International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

MACHINE LEARNING APPROACH OF AUTOMATIC IDENTIFICATION AND COUNTING OF BLOOD CELLS

FARHEEN ANJUM[1], Dr. GEETHA V[2]

[1]*PG Scholar, Department of Electronics and Communication, UBDT College of Engineering, Davangere, Karnataka, India.*

[2]*Associate Professor, Department of Electronics and Communication, UBDT College of Engineering, Davangere, Karnataka, India.*

Abstract - *One of the important medical test to evaluate overall human health condition is the Complete Blood Cell Count (CBC). Traditionally these blood cells were counted manually by visual inspection or by using the haemocytometer along with some chemical compounds and other scientific equipment's which is a tedious and more time consuming task. To avoid this problem, the proposed work here is the machine learning approach of automatic identification and counting of blood cells (RBC's, WBC's and Platelets) using 'You Only Look Once' in short YOLO object detection and classification algorithm to automatically identify and count the blood cells from the blood smear images. This YOLO framework will be trained the modified Blood Cell Count Data set (BCCD) of blood smear images to automatically identify and count the RBS's, WBC's and Platelets from the blood smear images. Thus overall the computer aided system of counting and detection enables to count the blood cells just in less than a second and this can be useful in the practical applications.*

Key Words: **Blood Smear images, BCCD Set, YOLO, Tiny YOLO, VGG-16 Architecture, Bounding Boxes, Automatic identification and counting.**

1.INTRODUCTION

Complete Blood Cell Count (CBC) is one the important medical test often requested by the medical professionals to evaluate overall human health conditions. Human blood composed of three important cells. Those are Red Blood Cells (RBC's), White Blood Cells (WBC's) and Platelets. RBCs are also called as the Erythrocytes constitute 40-45% of blood and the main function of these cells is they carry oxygen to our body tissues. Normal range of RBC is between 4.7 to 6.1 million cells per microliter of blood. WBCs are also called as Leukocytes constitutes 1% of blood and they fight against infections. Normal range of WBC is 4500 to 11000 cells per microliter of blood. Platelets are also called as Thrombocytes constitutes remaining part of blood and its function is it helps in clotting of blood. As these cells are huge in number traditional method of manually counting of blood cells using haemocytometer and other equipment's is time consuming and erroneous task and is completely depends on the skill of a laboratory analyst. With the development of machine learning techniques, image processing, object detection and classification becoming more accurate and robust. Generally, deep learning methods are being applied in different medical applications like detection of diabetic retinopathy in retinal fundus images, automatic segmentation of left ventricle in cardiac MRI, femoral neck fracture recovery. This model deploys a deep learning based object detection method to detect and count different blood cells from the blood smear images automatically. Among different object detection algorithms such as Region-based fully convolutional network (R-FCN), Region-based convolutional networks (R-CNN), Single Shot Detector (SSD) and You Only Look Once (YOLO) this model will be deployed with YOLO algorithm with VGG-16 architecture, which is three times faster than other algorithm. The YOLO framework will be retrain to automatically identify and count RBC's, WBC's and Platelets from the blood smear images. Verification method has also deployed in order to improve the counting accuracy in order to improve the counting accuracy and to avoid the double count by the framework.

2. RELATED WORK

Generally there are two different methods for automated counting of blood cells those are the image processing and the machine learning techniques. Acharjee et al. [3] The Hough transform is applied with specified diameter to detect the biconcave and oval shape of RBC. Cruz et al. [1] has designed the raspberry-pi based image analysis system for detection and counting of blood components from the microscopic blood images using Hue saturation and value (HSV) thresholding method. Zhao et al. [13] a system for automatic identification and classification system for WBCs using the convolutional neural network (CNN) has been proposed. They detected WBCs from the microscopic images first then CNN was used to detect different types of WBCs. Image processing technique for counting of RBCs along with the identification of normal and abnormal cells has been proposed by Acharya and kumar [10] where they used k-medoids for WBCs extraction and separating RBCs from WBCs using granulometric analysis and counting it using labelling algorithm and CHT transform. Circlet transform for counting of RBCs from grey scale images has been proposed by Sarrafzadeh et al. [11] where they have used the soft thresholding iterative method

for identification and counting. Patch size normalization has been applied on the pre-processed images, and then the RBCs shapes are classified by applying CNN from the microscopic images of sickle cell disease proposed by Xu et al. [15]. A system for classifying five different types of the WBCs using three different classifiers (two SVMs and one CNN) has been proposed by Habibzadeh et al. [2].

A complete different framework has been proposed here which deploys the YOLO for the identification and counting of all three different blood cells. This model does not requires the binary conversation of image and is fully automated, accurate and fast process.

3. MATERIALS AND METHOD

The goal is to use the YOLO algorithm which is an object detection and classification algorithm to detect and count the blood cells directly from blood smear images. The YOLO framework will be trained with the modified configuration and annotated blood cells training images.

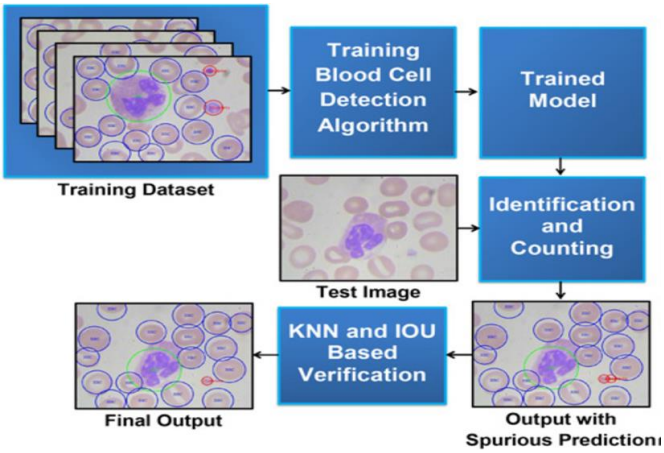

Fig -1: Automatic Blood Cell Detection and Counting System

3.1 Dataset:

The dataset used in this proposed methodology is called as BCCD – Blood Cell Count Dataset that contains the annotated images of blood cells, which are downloaded from the Kaggle website. Kaggle website is publically available community for the machine learning professionals. It helps users for searching, exploring also publishing of the datasets. It is also helpful in building of models based on web environment. Then the data will be splitted into training sets, validation sets. The framework will be trained with training data set by using training blood cell detection algorithm i.e. the YOLO algorithm. The model will be trained with the training images along with their annotation files. After training, the model will be tested by some of the validation datasets. With the identification and counting algorithm the model will automatically detects and counts the three different blood cells (RBC, WBC and Platelets) from the image and gives the output as the cell count number. Sometimes the model will give output with some spurious prediction which means sometimes it will double counts the platelet cells. In order to remove this spurious predication the model uses the k nearest neighbour algorithm along with IoU-Intersection over Union algorithm and gives the final output with the single count of the platelets as shown in the block diagram Fig -1.

3.2 YOLO Algorithm:

'YOLO' abbreviated as You Only Look Once is a real time convolutional deep neural network algorithm which will work for both detection also classification of the objects. This is the most popular algorithm because of its speed and accuracy. This algorithm will takes regression problem as object detection hence providing class probabilities of the detected objects in the image. This algorithm will applies single neural network to whole image. This algorithm will takes the entire image, detects bounding boxes also class probabilities just within one run of the algorithm hence the name 'you only look once'. The bounding boxes predicted will be weighted with their probabilities that are predicted. Excellent speed is it's biggest advantage. This can process upto 45 frames per second and this is incredibly fast compared with other object detection algorithms. The name itself suggest that for an image this algorithm will 'look only once', meaning that to make the prediction this will requires just a single forward pass propagation into the neural network. Let's see the working of YOLO algorithm by taking the example of the below image. First the algorithm will take the entire image and divides the image into grids as shown in Fig -2. Then the responsibility of every grid cell is to predict the bounding boxes. Then for each of the grid in the image, image classification and localization is applied.

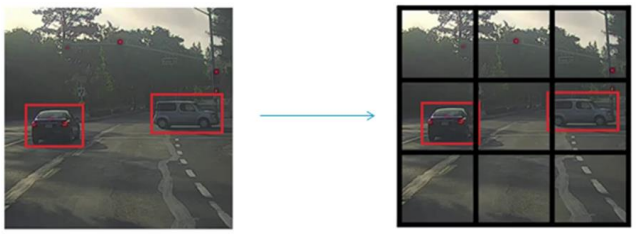

Fig -2: Image into Grids

Here image is divided to 3x3 grid size and three objects need to be detected from the image those are car, pedestrian, and motorcycle. The data which is labelled should be passed to the framework for training it. Thus each and every grid will have the y label is a vector of eight that is dimensional given as,

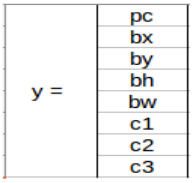

Fig -3: y – label

Here pc defines the probability of the presence of the object, bx, by, bw, bh are the bounding box specifications and c1, c2, c3 are the classes i.e. motorcycle, car, pedestrian. Lets take first grid from the image. Since there is no object is present in the grid hence pc=0 as shown in below Fig -4.

Fig -4: y – label for 1st grid.

The The '?' mark sign in the y-label specifies that it doesn't atter what the bx, by, bh, bw contains as there is no object is present in this grid. Take another grid which contains an object and lets see its y-labelling as shown in Fig -5. How does the algorithm will decide whether there is object present in the grid cell? In this example two objects are present i.e. two cars. The algorithm will take the midpoint of those two objects and assign those objects to the cell containing those objects midpoint. If grid contains the centre coordinate of a particular object then only that object is considered to be present in that particular cell or grid.

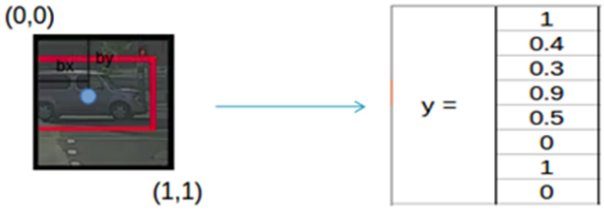

Fig -5: Labelling by the YOLO algorithm.

In this figure since object is present and object is a car hence class c2 = 1 and c1=c3=0. For this example that may be around bx= 0.4 and by = 0.3.The bounding box specification bx, by represents the x and y coordinates of the midpoint of the object related to a particular cell, bh and bw are given by the following formulae,

$$bh = \frac{\text{Height of Bounding box}}{\text{Height of corresponding grid cell}}$$

$$bw = \frac{\text{Width of Bounding box}}{\text{Width of corresponding grid cell}}$$

For this example bh and bw may be around 0.9 and 0.5 respectively. Thus in this way the YOLO algorithm will does the labelling of each grid cells present in the image.

Once the class probabilities have been predicted, intersection over union (IoU) is the next step of algorithm. Depending upon the class probabilities the YOLO algorithm will calculates too many anchor boxes. For eliminating of these unnecessary anchor boxes the algorithm will calculates the area of intersection over union as shown in below Fig -6.

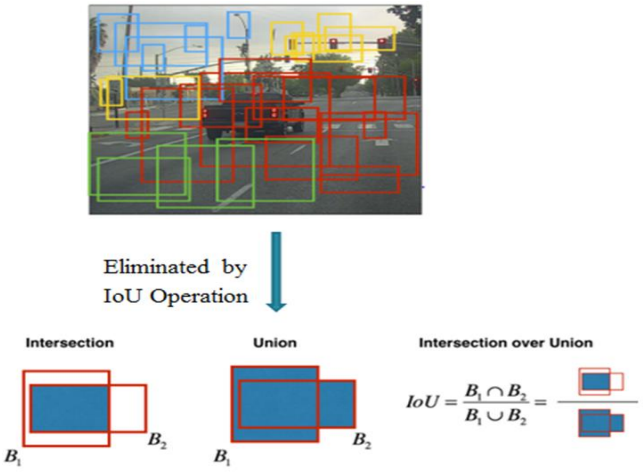

Fig 6: IoU Operation.

YOLO algorithm will calculates this area for all other bounding boxes with respect to the one having highest class probability bounding box. Then boxes having probabilities greater than the threshold all those bounding boxes will be rejected. This is because the algorithm will assumes same object will be covering by those bounding boxes but it will be having the low class probability for the same hence that boxes will be rejected. In this way the algorithm will remove the unwanted bounding boxes.

For the improvement of the accuracy of this YOLO algorithm there is one more technique called Non-Max Suppression. Major problem of object detection algorithms is rather than detecting the objects only one time it will be predicted so many times. In the Fig -7, same object (car) have been detected multiple times, this will be demolished by this Non-Max Suppression process until it will left with a single detection per object. First the algorithm will look at all the probabilities associated with each detection, and then it will selects the highest one from it and take it as output. Then it will look at all other boxes associated with the current selected output box and suppress all those boxes whose intersection over union is greater than the selected output box. All these above steps will be repeated until it will left with the final bounding boxes by selecting the correct bounding boxes and supressing the unnecessary ones. This is the operation of Non-Max Suppression. It will takes the boxes having the probability maximum and suppress the closed ones which having the Non – Max probability.

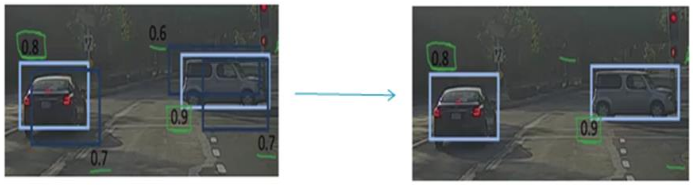

Fig -7: Non-Max Suppression operation.

3.3 Proposed algorithm:

This proposed model includes the modified training model where the final convolutional layers will be changed for three outputs (RBC, WBC and Platelet), with an appropriate threshold identifying the blood cells and counting them from their labels. There is no misunderstanding in this framework among the different cells like recognizing WBCs as RBCs, or RBCs as platelets, so on. There are cases where the model may predict and counts same platelet twice. It is because of the presence of same platelet at two corresponding grids so it will double counts the same platelet. This problem can be resolved by applying k-nearest neighbor (KNN) and IoU algorithms in each platelet. First the KNN is applied to every platelet for the determination of the nearest platelet. After that by applying Intersection over Union (IoU) among two of the platelets and calculates the range of the overlap between those two platelets. In this proposed model 10% of overlap between those two platelets is allowed and taken that as a threshold. If the overlap is greater that threshold then that platelet will be ignored as double count and thus it will get rid of spurious counting as shown in figure below. In this way this proposed model will detects the blood cells automatically from the blood images and gives their count.

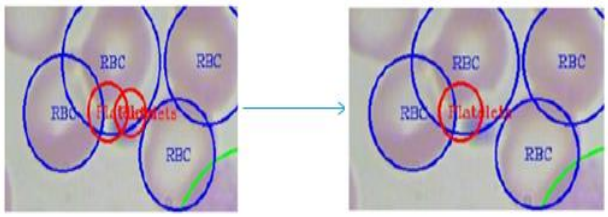

Fig -8: Discarding of Spurious prediction

Following are the algorithm steps used in this model,

1. Importing Test images
2. Importing trained weights
3. Prediction by using the object detection algorithm (YOLO)
4. Removing the fake predictions using appropriate threshold values.
5. Initializing the blood cell labels, i.e.,

   RBC = 0

   WBC = 0

   Platelet = 0

Loop over for all the cell predictions

6. for i in the range (length of prediction) do

   (x1, y1) = Bounding box's left top coordinate

   (x2, y2) = Bounding box's right bottom coordinate

   Label = label of the cell

Checking for spurious overlapping platelet prediction

7. if label == 'Platelet' then
8. Finding nearest platelet using k-nearest neighbor algorithm
9. Applying intersection over union (IoU) to calculate the overlap between detected platelet and nearest platelet

Allowing only 10%of overlap

10. If overlap > 10% then
11. Continue
12. end if

Cell counting

13. if label == 'RBC' then RBC → RBC+1
14. else if label == 'WBC' then WBC → WBC+1
15. else if label == 'Platelet' then Platelet → Platelet+1
16. end
17. Saving the output image.

3.4 FLOWCHART

The flowchart for the proposed framework is given in figure 9,

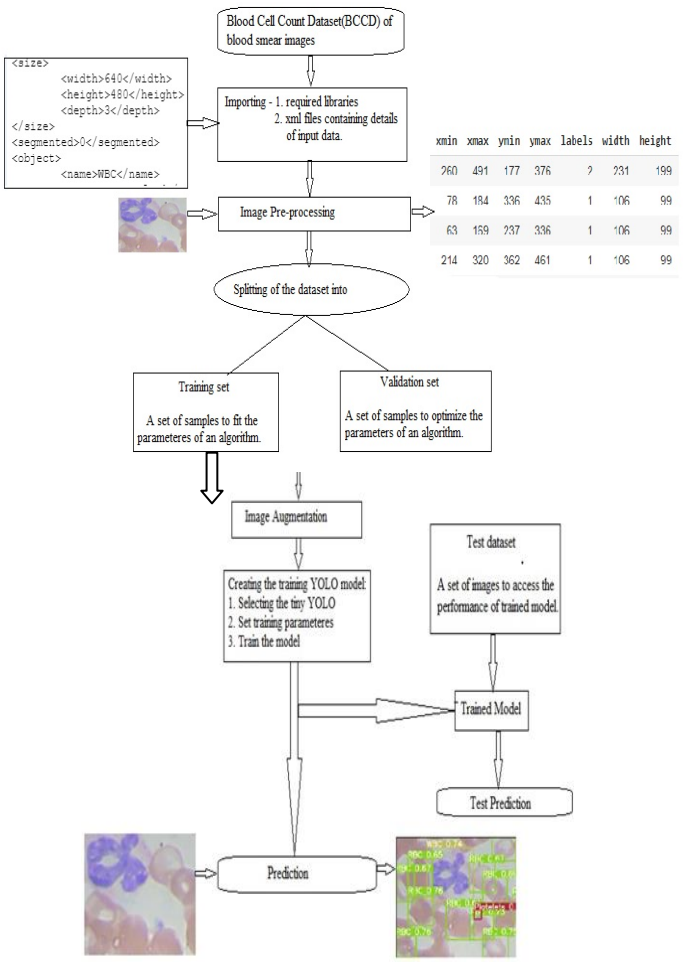

Fig -9: Flowchart of proposed algorithm.

3.5: Prediction of Proposed Model:

The proposed model will automatically identifies and count the RBCs, WBCs and Platelets from the blood smear images. The model has been tested with the 60 tests data set where the ground truth is known. First this model will counts the different cells present in the validation sets of smear image with different confidence truths. It has been noted that the threshold plays an important role in the YOLO algorithm as this algorithm uses this threshold for the grid cells not for the entire image. Grid cells don't contain the cells with low confidence. Hence by choosing appropriate confidence threshold we can get rid of the redundant and spurious predictions.

Then the average absolute error between ground truth and the estimated number cells in validation set is calculated. The minimum average absolute error values for each type of the cell is realized and select the confidence values for the identification process of the blood cells. The error can be computed as,

$$\varepsilon^{\text{cell}} = \frac{1}{N}\sum_{i=1}^{N} |\chi^{(i)}_{\text{groundtruths}} - \chi^{(i)}_{\text{estimated}}|$$

Where, cell indicates the type of cells - RBC, WBC, or platelets, N is the size of validation dataset, x is the number of cells, ϵ is the average absolute error value for the particular cell. In this way by choosing appropriate confidence threshold the model will automatically identifies the different type of blood cells and counts the same present in the blood smear images.

4. CONCLUSION

In this research based project paper, machine learning approach for the identification and counting of blood cells automatically using YOLO framework has been presented, which is a deep neural network object detection and classification algorithm. This framework will uses the blood smear image datasets as its input and gives the blood cell identification and its count as the output. It automatically detects the three different types of blood cells and gives the count of the respective blood cells. This model does not misinterpret between the types of blood cells and produces the accurate results. But sometimes it may double count the platelet cells which can be further removed by this algorithm by using the verification model. Thus the accuracy of the model is improved by employing the KNN – K nearest neighbor algorithm along with the IoU - Intersection Over Union operation to avoid the repeated counting of single object by the algorithm and gives the correct prediction with the single count of the cells. This model has been tested on various validation datasets of images of blood and it had outputted satisfactorily. Thus by its detection and the accuracy performances this model has gains potential to ease blood cell detection and counting manually which can be used in the practical applications.

REFERENCES

[1] Cruz D., Jennifer C., Valiente L.C., ET AL.: 'Determination of blood components (WBCs, RBCs, and platelets) count in microscopic images using image processing and analysis'. Proc. Int. Conf. on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), December 2017.

[2] Habibzadeh M., KrzyÅijak A., Fevens T.: 'White blood cell differential counts using convolutional neural networks for low resolution images'. Proc. Int. Conf. on Artificial Intelligence and Soft Computing, 2013, pp. 263–274.

[3] Acharjee S., Chakrabartty S., Alam M.I., ET AL.: 'A semiautomated approach using GUI for the detection of red blood cells'. Proc. Int. Conf. on Electrical, Electronics, and Optimization Techniques, 2016, pp. 525–529.

[4] Lou J., Zhou M., Li Q., ET AL.: 'An automatic red blood cell counting method based on spectral images'. Proc. Int. Congress on Image and Signal Processing, BioMedical Engineering and Informatics, October 2016, pp. 1391–1396.

[5] Islam M.T., Aowal M.A., Minhaz A.T., ET AL.: 'Abnormality detection and localization in chest X-rays using deep convolutional neural networks', arXiv preprint arXiv:1705.09850v3, 2017.

[6] Avendi M.R., Kheradvar A., Jafarkhani H.: 'A combined deep-learning and deformable model approach to fully automatic segmentation of the left ventricle in cardiac MRI', Med. Image Anal., 2016, 30, pp. 108–119.

[7] Gulshan V., Peng L., Coram M., ET AL.: 'Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs', J. Am. Med. Assoc., 2016, 316, (22), pp. 2402–2410.

[8] Ren S., He K., Girshick R., ET AL.: 'Faster R-CNN: towards teal-time object detection with region proposal networks', IEEE Trans. Pattern Anal. Mach. Intell., 2017, 39, (6), pp. 1137-1149.

[9] Redmon J., Divvala S., Girshick R., ET AL.: 'You only look once: unified, real-time object detection'. IEEE Conf. on Computer Vision and Pattern Recognition, December 2016.

[10] Acharya V., Kumar P.: 'Identification and red blood cell automated counting from blood smear images using computer-aided system', Med. Biol. Eng. Comput., 2018, 56, (3), pp. 483–489.

[11] Sarrafzadeh O., Dehnavi A.M., Rabbani H., ET AL.: 'Circlet based framework for red blood cells segmentation and counting'. IEEE Workshop on Signal Processing Systems, December 2015.

[12] Kaur P., Sharma V., Garg N.: 'Platelet count using image processing'. Int. Conf. on Computing for Sustainable Global Development (INDIACom), March 2016.

[13] Zhao J., Zhang M., Zhou Z., ET AL.: 'Automatic detection and classification of leukocytes using convolutional neural networks', Med. Biol. Eng. Comput., 2017, 55, (8), pp. 1287–1301.

[14] Habibzadeh M., Jannesari M., Rezaei Z., ET AL.: 'Automatic white blood cell classification using pre-trained deep learning models: ResNet and inception'. AIP Conf. Proc., 2017, vol. 1883, pp. 1–8.

[15] Xu M., Papageorgiou D.P., Abidi S.Z., ET AL.: 'A deep convolutional neural network for classification of red blood cells in sickle cell anemia', PLoS Comput. Biol., 2017, 13, (10), pp. 1–27.

[16] He K., Zhang X., Ren S., ET AL.: 'Deep residual learning for image recognition'. IEEE Conf. on Computer Vision and Pattern Recognition, December 2016.

[17] Szegedy C., Vanhoucke V., Ioffe S., ET AL.: 'Rethinking the inception architecture for computer vision'. IEEE Conf. on Computer Vision and Pattern Recognition, December 2016.

[18] Redmond J., Ali F.: 'YOLO9000: better, faster, stronger', arXiv preprint arXiv:1612.08242, 2016.

[19] Simonyan K., Zisserman A.: 'Very deep convolutional networks for large-scale image recognition'. Proc. Int. Conf. Learning Representations, San Diego, 2015, pp. 1–14.

[20] Howard A.G., Zhu M., Kalenichenko B.C.D., ET AL.: 'Mobilenets: efficient convolutional neural networks for mobile vision applications', arXiv preprint arXiv:1704.04861, 2017.