HemeAI

Amrita Ballurkar
amritab@vt.edu
Virginia Polytechnic Institute
Blacksburg, Virginia

Bhargav Iyer
biyer108@vt.edu
Virginia Polytechnic Institute
Blacksburg, Virginia

Ritvik Prabhu
ritvikp@vt.edu
Virginia Polytechnic Institute
Blacksburg, Virginia

ACM Reference Format:
Amrita Ballurkar, Bhargav Iyer, and Ritvik Prabhu. 2023. HemeAI. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 ABSTRACT

This project discusses an approach to automate the process of complete blood counts (CBCs) and peripheral blood smears. CBCs and peripheral blood smears are critical lab tests that doctors rely on to provide crucial information about the cellular components of blood. The process of manually reviewing cells after an abnormal CBC can lead to human error and long wait times for results. We propose using YOLO, a computer vision model, to classify the blood cells and identify abnormal cells within a blood sample. The approach adopted was to first use YOLO to perform a CBC by counting and classifying each cell. If there are abnormalities in the ratios between the cells, we will once again use YOLO to identify and count specific white blood cells and abnormal red blood cells, and diagnose from five diseases: anemia, thrombocytopenia, basophilia, eosinophilia, and leukemia. The first YOLO model, which was used for the preliminary analysis, yielded an F1 score of 0.85, while the second model, which was used to diagnose the disease, yielded an F1 score of 0.75.

2 INTRODUCTION

2.1 Motivation

Since the review of blood cells after abnormal CBCs are completed manually, they are subject to human error which can lead to dangerous false positives or false negatives. In addition to this, the review process can take a few days to complete, which can be critical when treating certain diseases. Furthermore, people in rural and developing areas have a hard time accessing this test and sometimes have to wait a long time to get their results.

2.2 Challenges

There are several potential challenges and risks associated with attempting to solve this problem. The first challenge is that the model is limited by the data that is used to train it. Cells that were stained in certain ways may not be able to be used to make accurate predictions as the model will be largely trained on Romanowsky-stained cells. Another challenge is that there will likely be limitations on the amount of data that is available for abnormal cells. Certain blood disorders are not that common and the images of the affected cells may not be open source for privacy reasons. Another limitation is that diseases that require the exact number of blood cells to be counted in the entire sample will probably not be able to be diagnosed. This is because of the sheer number of pictures that would have to be uploaded which would test computational limits.

2.3 Current Solutions

In 2019, researchers from the Bangladesh University of Engineering and Technology developed a machine learning approach to the CBC process. [Alam and Islam 2019] The algorithm created by this group used a combination of KNNs, SVMs, and CNNs in order to build an object detection and classification algorithm. The algorithm was relatively accurate (between 86-96%) and computationally efficient. This approach classifies the blood cells into 3 major types, which gives it the ability to perform CBCs on normal blood cells. However, there are some limitations to this method as it would not replicate the manual review process necessary to complete a CBC on abnormal blood cells.

The researchers for [Alam and Islam 2019] had used a Tiny YOLO which uses a combination of KNNs, SVMs, and CNNs. Tiny YOLO is a computer vision model. This project will use the latest version of YOLO (YOLOv8) instead of Tiny YOLO.

3 RELATED WORK

3.1 Literature Survey

Complete Blood Counts can be used to diagnose many diseases including anemia, blood cancers, and infections. Recently, it has been used as a tool to diagnose cases of COVID-19.[Pozdnyakova et al. 2020] Most people who have COVID-19 and other infections that are similar have abnormal white blood cell counts. Specifically, they have abnormally high neutrophil counts and abnormally low lymphocyte counts. It is important to determine the exact ratios of the white blood cells in the manual review process as that helps narrow down potential diagnoses. This paper highlights the criteria to determine if a CBC is abnormal in cases of suspected infection, and how the manual review process is crucial to the diagnostic process.

Deep learning is a powerful tool for cell image classification, however, past methods have not been able to achieve high accuracies when tested in real life.[Xu et al. 2022] Some progress has been made in the different fields of cell image analysis, including segmentation, classification, and tracking. For example, image segmentation models are moving away from anchor-based methods towards region-based methods that break images into places where cells may be and places where cell boundaries may be. Image

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

classification methods such as molecular embeddings and encoder-decoder networks have also shown promise. A thorough comparison of these methods found that in general they were more efficient than manual identification and are sometimes more accurate than manual methods.

In a paper called "A deep learning-based algorithm for 2-D cell segmentation in microscopy images", the authors described the deep learning model that they trained to segment images of cells. [Al-Kofahi et al. 2018] The deep learning algorithm that was developed was built with the intention that it could assist in isolating the cells for the purpose of identifying them later. The deep learning model was built using MXNet and UNet-like architecture. The algorithm that was developed was able to isolate whole cells and/or the nuclei of the cells. One of the drawbacks to the model that was developed was that it was not generic. This means that this model only works with specific staining techniques that highlight the nuclei of the cells.

In another paper called "An application of machine learning to haematological diagnosis", a machine learning approach to diagnosing blood disorders using hematological parameters is described.[Gunčar et al. 2018]. In this case, the parameters include some limited information about the CBCs that were run and several other lab tests. Notably, the authors used 3 different ML techniques (SVMs, Random Forest, and Smart Blood Analysis) to diagnose patients. The algorithm developed was found to be more accurate than some hematology specialists. In the future, the authors hope that machine learning can help physicians make more accurate diagnostic decisions using data from CBCs and other lab tests, especially in cases where a patient might have a rarer disease that might not first come to mind.

The YOLO algorithm is a method for image segmentation that is commonly used for image processing for bioinformatics. [Jiang et al. 2022] The YOLO (you only look once) algorithm has a very high performance when used for object detection. It is very fast and accurate. It is composed of Region Proposal Networks (RPNs) to do the object detection. When more complex tasks need to be done, different layers can be added to the model in order to achieve this. For example, a classification layer can be added in order to make image segmentation and classification a one step process.

3.2 Limitations of existing approaches

The paper mentioned above has a key limitation. It only replicates a CBC and not the peripheral blood smear (manual review) that needs to occur if the CBC is abnormal. This means that the model developed in this paper can only be used as a preliminary test. In order to conduct a more thorough analysis and detect diseases, a peripheral blood smear would still need to be conducted if the CBC is abnormal which may result in human error.

4 PROPOSED APPROACH

The problem we solved for the first milestone was to classify different types of cells within an image of blood cells. These images contained white blood cells, red blood cells, and platelets. These cells needed to be classified in order to perform a CBC. If the CBC turned out to be abnormal, the ratio of the cells along with images of any abnormal calls needed to be factored into the identification of the disease.

For milestone 2, we determined what disease the blood cell might have shown. We explored 3 different pretrained networks to accomplish this task. These were:

- YOLOv8
- VGG 16
- Resnet-18

The purpose for the choice of those Neural Networks is because of their popularity in Object Detection algorithms and ease in implementation of transfer learning on these architectures.Since YOLOv8 provided us with the best results, we decided to use that model for milestone 2

This data contained classes such as abnormal red blood cell, band neutrophil, basophil, eosinophil, erythroblast, lymphocyte, monocyte, myelocyte, neutrophil, and segmented neutrophil. Using the information obtained from milestone 1 and the presence of the classes in milestone 2, we made a diagnosis from one of five diseases (including their variations).

- Anemia
- Thrombocytopenia
- Basophilia
- Eosinophilia
- Leukemia

We have narrowed this list down from over twenty possible diseases due to the time taken in collecting, cleaning, and annotating all 20+ classes.

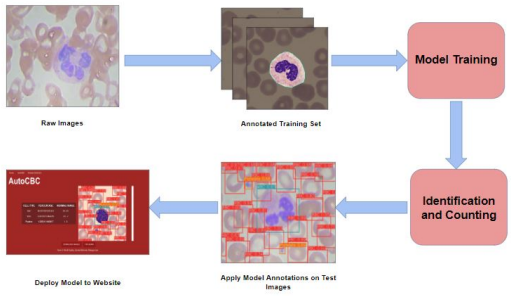

Figure 1: A diagram of the methodology followed by each milestone

The project had a pipeline that connected Milestone 1 (blood cell classification), Milestone 2 (disease detection), the docker file, and an application. Both Milestone 1 and Milestone 2 were done using different aspects of YOLOv8. Flask was the framework used for the backend server, and NextJS was used for the front end.

The dockerfile that was used was built using multistage builds to deal with the frontend and backend. The frontend installed npm and copied the package-lock.json and package.json, which had information regarding the version, dev, build, and how to start it. The backend used python version 3.8.13. The backend portion installed the required python files as well as installed all the required packages for app.py. After that, the frontend build was copied into the backend portion to make sure these two portions listened to each other and created a functioning image.

5 EXPERIMENT EVALUATION

5.1 Data-set Exploration

The first milestone's dataset contains images that contain different cells such as RBCs, WBCs, and platelets. According to Alam et al. "The dataset includes 100 images of resolution 3246×2448 acquired by Nikon V1 camera mounted on a Nikon ECLIPSE 50i microscope with a magnification of 100×." [Alam and Islam 2019]

These images were used in the paper "Machine learning approach of automatic identification and counting of blood cells" [Alam and Islam 2019]. The dataset is already split into training (300 images) and testing (64 images). Additionally, the dataset has been manually annotated to indicate the position of the RBCs, WBCs and platelets as seen in the image below:

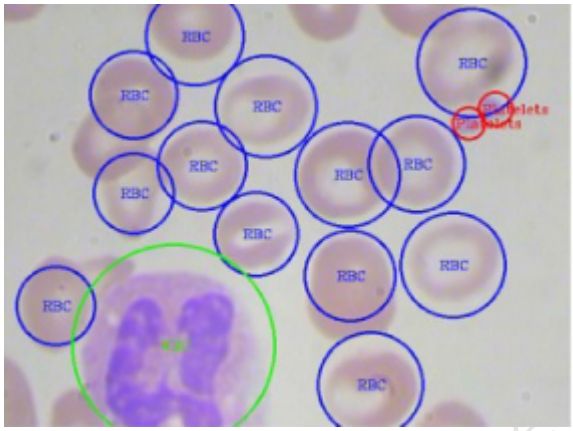

Figure 2: An example of the annotated dataset

These images were annotated in order to train the YOLOv8 model. The annotations were stored as an XML file (indicating the x and y positions of the different objects in the image) by the authors of the paper. A script was created that converted these annotations into a plain-text format that is compatible with YOLOv8 architecture.

The dataset can be found here: GitHub.

The dataset used in the second milestone was obtained from here: source. As mentioned in **??**, this dataset was annotated using Roboflow. 1,361 images were annotated with the following classes:

- abnormal red blood cell
- band neutrophil
- basophil
- eosinophil
- erythroblast
- lymphocyte
- monocyte
- myelocyte
- neutrophil
- segmented neutrophil

The distribution of the annotations is uniform (small standard deviation) to ensure that all classes are represented well in the training dataset. The dataset is split into 945 training images, 151 testing images, and 265 validation images.

6 PREDICTIVE MODEL

6.1 YOLO: You Only Look Once

The model being used is called YOLOv8. YOLO stands for "You Only Look Once" and it is a computer vision model. On top of the pre-trained object detection model, we will be retraining it such that it can be used to classify different blood cells in images.

YOLO is a computer vision model which has a backbone that uses 5 part steps. YOLOv8 is an anchor-free model that predicts directly the center of the object instead of the offset known as the anchor box. In CNN, anchor boxes are used to predict multiple objects within a picture. These are essentially the "bounding box" surrounding the different objects in a picture. YOLOv8 continually segments 5 times for a specific part of an image. This is used as its backbone in order to give a specific value to that area. It essentially works as a way to first value gotten, split the image, then comes a bottleneck when it detects an object or not which uses the value based on those sets of pixels, then comes concatenating the image, with a final check of the value within the different pixels. The detection of the object is done by checking whether the values gotten through the image segmentation match the object it is looking for. Then the head is used to detect between these 5 layers of segmentation to determine if there is an image.[sol [n.d.]]

6.2 RetinaNet

RetinaNet is a powerful object detection algorithm that uses deep neural networks and feature pyramids. It consists of a backbone network, a feature pyramid network, and two task-specific subnetworks for classification and regression. The feature pyramid representation allows for the detection of objects of different sizes, making it more accurate than other object detection algorithms.

The advantage of using RetinaNet is the way it handles class imbalances. The model does so by using a focal loss function. This function assigns higher weights to misclassified examples, helping the model focus on difficult examples and improving detection accuracy. Despite its high accuracy, RetinaNet's complex architecture makes it computationally expensive. [Tan et al. 2021]

6.3 ResNet-50

ResNet-50 is an image classification algorithm that uses a 50-layer deep residual neural network. It is a variant of a convolutional neural network and has 48 convolutional layers, 1 MaxPool layer, and 1 average pooling layer. Historically, it has been used because it was much faster than other deep-learning image classification algorithms available at the time. [Sarwinda et al. 2021]

6.4 AutoCBC Results

We can use metrics such as Precision and Recall to interpret the performance of the model at various confidence levels. Precision refers to the accuracy of positive predictions and Recall measures the completeness of positive predictions i.e. number of true positive predictions.

However, for our project, we use F-1 scores as they can be helpful in determining the confidence that balances the precision and recall values for the given model, hence is a good indicator of the overall model performance. [Lebiedzinski 2021]

In our project we noticed that as the confidence level increases, the precision value goes up and the recall value reduces as seen in the graphs below:

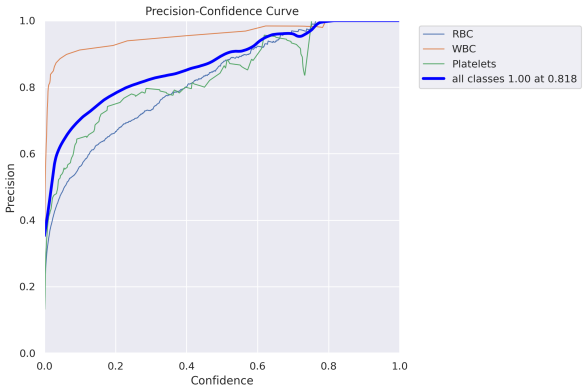

Figure 3: Precision Curve from training the model

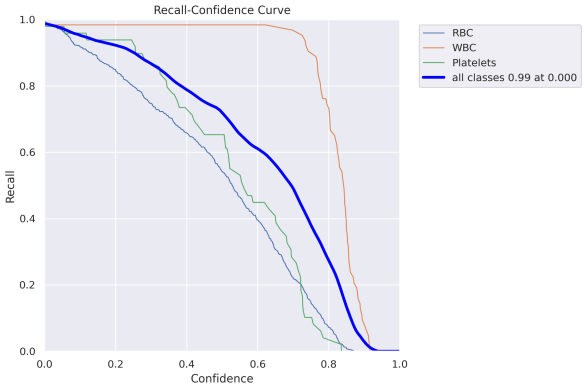

Figure 4: Recall Curve from training the model

We can combine the two metrics and obtain the F1 curve as shown below:

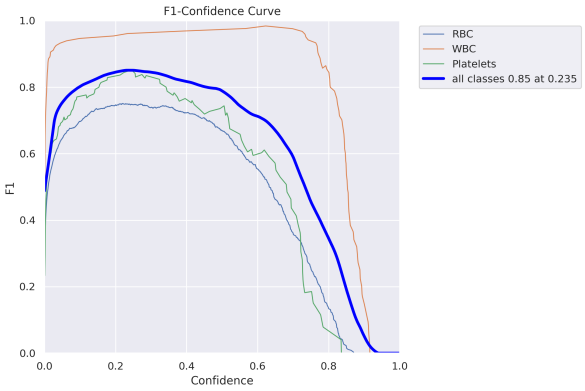

Figure 5: F-1 Curve generated from training the model

From the F1 curve, the confidence value that's optimal is about 0.25. Hence, it might be a good idea to select a confidence level of 0.25 as it returns an F1 score of 0.85, which is the highest produced by the model.

Furthermore, we use a confusion matrix to identify the ratio of true positive, true negative, false positive, and false negative cases during the training of the model. The confusion matrix obtained from the model has been shown below:

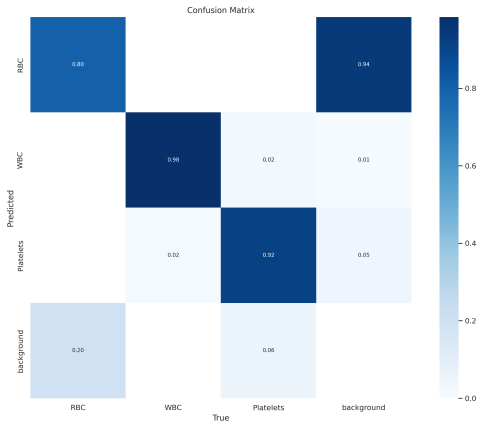

Figure 6: F-1 Curve generated from training the model

From the above figure, we can conclude that the model performs really well. The ratio of true positives for RBC, WBC, and platelets is 0.80, 0.98, and 0.92 respectively. However, we are alarmed by the False Positive Ratio for Background, since the model predicts RBC for the background relatively frequently (ratio = 0.94).

We finally tested our model on a random image (which the model was not trained on) to gauge the model's performance. It does not have much of a bearing on the evaluation, however, it helps us visualize the results of the model. An example output of the model is shown below:

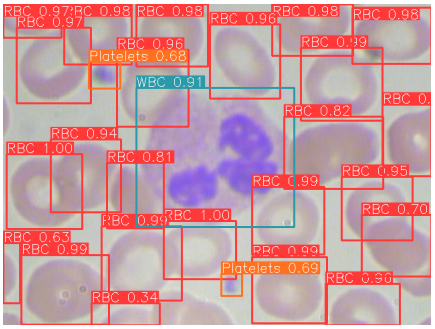

Figure 7: An example of the annotated output from the model

As you can see, the model has been able to predict every cell correctly. In fact, the model has been able to also correctly identify and subsequently predict overlapping cells as two separate entities.

6.5 Disease Detection Results

6.5.1 ***YOLO:.*** Using Precision and Recall curves we can determine the performance of the model at different confidence levels. Like stated in the previous section, Precision and Recall measures the completeness of positive predictions (false positives). This project's YOLO model uses F1 scores to balance Precision and Recall to become a good indicator of model performance.

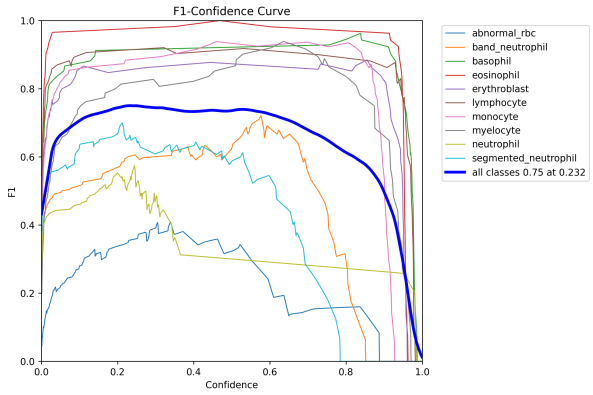

Figure 8: F1 curve of Disease Detection model using YOLO

This model gave our best results of an F1 score of 0.75 when its confidence is 0.25. Thus the confidence was set to 0.25 to give the best prediction to the model.

Then looking at the confusion matrix, it is possible to determine true positives, false positives, false negatives, and true negatives. Using this information it is possible to determine how well our model performed and see where it can be improved upon.

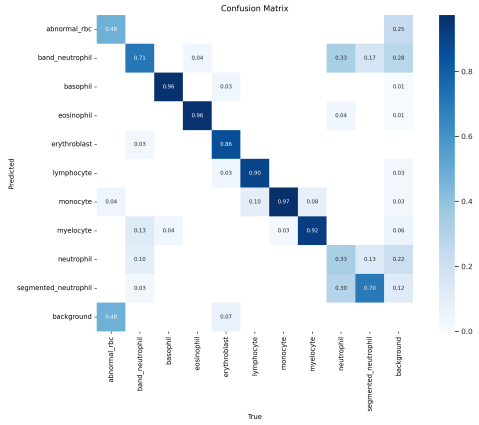

Figure 9: Confusion Matrix of Disease Detection model using YOLO

The confusion matrix showed that the model performed generally well. The confusion matrix showed the ratio of true positives for each specific WBC and abnormal RBC are:

- abnormal RBCs: 0.48
- band neutrophils: 0.71
- basophils: 0.96
- eosinophils: 0.96
- erythroblasts: 0.86
- lymphocytes: 0.90
- monocytes: 0.97
- myelocytes: 0.92
- neutrophils: 0.33
- segmented neutrophils: 0.70

Between looking at the Band Neutrophils, Segmented Neutrophils, and Neutrophils ratios in the confusion matrix, it was clear that the model was having trouble distinguishing the three of them together, given its ratio of true positive to false positive for those three values. By also looking at the confusion matrix, it is possible to see that the model had difficulty in detecting abnormal red blood cells. Abnormal RBCs had a ratio of 0.48 for true positives giving some concern to that given that there is a false positive ratio of abnormal RBCs with the background. This further reinforces the previous statement of the model having difficulty with detecting RBCs.

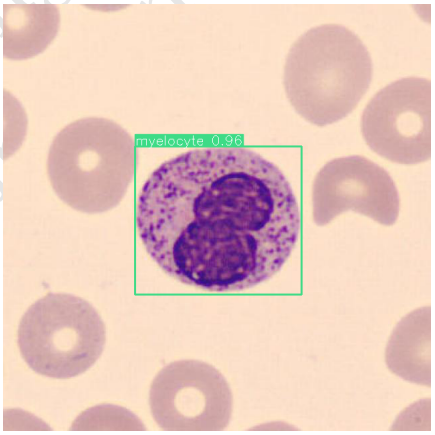

Figure 10: Example of annotated myelocyte output using YOLO

The image helps visualize the output of the model. The image above is an image of a myelocyte. This model correctly predicted the image as a myelocyte. However it did not notice the abnormal red blood cell next to it indicative of the models problem with abnormal RBCs.

6.5.2 ***RetinaNet:*** Unfortunately, we did not obtain any results using RetinaNet. The training time was too long, and we were limited by the number of GPU accelerators available. The model failed at epoch 53, and we were unable to obtain any further progress. The mAP for each class at the last epoch were as follows: Monocyte with 0.99, Segmented Neutrophil with 0.67, Abnormal RBC with 0.36, Lymphocyte with 1.00, Neutrophil with 0.46, Band Neutrophil with 0.54, and Myelocyte, Basophil, Eosinophil, and Erythroblast with 0.00. Considering the fact that this was only about a quarter of the way towards complete training, these results are impressive and show some promise.

6.5.3 ***ResNet 50:*** The ResNet 50 model was trained for 100 epochs. The overall accuracy was 53%. The accuracies for individual classes of cells were as follows: monocyte: 67%, lymphocyte: 100%, eosinophil: 50%, neutrophil: 0%. The model frequently confused neutrophils with other cells and was not able to identify them at all. Since the model did not perform as well as the YOLO model, it was ultimately not chosen for milestone 2.

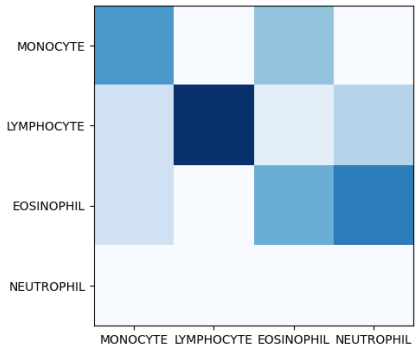

Figure 11: Confusion Matrix Using ResNet 50

7 FUTURE WORK

There are several things that can be done in the future in order to improve the project. The backend portion of the website could be improved using multithreading in order to make it annotate the images faster. If the precision of the model is improved, then more diseases can be identified. In addition to this, the models were only trained on cells that were stained using the Romanowsky technique. However, other common staining techniques include Wright's stain and Giemsa stain which are also commonly used in laboratory blood testing. In order to fix this, the models can be trained on images that contain blood cells that were stained using alternate methods.

8 CONCLUSION

YOLO has the potential to improve the accuracy and efficiency/turn-around time of CBCs and peripheral blood smears. The proposed approach involves using YOLO to classify the different types of cells within an image of blood cells, and then determine what disease the patient may have if abnormalities are found in the ratios between RBC, WBC, and Platelets. This can significantly reduce the time necessary to conduct a peripheral blood smear as that is subject to human error and may take several days to complete and deliver. The results from this approach have been promising. The first YOLO model yielded an F1 score of 0.85, while the second model yielded an F1 score of 0.75.

REFERENCES

[n.d.]. What is YOLOv8? The Ultimate Guide.

Yousef Al-Kofahi, Alla Zaltsman, Robert Graves, Will Marshall, and Mirabela Rusu. 2018. A deep learning-based algorithm for 2-D cell segmentation in microscopy images. *BMC Bioinformatics* 19, 1 (2018). https://doi.org/10.1186/s12859-018-2375-z

Mohammad Mahmudul Alam and Mohammad Tariqul Islam. 2019. Machine learning approach of automatic identification and counting of blood cells. *Healthcare Technology Letters* 6, 4 (Jul 2019), 103–108. https://doi.org/10.1049/htl.2018.5098

Gregor Gunčar, Matjaž Kukar, Mateja Notar, Miran Brvar, Peter Černelč, Manca Notar, and Marko Notar. 2018. An application of machine learning to haematological diagnosis. *Scientific Reports* 8, 1 (2018). https://doi.org/10.1038/s41598-017-18564-8

Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. 2022. A Review of Yolo Algorithm Developments. *Procedia Computer Science* 199 (2022), 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135 The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 2021): Developing Global Digital Economy after COVID-19.

Peter Lebiedzinski. 2021. A single number metric for evaluating object detection models. https://towardsdatascience.com/a-single-number-metric-for-evaluating-object-detection-models-c97f4a98616d#

Olga Pozdnyakova, Nathan T Connell, Elisabeth M Battinelli, Jean M Connors, Geoffrey Fell, and Annette S Kim. 2020. Clinical significance of CBC and WBC morphology in the diagnosis and clinical course of COVID-19 infection. *American Journal of Clinical Pathology* 155, 3 (2020), 364–375. https://doi.org/10.1093/ajcp/aqaa231

Devvi Sarwinda, Radifa Hilya Paradisa, Alhadi Bustamam, and Pinkie Anggia. 2021. Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer. *Procedia Computer Science* 179 (2021), 423–431. https://doi.org/10.1016/j.procs.2021.01.025 5th International Conference on Computer Science and Computational Intelligence 2020.

Lu Tan, Tianran Huangfu, Liyao Wu, and Wenying Chen. 2021. Comparison of RetinaNet, SSD, and Yolo V3 for real-time pill identification. *BMC Medical Informatics and Decision Making* 21, 1 (2021). https://doi.org/10.1186/s12911-021-01691-8

Junde Xu, Donghao Zhou, Danruo Deng, Jingpeng Li, Cheng Chen, Xiangyun Liao, Guangyong Chen, and Pheng Ann Heng. 2022. Deep learning in cell image analysis. *Intelligent Computing* 2022 (Sep 2022). https://doi.org/10.34133/2022/9861263