E-ISSN: 3107-5843
International Journal of Advanced Engineering and Management System
Volume 01, Issue 01, April-June 2025, pp. 29 - 41

A Hybrid Deep Learning Framework for Accurate Blood Cell Counting and Classification Using Advanced Segmentation and Feature Extraction

DHANYA. D[1] Electronic and Communication Engineering, Ponjesly College of Engineering Parvathipuram, Nagercoil, Tamil Nadu 629003,dhanyadharmamony@gmail.com

Dr, M.R. GEETHA[2], Head of the Department Electronic and Communication Engineering, Ponjesly College of Engineering, Parvathipuram, Nagercoil, Tamil Nadu 629003, geetha028@gmail.com

Abstract:

Blood cell counting is a crucial process in medical diagnostics for identifying and classifying red blood cells (RBCs), white blood cells (WBCs), and platelets. Accurate counting and classification are challenging due to overlapping cells, noise, and variations in cell shapes. To overcome these issues, we propose a robust approach integrating Segmentation using Canny Edge Detector and Watershed Segmentation. For Feature Extraction, we employ Texture-Based Features using Local Binary Patterns (LBP). Finally, Counting and Classification are performed using Convolutional Neural Networks (CNN). This combined method aims to enhance accuracy and efficiency in automated blood cell analysis.

Keywords: red blood cells, white blood cells, Canny Edge Detector, Watershed Segmentation, Convolutional Neural Networks.

1. Introduction

Medical diagnostic evaluations must include blood cell counting as an essential process which enables healthcare professionals to assess patient health along with providing early detection of various disorders from infections to anemia and blood cancer conditions. Red blood cells (RBCs), white blood cells (WBCs) along with platelets constitute the cell population in human blood through which these cells execute essential roles such as oxygen delivery and immune response and coagulation. Healthcare providers benefit from exact blood cell analysis because these measurements help track patients' medical status and treatment optimization and disease monitoring [1].

The process of blood cell segmentation plays an essential role in automatic blood assessment and researchers have implemented different methods to enhance accuracy levels [2]. The clustering methods K-means and Fuzzy C-means function by pixel groupings through color and intensity yet their performance declines in low contrast areas [3]. Active contour models show the ability to cope with irregular shapes although they remain vulnerable to noise contamination and their initial starting points [4]. Deep learning methods such as CNNs and U-Net deliver strong performance while being robust yet operate efficiently only with extensive labeled training data and substantial computational power.

Medical diagnosis relies on blood cell classification since it enables the proper identification and classification of red blood cells (RBCs), white blood cells (WBCs), and platelets [5]. Computer-based learning systems such as SVM, KNN and Decision Trees need manual extraction of features to identify simple datasets though they fall short when working with complex cell patterns and high-dimensional data. The implementation of hybrid models using several techniques results in better prediction accuracy at the cost of enhanced model complexity which raises the chance of overfitting the data. The present analytical restrictions demand improved efficient classification methods for obtaining dependable blood cell analysis outcomes. The key contribution of proposed method is….

Segmentation Approach:

- Canny Edge Detection and Watershed Segmentation work together to segment blood cells accurately when they overlap each other thus resolving the problems of cell boundary identification and separation.

Advanced Feature Extraction:

- The feature extraction process relies on texture characteristics detected with Local Binary Patterns (LBP) together with cellular dimensions obtained from area, perimeter, circularity measurements to assess RBCs, WBCs, and platelets properties.

Deep Learning for Classification and Counting:

- A Convolutional Neural Network (CNN) deployment serves for blood cell classification to detect RBCs, WBCs and platelets while providing accurate cell counting solutions and minimizing erroneous classifications.

The paper will follow this particular organization in its subsequent sections. Section 2 presents an overview of the literature review. Section 3 describes the proposed work. Section 4 discusses the study's results and assessments. Section 5 includes the final conclusion and plans for further work.

2. Literature review

Foy et al. [6] introduced Physiological inflammation reacts to tissue damage that occurs because of trauma and ischemia along with infection and various pathological causes. Through analysis of hospitalized patient longitudinal clinical measurements, they studied how patients recovered from inflammation that resulted from trauma and ischemia and infection. The method allows successful risk stratification of patients through WBC-PLT trajectories which demonstrate the highest associations and the most consistent recovery patterns compared to all other combinations of test results. The method depends on precise scheduling of inflammation occurrences yet these timelines remain uncertain in specific study groups while it also needs interpolated lab tests which might introduce measurement errors.

Kouzehkanan et al. [7] introduced accurate early identification of peripheral white blood anomalies functions as a fundamental measure for human well-being assessment simultaneously serving to identify various hematologic diseases. A free access database named Raabin-WBC was established for normal peripheral white blood cells featuring approximately 40,000 white blood cell and color spot images. The dataset contains many free-available white blood cell images from normal peripheral blood which significantly increases the diversity of data used for evaluating different machine learning applications such as detection and segmentation. The diversity of the dataset cannot prevent medical device variations including microscopes and cameras from causing potential biases which might reduce model generalization abilities.

3. Proposed method

The blood cell counting procedure along with classification method implements an organized process for improving both precision and operational effectiveness. Microscopic blood images which contain RBCs WBCs and platelets are acquired in Image Acquisition. Gaussian blur noise reduction occurs before moving onto adaptive histogram equalization for contrast enhancement as part of the preprocessing phase. Weak edges are first detected with the Canny Edge Detector then Watershed Segmentation divides the overlapping cells. Local Binary Patterns (LBP) capture texture information among other shape features in the Feature Extraction phase through a process of Texture-Based Features extraction. A Convolutional Neural Network (CNN) enables counting and classification by detecting RBCs WBCs and platelets as well as counting their appearance in the images. The Output and Visualization step displays final results and visual annotations of both count and classification across the input image after Post-processing has eliminated false positives through threshold-based techniques. The combined method works to solve problems with mixed cell boundaries and background problems along with cell shape inconsistencies and it delivers trustworthy blood cell diagnostic results. Figure 1 shows flow diagram of Blood Cell Counting and Classification.

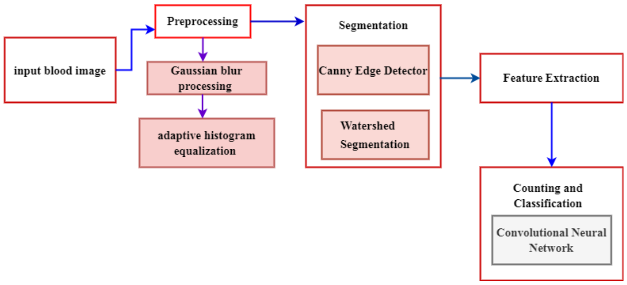


Figure 1: flow diagram of Blood Cell Counting and Classification

3.1. Image Acquisition

Early stages of image acquisition demand sharp micro images to detect all blood cell characteristics required for proper analysis results. A designed system works with multiple blood sample types to meet different clinical needs. Digital image storage allows efficient operations in following stages of automated blood cell counting which provides the basis for obtaining precise and dependable results.

3.2. Preprocessing

Multiple image enhancement methods operate in the preprocessing module to optimize blood smear images before cell detection takes place. Gaussian blur processing serves two functions by smoothing images through noisy reduction without affecting cell boundary preservation to keep irrelevant variations from affecting feature extraction tasks. The adaptive histogram equalization (AHE) method localizes contrast correction to improve visibility of cells that appear faint or with overlap between them. The cell boundaries receive expansion treatment through dilation along with erosion to reduce noise by shrinking structures that do not belong. The combination of preprocessing techniques produces images that are cleaned up with high contrast ready for accurate segmentation in the following stage.

3.3. Segmentation

The segmentation module aims to achieve precise separation of blood cells from background elements and individual cells and from other cells during combined or clustered situations. The identification of cell boundaries through edge tracing represents a main technique in detecting the edges of connected binary image elements. The Canny edge detector functions as a well-known edge detection technique that detects intense spatial gradient transitions which match cell edges. It works in multiple steps:

Canny Edge Detector

- A Gaussian Filter applies to smooth images while removing noise.
- The intensity gradients must be calculated through Sobel operators.
- The edge detection process becomes more precise through Non-Maximum Suppression because it removes pixels which are outside the regional maximum intensities.
- The process of double thresholding detects strong and weak edges depending on their intensity level relationship.

- Through the method of hysteresis edge tracking operators connect weak edges to strong edges whenever the weak edges belong to a continuous boundary.

The gradient G is calculated using the Sobel operators:

$$G = \sqrt{G_x^2 + G_y^2} \qquad (1)$$

The gradient values G_x and G_y define the horizontal and vertical movement direction in the image. Watershed segmentation process defines the boundaries for cells which overlap with each other. Pixel intensities function as a surface topography in this method that shows high-intensity cell borders as ridges and low-intensity cell interiors as basins. The algorithm performs basin flooding from predefined markers that usually represent cell centers until different basins intersect to create boundaries.

Watershed Segmentation

Canny performs edge detection yet it encounters difficulties when detecting overlapping or clustered cells. Watershed segmentation operates on the grayscale image as if it were a topographic surface through two processes: peaks represent boundaries and valleys contain cell interiors. The algorithm approaches each valley from markers which tend to be marked cell centers identified with distance transforms before region expansion leads to cell boundary formation. Watershed segmentation makes use of the following formula to determine the distance transform:

$$D(x, y) = \min_{(x^i, y^{'})} \sqrt{(x - x^{'})^2 + (y - y^{'})^2} \qquad (2)$$

The expression defines the boundary pixels as B while (x, y) represents the pixel coordinates inside the analyzed cell.

In combination:

- Canny successfully identifies distinct edges which benefit the separation of well-segregated cells.
- The watershed algorithm splits overlapping cells through boundaries which it generates based on intensity gradient computations.

3.4. Feature Extraction

The objective of this module involves identifying significant traits from separated cells which enable the classifier to identify red blood cells (RBCs), white blood cells (WBCs) and platelets. There are two principal categories that organize the features.

Geometric Features

The description of cellular geometry involves measurements of shape together with both dimension and architectural details. The key geometric features include:

- The detected cell boundary encompasses Area A which represents pixel numbers.

$$A = \sum_{(x,y)\in cell} 1 \qquad (3)$$

- The Euclidean distance between successive boundary points yields the entire cell boundary length which results in Perimeter P.

$$P = \sum_{i=1}^{N} \sqrt{\left(x_{i+1} - x_i\right)^2 + \left(y_{i+1} - y_i\right)^2} \qquad (4)$$

- **Aspect Ratio (AR):** The ratio of the cell's width ω to its height h.

$$AR = \frac{\omega}{h} \qquad (5)$$

- Circularity measurements (C) demonstrate the closeness of shape to circular form by producing values near 1 which represent highly circular shapes.

$$C = \frac{4\pi A}{P^2} \qquad (6)$$

Texture-Based Features (Local Binary Patterns - LBP)

The Lightness-Based Pixelation technique detects surface texture patterns of blood cells for WBC identification through their distinct pattern compared to smooth RBCs. To determine the LBP value for central pixel I_c an evaluation happens between pixel values of surrounding P neighbors within R radius circle.

$$LBP = \sum_{p=0}^{P-1} s\left(I_p - I_c\right) \times 2^p \qquad (7)$$

$$s(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & if\ x < 0 \end{cases} \qquad (8)$$

The LBP values are then combined into a histogram that represents the frequency of each pattern, creating a compact yet powerful texture descriptor.

Feature Vector

The extracted geometric and texture features get assembled into one single feature vector F:

$$F = \left[A, P, AR, C, LBP_1, LBP_2, \ldots\ldots, LBP_k\right] \qquad (9)$$

The number of LBP histogram bins appears as k in the context of this text.

After calculation the feature vector proceeds to a CNN classification module for training and prediction tasks.

3.5. CNN (Convolutional Neural Network)

A CNN model utilizes its design to extract hierarchical features from blood cell pictures so it can identify RBCs, WBCs and platelets. It consists of multiple layers:

Convolutional Layer

The layer applies filters known as kernels to the image while targeting local image patterns including edges as well as textures and shapes. The calculation for output feature map F follows this formula:

$$F(i,j)=\sum_m\sum_n I(i+m,j+n).K(m,n)+b \tag{10}$$

Where,

- I is the input image.
- K is the kernel/filter.
- b is the bias.
- m and n iterate over the kernel size.

Activation Function (ReLU)

The Rectified Linear Unit creates non-linear components by introducing new relationships:

$$\operatorname{Re}LU(x)=\max(0,x) \tag{11}$$

The network becomes efficient at learning sophisticated relationships by utilizing this feature.

Pooling Layer

Through max pooling the dimensions decrease yet the most essential attributes remain intact:

$$P(i,j)=\max\{F(2i,2j),F(2i+1,2j),F(2i,2j+1),F(2i+1,2j+1)\} \tag{12}$$

Fully Connected (Dense) Layer

The classification process occurs through fully connected layers after flattening the feature maps. The last output y employs the SoftMax activation function for classification procedures.

$$y_i=\frac{e^{zi}}{\sum_j e^{zj}} \tag{13}$$

where z_i is the output of the last dense layer.

Loss Function (Categorical Cross-Entropy)

The training of the CNN occurs through minimizing the cross-entropy loss.

$$L=-\sum_i y_i\log(\hat{y}_i) \tag{14}$$

where y_i is the true label and $\hat{y}_i$ is the predicted probability for class i.

3.5.1. Counting and Classification

The Counting and Classification Module uses CNN-trained output to arrange blood cells into three fundamental categories of RBCs, WBCs along with platelets. The results from the CNN's cell classification enable subsequent use in classifying and counting processes.

Classification

The fully connected layers of the CNN process the feature vectors that convolutional layers learned from individual cells. Using the SoftMax activation function the final classification gets performed.

$$\hat{y} = \arg\max\left(\frac{e^{zi}}{\sum_j e^{zi}}\right) \tag{15}$$

where z_i is the output of the final dense layer, and the index iii corresponds to one of the three classes (RBC, WBC, platelet).

Counting

Following classification occurs the system's iteration through the CNN's predictions allows it to determine the frequency of each detected class.

$$Count(c) = \sum_{i=1}^{N} I(y_i - c) \tag{16}$$

Where,

- N is the total number of detected cells.
- c is the class label (RBC, WBC, or platelet).
- I is an indicator function that equals 1 if the predicted class y_i matches c, and 0 otherwise.

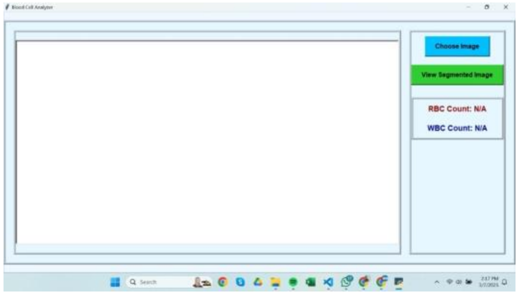
The program generates reports containing final counts for each class that users can view through the interface. The results may include:

- RBC count: Indicator of conditions like anemia or polycythemia.
- Medical practitioners use WBC count to determine infections, inflammation and conditions involving leukemia.
- A proper evaluation of clotting disorders requires measuring platelet count.

4. Results and performance evaluation

This research obtained blood cell pictures from microscopic images that contained red blood cells (RBCs) together with white blood cells (WBCs) and platelets

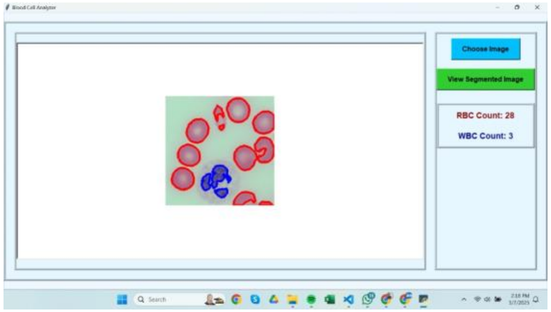
(PLTs). Image processing tasks with OpenCV library included noise removal and contrast enhancement while segmentation occurred through Canny Edge Detector and Watershed Segmentation methods.



RBC Count: N/A
WBC Count: N/A

Figure 2: Graphical User Interface of the Proposed Blood Cell Analyzer

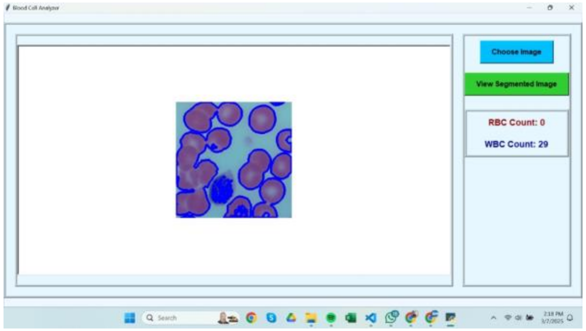
A Figure 2 presents its graphical user interface (GUI) in the provided image. Clients can view blood cell images while using buttons to select and segment pictures and observing RBC Count and WBC Count indicators despite their N/A labels. The platform features an interface made for automated blood cell counting operations which implements image processing technologies for segmenting cells.



RBC Count: 28
WBC Count: 3

Figure 3: Blood Cell Analyzer Displaying Segmented RBCs and WBCs with Count.

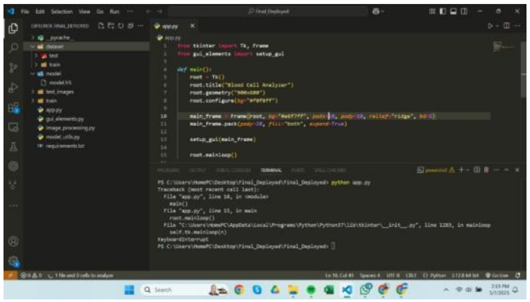
A Figure 3 appears on the Blood Cell Analyzer GUI as displayed in the image. The Blood Cell Analyzer GUI displays red blood cells with red segmentation while showing white blood cells with blue outline configuration after imaging blood samples. To the right side of the screen appear the cellular count results which display RBC Count: 28 and WBC Count: 3. The visualization demonstrates that blood cell segmentation together with counting procedures were successful.



RBC Count: 0
WBC Count: 29

Figure 4: Blood Cell Analyzer Displaying Segmented WBCs with Count and No RBCs Detected

The Figure 4 shows the outcome of a blood sample analysis which detects and segments white blood cells (WBCs) that appear in blue. Through blood analysis the system displays zero red blood cells and thirty-nine white blood cells. The data implies either the segmentation algorithm failed to perform correctly or WBCs comprise most of the image content. The results appear in an easy-to-read format on the right section of the display.

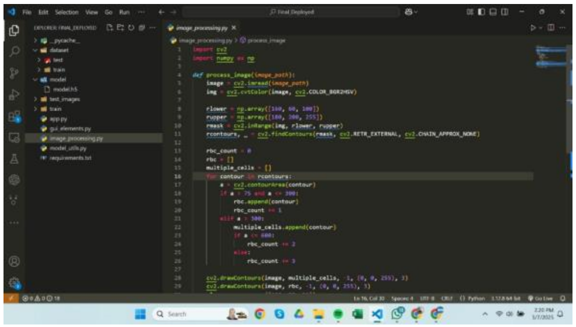


```python
from tkinter import Tk, Frame
from gui_elements import setup_gui

def main():
    root = Tk()
    root.title("Blood Cell Analyzer")
    root.geometry("900x600")
    root.configure(bg="#f0f8ff")

    main_frame = Frame(root, bg="#e6f7ff", padx=10, pady=10, relief="ridge", bd=5)
    main_frame.pack(pady=20, fill="both", expand=True)

    setup_gui(main_frame)

    root.mainloop()
```

```
PS C:\Users\HomePC\Desktop\Final_Deployed\Final_Deployed> python app.py
Traceback (most recent call last):
  File "app.py", line 18, in <module>
    main()
  File "app.py", line 15, in main
    root.mainloop()
  File "C:\Users\HomePC\AppData\Local\Programs\Python\Python37\lib\tkinter\__init__.py", line 1283, in mainloop
    self.tk.mainloop(n)
KeyboardInterrupt
PS C:\Users\HomePC\Desktop\Final_Deployed\Final_Deployed>
```

Figure 5: Python Code for Blood Cell Analyzer GUI using Tkinter

The Blood Cell Analyzer consists of its main interface constructed through the Tkinter library within Python's framework which is visible in the provided Figure 5. The application root contains parameters for title text and screen dimensions together with its background color specification. The Frame widget receives padding and a ridge border with specified background color. The setup_gui () function handles initial element configuration before the main loop of the root window executes the event-driven programming cycle. The execution of the program stopped when a user invoked Keyboard Interrupt from the terminal.



```python
import cv2
import numpy as np

def process_image(image_path):
    image = cv2.imread(image_path)
    img = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    rlower = np.array([160, 60, 100])
    rupper = np.array([180, 200, 255])
    rmask = cv2.inRange(img, rlower, rupper)
    rcontours, _ = cv2.findContours(rmask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)

    rbc_count = 0
    rbc = []
    multiple_cells = []
    for contour in rcontours:
        a = cv2.contourArea(contour)
        if a > 75 and a <= 300:
            rbc.append(contour)
            rbc_count += 1
        elif a > 300:
            multiple_cells.append(contour)
            if a <= 600:
                rbc_count += 2
            else:
                rbc_count += 3

    cv2.drawContours(image, multiple_cells, -1, (0, 0, 255), 3)
    cv2.drawContours(image, rbc, -1, (0, 0, 255), 3)
```

Figure 6: Blood Cell Segmentation and Counting using OpenCV

The image_processing.py script operates through OpenCV and NumPy libraries to process blood cell images according to the displayed Figure 6. Through process image () the function reads an image data, applies HSV color conversion and establishes binary mask thresholds based on upper and lower color range specifications. Contours are detected using cv2.findContours(). The software analyzes red blood cells (RBCs) through their contour area but uses different criteria to detect single cells and cell clusters. The visual identification of cells occurs through cv2.drawContours() drawing of marked cells to support precise counting for both RBCs and multiple-cell clusters.

5. Conclusion

Through the utilization of CNN-based classification and advanced segmentation the Blood Cell Analyzer successfully executes blood cell identification and counting operations. The system accurately detects WBCs through its segmented output and counted WBC population. The RBC segmentation process requires additional optimization because RBCs seem to be missing from the final count. The suggested system indicates potential success for automated blood cell analysis which supports reliable and efficient medical diagnostics.

Reference

[1] "Reassured diagnostics to inform disease control strategies, strengthen health

systems and improve patient outcomes | Nature Microbiology." Accessed: Mar. 07, 2025. [Online]. Available: https://www.nature.com/articles/s41564-018-0295-3

[2] S. Khan, M. Sajjad, T. Hussain, A. Ullah, and A. S. Imran,2021, "A Review on Traditional Machine Learning and Deep Learning Models for WBCs Classification in Blood Smear Images," *IEEE Access*, vol. 9, pp. 10657–10673, doi: 10.1109/ACCESS.2020.3048172

[3] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng, and A. K. Nandi, Sep. 2019, "Super Pixel-Based Fast Fuzzy C-Means Clustering for Color Image Segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 9, pp. 17531766, doi:10.1109/TFUZZ.2018.288908.

[4] Chunming Li, Chiu-Yen Kao, J. C. Gore, and Zhao Hua Ding, Oct. 2008, "Minimization of Region-Scalable Fitting Energy for Image Segmentation," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp.19401949, doi:10.1109/TIP.2008.2002304.

[5] J. Thachil and I. Bates, 2017, "Approach to the Diagnosis and Classification of Blood Cell Disorders," in *Dacie and Lewis Practical Haematology*, Elsevier. 497–510. doi: 10.1016/B978-0-7020-6696-2.00023-0.

[6] B. H. Foy, T. M. Sundt, J. C. T. Carlson, A. D. Aguirre, and J. M. Higgins, Aug. 2022, "Human acute inflammatory recovery is defined by co-regulatory dynamics of white blood cell and platelet populations," *Nat. Commun.*, vol. 13, no. 1, p. 4705, doi: 10.1038/s41467-022-32222-2.

[7] Z. M. Kouzehkanan *et al.*, Jan. 2022 "A large dataset of white blood cells containing cell locations and types, along with segmented nuclei and cytoplasm," *Sci. Rep.*, vol. 12, no. 1, p. 1123, doi: 10.1038/s41598-021-04426-x.