Autora: Cristina Martínez Toledo Tutora: Virginia Sanz Pérez Centro: IES Alkal’a Nahar

**DESARROLLO DE UN SOFTWARE PARA EL ANÁLISIS DE MIGRACIÓN CELULAR EN CULTIVOS**

**Resumen:**

La migración celular es el movimiento dirigido de las células en respuesta a un estímulo. Es necesaria en distintos procesos biológicos como la reparación de tejidos, el desarrollo embrionario o ciertas patologías como por ejemplo el cáncer, donde es necesario que las células tengan capacidad migratoria para poder metastatizar. En el desarrollo de fármacos contra el cáncer, los ensayos destinados a evaluar la capacidad de migrar de las células tumorales se realizan como un paso fundamental: impedir esta migración y, por lo tanto, la metástasis, es un objetivo clave. El ensayo de herida in vitro es el método más frecuentemente utilizado para el estudio de la migración celular, aunque el análisis de las imágenes obtenidas con los softwares tradicionalmente empleados es tedioso y requiere el ajuste manual de muchos parámetros, por lo que está sometido a la valoración subjetiva del investigador. El objetivo del proyecto es la adaptación de un software para ciencia de los materiales basado en LabVIEW con el objetivo de desarrollar un programa para el análisis de imágenes de ensayos de herida que resulte rápido de usar, fácil, preciso y objetivo.

**Abstract:**

Cell migration is the directed movement of a cell in response to a stimulus. It is fundamental for many biological processes such as tissue repair and embryo development. It is also related to some important pathologies such as cancer metastasis, where cell migration studies are extremely important to improve the treatment of malignant neoplasms. The *in vitro* scratch or wound healing assay is a frequently used method for the study of cell migration. However, obtaining statistically significant results requires time-consuming image analysis procedures. Furthermore, traditionally used softwares for image analysis like ImageJ have some downpoints, such as the manual adjustment of too many parameters. The aim of this study is to review the main characteristics of cell migration and how it is studied, as well as the development of a software prototype for the analysis of scratch assay images with a minimal set of manually adjustable parameters. The development of the software was based on the programming environment LabVIEW, and artificial intelligence algorithms (the neural network U-Net) were used for automated image segmentation. The program was tested for accuracy by comparing the data from previously analyzed scratch images with those obtained after the image analysis with the software developed for this study. Similar results were obtained in less time and in a more automated and reproducible way. Therefore, we have developed an user-friendly prototype software to accurately analyze scratch assay images, fast and easy to use without extensive computer skills.

**1. Introducción**

La migración celular es el movimiento dirigido de las células en respuesta a un estímulo. Es un proceso celular fundamental que comienza durante el desarrollo embrionario, y tiene gran importancia en procesos biológicos como la respuesta inmunitaria, el desarrollo del sistema nervioso o el mantenimiento de la homeostasis de los tejidos y la reparación de heridas, así como en la progresión de diversas patologías, como el cáncer. Su estudio y comprensión es fundamental para el desarrollo de nuevas estrategias terapéuticas para acelerar y mejorar los procesos de reparación de heridas, la formación de tejidos artificiales, o el tratamiento de las patologías relacionadas con este proceso (Hulkower and Herber, 2011; Fernández-Espartero, 2013; Merino-Casallo *et al*., 2022).

**1.1 Concepto de cáncer y metástasis**

El cáncer es una de las principales causas de mortalidad en el mundo. Cuando una célula normal envejece, se daña o tiene alguna anomalía, muere. Sin embargo, si surge algún fallo en este proceso, estas células anormales adquieren resistencia a la apoptosis y se dividen de forma descontrolada, formando un tumor maligno. Estas células son capaces de migrar e invadir el tejido que las rodea, propagándose a otras partes del cuerpo, originando metástasis. Los tumores metastásicos son los más peligrosos y los que causan la mayoría de muertes por cáncer, ya que experimentan una alta tasa de mutación espontánea y suelen ser extremadamente resistentes a las terapias actuales (Bravo-Cordero, Hodgson and Condeelis, 2012).

**1.2 Etapas de la oncogénesis, migración celular y metástasis**

Para que un tumor metastatice, debe pasar por distintas “etapas”. Primero, las células adquieren mayor capacidad proliferativa y acumulan alteraciones que las hacen más resistentes a la muerte celular (adquisición de inmortalidad). Después adquieren la capacidad de degradar los componentes de la matriz extracelular que forman el “límite” del tejido, se separan del tejido o epitelio y adquieren capacidad migratoria (individual o colectiva), proceso denominado “transición epitelio-mesenquimal” (EMT), que también es fundamental en otros procesos en los que interviene la migración celular (Benedetti and Reyes, 2015). Una vez en el estroma, las células siguen proliferando, generando diversas poblaciones celulares (invasión local); si las masas formadas por el tumor son grandes, acaban teniendo dificultad para nutrirse, por lo que ciertas células pueden adquirir capacidad angiogénica y/o linfangiogénica. A la célula tumoral le resultará más fácil moverse y pasar entre las células endoteliales de estos neovasos para finalmente alcanzar el torrente circulatorio. Aquellas células especialmente preparadas para resistir la acción del sistema inmunitario podrán sobrevivir y salir del torrente sanguíneo, llegando a otros órganos y colonizando su estroma, formando nuevas masas en el proceso inverso a la EMT denominado “transición mesénquimo-epitelial” (MET). El ciclo volverá a empezar y el cáncer se expandirá progresivamente (Arvelo, Sojo and Cotte, 2016; Hincapie and Gallego-Gómez, 2020).

Muchos de estos acontecimientos pueden ocurrir de forma simultánea, ya que los tumores no son masas homogéneas. Durante las sucesivas divisiones celulares, surgen diferentes poblaciones celulares que pueden presentar características diversas. Así, como si de un ejército invasor se tratara, surgen células capaces de degradar la matriz extracelular con mayor eficiencia, otras capaces de secretar factores inductores de la angiogénesis, células capaces de “engañar” al sistema inmunitario, otras productoras de factores que contribuyen a la supervivencia de las demás: un “ejército” cuya misión es generar y proteger poblaciones celulares capaces de atravesar las paredes vasculares, sobrevivir en el torrente circulatorio, llegar a un órgano diana, implantarse en dicho órgano y reiniciar el ciclo de proliferación-invasión característico de los tumores malignos.

En conclusión, la migración celular en tumores es fundamental para que estos puedan adquirir la capacidad de metastatizar. Debido a ello, el estudio y comprensión de este proceso permitiría el desarrollo de nuevas alternativas para el tratamiento de esta enfermedad, así como nuevas técnicas que puedan prevenir o tratar la metástasis de forma efectiva, lo que podría reducir significativamente el riesgo de muerte por cáncer. Además, frenar la migración en el tumor primario también podría ayudar a limitar o impedir la invasión local incluso antes de que se produzca metástasis (Palmer et al., 2011; Bravo-Cordero, Hodgson and Condeelis, 2012).

**1.3 Estudio de la migración celular y análisis tradicional de las imágenes: el ensayo de herida**

**1.3.1 Definición del ensayo de herida**

De los métodos utilizados para estudiar la migración celular colectiva, el ensayo de herida (“wound healing assay”) es el más conocido y utilizado debido a su simplicidad y relativo bajo coste. Se basa en la observación y estudio de una monocapa de células a la cual se le ha realizado una “herida” o brecha, la cual intentarán cerrar a lo largo del tiempo. Una vez creada la herida, se tomarán imágenes de esta en la misma zona de la brecha cada cierto intervalo temporal, así como al comienzo (0h) y al final del experimento. Debido a la importancia del estudio de la migración en el cáncer, los ensayos de herida son un paso fundamental en el desarrollo de medicamentos antineoplásicos (*Molecular Devices*, no date; Nyegaard, Christensen and Rasmussen, 2016; Nunes and Dias, 2017). Una vez adquiridas las imágenes, se deben analizar con un software de análisis de imagen.

**1.3.2 Análisis tradicional de las imágenes y limitaciones.**

Entre los programas utilizados para analizar las imágenes de los ensayos de herida, el más común es ImageJ, un software de libre acceso y dominio público desarrollado en Java y principalmente orientado al campo de las ciencias de la salud. A pesar de tener una gran variedad de usos y funciones, se trata de un programa muy complejo y poco intuitivo. Hay muchos parámetros y medidas que requieren ser ajustados manualmente, por lo que hay demasiada interacción subjetiva del investigador que puede llegar a afectar a los resultados de forma significativa. Además, debido a su complejidad, el análisis de imágenes resulta tedioso y consume mucho tiempo. Debido a esto, se han creado varios *plugins* y programas alternativos para facilitar el análisis de ensayos de herida, aunque estos suelen ser de pago, difíciles de conseguir, instalar o utilizar, o siguen dependiendo mucho de la interacción del investigador. (Michael et al., 2004; Suarez-Arnedo et al., 2020).

**2 Conceptos previos**

**2.1 LabVIEW**

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) es un entorno de programación con un lenguaje visual gráfico creado por National Instruments (NI). Este lenguaje se basa en la conexión mediante cables de iconos gráficos en un orden secuencial formando un diagrama. En cada programa hay dos ventanas; el panel frontal o interfaz de usuario y el diagrama de bloques, el programa propiamente dicho. Cada diagrama de bloques o programa independiente es llamado “VI”. En un VI se pueden introducir otros programas, llamados “subVI”, que realizarán funciones concretas dentro del principal.

**2.2 Deep learning y U-Net**

*Deep learning* es una rama del *machine learning*, a su vez parte del campo de la inteligencia artificial, que utiliza redes neuronales artificiales, algoritmos basados en el funcionamiento de las redes neuronales de los seres vivos. Este tipo de red neuronal aprende a reconocer patrones en una base de datos (un *input*) generando una predicción en datos del mismo tipo o similares (*output*). U-Net es una red neuronal convolucional de deep learning utilizada principalmente para tareas de visión artificial, sobre todo para segmentación semántica de imágenes. La segmentación de imágenes es el proceso de dividir una imagen en partes o regiones. La segmentación semántica es un algoritmo de deep learning que asocia una etiqueta o categoría a cada píxel presente en una imagen. Como su propio nombre indica, U-Net es una red con una estructura en forma de U. Funciona utilizando un algoritmo de optimización, el gradiente descendente, mediante el cual el modelo intenta minimizar lo máximo posible una función de error (*loss function*). Esta función cuantifica el error en el output producido por el modelo, es decir, cuánto varía la solución o predicción ofrecida por el programa de la “respuesta” que le hemos ofrecido en la base de datos. Después, el programa ajusta sus parámetros para minimizar esta función de error y acercarse más al resultado esperado. Además, debido a esta estructura, U-Net es bastante eficaz incluso con una base de datos limitada, al contrario que otros modelos de deep learning en los cuales se necesitan grandes series de datos para un correcto aprendizaje (Ronneberger, Fischer and Brox, 2015; Rédac, 2022) .

**2.3 Hiperparámetros en redes neuronales**

En una red neuronal, los hiperparámetros son las variables ajustables que controlan el proceso de entrenamiento de un modelo. Sus valores se indicarán antes del entrenamiento y serán fijos durante este, por lo que es importante buscar un valor adecuado para cada uno para optimizar lo máximo posible el proceso de aprendizaje. Además de ajustar los hiperparámetros, es importante para el entrenamiento tener cierta variedad en las imágenes de la base de datos (Na8, 2021). El *epoch* es el número de ciclos completos a través de todos los datos de entrenamiento. En general la base de datos no puede pasar completa por el programa, por lo que se divide en varios lotes o “*batches*”. El “*batch size”* es el número de datos dentro de cada uno de estos batches. Las “*iterations*” indican el número de veces que los parámetros del algoritmo son ajustados, número igual a la cantidad total de batches que hace falta para completar un epoch. Si el batch size es un número muy grande, a pesar de que entrene más rápido, puede que el modelo no aprenda las características, detalles y patrones importantes para hacer predicciones correctas (Brownlee, 2022). El *learning rate*, con un valor entre 0.0 y 1.0, controla cuánto cambian los parámetros del modelo cada vez que estos se actualizan. Si este valor es demasiado alto los cambios realizados serán más rápidos pero el modelo podría aprender de forma equivocada, y si es demasiado bajo el proceso tardaría demasiado y podría quedarse atascado (Jordan, 2020).

**3. Hipótesis y Objetivos**

Planteamos como hipótesis de trabajo que LabVIEW y U-Net podrían ser herramientas de utilidad para el desarrollo de un software capaz de realizar el análisis de ensayos de herida. Esto podría constituir un método preciso para el análisis de migración celular, más rápido, sencillo de utilizar y automático que los softwares tradicionales. Debido a las dificultades y problemas que presenta este análisis mediante métodos tradicionales, se ha planteado como objetivo principal del proyecto el desarrollo de un prototipo de software, basado en LabVIEW y U-Net, mediante la modificación de un programa para ciencia de los materiales y su adaptación para el análisis de ensayos de herida.

**4. Material y métodos**

**4.1 Materiales**

Se ha empleado la plataforma de desarrollo LabVIEW junto con el módulo NI Vision, una librería de LabVIEW utilizada principalmente para el desarrollo de aplicaciones de análisis y procesamiento de imagen, así como el software base previamente desarrollado para detección de bordes en materiales inorgánicos, cedido por el ingeniero Sergio Perosanz Amarillo (Universidad Politécnica de Madrid). También se utilizó un código de deep learning para la segmentación semántica de las imágenes basado en U-Net. Para la explicación y obtención de este código se ha realizado el curso online de Coursera “Deep Learning with PyTorch: image segmentation”. Para la programación del módulo de U-Net y su posterior uso en el software desarrollado, fue necesaria la descarga de la versión 3.9 de Python junto a las siguientes librerías: OpenCV, Albumentations, Pandas, Segmentation-models-pytorch y torch-torchvision. Para la creación del banco de datos necesario para entrenar y probar el modelo de deep learning, se utilizaron imágenes de ensayos de herida de uso público del repositorio “The Broad Bioimage Benchmark Collection” (BBBC), así como imágenes proporcionadas por la Dra. Marina Lasa (Universidad Autónoma de Madrid-CSIC) y por la Dra. María del Val Toledo (Dpto. Biomedicina y Biotecnología, UAH), siendo en total 126 imágenes con sus respectivas máscaras en blanco y negro. Por último, para la obtención y comparación de resultados se emplearon imágenes de la página de 4Dcell (4Dcell, no date) previamente analizadas con ImageJ.

**4.2 Método experimental: desarrollo y funcionamiento del software**

Algunos datos que se analizan en los estudios de migración celular, y que se obtienen con este programa, son el área de la herida, la distancia media entre los bordes de la brecha y la desviación estándar, además de una imagen en la que esté marcado el borde de la herida y una máscara en blanco y negro. También se obtiene una lista con todas las distancias entre bordes de la herida (por cada fila de píxeles).

**4.2.1 Entrenamiento del modelo de deep learning con U-Net**

Ciertos softwares de análisis de imagen que requieren del uso de funciones como thresholds, utilizados en este caso para la detección de bordes, es común que no funcionen de manera precisa si la imagen proporcionada tiene ruido o no tiene suficiente contraste, lo cual es muy común en las imágenes obtenidas en los ensayos de herida. Para solucionar este problema se ha utilizado un algoritmo de deep learning para la segmentación semántica de las imágenes. En este caso, las dos categorías que se diferenciarán serán la brecha y el resto de la imagen. El banco de imágenes introducido para entrenar al modelo consta de las imágenes de distintos ensayos de heridas y sus correspondientes máscaras de capa, en las cuales la brecha aparece de color negro y las células del cultivo en blanco. Para mejorar el rendimiento del proceso de aprendizaje del modelo y aumentar su eficiencia y precisión fue necesario ajustar los valores de los hiperparámetros; finalmente se determinó un epoch de 25, un learning rate de 0.003 y un batch size de 16.

**4.2.2 Programa de LabVIEW para detección de bordes y análisis**

El programa completo está compuesto de un VI principal, que contiene la interfaz de usuario, y cinco secundarios, cada uno de los cuales tiene una función concreta. Estas funciones se unen en el VI principal para que el programa completo trabaje de forma conjunta. El bloque principal de este VI es un “*Case structure*”: al pulsar cada uno de los botones se ejecutará la función correspondiente a ese “caso”. Hay 5 casos posibles, además del 0 (por defecto). La estructura general del programa e interfaz de usuario forman parte del programa original para ciencia de los materiales.

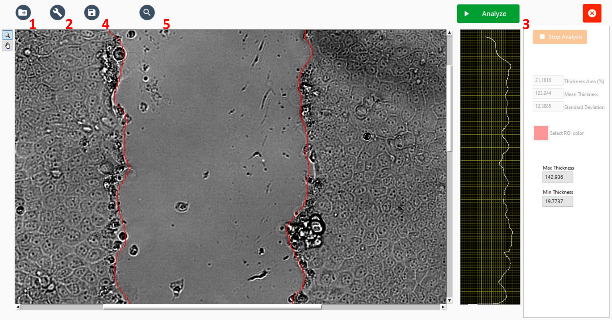


Figura 1: interfaz de usuario del software, donde se indica el botón correspondiente a cada caso, que se corresponde con el orden en el que el usuario deberá pulsar cada botón.

Primero (caso 1), se abrirá la imagen seleccionada por el usuario y traducirá a escala de grises U8 para poder ser utilizada en el programa. Después (caso 2), se deben calibrar las imágenes antes de realizar los cálculos necesarios, es decir, hacer la equivalencia entre los píxeles que tiene una determinada distancia en la imagen y los micrómetros a los que equivaldría en la realidad. Las dos medidas serán introducidas por el usuario: los píxeles se determinarán al “dibujar” una línea en la imagen, y el valor de micrómetros será ajustado manualmente. Este SubVI forma parte del programa original.

En el caso 3 hay varios módulos que intervienen. El primero contiene el algoritmo de U-Net, que está basado en el lenguaje de programación python, en su versión 3.9, por lo que para utilizarlo se debe añadir un módulo de python (*“Python node”*), al cual se le introducen distintos datos (la versión utilizada, el modelo previamente entrenado,...). El array que salga de este módulo será convertido de nuevo en una imagen, una máscara en blanco y negro, que estará comprimida, por lo que se deberá aumentar su tamaño al de la imagen original. Al ampliar la imagen aparecen ciertos píxeles grises que pueden dar lugar a error, por lo que mediante un threshold se toman los valores de gris entre 250 y 255 como blanco y los que están por debajo de 250 como negro. Por último, se calcula el porcentaje de área que ocupa la brecha sobre la imagen total.

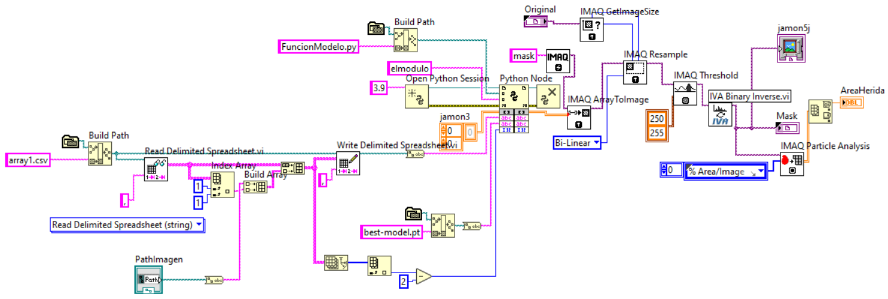
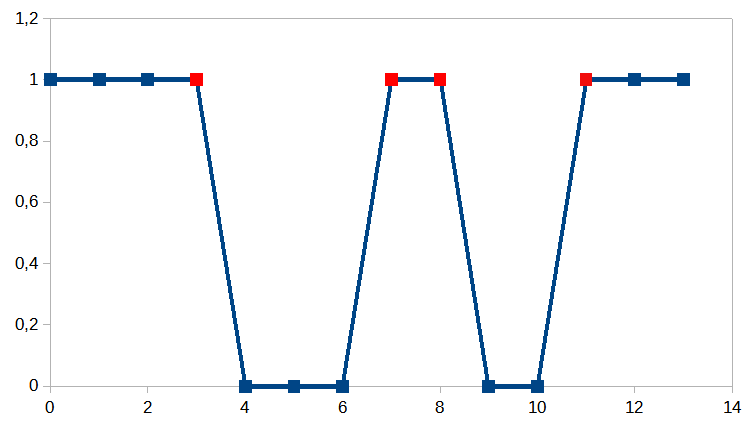


Figura 2: Diagrama de bloques del subVI que contiene el módulo de U-Net.

El siguiente módulo detectará y marcará el borde de la herida, y calculará el resto de datos necesarios para el análisis. El input será la máscara obtenida en el módulo de U-Net. Primero, por cada fila de píxeles se buscará cuáles son de color blanco, correspondientes a la zona de la imagen llena de células. Como la máscara está en binario, se guardarán los índices de los píxeles cuyos valores estén por encima de un threshold que mide 0.5, es decir, que sean blancos. En el siguiente bucle se busca dónde hay un “salto” de blanco a negro para encontrar los bordes. Para esto, a cada píxel de cada fila se le va sumando 1 y comparando los resultados esperados con los que el programa ha guardado en el bucle anterior. Se guardarán los píxeles en los cuales hay un salto entre valores (siempre que no sea el primero o el último) como bordes de la herida y se marcará un punto del color seleccionado por el usuario en todos ellos. Por último, se tomará el primer y último píxel guardado de cada fila para calcular la distancia entre bordes (espesor), que se guardarán en una matriz. Con estos datos el programa calculará la media de todas las distancias y la desviación estándar.

Figura 3: Gráfica de valores hipotéticos de una fila de píxeles. Los píxeles 0-3, 7-8 y 11 en adelante son blancos, por lo que estos números (índices) serán almacenados por el primer bucle. En el segundo bucle, al hacer la suma del píxel 3+1, se comparará el valor teórico que debería dar (4) con el que está guardado en el programa (7). Al ser estos valores distintos, se tomarán como borde los píxeles 3 y 7. Después se seguirá analizando de la misma forma cada píxel de cada fila de la imagen.

Por último (caso 4), se guardan los datos obtenidos por el programa. Al pulsar el botón aparecerá un menú preguntando al usuario cuáles de los datos obtenidos desea guardar. Dependiendo de qué opciones hayan sido seleccionadas, se ejecutará uno o varios de los siguientes subdiagramas. El subdiagrama 1 sirve para guardar todos los datos numéricos calibrados, el segundo sirve para guardar la máscara de capa en valores de 0-255, y el tercero sirve para guardar la imagen de la herida con el overlay. El caso 5 se puede pulsar en cualquier momento y sirve para ajustar el tamaño de la imagen al de la pantalla de la interfaz de usuario para visualizarla mejor. Este bloque forma parte del programa original para ciencia de los materiales.

**5. Resultados finales y discusión**

El programa es capaz de procesar las imágenes de forma rápida, y la interfaz de usuario es sencilla de comprender y utilizar. Únicamente se requiere que el usuario realice la calibración de la imagen, y el resto del análisis se completará en apenas unos segundos, por lo cual resulta ser un análisis automatizado y muy reproducible. Las líneas del borde obtenidas y la máscara en blanco y negro se ajustan con bastante precisión a la realidad, de forma similar e incluso superior a la proporcionada en varias de las imágenes de prueba (descritas en material y métodos). Los espacios libres de células de la herida, expresados en porcentaje sobre el total de la imagen, calculados con el programa tuvieron una diferencia del 7.38% de media al compararlos con los resultados proporcionados por la página de 4Dcells (Figuras 4 y 5). Esto se debe a la mayor precisión del programa desarrollado al determinar los bordes de la herida, por lo que el resultado es más aproximado a la realidad.

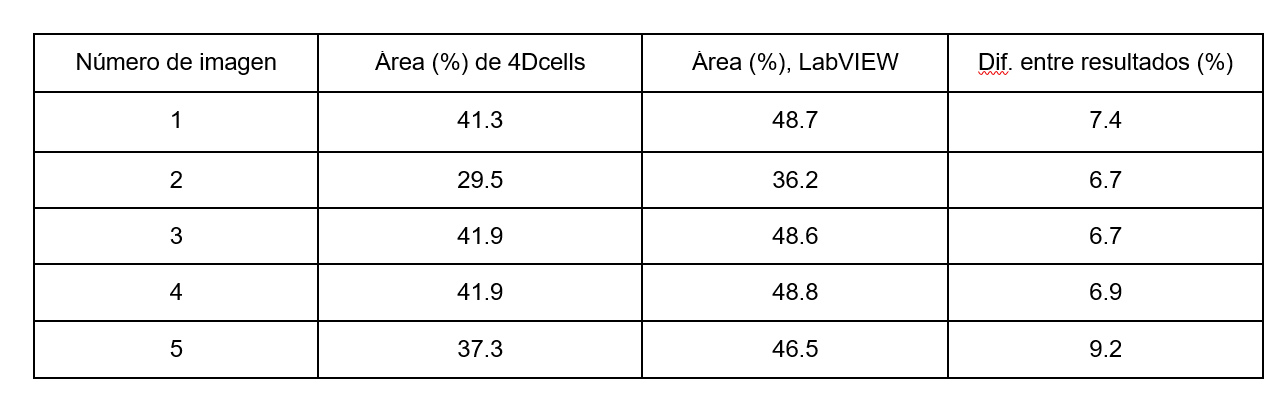
****

Figura 4: Porcentajes de área obtenidos al analizar varias imágenes con el software desarrollado, y comparación con los resultados ya publicados para las mismas imágenes, dando un porcentaje de diferencia entre ambos para el área de cada una. Al calcular la media de los distintos porcentajes de diferencia el resultado fue un 7.38% de error

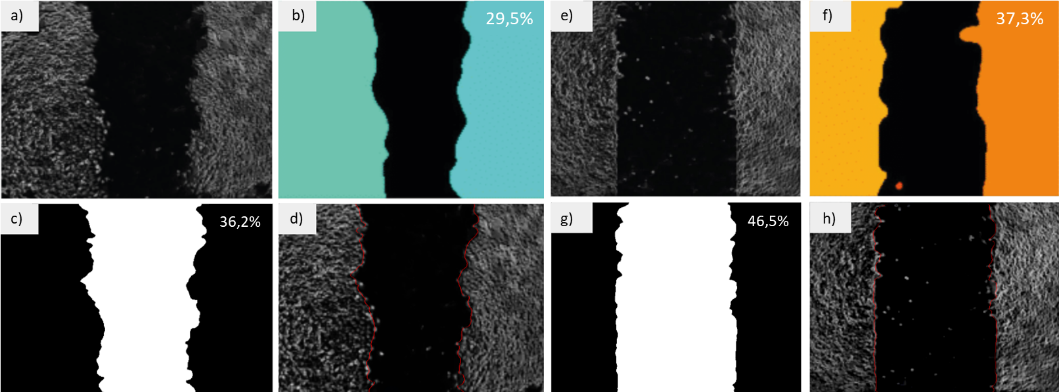


Figura 5: En esta figura se pueden ver las imágenes originales del ensayo de herida (a,e) y el análisis ya publicado obtenido con ImageJ (b,f), así como las máscaras de capa (c,g) y las imágenes con el borde de la herida marcado obtenidas con el software desarrollado (d,h).

La principal limitación que tiene el programa es la escasez de imágenes utilizadas para entrenar al modelo de U-Net, ya que es difícil encontrar un gran repertorio de imágenes de migración celular en el que se proporcionen la imagen original, la máscara en blanco y negro y el análisis. Además, el aspecto de las imágenes de ensayos de herida dependerá de muchos factores, como el tipo de célula, el aumento del microscopio o la luz al tomar la imagen. Debido a esto, las imágenes empleadas en el banco de datos no son lo suficientemente diversas para cubrir todos los casos posibles, por lo que, a pesar de que debido a las características de la red de U-Net el modelo sea bastante preciso, podría mejorarse aumentando la cantidad de imágenes empleadas. Otra posible mejora sería el añadido de un botón para rotar la imagen antes de analizarla. El módulo de detección de bordes programado analiza filas horizontales de píxeles, por lo cual únicamente es capaz de analizar aquellas imágenes que presenten la brecha de forma vertical. A pesar de que rotar la imagen en un editor de imagen es muy sencillo y apenas consume tiempo, para crear un programa completamente funcional por sí mismo sería importante crear este añadido.

**6. Conclusiones**

Mediante el uso de LabVIEW y U-Net se ha conseguido desarrollar un prototipo de software que es capaz de analizar las imágenes de ensayos de herida de forma más rápida, sencilla, reproducible y automática que los programas tradicionales. Es necesario continuar con el desarrollo de este prototipo, con el fin de optimizarlo y mejorar su precisión.

**7. Bibliografía**

Se podrá acceder a la bibliografía escaneando este código QR:

