Recognizing white blood cells with local image descriptors

Dan López-Puigdollers[a,1], V. Javier Traver[a], Filiberto Pla[a]

[a] *Institute of New Imaging Technologies, Jaume-I University,*
Av. de Vicent Sos Baynat, s/n
12071 Castelló de la Plana, Spain
Phone: +34 964 38 7767, Fax: +34 964 38 7678

Abstract

Automatic and reliable classification of images of white blood cells is desirable for inexpensive, quick and accurate health diagnosis worldwide. In contrast to previous approaches which tend to rely on image segmentation and a careful choice of ad hoc (geometric) features, we explore the possibilities of local image descriptors, since they are a simple approachthey require no explicit segmentation, and yet they have been shown to be quite robust against background distraction in a number of visual tasks. Despite its potential, this methodology remains unexplored for this problem. In this work, images are therefore characterized with the well-known visual bag-of-words approach. Three keypoint detectors and five regular sampling strategies are studied and compared.

The results indicate that the approach is encouraging, and that both the sparse keypoint detectors and the dense regular sampling strategies can perform reasonably well (mean accuracies of about 80% are obtained), and are competitive to segmentation-based approaches. Two of the main findings are as follows. First, for sparse points, the detector which localizes keypoints on the cell contour (oFAST) performs somehow better than the other two (SIFT and CenSurE). Second, interestingly, and partly contrary to our expectations, the regular sampling strategies including hierarchical spatial information, multi-resolution encoding, or foveal-like sampling, clearly outperform the two simpler uniform-sampling strategies considered.

From the broader perspective of expert and intelligent systems, the relevance of the proposed approach is that, since it is very general and problem-agnostic, it makes unnecesary human expertise to be elicited in the form of explicit visual cues; only the labels of the cell type are required from human domain experts.

Key words: White blood cells recognition, local image descriptors, SIFT, interest point detectors, visual vocabulary

1. Introduction

In blood tests, the count of the different types of white blood cells provides a good quantitative account of a person's health state, which is crucial for prevention or cure purposes. Since huge number of blood tests are performed on a daily basis world-wide, automatic, fast and accurate procedures are really called for. Methods based on computer vision and machine learning are still under investigation but represent promising inexpensive tools compared to manual and laser-flow approaches (Kamentsky, 1973; Saphiro, 2003), which are massive-counting procedures and are not very precise in discriminating particular cell types.

There are two main types of white blood cells, Granulocytes and Agranulocytes. In turn, Granulocytes include Neutrophil, Eosinophil, and Basophil, whereas Agranulocytes can be of types Lymphocyte and Monocyte. In addition, during the sample preparation process any cell of these types can be broken, which greatly distorts its appearance and can therefore be regarded as a class itself. By looking at examples of these classes (Fig. 1), it is hard to think of any simple rule of thumb or heuristics that can guide the design of discriminative image descriptors, since there are few clear distinctive features and significant overlap.

This situation seems to call for a feature learning scheme such as deep learning (DL). Despite the advantages of automatically learning features directly from data, DL is generally known to pay off when a great deal of training instances are available, often in the order of tens of thousands.

An alternative is segmenting the images so that the region(s) of interest corresponding mainly to the cell type are isolated from the background. After the segmentation, one can characterize the region(s) by using a combination of shape, color and texture features. The main drawback of this approach is that segmentation itself is a non-trivial problem, and it is generally hard to produce robust and reliable segmentations, being a critical preliminary step for subsequent feature computations (Bikhet, Darwish, Tolba & Shaheen, 2000; Sarrafzadeh, Dehnavi, Rabbani & Talebi, 2015).

Back in the 1970s, early efforts were performed to automate the task of classifying 5 types of white blood cells using 74 instances and four very specific features (size and color of nucleus and cytoplasm) (Young, 1972), and reported that granularity, texture and shape do not provide additional discriminative ability.

Email addresses: vtraver@uji.es (V. Javier Traver)

[1] Present address: Image Processing Laboratory, Universitat de València, C/ Catedrático José Beltran, 2, 46980 Paterna (València), Spain. Phone: +34 96 354 32 29, Fax: +34 96 354 32 61

Some authors (Gómez-Gil, Ramírez-Cortés, González-Bernal, Pedrero, Prieto-Castro, Valencia, Lobato & Alonso, 2008) use specific shape and area descriptors to classify six types of leukocytes, with encouraging results in a limited dataset of 54 instances. However, this approach requires segmentation, which is accomplished manually, and makes use of *ad hoc* features; additionally, 7 out of 9 segmented images per class are synthetically generated by a human artist. Similarly, geometric features computed on segmented images by simple global thresholding are considered in a simplified 3-class Leukocytes discrimination problem (Hiremath, Bannigidad & Geeta, 2010). There exist other works with similar approaches (Piuri & Scotti, 2004; Gautam, Singh, Raman & Bhadauria, 2016). One interesting idea is trying to increase the classification rate by taking into account the class predicted for co-occurring cells within a given specimen instead of classifying single cells individually (Song, Abu-Mostafa, Sill & Kasdan, 1997). In (Rezatofighi & Soltanian-Zadeh, 2011), after nucleus-cytoplasm automatic segmentation, morphological, color and texture features are extracted and selected, resulting in a good-performing approach. A similar framework including segmentation and texture features is described in a system requiring user interaction for some tasks such as supervising the segmentation results (Sabino, da Fontoura Costa, Rizzatti & Zago, 2004). Features derived from intensity histograms, and their projection with Kernel Principal Component Analysis (Habibzadeh, Krzyżak & Fevens, 2013) are tested for white blood cell discrimination in a small dataset of 140 low-resolution instances. They compared these features and linear Support Vector Machine (SVM) with a simple Convolutional Neural Network (CNN) model (Le-Net5). Despite their small dataset, the CNN is reported to perform on a par with or better than the SVM.

An interesting alternative are local image descriptors computed at interest points, and the use of a pooling strategy for characterizing each individual image with a fixed-length representation. This approach has proven to be very effective in a number of computer vision tasks (Laptev & Lindeberg, 2003; Mikolajczyk & Schmid, 2005; Laptev, 2005; Wang & Mori, 2009), because it tends to be robust against background clutter and requires no explicit segmentation of the relevant regions of the image. However, in spite of its potential, to the best of our knowledge, this approach has not been tested before on this problem. We therefore explored the possibility of characterizing white blood cells with local descriptors and the well-known bag of visual words as a pooling mechanism.

More concretely, this work focuses on exploring and comparing several different interest point sampling strategies within two big broad categories: interest point detection and regular dense-like sampling. Although interest point detection is aimed at selecting particularly good distinctive image locations and is therefore a promising procedure, many studies have revealed that dense-like sampling, i.e. ignoring the image contents for point selection, can perform similarly or better than point detectors. Since our main interest is in comparing these two main "detection" approaches, a common local descriptor, the well-known Scale Invariant Feature Transform (SIFT) (Lowe, 2004), will be used in all cases.

From the broader scope of expert and intelligent systems, different methodologies can differ in their potential for addressing practical, real-world issues. On the one hand, due to legal and other reasons, there is a growing interest towards making artificial systems explainable (Gunning, 2016; Ribeiro, Singh & Guestrin, 2016), which is understandably particularly important in the medical domain (Holzinger, Biemann, Pattichis & Kell, 2017). In this sense, and in the context of the problem addressed in this paper, methods that use visual cues that are easier to interpret by humans, such as segmentation-based and geometric features, might be preferable. On the other hand, problem-specific ad hoc image features may require some form of expert knowledge to be elicited and implemented, a process that can be vague and costly. In this respect, general, problem-agnostic methodologies, such as the one proposed in this work, based on general-purpose local descriptors, might be desirable. This situation represents a challenging dichotomy between conflicting properties of alternative methodologies, which is revisited later (Sect. 4.4).

In the rest of this manuscript, the sparse interest-point detectors (Sect. 2.1), the regular sampling strategies (Sect. 2.2) and the bag-of-words approach (Sect. 2.4) are described, and a summary of the complete process is provided (Sect. 2.5). Then, the experimental work, results and discussions are detailed (Sect. 3). Finally, some discussion (Sect. 4) and the main conclusions of the work are given (Sect. 5).

2. Methodology

2.1. Interest point detectors

Interest points, also called keypoints, are locally salient image points. Mainly meant for image matching, interest point detectors have been devised for robustness against view changes. Although for classification purposes the requirements may not be exactly the same as in matching, the properties of these detectors turn out to be generally useful also for image characterization. Among the many existing detectors (Tuytelaars & Mikolajczyk, 2008; Krig, 2014), we select and explore these three detectors: SIFT, oFAST, and CenSurE. They are representative of three broad kinds of keypoint detectors, as follows:

- SIFT relies on the concept of *scale-space*, which can be scale-invariant, but may lose some location precision due to the involvement of coarser-resolution levels in the pyramid;
- oFAST detects *corners*, which can be fast to compute and are stable image locations, robust to changes of view, but not to scale changes;
- CenSurE aims at achieving both the stability of scale-space methods and accuracy of corners, by finding extrema in the responses of *centre-surround* filters.

SIFT detector

The Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) includes both a detector and a descriptor. We focus now on its detection step. Its descriptor (Section 2.3) is applied for every detector considered in this work. Unlike previous detectors at that time, the SIFT detector was designed to be scale and rotation invariant which provides it with robustness against affine image transformations and, hence, repeatability.

SIFT uses a multi-scale scheme so that interest points at different scales can be found. To that end, a Gaussian pyramid is built, and then Differences of Gaussians (DoGs) are computed between the consecutive levels of the pyramid. Next, local maxima in the DoGs are identified not only within the 8 neighbors of a pixel in the DoG at a given scale but also within its 9 neighbors of the subsequent-level DoG.

To remove the responses to edge-like structures which are not precise locations of interest points, a second-derivative filtering mechanism using the Hessian matrix is applied. An intensity threshold of the responses is finally applied so that low-response points are filtered out. This thresholding allows the procedure to be more selective in the number of the resulting points. The adequate threshold to use is highly problem-dependent. In our case, in order to have an acceptable number of key points for adequately representing the images, a relatively low-intensity threshold was used (Table 1).

*oFAST (*Oriented *FAST)*

The FAST (*Features from Accelerated Segment Test*) detector (Rosten & Drummond, 2006) is computationally very light, and builds on the conceptually simple idea (Rosten & Drummond, 2005) of comparing a pixel with its 16 neighbouring pixels lying on a circle centred on the test pixel. A pixel is decided to be a corner if a number n of contiguous pixels on that circle are brighter or darker than the test pixel, for a given threshold for the intensity difference. With $n = 12$, a high-speed version of the test is possible by checking only 4 pixels on the circle, and classifying the pixel as a corner if 3 out of these 4 pixels are all brighter or darker than the test pixel. However, this quick test was found to have some drawbacks such as poor generalization for $n < 12$. To address this and other shortcomings, a decision tree is built that learns to classify corners from a properly generated training set which uses a slow but accurate corner detector. Then, the resulting tree is converted into nested *if-then-else* rules, which can subsequently be used as the fast and more accurate detector.

The oFAST (Rublee, Rabaud, Konolige & Bradski, 2011) includes the orientation component which is missing in FAST. This orientation is estimated by using the angle of the vector joining the centre of the detected corner and the centroid of the image patch around the corner's centre. Although the original FAST itself does not detect features at different scales, it can be applied at the different levels of a pyramid scale. This multi-scale FAST and a circular neighbourhood of radius 9 (FAST-9) are used in this work.

CenSurE

Like SIFT, the CENter SURround Extrema (CenSurE) detector (Agrawal, Konolige & Blas, 2008) uses a Laplace scale-space operator, but in contrast to SIFT, which uses a DoG approximation, CenSurE uses a a centre-surround one. Unlike SIFT, which subsamples images at larger scales, CenSurE uses the full-resolution images at any scale, which is an important difference. Then, since CenSurE aims at computing all features at all scales, fast computation is a must. Therefore, simplified bi-level kernels (values in the kernels are only -1 and $+1$) are used as centre-surround filters. These kernels are computed in constant time regardless of their size, which may render this descriptor real-time-amenable. CenSurE considers a suite of bi-level kernels with increasing symmetry degrees (boxes, hexagons, octagons and circles), which also represent approximations to the Laplacian operator with increasing computational cost. The circular filter (STAR) is the slowest but the best approximation to the Laplacian; the other approximations are computed efficiently by means of integral images (Viola & Jones, 2004). Since in our study accuracy is more important than efficiency, we use the circular filter. However, to gain some insight into the computation-recognition trade-off, the fastest and least accurate difference-of-boxes kernel (a basic Haar wavelet) is also briefly tested.

Additional configuration details of these three detectors are provided in Table 1. Most of these parameters take their corresponding default values in the libraries used. Some thresholds have been modified to increase the number of detected points, since it was too small in some images, particular in a few defocused ones. In addition, in oFAST the sensibility Harris factor was reduced to 0 to detect only pronounced corners.

Table 1: Configuration of the keypoint detectors. Default values given in parentheses if different to the values actually used

SIFT	
Edge threshold	10
Intensity threshold	0.5 (5)
Window size (σ)	2
Num. of octaves	9
Num. of levels per octave	3
oFAST	
Target number of features	200
Threshold	0.04
Sensibility Harris factor	0 (0.04)
Number of scales	8
Scale factor	1.2
CenSurE	
Bi-level filter	STAR
Non-maximum suppression threshold	0.01 (0.15)
Line threshold	50 (10)
Minimum and maximum scale	1 and 7

Examples of keypoints are given in Fig. 1, where it is noticeable how oFAST detects corner-like points around contours whereas SIFT and CenSurE detect mostly points inside the

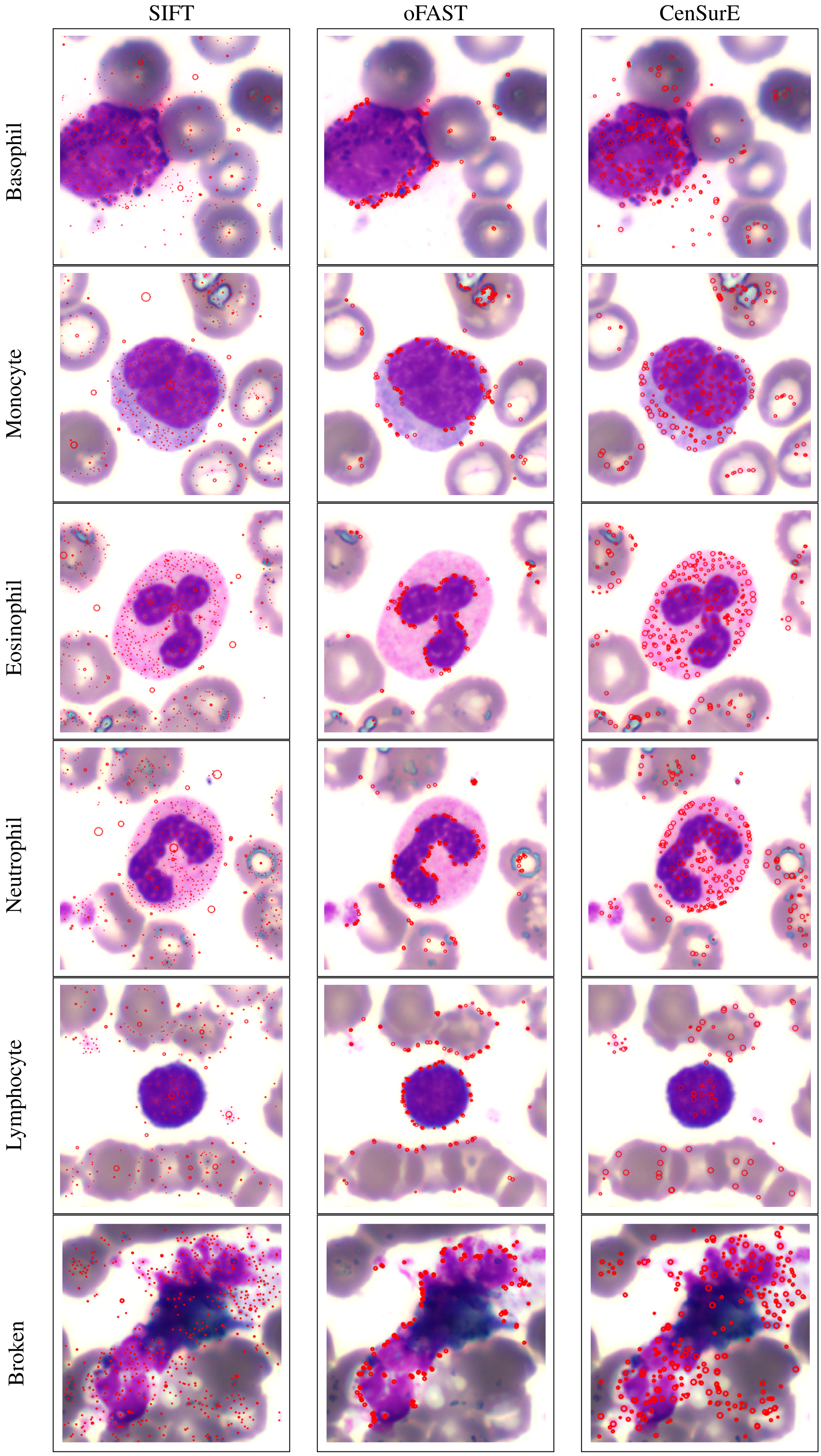


Figure 1: Keypoints detected with different detectors on example images of different white blood cell types

Table 2: Number of keypoints. For the regular sampling strategies this number is constant; for the sparse keypoint detectors, the average and standard deviation for a total of 1,315 images are given

	Detector			**Sampling strategy**				
	SIFT	CenSurE	oFAST	GUS	LUS	FS	*m*LUS	*h*LUS
Average	427	161	192	2,601	961	436	3,423	961
Std. dev.	209	115	31	-	-	-	-	-

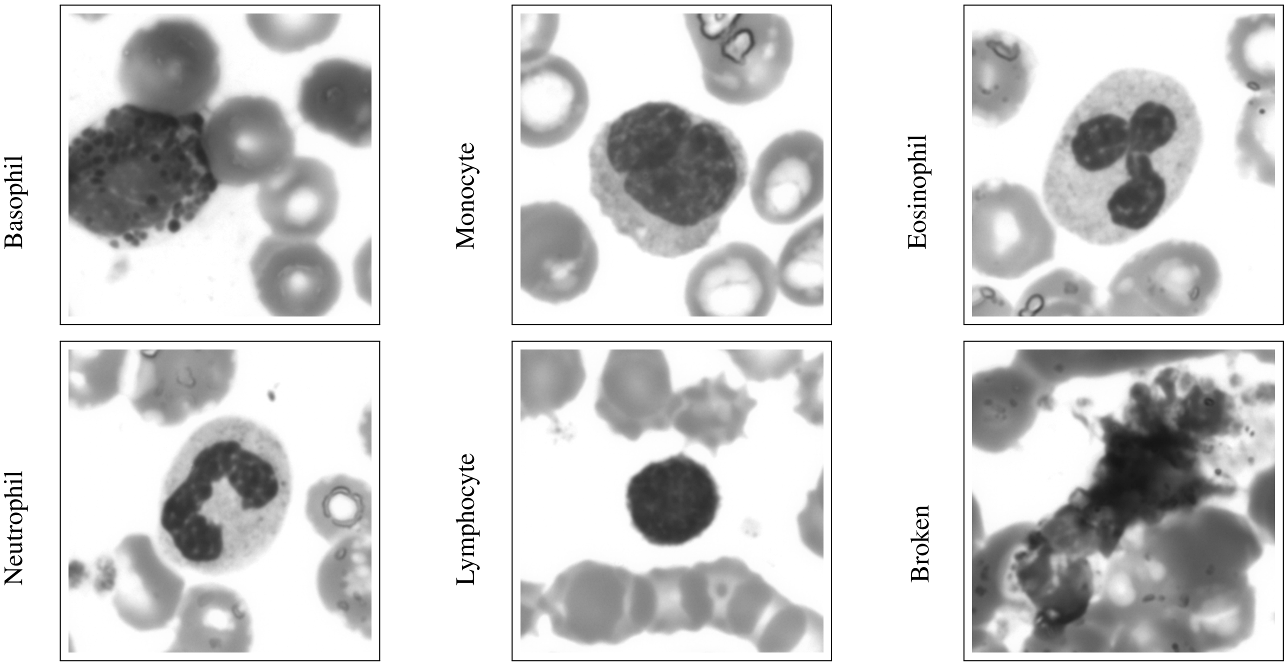


Figure 2: Gray-level images corresponding to images in Fig. 1. Keypoints detectors are actually computed on the gray-level images.

cells. Since keypoints are actually computed on gray-level images, the gray-level versions of these images are given in Fig. 2 as a reference. The number of keypoints that each detector finds are given in Table 2. On average, more than twice as many keypoints are detected with SIFT than with oFAST or CenSurE (Table 2). In oFAST, the mean number of keypoints per image is similar to the desired maximum number.

2.2. *Regular sampling strategies*

As an alternative to keypoint detectors, the location of interest points can be defined with some kind of regular grid. In this case, the points are not “interest” or “key” points anymore, since their geometric distribution is independent of the actual image contents. Although intuition may suggest that this “dense” pattern is less discriminative than locally salient points, previous works have shown that dense sampling can be effective in many visual tasks (Nowak, Jurie & Triggs, 2006; Wang, Ullah, Klaser, Laptev & Schmid, 2009). Since this might also be the case for this problem, we explore several sampling strategies, as discussed below.

Global uniform sampling (GUS)

The whole image is sampled at the same density, selecting one pixel every s pixels along both the horizontal and vertical axes. This amount s is commonly referred to as the *stride*. This sampling strategy makes sense when no clear insight is available on the possibly different relevance of different image regions. It is therefore implicitly assuming that all image parts may contribute equally to characterize the entire image contents.

Local uniform sampling (LUS)

As in GUS, a regular rectangular grid is defined, but instead of sampling the entire image, only a local region of interest (ROI) is covered. The ROI is expected to contain most of the discriminative information. In our problem, since the blood cell is generally roughly centred on the image and has a similar size across images in the dataset, a centred rectangular ROI of size 312×312 is consequently defined, so as to roughly cover the entire or most of the blood cell and not too much of the background. This ROI represents a 37% of the total area of the image.

Foveal-like sampling (FS)

This is a space-variant sampling technique, and represents a trade-off solution between GUS and LUS. Rather than sampling the whole image evenly in one case, or just focusing on one region and completely ignoring the rest of the image in the other,

Table 3: Strides and points per region at increasing distances to the centre for the foveal-like sampling (FS)

# regions	stride (s)	# points/region	# total points
4	8	49	196
8	16	16	128
20	32	4	80
32	64	1	32

we can sample more densely over the ROI, but still sample, even though less densely, the rest of the image. In particular, a foveal-like sampling is used with sampling density being inversely proportional to the distance to the centre. The rationale for this strategy is two-fold:

- On the one hand, since the extent of the blood cell varies across images, if we sample areas outside the ROI we may still capture some parts of the relevant blood cell which lie outside the ROI.
- On the other hand, contextual information is included in the representation. In some problems the context has been shown to play an important role in recognition. Therefore, we can evaluate how important is the context in this problem.

The space-variant sampling is implemented as follows. Non-overlapping square image regions, each of size 64×64, are considered, and they are sampled inversely to their distance to the centre. The number of regions, their sampling stride s, and the resulting number of points are given in Table 3, at increasing distance to the centre.

Sampling points on an example image for these three sampling strategies are given in Fig. 3.

Hierarchical uniform sampling (hLUS)

As in LUS, the image is sampled uniformly, but after the vocabulary of visual words has been built (Section 2.4), several histograms are computed, one at each cell in a quadtree, hence the name of this idea in the literature (Bosch, Zisserman & Muñoz, 2007), Pyramid of Histograms Of visual Words (PHOW). The rationale for this representation is to capture some spatial information that is otherwise lost with a single global histogram (Lazebnik, Schmid & Ponce, 2006). As in LUS, only descriptors in a central ROI of size 312×312 are considered. A 3-level pyramid is used, as in (Bosch, Zisserman & Muñoz, 2007), resulting in 1+4+16 = 21 histograms per image. Consequently, the feature vector describing each image turns out to be relatively high ($21 \cdot k$), with k being the vocabulary size. For computational and comparison reasons, we opted to reduce this dimensionality by applying Principal Component Analysis (PCA) (Alpaydın, 2004) and choosing k as the target dimensionality, to match that of the rest of the compared methods.

Multiresolution uniform sampling (mLUS)

This is also a dense sampling, but computed at several image resolutions. Since it also uses a local ROI, it can be seen as a multiresolution LUS (*m*LUS).[2]

For all the regular sampling but for FS, a common stride $s = 10$ is used. For GUS, and *m*LUS this stride results in a large number of descriptors per image (specifically, $3,423$ for *m*LUS), which raises memory issues in the clustering procedure (the overall requirement is of 1,800 MB). To circumvent this issue, and obtain a feasible reduced number of points, we tried two approaches, both seeking a fair comparison with the other strategies: (1) doubling the sampling stride s (i.e. halving the number of resulting points per image); and (2) selecting randomly a subset of m points per image for the clustering, with $m = 961$ corresponding to the number of points used in LUS. We tried both approaches, but found empirically the second one to be preferable andthus it is the one whose results are reported here.

The number of keypoints for each of these sampling strategies (Table 2), except for FS, is an order of magnitude larger than the number of keypoints selected by the sparse detectors.

2.3. SIFT descriptor

As said before, every keypoint in all the considered strategies is described locally using the SIFT descriptor (Lowe, 2004). This descriptor consists of the concatenation of 16 8-bin gradient orientation histograms computed over a neighbourhood of size 16×16 around the keypoint divided into 4×4 cells. For the detectors that compute an orientation (SIFT and oFAST), this is plugged into the SIFT descriptor, to endow the descriptor with rotation invariance. The result is a L_1-normalized vector of length 128 ($= 4 \cdot 4 \cdot 8$). Although the dataset used in the experiments consists of color images, in this work the SIFT descriptor is applied to their gray-level versions.

2.4. Vocabulary and bag of words

The set of SIFT descriptors (or a subset of them in the case of GUS and *m*LUS, as discussed above) of the images in the training set are clustered using k-means (Jain, 2010), for a user-provided value of k, and the centroids of each cluster represent the visual words of our vocabulary. To compute these clusters and their corresponding centroids $\{\mathbf{c}_i\}_1^k$, k-means starts by some (random) initial centroids and then proceeds with these two steps, which are repeated until convergence, with a maximum number of iterations:

1. Given the centroids, reassign each point $\mathbf{p} \in \{\mathbf{p}_i\}_1^n$ to the cluster corresponding to its closest centroid.
2. Given the clusters, recompute the centroid of each cluster.

Then, once this vocabulary is built, a given new local descriptor is assigned to the word corresponding to its closest centroid. Therefore, an image in both the training and test sets is coded as a k-bin histogram (the bag of words) with the counts of the words of all its descriptors. Formally, given m local (SIFT) descriptors in an image, $\{\mathbf{q}_j\}_1^m$, the k-bin histogram $\{\mathbf{h}_b\}_1^k$ corresponding to this given image is computed as follows:

[2] The function which extracts the data for our *m*LUS within the VLfeat package is called, and refers to, the PHOW method, but in our understanding PHOW actually refers to the hierarchical approach followed in our *h*LUS strategy.

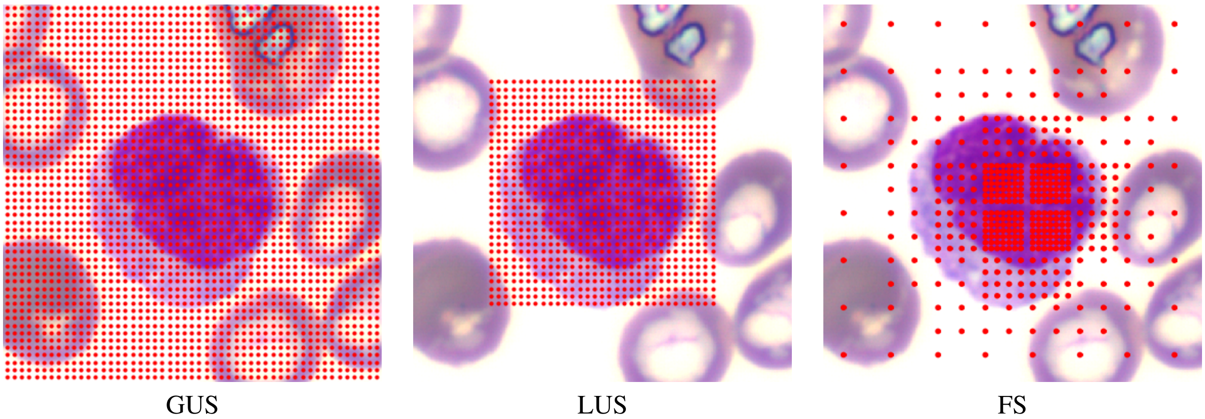


Figure 3: Sampling layouts for GUS, LUS, and FS. The other two approaches (mLUS and hLUS) are based on the same sampling as LUS

1. Find the closest centroid of each point $\mathbf{q} \in \{\mathbf{q}_j\}_1^m$, and assign it to the corresponding cluster. Let $c_j \in \{1, 2, \ldots, k\}$, $j \in \{1, \ldots m\}$ be these assignments of the m points to the k clusters.
2. Count the number of points assigned to each cluster, i.e. $\mathbf{h}_b = \sum_{j=1}^{m} \delta(c_j, b)$, where $\delta(x, y) = 1$ if $x = y$, and $\delta(x, y) = 0$ if $x \neq y$.

Finally, each of these histograms is L_1-normalized individually, per image.

2.5. Summarizing the complete procedure

To summarize (see the flowchart of the whole process in Fig. 4), at both the training and the prediction stages, keypoints are detected on the input images (Sect. 2.1) or, alternatively, selected with a regular sampling strategy (Sect. 2.2). In both cases, the points are described with the SIFT descriptor (Sect. 2.3).

At training time, from a set of SIFT descriptors from the training images, a vocabulary is built using vector quantization with k-means (Sect. 2.4). Through the vocabulary, one bag-of-words histogram (Sect. 2.4) is computed per image, both at training and prediction stages. Then, only at training, a set of histograms is used to train the SVM classifier (Sect. 3).

Only at the prediction stage, the trained classifier uses the BoW histogram of a novel image to predict its type of WBC. Therefore, the most costly procedures (BoW and SVM training) are performed only once, at the training stage. The input to the training stage is a set of training images with their corresponding ground-truth labels, and its output is a trained SVM. The ground-truth labels are only used during SVM training, since the k-means is completely unsupervised. As for the prediction stage, the input is a new, unlabelled image, and its output is the predicted label for the input image. The prediction stage uses the vocabulary and the trained SVM obtained at the training stage.

3. Experimental work

3.1. Setup

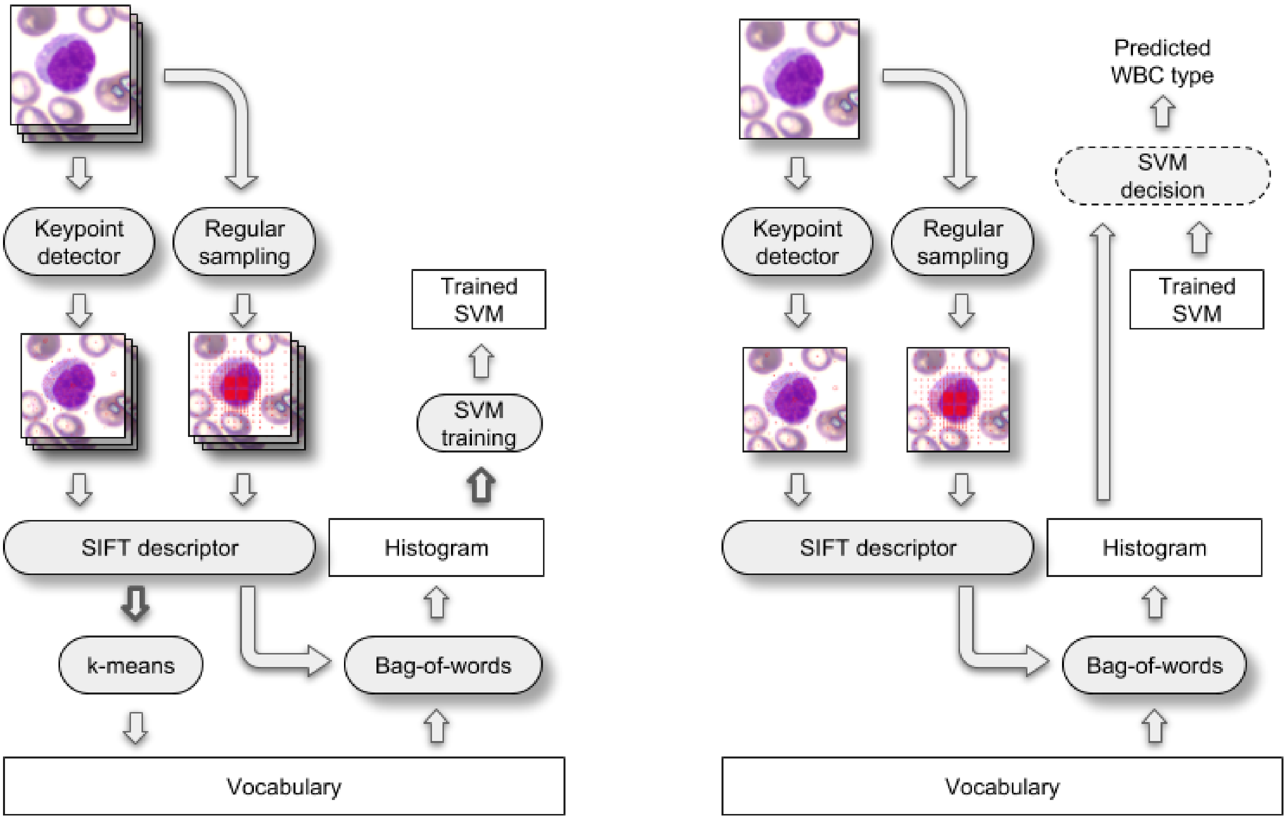
Dataset. The dataset provided by the hematologists at Hospital General, Castellón, Spain, consists of 1,315 512×512 color images of 6 types of blood cells (Table 4). The samples were randomly chosen from a set of alarms coming from the automatic differential flow system. These blood samples were considered as potentially abnormal so they had to be visually inspected by expert haematologists.

The samples were stained with the common May-Grumwald-Giemsa procedure. Hematologists labelled each individual image as one of the 6 classes, through a graphical user interface with the possibility of correcting their mistakes and navigating back and forth along the images. These labels are used as the ground-truth labels for training and testing purposes.

Table 4: Distribution of the classes of white blood cells in the dataset

Class	# instances	Ratio (%)
Lymphocyte	511	38.86
Neutrophil	476	36.20
Broken	185	14.07
Monocyte	99	7.53
Eosinophil	38	2.89
Basophil	6	0.46
Total	1,315	100

WBC localisation and counting. The images in this dataset were previously extracted in a pre-processing stage by an automatic machine vision system consisting of a microscope, a color camera, a robotic platform to move the blood smears under the microscope objective, and a personal computer to control and perform the image processing tasks. This system is able to scan the whole surface of a blood smear and it can easily differentiate at a given scale between white, red, and other main types of cells, because of their colour and sizes. Once a white cell is identified, the system zooms in and takes an image



(a) Training

(b) Prediction

Figure 4: Flowchart of the process at the (a) training and (b) prediction stages. See text for details

at a larger scale of the detected white cell. It is thus the output images of this system that are the input to our recognition approach, and the whole system would therefore be capable of counting the different types of WBC, as required for health reporting.

Vocabulary. To select the size of the vocabulary for the BoW model, tests in an initial data exploration phase were performed, and sizes $k = 150$ and $k = 500$ were found to give the overall best performance among a large set of sizes ranging from 10 to 1,000. Subsequent tests with these two sizes revealed that results with $k = 500$ were slightly better and thus used for the rest of the experimentation.

Classifier. A support vector machine (SVM) is used, with either linear or radial-basis function (RBF) kernels, selected by cross-validation. Given the distribution of instances per class (Table 4), there is a significant class imbalance, a situation known to require particular strategies to avoid a bias towards the majority classes. As a preliminary effort to deal with this issue, the Synthetic Minority Over-sampling Technique (SMOTE) was explored, both the SVM-SMOTE and the borderline SMOTE (Nguyen, Cooper & Kamei, 2011; Han, Wang & Mao, 2005). When using these approaches we appreciated a slight improvement in the recognition for the minority classes, but at the expense of a slight degradation of the recognition of the majority classes. Besides the class imbalance, too few instances of the minority classes are available, and therefore this issue is left for addressing it in future work with a bigger dataset.

Validation protocol and performance assessment. A hold-out ℓ-fold procedure is followed, which is adequate when datasets are not large, which is the case. We used $\ell = 5$ and, for every split, the whole procedure (clustering + histogram coding + training + testing) is repeated. Additionally, as a requirement of one statistical test (introduced below), a 2-fold hold-out was performed 5 times.

The common accuracy performance is complemented with precision and recall as well as with two more suitable measures under class imbalance, the F1 measure and the Matthews Correlation Coefficient (MCC) (Matthews, 1975; Boughorbel, Jarray & El-Anbari, 2017). Although MCC was originally thought for a binary case, its generalization for multi-class case has also been derived (Gorodkin, 2004):

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

where C_{ij} are the entries of the confusion matrix, i.e. the number of instances belonging to class i which are classified as the class j. The possible values for this coefficient are in the range $[-1, 1]$, the higher the better.

Box-and-whisker plots (Sahay, 2016) are provided for visually depicting the average performance and its variability across the 5 folds.

Hyperparameter selection. For each fold, a grid search of the SVM hyperparameters is performed to select those resulting in the highest precision. The hyperparameters included the kernel (linear or RBF), the regularization parameter (in the range $C \in \{1, 2, \ldots, 11\}$), and the scale parameter in the case of the RBF (in the range $\gamma \in \{10^j : j \in \{-6, -5, \ldots, 3, 4\}\}$). Note that ranges covering other more extreme values for C and γ were initially tested, but after observing the trend of the values being chosen, they were adjusted to the reported ones. The RBF kernel was chosen almost always for all the methods.

Statistical tests. To find out the statistically significant differences in performance among the different methods, we applied two significance tests: the corrected resampled t-test (Nadeau & Bengio, 2003) (CTT), and the 5×2cv paired t-test (5×2cv) (Dietterich, 1998), which are known to be suitable for comparing two methods on a single dataset. These test statistics follow a Student-t distribution t_f, with f degrees of freedom, and are computed as follows (Bouckaert & Frank, 2004):

$$\mathrm{CTT} \equiv \frac{\mu}{\sqrt{\left(\frac{1}{\ell} + \frac{n_{te}}{n_{tr}}\right) \cdot \sigma^2}} \sim t_{\ell-1},$$

$$5{\times}2\mathrm{cv} \equiv \frac{d_{11}}{\sqrt{\frac{1}{5} \sum_{j=1}^{5} \sigma_j^2}} \sim t_5,$$

where

- for CTT, ℓ is the total number of runs of the method; n_{tr} and n_{te} are, respectively, the number of training and testing instances in each run; and μ ad σ are the estimates of the mean and standard deviation of the ℓ differences of the chosen performance measure for the two compared methods; and
- for 5×2cv, let $a_{ij}, b_{ij}, i \in \{1, \ldots, \ell\}, j \in \{1, \ldots, r\}$ be the performance measures of two compared methods (say A and B) for the r-times ℓ-fold cross-validation (here, $\ell = 2, r = 5$), using exactly the same training and testing sets for both methods in every run; then $d_{ij} = a_{ij} - b_{ij}$ (the individual differences in performance), and $\sigma_j = \sum_{i=1}^{2}(x_{ij} - d_j)^2$, with $d_j = \frac{x_{1j}+x_{2j}}{2}$ (the mean difference for a single run j).

For the sake of convenience, the outcome of these tests will be presented graphically following the symbols in Table 5.

Software. As supporting software, we used the VLfeat library (Vedaldi & Fulkerson, 2008) for the SIFT detection, OpenCV (Howse, Joshi & Beyeler, 2016) for the SIFT description, and Python packages for sparse detectors (scikit-image (van der Walt, Schönberger, Nunez-Iglesias, Boulogne, Warner, Yager, Gouillart, Yu & the scikit-image contributors, 2014)), SVM, grid search and PCA (scikit-learn (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot & Duchesnay, 2011)), and clustering, vector quantization, and t-test lookups (scipy (Jones, Oliphant, Peterson et al., 2001–)).

Table 5: Visual representations of significance degree found by the statistical test given a p-value. The lower the p-value, the higher the significance

●	$p < 0.01$	◔	$p \leq 0.1$
◕	$p \leq 0.05$	○	$p > 0.1$ (no significance)

3.2. Results

Experimental results are discussed subsequently through the analysis of the boxplots, the confusion matrices, the statistical tests and the computation times. Performance comparison with other approaches are also provided as a general reference.

Boxplots. By observing the boxplots (Fig. 5), it can be found that purely uniform sampling strategies (GUS and LUS) perform the worst. Among them, it seems a better idea to focus on a central ROI (LUS), otherwise there seems to be too much noisy and irrelevant background included and this may end up being harmful. Interestingly, the foveal-like strategy (FS), by gradually decreasing the sampling density away from the image centre, provides a nice and better solution: the fact of capturing a bit of background with a coarser sampling seems to provide useful contextual information. There exists not much difference between the multi-resolution scheme (mLUS) and the hierarchical one (hLUS), and both tend to outperform the uniform sampling strategies either global (GUS) or local (LUS). A bit against our intuition, the multi-resolution (mLUS) and hierarchical (hLUS) strategies performed very well compared with the other regular sampling strategies and the keypoint detectors. This might be explained by the fact that some form of discriminative scale or texture-like information is better captured.

The three sparse detectors provide generally a good recognition rate, with oFAST possibly resulting in (slightly) better performance. Compared to the regular sampling strategies, oFAST has a comparable performance to mLUS or hLUS. The statistical significance analysis below provides some insight into the actual relevance of the similar or different performances among the methods.

Confusion matrices. Confusion matrices averaged over the ℓ folds (Fig. 6) provide insights into how the different types of white blood cells are recognized by the different strategies. Neutrophil and Lymphocyte are generally the classes with higher recognition rates, which might be related to the fact that these are the classes with the most instances. A further reason is that Lymphocyte is a relatively distinct class with respect to any other class. The strong visual similarity, even for a human observer, between Neutrophil and Eosinophil, possibly explains why all the methods tend to confuse them, and calls for very

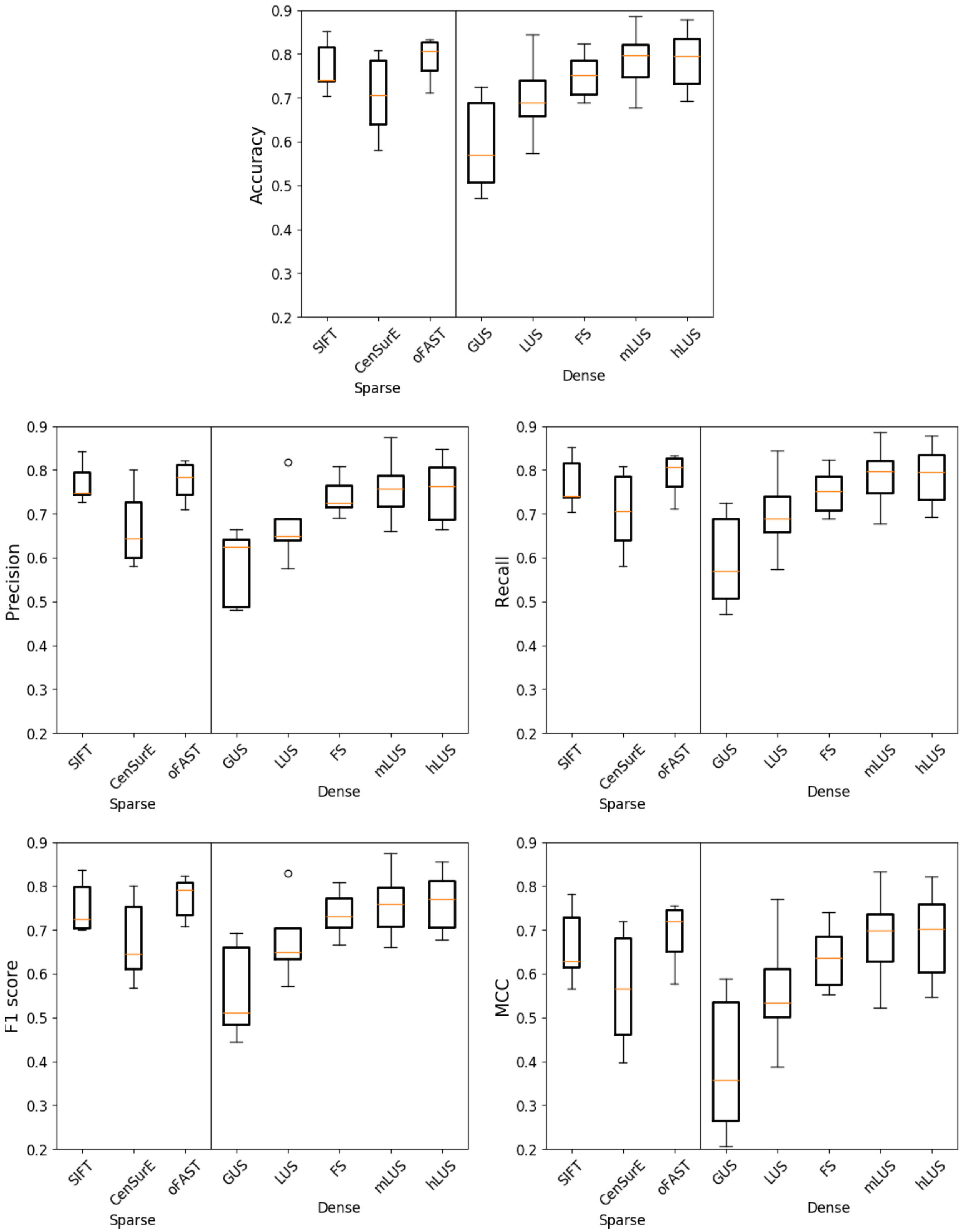


Figure 5: Recognition performance for the different detectors

specialized knowledge in terms of human expertise and/or automatic feature learning. Basophils are often confused with a variety of other blood types, with oFAST being the detector which leads to better performance. No blood cell instance is classified as Basophil, most likely due to the very few instances of this under-represented class. In particular, the frequent confusion of Basophils with Lymphocytes has also been reported by other authors (Habibzadeh, Krzyżak & Fevens, 2013). Monocytes are often confused with either Neutrophils or Lymphocytes. Interestingly, some strategies (LUS, GUS) misclassify Monocyte more as Neutrophil, and others (CenSurE, oFAST, FS) more as Lymphocyte, which indicates the different information captured by sparse detectors and foveal-like sampling on the one hand, and the uniform sampling on the other.

Significance analysis. The results of the significance tests roughly correspond to the intuitive visual interpretation one can get by comparing the box-plots. Furthermore, as it happens with these plots, the tests applied to different metrics are very similar; therefore, only results for a single metric (accuracy) are shown. For easier visual comparison, the visual representation of the significance level is given, instead of the the *p*-values (Table 6). Although both tests reveal many common differences among the methods, there are also some discrepancies. Overall, both tests reveal that the dense method LUS performs differently (worse) than most of the other methods (both dense and sparse). Both tests also agree to find CenSurE and oFAST different. And neither of the tests find any significant difference between the top performing regular-sampling methods (FS, *m*LUS, *h*LUS).

Comparison with other works. Since no standard public dataset of white blood cells is available for benchmarking, it is not possible to compare directly the different approaches. Furthermore, since the datasets used in these works are relatively small (often fewer than 100 instances), and some of them even use fewer classes (in one case, 3 instead of 6), the results are also less representative and/or the classification tasks easier. However, and with these caveats in mind, it can still be illustrative to compare ours with the performance reported in other studies with a similar problem but different datasets. Some segmentation-based approaches are reported to have accuracies about 90% or higher. With a maximum mean accuracy of 79% and a peak accuracy of 85%, the proposed strategy has comparable performance while being simpler and more robust. The particularly good performance of some works such as (Rezatofighi & Soltanian-Zadeh, 2011), with an accuracy of 96% (on a dataset of 251 instances), may suggest that carefully crafted and complex and possibly costly segmentation procedures such as snakes, the use of color and texture information, and many ad hoc design decisions and pre-processing steps (histogram equalization, color space transformation, image smoothing, etc.) may lead to good performance. In contrast, the approaches tested in our work are much more general, work on gray-level images, perform no image pre-processing at all, and have no explicit segmentation (besides a rough ROI central selection in some of the cases).

The promising results of the studied strategies and the observation of the reported results of the alternative approaches suggest possibilities for recognition improvement within the framework of local descriptors and BoW, thus getting closer to what experts in this domain consider acceptable (e.g. accuracies around 90%).

Computation times. Given the similar performance of several of the tested methods, it is important to compare them also in computational terms. The times in our computer (Intel©Core$^{\text{TM}}$ i5 650, 8-GB RAM, ATI Radeon HD4650 GPU) and (unoptimized) Python implementation are given in Fig. 7. The detection time has been computed on a subset of about 20% of the total dataset, and scaled proportionally to represent the full dataset.

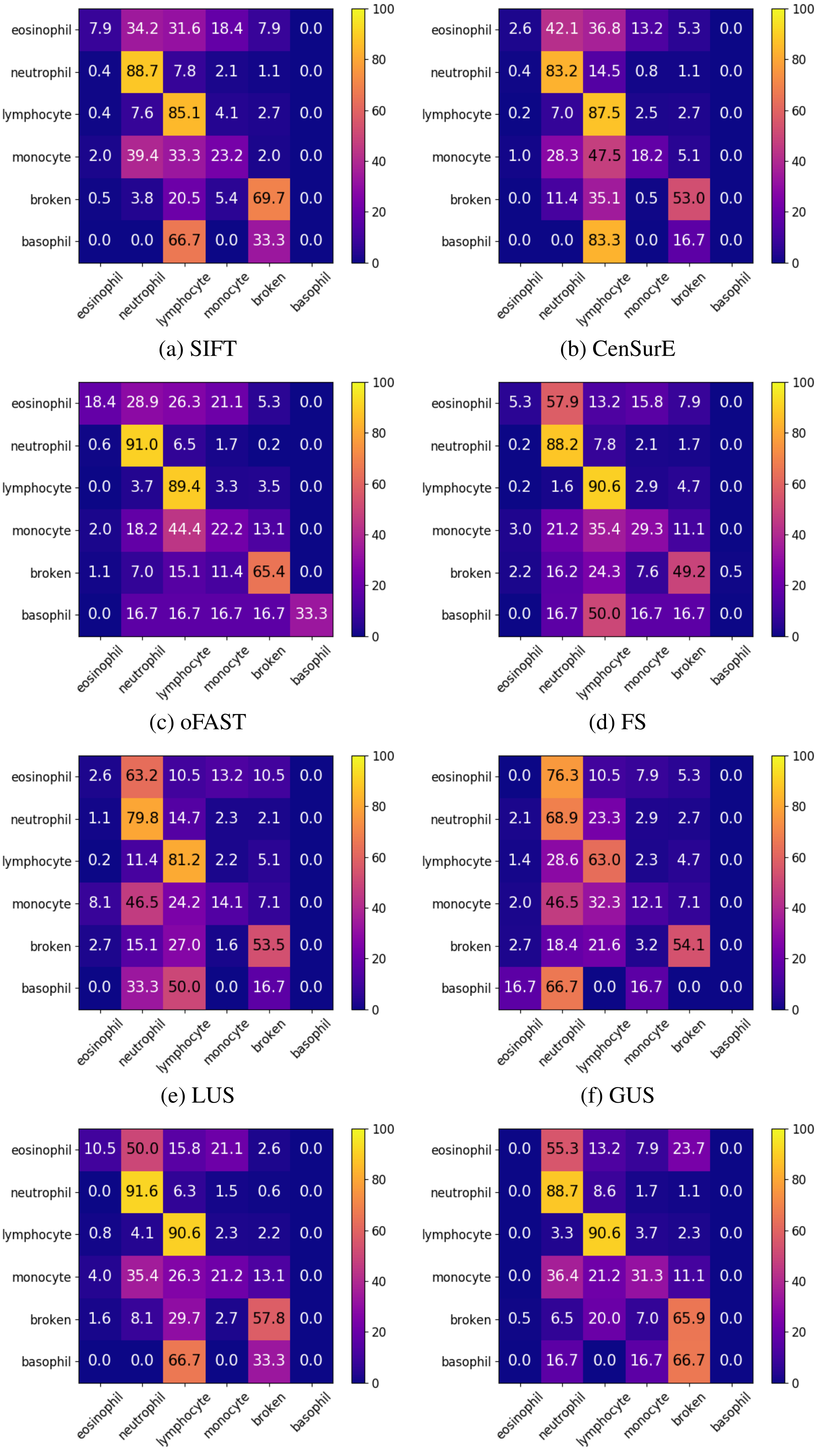
It can be seen that detecting or extracting the points is generally the most costly procedure, and apart from the *m*LUS case, it is naturally bigger when using actual detectors than in sampling strategies. The time for CenSurE is particularly large due to the bi-level filter chosen (STAR); other faster but less accurate filters are possible (see below). In contrast, the time for building the vocabulary, which is proportional to the number of points used in the clustering, is generally bigger with the dense methods. Similarly, computing the histograms depends on the number of points per image, and generally represents the smallest percentage of the total time. Finally, the training time of the classifier is essentially the same for all methods because the training data are the already-computed histograms of the same number of bins in all cases.

Taken together, and excluding the most costly CenSurE and *m*LUS, computation times for the sparse and dense methods are comparable, although the sparse keypoint detectors might be slightly advantageous. It is important to bear in mind that building the vocabulary and learning a classifier are off-line procedures, which usually need to be computed only once. At prediction time, the relevant costs include detecting and describing the keypoints, computing the histograms from the pre-computed vocabulary, and predicting the label from the already learned classifier. These steps can be computed fairly quickly and may be acceptable for real settings unless hard real-time constraints are demanded (e.g. for a responsive user interface).

Accuracy-time. When assessing the methods by considering jointly their recognition and computational performances (Fig. 8), the oFAST and the *h*LUS seem the most advantageous methods. Depending on the required classification-time tradeoff, the SIFT and FS methods can also be competitive alternatives. Although the mean accuracy is only a partial picture of the recognition ability, this scatter plot provides an interesting insight into the overall recognition and computational performances. Another fact worth observing is how a faster kernel (Difference of Box, DoB) in CenSurE results in lower discriminative power than the more costly STAR filter.

4. Discussion

This section is aimed at clarifying some aspects regarding the context, strengths and weaknesses of the proposed approach, as well as the limitations of the study, possible future work and the relevance of the contribution in the context of intelligent systems.



	eosinophil	neutrophil	lymphocyte	monocyte	broken	basophil
eosinophil	7.9	34.2	31.6	18.4	7.9	0.0
neutrophil	0.4	88.7	7.8	2.1	1.1	0.0
lymphocyte	0.4	7.6	85.1	4.1	2.7	0.0
monocyte	2.0	39.4	33.3	23.2	2.0	0.0
broken	0.5	3.8	20.5	5.4	69.7	0.0
basophil	0.0	0.0	66.7	0.0	33.3	0.0

(a) SIFT

	eosinophil	neutrophil	lymphocyte	monocyte	broken	basophil
eosinophil	2.6	42.1	36.8	13.2	5.3	0.0
neutrophil	0.4	83.2	14.5	0.8	1.1	0.0
lymphocyte	0.2	7.0	87.5	2.5	2.7	0.0
monocyte	1.0	28.3	47.5	18.2	5.1	0.0
broken	0.0	11.4	35.1	0.5	53.0	0.0
basophil	0.0	0.0	83.3	0.0	16.7	0.0

(b) CenSurE

	eosinophil	neutrophil	lymphocyte	monocyte	broken	basophil
eosinophil	18.4	28.9	26.3	21.1	5.3	0.0
neutrophil	0.6	91.0	6.5	1.7	0.2	0.0
lymphocyte	0.0	3.7	89.4	3.3	3.5	0.0
monocyte	2.0	18.2	44.4	22.2	13.1	0.0
broken	1.1	7.0	15.1	11.4	65.4	0.0
basophil	0.0	16.7	16.7	16.7	16.7	33.3

(c) oFAST

	eosinophil	neutrophil	lymphocyte	monocyte	broken	basophil
eosinophil	5.3	57.9	13.2	15.8	7.9	0.0
neutrophil	0.2	88.2	7.8	2.1	1.7	0.0
lymphocyte	0.2	1.6	90.6	2.9	4.7	0.0
monocyte	3.0	21.2	35.4	29.3	11.1	0.0
broken	2.2	16.2	24.3	7.6	49.2	0.5
basophil	0.0	16.7	50.0	16.7	16.7	0.0

(d) FS

	eosinophil	neutrophil	lymphocyte	monocyte	broken	basophil
eosinophil	2.6	63.2	10.5	13.2	10.5	0.0
neutrophil	1.1	79.8	14.7	2.3	2.1	0.0
lymphocyte	0.2	11.4	81.2	2.2	5.1	0.0
monocyte	8.1	46.5	24.2	14.1	7.1	0.0
broken	2.7	15.1	27.0	1.6	53.5	0.0
basophil	0.0	33.3	50.0	0.0	16.7	0.0

(e) LUS

	eosinophil	neutrophil	lymphocyte	monocyte	broken	basophil
eosinophil	0.0	76.3	10.5	7.9	5.3	0.0
neutrophil	2.1	68.9	23.3	2.9	2.7	0.0
lymphocyte	1.4	28.6	63.0	2.3	4.7	0.0
monocyte	2.0	46.5	32.3	12.1	7.1	0.0
broken	2.7	18.4	21.6	3.2	54.1	0.0
basophil	16.7	66.7	0.0	16.7	0.0	0.0

(f) GUS

	eosinophil	neutrophil	lymphocyte	monocyte	broken	basophil
eosinophil	10.5	50.0	15.8	21.1	2.6	0.0
neutrophil	0.0	91.6	6.3	1.5	0.6	0.0
lymphocyte	0.8	4.1	90.6	2.3	2.2	0.0
monocyte	4.0	35.4	26.3	21.2	13.1	0.0
broken	1.6	8.1	29.7	2.7	57.8	0.0
basophil	0.0	0.0	66.7	0.0	33.3	0.0

(g) *m*LUS

	eosinophil	neutrophil	lymphocyte	monocyte	broken	basophil
eosinophil	0.0	55.3	13.2	7.9	23.7	0.0
neutrophil	0.0	88.7	8.6	1.7	1.1	0.0
lymphocyte	0.0	3.3	90.6	3.7	2.3	0.0
monocyte	0.0	36.4	21.2	31.3	11.1	0.0
broken	0.5	6.5	20.0	7.0	65.9	0.0
basophil	0.0	16.7	0.0	16.7	66.7	0.0

(h) *h*LUS

Figure 6: Confusion matrices for the different detectors (a–c) and sampling strategies (d-h). True classes as rows, predicted ones as columns

Table 6: Results of significance tests on the accuracy

CTT test							
	CenSurE	oFAST	GUS	LUS	FS	*m*LUS	*h*LUS
SIFT	◔	○	●	●	○	○	○
CenSurE		●	◕	○	○	◕	◕
oFAST			●	◕	○	○	○
GUS				◕	●	◕	●
LUS					◕	◕	◕
FS						○	○
*m*LUS							○
5×2cv test							
	CenSurE	oFAST	GUS	LUS	FS	*m*LUS	*h*LUS
SIFT	○	○	◕	◔	○	○	○
CenSurE		●	○	○	◕	○	◔
oFAST			●	●	●	◕	◕
GUS				○	◕	◕	●
LUS					○	◔	◕
FS						○	○
*m*LUS							○

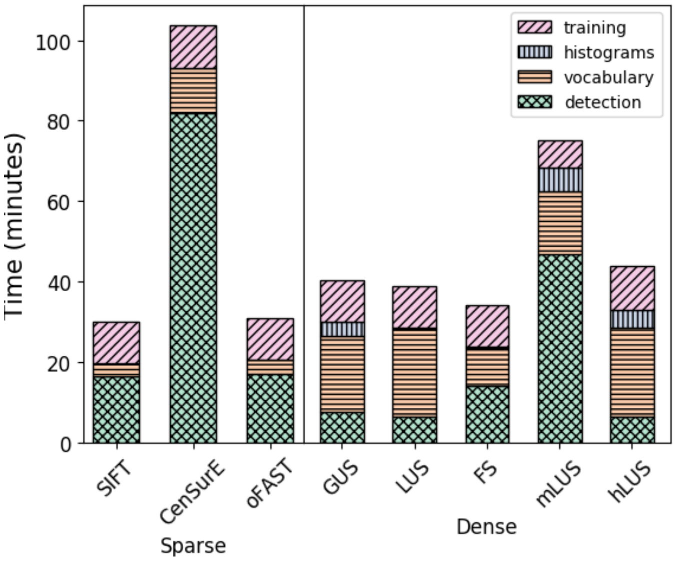


Figure 7: Time breakdown for the different steps of the BoW for each method

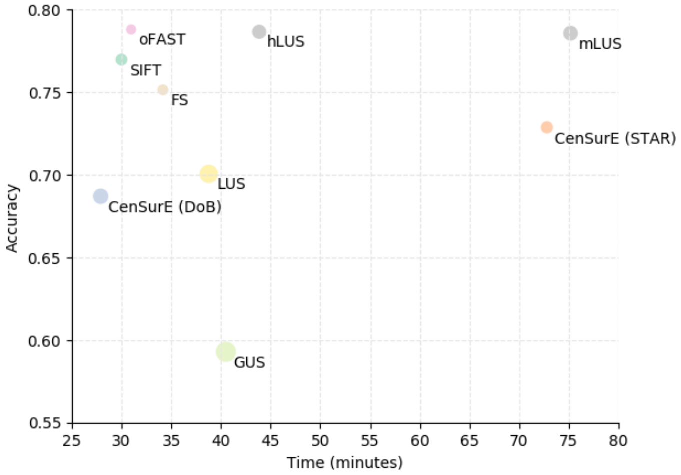


Figure 8: Accuracy-time scatter plot of the different strategies. The size of the circles is proportional to the variance in the accuracy

4.1. *Image segmentation*

It is important to distinguish two types of image segmentations that may arise in the context of the real-world problem: on the one hand, there are *segmentation-for-localisation* methods to roughly localise the cells of interest within a larger image, and whose solution in our work is taken for granted, since it is given by the robotized system, as described in Sect. 3, and it is out of the scope of this work. This segmentation is akin to the nucleus segmentation step in works such as (Rezatofighi & Soltanian-Zadeh, 2011). On the other hand, there are *segmentation-for-characterisation* methods, whose purpose is to extract features within the image region of the cell or within the nucleus and cytoplasm regions, separately (Rezatofighi & Soltanian-Zadeh, 2011).

In this respect, the proposed approach relies on some kind of segmentation-for-localisation procedure, and it is therefore sensitive to the potential failures of this system. However, it does not need any segmentation-for-characterisation method, and this fact is a strength of the approach (as discussed below in Sect. 4.3).

4.2. *Computational and robustness issues*

Regarding the computational cost, although a complete and rigorous comparison with alternative approaches is out of the scope of this paper, our estimated computational time and those reported in papers of related work are provided (Table 7), just as a rough guideline. Our reported times correspond to the most and the least costly strategies, which differ in one order of magnitude. The times for the remaining strategies lie in between these two times. The caveat for the times in the table is obviously that they correspond to very different settings, in terms of image sizes, implementation choices, and computing facilities, and are therefore not directly comparable, but they can be somehow illustrative of the computational efficiency of local-descriptor-based methods.

As for the robustness issue, general segmentation algorithms are known to easily over-segment or under-segment images (Estrada & Jepson, 2009). Because of their nature (low contrast, noise, etc.), medical images can be particularly challenging to segment and, despite the progress made, accurate segmentations are elusive (Elnakib, Gimel'farb, Suri & El-Baz, 2011), and robust segmentation remains an unsolved problem (Zhang & Metaxas, 2016). In contrast to segmentation-based methods, local descriptors do not rely on particular image regions being defined. The holistic and general purpose of this methodology makes it inherently robust to image contents, being tolerant to the presence of significant background clutter (Zhang, Marszałek, Lazebnik & Schmid, 2007).

4.3. Strengths and weaknesses

When comparing the proposed approach to segmentation-based approaches (Table 8), it can be observed that the same characteristic can be seen as an asset or become a weak point of the methodology. For instance, segmentation-for-characterisation (either segmenting the cell from the background or even separating the nucleus from cytoplasm) is a good asset since very specific features can be computed, but since robust segmentations tends to be elusive, the reliability of these features can be compromised. Likewise, the use of local descriptors makes only mild assumptions of image contents, and this is a great advantage. However, the approach might fail under strong deviations from these assumptions, which might happen if the cell is too small with respect to the background area, or there are too many distractive background features, or more than a single type of cell is present in the image. If the relevant cell region was highly off-centred, regular-sampling strategies may be affected significantly more than keypoint detectors.

Both approaches share the limitation of being dependent on the quality of the output of the segmentation-for-localisation step.

4.4. Future work

Exploring the discriminative role of color. This work used the SIFT descriptor on gray-level images since single-channel descriptors are most known and explored. However, since color information may provide discriminative information for WBC recognition, the role of color is an interesting open topic to explore. Possibilities range from early fusion (e.g. concatenating SIFT descriptors in different color bands) and late fusion (e.g. combining classifiers) to color extensions of the SIFT descriptor and alternative descriptors (van de Sande, Gevers & Snoek, 2010; Guo, Huang & Qiao, 2017). Depending on the problem, the selection of the descriptors with the adequate amount of invariance to illumination and color changes is recommended (van de Sande, Gevers & Snoek, 2010).

Combining segmentation-based and segmentation-free approaches. The former comparison (Sect. 4.3, Table 8), including the dichotomy between explainability and problem agnosticism discussed in the introduction, reveals that the strengths and weaknesses of local descriptors (segmentation-free) approach and segmentation-based approaches are essentially complementary. Consequently, the design of a system combining the strengths of both worlds may be considered. More specifically, an open issue would be how to endow approaches based on local descriptors with the problem-domain insights that can arguably be obtained more naturally with segmentation-based approaches.

Active learning. A common issue that the recognition problem tackled in this work and actually of many classifier-based learning systems is that they stop learning after a single (extensive) training stage. This poses a serious practical limitation since even though the experts may detect the system's misclassification during its predition stage, the system is unable to learn from its own mistakes. In this sense, less explored learning paradigms, such as on-line and incremental learning can be very valuable. Of particular interest in the scope of expert and intelligent systems is the combination of incremental and active learning (Settles, 2012), so that the system not only can keep learning incrementally from an initial extensive learning stage, but should also be able to, very selectively, ask the experts for the true label of cells and exploit this knowledge to update its model and perform better in the future. The human experts are thus not overloaded with many images to label explicitly. Their intervention is reduced to instances whose ground-truth label the system considers most valuable, thus making the most of human-expert skills and effort. As a longer-term, more ambitious endeavour, the problem can be studied with the perspective of never-ending learning (Mitchell, Cohen, R., Talukdar, Yang, Betteridge, Carlson, Mishra, Gardner, Kisiel, Krishnamurthy, Lao, Mazaitis, Mohamed, Nakashole, Platanios, Ritter, Samadi, Settles, Wang, Wijaya, Gupta, Chen, Saparov, Greaves & Welling, 2018), whose design can be more challenging.

5. Conclusions

This work has shown that white blood cells types can be recognized with reasonable performance by means of local image descriptors and a conventional bag-of-words pipeline, using only gray-level images, without the need of complex and error-prone explicit image segmentation, and without carefully crafted features, which makes this approach robust and much more general. A variety of sampling strategies and sparse detectors have been shown to produce generally satisfactory and very promising performances.

Among the three sparse detectors, oFAST tends to outperform the other two (SIFT and CenSurE). This suggests that visual structures nearby the border of the cell might be slightly more discriminative than its interior (CenSurE) or other sparse points distributed throughout the cell image (SIFT).

As for the sampling strategies, foveal, multi-resolution and spatial pyramids yields similar performance, without a clear

Table 7: Average times to process a single image

Approach		Image size	Time (s)
Gautam et al. (2016)		480 × 640	22
Rezatofighi & Soltanian-Zadeh (2011)		141 × 141	10
Our approach	CenSurE (STAR)	512 × 512	2.7
	LUS		0.3

Table 8: Comparison of segmentation-based and local descriptors-based approaches

Approach	Strengths (+) and weaknesses (−)
Segmentation-based	+ Specific features for the cell or nucleus and cytoplasm regions can be obtained + Meaningful expert-based ad-hoc features can be explored (shapes, geometric ratios, etc.) + In principle, more directly suitable for explainable solutions − Problem-specific features have to be designed and implemented − Correct segmentations are generally hard to obtain and may be initialisation-dependent − Subsequent steps (and in turn, results) rely on robust segmentations
Local descriptors	+ Generic, problem-agnostic methodology; no specific features have to be devised and implemented + Segmentation-free approach; only mild assumptions on cell-background contents + Robust against background contents; useful contextual information may be exploited − Potentially useful features such as shape or size cues are not explictly captured − Sensitive to strong deviations from the mild assumptions − In principle, less suitable for explainable solutions

winner, and outperform the pure uniform-sampling strategies either covering the whole image or a central region of interest. The result for the foveal-like sampling is interesting in that it suggests the importance of taking into account a moderate form of contextual information, i.e. a little of background information might help to discriminate, but too much of it may turn out to be harmful. Furthermore, the good performance of the multi-resolution and hierarchical strategies might relate to their ability to encode useful scale and texture information.

Given the different nature of some detectors and their (somehow) complementary performance, further work might explore some fusion strategies. It might also be investigated whether descriptors which capture color information and (more explicitly) shape and texture cues can provide higher discrimination ability. Another research avenue is the comparison of segmentation-based and segmentation-free approaches as well as smartly combining them to get the benefits of both. Studying incremental and active learning paradigms for this type of intelligent system would increase its utility in real-world medical settings.

Acknowledgments. This work has been partially supported by project P1·1B2014-09 of the *Pla de Promoció de la Investigació* from *Jaume-I University*. It has been carried out with the collaboration of the *Departament of Hematology* of the *Hospital General Universitario de Castellón*.

Sharing statement. The labelled image dataset will be made publicly available or upon request (publication permission and conditions are pending).

References

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

Alpaydın, E. (2004). *Introduction to Machine Learning*. The MIT Press.

Bikhet, S. F., Darwish, A. M., Tolba, H. A., & Shaheen, S. I. (2000). Segmentation and classification of white blood cells. In *International Conference on Acoustics, Speech, and Signal Processing* (pp. 2259–2261).

Bosch, A., Zisserman, A., & Muñoz, X. (2007). Image classification using random forests and ferns. In *International Conference on Computer Vision* (pp. 1–8).

Bouckaert, R. R., & Frank, E. (2004). Evaluating the replicability of significance tests for comparing learning algorithms. In *Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining* (pp. 3–12).

Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews correlation coefficient metric. *PLoS ONE*, *12*.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, *10*, 1895–1923.

Elnakib, A., Gimel'farb, G., Suri, J. S., & El-Baz, A. (2011). Medical image segmentation: A brief survey. In A. S. El-Baz, R. Acharya U, A. F. Laine, & J. S. Suri (Eds.), *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies: Volume II* (pp. 1–39).

Estrada, F. J., & Jepson, A. D. (2009). Benchmarking image segmentation algorithms. *International Journal of Computer Vision*, *85*, 167–181.

Gautam, A., Singh, P., Raman, B., & Bhadauria, H. (2016). Automatic classification of leukocytes using morphological features and Naïve Bayes classifier. In *Region 10 Conference* (pp. 1023–1027).

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

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

Gunning, D. (2016). *Explainable artificial intelligence (XAI)*. Technical Report DARPA-BAA-16-53 Defense Advanced Research Projects Agency, (DARPA).

Guo, S., Huang, W., & Qiao, Y. (2017). Improving scale invariant feature transform with local color contrastive descriptor for image classification. *Journal of Electronic Imaging*, *26*.

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

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In D.-S. Huang, X.-P. Zhang, & G.-B. Huang (Eds.), *International Conference on Intelligent Computing*.

Hiremath, P., Bannigidad, P., & Geeta, S. (2010). Automated identification and classification of white blood cells (leukocytes) in digital microscopic images. *International Journal of Computer Applications*, (pp. 59–63).

Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain? *CoRR*, *abs/1712.09923*.

Howse, J., Joshi, P., & Beyeler, M. (2016). *OpenCV: Computer Vision Projects with Python*. Packt.

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

Jones, E., Oliphant, T., Peterson, P. et al. (2001–). SciPy: Open source scientific tools for Python. `http://www.scipy.org`.

Kamentsky, L. A. (1973). Cytology automation. *Advances in Biological and Medical Physics*, *14*, 93–161.

Krig, S. (2014). Interest point detector and feature descriptor survey. In *Computer Vision Metrics: Survey, Taxonomy, and Analysis* (pp. 217–282). Berkeley, CA: Apress.

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*, 107–123.

Laptev, I., & Lindeberg, T. (2003). Space-time interest points. In *International Conference on Computer Vision* (pp. 432–439).

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2169–2178).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*, 91–110.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta – Protein Structure*, *405*, 442–451.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 1615–1630.

Mitchell, T. M., Cohen, W. W., R., H. J. E., Talukdar, P. P., Yang, B., Betteridge, J., Carlson, A., Mishra, B. D., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E. A., Ritter, A., Samadi, M., Settles, B., Wang, R. C., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., & Welling, J. (2018). Never-ending learning. *Communications of the ACM*, *61*, 103–115.

Nadeau, C., & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, *52*, 239–281.

Nguyen, H. M., Cooper, E. W., & Kamei, K. (2011). Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, *3*, 4–21.

Nowak, E., Jurie, F., & Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *European Conference on Computer Vision* (pp. 490–503).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

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

Rezatofighi, S. H., & Soltanian-Zadeh, H. (2011). Automatic recognition of five types of white blood cells in peripheral blood. *Computerized Medical Imaging and Graphics*, *35*, 333–343.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016* (pp. 1135–1144).

Rosten, E., & Drummond, T. (2005). Fusing points and lines for high performance tracking. In *International Conference on Computer Vision* (pp. 1508–1515). IEEE volume 2.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *European Conference on Computer Vision* (pp. 430–443). Springer.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision* (pp. 2564–2571). IEEE.

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

Sahay, A. (2016). *Data Visualization, Volume I*. Business Expert Press.

van de Sande, K., Gevers, T., & Snoek, C. (2010). Evaluating Color Descriptors for Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*, 1582–1596.

Saphiro, H. M. (2003). *Practical flow cytometry*. Wiley-Liss.

Sarrafzadeh, O., Dehnavi, A. M., Rabbani, H., & Talebi, A. (2015). A simple and accurate method for white blood cells segmentation using k-means algorithm. In *Workshop on Signal Processing Systems* (pp. 1–6).

Settles, B. (2012). *Active Learning*. Morgan & Claypool.

Song, X. B., Abu-Mostafa, Y. S., Sill, J., & Kasdan, H. (1997). Incorporating contextual information in white blood cell identification. In *Advances in Neural Information Processing Systems* (pp. 950–956).

Tuytelaars, T., & Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. *Foundations and Trends® in Computer Graphics and Vision*, *3*, 177–280.

Vedaldi, A., & Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`.

Viola, P. A., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*, 137–154.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., & the scikit-image contributors (2014). scikit-image: image processing in Python. *PeerJ*, *2*, e453.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., & Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference* (pp. 124.1–124.11). BMVA Press.

Wang, Y., & Mori, G. (2009). Human action recognition by semilatent topic models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*, 1762–1774.

Young, I. T. (1972). The classification of white blood cells. *IEEE Transactions on Biomedical Engineering*, *BME-19*, 291–298.

Zhang, J., Marszałek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, *73*, 213–238.

Zhang, S., & Metaxas, D. (2016). Large-scale medical image analytics: Recent methodologies, applications and future directions. *Medical Image Analysis*, *33*, 98–101.