|  |  |
| --- | --- |
| Descripción: tempimage1 | D:\Diseño\power point\Sin título-1.png  **FACULTAD DE CIENCIAS NATURALES Y EXACTAS**  **DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**  **PRÁCTICA LABORAL E INVESTIGATIVA II**  **SEGUNDO AÑO LICENCIATURA EN CIENCIA DE LA COMPUTACIÓN**  **Título del trabajo**  Análisis morfológico automático de células eritrocitos mediante una red neuronal convolucional.  **Autor**: Alejandro Fonseca Cuza  **Tutor**: Dr.C. Wilkie Delgado Font  **Diciembre de 2023** |

**Resumen**

La clasificación morfológica de las células es una herramienta fundamental en el campo de la medicina, ya que permite identificar y diferenciar diferentes tipos de células presentes en los tejidos y órganos del cuerpo humano. Esta clasificación se basa en la observación de la forma, tamaño, estructura y función de las células, lo que permite establecer diagnósticos precisos de enfermedades y trastornos. En este estudio, se automatizó el análisis morfológico de células de eritrocitos. Se clasificaron en tres categorías: circular, deformada alargada y deformada poco alargada, utilizando una red neuronal convolucional profunda para la extracción de características y clasificación de imágenes.

Las pruebas se realizaron sobre un conjunto de imágenes individuales creadas para este trabajo en estudios anteriores. Para mejorar el entrenamiento se llevaron a cabo experimentos con imágenes reales y generadas. Se utilizó un modelo de red neuronal convolucional con 3 capas convolucionales y 2 totalmente conectadas.

Los resultados obtenidos a través de la utilización de una red neuronal para la clasificación de células han demostrado ser altamente efectivos y precisos en términos de su capacidad para identificar y diferenciar diferentes tipos de células alcanzando una precisión prácticamente exacta. Estos hallazgos sugieren que la clasificación morfológica de las células a través del uso de técnicas de inteligencia artificial puede ser una herramienta valiosa en el campo de la medicina, especialmente en el diagnóstico y tratamiento de enfermedades relacionadas con las células de eritrocitos.

**Abstract**

The morphological classification of cells is a fundamental tool in the field of medicine, since it allows us to identify and differentiate different types of cells present in the tissues and organs of the human body. This classification is based on the observation of the shape, size, structure and function of cells, which allows accurate diagnoses of diseases and disorders to be established. In this study, the morphological analysis of erythrocyte cells was automated. They were classified into three categories: circular, elongated deformed, and slightly elongated deformed, using a deep convolutional neural network for feature extraction and image classification.

The tests were performed on a set of individual images created for this work in previous studies. To improve the training, experiments were carried out with real and generated images. A convolutional neural network model was used with 3 convolutional and 2 fully connected layers.

The results obtained through the use of a neural network for cell classification have proven to be highly effective and precise in terms of their ability to identify and differentiate different types of cells, achieving practically exact precision. These findings suggest that morphological classification of cells through the use of artificial intelligence techniques can be a valuable tool in the field of medicine, especially in the diagnosis and treatment of diseases related to erythrocyte cells.

Contenido

[1. Introducción 3](#_Toc118370654)

[1.1 Enunciado del problema a resolver. 3](#_Toc118370655)

[1.2 Análisis general dela solución del problema. 3](#_Toc118370656)

[1.3 Objetivos del trabajo (general y específicos). 3](#_Toc118370657)

[2. Desarrollo 4](#_Toc118370658)

[2.1 Modelos matemáticos de la solución. 4](#_Toc118370659)

[2.2 Diagrama UML de la(s) clase(s). 4](#_Toc118370660)

[2.3 Descripción de los algoritmos elaborados. 4](#_Toc118370661)

[2.4 Implementación 4](#_Toc118370662)

[3. Conclusiones 5](#_Toc118370663)

[Referencias Bibliográficas 6](#_Toc118370664)

# Introducción

## 1.1 Enunciado del problema a resolver.

Los eritrocitos, también conocidos como glóbulos rojos, son células sanguíneas que se encargan de transportar oxígeno desde los pulmones a los tejidos del cuerpo, y de llevar dióxido de carbono de vuelta a los pulmones para ser exhalado. Estas células tienen una forma de disco bicóncavo que les permite maximizar su área de superficie, lo que facilita su capacidad de intercambio gaseoso. Su formación tiene lugar en la médula ósea, a través de un proceso llamado eritropoyesis, en el cual las células madre hematopoyéticas se diferencian en eritroblastos y luego en eritrocitos maduros [1].

En cuanto a la tecnología, la inteligencia artificial (IA) y los modelos de redes neuronales de clasificación juegan un papel crucial en la detección y prevención de enfermedades relacionadas con los eritrocitos. La IA puede analizar grandes conjuntos de datos de pacientes, incluyendo recuentos de eritrocitos, su tamaño y forma, así como otras métricas de la sangre. Con esta información, estos algoritmos pueden identificar patrones que podrían indicar la presencia de enfermedades como anemia, talasemia, trastornos de coagulación, entre otros.

De igual forma los eritrocitos, se analizan empleando microscopios ópticos. Este análisis presenta dificultades relacionadas con el cansancio que provoca el tiempo que se emplea y la posición que adquieren los especialistas para analizar las muestras, la dificultad del propio medio de observación e incluso la variabilidad de criterios entre especialistas, lo que incide en la calidad de los resultados que se obtienen y provoca que el esfuerzo para lograrlos, debido a la sensibilidad el entorno valorado y a la necesidad de garantizar la veracidad de los mismos, sea realmente elevado [2].

Considerando el escenario descrito anteriormente, el problema de investigación de la presente investigación es la necesidad de comprobar el desempeño en la clasificación automática de células de eritrocito mediante diferentes métodos de aprendizaje profundo utilizado una red neuronal convolucional (CNN).

## 1.2 Análisis general de la solución del problema.

Ya se cuenta con resultados de clasificación obtenidos empleando características asociadas a funciones basadas en geometría integral y utilizando como algoritmo de clasificación KNN,SVM,Decision Trees que es de los más sencillos existentes. Los resultados obtenidos con estos métodos muestran valores superiores al 96% [3].

Aunque los resultados se consideran muy buenos, es de interés continuar elevando los resultados de la sensibilidad en la clasificación de los mismos, para lo cual se propone el empleo de otros algoritmos de clasificación considerados más potentes, en este caso, las redes neuronales convolucionales.

## 1.3 Objetivos del trabajo (general y específicos).

Este trabajo tiene como objetivo general, evaluar el comportamiento y resultado del modelo de una red neuronal profunda, en la clasificación de manera automática de células de eritrocitos según su morfología.

**Para el cumplimiento de este objetivo general, se requiere de los siguientes objetivos específicos:**

1. Revisión del estado del arte sobre los fundamentos del aprendizaje profundo, en concreto de las redes neuronales para la clasificación de imágenes.

2. A partir del conjunto de datos obtenido de anteriores estudios. Realizarle transformaciones de rotación, escalamiento para así lograr alcanzar una mayor cantidad de datos.

3. Crear, entrenar y probar un modelo de red neuronal convolucional que sea lo suficientemente bueno para obtener una buena precisión.

4. Comparar los resultados con otros obtenidos anteriormente.

# Desarrollo

**¿Qué es el Deep Learning?**

El Deep Learning es una subárea de la inteligencia artificial que se ocupa del aprendizaje automático de representaciones de datos mediante el uso de algoritmos que modelan estructuras y abstracciones jerárquicas. La idea es aprender a representar los datos con múltiples niveles de abstracción, lo que hace posible que el sistema pueda realizar tareas complejas sin depender de una programación específica para cada tarea.

**¿Qué son las Redes Neuronales Profundas?**

Las redes neuronales profundas son un componente fundamental del Deep Learning. Son un tipo de modelo de aprendizaje automático inspirado en la red neuronal biológica del cerebro humano. Estas redes consisten en capas de nodos (neuronas) que toman entradas, aplican transformaciones y generan salidas. La profundidad de estas redes se refiere a la presencia de múltiples capas ocultas entre la capa de entrada y la capa de salida.

**¿Cómo funcionan?**

Las redes son capaces de aprender patrones complejos a partir de los datos a través de la realización de operaciones matemáticas en las entradas que reciben. Cada capa de la red realiza transformaciones en los datos, permitiendo la extracción progresiva de características más abstractas a medida que avanzamos en las capas. Este proceso de extracción de características jerárquicas es la clave para el éxito de las redes neuronales profundas en tareas como reconocimiento de imágenes, procesamiento del lenguaje natural, entre otros.

**¿Cómo aprenden?**

Las redes neuronales profundas aprenden a través de un proceso llamado entrenamiento, que generalmente implica el uso de conjuntos grandes de datos para ajustar los pesos y sesgos de la red. Esto se hace mediante la minimización de una función de pérdida que mide la discrepancia entre las predicciones de la red y las salidas deseadas usando algoritmos como el descenso de gradiente y retro propagación hacia atrás.

## Modelos matemáticos de la solución.

Cada neurona en una capa oculta realiza un cálculo que implica una combinación lineal de las entradas seguida de una función de activación no lineal. Este proceso se repite para todas las neuronas en la capa. Las funciones de activación, como ReLU, tanh o sigmoid, introducen no linealidad en la red, lo que le permite aprender y modelar relaciones complejas en los datos.

Donde:

-

-

-

-

-

-

**Función de activación ReLU** **(Rectified Linear Unit)**

La función de activación ReLU es una de las funciones de activación más utilizadas en redes neuronales. ReLU opera elemento por elemento,toma un valor de entrada x y devuelve x si x es positivo, y devuelve 0 si x es negativo. Gráficamente, esto significa que la función tiene una pendiente de 1 para valores positivos de x y es horizontal para valores negativos.

Función de activación ReLU:

**Función de activación Softmax**

La función de activación Softmax se utiliza comúnmente en la capa de salida de una red neuronal para problemas de clasificación multiclase. Esta toma como entrada un vector de números reales y produce como salida un vector de la misma dimensión, donde los valores están en el rango (0, 1) y la suma de todos los valores es 1. Esto es especialmente útil en el contexto de clasificación, ya que proporciona una distribución de probabilidad sobre las clases.

Función de activación Software:

Donde:

## Diagrama UML de la(s) clase(s).

@startuml

class Net {

- num\_channels: int

- conv1: Conv2d

- conv2: Conv2d

- conv3: Conv2d

- fc1: Linear

- fc2: Linear

+ init(num\_channels)

+ forward(x)

}

class Conv2d

class Linear

Net "1" -- "1" Conv2d : has

Net "1" -- "1" Linear : has

@enduml

@startuml

class GetImage {

- base\_dir: str

- split: str

- transform: Any

- filenames: list

- targets: list

+ \_\_init\_\_(base\_dir, split="train", transform=None)

+ \_\_len\_\_()

+ \_\_getitem\_\_(idx)

}

class Dataset

class Image

GetImage --|> Dataset

GetImage "1" o-- "N" Image : has

@enduml

## Descripción de los algoritmos elaborados.

Para proceder con el proceso de entrenamiento cada clase fue etiquetada desde 0 hasta 2 (0 - circular, 1 - deformadas alargadas, 2 - deformadas poco alargadas u otras deformaciones), las cuales se cargaron de la siguiente forma:

Función: GetData(path='./',split='train',transform=None)

Entrada: path: ruta de las imagenes, split: nombre de la carpeta con las imagenes del grupo que quiere guardar, transform: La transformacion que se le hace a las imagenes.

Precondiciones: La carpeta que este en la ruta especificada y split especificado debe tener dentro tres carpetas con las imgenes de cada clase dentro y las etiquetas como nombre de las carpetas

Salida: Una lista de pares (imagen,etiqueta) con todas las imagenes correspondientes al split transformadas con el transform

Pasos:

data=[ ]

full\_path=os.path.join(path,split);

para i en os.listdir(full\_path) hacer:

file=os.path.join(full\_path,i)

si: os.path.isdir(file):

para img en os.listdir(file) hacer:

si: img.endswith('.png'):

path\_img=os.path.join(file,img);

image=Image.open(path\_img)

image = image.convert("RGB")

si: transform:

image=transform(image)

si: image.shape[0]>1:

image=image[0,:,:].unsqueeze(0) //Reducir las imagenes a un solo canal

fin

fin

data.append((image,int(i)))

fin

np.random.shuffle(data)

fin

Durante el aprendizaje profundo una de las principales limitaciones es la falta de datos de entrenamiento. Para evitar el problema del sobreajuste se utilizo el aumento de datos.

Para el aumento de los datos se aplicó la rotación con ángulos diferentes de manera aleatoria. También se aplicó dos técnicas de volteo (horizontal, vertical) y por ultimo se convirtió a tensor:

transform\_R = transforms.Compose([

transforms.RandomRotation(2 \* torch.pi),

transforms.RandomVerticalFlip(),

transforms.RandomHorizontalFlip(),

transforms.PILToTensor(),

])

Función: AumentoDatos(par):

Entrada: Un par (imagen,etiqueta)

Precondiciones: 0 <= etiqueta<3

Salida: Una lista de pares del mismo formato que el de entrada pero con los datos aumentados en cierta cantidad segun su tipo.

Pasos:

imgAum=[ ]

parAum=[ ]

imgaux=par[0]

rango =0

si: par[1] ==0:

rango = 16

fin

si: par[1] ==1:

rango = 24

fin

si: par[1] ==2:

rango = 4

fin

para i en el rango (rango) hacer:

imgaux = transform\_R(par[0])

si: imgaux.shape[0]>1:

imgaux=imgaux[0,:,:].unsqueeze(0)

fin

found\_match = False

si: image in imgAum:

si: torch.allclose(image, imgaux):

found\_match = True

break

fin

fin

si no: found\_match:

imgAum.append(imgaux)

parAum.append((imgaux,par[1]))

fin

fin

Algoritmo para el entrenamiento de la red:

Funcion: train(𝑑𝑠𝑡𝑟𝑎𝑖𝑛,𝑑𝑠𝑡𝑒𝑠𝑡,𝑒𝑝,𝑙𝑟, model, loss\_fn, optimizer,device)

Entrada: model: modelo a entrenar, 𝑒𝑝: Cantidad de épocas a entrenar, 𝑙𝑟: Tasa de aprendizaje, 𝑑𝑠𝑡𝑟𝑎𝑖𝑛: Datos de entrenamiento, 𝑑𝑠𝑡𝑒𝑠𝑡: Datos de prueba, optimizer: Optimizador, loss\_fn = Funcion de perdida

Resultado: Modelo entrenado

𝑀 = model

𝑂 = optimizer

𝑡𝑜𝑡𝑎𝑙𝑙𝑜𝑠𝑠 = 0

𝑡𝑜𝑡𝑎𝑙𝑐𝑜𝑟𝑟𝑒𝑐𝑡 = 0

para 𝑖 en rango(𝑒𝑝) hacer:

para cada 𝑏𝑎𝑡𝑐ℎ en 𝑑𝑠𝑡𝑟𝑎𝑖𝑛 hacer:

(𝑋,𝑦) = 𝑏𝑎𝑡𝑐ℎ

𝑝𝑟𝑒𝑑 = (𝑋) (Pasa los datos del 𝑏𝑎𝑡𝑐ℎ actual al modelo)

𝑙𝑜𝑠𝑠 = loss\_fn(𝑝𝑟𝑒𝑑,𝑦)

𝑂.𝑏𝑎𝑐𝑘𝑝𝑟𝑜𝑝𝑎𝑔𝑎𝑡𝑖𝑜𝑛()

𝑡𝑜𝑡𝑎𝑙𝑙𝑜𝑠𝑠 ← 𝑡𝑜𝑡𝑎𝑙𝑙𝑜𝑠𝑠 +𝑙𝑜𝑠𝑠

𝑡𝑜𝑡𝑎𝑙𝑐𝑜𝑟𝑟𝑒𝑐𝑡 ← 𝑡𝑜𝑡𝑎𝑙𝑐𝑜𝑟𝑟𝑒𝑐𝑡 +𝑐𝑜𝑢𝑛𝑡(𝑝𝑟𝑒𝑑 == 𝑦)

fin

fin

𝑛 = Cantidad de 𝑏𝑎𝑡𝑐ℎ𝑠 en 𝑑𝑠𝑡𝑟𝑎𝑖𝑛

𝑚 = Cantidad total de imágenes en 𝑑𝑠𝑡𝑟𝑎𝑖𝑛

𝑙𝑜𝑠𝑠𝑡𝑟𝑎𝑖𝑛 = 𝑡𝑜𝑡𝑎𝑙𝑙𝑜𝑠𝑠/𝑛

𝑎𝑐𝑐𝑢𝑟𝑎𝑐𝑦𝑡𝑟𝑎𝑖𝑛 = 𝑡𝑜𝑡𝑎𝑙𝑐𝑜𝑟𝑟𝑒𝑐𝑡/ 𝑚

(𝑎𝑐𝑐𝑢𝑟𝑎𝑐𝑦𝑡𝑒𝑠𝑡,𝑙𝑜𝑠𝑠𝑡𝑒𝑠𝑡) = Computar la pérdida usando el mismo algoritmo pero sin usar back-propagation.

* 1. **Implementación**

**Lenguajes de Programación Utilizados:**

-Python

-Javascript

**Módulos/Frameworks/Librerías:**

-Pytorch

-Matplotlib

-Numpy

-Flask

-OpenCV

-Pillow

-Os

**IDE (Entorno de desarrollo integrado):**

-Anaconda

-Google Colab

-Visual Studio Code

# Conclusiones

Los resultados alcanzados en este trabajo dan cuenta del cumplimiento del objetivo general y de los objetivos específicos propuestos inicialmente. Los mismos son descritos a continuación:

* La revisión del estado del arte sobre los fundamentos del aprendizaje profundo, en concreto de las redes neuronales para la clasificación de imágenes, permitió corroborar que es posible aplicar el aprendizaje profundo para realizar el análisis morfológico de manera automática de células de eritrocitos.
* Los resultados obtenidos alcanzados fueron excelentes alcanzando un valor superior al 99% en los datos de prueba.
* Al evaluar el comportamiento y resultado del modelo, se evidencia la capacidad del deep learning para la clasificación de imágenes por encima de los métodos tradicionales del machine learning en la mayoría de los casos.

# Referencias Bibliográficas

[1] Enderby, J E; Neilson, G W (1 de junio de 1981). [«The structure of electrolyte solutions»](https://web.archive.org/web/20211218145748/https:/iopscience.iop.org/article/10.1088/0034-4885/44/6/001). *Reports on Progress in Physics* **44** (6): 593-653. [ISSN](https://es.wikipedia.org/wiki/ISSN) [0034-4885](https://portal.issn.org/resource/issn/0034-4885). [S2CID](https://es.wikipedia.org/wiki/Semantic_Scholar) [250852242](https://api.semanticscholar.org/CorpusID:250852242). [doi](https://es.wikipedia.org/wiki/Digital_object_identifier):[10.1088/0034-4885/44/6/001](https://dx.doi.org/10.1088%2F0034-4885%2F44%2F6%2F001). Archivado desde [el original](https://iopscience.iop.org/article/10.1088/0034-4885/44/6/001) el 18 de diciembre de 2021. Consultado el 18 de diciembre de 2021.

[2] Herold, S. (2014). Modelos y métodos de la geometría estocástica para el procesamiento y análisis de imágenes de muestras biológicas, obtenidas mediante técnicas de microscopía [Tesis doctoral, Universitat Jaume I].