Leukemia Cancer Detection

Using Image Classification

Our team

Chinta Sai Ranganath

Sidramyna Rathnakar

Mohammad Fariduddin

Lunavath Thirupathi

Kota Amith

Likith Chowdary

Contents

Introduction

Leukemia poses a global health challenge, with current diagnostic methods relying on manual microscopic analysis of blood samples, which is slow and error-prone. In this project, we utilize deep learning techniques like CNN, ResNet, and VGG to enhance leukemia diagnosis accuracy and efficiency by classifying leukemic cells from normal ones in microscopic images.

This endeavor represents a significant advancement, aiming to overcome traditional diagnostic limitations and improve patient outcomes. Through our research, we aim to contribute to leukemia understanding and translate our findings into practical benefits for patients and healthcare providers, envisioning a future of faster, more reliable diagnosis.

Problem Statement

Manual leukemia diagnosis using microscopic blood sample analysis is slow, error-prone, and inefficient. To address this, we need automated machine learning algorithms capable of accurately distinguishing leukemic cells from healthy ones in microscopic images. Through the development of robust image classification models using deep learning techniques, we aim to enhance the efficiency and accuracy of leukemia diagnosis, leading to improved patient outcomes and reduced mortality rates.

Data Collection

01 The dataset was fetched from the website ‘https://universe.roboflow.com/custom-yolov5-o0hdb/leukemia-cancer-detection’

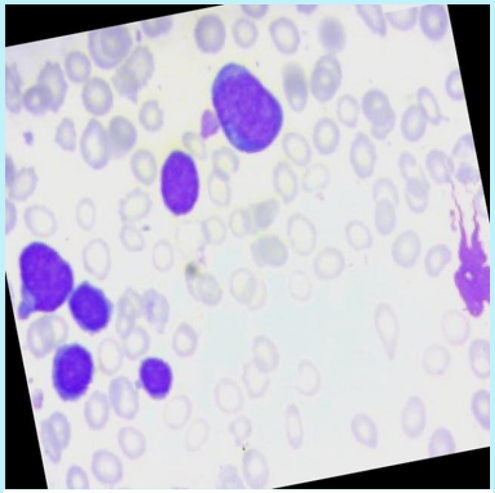
02 The dataset contains a total of 7535 images and labels.

03 Training Data: 6591 images and labels.
Testing Data: 312 images and labels.
Validation Data: 622 images and labels.

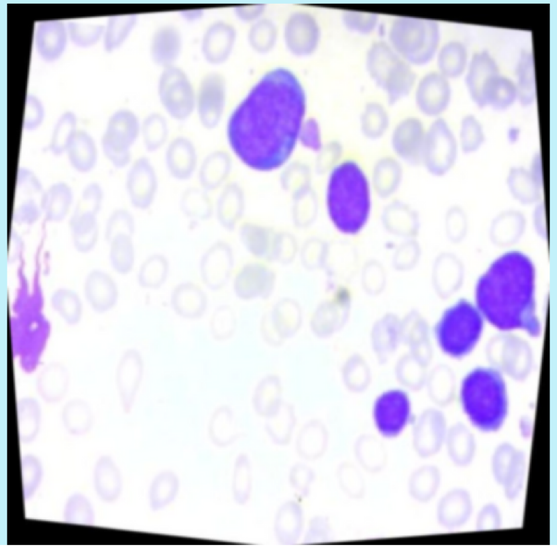
04 Images are in standard image formats (e.g., JPG,), while labels are in a format compatible with YOLOv5 object detection framework.

Data Preprocessing

1. Auto-orientation of pixel data



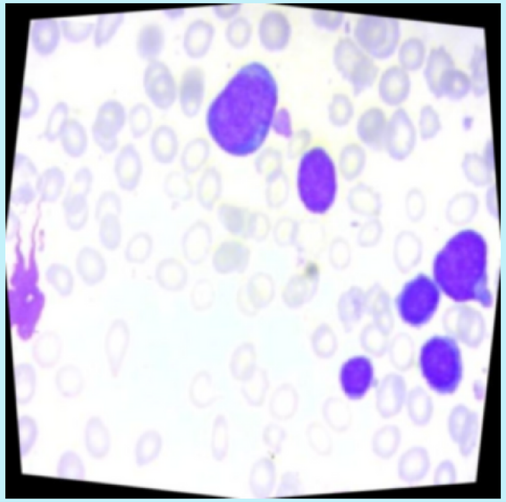
Original Image



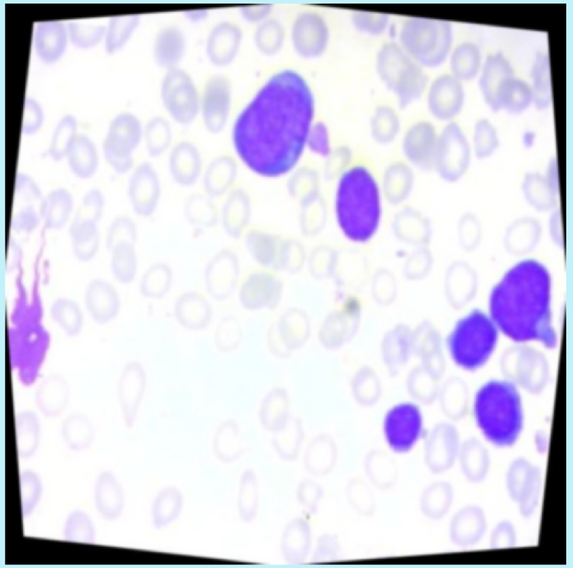
Auto-oriented Image

Data Preprocessing

2. Resizing image (640 X 640)



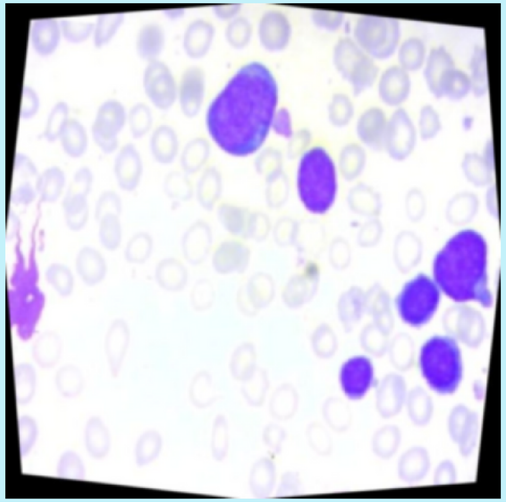
Original Image



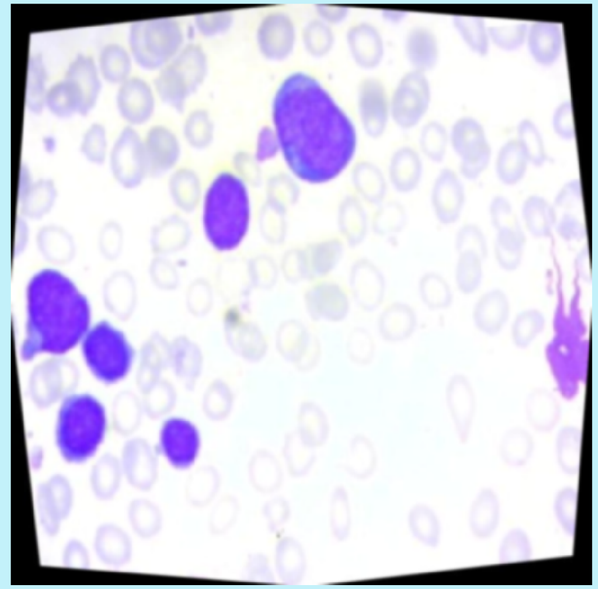
Resized Image

Data Augmentation

1. 50% probability of horizontal flip



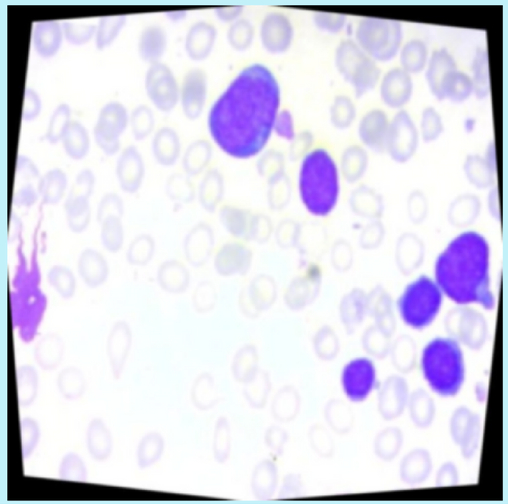
Original Image



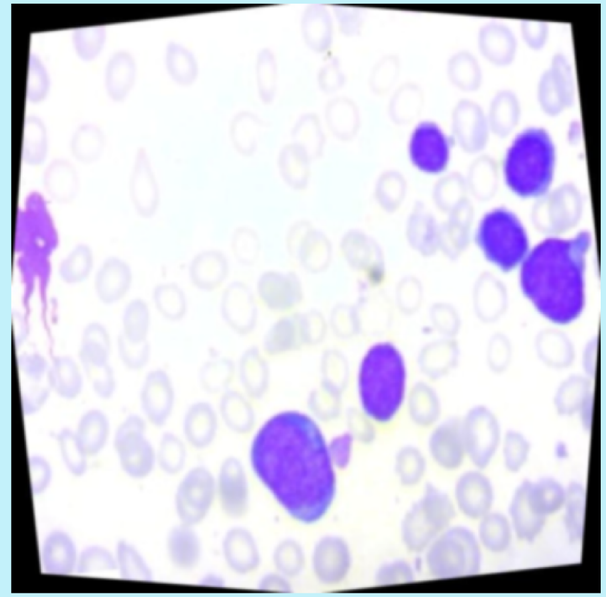
Horizontally Flipped Image

Data Augmentation

2. 50% probability of vertical flip



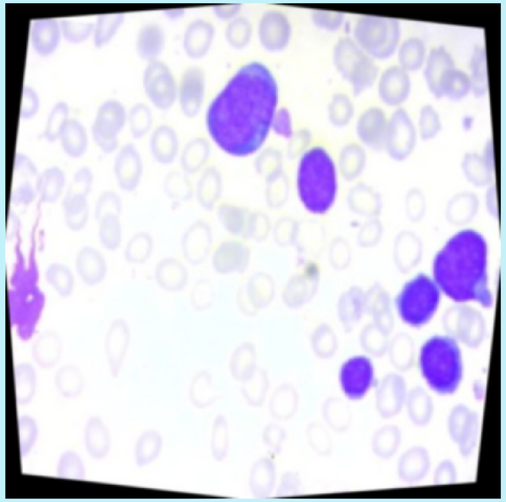
Original Image



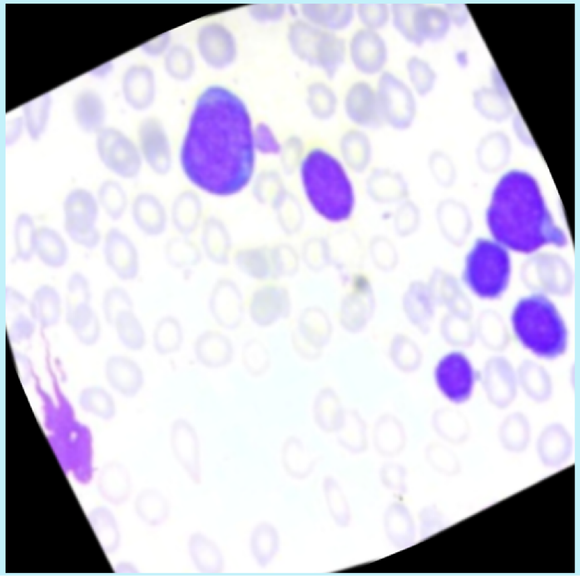
Vertically Flipped Image

Data Augmentation

3. Random rotation of between -30 and +30 degrees



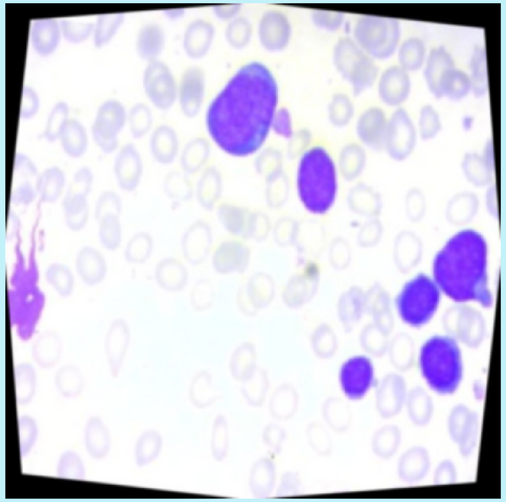
Original Image



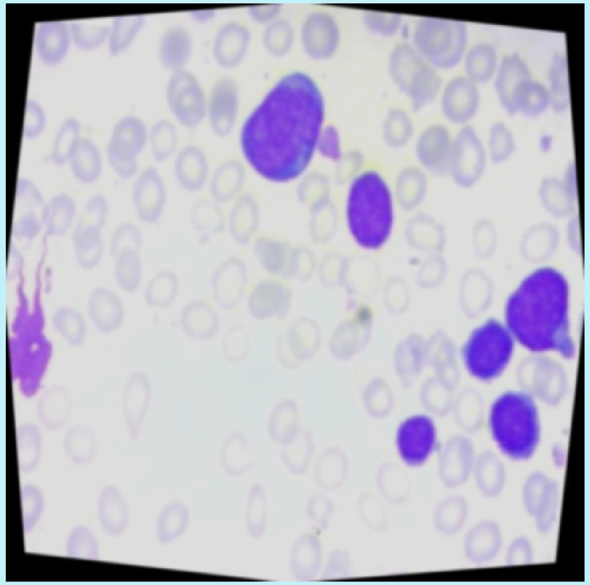
Rotated image

Data Augmentation

4. Random brightness adjustment of between -15 and +15 percent



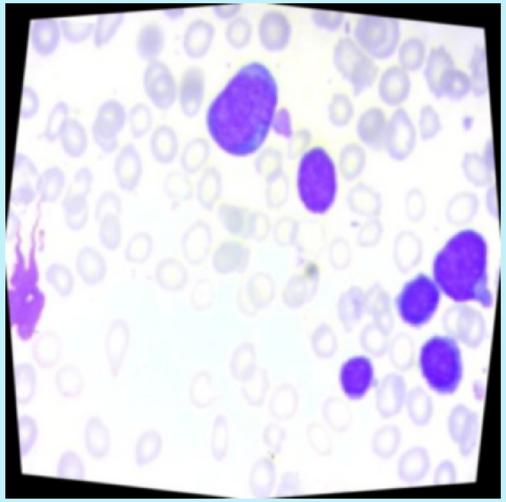
Original Image



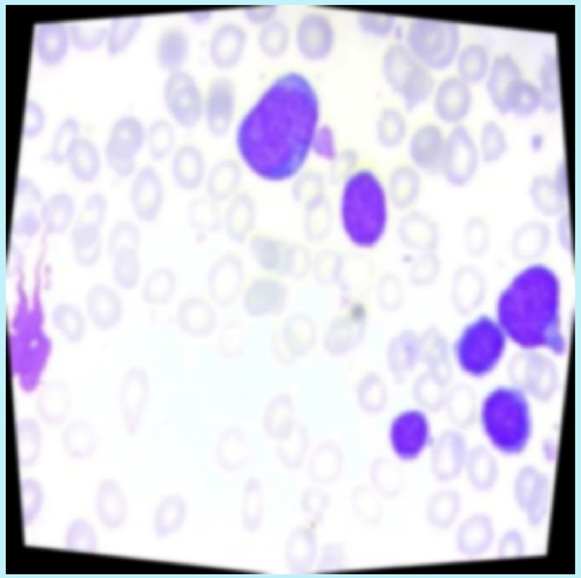
Brightness Adjusted Image

Data Augmentation

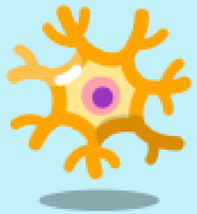
5. Random Gaussian blur of between 0 and 2 pixels



Original Image



Blurred Image



CNN Model Development

Convolutional Layers: These layers extract features from input images by applying filters, generating feature maps capturing patterns and edges. Multiple layers with increasing filters enable the model to learn increasingly complex features.

Activation Functions: These functions introduce non-linearity into the model, allowing it to learn more complex relationships between features. The code uses the ReLU (Rectified Linear Unit) activation function.

Pooling Layers: These layers downsample the data, reducing its dimensionality. This can help control overfitting and make the model more efficient. Here, max pooling is used, which takes the maximum value from a specific window size (e.g., 2x2) to create a smaller output.

Flatten Layer: After the convolutional layers, the data is flattened into a 1D vector before feeding it into dense layers.

Dense Layers: These fully connected layers learn more intricate relationships between the extracted features. The code uses two dense layers with dropout (a regularization technique) to prevent overfitting. generalize to unseen data.

Training & Evaluation

Training

```
accuracy: 0.8486 - loss: 0.1733
```

```
val_accuracy: 0.8788 - val_loss: 0.1224
```

At last, we gained accuracy of approx 84% and with a loss of 0.16%.

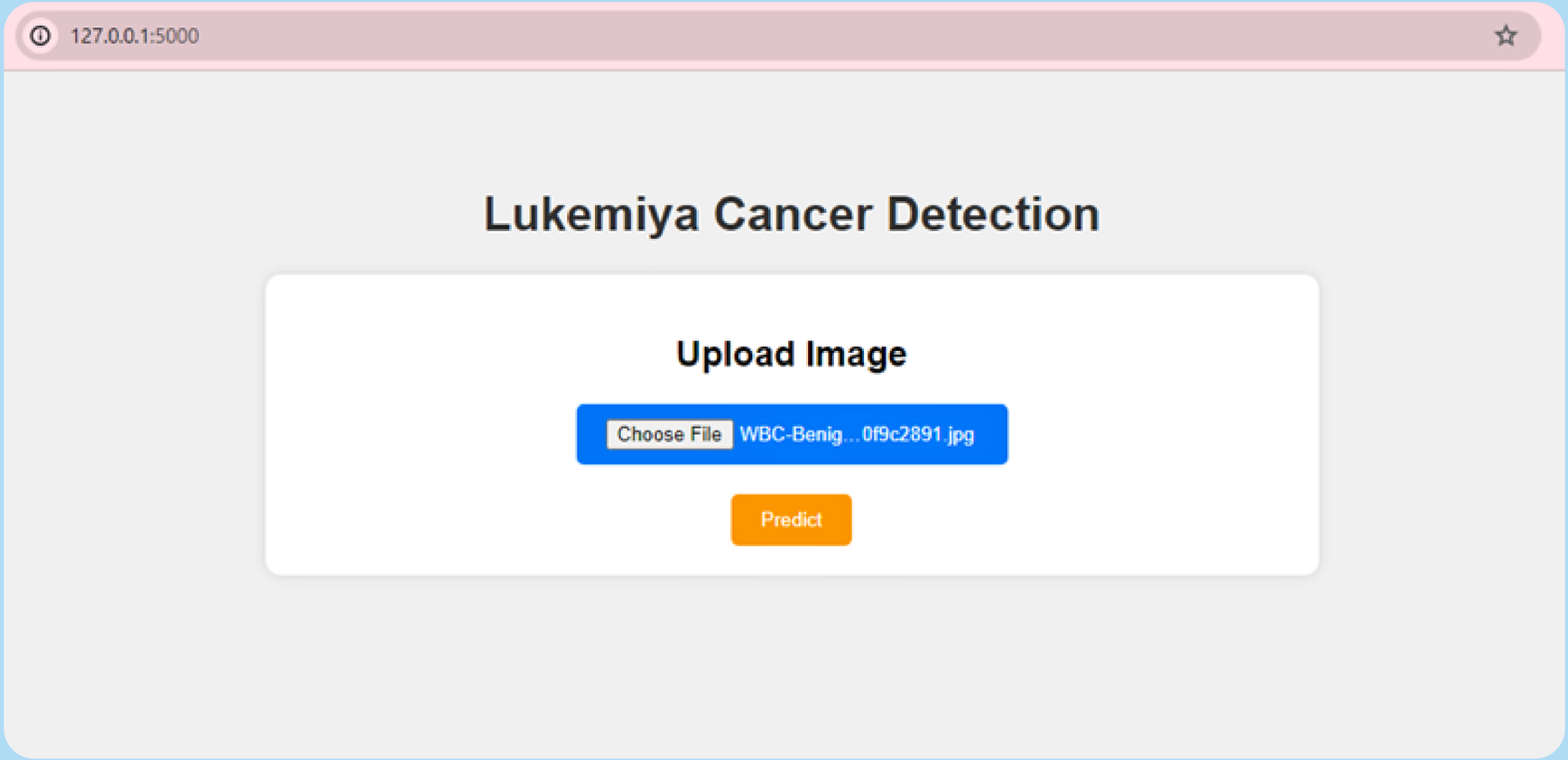
And val_accuracy is approx 87% with a loss of 13%.

Evaluation

```
Test Loss: 0.1172095462679863
Test Accuracy: 0.8814102411270142
```

After Evaluating the model, we gained Test accuracy is appox 88% with a loss of approx 12%.

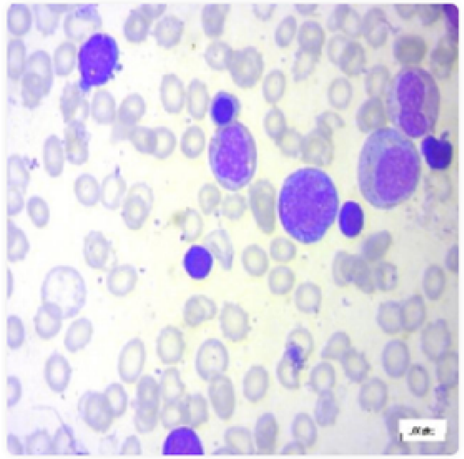
WEB APPLICATION



Result

127.0.0.1:5000/predict

Lukemiya Cancer Detection Results

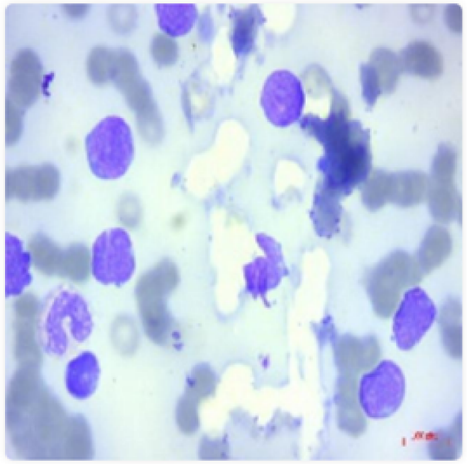


Result : Benign

Result

127.0.0.1:5000/predict

Lukemiya Cancer Detection Results

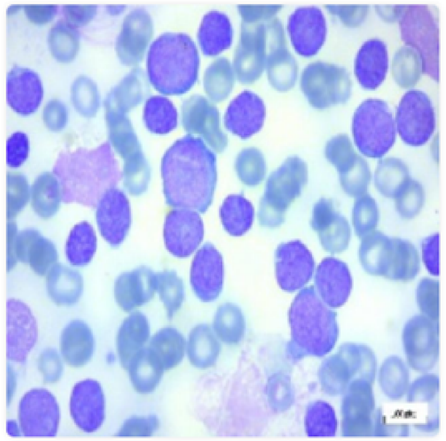


Result : Malignant Early

Result

127.0.0.1:5000/predict

Lukemiya Cancer Detection Results

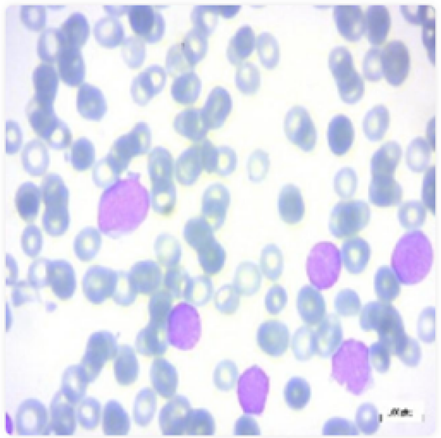


Result : Malignant Pre

Result

127.0.0.1:5000/predict

Lukemiya Cancer Detection Results

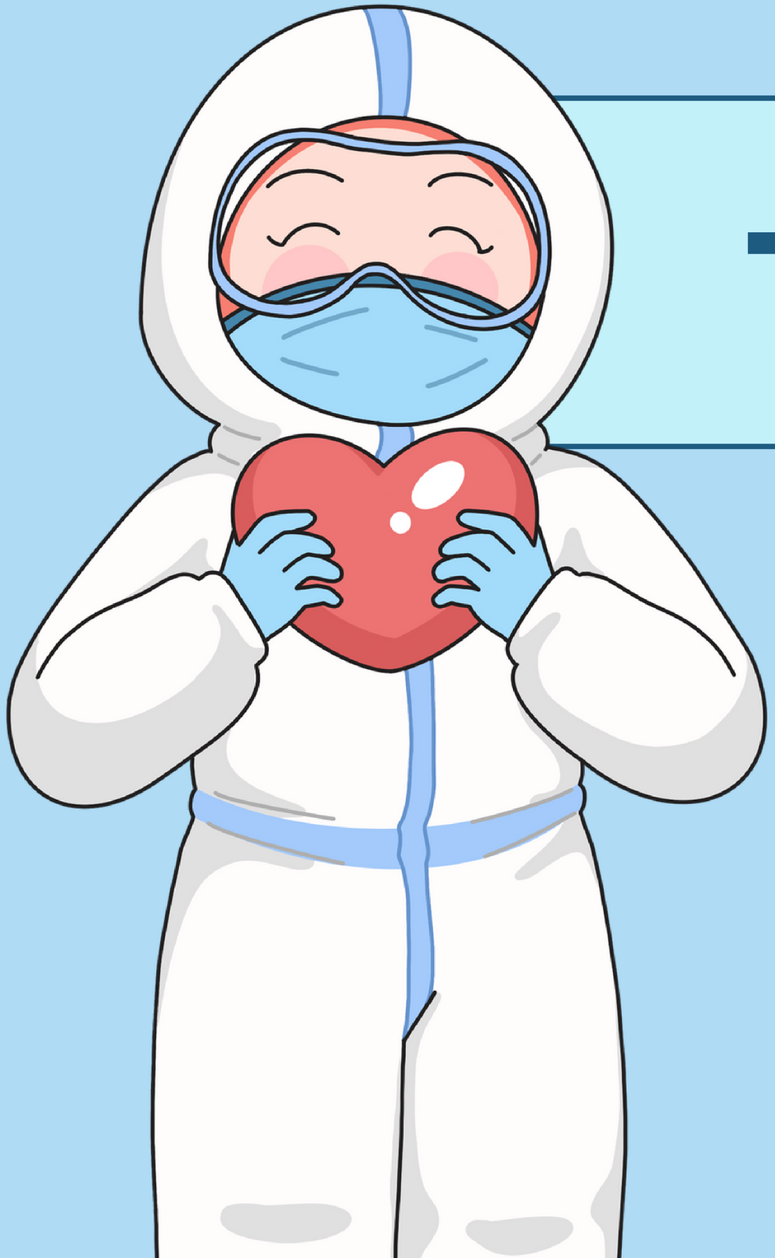


Result : Malignant Pro

Conclusion

In conclusion, our journey to leverage computer technology for improving leukemia diagnosis has been promising. Through rigorous testing of deep learning techniques, we aimed to enhance the speed and accuracy of detecting leukemia cells in microscopic images. Our primary objective was to enable early detection of leukemia, facilitating better treatment outcomes for patients.

By combining research efforts and collaborative initiatives, we are optimistic about the potential impact of our work in revolutionizing leukemia diagnosis. With continued dedication and innovation, we aspire to make significant strides towards improving healthcare outcomes for individuals affected by leukemia.



THANK YOU!