

Detección Rápida de Retinopatía Diabética Mediante Aprendizaje Profundo. Rapid Detection of Diabetic Retinopathy Through Deep Learning.

Darwin Patiño-Pérez, Ph.D.¹, Celia Munive-Mora, BS^{1,2,3}, Luis Chóez-Acosta, MSc¹, Alex Collantes-Farah, MSIG¹, Miguel Molina-Calderón, MSIG¹, Zynnica Reyessan-Sánchez, MSI⁴
¹Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Física, Ecuador, darwin.patinop@ug.edu.ec, celia.munivem@ug.edu.ec, luis.choeza@ug.edu.ec, alex.collantesf@ug.edu.ec, miguel.molinac@ug.edu.ec,
²St Luke's University Hospital Network, PA, United States, celia.munive@sluhn.org,
³De Sales University, Center Valley, PA, United States, cm3877@desales.edu,
⁴Universidad de Guayaquil, Facultad de Ciencias Médicas, Ecuador, zynnica.reyessan@ug.edu.ec

Resumen.— La enfermedad llamada retinopatía diabética (RD) ha generado una serie de estudios e investigaciones que buscan poder detectarla a tiempo, con la finalidad de contrarrestar los daños que esta puede causar a la retina en el ojo de las personas y evitar la ceguera; para su diagnóstico se requiere que un especialista la detecte mediante el análisis de una imagen ocular. Puesto que se requiere de una rápida detección, se planteó una investigación que fijo como objetivo implementar y analizar distintas técnicas basadas en aprendizaje automático o Machine Learning (ML), apropiadas para solucionar el presente problema y evaluar cuál de ellas es la más eficiente para la detección y clasificación de RD, destacando entre ellas las basadas en aprendizaje profundo o deep learning que actualmente se han convertido en un componente de la inteligencia artificial más utilizado en el campo de la biomedicina. Se determinó que uno de los modelos de predicción basado en aprendizaje profundo o deep learning, aprendió a realizar una rápida detección de la RD con detalles muy finos, como microaneurismas y algunas características más grandes como exudados duros, al analizar todas las retinografías mediante el proceso de clasificación alcanzándose el 85.9% de exactitud en la detección de la retinopatía diabética.

Palabras Claves: Retinopatía diabética, Aprendizaje automático, Aprendizaje profundo, Redes neuronales artificiales, Redes neuronales convolucionales.

Abstract. — The disease called diabetic retinopathy (DR) has generated a series of studies and investigations that seek to be able to detect it in time, to counteract the damage that it can cause to the retina in the eye of people and prevent blindness; For its diagnosis, a specialist is required to detect it by analyzing an ocular image. Since rapid detection is required, an investigation was proposed with the objective of implementing and analyzing different techniques based on automatic learning or Machine Learning (ML), appropriate to solve this problem and evaluate which of them is the most efficient for the detection and classification of DR, highlighting among them those based on deep learning or deep learning that currently have become a component of artificial intelligence most used in the field of biomedicine.

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).

It was determined that one of the deep learning-based prediction models learned to perform rapid DR detection with very fine details, such as microaneurysms and some larger features such as hard exudates, by analyzing all retinographies using the process classification, reaching 85.9% accuracy in the detection of diabetic retinopathy.

Keywords: Diabetic retinopathy, Machine learning, Deep learning, Artificial neural networks, Convolutional neural networks.

I. INTRODUCCION

La retinopatía es una enfermedad del ojo que afecta la retina, la capa de células sensibles a la luz en la parte posterior del ojo que es esencial para la visión según Fig.1. La retinopatía puede tener diferentes causas y grados de gravedad, pero en general se caracteriza por cambios en la retina que pueden afectar la visión.

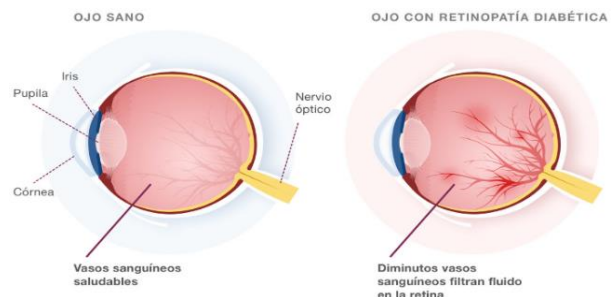


Fig. 1. Ojos Con y Sin Retinopatía

La retinopatía diabética (RD) es una de las causas principales de ceguera a nivel mundial. La retinopatía diabética es un conjunto de enfermedades, las cuales son el causante del impedimento de la circulación normal de la sangre, principalmente en el centro de la retina, esta es una complicación frecuente tanto en el tipo de diabetes 1 y tipo de diabetes 2. Esta enfermedad aparece en diabéticos crónicos y si la diabetes es mal controlada podría agravar mucho la condición visual, dando lugar a la cuarta causa de ceguera en el mundo [1].

El progreso de la retinopatía diabética va desde los estados más benignos hasta los más severos cuando no se aplica una intervención médica apropiada [2].

El objetivo de este estudio es utilizar inteligencia artificial implementando modelos de aprendizaje automático que previamente se analizan para determinar las técnicas y métodos más apropiados para el resolver el diagnóstico tardío de la retinopatía diabética.

Estos modelos utilizaran un dataset de Asia Pacific Tele-Ophthalmology Society (APTOS) publicado en la plataforma Kaggle con la finalidad de tener las retino grafías adecuadas para detectar el nivel de Retinopatía Diabética. Estos niveles se dividirán de 5 formas, las cuales son: No DR, Mild, Moderate, Severe, Proliferative; siendo el nivel más bajo No DR (No posee síntomas) y el nivel más alto “Proliferative” el grado de afectación más severo de la retinopatía diabética.

Considerando que la inteligencia artificial (IA) es una rama de la informática que se enfoca en el diseño y desarrollo de algoritmos y sistemas que pueden realizar tareas que normalmente requieren inteligencia humana, como el aprendizaje, la percepción, el razonamiento y la resolución de problemas[3]. La IA utiliza técnicas de la ciencia de la computación, la estadística, la lógica y la matemática para crear sistemas que pueden aprender de manera autónoma a partir de los datos y mejorar su desempeño con el tiempo.

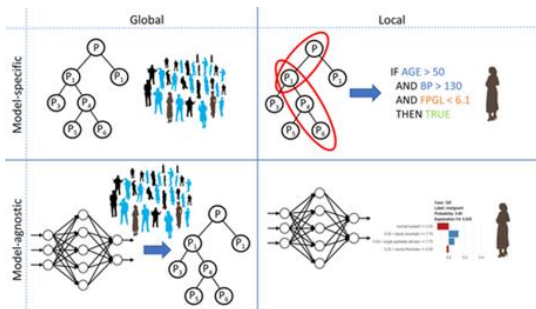


Fig. 2. Modelos de Deep Learning usados en el área de la salud

Puesto que los modelos de predicción son aquellos que proporcionan predicciones para un resultado dicotómico (resultados diferentes pero complementarios), en la Fig.2 se aprecian los modelos más relevantes que han sido usados en aplicaciones médicas [4]. En vista que el aprendizaje automático o Machine Learning, es uno de los campos o perspectivas principales de la *Artificial Intelligence*, la cual consiste en que los máquinas u ordenadores tengan la capacidad de aprender o analizar como lo haría una inteligencia humana [5], se ha realizado un estudio del comportamiento de algunos modelos basados en aprendizaje profundo, permitiendo que los modelos computacionales que se componen de múltiples capas de procesamiento aprendan de las representaciones de los datos en los múltiples niveles de abstracción[6].

En el presente estudio se evaluará el comportamiento de los modelos de redes neuronales artificiales según la Fig.3 en especial las denominadas convolucionales como ResNet18, ResNet50 así como el modelo predictivo que hace uso de *clustering* como *K-means* y otro que hace uso de árboles de decisión como *Random Forest*.

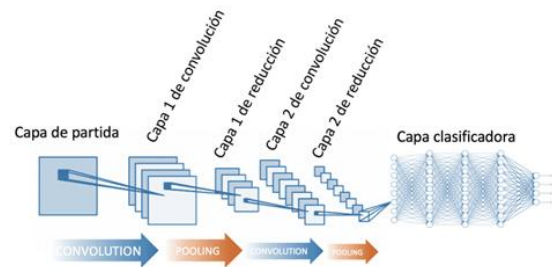


Fig. 3. Estructura de una CNN

II. MATERIALES Y METODOS

A. El Dataset

La unidad de estudio que se utilizará en la presente investigación consta de 3662 imágenes de retino grafías recolectadas de la comunidad en línea de *Kaggle* APTOS 2019 BD, las cuales se encuentran clasificadas de acuerdo con el nivel que poseen, es decir, si presentan poco, mucho o nada de retinopatía diabética.

Como población se cuenta con un total de 3662 imágenes médicas dentro de la comunidad en línea de *Kaggle*, las cuales se encuentran segmentadas de acuerdo con el siguiente cuadro:

TABLA I
RESUMEN DE SEGMENTACION DE IMAGENES MEDICAS

Clase	Nombre	Núm. Imágenes	Porcentaje
0	No DR	1805	49,29%
1	Mild	370	10,10%
2	Moderate	999	27,28%
3	Severe	193	5,27%
4	Proliferative	295	8,06%
Total		3662	100%

B. Tratamiento y Ajuste de Imágenes

Debido a que algunas imágenes del Data Set no fueron obtenidas en las mismas condiciones y con el mismo equipo, muchas de estas imágenes tienen diferentes porciones de la retina en distintas alineaciones y calidad, debido a esto se implementó técnicas de procesamiento de imágenes utilizando la biblioteca *OpenCV* en *Python*, con la finalidad de que los modelos puedan aprender las características de la retina de una manera más eficiente.

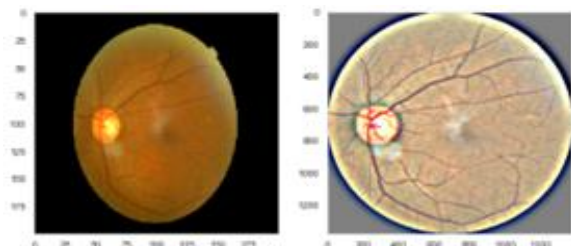


Fig.4. Antes/Después del Tratamiento

Se aplico el desenfoque gaussiano y un recorte circular, en la cual la imagen se traza un contorno en la imagen y se aplica un filtro gaussiano, este realiza la eliminación de los componentes de alta frecuencia, para que se puedan apreciar con más claridad las características de cada imagen.

C. Descripción de las Variables

En la Tabla-I se puede apreciar los diferentes niveles de RD los cuales son: el nivel 0 (cero) No DR esto quiere decir no patológico o que no posee retinopatía diabética. El nivel 1 (uno) Mild es retinopatía diabética no patológica leve, esto quiere decir que se han localizado micro aneurismas (manchas rojas) las cuales son la fuente del exudado duro (manchas amarillas de alto contraste). El nivel 2 (dos) Moderate, es la etapa en la cual se puede llegar a perder la capacidad del transporte de la sangre por medio de los vasos sanguíneos provocado por su distorsión e hinchazón. El nivel 3 (tres) es Severe, en esta etapa es impedido el suministro de sangre hacia la retina debido al bloqueo de los vasos sanguíneos. El nivel 4 (cuatro) Proliferative, es la etapa más avanzada, donde las características del crecimiento, que son secretadas por la retina activan la proliferación de nuevos vasos sanguíneos, los cuales crecen en el interior de la retina en un gel vítreo, llenando el ojo. Cada etapa posee sus características y propiedades propias, por lo que los médicos cuando realicen su análisis probablemente no tomen en cuenta algunos detalles, y por lo cual puede llegar a existir la probabilidad de que realicen un diagnóstico errado.

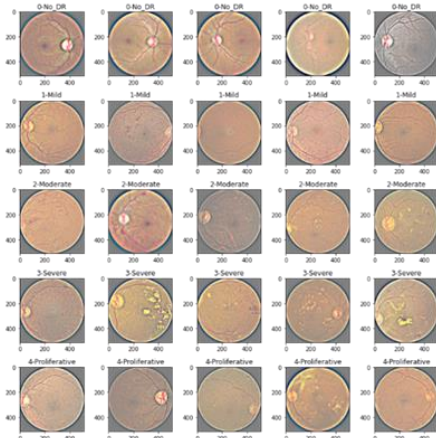


Fig. 5. Tipos de Retinopatía Diabética

D. Proceso de Análisis de Datos

En primera instancia se descargó y subió las imágenes a Google Drive, después se clasifico las imágenes en un directorio correspondiente a cada nivel de Retinopatía Diabética, con el fin de que estas seas diferenciadas correctamente. El elemento o punto clave que nos dará la certificación de si el resultado es correcto o no serán los datos de cada retino grafía que ya se encuentra dentro del almacenamiento en la nube. Con la finalidad de que los algoritmos que están Google Colab puedan acceder a esta información para su entrenamiento. Si la comparación entre el resultado obtenido entre el modelo o algoritmo clasificador y la información mostrada en la nube es la misma, se puede entonces deducir que el resultado es correcto, en caso de que no se cumpla esta condición el

resultado obtenido no fue correcto. Una vez que se haya realizado el entrenamiento de cada uno de los algoritmos y se pueda apreciar sus resultados, se realizara la comparación de cada uno de ellos para saber cuál es el más eficiente para la solución del problema propuesto.

E. Los modelos de ML usados

ResNet-18.- Es una red convolucional de 18 capas de profundidad según la Fig.6. El rendimiento de ResNet-18 puede retener las características de baja escala por la razón de que es superficial. Por lo tanto, se usa el modelo ResNet-18 como un extractor de funciones (codificador). La arquitectura ResNet-18 consta de 16 capas de convolución, 2 capas de reducción de resolución y algunas capas completamente conectadas [7].

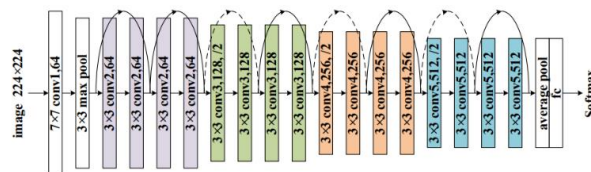


Fig. 6. Arquitectura de la ResNet18

ResNet-50.- Es una red neuronal artificial convolucional que posee 50 capas de profundidad según la Fig.7. Esta puede cargar una versión pre-entrenada en más de un millón de imágenes de la base de datos de Imagenet. Esta posee 48 capas de convolución junto a 1 de MaxPool y 1 capa de grupo promedio posee $3,8 \times 10^9$ operaciones de puntos flotantes [8].

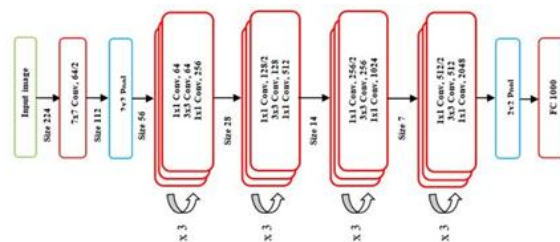


Fig. 7. Arquitectura de la ResNet50

Random Forest.- Este algoritmo es un clasificador basado en el aprendizaje conjunto que construye una colección de árboles individuales "independientes y no idénticos", que se centra en la "aleatorización". El método RF consta de un parámetro que consta de un vector aleatorio, una selección aleatoria de las características de las muestras y una selección aleatoria del conjunto de entrenamiento que es el subconjunto del conjunto de datos dado [9].

K-means.- Es un algoritmo bastante amplio que usan los investigadores en campos como el climatológico, además es uno de los algoritmos más notables dentro del machine learning ya que puede realizar análisis de variables y datos de meteorología entre otros [10].

Red Neuronal Convolutacional (CNN)

Las redes neuronales convolucionales, son algoritmos de inteligencia artificial, se basan en redes neuronales multicapa que aprenden características relevantes de las

imágenes, que pueden realizar varias tareas como clasificación, detección y segmentación de objetos [13]; están dentro del campo del *deep learning*[14].

Para la implementación de los algoritmos de *machine learning*, se usó Python que tiene las librerías para la creación de una gran gama de modelos de Deep Learning usando como backend otras librerías como TensorFlow, Keras y Theano [15].

Además, se ha usado el modelo PCA, que es una técnica estadística muy útil que ha encontrado aplicación en el campo del reconocimiento facial y la compresión de imágenes, es una técnica muy común para encontrar patrones en datos de gran dimensión [16].

En las redes neuronales convolucionales (CNN), se ha usado la función de activación ReLU, cuya función es la de aumentar la propiedad no lineal de activación sin que se modifiquen los cuerpos receptivos de dichas capas convolucionales [17].

Matriz de confusión

La Matriz de confusión es la que se encarga de la detección de errores permitiendo un análisis descriptivo y analítico, en donde se puede visualizar las diferentes asignaciones correctas o incorrectas [18].

RESNET (Red Residual)

A medida que la CNN se hace más profunda, tiende a producirse un gradiente de fuga que afecta negativamente al rendimiento de la red. La red residual funciona añadiendo mapeos de identidad en la parte superior de la red neuronal.

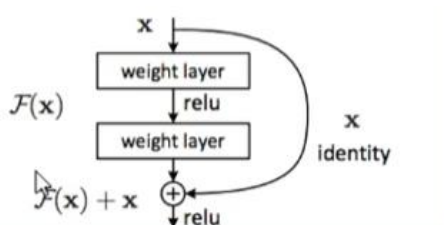


Fig.8. Estructura de una CNN

Convoluciones

Una convolución en una imagen por definición es una transformación píxel por píxel, realizada aplicando a una imagen alguna transformación definida por un conjunto de pesos, también conocida como filtro [19].

Capas convolucionales

Una capa convolucional consta de un conjunto de filtros que se pueden aprender. Cada filtro es espacialmente pequeño en ancho y alto, pero se extiende por toda la profundidad del volumen de entrada [19].

Submapping

La función de una capa de agrupación, también conocida como capa de submuestreo, es reducir progresivamente el tamaño espacial de la representación, para reducir la cantidad de parámetros y cómputo en la red [19].

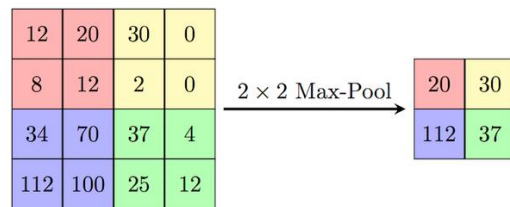


Fig. 9. Submapping

K-fold

El KCV consiste en dividir un conjunto de datos en k subconjuntos; luego, iterativamente, algunos de ellos se utilizan para aprender el modelo, mientras que los otros se explotan para evaluar su desempeño [20].

Pooling layer

Capa usada para reducir las dimensiones, con el objetivo de disminuir los tiempos de procesamiento reteniendo la información más importante, dando como resultado una reducción sobre el sobreajuste (overfitting) en la red y algo de invarianza a la traslación [21].

Retino grafías

La retino grafía es una prueba diagnóstica que toma una imagen de modo no invasivo e indoloro en color del fondo de ojo [22].

III. RESULTADO Y DISCUSIÓN

ResNet-18

Las líneas de código según Fig.10, reflejan el desarrollo del algoritmo de ResNet18, este es de mucha ayuda para la solución del proyecto actual, debido que la ResNet18 trata de conseguir un vector propio que contiene múltiples probabilidades, que se utilizan para indicar que la imagen de entrada pertenece a una cierta clase, y con eso la clase con probabilidad más alta será la salida resultante, de esa manera se puede lograr clasificar los diferentes tipos de retinopatía diabética.

```

input_shape = (256,256,3)
#Input tensor shape
X_input = Input(input_shape)
#Zero-padding
X = ZeroPadding2D((3,3))(X_input)
# 1 - stage
X = Conv2D(64, (7,7), strides=(2,2), name = 'conv1', kernel_initializer= glorot_uniform(seed = 0))(X)
X = BatchNormalization(axis = -3, name = 'bn_conv1')(X)
X = Activation('relu')(X)
X = MaxPooling2D((3,3), strides=(2,2))(X)
# 2- stage
X = res_block(X, filter= [64,64,256], stage= 2)
# 3- stage
X = res_block(X, filter= [128,128,512], stage= 3)
# 4- stage
X = res_block(X, filter= [256,256,1024], stage= 4)
#Average Pooling
X = AveragePooling2D((2,2), name = 'Average_Pooling')(X)
#Final layer
X = Flatten()(X)
X = Dense(5, activation = 'softmax', name = 'Dense_final', kernel_initializer= glorot_uniform(seed=0))(X)
model = Model( inputs=X_input, outputs = X, name = 'Resnet18')
model.summary()

```

Fig.10. Código de ResNet18 en Tensor Flow 2.0

En la Fig.11 se aprecian las etapas del entrenamiento y la cantidad de veces que el modelo será ejecutado con la finalidad de que la red pueda llegar a realizar la detección y clasificación de manera eficaz.

```

model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
#using early stopping to exit training if validation loss is not decreasing even after certain epochs (patience)
earlystopping = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=15)

#save the best model with lower validation loss
checkpointer = ModelCheckpoint(filepath="ResNet18\weights.hdf5", verbose=1, save_best_only=True)
history = model.fit(train_generator, steps_per_epoch = train_generator.n // 32, epochs = 2,
                    validation_data= validation_generator, validation_steps= validation_generator.n // 32,
                    callbacks=[checkpointer , earlystopping])

```

Fig.11. Algoritmo para el entrenamiento de la ResNet18

En la Fig.12. se puede apreciar que el algoritmo termino su ejecución en la época 2 y dio los siguientes resultados:

- Perdida en el entrenamiento (loss): 83%
- Precisión en el entrenamiento (accuracy): 68%
- Perdida en la validación (loss): 108%
- Precisión en la validación (accuracy): 26%

```

Epoch 1/2
77/77 [#####] - ETA: 0s - loss: 1.9494 - accuracy: 0.5980
Epoch 1: val_loss improved from inf to 1.95709, saving model to ResNet18\weights.hdf5
77/77 [#####] - 145s 20/step - loss: 1.9494 - accuracy: 0.5980 - val_loss: 1.9571 - val_accuracy: 0.2668
Epoch 2/2
77/77 [#####] - ETA: 0s - loss: 0.8374 - accuracy: 0.6851
Epoch 2: val_loss improved from 1.95709 to 1.86122, saving model to ResNet18\weights.hdf5
77/77 [#####] - 64s 820ms/step - loss: 0.8374 - accuracy: 0.6851 - val_loss: 1.8612 - val_accuracy: 0.2620

```

Fig.12. Finalización del Entrenamiento de ResNet18

Esto nos da a conocer que el modelo esta sobre ajustando los datos. En Fig.13 y Fig.14 se pueden apreciar los gráficos de pérdidas y precisión del modelo a través de las épocas o etapas del modelo.

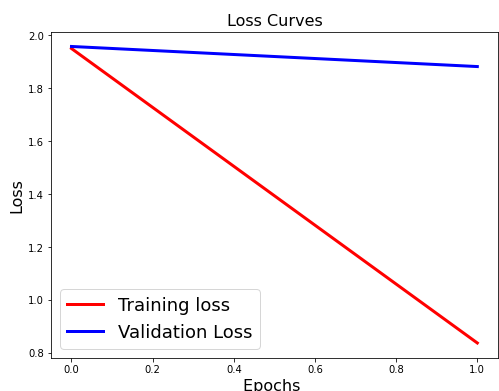


Fig.13. Pérdida del entrenamiento y validación

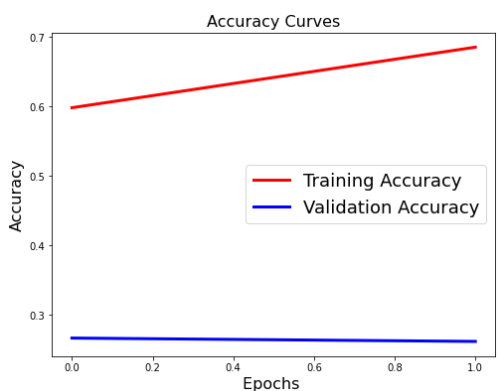


Fig.14. Precisión del entrenamiento y validación.

En la Fig.15. se puede apreciar que la mayoría de las muestras se clasificaron en la case 0 (No DR) con un valor de 330, la clase 1 (Mild) con un valor de 19, la clase 2 (Moderate) con un valor de 87, mientras que la clase 3 (Severe) con un valor de 20 y 35 (Proliferative).

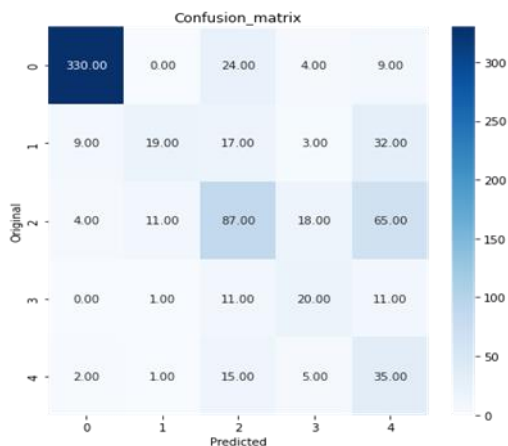


Fig. 15. Matriz de confusión de las clasificaciones con ResNet-18

ResNet-50

En Fig.16 se muestran las líneas de código, en las cuales está el desarrollo del algoritmo de Resnet50, el cual permite procesar y clasificar las imágenes.

```

def create_model(input_shape, n_out):
    input_tensor = Input(shape=input_shape)
    #Cargamos la base de Imagenet
    base_model = applications.ResNet50(weights="imagenet", include_top=False,input_tensor=input_tensor)

    x = GlobalAveragePooling2D()(base_model.output)
    x = Dropout(0.5)(x)
    x = Dense(2048, activation="relu")(x)
    x = Dropout(0.5)(x)
    final_output = Dense(n_out, activation="softmax", name='final_output')(x)
    model2 = Model(input_tensor, final_output)
    return model2

```

Fig.16. Código de ResNet50 en Keras y Tensorflow 2.0

Para que el modelo no tenga muchas complicaciones en su entrenamiento, se dividió el entrenamiento del modelo en dos fases: la primera fase es de calentamiento y la segunda es de ajuste fino, con la finalidad de tener un mejor rendimiento y menor tiempo de entrenamiento.

```

model = create_model(input_shape=(HEIGHT, WIDTH, CHANNEL), n_out=ALL_CLASSES)
for layer in model.layers:
    layer.trainable = False

for i in range(5, 0):
    model.layers[i].trainable = True

model.summary()

STEP_SIZE_TRAIN = train_generator.n//train_generator.batch_size
STEP_SIZE_VALID = validation_generator.n//validation_generator.batch_size
print(STEP_SIZE_TRAIN,STEP_SIZE_VALID)
model.compile(optimizer = optimizers.Adam(learning_rate=WARNUP_LEARNING_RATE), loss = 'categorical_crossentropy', metrics = ['accuracy'])

history_calentamiento = model.fit(train_generator, steps_per_epoch = STEP_SIZE_TRAIN, epochs = WARNUP_EPOCHS, validation_data= validation_generator,
                                validation_steps= STEP_SIZE_VALID, verbose=1)

model.save_weights("./Drive/MyDrive/Colab Notebooks/history_calentamiento.hdf5")

```

Fig.17. Algoritmo para la fase de calentamiento

En la Fig.17 se puede observar que para la primera fase se debe desactivar el parámetro “entrenable” de algunas capas, con la finalidad de congelar todas las capas de codificador en la red del modelo. Esto se lo realiza para poder calentar los pesos, es decir realizar la transferencia del aprendizaje en pequeños conjuntos de datos utilizando las inicializaciones de peso de Imagenet.

En la Fig. 18 se puede apreciar que el algoritmo terminó en el epoch 2, por lo que se pueden observar los siguientes resultados:

- Pérdida en el entrenamiento (loss): 132%
- Precisión en el entrenamiento (accuracy): 48%
- Pérdida en la validación (loss): 126%
- Precisión en la validación (accuracy): 54%

```
Epoch 1/2 [=====] - 5965 2s/step - loss: 1.9738 - accuracy: 0.4224 - val_loss: 1.2969 - val_accuracy: 0.4743
Epoch 2/2 [=====] - 339 112ms/step - loss: 1.3229 - accuracy: 0.4898 - val_loss: 1.2684 - val_accuracy: 0.5445
```

Fig.18. Finalización de la primera fase de entrenamiento de ResNet50

En la primera fase de entrenamiento el algoritmo puede llegar a mejorar su aprendizaje en la siguiente fase final. En la Fig. 19 y Fig.20 se pueden apreciar los gráficos de pérdidas y precisión del modelo en su primera fase.

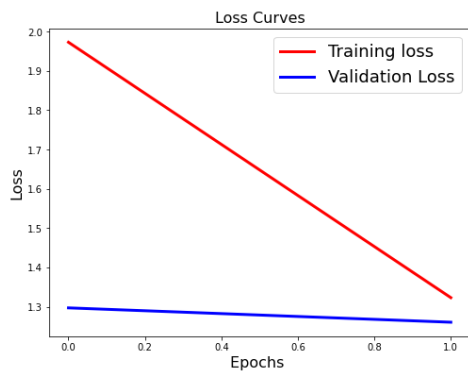


Fig. 19. Pérdida en la primera fase

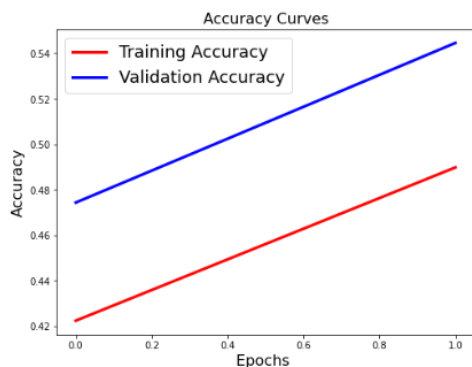


Fig. 20. Precisión en la segunda fase

La Fig. 21 se puede apreciar, que en la segunda etapa de ejecución se han descongelado y entrenado todas las capas del modelo.

```
for layer in model.layers:
    layer.trainable = True

es = EarlyStopping(monitor='val_loss', mode='min', patience=ES_PATIENCE, restore_best_weights=True, verbose=1)
r1rop = ReduceLROnPlateau(monitor='val_loss', mode='min', patience=R1ROP_PATIENCE, factor=DECAY_DROP, min_lr=1e-6, verbose=1)

callback_list = [es, r1rop]
optimizer = optimizers.Adam(learning_rate=LEARNING_RATE)
model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])
model.summary()

history_justado = model.fit(train_generator, steps_per_epoch = STEP_SIZE_TRAIN, epochs = EPOCHS,
                           validation_data = validation_generator, validation_steps = STEP_SIZE_VALID, callbacks=callback_list, verbose=1)

model.save_weights("./drive/MyDrive/Colab Notebooks/history_justado.h5")
```

Fig. 21. Código para la segunda fase de entrenamiento

Al finalizar la segunda fase en Fig.22, se puede apreciar que el modelo se detuvo en la etapa 11 y dio los siguientes resultados:

- Pérdida en el entrenamiento (loss): 6.5%

- Precisión en el entrenamiento (accuracy): 94%
- Pérdida en la validación (loss): 20%
- Precisión en la validación (accuracy): 83%

```
Epoch 9: ReduceLROnPlateau reducing learning rate to 4.899999873489376e-05.
Epoch 10/10 [=====] - 389 133ms/step - loss: 0.1138 - accuracy: 0.8869 - val_loss: 0.2136 - val_accuracy: 0.8881 - lr: 1.0000e-04
Epoch 11/10 [=====] - 389 133ms/step - loss: 0.0752 - accuracy: 0.9279 - val_loss: 0.2534 - val_accuracy: 0.7748 - lr: 5.0000e-05
Epoch 11/10 [=====] - ETA: 0s - loss: 0.0658 - accuracy: 0.9408Restoring model weights from the end of the best epoch: 6.
Epoch 11/10 [=====] - 389 133ms/step - loss: 0.0658 - accuracy: 0.9408 - val_loss: 0.2628 - val_accuracy: 0.8223 - lr: 5.0000e-05
Epoch 11: early stopping
```

Fig. 22. Finalización de la segunda fase

En Fig.23 y Fig.24 se pueden apreciar las escalas de pérdidas y precisión del modelo en esta segunda fase.

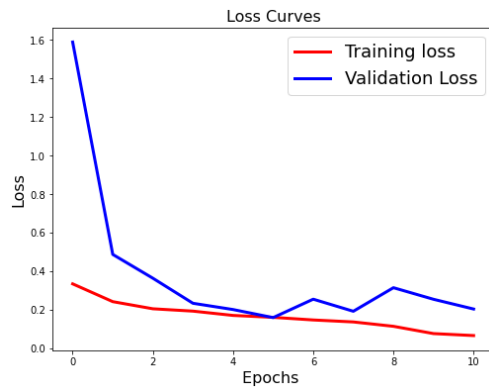


Fig. 23. Pérdida en la segunda fase

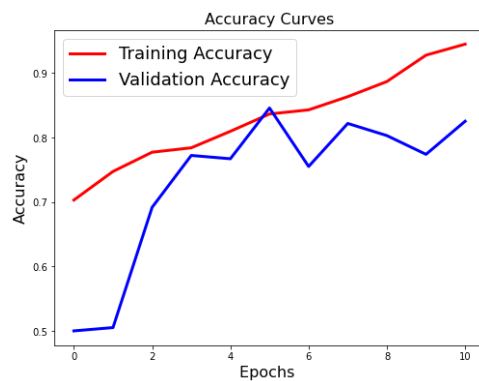


Fig. 24. Precisión en la segunda fase

En la Fig. 25 se aprecia que la mayoría de las muestras se clasificaron en la clase 0 (No DR) con un valor de 351, la clase 1 (Mild) con un valor de 8, la clase 2 (Moderate) con un valor de 162, mientras que la clase 3 (Severe) con un valor de 25 y 23 (Proliferative).

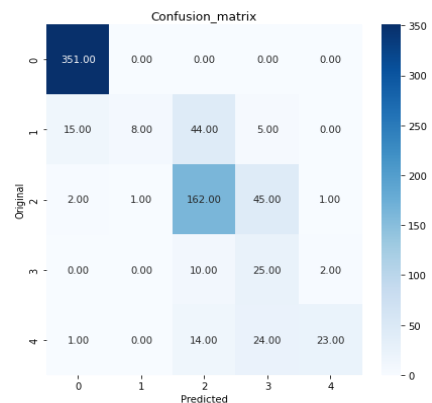


Fig. 25. Matriz de confusión ResNet50

Implementación de K-means y Random Forest.

Como el problema de investigación es de clasificación, también se evaluó la implementación de los siguientes modelos y se da a conocer los resultados de las simulaciones realizadas:

- K-means
- Random Forest

Para poder entrenar estos modelos de predicción se realizó la extracción de las características de las imágenes utilizando el algoritmo PCA, se redimensionaron las imágenes a (320x320), y se realizó la extracción con 150 componentes obteniéndose una buena varianza según la Fig.26.

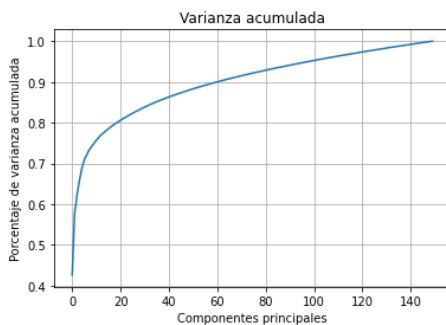


Fig.26. Varianza acumulada PCA

En el momento de entrenar los modelos se realizó la implementación de la técnica K-Folds estratificado, a pesar de haber realizado el submuestreo los datos todavía estaban desbalanceados.

La métrica de validación fue definida como una función que toma como entrada dos vectores, Y-predicción y Y-true, y retorna el valor que corresponde al porcentaje de las muestras que fueron mal clasificadas. El vector Y-true son las características verdaderas que corresponden a las estradas, mientras que Y-predicción son las predicciones realizadas por el modelo usando las mismas entradas.

K-means

Para poder aplicar K-means (vecinos cercanos) se realizó la evaluación del algoritmo con distintos vecinos, desde 1 hasta 39 vecinos, para así poder evitar problemas en la distribución de los vecinos, es decir al momento de clasificar una nueva muestra, esta posea una igual cantidad de vecinos en cada una de las clases. La mejor cantidad de vecinos que se encontró con este algoritmo fue de 3 vecinos, los resultados fueron los siguientes:

- Eficiencia de validación: 57%
- Intervalo de confianza en eficiencia de validación: 1%
- Eficiencia de entrenamiento: 43%
- Intervalo de confianza en la eficiencia de entrenamiento: 1%

Este resultado nos da a conocer un sub-ajuste de los datos. Mientras más crecían los vecinos se reducía el porcentaje en la validación y se incrementaba el porcentaje en el entrenamiento.

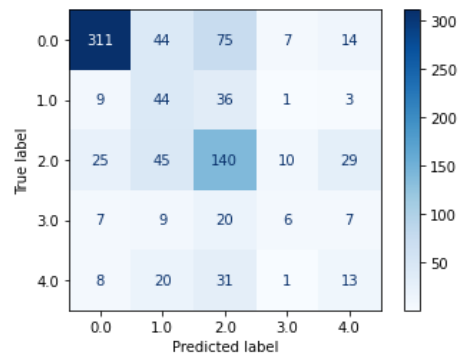


Fig.27. Matriz de confusión K-means

En la Fig.27 se puede apreciar que la mayoría de las muestras se clasificaron en la clase 0 (No_DR) y la clase 2, mientras que la clase 3 y 4 se clasificaron pocas imágenes, 0 y 2 representan a la parte mayor del conjunto de datos, es decir existe un desbalance en los datos.

Random Forest

Para poder realizar la evaluación de este modelo se utilizó K-Fold estratificado con 4 Folds. Esto quiere decir que 14108/4 = 3527 registros que se ubicaron en cada Fold. Finalmente, los resultados obtenidos por la implementación de Random Forest a nuestro conjunto de datos se refleja en la Fig.28 :

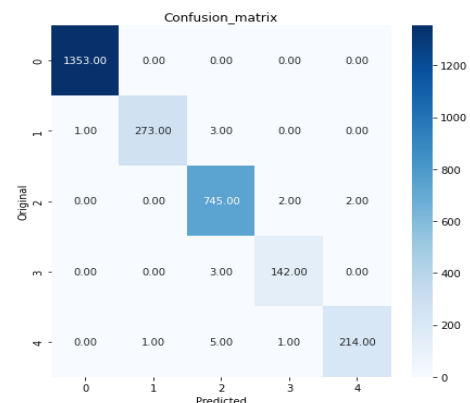


Fig.28. Matriz de confusión Random Forest

IV. CONCLUSIONES

Se puede apreciar que Random Forest y K-Means se sobreajustaron a los datos, esto quiere decir que el método PCA utilizado para extraer las características no fue muy eficiente, debido a que este pudo haber descartado ciertos parámetros o variables importantes para la detección y clasificación, pero en los modelos de aprendizaje profundo se puede apreciar una mayor eficiencia de validación. El modelo ResNet18 al ser una red que toma características superficiales no dio buenos resultados en el entrenamiento y validación a diferencia de ResNet50 el cual dio resultados en el entrenamiento y en la validación, dando como resultado un grado de precisión de 85.95%. el cual es un resultado aceptable para lograr la clasificación y detección de retinopatía diabética.

REFERENCIAS

Por lo que podemos asegurar que los algoritmos de aprendizaje automático son las mejores opciones, en especial ResNet-50 de entre todas las presentadas en este documento, dado que le permite al desarrollador investigar y experimentar el comportamiento de los modelos basados en Machine Learning para la resolución de problemas de clasificación y la detección de enfermedades como la RD a través del análisis de retino grafías.

Mientras más imágenes posea de cada tipo de RD el modelo en el proceso de aprendizaje este puede llegar a realizar una mejor clasificación. También que el tamaño del conjunto de datos puede incidir en el rendimiento del modelo, esto significa que mientras mayores sean los datos, mayor tiempo le tomaría al modelo entrenarse. La importancia y efectividad del uso de las redes neuronales convolucionales para el análisis de datos.

Por lo tanto, el modelo de Machine Learning para la predicción y clasificación del nivel de Retinopatía Diabética cumple con los requerimientos necesarios para ser utilizado como una herramienta tecnológica de clasificación confiable y precisa de acuerdo con los resultados obtenidos.

Se debe de tener en cuenta que para este problema de detección y clasificación de imágenes los modelos de aprendizaje profundo son muy eficientes para esto, debido a que utilizan estructuras complejas y optimizadas (redes neuronales) para el análisis, estos analizan los datos de entrada en múltiples capas de la red, y cada capa define entidades y patrones específicos en los datos, esto les permite detectar características como hemorragias, aneurismas, dilatación de vasos sanguíneos, las cuales son esenciales para la detección y clasificación de la retinopatía diabética.

El conjunto de imágenes de retinografías, deben de tener una buena calidad para que cuando estas sean usadas en la etapa de entrenamiento no se presenten inconsistencias en el modelo.

Si no se posee un computador con altas capacidades de RAM y GPU para poder efectuar cálculos complejos y realizar el análisis de datos de redes neuronales convolucionales, es más factible utilizar Google Colab, el cual otorga maquinas desde 12GB de RAM y 50GB de almacenamiento

La matriz de confusión se utiliza para evaluar la eficacia del modelo en la clasificación de datos.

Se comprobó que la ejecución de los dos modelos en GPU ofreció menor tiempo de ejecución que en CPU y dado que las precisiones alcanzadas en ambos escenarios son adecuadas y aceptables por encima del 80% de exactitud, se pueden estimar como factible la aplicación de ambos modelos ya que el estándar base promedio de recomendación de efectividad está entre el 75% y 80%.

- [1] Organización Mundial de la Salud, «TADDS* Instrumento para la evaluación de los sistemas de atención a la diabetes y a la retinopatía diabética.» 2015. [En línea]. Available: https://www.who.int/blindness/publications/TADDS_ES.pdf. [Último acceso: 13 Marzo 2022].
- [2] M. E. Macías Llanes, F. Figaredo Curiel, J. Núñez Jover y M. E. del Huerto Marimón, «Estrategias en el tratamiento de la retinopatía diabética,» *Revista cubana de oftalmología*, vol. 31, pp. 90-99, 2018.
- [3] F. Tablado, «Inteligencia artificial: Definición, tipos y aplicaciones,» 18 Junio 2020. [En línea]. Available: <https://protecciondatos-lopd.com/empresas/inteligencia-artificial/>.
- [4] Ewout W. Steyerberg, Andrew J. Vickers, Nancy R. Cook, Thomas Gerds, Mithat Gonen, Nancy Obuchowski, Michael J. Pencina y Michael W. Kattan, «Assessing the performance of prediction models: a framework for some traditional and novel measures,» *PubMed Centra*, vol. 21, n° 1, pp. 128-138, 2010.
- [5] L. Petteri Rouhiainen, «Inteligencia Artificial 101 cosas que debes saber hoy sobre nuestro futuro,» Noviembre 2018. [En línea]. Available: https://www.planetadelibros.com/libros_contenido_extra/40/39307_Inteligencia_artificial.pdf.
- [6] Yann LeCun, Yoshua Bengio y Geoffrey Hinton, «Deep learning,» *Nature*, p. 436–444, 2015.
- [7] Xianfeng Ou, Pengcheng Yan, Yiming Zhang, Bing Tu, Guoyun Zhang, Jianhui Wu y Wujing Li, «Moving Object Detection Method via ResNet-18 With Encoder–Decoder Structure in Complex Scenes,» *IEEE Access*, vol. 7, pp. 2-3, 2019.
- [8] Giuseppe G. A. Celano, «A ResNet-50-based Convolutional Neural Network Model for Language ID Identification from Speech Recordings,» 10 Junio 2021. [En línea]. Available: <https://aclanthology.org/2021.sigtyp-1.13.pdf>.
- [9] Mohapatra, Niva & Shreya, K. & Chinmay y Ayes, «ResearchGate GmbH,» Enero 2020. [En línea]. Available: https://www.researchgate.net/publication/338561346_Optimization_of_the_Random_Forest_Algorithm.
- [10] J. S. Ramírez Gómez, N. D. Duque Méndez y J. J. Vélez Upegu, «Normalización en desempeño de k-means sobre datos climáticos,» *Revista Vínculos: Ciencia, Tecnología y Sociedad*, vol. 16, n° 1, pp. 57-72, Enero-Junio 2019.
- [11] G. Moreno, «Las redes neuronales: qué son y por qué están volviendo,» 21 Junio 2016. [En línea]. Available: <https://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo>.
- [12] K. L. & R. Delgado, «Análisis de electroencefalograma usando redes neuronales artificiales,» *Acta Universitaria*, p. 29, 2019.
- [13] E. Todt y B. A. Krinski, «Convolutional Neural Network - CNN,» 30 Noviembre 2019. [En línea]. Available: https://www.inf.ufpr.br/todt/IAaplicada/CNN_Presentation.pdf.
- [14] J. Torres, «DEEP LEARNING Introducción práctica con Keras,» Junio 2018. [En línea]. Available: https://books.google.es/books?hl=es&lr=&id=ju1mDwAAQBAJ&oi=fnd&pg=PA78&dq=keras&ots=k_RhPC61RT&sig=x5RVXxscopMAIQBgyz0sNC6IbQ8#v=onepage&q=keras&f=false.
- [15] inLab FIB, «Plataforma para hacer competiciones de inteligencia artificial,» inLab FIB, 24 Agosto 2018. [En línea]. Available: <https://inlab.fib.upc.edu/es/blog/plataforma-para-hacer-competitions-de-inteligencia-artificial>. [Último acceso: Marzo 2022].
- [16] Changsheng Zhu, Christian Uwa Idemudia y Wenfang Feng, «Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques,» *Informatics in Medicine Unlocked*, vol. 17, pp. 100-179, 2019.
- [17] C. Bonilla Carrión, «REDES CONVOLUCIONALES,» Junio 2020. [En línea]. Available: <https://idus.us.es/bitstream/handle/11441/115221/TFG%20DGM%20Bonilla%20Carri%C3%B3n%2C%20Carmelo.pdf?sequence=1&isAllowed=y>.
- [18] «Análisis de Calidad Cartográfica mediante el uso de la Matriz de

- Confusión.» 1 Octubre 2016. [En línea]. Available: file:///C:/Users/erick/Documents/ERICK/Dialnet-AnalisisDeCalidadCartograficaMedianteElEstudioDeLa-5998855.pdf. [Último acceso: Marzo 2022].
- [19] D. H. Cámpora Pérez, «A practical approach to Convolutional Neural,» 5 Marzo 2019. [En línea]. Available: https://indico.cern.ch/event/766995/contributions/3295783/attachments/1802769/2940979/Daniel_Campora_-_A_practical_approach_to_Convolutional_Neural_Networks_slides.pdf.
- [20] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto y Ridella S, «The ‘K’ in K-fold Cross Validation,» *i6doc.com publ*, n° 6, pp. 441-446, 2012.
- [21] A. Pitarque, J. C. Ruiz y J. F. Roy, «Las redes neuronales como herramientas estadísticas no paramétricas de clasificación,» *Psicothema*, vol. 12, n° Su2, pp. 459-463, 2000.
- [22] L. Herguedas Fenoy, «Guía práctica clínica para la realización de una retinografía,» 2018. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/31988>.
- [23] J. Chua, X. Y. Lim, T. Y. Wong y C. Sabanayagam, «Diabetic Retinopathy in the Asia-Pacific,» *The Asia-Pacific Journal of Ophthalmology*, vol. 7, pp. 3-16, 2018.
- [24] Gregor Stiglic, Primoz Kocbek, Nino Fijacko, Marinka Zitnik, Katrien Verbert y Leona Cilar, «Interpretability of machine learning-based prediction models in healthcare,» *WIREs*, vol. 10, n° 5, 2020.
- [25] Zhi-Hua Zhou, «A brief introduction to weakly supervised learning,» *National Science Review*, vol. 5, p. 44–53, Enero 2018.
- [26] Xiang Lin, R D (Shawn) Blanton y Donald E Thomas, «Random Forest Architectures on FPGA for Multiple Applications,» *ACM Digital Library*, n° 4, pp. 415-418, 10 Mayo 2017.
- [27] S. C. Vaca Orellana, «Factores de riesgo que inciden en retinopatía diabética no proliferativa pacientes con diabetes mellitus tipo 2 mayores 40 años establecimiento de salud Carapungo 2, quito 2018,» 2019. [En línea]. Available: http://repositorio.unemi.edu.ec/bitstream/123456789/4638/1/TESIS%20VACA%20ORELLANA%20SUSAN%20CATHERINE_compressed.pdf.
- [28] Thippa Reddy Gadekallu, Neelu Khare, Sweta Bhattacharya, Saurabh Singh, Praveen Kumar Reddy Maddikunta, In-Ho Ra y Mamoun Alazab, «Early Detection of Diabetic Retinopathy Using PCA-Firefly Based Deep Learning Model,» 5 Febrero 2020. [En línea]. Available: <https://www.mdpi.com/2079-9292/9/2/274/htm>.
- [29] Jing Wei, Xuan Chu, Xiang-Yu Sun, Kun Xu, Hui-Xiong Deng, Jigen Chen, Zhongming Wei y Ming Lei, «Machine learning in materials science,» *InfoMat*, pp. 338-358, 2019.
- [30] Jianxin Wu, «National Key Lab for Novel Software Technology Nanjing University, China,» 1 Mayo 2017. [En línea]. Available: <https://cs.nju.edu.cn/wujx/paper/CNN.pdf>.
- [31] R. Gonzalez Gouveia, «Diferencias entre Inteligencia Artificial vs Machine Learning vs Deep Learning,» 14 Marzo 2021. [En línea]. Available: <https://www.youtube.com/watch?v=7WTEWyn9zfU>.
- [32] Geoffrey Hinton, Oriol Vinyals y Jeff Dean, «Distilling the Knowledge in a Neural Network,» *arXiv*, vol. 1, n° 9, 2015.
- [33] L. García Ferrer, M. Ramos López, Y. Molina Santana, M. Chang Hernández, E. Perera Miniet y K. Galindo Reydmund, «Estrategias en el tratamiento de la retinopatía diabética,» *Revista Cubana de Oftalmología*, pp. 90-99, 2018.
- [34] Banerjee Priyanka y Preissner Robert, «BitterSweetForest: A Random Forest Based Binary Classifier to Predict Bitterness and Sweetness of Chemical Compounds,» *Frontiers in Chemistry*, vol. 6, 11 Abril 2018.
- [35] Anirudha Ghosh, A. Sufian, Farhana Sultana, Amlan Chakrabarti y Debashis De, «ResearchGate GmbH,» Enero 2020. [En línea]. Available: https://www.researchgate.net/publication/337401161_Fundamental_Concepts_of_Convolutional_Neural_Network.
- [36] Aldo Glielmo, Brooke E. Husic, Alex Rodriguez, Cecilia Clementi, Frank Noé y Alessandro Laio, «Unsupervised Learning Methods for Molecular Simulation Data,» *ACS Publications*, p. 9722–9758, 2021.
- [37] Analytics Vidhya, «Analytics Vidhya,» 17 Febrero 2020. [En línea]. Available: <https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/>.