

Comparación de arquitecturas de redes neuronales convolucionales para el diagnóstico de COVID-19

Daniela Lopez-Betancur¹, Rembrandt Bosco Duran², Carlos Guerrero-Mendez^{3*}, Rogelia Zambrano Rodríguez⁴, Tonatiuh Saucedo Anaya^{3*}

¹ Universidad Politécnica de Aguascalientes,
Dirección de Posgrado e Investigación,
México

² Universidad Autónoma de Zacatecas,
Unidad Académica de Física,
México

³ Universidad Autónoma de Zacatecas,
Unidad Académica de Ciencia y Tecnología de la Luz y la Materia,
México

⁴ Universidad Autónoma de Zacatecas,
Unidad Académica de Contaduría y Administración,
México

daniela.betancur@upa.edu.mx, {remhug23, rodriguez.rogelia23}@gmail.com,
{tsaucedo, guerrero_mendez}@uaz.edu.mx

Resumen. Las redes neuronales convolucionales (CNNs, por sus siglas en inglés) han demostrado un gran potencial para resolver problemas de clasificación con imágenes médicas. En esta investigación, se evaluaron treinta y dos arquitecturas CNNs, y se compararon para realizar el diagnóstico COVID-19 mediante el uso de imágenes radiográficas. Se utilizó una colección de 5,953 imágenes de rayos X de tórax frontales (117 imágenes de pacientes diagnosticados con COVID-19, 4,273 de pacientes con neumonía no relacionada con COVID-19 y 1,563 imágenes etiquetadas como Normal provenientes de pacientes saludables) para entrenar y evaluar las arquitecturas. En este artículo, las métricas de evaluación implementadas están en concordancia con las condiciones requeridas para un conjunto de datos desequilibrado. Siete de los treinta y dos modelos evaluados lograron una clasificación de rendimiento excelente ($\geq 90\%$) según la métrica del Índice de precisión equilibrada (IBA, por sus siglas en inglés). Los tres modelos de CNNs que obtuvieron los mejores resultados en esta investigación fueron Wide_resnet101_2, Resnext101_32x8d y Resnext50_32x4d, los cuales obtuvieron un valor de precisión de clasificación del 97.75%. El problema de

sobreajuste en los modelos se descartó de acuerdo con el comportamiento de los valores de precisión tanto en el conjunto de datos de entrenamiento, como en los de prueba. El mejor modelo para el diagnóstico de COVID-19 es el Resnext101_32x8d, de acuerdo con la matriz de confusión y las métricas logradas de sensibilidad, especificidad, F1-score, G_mean, IBA y tiempo de entrenamiento de 97.75%, 96.40%, 97.75%, 97.06%, 94.34%, 76.98 min, respectivamente.

Palabras clave. Redes neuronales convolucionales, COVID-19, transferencia de aprendizaje.

Comparison of Convolutional Neural Network Architectures for COVID-19 Diagnosis

Abstract. Convolutional neural networks (CNNs) have shown great potential to solve several medical image classification problems. In this research, thirty-two CNN architectures were evaluated and compared to perform COVID-19 diagnosis by using radiographic images. A collection of 5,953 frontal chest X-ray images (117

patients diagnosed with COVID-19, 4,273 with Pneumonia not related to COVID-19, and 1,563 Normal or healthy) was used for training and testing those thirty-two architectures. In this article, the implemented metrics were according to the conditions of an imbalanced dataset. Seven of the thirty-two models evaluated achieved an excellent performance classification ($\geq 90\%$) according to the Index of Balanced Accuracy (IBA) metric. The top three CNN models selected in this research (Wide_resnet101_2, Resnext101_32x8d, and Resnext50_32x4d) obtained the highest classification precision value of 97.75%. The overfitting problem was ruled out according to the evolution of the training and testing precision measurement. The best CNN model for COVID-19 diagnosis is the Resnext101_32x8d according to the confusion matrix and the metrics achieved (sensitivity, specificity, F1-score, G_mean, IBA, and training time of 97.75%, 96.40%, 97.75%, 97.06%, 94.34%, 76.98 min, respectively) by the CNN model.

Keywords. Convolutional neural network, COVID-19, Transfer learning.

1. Introducción

El COVID-19 es una enfermedad infecciosa causada por el síndrome respiratorio agudo severo Coronavirus 2 (SARS-CoV-2), que es capaz de infectar y puede ser grave en humanos [1–5]. Inicialmente, esta enfermedad fue diagnosticada como casos de neumonía acompañada con otros síntomas, como fiebre, tos y dificultad respiratoria [6–8]. Los primeros casos fueron descubiertos en Wuhan China, a finales de diciembre de 2019, para después propagarse rápidamente hacia otras ciudades de China y posteriormente a diferentes países del mundo, convirtiéndose así en una emergencia de salud pública [9], y por consiguiente siendo declarada como una pandemia a nivel mundial en marzo del 2020 por la organización mundial de la salud (OMS) [10].

A pesar de las constantes advertencias y esfuerzos tanto gubernamentales como sociales, hoy en día se reportan 12,552,765 personas infectadas, y más de 561,617 muertes a causa del COVID-19 en 216 países [11]. El mayor número de muertes se ha producido en personas mayores y con dificultades de salud previas como enfermedades cardiovasculares, diabetes, hipertensión, asma y obesidad [12–14].

Con el fin de coadyuvar a esta emergencia de salud pública es necesario del uso y desarrollo de nuevas tecnologías, capaces de diagnosticar rápida y verazmente el COVID-19. Actualmente, una tecnología de gran utilidad en el desarrollo de “sistemas inteligentes” son las redes neuronales convoluciones (CNNs, por sus siglas en inglés). Las CNNs son algoritmos matemáticos que imitan el funcionamiento de la corteza visual humana a través de la aproximación de funciones continuas. Estas son capaces de aprender a detectar patrones simples o complejos en imágenes con la finalidad de clasificarlas [15]. Generalmente, los modelos de CNNs suelen ser tratadas como modelos de “cajas negras” debido a la gran cantidad de operaciones que se realizan en ellas [16].

Las arquitecturas de modelos CNNs implementados en tareas de clasificación están compuestas por dos secciones. La primera, conocida como “Feature extractor”, donde se introduce la imagen y se calculan las principales características de esta. Dependiendo de la arquitectura del modelo se podrán extraer diferentes características de la imagen de entrada [17].

La segunda sección conocida como “Classifier” es donde se utilizan las características extraídas de la imagen de entrada para realizar la clasificación de esta. Por lo tanto, para determinar el mejor modelo en una tarea de clasificación es imperativo analizar las diferentes arquitecturas para comprobar cuál arquitectura logra una clasificación adecuada según las características extraídas de las imágenes de entrenamiento, ya que dependiendo de la naturaleza de las imágenes de entrada es que diferentes arquitecturas podrán obtener diferentes resultados de clasificación.

Específicamente, las CNNs han demostrado ser eficientes en la clasificación y en la detección de afecciones medicas como tumores hepáticos [18–20], tumores de células en huesos [21], cáncer de mama [22–24], cáncer de piel [25–27], en la corrección de cantidades de calcio coronario [28], etc. Diversas investigaciones han incursionado en la clasificación de imágenes radiológicas de pecho con COVID-19 [29, 30], aunque solo consideran comparativas entre una o un pequeño grupo de arquitecturas de CNNs.

Por otro lado, existen investigaciones que consideran la detección de afecciones médicas (no relacionadas con COVID-19) usando imágenes radiológicas de tórax [31, 32]. Si bien, las capacidades de las CNNs, junto con otro tipo de algoritmos, hacen posible realizar tareas de detección, emplear sistemas de detección para el COVID-19 tendría inconveniente de ser una ardua tarea en la segmentación de las regiones de interés. Segmentar o señalar la presencia de las lesiones relacionadas con COVID-19 en imágenes radiológicas de un conjunto de datos lo suficientemente grande para entrenar un modelo requiere ser realizada por un grupo de médicos expertos apoyados por diferentes estudios como biopsias y entre otros. Lo cual sería difícil de encontrar actualmente.

En esta investigación se tiene como objetivo comparar y evaluar, con métricas acordes al conjunto de datos, el aprendizaje y desempeño de las arquitecturas o modelos de CNNs, para poder determinar cuál arquitectura es la más conveniente en el diagnóstico del COVID-19 utilizando imágenes de rayos X de tórax.

Se espera que esta investigación sirva como guía para que la comunidad científica pueda desarrollar sistemas inteligentes robustos para el diagnóstico del COVID-19 sin la necesidad de realizar la monótona comparación del desempeño de las principales arquitecturas de CNNs, y poder implementar directamente el modelo más conveniente para esta tarea. Todos los modelos CNNs analizados en esta investigación puede obtenerse del paquete de código abierto de la librería PyTorch Torchvision del lenguaje de programación Python. Cabe mencionar que los resultados expuestos en este documento son para fines educativos, debido a que no han sido validados médicamente.

El presente artículo se encuentra organizado de la siguiente manera: Materiales y métodos son descritos en la sección 2, donde se describe el conjunto de datos utilizado, se expone el problema de conjuntos de datos desbalanceados, se especifica los procedimientos utilizados para configurar y poner en marcha los diferentes modelos de CNNs, se enlistan los parámetros experimentales utilizados, y se detallan las métricas utilizadas para la evaluación del desempeño de las CNNs.

Por otra parte, en la sección 3 se presenta, compara y discute los resultados obtenidos para todos los modelos de CNN. Finalmente, en la sección 4 se presenta las conclusiones y el trabajo a futuro de esta investigación.

2. Materiales y métodos

En esta investigación, se evaluó el aprendizaje y desempeño de diversos modelos pre-entrenados de CNNs para la clasificación del COVID-19, utilizando Transfer Learning en fine-tuning e imágenes de rayos X de tórax. La evaluación del desempeño de los modelos se calculó en base a métricas especializadas para bases de imágenes desbalanceadas.

2.1. Problema de base de datos desbalanceadas

Una base de datos desbalanceada es aquella que presenta una distribución desigual entre sus clases, y es la que usualmente podemos encontrar en problemas de clasificación [33]. Particularmente en aplicaciones médicas, el registro de imágenes de rayos X de casos positivos de una enfermedad a diagnosticarse es menor que el registro de imágenes de pacientes sanos, generándose así una base de datos desbalanceada.

Utilizar bases de datos con distribuciones desiguales entre sus clases para entrenar un "sistema inteligente" ocasionará que el clasificador obtenga una mayor exactitud sobre la clase con un mayor número de datos (imágenes de pacientes sanos), y una menor exactitud sobre la clase con menor número de datos (imágenes de pacientes enfermos). Por lo tanto, la métrica exactitud que evalúa el desempeño del clasificador dependiendo del número de casos correctos sin importar el tipo de clase, es inapropiada, y solo es útil en bases de datos con distribuciones iguales entre sus clases. Por lo tanto, el desempeño de clasificadores entrenados con bases de datos medicas debe realizarse de acuerdo a métricas especiales (comúnmente usadas en bases de datos desbalanceadas) [34], y que no estén influenciadas por la distribución de los números de casos.

2.2. Bases de datos de imágenes

Los modelos de CNNs analizados y evaluados en esta investigación fueron entrenados utilizando imágenes de rayos X de tórax extraídas de la base de datos de Kaggle “chest X-Ray Images (Pneumonia)”. Esta base de datos consiste en 5,863 imágenes que están distribuidas en dos categorías (Normal, y Neumonía ocasionada por bacterias y virus que no tiene relación al COVID-19) [35].

En el caso de las imágenes con casos positivos de COVID-19, estas fueron extraídas del repositorio de GitHub “COVID-19 image data collection” [36], el cual recolecta imágenes de fuentes públicas de hospitales, médicos y artículos científicos sobre los casos positivos de COVID-19 y otros casos de neumonías ocasionadas por virus y bacterias, aunque para esta investigación solo se recolectaron los casos positivos de COVID-19. El total de imágenes fue dividido de acuerdo al tipo de diagnóstico obtenido de las fuentes de descarga, por lo que se creó una base de datos con las siguientes categorías o clases: “COVID-19”, “Normal” y “Neumonía”.

En total se recolectaron 117 imágenes diagnosticadas con COVID-19, 4,273 imágenes con neumonía y 1,563 imágenes de personas con un diagnóstico normal o libre de afecciones médicas, obteniéndose así una base de datos desbalanceada con tres clases. Cada clase fue dividida aleatoriamente en una relación de 80% para los datos de entrenamiento y 20% para los datos de prueba. La distribución de clases implementada en esta investigación se observa en la Fig. 1.

2.3. Transfer Learning

Se conoce como *Transfer Learning (TL)* al proceso de utilizar un modelo pre-entrenado de CNN para resolver una nueva tarea de clasificación de imágenes. Generalmente, la mayoría de los modelos CNNs son entrenados y han probado ser eficientes en la clasificación de imágenes de la base de datos ImageNet, la cual contiene 1.2 millones de imágenes con 1,000 categorías de objetos y animales comunes [37]. Por lo tanto, al usar TL en modelos pre-entrenados estaremos aprovechando las capacidades de

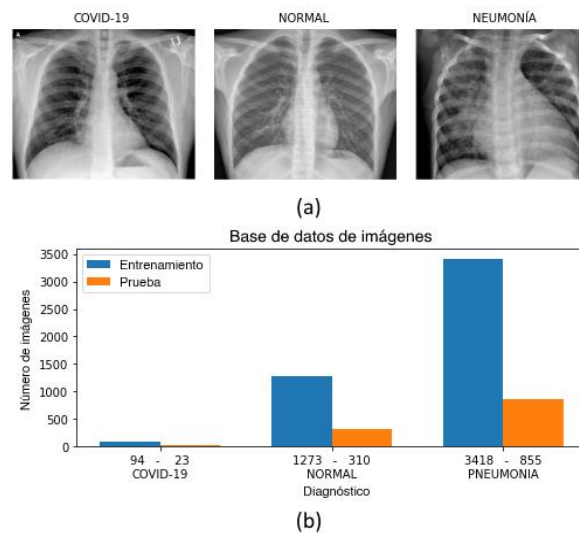


Fig. 1. Base de datos utilizada: (a) Imágenes de ejemplo de los tres tipos de diagnósticos, y (b) distribución numérica de casos en las clases

calcular patrones de modelos, además de todo el conocimiento de clasificación de objetos previamente obtenido para reutilizarlo en la elaboración de una nueva tarea de clasificación.

Existen dos técnicas para implementar el TL. La primera técnica se conoce como *Feature extraction*. En esta, se aprovecha los pesos de la mayoría de las neuronas entrenadas, así como de la arquitectura de CNN para buscar patrones, y solo se modificará el número de neuronas de la capa de salida en relación con el número de clases de la nueva tarea. Por último, los pesos de las nuevas neuronas en la capa de salida son calculados en un nuevo proceso de entrenamiento utilizando la nueva base de datos.

Técnicamente, el modelo utilizará los pesos de las neuronas ya calculados en un entrenamiento previo para clasificar las nuevas imágenes de nueva tarea. La segunda técnica de TL es llamada *Fine-tuning*. En esta, la última capa de la CNN es también redimensionada según el número de clases de la nueva tarea, pero a diferencia de la primera técnica, todos los parámetros de pesos de las neuronas artificiales son actualizados durante el nuevo proceso de entrenamiento, y solo aprovecharemos la arquitectura del modelo de la CNN.

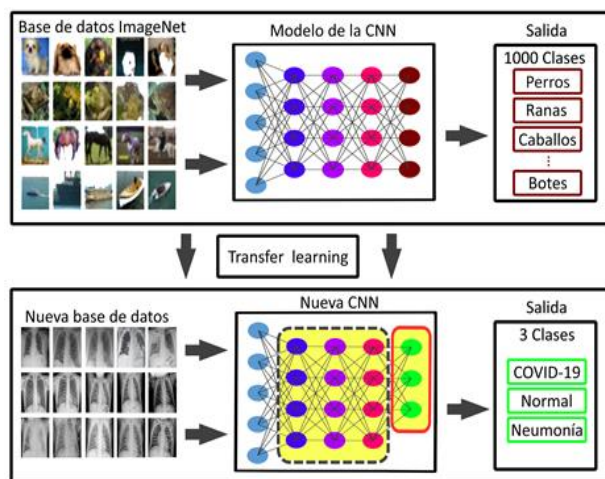


Fig. 2. Representación del proceso de TF y configuración de la CNN. El cuadro superior representa la CNN pre-entrenada en su estado original, la cual es capaz de clasificar 1000 categorías diferentes, al usar *fine-tuning* la última capa es redimensionada y los pesos de las neuronas cambian. La nueva CNN tiene como objetivo clasificar tres nuevas clases

Esta última técnica es útil cuando la naturaleza de las imágenes no tiene relación con el tipo de imágenes del entrenamiento previo. Cabe resaltar, que el TL es útil cuando el número de imágenes de entrenamiento es reducido y los recursos computacionales son limitados [38]. En la Fig. 2 se muestra el proceso de TL en *Fine-tuning* utilizado en esta investigación.

2.4. Data augmentation

Una técnica común y muy útil para incrementar el número de datos disponibles en el proceso de entrenamiento es el Data augmentation. Esta técnica consiste en generar digitalmente nuevas imágenes de acuerdo a una lista de transformaciones. De esta manera se evita tener poco número de ejemplos para el entrenamiento y que la CNN se sobreentrene (se tenga *overfitting*) [39–41].

Con la generación de nuevas imágenes se puede aumentar la robustez de un clasificador de imágenes. Además, al usar data augmentation se reduce el repetitivo proceso de registro,

procesamiento, etiquetado y almacenamiento de nuevas imágenes.

En esta investigación se implementaron las transformaciones: *RandomResizedCrop*, donde una imagen es recortada aleatoriamente, y posteriormente es redimensionada de acuerdo con los requerimientos de tamaño de la imagen de entrada de la CNN; *RandomHorizontalFlip*, donde se generan reflexiones verticales aleatorias en las imágenes de entrada; y *RandomRotation* donde se rota aleatoriamente la imagen de entrada dentro de un rango determinado. En esta investigación se seleccionó un rango de 0 a 20 grados. En la Fig. 3 se muestran las diferentes transformaciones realizadas aleatoriamente en las imágenes de la base de datos de entrenamiento.

2.5. Hiperparámetros

Se le llaman hiperparámetros a aquellos elementos que rigen el comportamiento de aprendizaje en un algoritmo inteligente. Estos hiperparámetros están directamente asociados con la eficiencia del proceso de entrenamiento. En esta investigación, los hiperparámetros fueron establecidos cuidadosamente para alcanzar el óptimo desempeño de los modelos usados. Un hiperparámetro de suma importancia es el número de épocas del proceso de entrenamiento. El número de épocas indicará el número de veces en que el total de las imágenes de entrenamiento son introducidas en la CNN, la cual buscará patrones y tratará de clasificar en base a esos patrones el tipo de imagen de entrada.

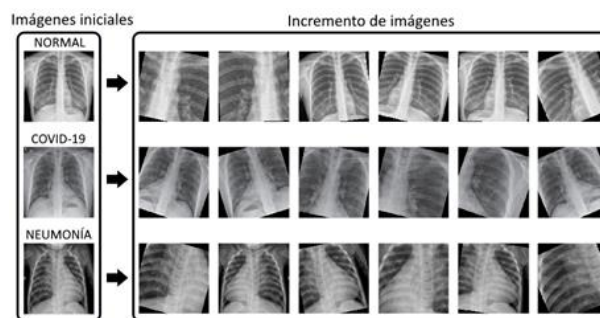
En cada época, la CNN modificará los pesos de las neuronas hasta obtener un mínimo error en la clasificación de imágenes. En nuestro caso, el entrenamiento de las arquitecturas de las CNNs seleccionadas fue establecido en 50 épocas.

La selección del número de épocas se realizó en base al comportamiento de aprendizaje de los modelos, así como también, con el fin de evitar que la CNN alcance *overfitting*, que es cuando el modelo se memoriza todos los ejemplos de entrenamiento.

A su vez, las arquitecturas fueron entrenadas utilizando el algoritmo optimizador de aprendizaje momentum de 0.9 y un batch size de 16. La Tabla 1 enlista los hiperparámetros utilizados en el entrenamiento de todos los modelos de CNNs.

Tabla 1. Hiperparámetros utilizados

Hiperparámetro	Valor
Optimizador	SGD
Tasa de aprendizaje	0.001
Momentum	0.9
Épocas	50
Batch size	16

**Fig. 3.** Data augmentation aplicada en imágenes tomadas aleatoriamente de cada clase de la base de datos de esta investigación

2.6. Modelos CNN pre-entrenados

El algoritmo de entrenamiento y prueba de esta investigación fue desarrollado utilizando el lenguaje Python 3.8.2, y se ejecutó en el servicio de Google Colaboratory, el cual se asemeja a un ambiente similar a Jupyter notebook que no requiere de instalación, y se corre desde la nube. Así mismo, se usó PyTorch 1.4.0 y el paquete de Torchvision 0.5.0 para la obtención de los 32 modelos de CNNs pre-entrenados [42]. Además, se utilizó la librería imbalanced-learn para verificar las métricas de evaluación de aprendizaje y desempeño de los modelos implementados.

Los modelos de CNNs implementados en esta investigación son: Wide_resnet101_2 [43], Resnext101_32x8d [44], Resnext50_32x4d [44], Inception_V3 [45], Mnasnet1_0 [46], Wide_resnet50_2 [43], Mobilenet_v2 [47], Shufflenet_v2_x0_5 [48], Alexnet [49], Squeezenet1_0 [50], Vgg19_bn [51], Vgg13 [51],

Squeezenet1_1 [50], Densenet201 [52], Vgg11 [51], Vgg16_bn [51], Resnet101 [53], Resnet50 [43], Resnet18 [53], Vgg13_bn [51], Densenet169 [52], Vgg19 [51], Vgg11_bn [51], Googlenet [54], Densenet121 [52], Resnet34 [53], Vgg16 [51], Resnet152 [53], Mnasnet1_3 [46], Shufflenet_v2_x1_0 [48], Mnasnet0_5 [46] y Mnasnet0_75 [46].

2.7. Métricas de evaluación de desempeño

Una herramienta útil en la evaluación de desempeño de un modelo de clasificación es la matriz de confusión [55]. Con la ayuda de la matriz de confusión es posible obtener cuatro valores importantes o términos fundamentales. El primero de estos términos es conocido como Verdaderos-Positivos (TP), que se relaciona al número de casos que son predichos como pertenecientes a una clase, y que efectivamente pertenecen a esa clase, es decir, predicciones correctas.

El segundo término es Verdaderos-Negativo (TN), e indica el número de casos predichos como no pertenecientes a una clase, y que verdídicamente no pertenecen a esa clase. Por otro lado, los casos predichos como pertenecientes a una clase y que realmente no pertenecen a la clase son Falsos-positivos (FP). Finalmente, el último término es Falsos-negativos (FN), y se refiere a los elementos predichos como no pertenecientes a una clase, cuando en realidad estos sí pertenecen a esa clase. Estos cuatro términos conforman la matriz de confusión como se ve en la Fig. 4.

Al emplear los términos extraídos de la matriz de confusión se pueden calcular métricas que ayudarán a determinar adecuadamente el desempeño de un clasificador.

Las métricas utilizadas en esta investigación son: la Precisión que mide la habilidad del clasificador en determinar la clase de una imagen correctamente; la Sensibilidad que se refiere a la habilidad del clasificador en predecir las muestras de interés o muestras positivas de la enfermedad a clasificar; la Especificidad que estima el porcentaje de número de casos negativos entre un grupo de casos que no tienen la enfermedad o la afección a clasificar; la F1-score que indica la relación del clasificador en alcanzar mejores resultados en las clases de interés o prioritarias; el

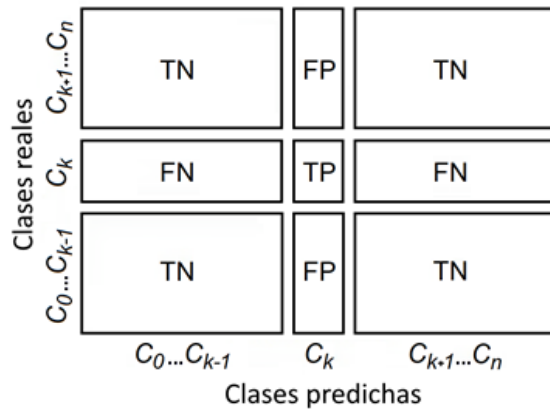


Fig. 4. Elementos en una matriz de confusión con clases C, y números n de clases.

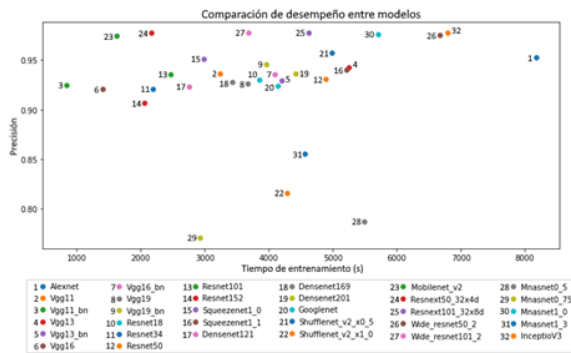


Fig. 5. Precisión alcanzada por cada modelo de CNN en función del tiempo.

Tabla 2. Métricas de evaluación

Métrica	Fórmula
Precisión	$TP / (TP + FP)$
Sensibilidad	$TP / (TP + FN)$
Especificidad	$TN / (TN + FP)$
F1-Score	$\frac{2 * Precisión * Sensibilidad}{Precisión + Sensibilidad}$
G_mean	$\sqrt{(Sensibilidad * Especificidad)}$
IBA	$[1 + (0.1)(Sensibilidad - Especificidad)][G_mean^2]$

G_mean que mide el balance de desempeño sobre las clases con mayor y menor número de

elementos, y el Index Balanced Accuracy (IBA) que calcula el promedio de la exactitud obtenida entre las clases con mayor y menor número de casos predichos correctamente [56].

Las expresiones matemáticas de las métricas implementadas para la evaluación de los modelos de CNNs son enlistadas en la Tabla 2.

3. Resultados y discusiones

En esta investigación se evalúa el desempeño de 32 modelos de CNNs como clasificadores para el diagnóstico de casos de COVID-19, neumonía y normales (pacientes sanos). Las CNNs utilizaron *fine-tuning* e imágenes de rayos X frontales de tórax para su entrenamiento. Las arquitecturas de CNNs pre-entrenadas fueron obtenidas del paquete Torchvision de la librería de PyTorch.

Los resultados de desempeño de los modelos seleccionados son enlistados en la Tabla 3. Cabe resaltar, que todos los modelos fueron entrenados usando 50 épocas de entrenamiento, sin embargo, solo se reporta la época con mejor desempeño de precisión para cada modelo. Además, el tiempo tomado por cada modelo para llegar a la época con mejor desempeño también es reportado.

De acuerdo con los resultados reportados en la Tabla 3, los primeros siete modelos superan el 90% en la métrica IBA y 97% en precisión, lo que nos indica que obtuvieron un excelente desempeño de predicción en todas sus clases con un alto grado de clasificación en cada clase.

Ahora bien, con el fin de determinar cuál es la mejor arquitectura para clasificar imágenes del COVID-19 se requiere de otra métrica, por lo cual utilizaremos la métrica de precisión. En la Fig. 5 se muestra los valores de precisión obtenidos por cada modelo de CNN en función del tiempo de entrenamiento con las imágenes de la base de datos para prueba.

De acuerdo con los resultados presentados en la Tabla 3 y en la Fig. 5, el valor de precisión más alto fue del 97.75%, el cual fue obtenido por tres arquitecturas: Wide_resnet101_2, Resnext101_32x8d y Resnext50_32x4d.

Por otro lado, cabe resaltar los valores obtenidos por la topología Inception_V3 que presenta los mejores resultados en especificidad, G_mean e IBA, lo que indica que obtuvo buenos

Tabla 3. Métricas de desempeño para cada modelo de CNN

Modelo (mejor época/total épocas)	MÉTRICAS DE DESEMPEÑO						Tiempo (min)
	Precisión	Sensibilidad	Especificidad	F1-score	G_mean	IBA	
Wide_resnet101_2 (26/50)	97.75	97.75	96.76	97.75	97.25	94.66	61.45
Resnext101_32x8d (21/50)	97.75	97.75	96.40	97.75	97.06	94.34	76.98
Resnext50_32x4d (10/50)	97.75	97.75	96.07	97.75	96.89	94.04	36.10
Inception_V3 (47/50)	97.69	97.67	97.16	97.67	97.41	94.94	113.28
Mnasnet1_0 (49/50)	97.58	97.50	97.03	97.52	97.26	94.63	95.22
Wide_resnet50_2 (44/50)	97.50	97.50	96.16	97.50	96.82	93.86	111.27
Mobilenet_v2 (9/50)	97.42	97.42	95.64	97.41	96.50	93.29	26.97
Shufflenet_v2_x0_5 (48/50)	95.70	95.67	93.57	95.67	94.57	89.62	83.00
Alexnet (25/50)	95.26	95.26	91.41	95.21	93.20	87.20	136.25
Squeezenet1_0 (33/50)	95.09	95.09	93.06	95.09	94.03	88.60	49.78
Vgg19_bn (39/50)	94.54	94.51	92.31	94.48	93.34	87.34	66.07
Vgg13 (41/50)	94.25	94.26	92.15	94.25	93.15	86.95	87.48
Squeezenet1_1 (20/50)	93.98	93.68	93.80	93.76	93.74	87.86	86.92
Densenet201 (44/50)	93.61	93.59	91.11	93.60	92.29	85.39	73.63
Vgg11 (33/50)	93.60	93.09	93.37	93.19	93.21	86.88	53.98
Vgg16_bn (40/50)	93.56	93.51	91.11	93.49	92.23	85.29	68.18
Resnet101 (26/50)	93.51	93.26	93.01	93.33	93.13	86.75	41.18
Resnet50 (43/50)	93.05	93.09	89.55	93.06	91.20	83.48	81.53
Resnet18 (30/50)	92.98	92.85	90.85	92.88	91.79	84.44	64.13
Vgg13_bn (44/50)	92.92	92.43	92.64	92.55	92.53	85.60	69.98
Densenet169 (35/50)	92.79	92.18	93.36	92.32	92.77	85.97	57.10
Vgg19 (36/50)	92.61	92.26	90.80	92.28	91.46	83.82	61.25
Vgg11_bn (9/50)	92.47	92.35	89.62	92.32	90.87	82.84	13.98
Googlenet (39/50)	92.39	92.26	89.39	92.27	90.73	82.56	69.10
Densenet121 (22/50)	92.30	92.01	90.74	92.09	91.34	83.54	45.83
Resnet34 (20/50)	92.05	91.68	91.00	91.79	91.32	83.45	36.53
Vgg16 (14/50)	92.02	92.01	88.88	92.02	90.34	81.87	23.48
Resnet152 (21/50)	90.64	90.60	84.93	90.45	87.43	76.93	34.27
Mnasnet1_3 (50/50)	85.58	82.95	83.33	82.57	82.43	68.90	76.13
Shufflenet_v2_x1_0 (47/50)	81.59	83.78	77.46	82.62	78.76	64.62	71.52
Mnasnet0_5 (50/50)	78.75	71.96	33.55	60.99	21.43	60.06	91.53
Mnasnet0_75 (28/50)	77.06	80.03	66.30	78.40	69.84	51.17	48.73

Tabla 4. Características de los modelos elegidos

Modelo	Total de parámetros	Peso total (Mb)
Wide_resnet101_2	126,886,696	1,018.61
Resnext101_32x8d	88,791,336	1,111.83
Resnext50_32x4d	25,028,904	457.83

resultados en relación al número de casos positivos y negativos, aunque el tiempo total de entrenamiento es de los más altos. Respecto al tiempo total de entrenamiento, la Vgg11_bn fue

entrenada en un menor lapso de tiempo, aunque sus resultados de precisión de clasificación no son buenos, estando por debajo del 93%.

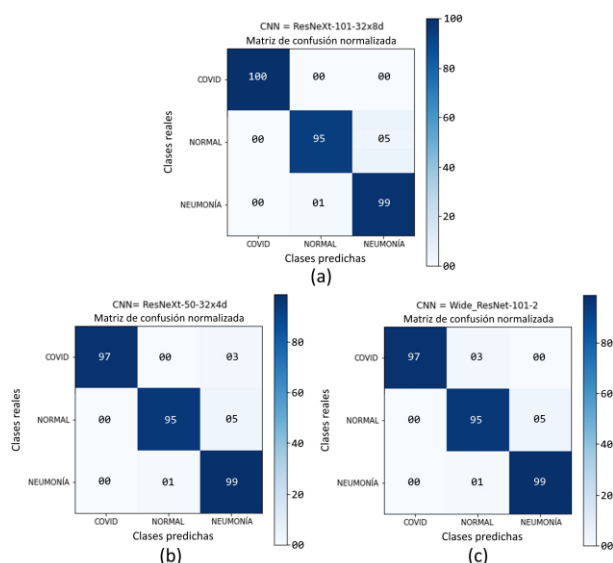


Fig. 6. Matriz de confusión obtenida de las tres mejores arquitecturas: Wide_resnet101_2, Resnext101_32x8d, y Resnext50_32x4d

Adicionalmente, los parámetros a entrenar (pesos y bias) de las tres mejores arquitecturas obtenidas se enlistan en la Tabla 4. Además, se agrega el peso de cada arquitectura. Esta tabla proporciona una idea de la complejidad computacional asociada a los modelos.

Un aspecto interesante es la relación del tiempo de entrenamiento y el peso en disco de los modelos seleccionados, a mayor peso de almacenamiento mayor es el tiempo tomado para el entrenamiento del modelo.

Para profundizar en el análisis del proceso de aprendizaje de las tres arquitecturas se puede analizar la matriz de confusión obtenida para cada uno de los modelos seleccionados. En la Fig. 6 se presenta las matrices de confusión de los modelos.

Para complementar el análisis de las tres mejores arquitecturas, en la Fig. 7 se muestra la precisión obtenida por cada arquitectura tanto con la base de datos de entrenamiento como con la de prueba en función del número de épocas de entrenamiento. Además, al analizar la Fig. 7 se observan los comportamientos de las arquitecturas seleccionadas, las cuales no presentan problemas de overfitting, dado que los

valores de precisión de entrenamiento como de prueba presentan un comportamiento ascendente. De acuerdo con los resultados obtenidos, la arquitectura Wide_resnet101_2 alcanzó el mejor desempeño numérico en todas las métricas, en comparación con las otras dos arquitecturas seleccionadas. No obstante, al analizar su matriz de confusión se nota un error del 3%, el cual lo presenta al confundir pacientes con COVID-19 con pacientes sanos. Esta confusión es algo alarmante, debido a que enviaría a casa a un paciente infectado de COVID-19 sin ningún tratamiento médico o precauciones. Es por tal motivo, que esta arquitectura se considera en el tercer lugar.

Por otra parte, un escenario en donde la arquitectura confunde pacientes con COVID-19 como pacientes con neumonía, este caso sería un poco más certero, pero no alarmante, debido a que los pacientes de neumonía que además presenten fiebre, tos y debilidad respiratoria son casos positivos de COVID-19 [57]. Por tal motivo, la arquitectura Resnext50_32x4d se considera como la segunda mejor arquitectura, la cual confunde los casos de COVID-19 como casos con neumonía en un 3%.

Ahora bien, de acuerdo con los valores obtenidos en la matriz de confusión, la mejor arquitectura para clasificar casos de pacientes con COVID-19, pacientes normales (sanos) y pacientes con neumonía es la Resnext101_32x8d. Esta arquitectura presenta una clasificación perfecta en el diagnóstico de pacientes con COVID-19, y solo presenta pequeños errores entre la clasificación de pacientes sanos (5%) y con neumonía (1%). Es importante resaltar que las arquitecturas ResNeXt son altamente implementadas en tareas de clasificación de imágenes, y han demostrado ser precisas en el diagnóstico de enfermedades médicas, obteniendo excelentes desempeños [58–63].

Esta alta eficiencia en tareas de clasificación de imágenes usando modelos ResNeXt se atribuye a que son arquitecturas altamente modularizadas, y que introducen el concepto de cardinalidad, el cual ha demostrado que a mayor cardinalidad se tiene mayor precisión en las tareas de clasificación de imágenes [58].

Ante la presente emergencia de salud pública generada por el COVID-19, las imágenes

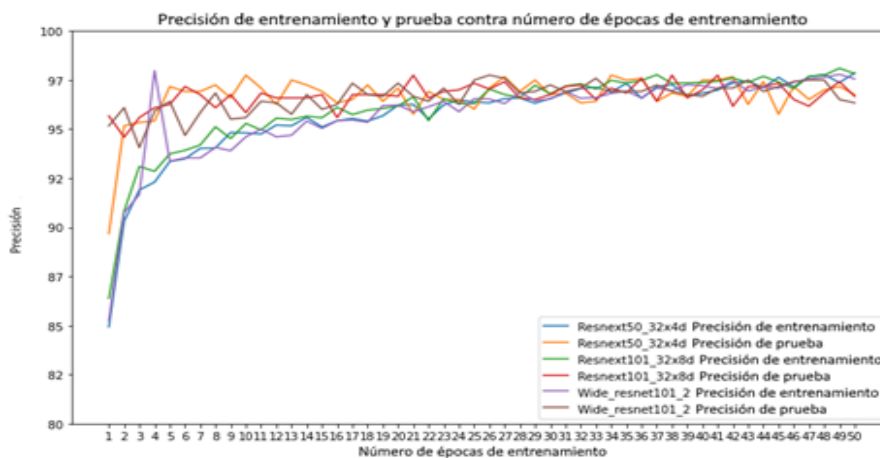


Fig. 7. Precisión obtenida por las tres mejores arquitecturas tanto con la base de datos de entrenamiento como con la de prueba en función del número de épocas de entrenamiento.

radiológicas han demostrado ser de gran utilidad [64–67].

El diagnóstico de pacientes con COVID-19 es una ardua tarea que consume demasiado tiempo y recursos hospitalarios, y a su vez que suele confundirse con casos de neumonía en pacientes asintomáticos, lo que genera resultados inconclusos.

Este tipo de confusiones, no permiten un control adecuado para esta emergencia sanitaria [68], por lo cual se requiere de sistemas inteligentes que ayuden a diagnosticar y confirmar hasta en pacientes asintomáticos, con el fin de tener tratamientos oportunos, un uso correcto de los recursos disponibles y un aislamiento pertinente.

4. Conclusiones

En esta investigación se presenta una comparación exhaustiva entre las principales arquitecturas de CNNs como clasificadores del COVID-19, con el objetivo de determinar que arquitecturas son las más confiables para este diagnóstico, y cuales suelen trabajar de mejor manera (aprender características) con imágenes radiológicas de pacientes con COVID-19. Un total de 32 arquitecturas pre-entrenadas fueron extraídas de la librería Torchvision de PyTorch de código abierto.

Las arquitecturas fueron analizadas y evaluadas en función a su desempeño de aprendizaje al utilizar bases de datos médicas que son totalmente desbalanceadas. Las mejores arquitecturas seleccionadas son capaces de clasificar el COVID-19 con una precisión del 97.75%. Estas arquitecturas presentaron un excelente rendimiento de acuerdo a métricas como la precisión, IBA, entre otras., lo que garantiza que las métricas de rendimiento tuvieron un excelente equilibrio entre las predicciones de las tres clases.

Se espera que esta investigación pueda servir como guía para la comunidad científica y a futuros desarrolladores de sistemas inteligente para el apoyo a médicos expertos en el diagnóstico de COVID-19.

Como trabajo a futuro, se requiere de una colaboración directa con los centros médicos para aumentar el número de imágenes en la base de datos de entrenamiento y mejorar los resultados de clasificación. Además, se pretende crear una interfaz basada en las mejores CNNs, que sea capaz de ayudar a la comunidad médica en el diagnóstico del COVID-19.

Referencias

1. Petrosillo, N., Viceconte, G., Ergonul, O., Ippolito, G., Petersen, E. (2020). COVID-19,

- SARS and MERS: are they closely related? *Clinical Microbiology and Infection*. DOI: 10.1016/j.cmi.2020.03.026.
2. **Lippi, G., Plebani, M., Henry, B.M. (2020).** Thrombocytopenia is associated with severe coronavirus disease 2019 (COVID-19) infections: A meta-analysis. *Clinica Chimica Acta*, Vol. 506, pp. 145–148. DOI: 10.1016/j.cca.2020.03.022.
 3. **Zhou, Z., Guo, D., Li, C., Fang, Z., Chen, L., Yang, R., Zeng, W. (2020).** Coronavirus disease 2019: initial chest CT findings. *European Radiology*. DOI: 10.1007/s00330-020-06816-7.
 4. **Ma, Y., Zhao, Y., Liu, J., He, X., Wang, B., Fu, S., Luo, B. (2020).** Effects of temperature variation and humidity on the death of COVID-19 in Wuhan, China. *Science of The Total Environment*. DOI: 10.1016/j.scitotenv.2020.138226.
 5. **Pan, F., Ye, T., Sun, P., Gui, S., Liang, B., Li, L., Zheng, C. (2020).** Time course of lung changes on chest ct during recovery from 2019 novel coronavirus (COVID-19) pneumonia. *Radiology*. DOI: 10.1148/radiol.2020200370.
 6. **Ye, Z., Zhang, Y., Wang, Y., Huang, Z., Song, B. (2020).** Chest CT manifestations of new coronavirus disease 2019 (COVID-19): A pictorial review. *European Radiology*. DOI: 10.1007/s00330-020-06801-0.
 7. **Sohrabi, C., Alsafi, Z., O'Neill, N., Khan, M., Kerwan, A., Al-Jabir, A., Agha, R. (2020).** World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). *International Journal of Surgery*, Vol. 76, pp. 71–76. DOI: 10.1016/j.ijssu.2020.02.034.
 8. **Kooraki, S., Hosseiny, M., Myers, L., Gholamrezanezhad, A. (2020).** Coronavirus (COVID-19) outbreak: What the department of radiology should know. *Journal of the American College of Radiology*, Vol. 17, No. 4, pp. 447–451. DOI: 10.1016/j.jacr.2020.02.008.
 9. **Li, H., Liu, S.-M., Yu, X.-H., Tang, S.-L., Tang, C.-K. (2020).** Coronavirus disease 2019 (COVID-19): Current status and future perspective. *International Journal of Antimicrobial Agents*. DOI: 10.1016/j.ijantimicag.2020.105951.
 10. **World Health Organization (2020).** Novel Coronavirus (2019-nCoV) situation reports - 51. World Health Organization. <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports>.
 11. **World Health Organization (2020).** Coronavirus disease (COVID-19) pandemic. World Health Organization, Geneva. <https://www.who.int/emergencies/diseases/novel-coronavirus-2019>.
 12. **Wu, Z., McGoogan, J.M. (2020).** Characteristics of and important lessons from the coronavirus disease 2019 (COVID-19) Outbreak in China: Summary of a Report of 72 314 cases from the Chinese center for disease control and prevention. *JAMA*. DOI: 10.1001/jama.2020.2648.
 13. **Tian, S., Hu, W., Niu, L., Liu, H., Xu, H., Xiao, S.-Y. (2020).** Pulmonary pathology of early-phase 2019 novel coronavirus (COVID-19) pneumonia in two patients with lung cancer. *Journal of Thoracic Oncology*, Vol. 15, No. 5, pp. 700–704. DOI: 10.1016/j.jtho.2020.02.010.
 14. **Montero-Feijoo, A., Maseda, E., Adalia-Bartolomé, R., Aguilar, G., González de Castro, R., Gómez-Herreras, J. I., Varela Durán, M. (2020).** Recomendaciones prácticas para el manejo perioperatorio del paciente con sospecha o infección grave por coronavirus SARS-CoV-2. *Revista Española de Anestesiología y Reanimación*. DOI: 10.1016/j.redar.2020.03.003.
 15. **Hajian, A., Styles, P. (2018).** Application of soft computing and intelligent methods in geophysics. Springer.
 16. **Liu, M., Shi, J., Li, Z., Li, C., Zhu, J., Liu, S. (2017).** Towards Better Analysis of Deep Convolutional Neural Networks. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 23, No. 1, pp. 91–100. Presented at the IEEE Transactions on Visualization and Computer Graphics. DOI: 10.1109/TVCG.2016.2598831.

17. **Tan, M., Yuan, S., Su, Y. (2018).** Content-Based Similar Document Image Retrieval Using Fusion of CNN Features. In **Huet, B., Nie, L., Hong, R. editors**, *Internet Multimedia Computing and Service*, pp. 260–270, Singapore Springer. DOI: 10.1007/978-981-10-8530-7_25.
18. **Hamm, C.A., Wang, C.J., Savic, L.J., Ferrante, M., Schobert, I., Schlachter, T., Letzen, B. (2019).** Deep learning for liver tumor diagnosis part I: Development of a convolutional neural network classifier for multi-phasic MRI. *European Radiology*, Vol. 29, No. 7, pp. 3338–3347. DOI: 10.1007/s00330-019-06205-9.
19. **Wang, C.J., Hamm, C.A., Savic, L.J., Ferrante, M., Schobert, I., Schlachter, T., Letzen, B. (2019).** Deep learning for liver tumor diagnosis part II: Convolutional neural network interpretation using radiologic imaging features. *European Radiology*, 29(7), pp. 3348–3357. DOI: 10.1007/s00330-019-06214-8.
20. **Özyurt, F., Tuncer, T., Avci, E., Koç, M., Serhatlioğlu, İ. (2019).** A novel liver image classification method using perceptual hash-based convolutional neural network. *Arabian Journal for Science and Engineering*, Vol. 44, No. 4, 3173–3182. DOI: 10.1007/s13369-018-3454-1.
21. **He, Y., Guo, J., Ding, X., van Ooijen, P.M.A., Zhang, Y., Chen, A., Xie, X. (2019).** Convolutional neural network to predict the local recurrence of giant cell tumor of bone after curettage based on pre-surgery magnetic resonance images. *European Radiology*, Vol. 29, No. 10, pp. 5441–5451. DOI: 10.1007/s00330-019-06082-2.
22. **Ortiz-Rodriguez, J.M., Guerrero-Mendez, C., Martinez-Blanco, M.R., Castro-Tapia, S., Moreno-Lucio, M., Jaramillo-Martinez, R., Garcia, J.A.B. (2017).** Breast Cancer Detection by Means of Artificial Neural Networks. *Advanced Applications for Artificial Neural Networks*. DOI: 10.5772/intechopen.71256.
23. **Rouhi, R., Jafari, M., Kasaei, S., Keshavarzian, P. (2015).** Benign and malignant breast tumors classification based on region growing and CNN segmentation. *Expert Systems with Applications*, Vol. 42, No. 3, pp. 990–1002. DOI: 10.1016/j.eswa.2014.09.020.
24. **Borges-Sampaio, W., Moraes-Diniz, E., Corrêa-Silva, A., Cardoso de Paiva, A., Gattass, M. (2011).** Detection of masses in mammogram images using CNN, geostatistic functions and SVM. *Computers in Biology and Medicine*, Vol. 41, No. 8, pp. 653–664. DOI: 10.1016/j.compbiomed.2011.05.017.
25. **Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S. (2017).** Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, Vol. 542, No. 7639, pp. 115–118. DOI: 10.1038/nature21056.
26. **Dorj, U.-O., Lee, K.-K., Choi, J.-Y., Lee, M. (2018).** The skin cancer classification using deep convolutional neural network. *Multimedia Tools and Applications*, Vol. 77, No. 8, pp. 9909–9924. DOI: 10.1007/s11042-018-5714-1.
27. **Tschandl, P., Rosendahl, C., Akay, B.N., Argenziano, G., Blum, A., Braun, R.P., Kittler, H. (2019).** Expert-level diagnosis of nonpigmented skin cancer by combined convolutional neural networks. *JAMA Dermatology*, Vol. 155, No. 1, pp. 58–65. DOI: 10.1001/jamadermatol.2018.4378.
28. **Zhang, Y., van der Werf, N.R., Jiang, B., van Hamersvelt, R., Greuter, M.J.W., Xie, X. (2020).** Motion-corrected coronary calcium scores by a convolutional neural network: A robotic simulating study. *European Radiology*, Vol. 30, No. 2, pp. 1285–1294. DOI: 10.1007/s00330-019-06447-7.
29. **Khan, A.I., Shah, J.L., Bhat, M.M. (2020).** CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Computer Methods and Programs in Biomedicine*, Vol. 196. DOI: 10.1016/j.cmpb.2020.105581.
30. **Moutounet-Cartan, P.G.B. (2020).** Deep convolutional neural networks to diagnose COVID-19 and other pneumonia diseases from posteroanterior chest x-rays. arxiv.org/abs/2005.00845.

31. **Li, Z., Wang, C., Han, M., Xue, Y., Wei, W., Li, L.-J., Fei-Fei, L. (2018).** Thoracic disease identification and localization with limited supervision.
32. **Lu, L., Wang, X., Carneiro, G., Yang, L. (2019).** Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics. Springer Nature.
33. **Kellenberger, B., Marcos, D., Tuia, D. (2018).** Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning. *Remote Sensing of Environment*, Vol. 216, pp. 139–153. DOI: 10.1016/j.rse.2018.06.028.
34. **Johnson, J.M., Khoshgoftaar, T.M. (2019).** Survey on deep learning with class imbalance. *Journal of Big Data*, Vol. 6, No. 1, pp. 27. DOI: 10.1186/s40537-019-0192-5.
35. **Mooney, P. (2020).** Chest x-ray images (Pneumonia). Kaggle.
36. **Cohen, J.P., Morrison, P., Dao, L., Roth, K., Duong, T.Q., Ghassemi, M. (2020).** COVID-19 Image data collection: Prospective predictions are the future. *Arxiv.org/abs/2006.11988*.
37. **Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012).** ImageNet classification with deep convolutional neural networks. In **Pereira, F., Burges, C.J.C.L., Bottou, K., Weinberger, Q. editors**, *Advances in Neural Information Processing Systems*, Vol. 25, pp. 1097–1105, Curran Associates, Inc.
38. **Guerrero-Mendez, C., Saucedo-Anaya, T., Moreno, I., Araiza-Esquivel, M., Olvera-Olvera, C., Lopez-Betancur, D. (2020).** Digital holographic interferometry without phase unwrapping by a convolutional neural network for concentration measurements in liquid samples. *Applied Sciences*, Vol. 10, No. 14. DOI: 10.3390/app10144974.
39. **Shorten, C., Khoshgoftaar, T.M. (2019).** A survey on image data augmentation for deep learning. *Journal of Big Data*, Vol. 6, No. 1, pp. 60. DOI: 10.1186/s40537-019-0197-0.
40. **Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., Batra, D. (2016).** Reducing overfitting in deep networks by decorrelating representations. *arxiv.org/abs/1511.06068*.
41. **Mikołajczyk, A., Grochowski, M. (2018).** Data augmentation for improving deep learning in image classification problem. *International Interdisciplinary PhD Workshop (IIPhDW)*, pp. 117–122. Presented at the 2018 International Interdisciplinary PhD Workshop (IIPhDW). DOI: 10.1109/IIPHDW.2018.8388338.
42. **PyTorch. (2019).** Torchvision.models. PyTorch master documentation.
43. **Zagoruyko, S., Komodakis, N. (2017).** Wide Residual Networks. *arXiv:1605.07146*. Retrieved from <http://arxiv.org/abs/1605.07146>
44. **Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K. (2017).** Aggregated Residual Transformations for Deep Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995. Presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR`17). DOI: 10.1109/CVPR.2017.634.
45. **Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2015).** Rethinking the Inception Architecture for Computer Vision. *arxiv.org/abs/1512.00567*.
46. **Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V. (2019).** MnasNet: Platform-Aware Neural Architecture Search for Mobile. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828.
47. **Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C. (2018).** MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520.
48. **Ma, N., Zhang, X., Zheng, H.-T., Sun, J. (2018).** ShuffleNet V2: Practical guidelines for efficient cnn architecture design. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 116–131.
49. **Krizhevsky, A. (2014).** One weird trick for parallelizing convolutional neural networks. *arxiv.org/abs/1404.5997*.

50. **Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K. (2016).** SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arxiv.org/abs/1602.07360*.
51. **Simonyan, K., Zisserman, A. (2015).** Very Deep Convolutional Networks for Large-Scale Image Recognition. *arxiv.org/abs/1409.1556*.
52. **Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q. (2017).** Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708.
53. **He, K., Zhang, X., Ren, S., Sun, J. (2016).** Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
54. **Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015).** Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9.
55. **Fawcett, T. (2006).** An introduction to ROC analysis. *Pattern Recognition Letters*, Vol. 27, No. 8, pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.
56. **Akosa, J. (2017).** Predictive accuracy: A misleading performance measure for highly imbalanced data. *Proceedings of the SAS Global Forum*, pp. 12.
57. **Pan, Y., Guan, H. (2020).** Imaging changes in patients with 2019-nCoV. *European Radiology*. DOI: 10.1007/s00330-020-06713-z.
58. **Zhang, G., Yang, Z., Gong, L., Jiang, S., Wang, L., Zhang, H. (2020).** Classification of lung nodules based on CT images using squeeze-and-excitation network and aggregated residual transformations. *La radiologia medica*, Vol. 125, No. 4, pp. 374–383. DOI: 10.1007/s11547-019-01130-9.
59. **Bajwa, M.N., Muta, K., Malik, M.I., Siddiqui, S.A., Braun, S.A., Homey, B., Ahmed, S. (2020).** Computer-Aided Diagnosis of Skin Diseases Using Deep Neural Networks. *Applied Sciences*, Vol. 10, No. 7, pp. 2488. DOI: 10.3390/app10072488.
60. **Zhou, H., Xie, F., Jiang, Z., Liu, J., Wang, S., Zhu, C. (2017).** Multi-classification of skin diseases for dermoscopy images using deep learning. *IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1–5. DOI: 10.1109/IST.2017.8261543.
61. **Koné, I., Boulmane, L. (2018).** Hierarchical ResNeXt models for breast cancer histology image classification. **Campilho, A., Karray, F., ter Haar Romeny, B. editors**, *Image Analysis and Recognition*, pp. 796–803. Cham: Springer International Publishing. DOI: 10.1007/978-3-319-93000-8_90.
62. **Li, H., Cheng, J.-Z., Chou, Y.-H., Qin, J., Huang, S., Lei, B. (2019).** AttentionNet: Learning where to focus via attention mechanism for anatomical segmentation of whole breast ultrasound images. *IEEE 16th International Symposium on Biomedical Imaging (ISBI'19)*, pp. 1078–1081. DOI: 10.1109/ISBI.2019.8759241.
63. **Lee, K., Ko, J.G., Yoo, W. (2019).** An intensive study of backbone and architectures with test image augmentation and box refinement for object detection and segmentation. *International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 673–677. DOI: 10.1109/ICTC46691.2019.8939591.
64. **Himoto, Y., Sakata, A., Kirita, M., Hiroi, T., Kobayashi, K., Kubo, K., Umeoka, S. (2020).** Diagnostic performance of chest CT to differentiate COVID-19 pneumonia in non-high-epidemic area in Japan. *Japanese Journal of Radiology*. DOI: 10.1007/s11604-020-00958-w.
65. **Chung, M., Bernheim, A., Mei, X., Zhang, N., Huang, M., Zeng, X., Shan, H. (2020).** CT imaging features of 2019 novel coronavirus (2019-nCoV). *Radiology*, Vol. 295, No. 1, pp. 202–207. DOI: 10.1148/radiol.202000230.
66. **Kim, H. (2020).** Outbreak of novel coronavirus (COVID-19): What is the role of radiologists? *European Radiology*. DOI: 10.1007/s00330-020-06748-2.

67. **Li, K., Fang, Y., Li, W., Pan, C., Qin, P., Zhong, Y., Li, S. (2020).** CT image visual quantitative evaluation and clinical classification of coronavirus disease (COVID-19). *European Radiology*. DOI: 10.1007/s00330-020-06817-6.
68. **Wu, X., Zhao, Y., Radev, D., Malhotra, A. (2020).** Identification of patients with carotid

stenosis using natural language processing. *European Radiology*. DOI: 10.1007/s00330-020-06721-z.

*Article received on 12/07/2020; accepted on 24/03/2021.
Corresponding author is Daniela Lopez-Betancur.*