Erythrocyte classification in sickle cell disease

Facundo Di Yelsi[1], Iván Bumaschny[1], Juan Marí[1]

[1]*Facultad de Ingeniería y Ciencias Exactas y Naturales, Universidad Favaloro, Buenos Aires, Argentina.*

***Abstract*— Taking peripheral blood smear images from patients suffering from sickle cell disease, the aim of this project was to classify the erythrocytes according to their shape. To do so, the first step was to apply image segmentation and noise reduction techniques, while a second stage included cell classification and performance comparison among different classifiers.**

***Keywords*— Erythrocyte, machine learning, processing, classification.**

***Resumen*— A partir de imágenes de frotis de sangre periférica de pacientes con anemia falciforme, el presente trabajo tiene como objetivo clasificar los eritrocitos según su forma. Para ello, se aplicó una primera etapa de segmentación y eliminación de ruido, y una segunda de clasificación, en la cual se compara la performance de distintos clasificadores utilizados.**

***Palabras clave*— Eritrocito, aprendizaje de máquina, procesamiento, clasificación.**

I. INTRODUCCIÓN

La anemia falciforme es un enfermedad que se caracteriza por la presencia de eritrocitos deformados, generalmente con forma elongada, y cuya funcionalidad fisiológica está altamente disminuida.

Se han presentado distintos trabajos al respecto, como [1], en los cuales se toman imágenes de frotis de sangre de estos pacientes y se aplican distintos algoritmos, con la finalidad de clasificar los glóbulos rojos.

Si bien los resultados obtenidos son buenos, un gran defecto es que no se tienen en cuenta las células pegadas, sino que se descartan, lo que lleva a pérdida de información.

Por otra parte, [2] presenta una solución a esto basada en el uso de la transformada watershed, pero aplicado a la clasificación de leucocitos.

Por lo tanto, el presente trabajo tiene como finalidad tomar los mejores aspectos de los trabajos mencionados y aplicarlos al problema de clasificar los glóbulos rojos.

Esto podría ser de gran utilidad en la determinación de la gravedad de la enfermedad en un determinado paciente, según qué proporción de sus eritrocitos se vean afectados.

II. MATERIALES Y MÉTODOS

El trabajo fue realizado en MATLAB, ya que cuenta con gran cantidad de funciones implementadas, y se dividió, a grandes rasgos, en dos partes: segmentación y clasificación.

El objetivo de la primera etapa fue la identificación de las células, eliminando aquellas que no fueran eritrocitos y el ruido propio de la imagen, y se separaron las estaban pegadas mediante el uso de la transformada watershed. Luego, se extrajeron descriptores de todas las células.

En la segunda parte, se tomaron estos descriptores y se aplicaron distintos algoritmos de clasificación, comparando por último la performance de los distintos clasificadores.

En la Fig. 1 se observa un diagrama de flujo con los pasos generales seguidos.

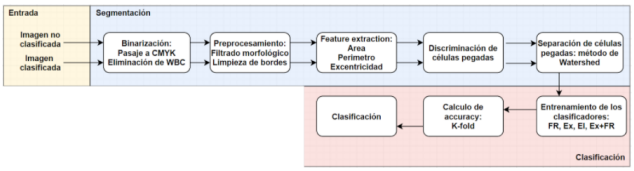


Fig. 1: Diagrama de flujo del trabajo.

A. *Dataset*

El dataset utilizado fue descargado de internet [3], y se compone de 624 imágenes de frotis de sangre periférica, pertenecientes a paciente con anemia falciforme, y cuyas células fueron clasificadas en circulares, elongadas u otras, además de 196 imágenes no clasificadas, como se muestra en la Fig. 2 y Fig. 3.

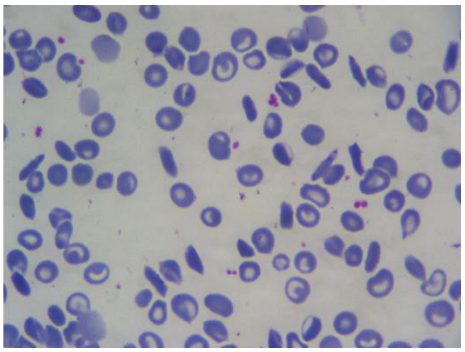


Fig. 2: Imagen original de frotis sanguíneo, sin procesamiento, de las células no clasificadas.

B. *Binarización y eliminación de WBC*

El primer paso fue la eliminación de los leucocitos, para evitar que se solaparan con los eritrocitos de interés.
Analizando la imagen original en los cuatro canales correspondientes al sistema CMYK, se detectó que el mayor contraste entre los glóbulos blancos y los rojos se presentaba en el canal magenta.

Fig. 3: Imagen de las células ya clasificadas, que se usaron para entrenar los clasificadores.

Así, utilizando un umbral empírico de 0.85 determinado a partir del análisis del histograma de colores, se binarizó la imagen de este canal, reteniendo así solamente los leucocitos.

Esta imagen se utilizó para eliminar los glóbulos blancos de la imagen no binarizada, obteniendo así, luego de la aplicación de distintos filtros morfológicos, una imagen en escala de grises.

C. Preprocesamiento y feature-extraction

A partir de la imagen obtenida, se la binarizó nuevamente, utilizando ahora el método de Otsu, y se la segmentó utilizando connected-component labeling. Se eliminaron todos los segmentos con áreas muy pequeñas, que corresponden a ruido o a elementos formes que no son de interés, tales como plaquetas.

Se eliminaron las células pegadas a los bordes, ya que muy probablemente estuvieran cortadas por la imagen y no fueran realmente representativas (Fig. 4).

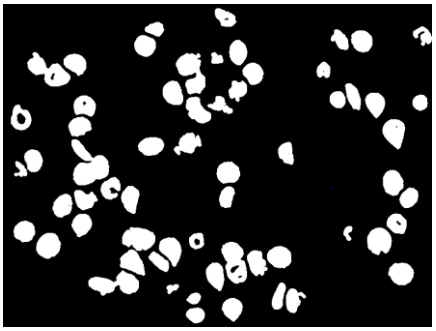


Fig. 4: Imagen obtenida luego de la binarización, antes de la clausura morfológica de las células. Se observan algunas células pegadas.

A las células restantes se les construyeron bounding box individuales, para aplicarles luego, a cada una por separado, una clausura morfológica, eliminando los huecos internos, y se les extrajeron descriptores, que se usarían luego para la clasificación.

Estos descriptores fueron área y perímetro, con los que se calcula el factor de redondez, y la excentricidad, que se detallan más adelante.

D. Separación de células pegadas

Analizando el histograma de áreas, se ajustó una distribución normal $N(\mu, \sigma)$, y se tomaron como válidas las células cuya área estuviera en el intervalo $\mu \pm 1.5\sigma$; las más pequeñas se eliminaron, mientras que las más grandes se tomaron como si fueran dos o más células pegadas y se las pasó por una etapa extra de separación, luego de la cual se volvieron a extraer los descriptores ya mencionados.

En la Fig. 5, es posible apreciar un ejemplo de tres células pegadas, superpuestas entre sí. Debido a que las superficies solapadas son relativamente pequeñas en relación al área total, fue posible separarlas, y poder así sumarlas a los datos disponibles.

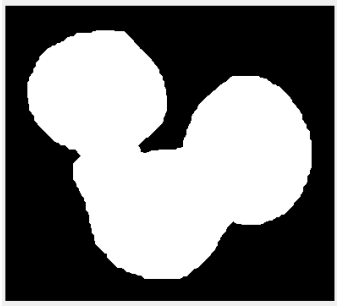


Fig. 5: Tres células pegadas, en su máscara binaria.

Resulta imperativo hacer este procesado, ya que es muy recurrente la superposición y/o unión por error de procesado de las células; si se fuera a descartar todas ellas, sin intentar separarlas, se perdería un buen porcentaje del total de células analizables.

Debido a la necesidad de enmarcar los límites entre las superficies de las células de manera tal que sean divididas de la mejor manera posible, conservando su forma original, se decidió utilizar la transformada *watershed*. Esto se debe a que otros métodos de separación de elementos de manera morfológica dejan de ser de utilidad cuando las figuras comparten una considerable superficie; mas aún, al ser células de diferentes formas geométricas, la elección de diferentes elementos estructurales condiciona el resultado.

Por otra parte, el método de *watershed* utiliza un enfoque alternativo: en vez de trabajar con las formas geométricas en sí, este método procesa la topología original de las figuras para crear un gradiente de magnitudes, el cual estará (en el mejor de los casos) relacionado con las diferentes *alturas* de la figura original. Obviamente, al ser imágenes en dos dimensiones, aquí no se trabaja con gradientes de alturas, sino que, en cambio, las líneas divisoras estarán originadas por las secciones de la imagen enmarcadas por cada célula.

Para lograr esto, es necesario especificar cuales son dichas secciones, para lo cual se utilizó el Algoritmo de Inundado de Meyer. Este algoritmo, específicamente concebido para la transformada *watershed*, consiste en "inundar" la imagen desde ciertos puntos (en el caso particular de este proyecto, se utilizaron los centroides de las células) y, cuando los flujos se encuentran, crear las líneas divisoras ahí mismo (Fig. 6).

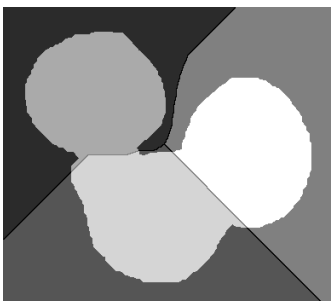


Fig. 6: *Ridge Lines* (líneas divisoras) resultantes, aplicando la transformada *watershed*, en conjunto con el algoritmo de inundado de Meyer. Estas líneas indican dónde se realizará el corte para separar las células.

Es importante destacar varias cosas. Dado que el resultado de la división por el algoritmo de Meyer es dependiente de la semilla (en este caso, el centroide de las células), una buena elección de las mismas llevará a un buen resultado a la hora de aplicar la transformada *watershed*.

En el mejor de los casos, el centroide correspondiente a cada célula es fácil de encontrar con operaciones morfológicas como la erosión (Fig. 7); en cambio, cuanto más superpuestas se encuentren las mismas, más difícil será hallar dicho centroide.

Fig. 7: El centroide de cada célula de la Fig. 5.

En algunos casos, resulta prácticamente imposible encontrar un solo punto central usando solamente erosión, ya que el elemento resultante más pequeño será, en el mejor de los casos, del tamaño del elemento estructurante.

En estas situaciones, existen métodos alternativos, como son la transformada circular de Hough o incluso métodos estadísticos como vecinos más cercanos.

Sin embargo, en este proyecto se utilizó, sumada de la erosión, la transformada de distancia euclídea. Esta operación consiste en guardar, en cada píxel de la imagen, el valor correspondiente a la distancia mínima entre el píxel en cuestión y el píxel igual a cero más cercano (Fig. 8).

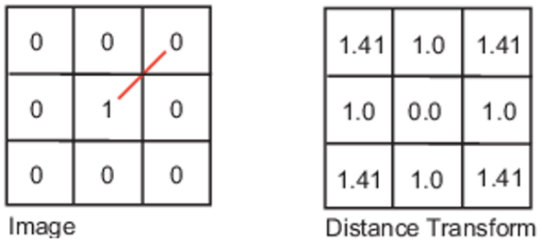


Fig. 8: Matriz de 3x3 resultante, aplicando la transformada de distancia euclídea. Fuente: Documentación de Matlab de *Distance Transform*

Debido a las geometrías circular y cilíndrica de las células, el punto con el valor más alto corresponderá, en general, al centro de la imagen (Fig. 9).

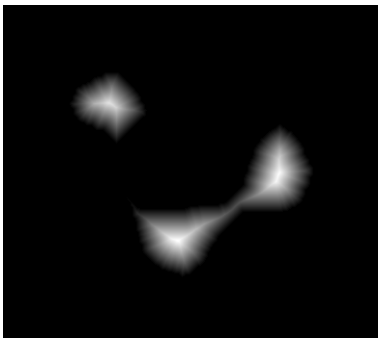


Fig. 9: Imagen resultante al aplicar erosiones consecutivas y la transformada de distancia, en ese orden, a la Fig. 5.

E. Clasificación

Como se comentó anteriormente, para cada célula se calculó, usando su área y perímetro, el factor de redondez, dado por la ecuación 1.

$$FR = 4\pi \frac{Area}{Perímetro^2} \quad (1)$$

Un valor de 1 se corresponde con un círculo perfecto, mientras que uno menor a 1 se va asemejando más a una elipse.

Por otra parte, se calculó la excentricidad, dada por la ecuación 2.

$$Ex = \sqrt{1 - (\frac{Eje\ menor}{Eje\ mayor})^2} \quad (2)$$

En un círculo perfecto, ambos ejes son iguales, y Ex tiende a 0. En cambio, cuanto más elongada es la figura, Ex tiende a 1.

Con estos dos descriptores, se define un tercero, la elipticidad El, dado por la ecuación 3.

$$El = \frac{FR}{Ex} \quad (3)$$

Al momento de decidir qué clasificador utilizar, el primer paso fue analizar la distribución de las tres variables en las tres clases (circular, elongada u otra), ya que algunos asumen normalidad de los datos.

Como se vio analizando los histogramas, y como se confirmó con un test de Shapiro-Wilk, la mayor parte de los datos no estaban normalmente distribuidos, por lo que se descartó el uso de LDA y QDA, y se eligieron KNN, árboles de decisión y el clasificador de Naive-Bayes.

Tomando los descriptores extraídos de las células ya clasificadas, como las de la Fig. 3, se dividieron en un set de train y uno de test, en relación 90/10.

Para distintas combinaciones de descriptores, se utilizó el set de train para entrenar a los clasificadores. Como métrica de comparación, se eligió el accuracy, calculado según la ecuación 4.

$$Accuracy = \frac{Cantidad\ bien\ clasificados}{Cantidad\ total} \quad (4)$$

Claramente, cuanto mayor sea el *accuracy*, mejor será el clasificador.

Mediante K-fold CV (K=5), se calculó el valor de esta métrica para cada uno de los clasificadores entrenados.

Por último, se evaluaron los mismos sobre los sets de test correspondientes en cada caso, y se recalculó el *accuracy*. La idea de esto es comparar los valores obtenidos para ambos sets, para determinar así si el clasificador tiene overfitting o si, en cambio, generaliza bien.

Con el objetivo de aumentar aún más el valor de *accuracy* obtenido, se armó un ensamble con los tres clasificadores, que consistió en elegir como clasificación de cada célula del set de test a la clase más votada por los tres.

III. Resultados

Las tablas I a IV muestran los resultados de *accuracy* obtenidos con los tres clasificadores ya mencionados y el ensamble, utilizando distintas combinaciones de los descriptores.

TABLA I
RESULTADOS DE KNN

Descriptor	Accuracy K-fold	Accuracy test
Factor de redondez	87.1%	88.9%
Excentricidad	85.0%	88.5%
Elipticidad	84.3%	83.1%
FR+Ex	92.9%	93.9%

Resultados de *accuracy* obtenidos con el clasificador KNN, tanto sobre el set de train (con K-fold) como sobre el de test.

TABLA II
RESULTADOS DE ÁRBOLES DE DECISIÓN

Descriptor	Accuracy K-fold	Accuracy test
Factor de redondez	88.0%	89.0%
Excentricidad	86.3%	90.0%
Elipticidad	86.0%	85.7%
FR+Ex	95.0%	94.4%

Resultados de *accuracy* obtenidos usando árboles de decisión, tanto sobre el set de train (con K-fold) como sobre el de test.

TABLA III
RESULTADOS DE NAIVE-BAYES

Descriptor	Accuracy K-fold	Accuracy test
Factor de redondez	91.8%	89.0%
Excentricidad	87.8%	87.1%
Elipticidad	89.1%	84.3%
FR+Ex	94.8%	95.0%

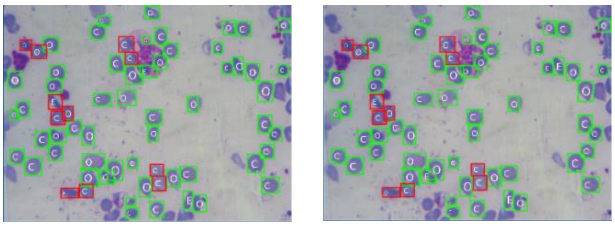
Resultados de *accuracy* obtenidos con el clasificador de Naive-Bayes, tanto sobre el set de train (con K-fold) como sobre el de test.

TABLA IV
RESULTADOS DEL ENSAMBLE DE CLASIFICADORES

Descriptor	Accuracy test
Factor de redondez	90.3%
Excentricidad	90.3%
Elipticidad	85.5%
FR+Ex	95.1%

Resultados de *accuracy* obtenidos con el ensamble de clasificadores, tanto sobre el set de train (con K-fold) como sobre el de test.

La Fig. 10 muestra la clasificación obtenida utilizando el ensamble de clasificadores.



(a) (b)

Fig. 10: Clasificación final. En verde, las células naturalmente separadas. En rojo, las separadas con transformada watershed. (a) Utilizando solo el factor de redondez. (b) Utilizando FR+Ex.

IV. CONCLUSIONES

La principal innovación de este trabajo con respecto a otros anteriores es la separación de las células pegadas en lugar de ser descartadas, lo que logra reducir la pérdida de información.

Como se observa en las tablas de resultados, los mejores valores de *accuracy*, tanto sobre el set de train como sobre el de test, se obtuvieron utilizando la combinación de factor de redondez y excentricidad.

Esto se observa también en la Fig. 10; hay algunas células que, a simple viste, son claramente elongadas, y que son mal clasificadas utilizando solamente FR, mientras que son bien etiquetadas al usar FR+Ex.

Por otra parte, el mejor clasificador fue el ensamble, ya que se ve una mejora notable en su *accuracy* con respecto a los tres individuales.

Como posible continuación del trabajo a futuro, podría pensarse en adaptar algún modelo de *deep learning* para tratar de obtener mejores resultados, aunque esto requeriría de un dataset de mucho mayor tamaño.

También podría ser de utilidad analizar otras métricas al momento de comparar el desempeño de los clasificadores, ya que cada una resalta distintos aspectos de los mismos.

REFERENCIAS

[1] Tomari, Razali & Wan Zakaria, W.N & Abdul Jamil, Muhammad Mahadi & Nor, Faridah & Fuad, Nik. (2014). Computer Aided System for Red Blood Cell Classification in Blood Smear Image. Procedia Computer Science. 42. 10.1016/j.procs.2014.11.053.

[2] Di Ruberto, Cecilia & Puztu, Lorenzo. (2013). White Blood Cells Identification and Counting from Microscopic Blood Image. International Journal of Medical, Health, Biomedical and Pharmaceutical Engineering. 7. 15-22.

[3] Gonzalez-Hidalgo, M.; Guerrero-Pena, F.A.; Herold-Garcia, S.; Jaume-i-Capó, A.; Marrero-Fernández, P.D. Red blood cell cluster separation from digital images for use in sickle cell disease. IEEE J. Biomed. Health Inform. 2014, 19, 1514–1525