

Segmentación de heridas crónicas en imágenes RGB a través de técnicas de adaptación de dominio y aprendizaje profundo

Fabian Perez, Daniel Ballesteros, Cristian Rey, Karen Sanchez, Henry Arguello

Resumen—La diabetes tipo-2, la lepra, y otras enfermedades infecciosas, pueden ocasionar daños severos al organismo, tales como, heridas en la piel. De no ser tratadas adecuadamente, estas heridas o úlceras pueden generar la amputación de extremidades del paciente. A partir del uso de redes neuronales convolucionales, particularmente, con arquitecturas como U-net o MobilNet, es posible asistir de manera computarizada la segmentación de las heridas a partir de fotografías, y calcular su área y perímetro. El análisis de múltiples fotografías de un paciente, tomadas en diferentes días, puede permitir realizar el seguimiento de la curación apropiada (o no) de una herida.

Palabras Clave—chronic ulcer, medical imaging, deep learning, Adaptación de dominio, data augmentation, convolutional neural networks.

1 INTRODUCCIÓN

LA diabetes es una enfermedad que se presenta cuando el nivel de glucosa en sangre, también conocido como azúcar en sangre, es demasiado alto. Es considerada una enfermedad metabólica crónica que con el tiempo conduce a daños graves en el corazón, vasos sanguíneos, ojos, riñones y nervios [1]. El daño en los nervios o neuropatía diabética, puede desencadenar la generación de heridas abiertas en la piel, las cuales, requieren de un tratamiento especializado que prevenga su infección. Esta enfermedad también puede disminuir la cantidad de flujo sanguíneo que llega a los pies, lo que hace más difícil que la herida sane [2] [3]. Además de la diabetes, otra causa principal de la generación de heridas crónicas en la piel es la enfermedad de Hansen, antes conocida como Lepra.

Actualmente, se presenta gran número de casos de úlceras en extremidades inferiores debido a estas dos condiciones. El manejo de estas úlceras presenta un gran desafío clínico, ya que la respuesta a los tratamientos en la mayoría de los casos es deficiente y los gastos elevados, tanto para los proveedores de atención médica como para el paciente [4]. En Estados Unidos las heridas relacionadas con diabetes representan un problema de salud que afecta a 5-6 millones de personas [5]. Además, provocaron aproximadamente 73,000 amputaciones de miembros inferiores en el año 2010. A pesar de que se pueden prevenir con las medidas adecuadas de intervención médica y autogestión del paciente, la tasa de incidentes de las úlceras del pie están aumentando y el cuidado efectivo de las mismas impone una carga laboral considerable a los médicos.

Para apoyar a los profesionales clínicos en el análisis rápido y preciso de la evolución de las heridas por úlceras, se han diseñado estrategias tecnológicas novedosas

de inteligencia artificial, que realizan tareas de medición de área y perímetro de las heridas a partir de fotografías convencionales.

Estos estudios realizan tareas de clasificación o segmentación, a partir de métodos tradicionales o de aprendizaje profundo. En uno de los reportes encontrados en el estado del arte se utilizó la técnica de agrupación por color seleccionado, la imagen es capturada por una cámara digital usando un filtro 5x5 y difusión anisotrópica para la homogeneización del color. A partir de esto se segmentaron las heridas utilizando técnicas de agrupación para los canales de color. Los autores reportan resultados con distintas métricas para cada media de los canales de color con una precisión del 74,39 % [6]. Otro de los trabajos propuso un marco de segmentación de heridas usando métodos tradicionales de procesamiento de imágenes con un algoritmo de detección en heridas con robustez y entregando mejores resultados de segmentación, en este trabajo cambiaron el método de profundidad de Mobile Net en redes con Depth Multiplier igual al 25%, 50% y 100%. Este modelo añade el pos-procesamiento de la corrección semántica al modelo de pre-procesamiento [7]. Además, se encontraron otros trabajos en los cuales hacen uso de la segmentación y clasificación con redes convolucionales para segmentar úlceras en las extremidades utilizando validación cruzada y logrando un coeficiente de similitud de 0,794 ($\pm 0,104$) para la región de la úlcera, 0,851 ($\pm 0,148$) para la región de la piel circundante, y 0,899 ($\pm 0,072$) para la combinación de ambas regiones. Por otra parte, en [8] se presenta un enfoque con clasificación haciendo uso de máquinas de soporte vectorial (SVM) para determinar los límites de la herida capturadas con un cuadro de capturas de imágenes, implementando la clasificación en una plataforma de teléfono Nexus 5. Los resultados se comparan con otros clasificadores y muestran que se proporcionan altas tasas de rendimiento global (sensibilidad promedio = 73,3 %, especificidad = 94,6 %) y es lo suficientemente eficiente para un análisis de imágenes

- F. Perez, D. Ballesteros y C. Rey son estudiantes de la Escuela de Ingeniería de Sistemas e Informática, Universidad Industrial de Santander, Colombia, 680002.

basado en teléfonos inteligentes. [9] Adicionalmente, se presenta otro método de clasificación el cual consta de dos etapas que son la extracción del ROI y la clasificación de tejidos. Los resultados preliminares obtenidos: "Dice similarity coefficient" 92%, "percentage area distance" 13% y "area under the curve" (AUC) 95% [10].

Sin embargo, aunque varios de los trabajos expuestos anteriormente presentan una buena tasa de rendimiento, se hace necesario aumentar el número de imágenes de úlceras para obtener resultados más fiables, implementar nuevas modalidades para aumentar el rendimiento del sistema, y resolver el problema de la detección de piel en algunos fondos de fotos muy complejas. Este problema lleva a que algunos fondos que no son de piel se juzguen erróneamente como heridas en los resultados finales.

Por lo tanto, este trabajo aborda una estrategia para la segmentación de heridas crónicas con aprendizaje profundo articulado con técnicas de aumento de datos, ajuste de hiperparámetros y adaptación de dominio.

2 MÉTODO PROPUESTO

En el presente proyecto de investigación se busca segmentar las heridas crónicas de un dataset público de Polonia [13]. Para esto, se trabajó con un conjunto de imágenes las cuales servirán para entrenar una red neuronal que se usará para automatizar la segmentación de las heridas, la cual es una arquitectura de red convolucional U-net 3. Este modelo de red será entrenado con un dataset adicional de imágenes de heridas del municipio de Contratación, el cual será usado en la adaptación de dominio como el dataset fuente, que será adaptado al dataset objetivo (Polonia), el proceso para llevar a cabo la adaptación de dominio fue testeado con diversos métodos, para así quedarnos con el mejor.

Mediante Adaptación de dominio se producirá un nuevo dataset a partir de las imágenes del dataset de Contratación adaptadas al de Polonia. adicionalmente, durante el proceso de entrenamiento las imágenes que la red observa en cada época pasaran por un proceso de aumento de datos. Para la red se implementa el modelo de una red neuronal convolucional de la arquitectura U-net, a la cual se le introducirán las imágenes del dataset y como ground truth las imágenes segmentadas por el médico especialista para que durante seguidas iteraciones la red aprenda a generalizar la segmentación de una herida.

A continuación, procederemos a detallar cada uno de los pasos llevados a cabo en la estructura de nuestro método desarrollado.

2.1 Red neuronal convolucional: U-net

La principal tarea a realizar en este proyecto es la segmentación de imágenes de heridas crónicas, en donde la salida deseada debe ser una imagen binaria cuyos píxeles de color blanco corresponden a lo que se define como herida, y los píxeles negros lo que no es herida (piel normal, fondo de la foto, etc). En una primera instancia, para realizar la correspondiente tarea de segmentación, se utilizó la arquitectura convolucional U-net 1, compuesta por mapas de características multicanal.

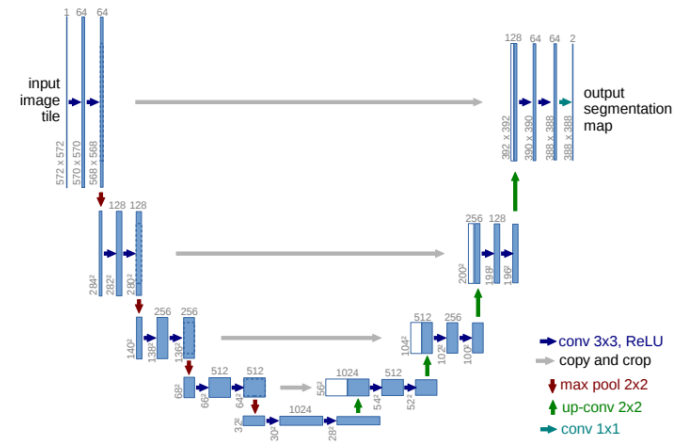


Fig. 1: Arquitectura neuronal U-net (32x32 píxeles en la resolución mas baja). Cada cuadro azul corresponde a un mapa de características multicanal.

A la red convolucional, entra una imagen RGB de una herida de dimensiones 256x256 normalizada entre 0 y 1, y sale una imagen binaria de las mismas dimensiones correspondiente a la segmentación de la herida.

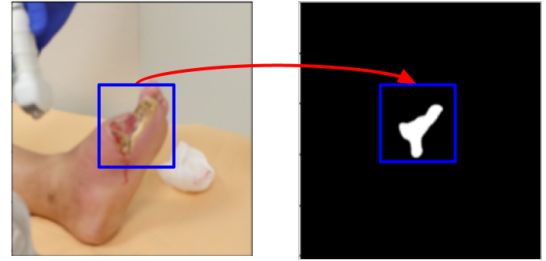


Fig. 2: A la izquierda, una imagen de entrada. A la derecha, segmentación binaria de la herida mediante la red neuronal U-net.

2.2 Métricas

Empleamos dos métricas para evaluar el rendimiento de la red; el "dice" y la "loss" de la binary cross entropy, siguiendo las medidas de verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN), las cuales se pueden calcular teniendo la segmentación verdadera y y la máscara predicha por la red \hat{y} .

$$TP \ \& \ TN = \sum y \odot \hat{y} \quad (1)$$

$$FP = \sum (1 - y) \odot \hat{y} \quad (2)$$

$$FN = \sum y \odot (1 - \hat{y}) \quad (3)$$

Donde \odot corresponde al producto por elementos.

2.2.1 Dice

El coeficiente dice, muestra la similitud entre la segmentación y el ground truth. Este coeficiente se calcula por

$$Dice = \frac{2TP}{TP + FP + FN} \quad (4)$$

2.2.2 Función de pérdida

La función de pérdida es el método por el cual se evalúa la precisión de un algoritmo para modelar los datos otorgados. La función de pérdida empleada en este proyecto de investigación es la pérdida de entropía cruzada o "binary cross entropy" ya que se ajusta a nuestro objetivo. La imagen resultado contiene solo dos posibles pixeles, pixeles de herida o de no-herida. A continuación se muestra la ecuación de la "binary cross entropy":

$$J(\mathbf{w}) = -\frac{1}{N} \sum_{j=1}^N \left[y^{(j)} \log(\hat{y}^{(j)}) + (1 - y^{(j)}) \log(1 - \hat{y}^{(j)}) \right] \quad (5)$$

Donde N es el número de muestras definido por el batch size, j es cada valor de la iteración en la sumatoria, y la matriz de la máscara real y \hat{y} la matriz de la máscara predicha

2.3 Optimizador

Las redes neuronales son algoritmos que modelan un problema con base a unos parámetros, también conocidos como pesos. El trabajo de un optimizador es reducir el error cometido por la red optimizando el valor de los parámetros. El optimizador empleado para nuestra red es la estimación adaptativa de momentos, o Adam. Los momentos permiten determinar el incremento aplicado a los parámetros en cada iteración y determina la siguiente actualización como una combinación lineal entre el incremento de la actual iteración y el incremento realizado anteriormente.

$$m = \beta_1 * m + (1 - \beta_1) * \Delta W \quad (6)$$

$$v = \beta_2 * v + (1 - \beta_2) * \Delta W^2 \quad (7)$$

Donde m y v representan los dos momentos, siendo m el que modela la media de los gradientes a lo largo del tiempo, mientras que v hace lo mismo con la varianza. El valor de β_1 en la mayoría de los casos, equivale a 0.9, mientras que el valor de β_2 se fija en 0.99

2.4 Hiperparámetros

Las redes neuronales de aprendizaje profundo se entrenan utilizando el algoritmo de descenso de gradiente estocástico. El descenso del gradiente estocástico es un algoritmo de optimización que estima el gradiente de error para el estado actual del modelo utilizando ejemplos del conjunto de datos de entrenamiento, luego se actualizan los pesos del modelo utilizando el algoritmo de propagación hacia atrás de errores, llamado back propagation. La magnitud de cambio con la que se actualizan los pesos durante el periodo de entrenamiento se denomina learning rate. Específicamente, el learning rate es un hiperparámetro configurable utilizado en el entrenamiento de las redes neuronales que tiene

normalmente un valor positivo entre el rango de (0.0, 1.0].

En la primera instancia del proyecto, se modificó los valores del learning rate para estabilizar las métricas hasta encontrar aquel menor valor en la métrica loss en la última época, y de esta manera, encontrar el mejor modelo. Para esto, se realizó una primera ejecución para determinar el rango en el que se encontraba el valor más óptimo de learning rate

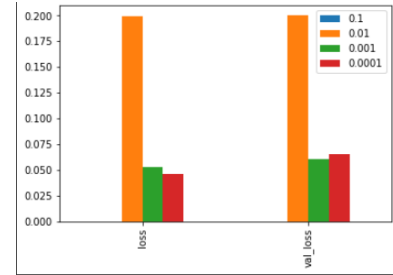


Fig. 3: Gráfica de los resultados obtenidos, cada barra corresponde a un valor de learning rate y su altura corresponde al valor de la loss en la época final

Se observó que en la figura 3 los mejores valores se encontraban en el intervalo de [0.001, 0.0001], es por esto, que se realizó una segunda ejecución para precisar este valor óptimo de learning rate

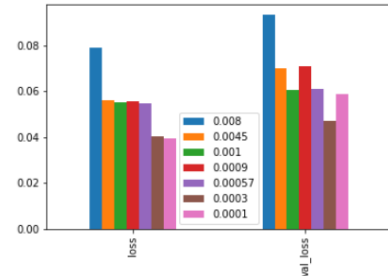


Fig. 4: Resultados obtenidos para la segunda ejecución con distintos valores de learning rate, la altura corresponde al valor de la loss en la época final

A partir del análisis de la figura 4 concluimos que el mejor valor del learning rate es **0.0003**

Una vez establecido el valor del learning rate con el cual se iba a trabajar, determinamos el tamaño del lote que define el número de muestras que se propagan a través de la red para realizar la actualización de los pesos. Realizando varias pruebas y graficando los resultados, se logró determinar el valor apropiado del tamaño del lote que se debía usar: 5

A pesar de que normalmente, se establece que el mejor batch-size es el mayor posible, decidimos realizar esta búsqueda y encontramos que aunque los valores están muy cercanos, no necesariamente tiene que ser el mayor valor de batch-size el mejor.

De esta manera se concluyó que el valor más óptimo para el tamaño del lote para utilizar en el entrenamiento de la red es 5, a saber, la red verá 5 imágenes, calculará la función de pérdida y actualizará los pesos.

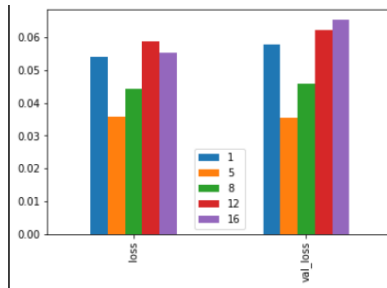


Fig. 5: Grafica de los resultados obtenidos, cada barra corresponde a un valor de batch size y la altura al valor de la loss en la época final del entrenamiento.

2.5 Aumento de datos

Obtener muchas imágenes de heridas etiquetadas es algo costoso, es por eso que en este proceso se hacen ligeras modificaciones a cada imagen de modo que se obtengan más muestras para el entrenamiento de la red. En nuestro caso se utilizaron las siguientes transformaciones: rotación variando el ángulo, dar la vuelta a la imagen horizontal y verticalmente, mover la imagen y cambiando la escala de la imagen.

Todas estas transformaciones nos dan como resultado un mayor número de imágenes para el entrenamiento, y por lo tanto, un mejor aprendizaje de la red neuronal. Este proceso es fundamental para enseñarle a la red la varianza deseada y propiedades de robustez cuando no se disponen de muchas muestras de entrenamiento.

Las transformaciones definidas anteriormente se realizan de manera arbitraria y la intensidad de cada una de ellas varía de forma aleatoria dentro de un rango de valores establecido y en cada iteración se escoge un valor.

Además, el aumento de datos modifica cada imagen original del conjunto de datos pero con valores diferentes en las transformaciones previamente mencionadas. A las respectivas máscaras de segmentación de cada imagen modificada con el aumento de datos, se le realizaron los mismos filtros aplicados a las imágenes para que así la herida siga relacionada con la máscara.



Fig. 6: Resultado de realizar distintas transformaciones a una misma imagen. A la izquierda se muestra la imagen sin modificaciones, a la derecha se muestran múltiples muestras de modificaciones aleatorias.

El aumento de datos aplicado en nuestro proyecto se podía llevar a cabo en con dos métodos diferentes:

2.5.1 Método 1

El primer método, consiste en aplicar el debido proceso del aumento de datos en cada época, la red solicita imágenes, se procesan y se modifican las imágenes en RAM, entran a la

red y una vez la red las procesaba, las imágenes modificadas se eliminan de la memoria RAM y pasa a la siguiente época, para volver a generar nuevas modificaciones a partir de la imagen original, a saber, las imágenes modificadas se generan en la transición de cada época.

2.5.2 Método 2

El segundo método planteado consiste en que el aumento de datos se realiza antes de el entrenamiento, las imágenes modificadas se guardan en el disco duro del dataset para después ser cargadas y puedan ser procesadas por la red. La ventaja de este método es que se pueden controlar el total de nuevas imágenes modificadas generadas, por otro lado, consume gran cantidad de espacio digital.

Se decidió llevar a cabo el 1er método descrito en 2.5.1 para el aumento de datos, debido a que presentaba una mayor velocidad de ejecución respecto al otro método y no era necesario ocupar espacio en el disco.

2.6 Adaptación de dominio

A partir de la definición del dataset fuente y el dataset objetivo se realiza un proceso en el cual se cambian algunos parámetros de las imágenes del dataset fuente para que se adapten al dataset objetivo generando un dataset con características similares al dataset objetivo, eliminando la varianza y el ruido que se presentaban con el dataset original.

Para realizar este proceso se probó con varios métodos, a continuación se nombrará cada uno de los métodos probados de forma cronológica: fast style transfer, sim2real GAN, cyclegan, transferencia de paleta de color.

2.6.1 Fast style transfer

La transferencia de estilo es una técnica de optimización utilizada para tomar dos imágenes, una imagen fuente y una imagen objetivo y combinarlas para que la imagen de salida se vea como la imagen objetivo, pero con el estilo transferido.

Esto se implementa mediante la optimización de la imagen de salida para que coincida con las características de la imagen fuente y las características extraídas de la imagen objetivo. Estas características son extraídas usando una red neuronal convolucional. Se usó la implementación oficial hecha por tensorflow usando los pesos de una red previamente entrenada [14].

2.6.2 Sim2real GAN

En el proyecto original de sim2real se generaban imágenes sintéticas en un entorno 3D, que se asemejaban más o menos a escenas de cirugía hepática laparoscópica. Luego entrenaban a un grupo de redes adversarias generativas (GAN) para traducir estas imágenes para que parezcan imágenes laparoscópicas reales [15], teniendo clara esta metodología, lo que se hizo fue modificarla para transferir el estilo de una imagen del dataset objetivo a una imagen del dataset fuente.

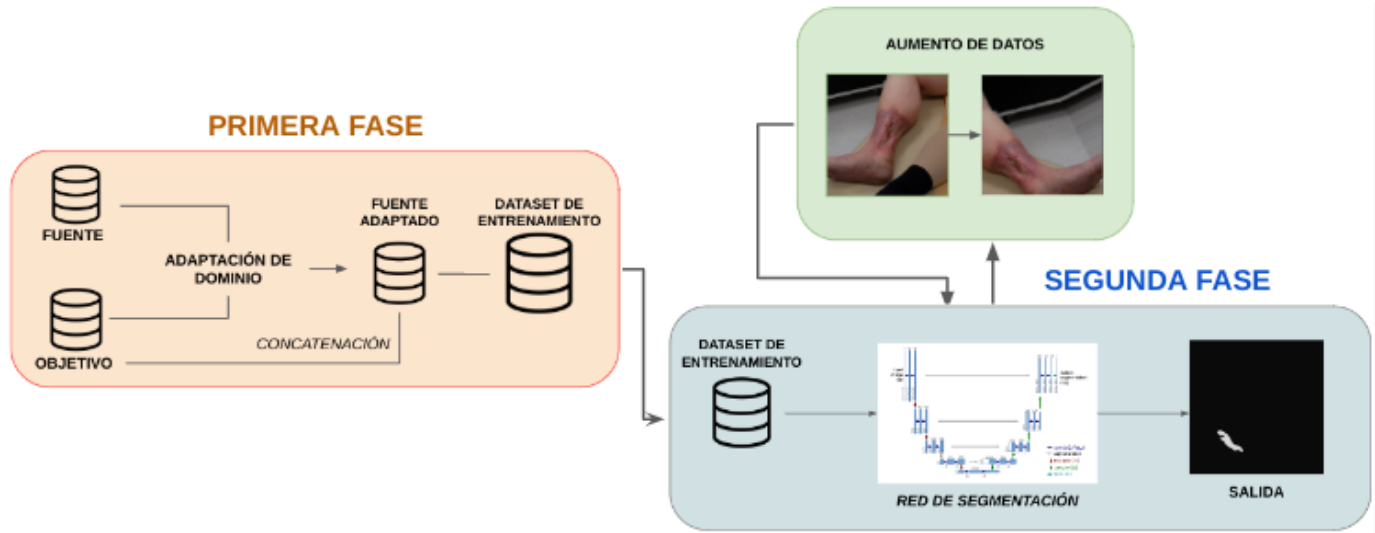


Fig. 7: Estructura general de la implementación, cuya primera parte esta compuesta del proceso de adaptación de dominio, en el que se logra adaptar el dominio del dataset fuente al dataset de origen y de esta manera obtener un nuevo dataset que se concatena con el dataset objetivo. En la segunda fase se introduce este dataset a la red de segmentación y a través de un proceso de aumento de datos, la red es entrenada para obtener la respectiva segmentación.

2.6.3 Cyclegan

Este es un tipo de red adversaria generativa en la que se trata el problema de transferencia de estilo a partir de la reconstrucción de imágenes, funciona a partir de un generador y un discriminador, el trabajo del generador es producir las muestras a partir de el dominio deseado y el trabajo del discriminador es averiguar que la muestra es de el conjunto real o de la que son hechas por el generador, para nuestro caso se utilizo la implementación oficial de tensorflow.

2.6.4 Transferencia de paleta de color

Este método radica en que se tienen dos imágenes, una imagen destino y una imagen paleta, el resultado de la transferencia es hacer que los colores de la imagen destino coincidan con los de la imagen paleta, el método usado es el propuesto por Reinhard y sus colegas en una versión mejorada. Este procedimiento lleva el espacio de color RGB de una imagen a un espacio de color diferente para así poder realizar la transferencia, para mas información acerca de este método dirigirse aquí [16].

Dataset generado. A partir de los procesos descritos anteriormente, se genera un nuevo dataset el cual contiene las imágenes del dataset fuente adaptadas al dataset objetivo, este dataset se concatena con las imágenes del dataset objetivo para así tener un mayor volumen de muestras para entrenar la red neuronal convolucional.

3 RESULTADOS OBTENIDOS

3.1 Datos

Para la implementación del modelo de segmentación se trabajó con dos datasets de imágenes, uno proveniente de un hospital de Polonia de acceso público [14] que contiene

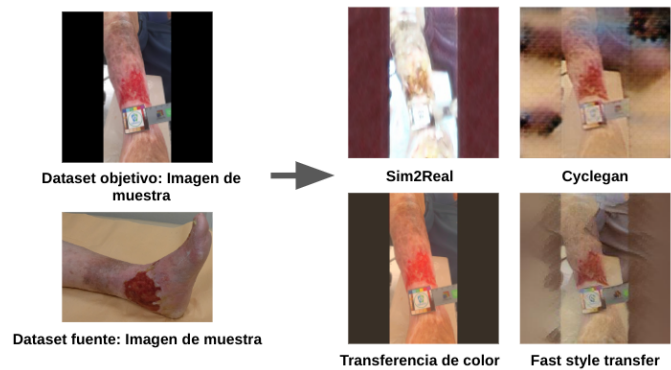


Fig. 8: Muestra de resultados obtenidos para una imagen en específico al pasar por la adaptación de dominio

un total de 188 imágenes de tamaño 1024×1024 a color de heridas crónicas, tomadas de diferentes pacientes con sus respectivas mascarar binarias segmentadas y etiquetas por un doctor. Y el otro dataset local proveniente del municipio de Contratación del departamento de Santander que consta de 182 imágenes de tamaño 640×640 a color de heridas crónicas, de las cuales 12 fueron eliminadas ya que no tenían su respectiva máscara binaria.



Fig. 9: Imágenes de muestra del dataset de Contratación

TABLE 1: Resultados obtenidos en este proyecto

Dataset	método	métricas	
		loss	dice
Polonia	U-net base	0.047	0.5006
	U-net + hiperparametros ajustados	0.027	0.5689
	U-net + hiperparámetros + aumento de datos	0.018	0.7784
Polonia + Contratacion + Adaptacion de dominio	fast style transfer	0.020	0.6548
	sim2real	0.025	0.6797
	cycleGAN	0.026	0.3838
	transferencia paleta de color	0.019	0.7036



Fig. 10: Imágenes de muestra del dataset de Polonia

3.2 Detalles de la implementación

Datasets. Las imágenes se reajustaron para evitar el error de OOM (Out of memory) ya que el código fue ejecutado en Google Colab, de manera que cada imagen se reajusta a un tamaño de 256 píxeles de alto por 256 píxeles de ancho. Las respectivas máscaras de cada dataset de imágenes, proporcionan el respectivo ground truth para la red de segmentación. Al concatenar el dataset fuente adaptado y el dataset objetivo, surge un dataset con 333 imágenes con las mismas medidas anteriormente mencionadas.

Código. Para llevar a cabo la implementación de el modelo de aprendizaje automático, se implementó el código en el lenguaje de programación python, en la herramienta proporcionada por Google, Google Colab, la cual nos permitió ejecutar el código en Python en nuestro navegador web. Hicimos uso de algunas librerías como Tensorflow en su version 2.8.2, Keras, Pytorch, Sklearn, ImageDataAugmentor y Albumentations; Tensorflow que es una biblioteca de código abierto dirigida al aprendizaje automático además de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones usados por humanos; Keras, biblioteca de código abierto de redes neuronales, cuyo fin es realizar la construcción por bloques de la arquitectura de cada red neuronal, Sklearn que nos proporciona múltiples funciones para cargar datos y organizarlos; ImageDataAugmentor es una librería que está dirigida a facilitar la implementación del aumento de datos para cualquier red neuronal; Albumentations facilita la implementación de las transformaciones para cada imagen, con múltiples métodos para escoger la modificación que mas nos convenga.

Adaptación de dominio Para la adaptación, se utilizó la implementación oficial de cada método y se entrenó para posteriormente guardar los resultados para cada imagen, en el caso de la transferencia de paleta de color, se construyó un script adicional para conseguir la transferencia de un lote de imágenes a partir de una imagen paleta.

Segmentación. El modelo de red neuronal U-net fue implementado en tensorflow para llevar a cabo la tarea de

segmentación, la cual fue entrenada con el dataset concatenado entre dataset adaptado y el dataset objetivo con 100 épocas y los hiperparámetros definidos en la sección 2.4, las imágenes antes de entrar a la red se normalizan entre 0 y 1. Del dataset de Polonia, se utilizó el 20% de las imágenes para testear, a saber, 150 imágenes para entrenamiento y 38 para test, y de ahí sacar las métricas que se muestran en la tabla 1

3.3 Estudio de ablación

Los resultados de todas las simulaciones se encuentran en la tabla 1, allí se muestran las métricas del modelo respecto a los datos de test para cada una de los métodos utilizados, para un dataset sin adaptación de dominio, se muestra el procedimiento usado para cada simulación, posteriormente para los 2 datasets se estableció como base el procedimiento de 1 dataset con unet + hiperparámetros + aumento de datos, y a partir de esta base se usaron diferentes métodos de adaptación de dominio que se muestran en la tabla con sus respectivos resultados para cada métrica.

3.4 Resultados cualitativos

Como se puede observar en la tabla 1, el mejor resultado para las dos métricas utilizadas se da cuando se utiliza un dataset, se observa que se sigue un aumento progresivo en la métrica: dice, conforme se agregaban procesos adicionales al método, como pueden ser el ajuste de hiperparametros y el uso de aumento de datos. Para el caso de el entrenamiento de los dos datasets con adaptación de dominio, los métodos de fast style transfer y sim2real tiene resultados similares entre si pero inferior para el mejor resultado con 1 dataset, cuando se hace la adaptación con una cycleGAN, las métricas tienen resultados pobres incluso menores que cuando se utiliza la unet-base, esto se puede explicar ya que la cycleGAN reconstruye la imagen y durante ese proceso de reconstrucción destruye la forma de la herida y agrega demasiado ruido a la imagen, para el caso de la transferencia de paleta de color es el que muestra mejores resultados cuando se utiliza adaptación de dominio, ya que este proceso de adaptación no destruye la forma de la herida y relaciona los colores entre los datasets, que es un factor importante para la segmentación en la red, sin embargo, sus resultados no alcanzan los conseguidos con el entrenamiento de 1 solo dataset

Durante el entrenamiento del mejor resultado: unet+ hiperparametros + aumento de datos, se observó que la métrica dice aumenta constantemente conforme las épocas aumentan.

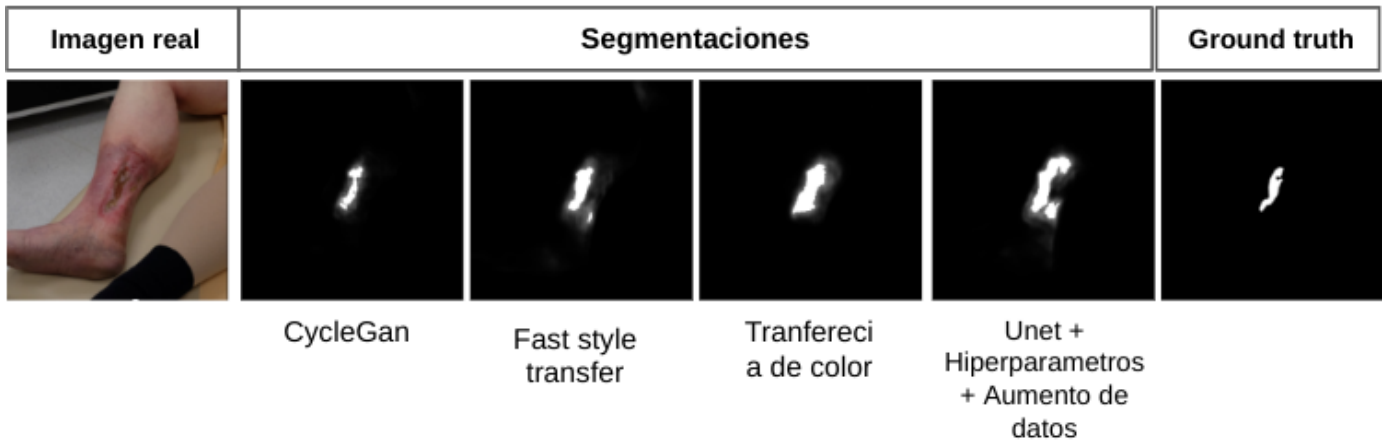


Fig. 11: Resultados de la segmentación de una herida usando los diferentes métodos usados entrenados con 100 épocas



Fig. 12: dice vs épocas

Es por esto que con la expectativa de maximizar la métrica dice en esta red se entrenó con 100 épocas más, para un total de 200 épocas

método	métricas	
	loss	dice
Unet + hiperparametros + aumento de datos	0.014	0.855

TABLE 2: Resultado obtenidos para red entrenada con 200 épocas

Como se observa en la tabla 2 efectivamente al aumentar el numero de épocas, la métrica dice aumenta su valor, es esta red entrenada la propuesta en la que la generalización y la segmentación consiguen mejores resultados.

3.5 Resultados visuales

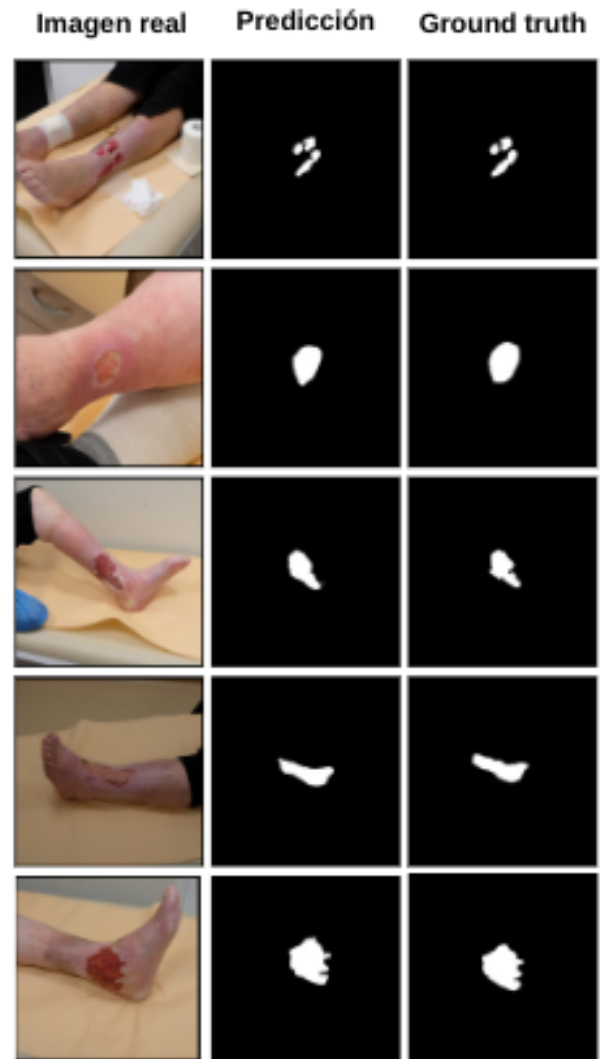


Fig. 13: Resultados visuales para la red entrenada con 200 épocas en el dataset de Polonia

4 CONCLUSIONES

En este proyecto se desarrollo una estrategia para la segmentacion de heridas en fotografias RGB. Se exploraron diversas tecnicas de programacion como el aumento de datos, ajustes de hiperparametros y la adaptacion de dominio. El objetivo de este proyecto es la segmentacion de heridas de Polonia, se exploro el aumento de la exactitud apoyandose en un dataset de contratacion, sin embargo, esto no aumento la precision. Finalmente, la maxima exactitud de segmentacion alcanzada fue 0.855 en la metrica "dice".

REFERENCES

- [1] "Diabetes - OPS/OMS | Organización Panamericana de la Salud". PAHO/WHO | Pan American Health Organization. <https://www.paho.org/es/temas/diabetes> (accedido el 28 de agosto de 2022).
- [2] "La diabetes y los problemas de los pies | NIDDK". National Institute of Diabetes and Digestive and Kidney Diseases. <https://www.niddk.nih.gov/health-information/informacion-de-la-salud/diabetes/informacion-general/prevenir-problemas/pies-piel> (accedido el 28 de agosto de 2022).
- [3] Olaf Ronneberger, Philipp Fischer, Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", Computer Science Department and BIOSS Centre for Biological Signalling Studies, 2015.
- [4] "Withdrawal: Torsello A, et al. Congenital Peritoneal Encapsulation and Nephrotic Syndrome. J Forensic Sci 2020 (<https://onlinelibrary.wiley.com/doi/10.1111/1556-4029.14292>)", Journal of Forensic Sciences, vol. 65, n.º 3, p. 1026, enero de 2020. Accedido el 28 de agosto de 2022. [En línea]. Disponible: <https://doi.org/10.1111/1556-4029.14292>
- [5] K. M. Buckley et al., "Reducing the risks of wound consultation: adding digital images to verbal reports," Wound Ostomy Cont. Nurs., vol. 36, no. 2, pp. 163–170, 2009.
- [6] Y. Manoj Kumar, M. Dhane Dhiraj, M. Gargi y C. Chandan, "Segmentation of Chronic Wound Areas by Clustering Techniques Using Selected Color Space", Ingenta connect, vol. 3, 2013.
- [7] C. Wang, X. Liu, Y. Peng, S. Jin y F. Li, "A Composite Model of Wound Segmentation Based on Traditional Methods and Deep Neural Networks", Hindawi, 2018, art. n.º 4149103.
- [8] M. Goyal, N. D. Reeves, A. K. Davison, S. Rajbhandari, J. Spragg y M. H. Yap, "DFUNet: Convolutional Neural Networks for Diabetic Foot Ulcer Classification", IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, n.º 5, pp. 728–739, octubre de 2020. Accedido el 28 de agosto de 2022. [En línea]. Disponible: <https://doi.org/10.1109/tetci.2018.2866254>
- [9] L. Wang, P. C. Pedersen, E. Agu, D. M. Strong y B. Tulu, "Area Determination of Diabetic Foot Ulcer Images Using a Cascaded Two-Stage SVM-Based Classification", IEEE Transactions on Biomedical Engineering, vol. 64, n.º 9, pp. 2098–2109, septiembre de 2017. Accedido el 28 de agosto de 2022. [En línea]. Disponible: <https://doi.org/10.1109/tbme.2016.2632522>
- [10] B. García-Zapirain, M. Elmogy, A. El-Baz y A. S. Elmaghraby, "Classification of pressure ulcer tissues with 3D convolutional neural network", Medical & Biological Engineering & Computing, vol. 56, n.º 12, pp. 2245–2258, junio de 2018. Accedido el 28 de agosto de 2022. [En línea]. Disponible: <https://doi.org/10.1007/s11517-018-1835-y>
- [11] "Cómo Entrenar una CNN Usando Adam en Keras - DataSmarts Español". DataSmarts Español. <https://datasmarts.net/es/como-entrenar-una-cnn-usando-adam-en-keras/> (accedido el 28 de agosto de 2022).
- [12] O. Ronneberger, P. Fischer y T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", UNI Freiburg, vol. 9351, mayo de 2015.
- [13] M. Krecichwost et al., "Chronic wounds multimodal image database," Computerized Medical Imaging and Graphics, vol. 88, p. 101844, Mar. 2021, doi: 10.1016/j.compmedimag.2020.101844.
- [14] "Fast Style Transfer for Arbitrary Styles | TensorFlow Hub," TensorFlow, Apr. 27, 2022. https://www.tensorflow.org/hub/tutorials/tf2_arbitrary_image_stylization
- [15] Micha Pfeiffer, Isabel Funke, Maria R. Robu, Sebastian Bodendstedt, Leon Strenger, Sandy Engelhardt, Tobias Roß, Matthew J. Clarkson, Kurinchi Gurusamy, Brian R. Davidson, Lena Maier-Hein, Carina Riediger, Thilo Welsch, Jürgen Weitz and Stefanie Speidel. Generating large labeled data sets for laparoscopic image processing tasks using unpaired image-to-image translation, 2019; arXiv:1907.02882.
- [16] T. Johnson, "Enhanced Image-Colour-Transfer-2," GitHub, Jun. 2022. <https://github.com/TJCoding/Enhanced-Image-Colour-Transfer-2> (accessed Aug. 30, 2022).