UNIVERSITAT JAUME I

Trabajo de fin de Máster

Máster en Sistemas Inteligentes
Curso 2016 - 2017

Caracterización y clasificación de glóbulos blancos mediante descriptores locales de imágenes

---

Autor:
Dan López Puigdollers

Tutores académicos:
Vicente Javier Traver Roig
Filiberto Pla Bañón

Fecha de lectura: 25 de septiembre de 2017

Si es bueno vivir, todavía es mejor soñar, y lo mejor de todo, despertar.
Antonio Machado (1875-1939)

Agradecimientos

Quería dedicar este pequeño apartado para acordarme y agradecer a todas aquellas personas que habéis estado cerca en mi travesía académica en estos últimos años y que, de alguna forma u otra, me habéis ayudado a no rendirme y a forjar un destino del cual no hubiese sido capaz de alcanzar por mis propias fuerzas.

En primer lugar, gracias a Dios por permitirme llegar a este punto de mi vida y por acompañarme a lo largo de todos estos años en la universidad.

En segundo lugar, quería agradecer a mis padres su inefable e incondicional apoyo en todo este tiempo. Gracias a mi madre por su paciente oído detrás del teléfono durante mis años de carrera y por sus palabras de ánimo en los momentos difíciles. Gracias a ello pude llegar bien a la meta. Gracias también a mi padre por aconsejarme y decirme las palabras que necesito escuchar en todo momento.

Tampoco quería olvidarme de mi hermana María. De nuevo, gracias por inspirarme con tu ejemplo de esfuerzo y excelencia. Sin ti no sería la persona que soy a día de hoy.

Muchas gracias también a mis tutores Vicente Javier y Filiberto. Por haberme brindado vuestra valiosa ayuda en la labor investigadora y en la elaboración de este trabajo y por concederme formar parte como becario de este línea de investigación. Gracias también al Plan de Promoción de la Investigación de la UJI por la financiación de este trabajo en el proyecto P11B2014-09. Sin vosotros este trabajo no hubiese sido posible.

A todos vosotros mi más sincera gratitud.

Resumen

La pretensión de este trabajo es ofrecer un estudio de análisis, exploración y comparación de una serie de extractores de características o puntos de interés locales de diferentes tipos, con el fin de formar parte de la descripción de vectores de atributos requeridos en un esquema de clasificación posterior denominado Bolsa de Palabras Visuales (*Visual* BoW). Se trata de un modelo de aprendizaje empleado en un amplio rango de disciplinas, pero inédito en la competencia que se plantea en este trabajo: clasificar automáticamente imágenes en color relacionadas con ejemplos de glóbulos blancos de distintas clases. Por tanto, comparar varios detectores de características permite conocer el tipo de característica local detectada disponibles que más se adecua a la problemática planteada, consiguiendo mejorar la precisión y la robustez del método de clasificación elegido. Junto a este estudio también se tratan otras fases del proceso de la *Visual* BoW, como son la elección de parámetros en el proceso de formación del "vocabulario" y el algoritmo de aprendizaje supervisado.

Palabras clave

Clasificación de glóbulos blancos, Bolsa de Palabras Visuales, *Visual* BoW, detección de características locales, SIFT.

Abstract

The purpose of this work is to offer a study of the analysis, exploration and comparison of a series of extractors of local features or points of interest of different types, in order to be part of the description of required attribute vectors in a classification scheme later denominated Bag of Visual Words (Visual BoW). It is a model of learning used in a wide range of disciplines, but unexplored for this problem that arises this work: automatic classification of color images with examples of white blood cells of different classes. Therefore, several feature detectors allow us to know the type of detected feature available that best fits the problematic raised, improving the accuracy and robustness of the chosen classification method. In addition to this study, other phases of the *Visual* BoW process are also discussed, such as the choice of parameters in the process of "vocabulary" formation and the supervised learning algorithm.

Keywords

Classification of white blood cells, Bag of Visual Words, Visual BoW, detection of local features, SIFT.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Contextualización

Los avances en análisis y diagnóstico asistido por imagen han sido notorios en los últimos años, a razón del nivel actual de desarrollo tecnológico de los sistemas que intervienen en dichos procesos. Para el recuento automático de células sanguíneas es habitual el empleo de la citometría de flujo, el cual consiste en identificar y clasificar las células mediante el uso de tecnología láser gracias a la explotación de las características morfológicas, empleo de biomarcadores o ingeniería de proteínas [21].

A pesar de obtenerse resultados precisos, dicha técnica presenta un conjunto de limitaciones a la hora de procesar una amplia variedad de subclases de leucocitos y, especialmente, las anómalas, que suscitan un especial interés en el diagnóstico de una amplia categoría de enfermedades. Hasta el momento, debido a la menor frecuencia en la presencia de este tipo de células, era necesaria la intervención manual de un especialista mediante análisis visual directo de las muestras.

Actualmente existen diversas propuestas sofisticadas y precisas de clasificadores automáticos basados en análisis de imagen para un amplio rango de aplicaciones médicas [36]. Los modelos orientados a células sanguíneas permiten lidiar con varios tipos de glóbulos blancos, aunque su rendimiento en la clasificación o el recuento diferencial es insuficiente para la práctica clínica, puesto que emplean métodos tradicionales de segmentación, caracterización y clasificación, habitualmente rígidos, que no permiten redefinir y evolucionar a través de la retroalimentación recibida por los hematólogos de manera constante.

Para resolver estas limitaciones se propone el uso de un marco de técnicas de aprendizaje automático capaces de trabajar con múltiples clases de glóbulos blancos, con especial interés en aquellas esenciales y menos presentes en el diagnóstico, como son las células anómalas, a través de la extracción de características de las imágenes y su posterior clasificación por medio de técnicas ajustadas al problema dado.

1.2. Aprendizaje automático en clasificación de imágenes

La intención del proceso de clasificación en este aspecto consiste en categorizar una imagen digital dentro de una de las distintas clases disponibles. Para ello, el objetivo pasa por identificar y representar la diversidad de características que ocurren y definen una imagen, tales como la disposición espacial y naturaleza de los píxeles, las relaciones que se establecen entre ellos, etc. Estas características o atributos pueden codificarse de manera numérica, por ejemplo, nivel de intensidad, color o propiedades geométricas; o de forma categórica, es decir, propiedades que se expresan textualmente.

La clasificación de imágenes es una parte importante del análisis de imágenes digitales. En la literatura predominan dos métodos principales de aprendizaje automático enfocados en este aspecto: aprendizaje supervisado y no supervisado.

1.2.1. Aprendizaje supervisado

En el aprendizaje supervisado se identifican y etiquetan previamente ejemplos de las clases que conforman el problema de clasificación. Estos ejemplos se proporcionan al algoritmo de aprendizaje como datos de entrada, también denominados como "conjuntos de entrenamiento", en el proceso de creación del modelo predictivo. Una vez formulado el modelo de esta manera, éste es capaz de predecir la etiqueta de ejemplos no observados. Es el escenario más común asociado con clasificación, regresión y problemas de *ranking* [28].

Formalmente, dado un conjunto N de ejemplos de entrenamiento de la forma $\{(x_1, y_1), \ldots, (x_N, y_N)\}$, tal que x_i es un vector de características del ejemplo i-ésimo e y_i, su respectiva etiqueta o clase, un algoritmo de clasificación busca la función $g : X \to Y$, donde X es el espacio de entrada e Y es el espacio de salida. La función g es un elemento de algún espacio de las posibles funciones de G, habitualmente llamado *espacio hipotético*. A veces es conveniente representar g usando una función de puntuación $f : X \times Y \to \Re$, tal que la función g quede definida de forma que devuelva y con la máxima puntuación: $g(x) = \arg\max_y f(x, y)$.

Existen muchos algoritmos de aprendizaje que pretenden encontrar la función g, por ejemplo, formulando el modelo como un problema probabilístico $g(x) = P(y|x)$, o f tomando forma de modelo de probabilidad conjunta $f(x, y) = P(x, y)$. Por ejemplo, *naive Bayes* y *análisis de discriminantes lineales* (*Linear Discriminant Analysis*, LDA) son modelos de probabilidad conjunta, mientras que la *regresión logística* es un modelo de probabilidad condicional.

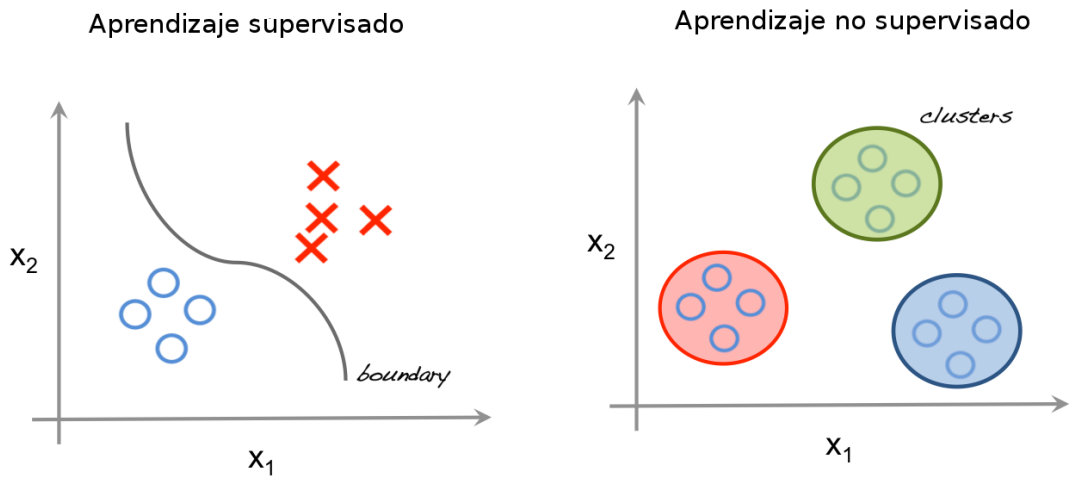


Figura 1.1: Comparación entre aprendizaje supervisado y no supervisado. Imagen de: http://beta.cambridgespark.com

1.2.2. Aprendizaje no supervisado

Por otro lado, el aprendizaje no supervisado se parte de un conjunto de imágenes que no han sido previamente etiquetadas. La idea principal es emplear algoritmos que examinen un conjunto de atributos desconocidos y que consigan dividir en un número determinado de grupos naturales presentes y similares a los que definen las clases de las imágenes analizadas a través de una medida de similitud.

A diferencia de la clasificación supervisada, la clasificación no supervisada no requiere datos de entrenamiento especificados por un analista. La premisa básica es que los valores dentro de un tipo de clase dado deben estar cerca juntos en el espacio de medición (por ejemplo, tener niveles de gris similares), mientras que los datos en diferentes clases deben estar comparativamente bien separados (por ejemplo, tener niveles de grises muy diferentes).

Debido a que no existen ejemplos etiquetados, puede resultar más complicado evaluar cuantitativamente el rendimiento del modelo de aprendizaje. Métodos de *clustering* y de reducción de la dimensionalidad son ejemplos de problemas de aprendizaje no supervisado [28].

La comparación entre ambos métodos es observable en la Figura 1.1. A la izquierda se disponen de muestras etiquetadas representadas por distintas formas geométricas. El algoritmo de aprendizaje se encarga de modelar la frontera de decisión entre las clases a partir de estas muestras. A la derecha de la figura se parte de ejemplos no etiquetados. Los algoritmos tratan de buscar las agrupaciones naturales de estos ejemplos por medio de medidas que aseguren la máxima similitud *intraclase* y disimilitud *interclase*.

1.3. Clasificación de glóbulos blancos

En hematimetría clínica resulta de vital importancia hacer una identificación y cuantificación precisa de los diferentes conjuntos de grupos celulares, glóbulos rojos (hematíes), glóbulos blancos (leucocitos), plaquetas, contenido de hemoglobina y otros parámetros asociados con la cantidad, forma y contenido. El fin es adquirir datos de salud relacionados con la presencia de posibles enfermedades: anemia, enfermedades generales o diferentes tipos de cáncer.

Respecto al tópico de estudio que se centra este trabajo, los glóbulos blancos son los encargados de las defensas de la persona. Por ello, en posibles cuadros de infección su cantidad es más elevada, o en ciertas enfermedades están disminuidos. También es importante saber cuáles son las poblaciones de cada tipo de leucocitos. Desde un primer acercamiento a la problemática dada podemos hacer una previa distinción taxonómica de los glóbulos blancos en dos grupos: granulocitos y agranulocitos [35].

1.3.1. Granulocitos

Son el tipo más común de glóbulos blancos en el cuerpo humano, con una proporción alrededor del 70-75 % del total de glóbulos blancos. La razón del nombre de este tipo de células es por el contenido de pequeños y visibles gránulos dentro del citoplasma, claramente observables bajo el efecto de coloración mediante tintes [8]. Los granulocitos se pueden subdividir en: neutrófilos, basófilos y eosinófilos.

Neutrófilos

Los neutrófilos son el tipo más abundante de granulocitos, aproximadamente 40-75 % del total de glóbulos blancos. Forman una parte esencial del sistema inmune innato.

Se forman a partir de células madre en la médula ósea. Los neutrófilos pueden subdividirse en neutrófilos segmentados y neutrófilos unidos.

Tienen un tamaño aproximado de 10-12 μm, con un núcleo multilobulado. Sus gránulos son finos y ligeramente rosados en presencia de tinción.

Basófilos

Los basófilos son los menos comunes dentro de los granulocitos, representando alrededor de 0.5-1 % del total de glóbulos blancos. Sin embargo, son el tipo más grande de granulocitos.

Tienen un tamaño aproximado de 12-15 μm, con un núcleo bilobulado o trilobulado. Sus gránulos presentan un color azul oscuro en presencia de tinción.

Eosinófilos

Los eosinófilos componen aproximadamente 2-4 % del total de glóbulos blancos. Se ocupan principalmente de las infecciones parasitarias.

Su núcleo es, habitualmente, bilobulado con un tamaño aproximado de 10-12 μm. Los lóbulos están conectados por un cordón delgado. El citoplasma está lleno de gránulos que asumen un color rosa-naranja característico con tinción de eosina.

1.3.2. Agranulocitos

Por otro lado, a diferencia de los granulocitos, los agranulocitos se caracterizan por no presentar gránulos en su citoplasma. Tampoco disponen de una cobertura de membrana, propia de los granulocitos. Los agranulocitos pueden clasificarse en linfocitos y monocitos.

Linfocitos

Los linfocitos son más comunes en el sistema linfático que en el flujo sanguíneo. Se pueden clasificar en tres tipos: células NK, células T y células B.

Puede presentar un tamaño variable que oscila en dos rangos. Por un lado, los linfocitos pequeños pueden tener un tamaño aproximado de 7-8 μm. Los linfocitos grandes pueden tener un tamaño que se mueve en el rango de 12-15 μm. La forma de su núcleo es excéntrica y presenta un color intenso, producto de la tinción.

Monocitos

Los monocitos son un tipo de glóbulo blanco que se encuentra aproximadamente en 5.3 % del total de glóbulos blancos. Generalmente, abandonan el flujo sanguíneo y se convierten en macrófagos de tejidos, encargándose de eliminar restos de células muertas, así como del ataque de microorganismos.

Presentan un tamaño dentro del rango aproximado de 12-15 μm. La forma de su núcleo es ligeramente arriñonada y, bajo el efecto de la tinción, presenta un ligero color rosado.

En la Figura 1.2 se pueden observar cinco instancias o imágenes de los tipos

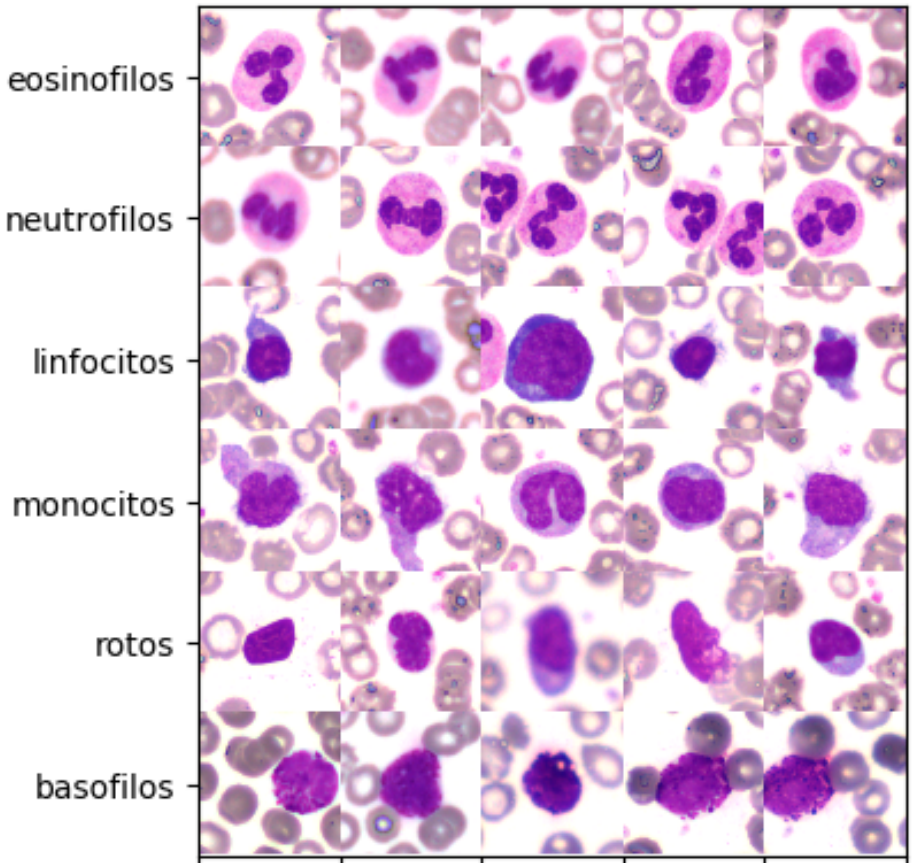


Figura 1.2: Ejemplos de 5 imágenes de glóbulos blancos para cada una de las distintas clases observadas en nuestra base de datos.

de glóbulos blancos comentados anteriormente y disponibles en la base de datos proporcionada para este trabajo. Junto a las clases anteriores se añade una nueva, denominada "rotos", que hace referencia a cualquier tipo de glóbulo blanco sobre el que se ha desprendido su citoplasma y el contenido de éste en el proceso de adquisición de las imágenes. La naturaleza de la base de datos se detalla en más profundidad en la sección 3.2.

1.4. Estado del arte

Los procedimientos de segmentación manuales, pese a la precisión que se puede garantizar gracias a la clasificación de las muestras por interacción de un experto médico, son incapaces de evolucionar y aprender de esta retroalimentación recibida, además de ser temporalmente muy costosos para el personal médico al tener que realizar un riguroso estudio de los parámetros determinantes en la diferenciación de cada tipo de leucocito en particular. En la literatura podemos encontrar distintos enfoques que intentan crear modelos orientados a la clasificación automática de glóbulos blancos basados en imágenes, puesto que son más rápidos y menos laboriosos que los métodos tradicionales de clasificación de los diferentes tipos de células sanguíneas de forma manual y sistemática. Además, en un sistema de clasificación

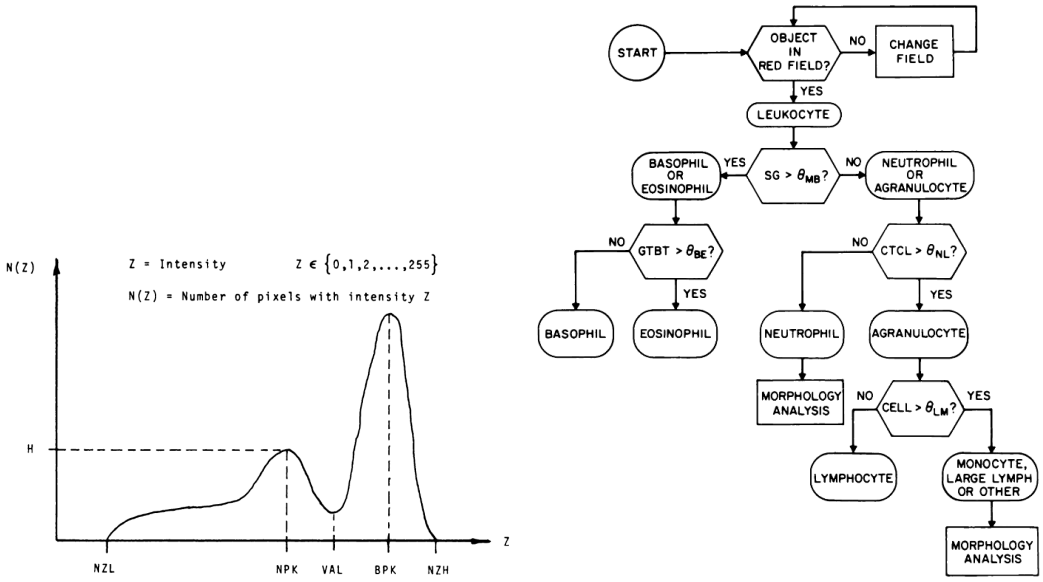
automática adaptativo, el hematólogo puede ser capaz de adaptar las decisiones automáticas del sistema, a la vez que se detecta una muestra errónea. Mediante estos enfoques también es posible reducir la rigidez que supone preprocesar y segmentar las imágenes de forma manual. En contrapartida, estos sistemas requieren de conjuntos de entrenamiento relativamente grandes para generar el modelo de clasificación y poder competir en precisión respecto a los métodos más tradicionales.

Por regla general, existen dos protocolos empleados en el conteo de células sanguíneas usados en la diagnosis clínica empleado por los expertos. Uno es el conteo completo de sangre (*Complete Blood Count* , CBC), también conocido como hemograma, y otro es el conteo diferencial de sangre (*Differential Blood Count* , DBC).

En el protocolo CBC se realiza automáticamente el conteo usando un citómetro. En el caso de DBC, se trata de un método muy empleado en la diagnosis de enfermedades relacionadas con la sangre. Consiste en calcular el porcentaje de ocurrencias de todos los tipos de células sanguíneas en las imágenes con las células marcadas, también denominados especímenes, por medio de la coloración de su citoplasma mediante algún tipo de tinción, también denominado tinción de Romanowsky (más tarde aparecieron otro tipo de tinciones que se emplean en tipos de glóbulos blancos concretos, como la tinción *Giemsa*, *Jenner*, *Wright*, *Field* o *Leishman*), encontrando y clasificando una muestra con 100 leucocitos con ayuda del microscopio [27]. La calidad del DBC es dependiente de la experiencia del experto. Además, se trata de en un proceso más complejo y con mayor coste temporal por su prevalencia manual.

Los primeros métodos automáticos basados en DBC realizaban la clasificación teniendo en cuenta criterios basados en la textura y forma del núcleo, presencia de gránulos y color del citoplasma del glóbulo coloreado con el tinte. La propuesta presentada en este artículo [41] sugiere un método de clasificación basado en el diseño un algoritmo procedimental de decisión secuencial. Facilita el análisis de especímenes afectados por distintos tipos de tinción de Romanowsky y permite orientar al experto hematológico en la misión de categorizar nuevas muestras a través de la inspección de éstas mediante la visualización y cálculo de cinco patrones de atributos de las imágenes digitales extraídas por el microscopio. Estos atributos se extraen a partir de los histogramas de intensidad en tres bandas de longitud de onda (rojo, azul y verde) bajo tres condiciones de iluminación. Para cada histograma se analizan los parámetros NZL y NZH, relacionados con la ubicación del valor de inicio y fin de cada histograma; H y NPK, referidos como la altura y ubicación del pico relacionado con la célula; BPK, posición del pico secundario relacionado con el fondo de la imagen; y VAL, posición del umbral que separa la información vinculada a la célula y la información relacionada con el fondo (Figura 1.3a). A partir de éstos se extraen una serie de parámetros estadísticos que permiten definir visualmente las fronteras de decisión entre los 5 tipos de glóbulos blancos. El diagrama de flujo resultante del análisis de los parámetros mencionados con un conjunto de entrenamiento dado se resume en la Figura 1.3b.

Más tarde se mejoró de forma considerable el rendimiento del proceso, gracias a la aparición del método de resistencia eléctrica. Las células sanguíneas tienen la propiedad de no conducir la electricidad. El cambio y la magnitud de la resistencia



(a) Esquemático de los parámetros definidos en los histogramas de intensidad.

(b) Diagrama de flujo del algoritmo de decisión secuencial.

Figura 1.3: Método de clasificación procedimental de leucocitos construido a través del análisis de imágenes digitales [41].

detectada en el paso por una minúscula obertura determina el tamaño de las células que pasan a través de la máquina en el interior de un líquido donde se conduce electricidad, pudiendo distinguirlas.

El progreso de esta técnica fue en aumento gracias a la aparición de la citometría por flujo óptico [14], técnica implementada en la mayor parte de los dispositivos comerciales actuales. Está técnica se basa en la dispersión de luz generada en la reflexión de la luz láser en las células afectadas por un químico fluorescente.

Los métodos de clasificación se han realizado de forma paralela, desde expertos hematológicos hasta técnicas basadas en el procesado de las imágenes de glóbulos tintados y técnicas de reconocimiento de patrones [37, 30].

Puesto que ambos métodos se basan en células afectadas por químicos colorantes, es posible combinarlas para obtener mejores resultados. La clasificación basada en imagen puede funcionar como un "consejero" del experto hematológico, o bien, realizar una rutina DBC completamente autónoma, excepto en casos concretos donde es necesaria la intervención de un especialista.

Los enfoques actuales se centran en emplear los algoritmos de aprendizaje con mejor rendimiento y desempeño dentro del mundo del reconocimiento de patrones junto con técnicas de procesamiento digital de imagen. Por ejemplo, las redes neuronales (referidas en la literatura como *Neural Networks*, NN) han sido un frecuente en uso dentro de la clasificación automática de glóbulos blancos. Sin embargo, la mejor precisión ofrecida por éstas da lugar a margen y espacio para futuras mejoras en los algoritmos propuestos.

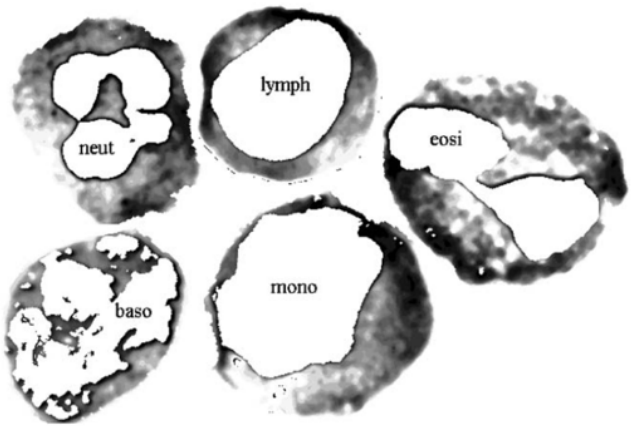


Figura 1.4: Cuantificación en 16 niveles de gris y segmentación del citoplasma en los cinco tipos normales de glóbulos blancos [34].

Por ejemplo, varios autores [14, 30, 18] realizan un análisis de las propiedades morfológicas de los glóbulos blancos de estudio mediante el procesado previo de los niveles de gris o canales RGB de las imágenes, segmentación y umbralizado. De las imágenes binarias resultantes se analiza el área (entendido como la cantidad de píxeles iguales a 1), la solidez (el área dividido por el área de la envolvente convexa que compromete los píxeles del núcleo), circularidad (área del núcleo divida por el cuadrado de la circunferencia del núcleo) o excentricidad.

También se resalta la importancia de determinar y cuantificar la información sobre la textura del núcleo afirmando que el análisis exclusivo de la forma es insuficiente para obtener buenos resultados, ya que el aspecto visual de los leucocitos es similar entre ellos y su variación es amplia. Algunos autores añaden a los atributos anteriores basados en la forma del núcleo, un análisis de la textura del citoplasma a través de la segmentación y transformación del espacio en color a niveles de gris (Figura 1.4). El citoplasma, a diferencia del núcleo, presenta poca variación y amplitud de color. Se realiza una extracción de cinco atributos de textura basados en las matrices de concurrencia en niveles de gris (GLCM), como es la inercia, entropía, energía, homogeneidad local y correlación [34].

Otros autores [15] se centran en analizar la forma de los leucocitos por medio de operaciones de morfología matemática, con especial interés en el espectro de patrones o *pecstrum* (Figura 1.5). Se trata de un operador que descompone una imagen binaria segmentada en componentes morfológicos de acuerdo a la forma y tamaño de un elemento estructurante. Provee un análisis cuantitativo del contenido morfológico de las imágenes para ser usado posteriormente como vectores de atributos de características para un determinado algoritmo de aprendizaje, como puede ser una NN, máquinas de vectores de soporte (*Support Vector Machines*, SVM) , distancia euclídea, k-vecinos más próximos (k-*Nearest Neighbors*, k-NN) o redes neuronales retroalimentadas (*Feedforward Neural Network*, FFNN).

Toda esta información se proporciona como vectores de características al modelo de aprendizaje, en este caso, una NN que puede tener una o varias capas ocultas

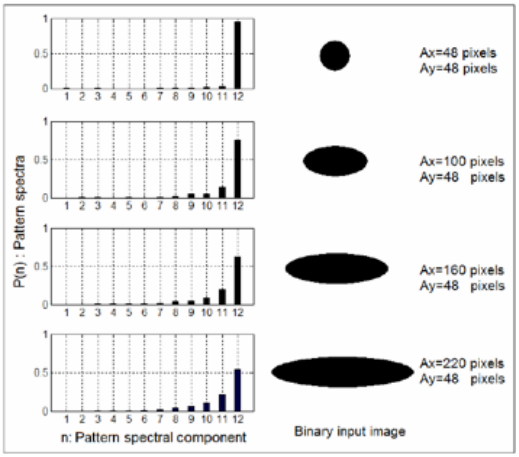


Figura 1.5: Ejemplo de análisis del contenido morfológico mediante el operador morfológico *pecstrum* [15].

dependiendo de la superposición del espacio de entrada, ya que los diferentes tipos de glóbulos blancos pueden ser visualmente similares. Por ejemplo, dentro de los linfocitos podemos encontrar un rango de tamaño más variable, que van desde los 8-10 μm hasta los 15 μm. En este caso, podemos separar dos espacios de entrada diferentes para una misma clase. El tamaño de la red no puede ser excesivo, con el fin de evitar problemas de "sobreajuste".

La tendencia más actual y clara en clasificación de imágenes se centra en el "aprendizaje profundo" o *deep learning*. En 2012 se presentó un modelo de aprendizaje basado en redes neuronales convolucionales profundas (*Convolutional Neural Networks*, CNNs) en el concurso anual de *ILSVRC* (*Large Scale Visual Recognition Challenge*. Más detalles en: http://www.image-net.org/challenges/LSVRC/), empleando un subconjunto de datos de *ImageNet*, una base de datos compuesta por cientos de miles de imágenes dentro de 1000 clases diferentes. Alex Krizhevsky et al. [23] consiguieron arrojar los mejores resultados publicados hasta esa fecha. Desde entonces, la popularidad de este tipo de modelos ha crecido y su aplicabilidad a problemas de visión por ordenador por sus buenos resultados. Además, éstos no requieren de extracción de características previa, puesto que toman las imágenes íntegras como entrada al sistema y las características útiles para el problema se aprenden automáticamente ajustando una enorme cantidad de parámetros en las arquitecturas de múltiples capas empleadas.

Es posible encontrar experimentos realizados en la clasificación de glóbulos blancos, por ejemplo, haciendo uso de una arquitectura de redes convolucionales a través de una estructura *LeNet* con un pequeño problema conformado por pocos ejemplos distribuidos en las cinco posibles clases de glóbulos blancos: eosinófilos, neutrófilos, basófilos, monocitos y linfocitos.

En el primer caso [17], se realiza una división de clases se transforma a un problema binario basado en glóbulos blancos mononucleares (monocitos, linfocitos y basófilos) y polinucleares (neutrófilos y eosinófilos). Los resultados mostraban un rendimiento excelente a pesar de la distribución sesgada de los ejemplos y del tamaño

reducido del conjunto de datos, disponiendo únicamente de 352 ejemplos.

En el segundo caso [29], se tienen también ejemplos de las mismas clases, contando con un conjunto de datos de 115 para las muestras de entrenamiento y 25 para el conjunto de prueba. Además, se disponían de imágenes de baja resolución. Los resultados mostraban que se obtenían resultados mejores o equiparables a clasificadores SVM que emplean extracción de características basadas en histogramas e intensidad. Además, como en otros modelos de aprendizaje automático, el rendimiento de las CNN puede mejorarse a través del tiempo a medida que se aumenta el volumen de datos.

1.5. Motivación y objetivos

El trabajo aquí presentado forma parte del proyecto financiado por el Plan de Promoción de la Investigación de la UJI "Técnicas de aprendizaje adaptativo y extracción de características en imágenes digitales para el reconocimiento automático de células sanguíneas" (P11B2014-09) con vigencia del 01/01/15 al 31/12/17.

La idea principal del presente proyecto parte en estudiar la naturaleza de la base de datos formada por imágenes de glóbulos blancos de distintos tipos, con el fin de diseñar e incorporar técnicas de aprendizaje adaptativo específicas a un sistema de clasificación de glóbulos blancos ajustado a unas singularidades y necesidades concretas: grado de automatización del método, flexibilidad, robustez, precisión, etc. La fase más crítica de este problema que garantiza la consecución global de los objetivos, como se ha podido dilucidar en el actual estado del arte, consiste en enmarcar un conjunto de atributos en las imágenes que describan con exactitud cada una de las clases representadas y sean capaces de delimitar y diferenciarlas sin ambigüedad. Esta selección previa permite optimizar multitud de factores en las fases posteriores del proceso de aprendizaje.

Aunque en un inicio se planteó la idea de apostar por un enfoque basado en *deep learning*, dado a sus buenos resultados y por su omisión de una fase previa de extracción de características, el limitante que nos hizo descartarla fue el tamaño de la base de datos proporcionada. Para obtener un buen rendimiento en soluciones basadas en "aprendizaje profundo", éstas requieren normalmente de "ingentes" cantidades de datos en la fase de aprendizaje [36]. Disponer de una base de datos relativamente pequeña no garantiza que enfoques basados en CNNs, presumiblemente costosos, consigan justificar mejoras significativas en los resultados respecto a otro tipo de esquemas de aprendizaje disponibles y más configurables.

También se contempló la idea de segmentar las imágenes previamente a construir el modelo de clasificación. Este concepto se basa en aplicar un preprocesado en las imágenes para normalizarlas y mitigar el efecto de posibles artefactos para ser procesadas. Este análisis consiste en extraer características relacionados con el color, propiedades morfológicas o categóricas (por ejemplo, indicar si una imagen concreta está desenfocada o la célula contenida está ocluida). Estas características siguen el

hilo observado en el actual estado del arte y son propicias para ser utilizadas como vectores de características en posteriores algoritmos de aprendizaje y clasificación.

No obstante, el objetivo del proyecto era diseñar en última instancia una herramienta para los hematólogos que fuera capaz de adaptarse y, como se ha comentado anteriormente, realizar una segmentación basada en estas características limita las posibilidades de generar modelos flexibles a largo plazo cuando podrían cambiar las propiedades observadas en las imágenes si se introduce nuevos dispositivos, métodos o formas de adquirir las imágenes. Rechazando un posible enfoque basado en *deep learning* y aquellos fundamentados en una previa segmentación de las imágenes, se decide finalmente abordar el problema desde una perspectiva intermedia entre las anteriores mediante detectores automáticos de características locales en las imágenes, puesto que es un procedimiento más autónomo que realizar una segmentación, pero menos independiente que uno basado en una arquitectura CNN, ya que este último no demanda de extracción y análisis previo de las imágenes. Éstos permiten extraer información robusta de las imágenes con propiedades que los hacen ideales para ser integrados en un posterior esquema de clasificación mediante un conjunto amplio de algoritmos de detección de diferentes tipos disponibles.

Por tanto, este trabajo arranca desde la necesidad de investigar, explorar y comparar diferentes detectores y caracterizadores de puntos de interés locales adecuados para realizar una posterior extracción de características de las imágenes de interés [25, 40, 6]. Emplear la extracción de características locales de imágenes puede considerarse como una fase relevante en el proceso de clasificación de objetos visuales, puesto que se integra muy bien con un amplio y conocido conjunto de métodos de análisis de imágenes [39]. El estudio se basa a su vez en conocer la configuración más adecuada de los detectores para su uso con imágenes de glóbulos blancos.

Además, se dispone de la posibilidad de comparar detectores densos y dispersos en términos de complejidad computacional y prestaciones de clasificación del conjunto de datos sobre glóbulos blancos. Para alcanzar dicha meta se contempla la consecución del siguiente esquema de objetivos centrados en la materia de este trabajo:

- Conocer las últimas tendencias de aprendizaje automático aplicado a problemas de visión por ordenador con especial atención en el estudio del estado del arte actual bajo el tópico de clasificación automática de glóbulos blancos.

- Estudiar la caracterización de las imágenes de glóbulos blancos por métodos basados en detección de puntos característicos.

- Manejar métricas apropiadas y válidas para la naturaleza del conjunto de datos disponible para generar resultados que atiendan de la mejor manera posible a las particularidades del problema.

- Diseñar un método de aprendizaje completo y eficaz que integre la extracción de puntos característicos locales en las imágenes. El proceso de clasificación debe cubrir los requerimientos propuestos en cuanto a prestaciones que

desean obtenerse teniendo en cuenta la naturaleza inherente del problema. Dicho esquema permitirá evaluar y comparar el desempeño de cada algoritmo de extracción de puntos característicos locales propuestos.

Capítulo 2

Metodología

2.1. Descripción general

En el actual trabajo se propone realizar un método o proceso de clasificación de imágenes de glóbulos blancos a través de un esquema basado en el concepto de "bolsa de palabras visuales" (en inglés, *Bag of Visual Words*, *Bag of Features* o *Bag of Keypoints*). Se trata de un enfoque que toma por analogía a los métodos de aprendizaje que emplean el método de bolsa de palabras para categorización de texto [42].

Su uso como procedimiento de clasificación no está presente en el actual estado del arte referente al tópico de interés de este trabajo, pero es bien aplicado en otros ámbitos y tareas de clasificación: objetos [13], gestos [39] o sistemas de diagnosis computarizado (CAD) [9].

Este método se resume principalmente las siguientes partes:

1. **Detección de puntos característicos locales en las imágenes**. Existe una gran diversidad de algoritmos de representación del contenido de la imagen [40]: SIFT, SURF, FAST, CenSurE, Harris, MSER, etc.

2. **Descripción de las regiones subyacentes a los puntos detectados anteriormente**. La descripción de las regiones puede ser binaria, por ejemplo, BRISK, BRIEF y FREAK; o basados en histogramas de gradientes orientados (en inglés, *Histogram of Oriented Gradients* o HOG), por ejemplo, SIFT, SURF y GLOH [26].

3. **Asignar los descriptores de las regiones a un predeterminado número de clústeres**. Estos grupos también se denominan "palabras" y al conjunto de "palabras" se denomina "vocabulario"(en inglés, *codebook*), halladas por medio de un algoritmo de cuantificación. Los clústeres se constituyen y definen por los vectores representantes o centroides. Generalmente, el algoritmo más

habitual de *clustering* empleado en la literatura para realizar la construcción del "vocabulario" es *k-means*.

4. **Construir la "bolsa de palabras visuales"** (*Bag of Visual Words*, *Visual* BoW). Deben calcularse el histograma o distribución de descriptores asociados a cada clúster y para cada imagen.

5. **Aplicar un algoritmo de aprendizaje multiclase**. Se toma la *Visual* BoW como los vectores de características necesarios para generar un modelo de clasificación resultante de una fase de aprendizaje con el fin de determinar las categorías asociadas a cada imagen.

2.2. Detección de características locales

El objetivo principal de los algoritmos de detección de puntos locales consiste en codificar la estructura local distintiva de una imagen por medio de la búsqueda de patrones que difieren del vecindario próximo. Se asocian, generalmente, a cambios en una o varias propiedades de la imagen de manera simultánea. Estas propiedades pueden ser la intensidad, color y textura. Las características locales pueden ser puntos, esquinas o pequeños *blobs* en la imagen. Estas características se miden y codifican posteriormente por medio la descripción de la región centrada en el resultado de la detección.

Las buenas características deben reunir las siguientes propiedades [40]:

- **Repetitividad**. Dado un conjunto de imágenes del mismo objeto o escena tomadas bajo diferentes condiciones de visualización, un alto porcentaje de las características detectadas en la parte visible de la escena observadas en el conjunto de imágenes deben encontrarse en la totalidad de imágenes relacionadas.

- **Diferenciabilidad**. Los patrones de intensidad subyacentes a las características detectadas deben mostrar mucha variación, de modo que las características se pueden distinguir y combinar.

- **Localidad**. Las características deben ser locales, para reducir la probabilidad de oclusión y permitir aproximaciones de modelo simples de las deformaciones geométricas y fotométricas entre dos imágenes tomadas bajo diferentes condiciones de visualización.

- **Cantidad**. el número de características detectadas debe ser suficientemente grande, de tal manera que se detecte un número razonable de características incluso en objetos pequeños. Sin embargo, el número óptimo de características depende de la aplicación. Idealmente, el número de características detectadas debe ser adaptable en un rango amplio por un umbral simple e intuitivo. La densidad de las características debe reflejar el contenido de la información de la imagen para proporcionar una representación compacta de la imagen.

- **Precisión**. Las características detectadas deben estar localizadas con precisión, tanto en la ubicación de la imagen, como con respecto a la escala y la forma.
- **Eficiencia**. Preferentemente, la detección de características en una nueva imagen debería ser posible en aplicaciones sujetas a exigencias temporales.

La repetitividad posiblemente es la propiedad más importante de todas y se puede lograr de dos maneras diferentes: invariancia o robustez:

- **Invariancia**. Cuando se esperan grandes deformaciones, se prefiere modelar tales cambios matemáticamente, si es posible, para desarrollar métodos para la detección de características que no se ven afectadas por estas transformaciones matemáticas.
- **Robustez**. En el caso de deformaciones relativamente pequeñas, a menudo es suficiente hacer que los métodos de detección de características sean menos sensibles a tales deformaciones, es decir, la precisión de la detección puede disminuir, pero no drásticamente. Las deformaciones típicas que se abordan haciendo énfasis en la robustez son el ruido de imagen, efectos de discretización, artefactos de compresión, desenfoque, etc. También las desviaciones geométricas y fotométricas del modelo matemático utilizado para obtener invariancia se superan con frecuencia incluyendo más robustez.

Debido a las propiedades que reúnen este tipo de características, su detección tiene una potencial aplicabilidad en multitud de escenarios:

- **Generación de panoramas grandes a partir de imágenes**. Se involucran tareas de registrado y búsqueda de correspondencias.
- **Detección del movimiento con gran amplitud**. Encontrar la correspondencia en las características locales entre *frames* consecutivos en una secuencia de vídeo hace posible realizar un seguimiento continuo donde otro tipo de técnicas de seguimiento basadas en flujo óptico pueden quedar limitadas.
- **Visión estereoscópica y reconstrucción tridimensional**. De nuevo, la búsqueda de correspondencias entre imágenes puede resultar de utilidad para realizar tareas de triangulación y cálculo de la posición relativa de la cámara.
- **Reconocimiento de objetos**. Por último, bajo el tópico de interés en tareas que involucran la clasificación de imágenes, las propiedades de las características locales permiten identificar patrones en éstas y construir posteriormente vectores de características útiles y válidos en algoritmos de clasificación y aprendizaje.

Desde el punto de vista de clasificación de imágenes, la detección de características locales resulta de especial interés, puesto que permiten reconocer patrones, escenas y objetos sin necesidad de realizar una segmentación previa. Las propiedades

que deben presentar las características aptas para la clasificación son aquellas que se relacionen con la búsqueda de patrones claros en las imágenes. Éstas son, principalmente, repetitividad, diferenciabilidad y precisión, puesto que entre imágenes de un mismo tipo de glóbulo blanco deben encontrarse características similares con cierta periodicidad sin probabilidad de confusión ante pequeños cambios de iluminación, deformación u oclusión entre ellas. Quizá una propiedad que puede parecer menos relevante es la cantidad, pero que puede ser determinante para resolver casos en los que el clasificador necesite cierta resolución para definir las fronteras de decisión.

Como se ha comentado anteriormente, un extractor determinado busca características locales que se materializan en forma de puntos, esquinas o pequeños *blobs*. Este hecho puede ser interesante cuando se intenta realizar una interpretación semántica para un contexto limitado y para una determinada aplicación. Por otro lado, uno puede estar interesado más en la repetitividad de las características locales, puesto que proveen de un acotado conjunto de puntos de referencia bien localizados e individualmente identificables. Desde este enfoque no resulta interesante lo que semánticamente pueden representar los puntos, sino que su localización pueda determinarse de forma precisa y estable a través del tiempo.

A priori, resulta una tarea complicada determinar el tipo de característica local más afín a la hora de codificar la información contenida en las imágenes de los glóbulos blancos. Por este motivo, el objetivo de este trabajo es comparar una batería de extractores de características disponibles gobernados por criterios de análisis distintos. Al final de este estudio se pretende extraer conclusiones oportunas en el comportamiento de cada uno, con el fin de dirigir y escoger la mejor opción disponible para la problemática inherente del conjunto de datos. Además, nos situamos en la fase más relevante del método de clasificación, donde prima sustancialmente la calidad de la obtención de vectores de características en cuanto a categorización de las imágenes dentro de las respectivas clases.

A la hora de escoger el conjunto de detectores de características se persigue el querer reunir cierta heterogeneidad entre los distintos detectores de características locales. Por tanto, en nuestra selección decidimos integrar un total de cinco detectores, los cuales podemos clasificar entre densos y dispersos basados en HOG, desde unos más clásicos hasta otros más recientes: SIFT, dSIFT, oFAST, PHOW, CenSurE.

A continuación, se detallan brevemente el método de extracción y búsqueda de características locales asociado a cada algoritmo y su disposición visual en las imágenes de nuestra base de datos bajo una cierta configuración.

2.2.1. SIFT

Scale-Invariant feature transform (por sus siglas, SIFT) es un algoritmo de extracción y descripción de características locales publicado por primera vez en 1999 [25] y patentado más tarde en 2004. A diferencia de otros detectores de característi-

cas locales basados en esquinas publicados en ese momento, SIFT se formula como un algoritmo robusto, puesto que es invariante a escala y rotación. Debido a esta propiedad, SIFT permite encontrar los mismos puntos ante cambios o transformaciones afines en la imagen, dando lugar a puntos que pueden ser repetibles. Es por esta razón que el uso de SIFT puede extenderse a multitud de tareas dentro de visión artificial.

SIFT se puede separar en extractor y descriptor de puntos característicos. El principal problema que resuelve es la percepción de las esquinas a medida que modificamos la escala, puesto que una esquina puede dejarse de percibirse como tal cuando la escala se ve modificada. Para pequeños bordes, mantener la misma ventana es suficiente. No obstante, si se desea detectar esquinas más grandes, debemos recurrir a ventanas más grandes. Por ello se emplea un filtrado de escala y espacio por medio del operador Laplaciano de Gauss (*Laplacian of Gaussian*, LoG), el cual actúa como un detector de *blobs* con varios valores de σ que actúan como parámetro de escala. El procedimiento de detección de puntos característicos locales utilizado por el algoritmo SIFT se resume en los siguientes pasos:

- En cada escala se construye una lista con los principales puntos buscando los máximos locales. Puesto que LoG es un operador costoso, SIFT utiliza una aproximación conocida como diferencia de Gaussianas (*Difference of Gaussians*, DoG). DoG obtiene la diferencia entre dos valores distintos de escala, σ y $k\sigma$. Este proceso se realiza en distintas octavas de la imagen a través de una pirámide gaussiana (Figura 2.1a):

  $$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y),$$

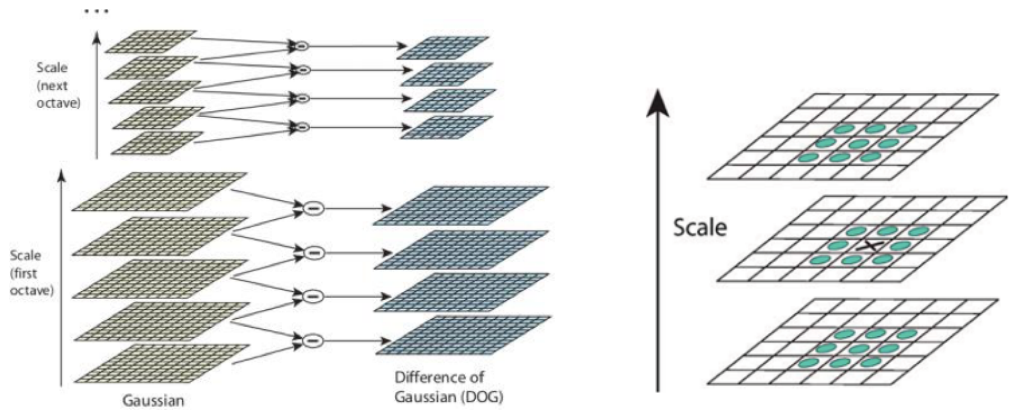
  donde $k > 1$ es el espacio entre las escalas adyacentes (típicamente k es $2^{1/4}$, $2^{1/3}$ o $2^{1/2}$).

- Una vez calculada la pirámide gaussiana se buscan los puntos característicos potenciales comparando cada píxel con sus 8 vecinos, así como de los 9 píxeles en la siguiente escala y en las anteriores (Figura 2.1b). Si se trata de un extremo local, es decir, $|D(x, y, \sigma)|$ es máximo, entonces resulta en un punto característico.

- A continuación, se refinan los puntos encontrados con el fin de generar resultados más precisos. Se emplea la expansión de las series de Taylor, además de un umbral de intensidad de los extremos. DoG tiene una fuerte respuesta para los bordes, por lo que éstos deben eliminarse. Para solventarlo se emplean los valores propios de la matriz Hessiana de 2×2 basada en la segunda derivada para calcular la curvatura principal. Se emplea una función de enventanado $w(x, y)$ para una imagen bidimensional en nivel de intensidad $I(x, y)$:

  $$H = \sum_{x,y} w(x, y) \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \tag{2.1}$$

  Un umbral regula el descarte de los bordes que exceden un determinado radio de curvatura. Con esta medida se descartan los puntos de bajo contraste y los que están asociados a bordes.



(a) **(b)**

Figura 2.1: (a) Cálculo de la diferencia de Gaussianas empleado en SIFT. Después de cada octava, la imagen se muestrea de nuevo con un factor 2. (b) Valor máximo y mínimo de la DoG detectados por comparación con los vecinos.

- Por último, con el fin de añadir invariancia a rotación, se añade información de rotación a cada punto. Para cada uno, se toma el vecindario y se calcula la magnitud y dirección del gradiente de cada región. Se construye un histograma de rotación de 36 bins cubriendo los 360 grados. Se toma el mayor pico del histograma y se calculan versiones del mismo punto con direcciones del histograma de gradiente por encima del 80 % de este valor. Posteriormente a este proceso de detección de puntos, se realiza la fase de descripción detallado en la sección 2.3.

En la implementación usada de SIFT se escogen los siguientes parámetros:

- **Umbral de bordes**: 10.
- **Umbral de intensidad**: 0,5.
- **Tamaño de la ventana (σ)**: 2.
- **Número de octavas**: máximo (depende del tamaño de la imagen). Es aproximadamente $\log_2(\text{mín}(anchura, altura))$. Con las imágenes de nuestra base de datos, con tamaño 512×512, este valor es 9.
- **Niveles por octava**: 3.

Para mejorar el rendimiento de la *Visual* BoW, debemos obtener una cantidad aceptable de puntos característicos. Para ello se decide escoger un umbral de intensidad bajo, consiguiendo una gran cantidad de puntos gracias a la aceptación de puntos de menor contraste.

Con dicha configuración se obtiene un promedio de 426 ± 209 puntos característicos por imagen en un total de 1315 ejemplos distribuidos en 6 clases. La localización de estos puntos tiende a concentrarse alrededor del glóbulo blanco de interés (Figura 2.2).

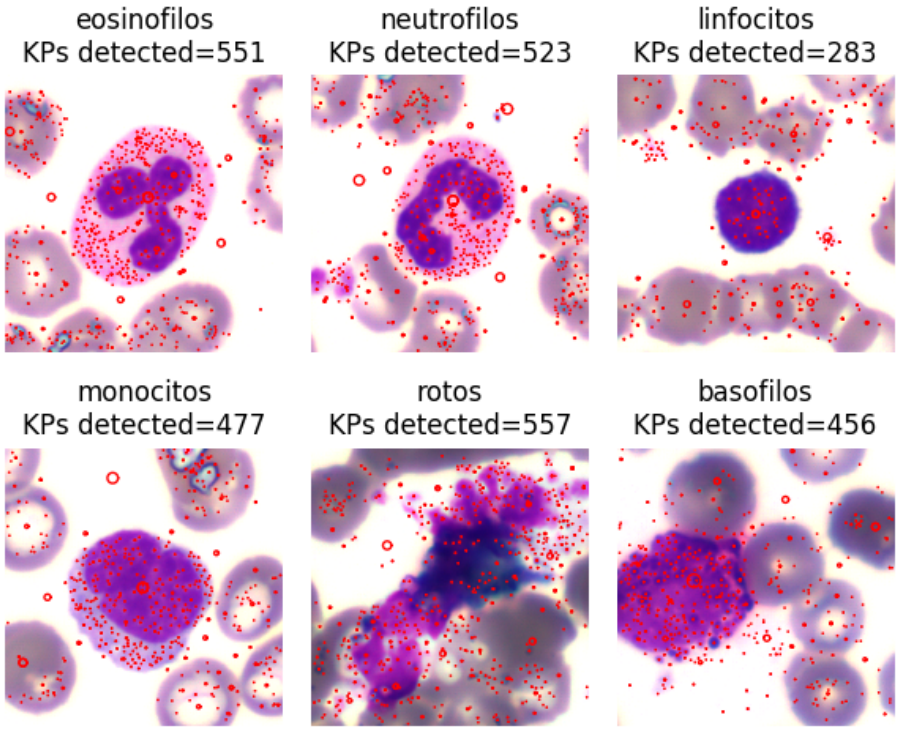


Figura 2.2: Ejemplos de distribución de puntos característicos SIFT encontrados en las imágenes de las diferentes clases del conjunto dado.

2.2.2. oFAST (*Oriented* FAST)

El algoritmo FAST (*Features from Accelerated Segment Test*) [32] es bien conocido por su rendimiento y sus prestaciones computacionales. Además, este algoritmo prima por la sencillez del método de cálculo de los puntos característicos, puesto que parte de las ideas introducidas en el algoritmo SUSAN (*Smallest Univalue Segment Assimilating Nucleus*). Primordialmente, el método inicial propuesto por Edward Rosten y Tom Drummond [31] se realizaba de la siguiente manera:

- Selección de un píxel p con intensidad I_p en la imagen para ser o no identificado como esquina.
- Elección de un umbral con valor t. Considerar un círculo de radio 16 alrededor del píxel de interés.
- El píxel p se considera esquina si existe un conjunto de n píxeles contiguos al círculo que tienen un valor de intensidad superior a $I_p + t$ o inferior a $I_p - t$.
- Para descartar rápidamente píxeles que no son considerados esquinas se utiliza un test que examina solo los píxeles 1, 9, 5 y 13. Si los píxeles 1 y 9 cumplen la condición anterior se procede a comprobar los píxeles 5 y 13. Si p es una esquina, debe cumplirse la condición al menos para 3 de los 4 píxeles.

Este enfoque presentaba limitaciones en cuanto a eficiencia, puesto que múltiples características detectadas son adyacentes unas de otras. Además, la eficiencia

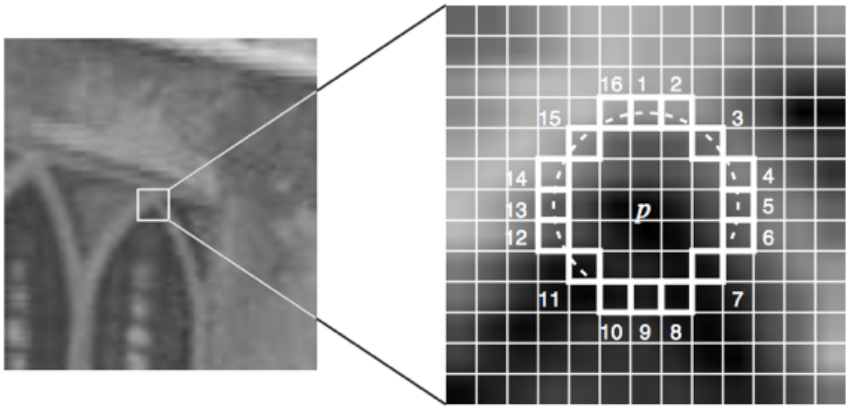


Figura 2.3: Ejemplo de test de identificación de esquinas en un círculo de Bresenham de 12 puntos para una región de la imagen [32].

depende del ordenamiento de la distribución de las esquinas.

FAST resuelve estas limitaciones por medio de una comprobación de los píxeles basada en aprendizaje automático por medio del algoritmo ID3 (fundamentado en árboles de decisión) y el método de supresión de no máximos.

Sin embargo, FAST no tiene componente de orientación, un parámetro muy importante y necesario si se requiere de descripción de características mediante SIFT o garantizar invariancia a rotación. El algoritmo ORB [33] solventa esta carencia añadiendo el cálculo de la componente de orientación en la detección FAST. Esta versión se conoce como oFAST.

Tal como se explica en su respectivo artículo, primero se obtienen los puntos FAST tomando únicamente como parámetro de control el umbral de intensidad entre un píxel central y aquellos contenidos en un anillo circular sobre éste. Se utiliza FAST-9 (radio circular de 9 píxeles), el cual obtiene buenos resultados. Similar a SIFT, FAST genera una respuesta fuerte en los bordes. Se emplea una medida de esquinas de Harris para ordenar los puntos, de manera que se puede elegir un número N objetivo de puntos en la detección. Esta medida categoriza matemáticamente las ubicaciones asociadas a esquinas, bordes o regiones planas. Estos casos se determinan en función del cambio significativo detectado en todas las direcciones. La detección de Harris busca el cambio de intensidad en el desplazamiento en la región $[u, v]$ por medio de una función de enventanado $w(x, y)$ para una imagen bidimensional en nivel de intensidad $I(x, y)$:

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2$$

En función de la respuesta de la región determinada por el término $[I(x+u, y+v) - I(x, y)]$, dará como resultado uno de los tres posibles casos anteriores. En cualquier caso se busca maximizar la siguiente expresión:

$$\sum [I(x + u, y + v) - I(x, y)]^2$$

A través de la aproximación de primer orden de las series de Taylor se llega a la expresión siguiente:

$$\approx \sum [I(x,y) + uI_x + vI_y - I(x,y)]^2$$

Reescribiendo la expresión anterior en forma de matriz:

$$= [u \quad v] \left(\sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

Para pequeños desplazamientos $[u, v]$ se tiene la siguiente aproximación bilineal:

$$E(u,v) \cong [u,v] M \begin{bmatrix} u \\ v \end{bmatrix},$$

donde M es una matriz 2×2 calculada desde las derivadas de la imagen. Ésta es conocida como la matriz de Harris:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{2.2}$$

La distribución de las derivadas en x e y se caracterizan por la forma y tamaño de los componentes principales de la elipse mediante los valores propios λ_1 y λ_2. Las esquinas se caracterizan por valores elevados de λ_1 y λ_2.

Debido a que FAST original no emplea detección de características en multi-escala, se calculan los puntos dentro de una pirámide de escalas filtradas por Harris para cada nivel.

Para el cálculo de las orientaciones se emplea una medida efectiva basada en la intensidad de los centroides. Esta medida asume que la intensidad de la esquina está desplazada de su centro. Se basa en calcular los momentos de una región como:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y)$$

Una vez hallados, se calculan los centroides:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

Con estos datos, se calcula el vector distancia $\overrightarrow{OC}$ entre el centroide C y el centro de la esquina. Por tanto, la orientación de la región se halla de la siguiente forma:

$$\theta = \text{atan2}(m_{01}, m_{10}),$$

donde atan2 no tiene en cuenta el signo de la orientación, por tanto, no es relevante si la esquina es oscura o clara.

En la implementación usada de oFAST, se escogen como parámetros:

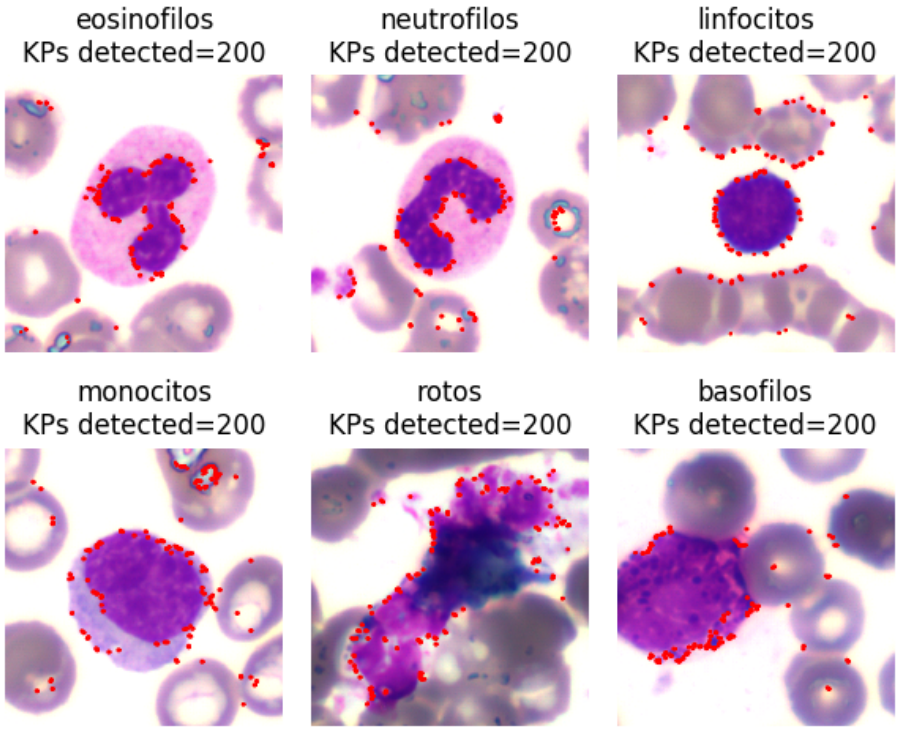


Figura 2.4: Ejemplos de distribución de puntos característicos oFAST encontrados en las imágenes de las diferentes clases del conjunto dado.

- **Número objetivo N de puntos característicos**: 200.
- **FAST-9**.
- **Umbral t**: 0,04.
- **Factor de sensibilidad de Harris**: 0.
- **Número de escalas**: 8.
- **Factor de escala**: 1,2.

Con dicha configuración se obtiene un promedio de 192 $\pm$ 31 puntos característicos por imagen en un total de 1315 ejemplos distribuidos en 6 clases. La localización de estos puntos tiende a concentrarse en los contornos o bordes de las formas de la imagen (Figura 2.4).

2.2.3. CenSurE

Agrawal et al. [6] introdujeron en la conferencia europea de visión por ordenador (*ECCV*) en su edición de 2008 un nuevo detector invariante a escala con un método de filtrado alrededor del centro llamado CenSurE (*Center Surround Extrema*). Este detector tiene un enfoque simple cuyo coste es independiente de la escala de la

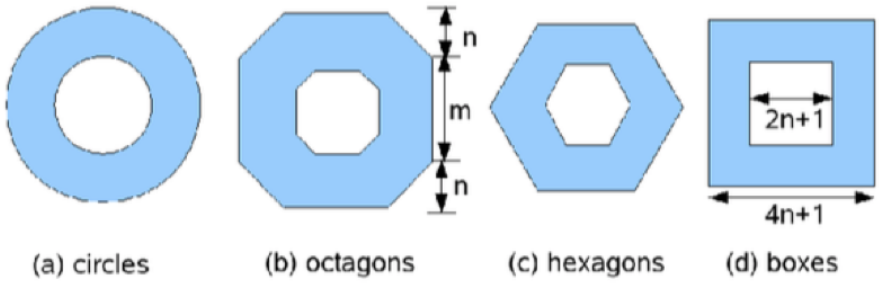


Figura 2.5: Progresión de los filtros bi-nivel disponibles en CenSurE [6].

pirámide, en contraposición a SIFT. Por tanto, se trata de una propiedad que permite al algoritmo CenSurE ser viable en tareas con exigencias temporales.

Los valores extremos de la función Laplaciana a través de las escalas muestran un comportamiento muy estable, por lo que se toma un operador más general, como es el caso del valor de los extremos en la respuesta alrededor del centro. Los autores demostraban también que este nuevo detector superaba al resto de detectores invariantes a escala en términos de cantidad de correspondencias en las características encontradas y en su precisión.

A diferencia de SIFT, CenSurE filtra los bordes a través de un filtro Harris (Ecuación 2.2), en vez de emplear una matriz Hessiana (Ecuación 2.1). Aseguran que obtiene mejor rechazo de bordes respecto al método empleado por SIFT. Como se ha comentado anteriormente, Lowe empleaba una aproximación del operador Laplaciano denominado DoG. En CenSurE se emplea una aproximación más sencilla basada en un filtro alrededor del centro mediante filtros bi-nivel. Consiste en multiplicar la imagen por 1 y -1. Estos filtros pueden ser, en progresión por grado de simetría (Figura 2.5):

- **Círculo**. La mejor aproximación del operador Laplaciano, pero con mayor coste. Este método recibe el nombre del Laplaciano de Gauss bi-nivel (*Bi-Level Laplacian of Gaussian*, BLoG).
- **Octágono**. Buen compromiso entre rendimiento y coste.
- **Hexágono**. Comportamiento similar al filtro tipo octágono.
- **Cajas**. Consiste en dos cuadrados, donde un cuadrado de tamaño $(2n+1) \times (2n+1)$ se sitúa en el interior de uno más grande de tamaño $(4n+1) \times (4n+1)$, siendo n el tamaño del bloque del filtro. Este método se denomina diferencia de caja (*Difference of Box*, DOB). Tiene un coste reducido, pero emplea un kernel que no es invariante a rotación; por tanto, es menos preciso que los anteriores.

Se emplean siete tamaños de bloque, $n \in \{1, 2, 3, 4, 5, 6, 7\}$, de la *wavelet* de Haar para encontrar las características. Para asegurar repetitividad se emplea la supresión

de no máximos con el fin de garantizar características con una buena respuesta en todas las escalas.

Las características que se sitúan a lo largo de una línea o borde se localizan de manera insegura y son menos estables. Así como ocurría con SIFT, se rechazan las respuestas de las líneas con un radio de curvatura por encima de un umbral por medio de la medida de Harris. Tiene un mayor coste, pero ofrece mejor precisión que la matriz Hessiana.

En la implementación usada de CenSurE se escogen como parámetros:

- **Tipo de filtro bi-nivel**: STAR (implementación computable del círculo).
- **Umbral de supresión de no máximos**: 0.01.
- **Umbral de rechazo de la medida de Harris**: 50.
- **Escala mínima**: 1.
- **Escala máxima**: 7.

Con dicha configuración se obtiene un promedio de 161 ± 115 puntos característicos por imagen en un total de 1315 ejemplos distribuidos en 6 clases. La localización de los puntos tiende a concentrarse en zonas de gran contraste y alrededor de *blobs* en las formas de la imagen (Figura 2.6).

2.2.4. dSIFT (*dense* SIFT)

Como se ha explicado anteriormente, SIFT se compone principalmente de cuatro fases: detección de extremos en escala y espacio, localización de puntos característicos, asignación de orientación y descripción de los puntos detectados. SIFT es aplicable a problemas de detección y reconocimiento de objetos. Sin embargo, tal como se detalla en una publicación de Wang et al. [43], la detección es limitada cuando se aplica a reconocimiento de caras, un escenario donde puede darse falta de textura, escasez de iluminación y baja resolución en las imágenes. Por tanto, pocos puntos son detectados. Lo mismo ocurre en problemas de reconocimiento de las venas o impresión de la huella de la palma de la mano. Para solventar este problema, en el algoritmo SIFT denso (*dense* SIFT, dSIFT) se omiten las tres primeras fases del algoritmo SIFT relacionados con la detección de puntos. El proceso de detección realizada previamente en SIFT se sustituye por un proceso de construcción de una malla de puntos predefinida y se realiza únicamente la descripción de las características locales sobre estos puntos.

Escoger una definición de puntos densa se basa en una primera hipótesis que nos conduce a intuir que la información sensible de la imagen se tiende a concentrar en el centro de la imagen, donde se sitúa normalmente el glóbulo de interés. Es por este

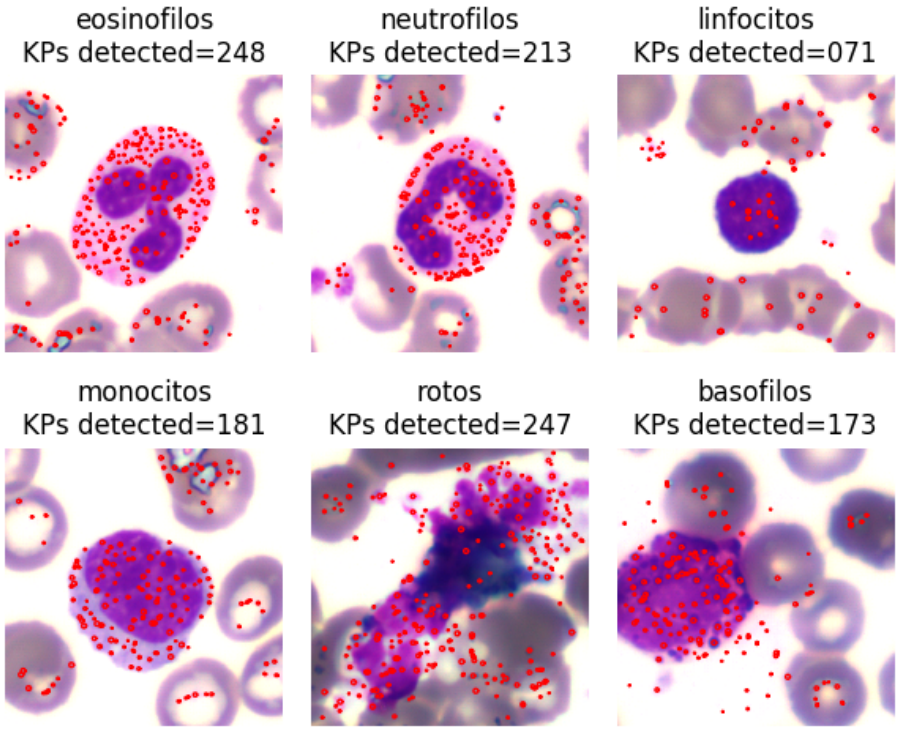


Figura 2.6: Ejemplos de distribución de puntos característicos CenSurE encontrados en las imágenes de las diferentes clases del conjunto dado.

motivo que resulta interesante definir y describir una mayor cantidad de puntos en esta zona. Por otro lado, la zona del contorno de la imagen no contiene información directamente relacionada con el glóbulo de interés y, por este motivo, se puede optar por prescindir o reducir la cantidad de puntos en esta localidad de la imagen. Con este planteamiento es posible escoger y forzar un número determinado de puntos en la imagen. Por otro lado, no se garantiza la propiedad de repetitividad entre los puntos de imágenes distintas al disponer de una malla de puntos fija.

Puesto que la malla de puntos densa se genera de forma manual, existen tres formas de definirla dependiendo del grado de compromiso que se requiera entre tiempos de cómputo razonables y rendimiento de la clasificación en etapas posteriores:

- **Invariancia espacial**: por defecto, en el algoritmo dSIFT solo se debe especificar la separación en píxeles entre los puntos de la rejilla para que ésta se construya de manera uniforme y automáticamente de acuerdo al tamaño de la imagen. Esta propiedad de dSIFT por defecto de rejilla uniforme en toda la localidad de la imagen se denomina invariancia espacial.

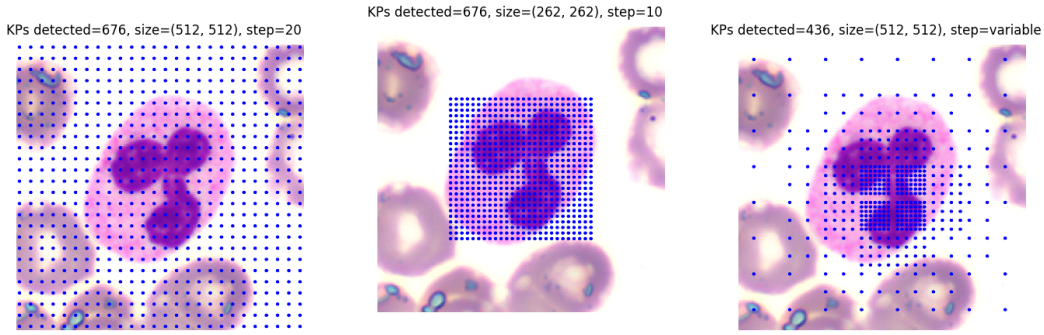
- **Región de interés** (*Region of interest*, ROI): si se quiere obtener mayor densidad de puntos sin elevar en exceso la cantidad de puntos global debido a su mayor sobrecoste en las fases posteriores del algoritmo, se puede definir una ROI en torno al centro de la imagen. En esta localidad de la imagen tiende a situarse el glóbulo blanco de interés, razón por la cual se quiere definir una mayor densidad de puntos. Este enfoque pretende poner a prueba la hipótesis

previamente formulada. Por otro lado, al definir una ROI se podría perder la información de contexto relacionada con la parte del fondo (*background*). En términos generales, en problemas de clasificación de imágenes, un tipo concreto de objeto o entidad de interés en una imagen lleva asociado un determinado contenido en el fondo (por ejemplo, una fotografía de un coche contiene normalmente una carretera como fondo). *A priori* se desconoce si esta información puede resultar útil en un problema de clasificación de glóbulos blancos observando las imágenes de la base de datos. No obstante, es interesante investigar esta propuesta y comparar que rendimiento se obtiene con dicha configuración de malla respecto al resto de formas de definirla, con el fin de ponderar el peso que tiene en el rendimiento general del método de clasificación el omitir este tipo de información.

- **Variancia espacial**: para obtener un compromiso entre resolución suficiente de puntos en la región del glóbulo de interés y una cierta definición del fondo de la imagen sin elevar en exceso la cantidad global de puntos, se recurre a definir una malla o rejilla de densidad variable en función de la localidad de la imagen. Este enfoque de densidad variable de puntos adquiere la propiedad de variancia espacial y es una situación de compromiso entre tiempos de cómputo asequibles en las fases posteriores junto con la obtención de un buen comportamiento del esquema de clasificación.

En la Figura 2.7 puede observarse estas tres formas propuestas de definir la distribución de la malla de puntos. Al definir una ROI se aumenta la densidad de puntos (separación de puntos más pequeña) sin aumentar la cantidad de puntos con el coste de dejar puntos sin definir en algunas localidades de la imagen. En la sección 3.3.2 se ha realizado un pequeño estudio de cómo afecta la elección de cada una de las distintas posibilidades de estrategias de muestreo de puntos propuestas a efectos de la clasificación de la *Visual* BoW de las fases posteriores y, por tanto, determinar cuales podrían ser más apropiadas para este problema.

Para este trabajo se apuesta por el enfoque que emplea una rejilla variable con variancia espacial. Para implementarlo, primero se realiza una subdivisión de las imágenes en 64 cuadrantes o subimágenes. La información sensible se concentra generalmente en el centro de la imagen, donde está definido el glóbulo blanco de interés. Para calcular la densidad de la rejilla en una localización dada de la imagen se tiene en cuenta la distancia euclídea en píxeles que existe entre el cuadrante central respecto a un cuadrante dado. A mayor distancia, menor densidad de puntos o mayor separación entre éstos. Las imágenes tienen un tamaño fijo de 512×512, por lo que la subdivisión realizada en todas las imágenes del conjunto de datos es constante. En total se tiene 8×8 cuadrantes o subimágenes de tamaño 64×64 píxeles manejados por índices que indican su posición en términos de fila y columna respecto a la imagen original. La rejilla definida en cada cuadrante es proporcional tanto a su tamaño constante e inversamente proporcional a la distancia respecto al cuadrante central. Si la separación entre píxeles es igual al tamaño del cuadrante, solo dispondremos de un punto definido en tal cuadrante. Si la separación es la mitad del tamaño del cuadrante, dispondremos de 2 puntos y así sucesivamente.



(a) Invariancia espacial.

(b) Invariancia espacial en una región de interés.

(c) Variancia espacial.

Figura 2.7: Formas propuestas de definir la malla de puntos en dSIFT.

Por último, comentar que la formación de la rejilla a partir de los parámetros anteriores (separación entre píxeles y tamaño del cuadrante) mediante la implementación encontrada en la librería de Python especializada en extracción de características locales, *VLFeat* [5], genera un borde en los extremos de las imágenes. Este borde está causado por el *offset* de 4,5 píxeles que se introduce por defecto en la ubicación de los puntos. Tal como esta implementado no es posible modificarlo y la única solución posible es introducir un solapamiento entre los cuadrantes. No obstante, no es un detalle demasiado relevante que afecte al rendimiento posterior de este extractor, por tanto, se mantiene la configuración original. El efecto de bordes comentado es apreciable en la Figura 2.8.

En nuestra implementación, este procedimiento de formación de la rejilla a nivel de imagen da como resultado la siguiente distribución de puntos:

- 32 cuadrantes con separación 64: 32 puntos característicos en total.
- 20 cuadrantes con separación 32: 80 puntos característicos en total.
- 8 cuadrantes con separación 16: 128 puntos característicos en total.
- 4 cuadrantes con separación 8: 196 puntos característicos en total.

Por tanto, se obtiene un total de 436 puntos característicos fijos por imagen con la configuración elegida, la cual ofrece una razonable solución de compromiso entre rendimiento y coste de ejecución en todo el proceso de la *Visual* BoW (Figura 2.8).

2.2.5. PHOW

Las características PHOW (*Pyramid of Histograms of Visual Words*) son una variante de la descripción de SIFT densa cuya descripción se realiza a múltiples escalas [11]. Una "bolsa de palabras visuales", entendida como las características de la imagen, es un conjunto de vectores disperso con la frecuencia de las "palabras"

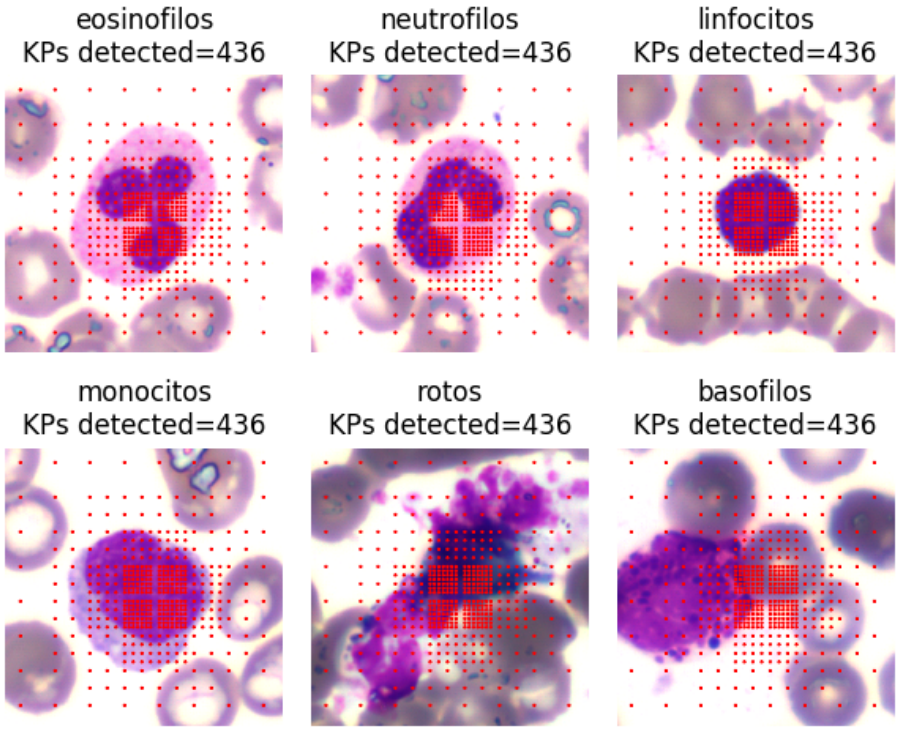


Figura 2.8: Ejemplos de distribución de puntos característicos dSIFT encontrados en las imágenes de las diferentes clases del conjunto dado.

repetidas en una imagen. El principal problema con el modelo de "bolsas de palabras visuales" es que la información espacial de las características de la imagen ya no está disponible en la representación del modelo. En *Visual* BoW sabemos que una característica particular existe en la imagen, y sabemos con qué frecuencia, pero no podemos decir dónde en la imagen. Este enfoque aborda este problema construyendo una pirámide espacial.

Se realiza subdividiendo la imagen en una creciente rejilla más fina por medio de una descomposición *quadtree*. De esta forma se obtiene una secuencia de rejillas desde el nivel 0 hasta el nivel L. Las características PHOW se calculan en cada subregión y a diferentes niveles por medio de la descripción SIFT densa. Posteriormente, se realiza una cuantificación realizando un agrupamiento mediante *k-means* con $k = 1000$ y concatenando los histogramas resultantes para generar los descriptores.

En la implementación usada de PHOW en este trabajo, para definir la rejilla, se selecciona previamente una región de interés (ROI), ya que el glóbulo blanco de interés se encuentra en el centro en la mayoría de las imágenes. Las imágenes tienen un tamaño fijo de 512×512 y la ROI recorta 125 píxeles de cada extremo, dando lugar a una subimagen de 262×262. Mediante esta estrategia se consigue reducir considerablemente la cantidad de puntos y, por consecuencia, el tamaño de la descripción de cada imagen respecto a emplear la imagen en su totalidad.

En el caso de PHOW no conviene separar las imágenes en cuadrantes, puesto que se pierde mucha información en la descripción debido al efecto de introducción de

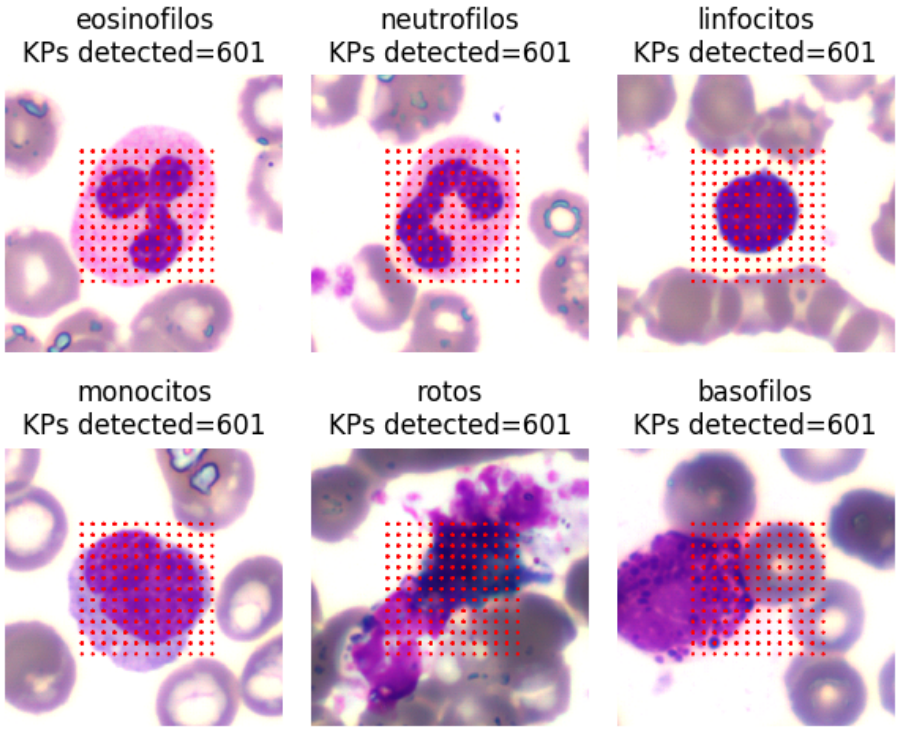


Figura 2.9: Ejemplos de distribución de puntos característicos PHOW encontrados en las imágenes de las diferentes clases del conjunto dado.

bordes que realiza la implementación encontrada de PHOW en la librería de *VLFeat*, al igual que ocurría en el caso de dSIFT. Esta falta de continuidad podría resultar crucial en los fases posteriores de dicho algoritmo.

Por tanto, únicamente la definición de la ROI y la separación s de 20 píxeles entre puntos consecutivos son los encargados de generar la rejilla final de cada imagen. El número de puntos característicos queda determinado por la siguiente relación:

$$keypoints \approx \left(\frac{anchura \cdot s}{altura} \right)^2 \cdot 3$$

Obteniendo 601 puntos característicos fijos por imagen con la configuración elegida (Figura 2.9).

2.3. Descripción de las características locales

La descripción de las regiones subyacentes al conjunto de características detectadas en las imágenes es un proceso intermedio en la obtención del vector de características habitual de cualquier algoritmo de aprendizaje empleado en el método de bolsa de palabras visuales sugerido en este trabajo. Este vector de características se halla por medio de los histogramas de pertenencia a las posibles palabras del vocabulario creado por un algoritmo de *clustering*, habitualmente *k-means*.

El proceso de agrupamiento *k-means* emplea la métrica de la distancia euclídea y la varianza como medida de dispersión de los grupos [22]. Por este hecho, debemos plantearnos dejar de lado propuestas de descripción binaria, aun siendo más eficientes y precisos, como son el caso de BRIEF [12], BRISK [24] o FREAK [7], recurriendo a posibles descriptores basados en la familia de los histogramas de gradientes orientados (HOG), como SIFT [25], GLOH [26], SURF [10] o DAISY [38].

Formalmente, la descripción de características busca que, dado un punto característico en la posición $\overrightarrow{x}$, escala σ y orientación θ, describimos la estructura de la imagen en una vecindad de $\overrightarrow{x}$, alineada con θ, y proporcional a σ. Para facilitar la coincidencia, el descriptor debe ser distintivo e insensible a las deformaciones locales de la imagen.

Como se ha comentado anteriormente en el apartado 2.2, SIFT es un extractor-descriptor de características que aúna ambos procesos. Puesto que SIFT es un algoritmo clásico de la literatura que ha demostrado un fabuloso desempeño en multitud de escenarios y aplicaciones, se decide emplear este tipo de descripción como parte de este proceso a continuación de todos los algoritmos de detección de características anteriores. En esta sección, se detalla el procedimiento de descripción utilizado en el algoritmo SIFT. Este proceso consiste en los siguientes pasos:

1. Dado un punto característico, se rodea la región alrededor de éste y se transforma dicha región a una rotación y escala canónica. Este espacio se reescala a una región de 16 píxeles.

2. Tras ello, se calculan la magnitud y orientación del gradiente para cada uno de los píxeles:
$$|\nabla L| = \sqrt{L_x^2 + L_y^2}$$
$$\arg \nabla L = \text{atan2}(L_y, L_x)$$

3. A continuación, se dividen las regiones de 16×16 píxeles en cuadrantes de 4×4 píxeles.

4. Para cada cuadrante se calcula el histograma de la dirección del gradiente por medio de 8 bins (Figura 2.10). Todos los histogramas se concatenan formando un vector de 128 (16×8) y se normalizan respecto a la unidad con el fin de mejorar la invariancia a cambios de iluminación afines. Para mitigar los efectos de iluminación no lineal se aplica un umbral y se vuelve a normalizar el vector. Gracias a este procedimiento, SIFT es invariante a rotación y cambios de iluminación, ya que los histogramas no contienen información geométrica.

2.4. Construcción del "vocabulario": *clustering*

Una vez descritas todas las características detectadas en las imágenes se desea crear un vocabulario que englobe todas las "palabras" que pueden darse en la des-

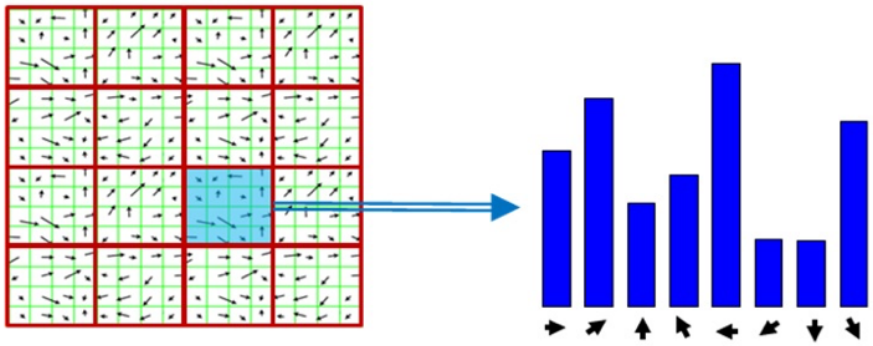


Figura 2.10: División de la región 16×16 píxeles en celdas de 4×4 píxeles. Extracción de los histogramas de la dirección del gradiente para cada celda. Imagen de: https://gilscvblog.com

cripción. *A priori*, no es posible determinar su cantidad, pero sabemos que descripciones similares dentro del espacio de descripción de SIFT pueden darse por regiones características con una cierta repetitividad. Generalmente, estos conjuntos de instancias similares forman agrupaciones.

La manera de encontrar estas agrupaciones es recurrir a un algoritmo no supervisado de *clustering*. El algoritmo *k-means* [20] es uno de los métodos más empleados en la literatura que, a pesar de tener una complejidad NP-hard, es eficiente y preciso gracias a la gran variedad de heurísticos disponibles. En *k-means* debe escogerse el valor de k o números de clústeres estimados presentes en el espacio euclídeo. Los descriptores SIFT tienen una dimensionalidad $d = 128$, como se ha visto en el apartado anterior. Las instancias son el número n de características descritas por imagen. Sabiendo esto, el problema puede resolverse en $O(n^{dk+1})$ en la aproximación original.

Este algoritmo consiste en calcular la posición de los centroides de los k clústeres que se quieren encontrar. Estos centroides sirven como "representantes" de los clústeres o, en el caso de *Visual* BoW, de las "palabras" del vocabulario.

El proceso de cálculo de *k-means* comienza inicializando de manera aleatoria la localización de los k centroides. Tras esto, se iteran dos pasos consistentes en:

1. **Asignar** cada observación x_p al centroide m_i del clúster $S_i^{(t)}$ más cercano.

$$S_i^{(t)} = \left\{ x_p \colon \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2, \quad 1 \leq j \leq k \right\} \tag{2.3}$$

2. **Actualizar** cada centroide $m_i^{(t+1)}$ con la media de sus observaciones asociadas.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{2.4}$$

Lo que busca *k-means* es intentar minimizar la distorsión de la suma cuadrática de las distancias entre las observaciones x_p asignadas a cada clúster $S_i^{(t)}$ y su centroide más cercano (ecuación 2.3). Puesto que los centroides se ajustan en el segundo

paso (ecuación 2.4), iterativamente se va modificando este valor de distorsión. El algoritmo finaliza cuando converge, es decir, el cambio de la distorsión entre sucesivas interacciones es menor que un umbral establecido, o bien se alcanza un valor máximo de iteraciones.

2.5. Cuantificación y obtención de histogramas

El proceso de obtención de los histogramas consiste en formar los vectores de características que describen cada una de las imágenes de acuerdo a la distribución de las características observadas respecto a la pertenencia de los clústeres o "palabras" calculadas.

Para cada imagen tenemos un número n de características u observaciones. Para cada una de ellas debe encontrarse la pertenencia al clúster más cercano. La distribución de este número n de características entre los k clústeres calculados conforma un histograma propio de cada imagen. Esta distribución o histograma es el vector de características necesario para cualquier algoritmo de aprendizaje supervisado elegido en la última fase.

La cuantificación se realiza por medio de la asignación de códigos del vocabulario generado a cada una de las observaciones. Suponiendo que tenemos la descripción de n características con una dimensionalidad d por imagen, esto es $n \times d$. Para cualquier punto, independientemente del algoritmo de detección empleado anteriormente, recordar que la descripción de cada punto se realiza por medio de SIFT. Esta descripción tiene una longitud $d = 128$.

El vocabulario que se genera tras el *clustering* mediante el algoritmo *k-means* tiene un tamaño de $k \times 128$, siendo k el número de "palabras" representadas por los centroides con dimensionalidad d. El código que se genera por imagen se corresponde a los índices de pertenencia de cada observación al clúster más cercano a través de un vector de tamaño d.

Por último, se obtiene el histograma por código, que consiste en medir la ocurrencia de cada índice, el cual tiene un tamaño k.

2.6. Aprendizaje y clasificación: SVM

Las máquinas de vectores de soporte (*Support Vector Machines*, SVM) es un algoritmo de aprendizaje supervisado que puede emplearse tanto para tareas de clasificación como para regresión, aunque su uso habitual se sitúa en aplicaciones de clasificación.

El algoritmo SVM se basa en la idea de encontrar un hiperplano que mejor divide

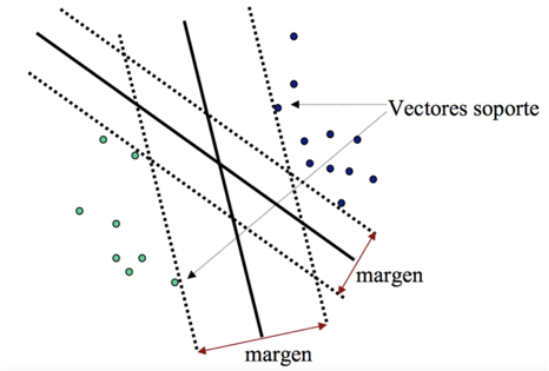


Figura 2.11: Definición del hiperplano que maximiza la separación entre las dos clases a través de los vectores de soporte.

un conjunto de datos en dos clases, como se muestra en la Figura 2.11. La manera más simple de realizar la separación es mediante una línea recta, un plano recto o un hiperplano N-dimensional.

Los vectores de soporte son los puntos de datos más cercanos al hiperplano, aquellos de un conjunto de datos que, si se eliminan, alteran la posición del hiperplano divisor. Debido a esto, pueden ser considerados los elementos críticos de un conjunto de datos.

Idealmente, el modelo basado en SVM debería producir un hiperplano que separe completamente los datos del universo estudiado en dos categorías. Sin embargo, una separación perfecta no siempre es posible y, si lo es, el resultado del modelo no puede ser generalizado para otros datos. Esto se conoce como "sobreajuste" (*overfitting*).

Con el fin de permitir cierta flexibilidad, los SVM manejan un parámetro C que controla el compromiso entre la complejidad del modelo y el número de datos que no son linealmente separables. Este parámetro se escoge empíricamente, generalmente por validación cruzada. Esto permite la creación de un margen blando (*soft margin*), que tolera ciertos errores en la fase de entrenamiento.

Generalmente, lo más habitual es encontrar problemas de clasificación donde las diferentes clases del conjunto de datos no son linealmente separables. Debido a la limitación de la idea principal basada en definir un hiperplano de separación, se ofrece una solución al problema gracias al mapeado del conjunto de datos en un espacio de mayor dimensionalidad donde si es posible definir más fácilmente el hiperplano de separación mediante un conjunto de funciones *kernel*, $K(x_i, x_j)$, método conocido como *kernel trick* [19].

Entre las posibles funciones *kernel* disponibles, este trabajo se emplea tanto el *kernel* lineal como la función de base radial (RBF). Éste último se encarga de realizar un *mapping* del espacio de entrada de la siguiente forma:

$$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}),$$

donde $\|x_i - x_j\|^2$ es la distancia cuadrática euclídea entre los vectores y σ es el

parámetro libre. También podemos simplificar la expresión anterior definiendo el parámetro $\gamma = \frac{1}{2\sigma^2}$, el cual es más común encontrarlo en la literatura:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$$

La principal ventaja de los SVM es la robustez en la generalización, una propiedad a tener en cuenta si lo que se busca es optimizar la precisión en la clasificación.

Aunque el SVM básico se formula para un problema binario (dos clases), es posible utilizarlo en problemas multiclase. Las dos opciones más habituales para ello son:

- Construir clasificadores binarios que distingan una clase respecto al resto. Este escenario se conoce como "uno contra todos". El clasificador que asigna la etiqueta final es aquel que maximiza una función de salida.
- Se construyen $k \cdot (k-1)/2$ modelos, donde k es el número de clases. Este enfoque se conoce en la literatura como "uno contra uno". La votación de la etiqueta final comienza asignando la instancia a una de las dos clases en cada clasificador. Acumulando los votos de todos los clasificadores, la clase más votada determina la etiqueta de salida.

Respecto a la versión de SVM que se emplea en este trabajo, trabajamos con la filosofía "uno contra uno". En cuanto al problema de seleccionar los valores de los hiperparámetros (*kernel* y C del SVM), le añadimos un método interno de fuerza bruta basado en la búsqueda exhaustiva por barrido de hiperparámetros, con el fin de garantizar la mejor elección del *kernel* y sus respectivos hiperparámetros para un conjunto de datos dado. Los hiperparámetros a optimizar son:

- C, en el caso del *kernel* lineal y RBF.
- γ, en el caso del *kernel* RBF.

El objetivo es maximizar una función de *score* objetivo mediante la elección de un modelo entre todos formado por una configuración concreta de hiperparámetros. Esta función de *score* puede ser una de las métricas más habituales empleadas en clasificación, las cuales se detallan en la sección 3.3.4. Todos los modelos generados se validan a través de una validación cruzada de 5 *splits*.

La elección final del *kernel* y sus hiperparámetros en el modelo resultante para un conjunto de datos dado describe en cierta medida la complejidad del problema, pudiendo analizar la eficacia de representación de la información realizada por los pasos anteriores de la *Visual* BoW. Por ejemplo, un valor elevado de C es indicativo de que se están tolerando muchos errores en el entrenamiento que no cumplen las restricciones impuestas, probablemente debido a un problema de clasificación cuasi-separable o no separable linealmente.

Capítulo 3

Experimentación y resultados

3.1. *Software* utlizado

El desarrollo del siguiente trabajo se ha realizado íntegramente bajo programación en Python. Se trata de un lenguaje multiparadigma con una gran cantidad de librerías muy completas para un compendio de aplicaciones muy variado: desarrollo web, aprendizaje automático, minería de datos, bases de datos, GUI, gestión matemática avanzada, etc. Esta flexibilidad unido a ser un lenguaje de licencia BSD (*software* libre) son las razones de su gran acogida en la actualidad, además de ser un lenguaje interpretable, legible y dinámico.

Llevando a cabo el desarrollo del trabajo, para cada fase que compone el método de la *Visual* BoW presentado se ha podido encontrar su respectiva implementación en este lenguaje a través de las distintas librerías disponibles para este fin. Además, cabía la posibilidad de escoger un determinado algoritmo entre varias implementaciones realizadas en distintas librerías a conveniencia. Por ejemplo, SIFT se trata de un algoritmo bajo patente (US 6711293 B1) y es difícil encontrarlo en los módulos genéricos de algunas librerías. OpenCV sí que lo implementa en una versión de C++, pero es necesario instalar los módulos no gratuitos, así como en el caso de la librería *VLFeat*. No obstante, es posible conseguir la implementación realizando una interfaz para Python del binario de SIFT.

En la Tabla 3.1 aparecen las referencias a las funciones y librerías que se han empleado en este trabajo para la realización de cada fase que compone el esquema completo de la *Visual* BoW propuesto. Aunque no figura en el contenido de esta tabla, el soporte de generación gráfico utilizado para obtener las correspondientes figuras de algunos resultados del trabajo se ha realizado con la librería *Matplotlib* (https://matplotlib.org/). La gestión y manipulación de datos en las fases intermedias del proceso se ha llevado a cabo mediante el empaquetado de *NumPy* (https://numpy.org/).

Comentar también que en el caso de los extractores de características empleados

Tabla 3.1: Resumen de las funciones y librerías empleadas en el trabajo para cada fase de la *Visual* BoW.

Fase	Algoritmo	Función	Librería
Detección de características	CenSurE	`CENSURE`	scikit-image [2]
	oFAST	`ORB`	scikit-image
	SIFT	`sift`	VLFeat [5]
	dSIFT	`vl_dsift`	VLFeat
	PHOW	`vl_phow`	VLFeat
Descripción de características	SIFT	`SIFT_create`	OpenCV [1]
Construcción del *codebook*	k-means	`kmeans`	SciPy [4]
Extracción de histogramas	Vector quantization (VQ)	`vq`	SciPy
	Cálculo de histograma	`histogram`	SciPy
Aprendizaje	SVM	`SVC`	scikit-learn [3]
	Búsqueda de parámetros	`GridSearchCV`	scikit-learn

solo se hace uso de la faceta de detección, puesto que éstos disponen en su gran mayoría de métodos para realizar la descripción de las regiones subyacentes a los puntos. La fase de descripción es común a todos ellos y se realiza a través de la descripción SIFT de la librería de OpenCV, una vez ajustado el formato de los puntos respecto a la estructura que define la interfaz de OpenCV para Python. Aunque se mencione que la función `ORB` se emplea para detectar puntos oFAST, matizar de nuevo que ORB es un algoritmo unificado que integra dos procesos: detección de puntos oFAST y descripción rBRIEF.

El entorno de desarrollo utilizado es PyCharm Community Edition 2017, puesto que simplifica la construcción de proyectos basados en Python. Tanto la escritura del código como el mantenimiento de los paquetes instalados se realiza de una forma más rápida y cómoda. El sistema operativo es la distribución de Ubuntu 16.04 LTS en un PC HP Intel® Core™ i5 CPU 650 con 8 GB de RAM.

3.2. Base de datos

Una vez introducida la metodología y configuración del proceso de clasificación automática propuesto en sus respectivas fases, se procede a describir la experimentación a realizar con el fin de validar el diseño del procedimiento llevado a cabo. La correcta experimentación permite extraer conclusiones relevantes en el funcionamiento del proceso.

El siguiente paso en el diseño de los experimentos a realizar en este trabajo continua por conocer la naturaleza de la base de datos proporcionada. Esta base de datos consiste en un conjunto de imágenes de tamaño fijo adquiridas y etiquetadas previamente por un especialista médico en el Servicio de Hematología del Hospital

General Universitario de Castellón de la Plana. Estas imágenes contienen, generalmente, una muestra o espécimen concreto de un tipo de glóbulo blanco visto en el primer capítulo, al que se le ha sometido a un proceso de tinción. En este conjunto de datos se ha decidido realizar la siguiente categorización en 6 etiquetas distintas: "eosinofilos", "neutrofilos", "linfocitos", "monocitos", "rotos" y "basofilos".

Se compone de un total de 1315 imágenes o ejemplos en color distribuidos en las clases anteriores con un tamaño fijo de 512×512. En la Tabla 3.2 se muestra la distribución del número de ejemplos en las distintas clases disponibles. Es observable el claro desbalance que existe entre las distintas clases, puesto que es común que algunas clases salgan favorecidas en cantidad de ejemplos por la abundancia natural en el flujo sanguíneo, como es el caso de los "neutrofilos" y "linfocitos". En el caso contrario encontramos "eosinofilos" y "basofilos", donde la combinación de ambas clases compone únicamente el 4 % del total de la base de datos.

Tabla 3.2: Distribución del número de imágenes en las etiquetas disponibles en la base de datos.

Clase	**# instancias**	**Porcentaje (%)**
linfocitos	511	38,86
neutrofilos	476	36,20
rotos	185	14,07
monocitos	99	7,53
eosinofilos	38	2,89
basofilos	6	0,46
Total	1315	100

A priori, podemos intuir que el desbalance entre clases puede ser un hándicap para las clases minoritarias, ya que la elaboración de un clasificador fundamentado en SVM o en otros tipos de algoritmos de aprendizaje requieren de un número de muestras suficiente para definir las fronteras de decisión con relativa precisión. Por tanto, es necesario un conjunto de entrenamiento con un tamaño acorde a la complejidad inherente del problema para generalizar las fronteras y no tener el caso contrario conocido como *underfitting*. Además, en problemas de desbalance entre clases deben definirse métricas y procedimientos apropiados que tengan en cuenta el sesgo a favor de las clases mayoritarias y no "camuflen" los resultados obtenidos en las clases minoritarias.

3.3. Descripción de los experimentos

3.3.1. Estimación del número de clústeres

Ya definidos anteriormente los detectores de características y sus respectivas configuraciones con las que se desea experimentar, partimos desde el inicio del sistema

de aprendizaje y clasificación con los descriptores SIFT de los puntos de detectados en cada imagen. Por tanto, para un ejemplo o imagen dada, tenemos un conjunto asociado de descripciones de las regiones de los puntos detectados de longitud $d \times 128$, siendo d el número de puntos detectados. En la Tabla 3.3 se presenta de forma resumida la categoría o tipo de característica, cantidad promedio y desviación de los puntos que son detectados en las imágenes de glóbulos blancos de nuestra base de datos bajo la configuración asignada a cada uno en los apartados anteriores.

Tabla 3.3: Resumen y comparación de los detectores de características empleados en este trabajo.

	Categoría				
	Disperso			**Denso**	
Detector	**SIFT**	**oFAST**	**CenSurE**	**dSIFT**	**PHOW**
Tipo de detección	región (*blob*)	borde (*edge*)	región (*blob*)	denso	denso
# puntos promedio	426	192	161	436	601
Desviación estándar	209	31	115	-	-

Se desea conocer de antemano el vocabulario o número de "palabras" aproximadas que definen de la mejor manera las particularidades del problema, es decir, la mejor configuración de k clústeres en el proceso de *clustering*. La estimación y elección de este parámetro tienen un impacto directo en el resto de fases del proceso y, por consiguiente, en el modelo y precisión final del algoritmo de aprendizaje empleado. En nuestro caso, se hace uso de SVM.

Para estimar el parámetro k se decide realizar un barrido de éste en un rango de valores discretos. Para cada valor de k se evalúa el proceso completo con el mismo conjunto de datos por medio del método de validación *holdout*. La partición se realiza destinando un 80 % al conjunto de entrenamiento y un 20 % al conjunto de validación. Esta partición se mantiene para todos los valores de k evaluados.

Para realizar esta experimentación se escogen únicamente las dos clases mayoritarias, es decir, los "neutrofilos" y "linfocitos". Ambas clases corresponden en conjunto al 75 % (987 de 1315) del total de la base de datos. Esta elección pretende buscar la inexistencia de *underfitting* en la fase de algoritmo de aprendizaje, puesto que el objetivo de este experimento se centra más en la parte anterior: *clustering* y cuantificación y extracción de histogramas.

La búsqueda de valores realizada en la búsqueda exhaustiva interna de SVM se constituye a través de un rango de valores determinado para cada hiperparámetro y para cada *kernel*, barajando todas las combinaciones posibles entre ellas para un *kernel* dado. En el caso del *kernel* lineal, se busca optimizar el rendimiento de la clasificación respecto al parámetro C que toma los siguientes valores: $C \in \{10^i : i \in \{-5, -4, \ldots, -1, 1, 2, \ldots, 5\}\}$. En el caso del *kernel* RBF se barre el rango de C anterior junto al hiperparámetro γ con el siguiente rango: $\gamma \in \{10^i : i \in \{3, 5, 7\}\}$.

Entre todos los modelos formados a través de los dos *kernels* definidos y su correspondiente barrido de parámetros se escoge aquel que maximice una función de

score en la clasificación. Esta función de *score* que se desea optimizar es "precision", definida en el apartado 3.3.4.

Lo que se pretende evaluar al final del proceso es la tasa de acierto para cada posible valor de k con el fin de elegir y fijar el más conveniente para el resto de los experimentos posteriores a realizar. El mismo experimento se replica para las descripciones obtenidas desde los distintos extractores de características definidos en el capítulo anterior: PHOW, SIFT denso (dSIFT), SIFT, oFAST y CenSurE. En la misma medida que podemos observar la tendencia en la tasa de acierto para un determinado número de k clústeres, también podemos comparar el comportamiento entre los distintos extractores y obtener unas conclusiones preliminares. Sabiendo que el proceso de *clustering* es más costoso a medida que aumentamos el número de clústeres con un nivel elevado de muestras, se realiza un barrido en dos escalas distintas: una más lineal y acotada a un valor máximo de 90; la otra, de tendencia más exponencial y con un valor máximo de 1000. El objetivo es analizar el rendimiento para dos niveles de detalle:

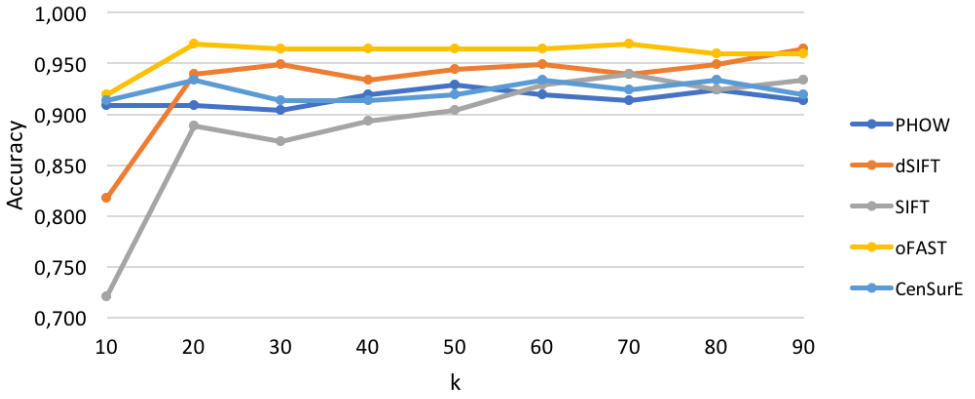
- **Primer barrido**: $k \in \{10, 20, 30, 40, 50, 60, 70, 80, 90\}$.
- **Segundo barrido**: $k \in \{10, 20, 40, 80, 150, 300, 500, 1000\}$.

Los resultados de esta experimentación se observan en la Figura 3.1 para los dos rangos de k propuestos. Se observa una tendencia general creciente en la tasa de acierto a medida que aumentamos k. Para todos los detectores de características, esta mejora es más notable entre saltos cuando el valor de k es bajo. A partir de un valor 80, prácticamente la tasa de acierto en todos los detectores se estabiliza y converge, haciendo que la mejora respecto a un valor superior sea más pequeña. Un caso excepcional ocupa el extractor PHOW, pues la tasa de acierto entre valores de k sucesivos se mantiene constante desde un principio. CenSurE tampoco consigue mejorar demasiado la tasa de acierto en los rangos de k definidos. oFAST parece mostrar el mejor rendimiento entre todos los extractores de características evaluados, manteniendo una tasa de acierto superior a 0,9 en todos los valores de k.

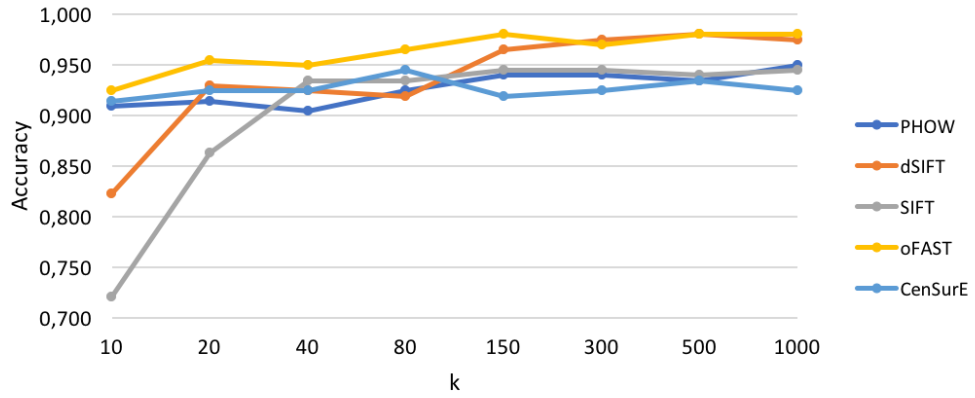
A vista de los resultados obtenidos, se decide tomar un valor fijo de k de 500 para los experimentos posteriores, un valor de compromiso entre precisión en la clasificación y coste computacional.

3.3.2. Estudio del rendimiento de dSIFT

Otro aspecto que ocupa la atención de este trabajo es la búsqueda de la mejor configuración en términos de rendimiento, estrategia de muestreo de puntos en función de la localidad de la imagen y cantidad de puntos en los algoritmos de detección densa, especialmente el caso del algoritmo dSIFT. Visto en su respectivo apartado las distintas opciones que se han formulado para definir la malla de puntos, el objetivo de este apartado es validar las hipótesis que se han realizado anteriormente:



(a) Primer barrido del rango de valores de k.



(b) Segundo barrido del rango de valores de k.

Figura 3.1: Rendimiento de la clasificación en términos de tasa de acierto entre los detectores de características definidos para las clases mayoritarias frente al tamaño del diccionario, k.

importancia de la densidad de puntos en el glóbulo de interés, cantidad de puntos global y necesidad de no desestimar la información de contexto.

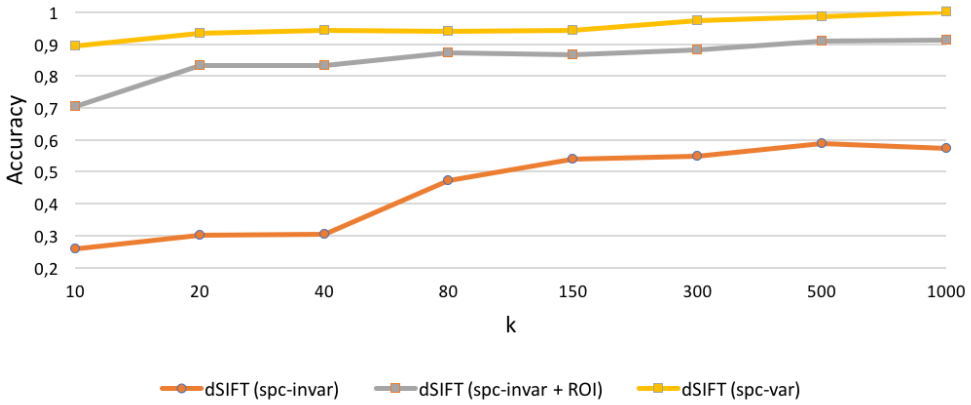
Para llevarlo a cabo se propone un experimento sencillo que nos permita dilucidar, entre las distintas formas propuestas de definir la malla de puntos, aquella más conveniente para la problemática dada. De este estudio se pueden extraer conclusiones muy importantes respecto a las hipótesis que se habían realizado previamente. La mejor configuración entre las propuestas formará parte de la experimentación presentada posteriormente. Se propone comparar las tres formas de definir las mallas vistas en la Figura 2.7 del capítulo anterior junto con la configuración que aparece en ésta en cuanto a distribución de puntos. Estas formas de definir la malla son, junto a su abreviatura:

- Malla con invariancia espacial ("spc-invar").
- Malla con invariancia espacial en una región de interés ("spc-invar + ROI").
- Malla con variancia espacial ("spc-var").

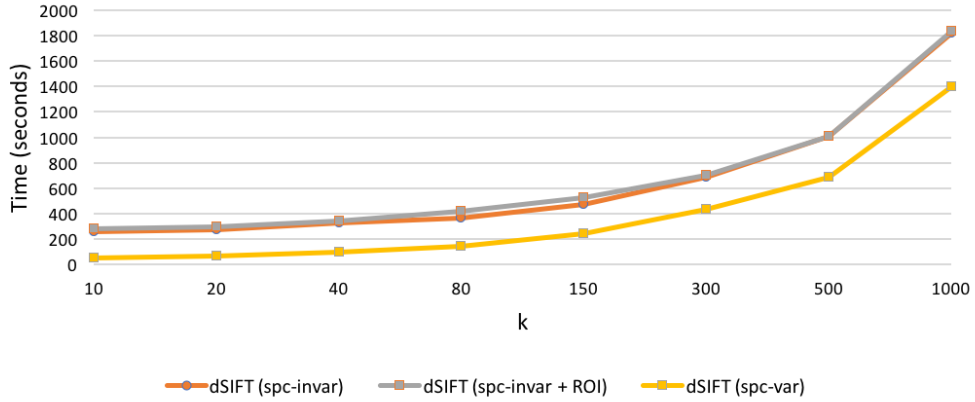
Para ello, se desea comparar la afección en la tasa de acierto y el tiempo de cómputo del esquema de clasificación empleando de nuevo las dos clases mayoritarias ("neutrofilos" y "linfocitos") respecto al recorrido del parámetro k del segundo barrido realizado en la anterior sección.

En la Figura 3.2 se puede analizar la diferencia que existe entre estas configuraciones. Respecto la Figura 3.2a, es fácilmente apreciable la pobre tasa de acierto que se obtiene utilizando una malla fija, invariante a la localidad de la imagen y sin definición de una ROI. Esto nos lleva a pensar que la primera hipótesis podría ser cierta y es que, cuanta mayor densidad de puntos se describa en la región de interés, mejores son los resultados de la clasificación posteriormente. Por otro lado, también se observa que utilizando una rejilla variable (variante al espacio) se consiguen resultados excelentes en la clasificación. Por ejemplo, para $k = 1000$ la clasificación que se hace de las muestras es perfecta. Por este lado, podemos concluir que el fondo tiene una información contextual que resulta relevante siempre que no sobremuestremos esta parte de la imagen respecto a la zona central. Puesto que en esta región de la imagen, donde se localiza la mayor parte del glóbulo (región de interés), parece que contiene la información más relevante y resulta conveniente muestrear con mayor densidad dicha zona. Por ello, el muestreo espacio variante parece ser la mejor estrategia.

Respecto la Figura 3.2b, se observa el impacto que tiene en la clasificación aumentar la cantidad de puntos global de la rejilla. La versión de dSIFT variante a espacio consigue escalar mejor con los procesos que requieren mayor complejidad de cómputo: *clustering* con k-means y SVM. Estos métodos tienen en conjunto un coste temporal lineal a medida que incrementamos el valor del parámetro k de k-means. En las condiciones de ejecución del método, una diferencia de 200 puntos adicionales en la malla supone 7 minutos más de tiempo de ejecución con $k = 1000$. Además, este aumento en la cantidad de puntos no está justificado, ya que la mejor



(a) Tasa de acierto frente al tamaño del diccionario, k.



(b) Tiempo de cómputo frente al tamaño del diccionario, k.

Figura 3.2: Comparación de rendimiento entre las formas propuestas de definir la malla de puntos en dSIFT para las clases mayoritarias frente al tamaño del diccionario, k.

Tabla 3.4: Resumen comparativo de la experimentación con las propuestas de malla definidas en dSIFT (se muestran los valores promedio).

	spc-invar	**spc-invar + ROI**	**spc-var**
Tasa de acierto	0,45	0,85	0,95
Tiempo (segundos)	650,52	676,79	389,83
# puntos definidos	676	676	436

configuración que obtiene los mejores resultados es aquella con menor cantidad de puntos (dSIFT con variancia espacial), una vez alcanzado un número suficiente de puntos para caracterizar la región. El factor más determinante que proviene de las conclusiones extraídas en este estudio es la localidad estratégica de los puntos en los algoritmos de detección densa. En la Tabla 3.4 se resumen el resultado promedio de la experimentación previamente realizada.

3.3.3. Elección del umbral de SIFT

El objeto de este apartado es evaluar el comportamiento de la *Visual* BoW al hacer variar un valor de umbral en la detección de puntos mediante el algoritmo SIFT. Este valor de umbral controla la selección de extremos locales en la búsqueda de máximos en la pirámide gaussiana, $|D(x, y, \sigma)|$, y se conoce también como umbral de intensidad. A medida que este valor decrece, se tolera una mayor cantidad de puntos que tienen una respuesta suficiente para ser detectados como máximos en las regiones de búsqueda de la pirámide gaussiana.

Puesto que es un parámetro que permite regular la cantidad de puntos detectados sin afectar a la ubicación de los puntos restantes, se pretende en esta experimentación evaluar si existe una dependencia entre esta cantidad y la robustez y rendimiento del esquema de clasificación. Este punto de vista es interesante para conocer de nuevo si este factor es determinante en el método escogido en general o, en caso contrario, no es suficientemente relevante respecto a la calidad de otro tipo de factores (ubicación de los puntos, tipo de característica, valor de k, algoritmo de aprendizaje escogido, etc.).

La experimentación aplicada consiste en evaluar la tasa de acierto en la clasificación y la cantidad de puntos detectada, así como la variación o desviación típica de puntos detectados entre las imágenes del conjunto de datos. De nuevo se emplean las dos clases mayoritarias ("neutrofilos" y "linfocitos") y se realiza un recorrido del parámetro del tamaño del diccionario, k, del segundo barrido utilizado en el anterior estudio de dSIFT.

Se desea evaluar un cierto rango de valores del umbral de intensidad manteniendo el resto de la configuración definida en la sección 2.2.1. Se dispone un total de 4 valores, $\{0, 5, 1, 5, 8\}$, un rango representativo de aquellos valores que son apropiados para la detección y fases posteriores. Esto quiere decir que, para valores inferiores

de 0,5 en el umbral de intensidad, se detecta una enorme cantidad de puntos y, por tanto, los recursos *hardware* en nuestro caso no son suficientes para gestionar en memoria todos los descriptores de los puntos de las imágenes en las clases seleccionadas. Este fenómeno marca una primera cota en las limitaciones para efectuar experimentos: los datos de los conjuntos de entrenamiento y validación formados durante la fase de formación deben ser inferiores respecto a la capacidad de la memoria volátil disponible para efectuar correctamente las pruebas. Por otro lado, valores de umbral superiores a 8 impiden la detección de suficientes puntos en algunas imágenes y se tiene, en consecuencia, algunas instancias vacías. Esta situación tampoco permite la ejecución del esquema de clasificación en las fases posteriores porque es equivalente a submuestrar de forma aleatoria los conjuntos de entrenamiento y validación respecto al resto de experimentos.

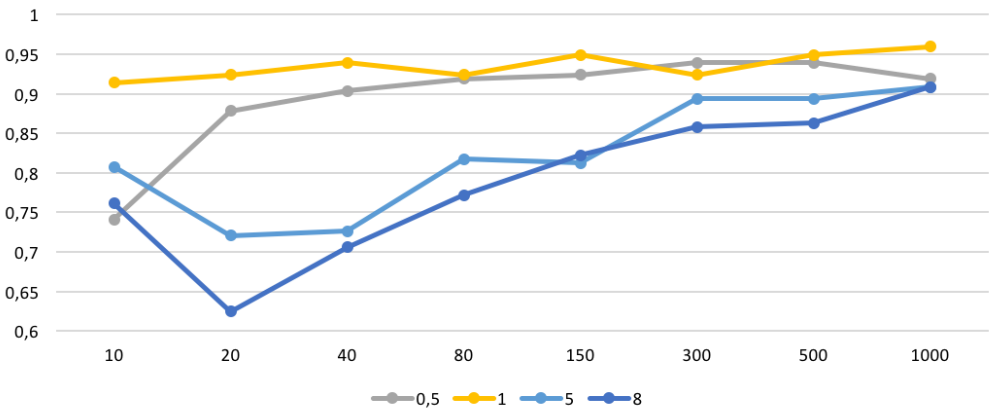
En la Figura 3.3 se detalla el resumen de la experimentación. Por un lado, en la figura superior se puede estimar el comportamiento en la clasificación para cada uno de los casos de detección de puntos con un determinado valor de umbral de intensidad. Los resultados muestran que disponer de un determinado número de puntos medios detectados por imagen es importante, tal como apuntan los dos valores de umbral más pequeños. Los umbrales de 0,5 y 1 consiguen mantener una mejor y más regular tasa de acierto en torno a diferentes tamaños de diccionario. El resto de casos se resienten más con la falta de definición de un tamaño de diccionario suficientemente grande, como se puede observar en sus tendencias crecientes e inferiores a los umbrales de 0,5 y 1.

Fijándonos en la figura inferior, si nos centramos de nuevo en los umbrales de 0,5 y 1, vemos la diferencia en la cantidad media de puntos entre un caso y otro. Un umbral de 1, a pesar de detectar aproximadamente la mitad de puntos medio por imagen respecto al umbral de 0,5, mantiene un comportamiento un poco más estable en los distintos tamaños de diccionario evaluados. Esta elección apunta a una mejor robustez general, puesto que su rendimiento es más independiente del valor de k.

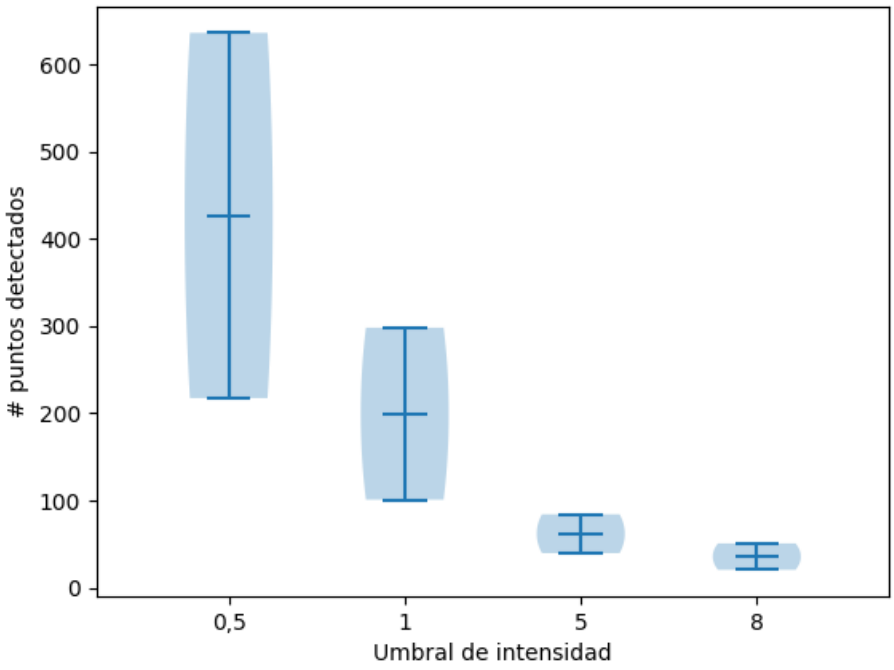
Junto al estudio realizado de dSIFT, se afirma que el número de puntos detectada por imagen es importante en términos generales. Tiene que ser suficientemente grande para obtener un rendimiento aceptable, pero el aumento de esta cantidad no va unido con una creciente mejora en la tasa de acierto del clasificador. Por tanto, este factor puede situarse en un punto medio entre rendimiento y exigencia computacional.

3.3.4. Comparación del rendimiento entre los distintos extractores de características

Una vez fijados y establecidos los parámetros requeridos en la *Visual* BoW, se pretende diseñar un marco de métricas y procedimiento de validación apropiado para la base de datos íntegra, donde se tiene en cuenta todas las clases de glóbulos blancos disponibles vistos en el apartado 3.2.



(a) Tasa de acierto frente al tamaño del diccionario, k.



(b) Número de puntos medio detectado por imagen y su desviación típica frente al tamaño del diccionario, k.

Figura 3.3: Comparación de rendimiento con varios valores de umbral de intensidad de SIFT para las clases mayoritarias frente al tamaño del diccionario, k.

Esquema de validación

Anteriormente se empleaba un método no exhaustivo en la partición de los datos, conocido como *holdout*. Para la extracción de resultados conclusivos no es suficiente, pues se requiere de mayor rigor en el análisis estadístico que minimice la dependencia en la elección previa de los respectivos conjuntos de entrenamiento y test.

Por un lado, el método *holdout* solo evalúa una de las posibles formas de separar los conjuntos de entrenamiento y, aunque esta asignación suele ser aleatoria, puede darse el caso que la partición realizada incluya ejemplos más "sencillos" en el proceso de entrenamiento de SVM y, por consiguiente, se generalice mejor las fronteras de decisión. Podría darse también el caso contrario y obtener resultados variables entre experimentos.

Por tanto, con el fin de mitigar tal efecto en la dependencia de la elección de los conjuntos de datos, se decide emplear un esquema de *k-fold cross-validation*. Este método de validación consiste en subdividir de forma aleatoria el conjunto original de datos en k subconjuntos de igual tamaño. Uno subconjunto se emplea como conjunto de validación y los $k-1$ subconjuntos restantes se emplean como conjunto de entrenamiento. Este proceso se realiza k veces, con el fin de evaluar cada uno de los posibles subconjuntos de prueba. Con este método podemos extraer propiedades estadísticas a partir de los resultados de las distintas iteraciones, como la media o la varianza. El número de *folds* elegido es 5 para todos los experimentos.

La varianza permite hablarnos de la robustez del clasificador, pues variaciones pequeñas entre iteraciones pueden significar un grado de independencia en la elección de los conjuntos de datos y, por tanto, mayor facilidad para encontrar las generalidades del problema.

Respecto al esquema anterior, se desea contemplar la misma proporción de ejemplos de cada clase en los respectivos subconjuntos (entrenamiento y validación), ya sean clases mayoritarias o minoritarias. Este método preserva una representación justa y equitativa de cada clase y se conoce como un esquema de validación *stratified k-fold cross-validation*.

Métricas de evaluación

A continuación, se definen el conjunto de métricas de rendimiento que resumen la clasificación obtenida en todo el proceso que abarca la *Visual* BoW, desde la extracción y descripción de puntos en las imágenes hasta la formación del modelo SVM por medio de los histogramas de "palabras". Teniendo en cuenta el desbalance observado entre clases y la presencia de un paradigma multiclase (más de 2 clases), se sugieren las siguientes métricas:

- **"Accuracy"**: es la medida de rendimiento más intuitiva. Simplemente es una relación entre las observaciones categorizadas como correctamente predichas,

verdaderos positivos y negativos (t_p y t_n, respectivamente), con respecto a las observaciones totales, donde se incluyen también las categorías de falsos positivos o f_p (en un caso binario es la situación consistente en predecir la etiqueta real cuando no está presente la etiqueta real) y los falsos negativos o f_n (en un caso binario es la situación consistente en no predecir la etiqueta real cuando sí está presente la etiqueta real). Puede tomar valores entre 0 y 1 (a mayor valor, mejor será el modelo). Esta métrica se define como:

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

Es una métrica sencilla y precisa de determinar la bondad de un clasificador, pero solo en el caso que las clases estén uniformemente distribuidas, pues solo tiene en cuenta el total. En el caso de desbalance entre clases debe complementarse con otro tipo de métricas que ponderen de igual forma los verdaderos y falsos positivos y negativos de todas las clases.

- **Coeficiente de correlación de Matthews** (*Matthews Correlation Coefficient*, MCC): se utiliza en aprendizaje automático como una medida de la calidad de las clasificaciones basadas en dos clases, introducida por el bioquímico Brian W. Matthews en 1975. Tiene en cuenta los verdaderos y falsos positivos y negativos por separado, y se considera generalmente como una medida equilibrada utilizable en casos de desbalance entre clases. Como su nombre indica es una medida de correlación que puede tomar valores entre -1 y $+1$, siendo un coeficiente de $+1$ una representación de predicción perfecta; 0, una predicción aleatoria; y -1, una discrepancia completa entre predicción y etiqueta real. MCC se define de la siguiente forma para el caso binario:

$$MCC = \frac{t_p \cdot t_n - f_p \cdot f_n}{\sqrt{(t_p + f_p)(t_p + f_n)(t_n + f_p)(t_n + f_n)}}$$

Inicialmente propuesto para problemas binarios con desbalance, MCC se puede generalizar para un problema multiclase, aunque el rango de los posibles valores del coeficiente pasa a situarse desde un valor mínimo entre -1 y 0 (dependiendo de la distribución) y un valor máximo de $+1$, siendo éste último valor de nuevo una representación de predicción perfecta. Esta generalización recibe por su autor el nombre de estadístico R_K (K clases distintas) o generalización discreta del coeficiente de correlación de Pearson [16]. Es la métrica utilizada en nuestros experimentos. Se define en términos de una matriz C de $K \times K$ dimensiones:

$$R_K = \frac{\sum_{klm} C_{kk} C_{lm} - C_{kl} C_{mk}}{\sqrt{\sum_k \left(\sum_l C_{kl} \right) \left(\sum_{\substack{l' \\ k' \neq k}} C_{k'l'} \right)} \sqrt{\sum_k \left(\sum_l C_{lk} \right) \left(\sum_{\substack{l' \\ k' \neq k}} C_{l'k'} \right)}},$$

donde C_{kl} son los elementos de la matriz de confusión.

- **"Precision"**: es la relación entre las observaciones positivas correctamente predichas y las observaciones positivas totales. Esta métrica se define como:

$$Precision = \frac{t_p}{t_p + f_p}$$

  Con alta "precision" nos referimos a una baja tasa de falsos positivos.

- **"Recall"**: es la relación entre las observaciones positivas correctamente predichas y las observaciones totales de la clase actual. Esta métrica se define como:

$$Recall = \frac{t_p}{t_p + f_n}$$

  Alto "recall" se traduce como una baja tasa de falsos negativos.

- **F1-score**: puede interpretarse como una media ponderada de "precision" y "recall", donde se alcanza el mejor valor en 1 y el peor en 0. La contribución relativa de "precision" y "recall" al F1-score son iguales. Se define de la siguiente manera:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

  Intuitivamente es más difícil de entender que "accuracy", pero es más útil que la anterior en el caso de desbalance entre clases.

- **Matriz de confusión**: es una tabla que resume de forma global el rendimiento de un algoritmo de aprendizaje supervisado. Para cada clase, podemos observar las predicciones realizadas respecto al resto de clases en términos de verdaderos y falsos positivos y negativos.

- ***Boxplot***: también conocido como diagrama de caja y bigotes, es un gráfico que está basado en cuartiles y mediante el cual se visualiza la distribución de un conjunto de datos. Está compuesto por un rectángulo (la "caja") y dos brazos (los "bigotes"). Suministra información sobre los valores mínimo y máximo, los cuartiles Q1, Q2 o mediana y Q3, y sobre la existencia de valores atípicos (*outliers*) y la simetría de la distribución.

Realmente todas las medidas sugeridas anteriormente se definen originalmente para un problema de clases binario, aunque es posible calcularlas en un problema multiclase por medio de métodos de promediado. El método empleado es el "ponderado", similar al "macro-average", que consiste en calcular métricas para cada etiqueta y encontrar el promedio entre ellas ponderando el número de instancias verdaderas para cada etiqueta. Este método es el que mejor tiene en cuenta el desbalance entre clases.

Junto a estas métricas, se añade el coste temporal promedio para la *Visual* BoW desde el punto de vista de cada detector de características. Este tiempo tiene en cuenta el proceso que se inicia desde el *clustering* hasta la clasificación por SVM incluida, es decir, no tiene en cuenta el tiempo de extracción y descripción de los puntos en las imágenes.

3.4. Discusión de resultados

Finalmente, vista la elección de parámetros y diseño de la experimentación a realizar en cuanto a métricas representadas en la evaluación de la clasificación, en la Figura 3.4 se resume y detalla en formato *boxplot* los resultados de clasificación de la *Visual* BoW para los distintos extractores de características en términos de las métricas comentadas anteriormente.

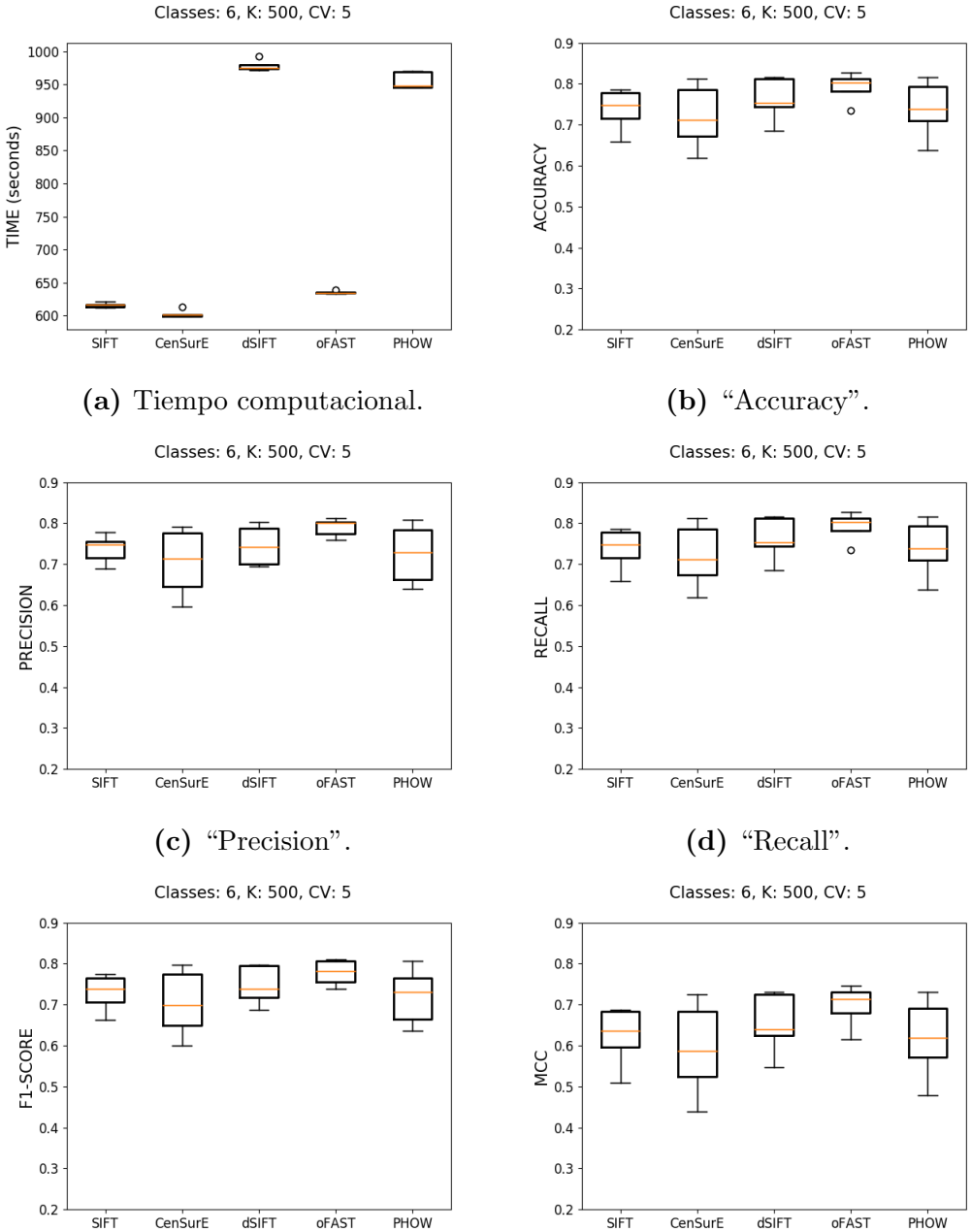
Debido a que previamente se había escogido una configuración *stratified 5-fold cross-validation*, en cada uno de los 5 splits hay una representación proporcional de muestras de cada clase respecto a su total. Por ejemplo, si de cada *split* se decide destinar un 80 % de muestras al conjunto de entrenamiento, entonces se dispone en dicho conjunto un 0,8 muestras de cada una de las clases disponibles en el *split*. Sumando estas representaciones por separado se obtiene exactamente el 80 % del total de muestras del *split* original. Mediante este esquema favorecemos que las clases minoritarias tengan su representación proporcional en cada conjunto creado.

Tras el proceso completo de validación disponemos de 5 medidas distintas para cada métrica y el gráfico *boxplot* es una excelente herramienta visual para capturar los estimadores estadísticos extraíbles a partir de estas medidas.

Desde un primer vistazo a las gráficas, cabe destacar el buen desempeño de oFAST en todas las métricas evaluadas, siendo este extractor el que obtiene valores más altos y menor varianza. Si esta observación la unimos a los hechos observados en el experimento de elección del valor k en la fase de *clustering*, concluimos que oFAST es el detector de características que mejor se adapta a la problemática del problema: mantiene su rendimiento independientemente de los conjuntos de entrenamiento y test elegidos, así como en la elección del número k de clústeres. Además, a excepción de PHOW y dSIFT, que tiene una configuración de puntos fija en todas las imágenes, oFAST es el extractor que mantiene mejor el número de puntos detectados entre todas las imágenes, a razón de la elección previa del número N de puntos objetivo (si se detecta un mayor número de puntos en las imágenes, se realiza un *ranking* y se seleccionan aquellos que tengan una respuesta mayor). En tiempo computacional, compite muy bien con los mejores en este aspecto, que son CenSurE y SIFT. Se observa en este aspecto el coste que implica aumentar el número de puntos, como el caso de dSIFT y PHOW, los cuales tienen tiempos promedio de ejecución similares de la *Visual* BoW y suponen una variación aproximada del 60 % en tiempo respecto al resto de detectores no densos.

También es observable la correlación relativa entre las distintas métricas para cada uno de los extractores, pues la tendencia y ubicación de cada uno de los extractores se mantiene entre estas medidas.

No obstante, omitiendo el caso de la desviación estándar, las medias obtenidas entre los distintos extractores para cada una de las métricas son muy similares, puesto que no difieren de 0,1 en ningún caso. Este fenómeno permite afirmar que tampoco hay una correlación clara entre la cantidad de puntos y la tasa de acierto.



(a) Tiempo computacional. **(b)** "Accuracy".

(c) "Precision". **(d)** "Recall".

(e) F1-score. **(f)** MCC.

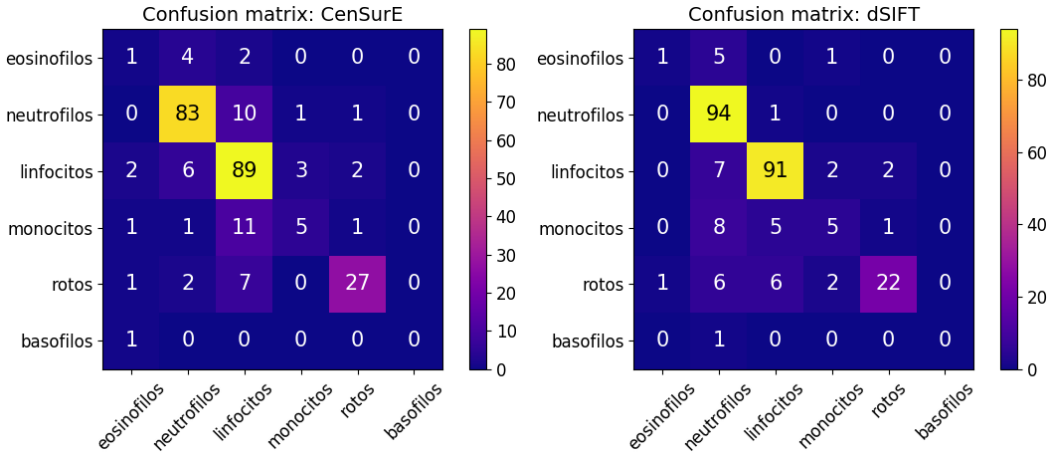
Figura 3.4: Comparativa del rendimiento de clasificación de la *Visual* BoW con los extractores de características propuestos.

La diferencia observada en la desviación estándar más acusada en el caso de CenSurE se relaciona más con la ubicación de los puntos, pues como se ha visto en el capítulo anterior, CenSurE detecta puntos en *blobs* en las formas presentes en las imágenes. En cambio, oFAST localiza los puntos, generalmente, en los bordes o contornos de las células (*edges*). Por tanto, se puede concluir que el factor relevante que afecta a la robustez y rendimiento del proceso de la *Visual* BoW se relaciona con el tipo de característica del detector empleado.

Como complemento a las gráficas anteriores, en la Figura 3.5 podemos observar las matrices de confusión extraídas del primer *split* seleccionado de cada uno de los extractores. Por tanto, se comparan los distintos extractores con los mismos conjuntos de entrenamiento y validación. En las respectivas matrices de confusión podemos observar tanto el rendimiento general de cada una de las propuestas como la predicción realizada en una clase o etiqueta particular. Por ejemplo, SIFT no es el detector que consigue el mayor número de verdaderos positivos en las clases mayoritarias, pero obtiene una buena tasa de acierto con la clase "rotos".

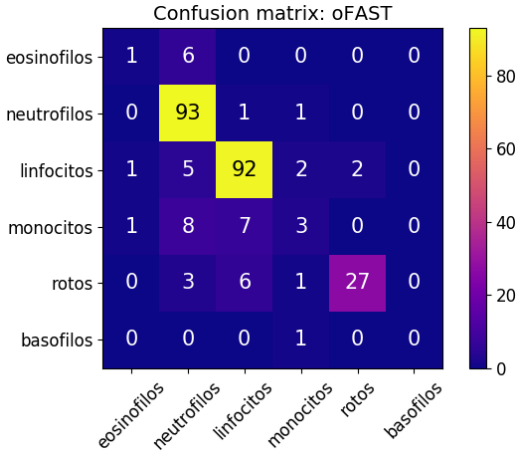
El caso contrario sucede con dSIFT, donde las tasas de acierto son superiores en las clases mayoritarias respecto a las clases minoritarias. Sin embargo, en las matrices de confusión no se refleja la varianza entre distintos *splits*. Por esa razón, esta información debe complementarse con la anterior.

Debido al desbalance entre clases tampoco se pueden hacer asunciones importantes sobre el comportamiento de las clases minoritarias, como es el caso de los "eosinofilos" y "basofilos", ya que su número es tan bajo en el conjunto de validación que no se puede afirmar con certeza las posibles causas en la tasa de acierto tan baja que existe en estas clases. La clase "monocitos" constituye un 7 % del total de la base de datos y es curioso observar como el clasificador confunde esta clase más fácilmente con las clases "neutrofilos" y "linfocitos" que con la clase verdadera. Se entiende que el número no es suficiente para separar con exactitud las fronteras de decisión de la clase "monocitos" con el resto de clases.

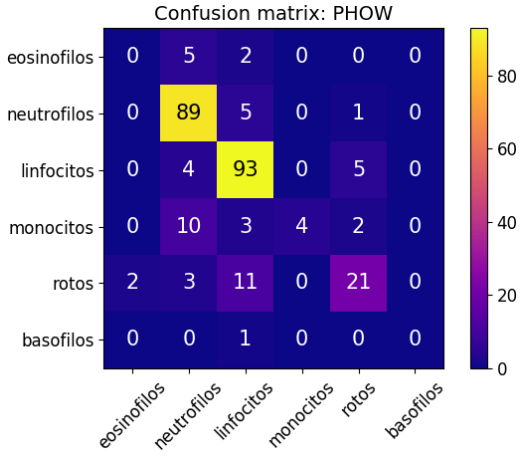


(a) CenSurE

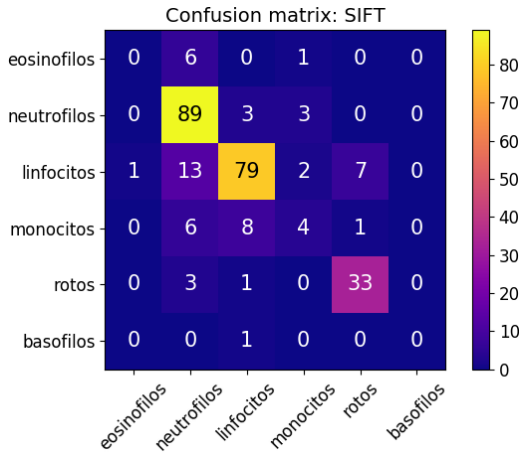
(b) dSIFT



(c) oFAST



(d) PHOW



(e) SIFT

Figura 3.5: Matrices de confusión sin normalizar de la *Visual* BoW para cada uno de los extractores de características (las filas son las etiquetas reales y las columnas las etiquetas predichas).

Capítulo 4

Conclusiones

En este trabajo se ha realizado una comparación de distintos detectores de puntos de interés locales en un procedimiento o esquema de clasificación automático denominado *Bolsa de Palabras Visuales*. Este esquema es un método semi-automático que se encuentra en un escalón intermedio entre procedimientos sin extracción previa de características, como es el caso de la actual tendencia del *deep learning* en tareas de visión por ordenador y los procedimientos más tradicionales mediante técnicas de procesamiento digital de imagen. La *Visual* BoW solo requiere de la introducción al sistema de un conjunto de descriptores de puntos o regiones de la imagen y la sintonización de parámetros relacionados con los algoritmos de aprendizaje supervisado (en nuestro caso, SVM) y no supervisado (*clustering*). Por este motivo, el procedimiento se puede descomponer en fases y esta modularidad se da libertad para experimentar distintos algoritmos de aprendizaje y detectores de características.

De la experimentación realizada podemos extraer las siguientes conclusiones relevantes:

- Respecto al tópico de estudio de este trabajo, el factor más importante que afecta directamente a la robustez y rendimiento del proceso se centraba concretamente en la elección del tipo de característica detectada. Entre los posibles tipos que podemos encontrar en la literatura, la característica de interés tipo *edge*, tratada en este trabajo por el detector oFAST, parece ser que es la que mejor describe la información contenida en las imágenes de glóbulos blancos.
- En el caso de la *Visual* BoW, la cantidad de puntos detectados del conjunto de descriptores de cada imagen no parece ser un factor determinante. Gracias a esto, no es necesario realizar esfuerzos en mejorar y enfocarse en este aspecto, pues, como se ha visto, este proceso no escala bien debido a la complejidad computacional en la fase de *clustering*.
- Al centrarnos en la elección del parámetro k en el *clustering*, se realizaba una evaluación del rendimiento haciendo variar este valor y observando el comportamiento del resto del procedimiento a través de la tasa de clasificación. De este apartado concluíamos que al aumentar k, se tendía a mejorar en cualquier

caso la tasa de acierto, pero se elevaba el coste computacional. Un valor k de 500 conseguía marcar una situación de compromiso entre el coste y el rendimiento. Seguir aumentando este valor no mejoraba de forma significativa la mayoría de extractores analizados.

- Otro aspecto a considerar es la naturaleza de la base de datos. En este caso, la base de datos muestra un problema de desbalance claro, donde de las 6 clases disponibles 2 de ellas ocupan el 75 % de muestras en total en combinación, mientras que otras 2 de ellas solo alcanzan el 4 % del total. Debido a esto, las clases mayoritarias salen especialmente beneficiadas en el proceso de entrenamiento, pues se tiene un número aceptable y suficiente. Esto se refleja en los buenos resultados obtenidos en las clases "linfocitos" y "neutrofilos", mientras que, en las clases minoritarias, su número es muy bajo y se disponen de pocas muestras para entrenar y para validar. Por tanto, en estas clases el rendimiento y la tasa de clasificación es pobre.

- Entre los detectores densos, dSIFT y PHOW, se benefician de estar definidos con resolución variable en el espacio. Se consigue reducir la cantidad de puntos definidos por imagen y se mejora el rendimiento del clasificador respecto a la versión de mallado uniforme de puntos. Ambos son muy similares en cuanto a rendimiento, aunque PHOW obtiene un mejor desempeño general, pues reduce ligeramente la tasa de falsos positivos respecto a dSIFT.

- En el estudio de las diferentes estrategias para definir los puntos de dSIFT concluíamos que la información relevante se concentra en torno a la localidad situada en el glóbulo de interés, razón por la que es importante definir una mayor densidad de puntos en esta región.

- Por último, comentar que la información de contexto tiene cierto peso en el rendimiento de la clasificación, debido a la obtención de mejores resultados realizando un muestreo de puntos espacio variante.

A vista de los resultados, el proceso *Visual* BoW parece ser un enfoque oportuno para la tarea de clasificación de glóbulos blancos a través de los extractores de características analizados. Al realizar esta comparación se han extraído importantes conclusiones para definir una hoja de ruta en la configuración de futuros experimentos y mejoras de este proceso.

4.1. Trabajo futuro

En este trabajo nos hemos centrado en cinco extractores de características conocidos en la literatura junto a una descripción de las regiones adyacentes a los puntos detectados mediante SIFT en todos ellos. Dependiendo la fase del esquema en el que nos queramos centrar, con el fin de mejorar los resultados obtenidos, se abre un abanico de posibilidades muy grande en cuanto a la cantidad de técnicas y perspectivas en el enfoque del problema.

Volviendo de nuevo a la fase de extracción de características, se pueden probar otro tipo de extractores o descriptores no explorados, como es el caso de aquellos que se enfocan en las características tipo textura, por ejemplo, *Local Binary Patterns* (LBP), y comprobar la diferencia respecto a la descripción SIFT.

Una idea muy interesante es explorar la información que puede aportar el color, ya que hasta este momento se ha realizado la detección de puntos de todos los algoritmos presentados en niveles de intensidad. Esta nueva información puede ser muy importante, como mencionan algunos autores en el actual estado del arte. Para hacerlo es posible utilizar métodos de fusión. Estos mecanismos se diferencian entre sí dependiendo de la fase del esquema de clasificación que se desea concatenar información. Por ejemplo, hacer uso de un esquema *early-fusion* consistiría en concatenar las descripciones realizadas de la detección de los puntos en tres canales RGB de la imagen, en vez de hacer una única descripción en niveles de gris. Mediante este método, se aumenta en tres la dimensionalidad anterior y, en consecuencia, el coste computacional de todo el proceso, pero se tiene en cuenta nueva información que podría mejorar el rendimiento general del clasificador. Esta fusión también se puede realizar en fases posteriores del esquema de la *Visual* BoW por medio de un método *late-fusion*, por ejemplo, combinando histogramas en el proceso de cuantificación procedentes de la descripción de cada canal.

Otra idea distinta sería combinar las descripciones de distintos detectores o clasificadores. Cada uno por separado tienen propiedades distintas y complementarias. Sería interesante realizar una combinación de ellas, de nuevo por medio de algún método de fusión. Por ejemplo, con un método *late-fusion* se pueden formar varios "vocabularios" desde descripciones con detectores distintos y concatenar los histogramas resultantes previamente a la fase de entrenamiento mediante un algoritmo de aprendizaje.

Respecto a SVM, su rendimiento es bueno, aunque en el proceso de creación del modelo se requiere de ajustar los hiperparámetros y la elección de un *kernel* mediante una búsqueda por rejilla exhaustiva. Otra opción consiste en medir las ventajas e inconvenientes de otros algoritmos de aprendizaje de categorías distintas en esta fase de entrenamiento: probabilísticos (por ejemplo, *redes Bayesianas*), no paramétricos (por ejemplo, *k-NN*), redes neuronales, etc. También se podría combinar distintos clasificadores por medio de la suma ponderada de los *scores* de los clasificadores individuales y mejorar los resultados, idea similar a lo que se realiza con los algoritmos de *Boosting*.

Todas estas ideas se centran en intentar mejorar la tasa de acierto y la robustez del esquema de clasificación, aunque siempre está la posibilidad de hacer esfuerzos en optimizar otro tipo de factores: requerimientos de memoria, coste computacional, dimensionalidad, etc.

No obstante, el problema de interés principal encontrado es el desbalanceo entre clases desde el punto de vista del clasificador. Este tema debe tratarse con mayor prioridad para mejorar sustancialmente las diferencias en tasas de precisión entre clases mayoritarias y minoritarias. Para ello, una opción es flexibilizar el procedimiento

de clasificación automático descrito en este trabajo para implementar un algoritmo adaptativo que pueda recibir retroalimentación a través de nuevas instancias o correcciones hechas a lo largo del tiempo. También se deja abierta la posibilidad de poder ampliar en un futuro la cantidad de los datos en las clases minoritarias para mejorar sustancialmente el rendimiento general del clasificador, ya sea por la adquisición de nuevas muestras de manera sintética o por su cesión a través de un hematólogo.

Bibliografía

[1] OpenCV-Python: Library of Python bindings designed to solve computer vision problems. https://opencv-python-tutroals.readthedocs.io/en/latest/. Accedido: 2017-09-12.

[2] Scikit-image: Image processing in Python. http://scikit-image.org/. Accedido: 2017-09-12.

[3] Scikit-learn: Machine learning in Python. http://scikit-learn.org/stable/. Accedido: 2017-09-12.

[4] SciPy: Python-based ecosystem of open-source software for mathematics, science, and engineering. https://www.scipy.org/. Accedido: 2017-09-12.

[5] VLFeat: Popular computer vision algorithms specializing in image understanding and local features extraction and matching. http://www.vlfeat.org/. Accedido: 2017-09-12.

[6] Agrawal, M., Konolige, K., and Blas, M. R. Censure: Center surround extremas for realtime feature detection and matching. In *European Conference on Computer Vision* (2008), Springer, pp. 102–115.

[7] Alahi, A., Ortiz, R., and Vandergheynst, P. Freak: Fast retina keypoint. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on* (2012), Ieee, pp. 510–517.

[8] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. *Molecular Biology of the Cell, Fourth Edition*, 4 ed. Garland Science, 2002.

[9] Barata, C., Figueiredo, M. A., Celebi, M. E., and Marques, J. S. Local features applied to dermoscopy images: Bag-of-features versus sparse coding. In *Iberian Conference on Pattern Recognition and Image Analysis* (2017), Springer, pp. 528–536.

[10] Bay, H., Tuytelaars, T., and Van Gool, L. Surf: Speeded up robust features. *Computer vision–ECCV 2006* (2006), 404–417.

[11] Bosch, A., Zisserman, A., and Munoz, X. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), IEEE, pp. 1–8.

[12] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010* (2010), 778–792.

[13] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV* (2004), vol. 1, Prague, pp. 1–2.

[14] de Jonge, R., Brouwer, R., van Rijn, M., van Acker, B. A., Otten, H. J., and Lindemans, J. Automated analysis of pleural fluid total and differential leukocyte counts with the sysmex xe-2100. *Clinical Chemical Laboratory Medicine 44*, 11 (2006), 1367–1371.

[15] Gómez-Gil, P., Ramírez-Cortés, M., González-Bernal, J., Pedrero, Á. G., Prieto-Castro, C. I., Valencia, D., Lobato, R., and Alonso, J. E. A feature extraction method based on morphological operators for automatic classification of leukocytes. In *Artificial Intelligence, 2008. MICAI'08. Seventh Mexican International Conference on* (2008), IEEE, pp. 227–232.

[16] Gorodkin, J. Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry 28*, 5 (2004), 367–374.

[17] Habibzadeh, M., Krzyżak, A., and Fevens, T. White blood cell differential counts using convolutional neural networks for low resolution images. In *International Conference on Artificial Intelligence and Soft Computing* (2013), Springer, pp. 263–274.

[18] Hiremath, P., Bannigidad, P., and Geeta, S. Automated identification and classification of white blood cells (leukocytes) in digital microscopic images. *IJCA special issue on "recent trends in image processing and pattern recognition" RTIPPR* (2010), 59–63.

[19] Hofmann, M. Support vector machines-kernels and the kernel trick. *An elaboration for the Hauptseminar Reading Club SVM* (2006).

[20] Jain, A. K. Data clustering: 50 years beyond k-means. *Pattern recognition letters 31*, 8 (2010), 651–666.

[21] Kamentsky, L. A. Cytology automation. *Adv Biol Med Phys 14*, 93 (1973), 1–1.

[22] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence 24*, 7 (2002), 881–892.

[23] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.

[24] Leutenegger, S., Chli, M., and Siegwart, R. Y. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 2548–2555.

[25] Lowe, D. G. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (1999), vol. 2, Ieee, pp. 1150–1157.

[26] Mikolajczyk, K., and Schmid, C. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence 27*, 10 (2005), 1615–1630.

[27] Mirčić, S., and Jorgovanović, N. Automatic classification of leukocytes. *Journal of automatic control 16*, 1 (2006), 29–32.

[28] Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2012.

[29] Parthasarathy, D. Classifying White Blood Cells With Deep Learning. https://blog.athelas.com/classifying-white-blood-cells-with-convolutional-neural-networks-2ca6da239331, 2017. Accedido: 2017-07-27.

[30] Piuri, V., and Scotti, F. Morphological classification of blood leucocytes by microscope images. In *Computational Intelligence for Measurement Systems and Applications, 2004. CIMSA. 2004 IEEE International Conference on* (2004), IEEE, pp. 103–108.

[31] Rosten, E., and Drummond, T. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 2, IEEE, pp. 1508–1515.

[32] Rosten, E., and Drummond, T. Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006* (2006), 430–443.

[33] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on* (2011), IEEE, pp. 2564–2571.

[34] Sabino, D. M. U., da Fontoura Costa, L., Rizzatti, E. G., and Zago, M. A. A texture approach to leukocyte recognition. *Real-Time Imaging 10*, 4 (2004), 205–216.

[35] Saladin, K. S., and Miller, L. *Anatomy & physiology*. WCB/McGraw-Hill New York (NY), 1998.

[36] Seguí, S., Drozdzal, M., Pascual, G., Radeva, P., Malagelada, C., Azpiroz, F., and Vitrià, J. Generic feature learning for wireless capsule endoscopy analysis. *Computers in biology and medicine 79* (2016), 163–172.

[37] Theera-Umpon, N., and Gader, P. D. System-level training of neural networks for counting white blood cells. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 32*, 1 (2002), 48–53.

[38] Tola, E., Lepetit, V., and Fua, P. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence 32*, 5 (2010), 815–830.

[39] Traver, V. J., Latorre-Carmona, P., Salvador-Balaguer, E., Pla, F., and Javidi, B. Human gesture recognition using three-dimensional integral imaging. *JOSA A 31*, 10 (2014), 2312–2320.

[40] Tuytelaars, T., Mikolajczyk, K., et al. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision 3*, 3 (2008), 177–280.

[41] Tycko, D., Anbalagan, S., Liu, H., and Ornstein, L. Automatic leukocyte classification using cytochemically stained smears. *Journal of Histochemistry & Cytochemistry 24*, 1 (1976), 178–194.

[42] Wallach, H. M. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning* (2006), ACM, pp. 977–984.

[43] Wang, J.-G., Li, J., Lee, C. Y., and Yau, W.-Y. Dense sift and gabor descriptors-based face representation with applications to gender recognition. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on* (2010), IEEE, pp. 1860–1864.