arxiv.org/pdf/1512.03385>.