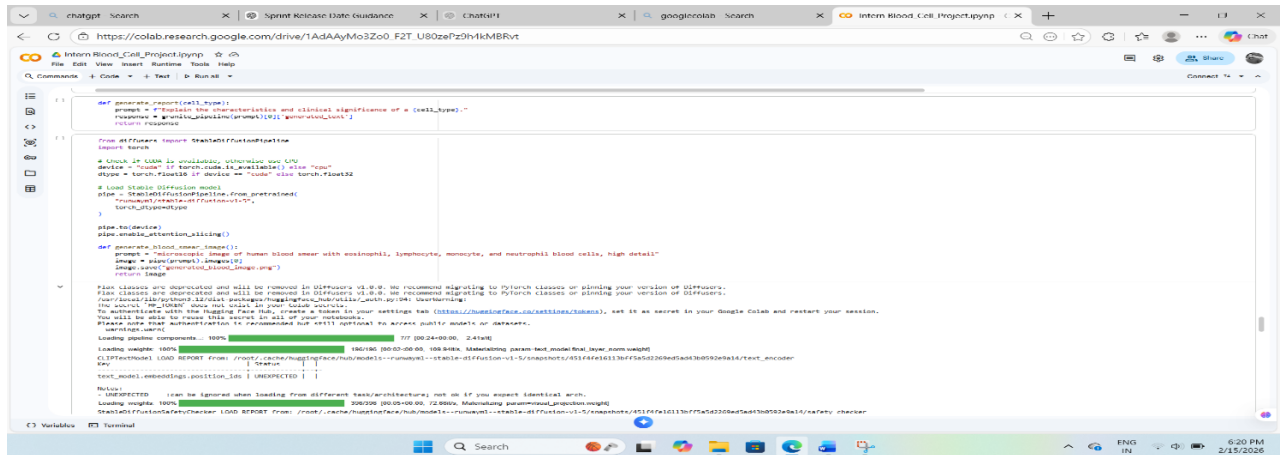


## Project Demonstration Phase

### Stable Diffusion Model Initialization and Device

This section of the code initializes the Stable Diffusion v1.5 model using the Hugging Face Diffusers library. It first checks whether CUDA (GPU) is available and selects either GPU or CPU for execution to optimize performance. The appropriate data type (float16 for GPU or float32 for CPU) is selected to reduce memory usage. The pipeline is then loaded and moved to the selected device, enabling efficient image generation.



```
def generate_response(cell_type):
    prompt = f"Provide the characteristics and clinical significance of a {cell_type}."
    response = generate_blood_smear_image(cell_type, generate_prompt)
    return response

def generate_blood_smear_image():
    # Check if CUDA is available, otherwise use CPU
    device = "cuda" if torch.cuda.is_available() else "cpu"
    dtype = torch.float16 if device == "cuda" else torch.float32

    # Load Stable Diffusion model
    pipe = StableDiffusionPipeline.from_pretrained(
        "stabilityai/stable-diffusion-v1-5",
        torch_dtype=dtype
    )
    pipe.to(device)
    pipe.enable_attention_slicing()

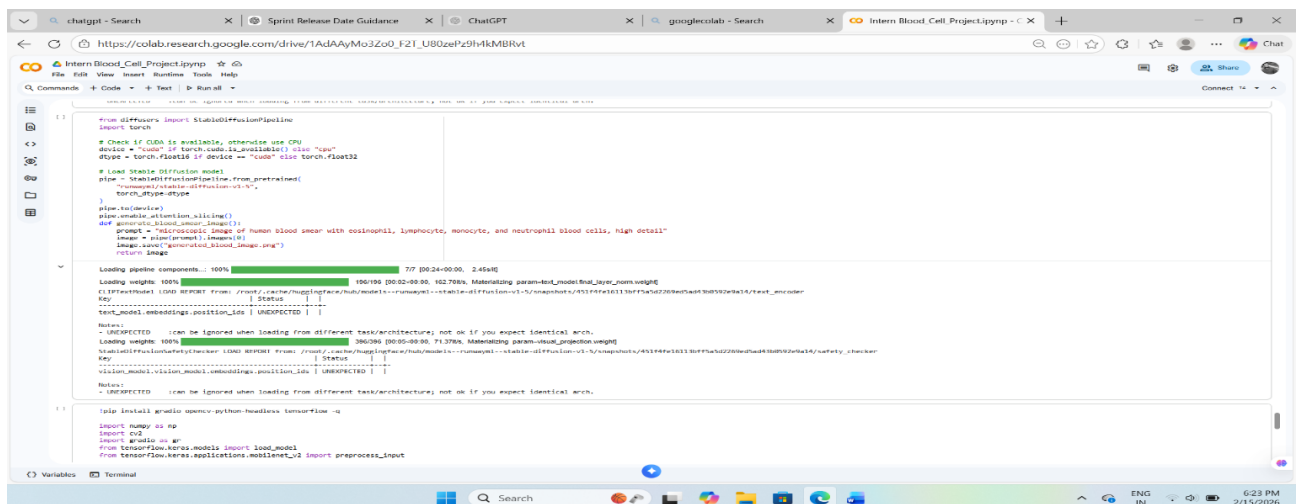
    # Generate blood smear image
    prompt = "Microscopic image of human blood smear with eosinophils, lymphocyte, monocyte, and neutrophil blood cells, high detail"
    image = pipe(prompt).images[0]
    image.save("generated_blood_image.png")
    return image

# Main function to generate blood smear image
def generate_blood_smear_image():
    prompt = "Microscopic image of human blood smear with eosinophils, lymphocyte, monocyte, and neutrophil blood cells, high detail"
    image = pipe(prompt).images[0]
    image.save("generated_blood_image.png")
    return image

# Main function to generate blood smear image
def generate_blood_smear_image():
    prompt = "Microscopic image of human blood smear with eosinophils, lymphocyte, monocyte, and neutrophil blood cells, high detail"
    image = pipe(prompt).images[0]
    image.save("generated_blood_image.png")
    return image
```

### Blood Smear Image Generation Using Text-to-Image Prompt

This part of the code defines the function generate\_blood\_smear\_image() to create synthetic microscopic blood smear images. A detailed medical prompt describing eosinophils, lymphocytes, monocytes, and neutrophils is provided to the model. The Stable Diffusion pipeline generates a high-resolution image based on this prompt and saves it as generated\_blood\_image.png. The console output below confirms successful loading of model weights and components.



```
def generate_blood_smear_image():
    # Check if CUDA is available, otherwise use CPU
    device = "cuda" if torch.cuda.is_available() else "cpu"
    dtype = torch.float16 if device == "cuda" else torch.float32

    # Load Stable Diffusion model
    pipe = StableDiffusionPipeline.from_pretrained(
        "stabilityai/stable-diffusion-v1-5",
        torch_dtype=dtype
    )
    pipe.to(device)
    pipe.enable_attention_slicing()

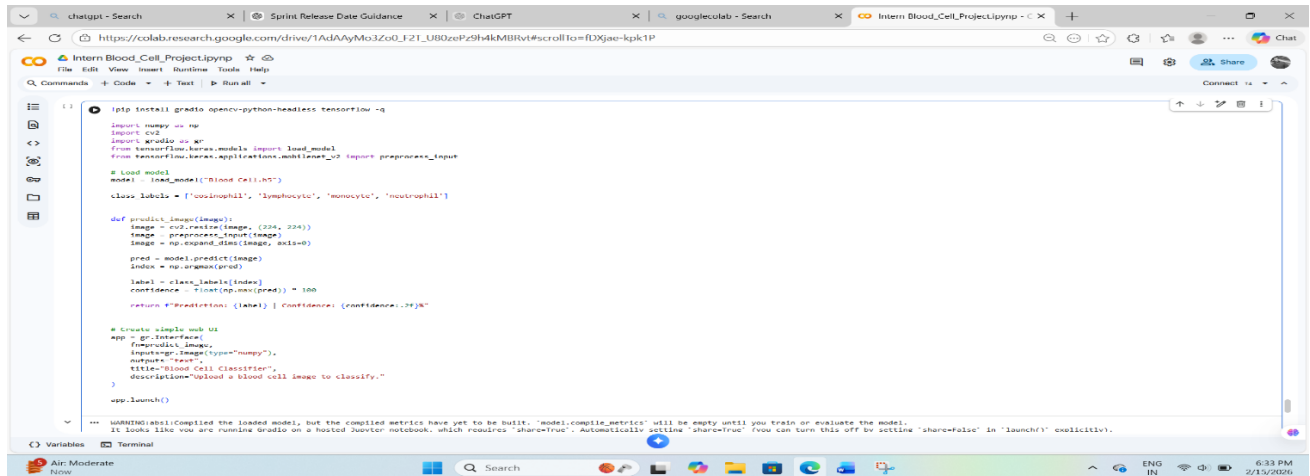
    # Generate blood smear image
    prompt = "Microscopic image of human blood smear with eosinophils, lymphocyte, monocyte, and neutrophil blood cells, high detail"
    image = pipe(prompt).images[0]
    image.save("generated_blood_image.png")
    return image

# Main function to generate blood smear image
def generate_blood_smear_image():
    prompt = "Microscopic image of human blood smear with eosinophils, lymphocyte, monocyte, and neutrophil blood cells, high detail"
    image = pipe(prompt).images[0]
    image.save("generated_blood_image.png")
    return image

# Main function to generate blood smear image
def generate_blood_smear_image():
    prompt = "Microscopic image of human blood smear with eosinophils, lymphocyte, monocyte, and neutrophil blood cells, high detail"
    image = pipe(prompt).images[0]
    image.save("generated_blood_image.png")
    return image
```

## Gradio-Based Deployment of Blood Cell Classification Model

This section of the code deploys the trained Blood Cell classification model using the Gradio library. The saved model Blood\_Cell.h5 is loaded, and class labels such as eosinophil, lymphocyte, monocyte, and neutrophil are defined. The predict\_image() function preprocesses the uploaded image by resizing it to 224×224 pixels, applying MobileNetV2 preprocessing, and expanding dimensions before making predictions. The predicted class is determined using argmax, and the confidence score is calculated and displayed as output.



```
!pip install gradio opencv-python-headless tensorflow -q

import numpy as np
import cv2
import gradio as gr
from tensorflow.keras.models import load_model
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input

# Load model
model = load_model('Blood_Cell.h5')
class_labels = ['eosinophil', 'lymphocyte', 'monocyte', 'neutrophil']

def predict_image(image):
    image = cv2.resize(image, (224, 224))
    image = preprocess_input(image)
    image = np.expand_dims(image, axis=0)
    pred = model.predict(image)
    index = np.argmax(pred)
    label = class_labels[index]
    confidence = float(np.max(pred)) * 100
    return f"Prediction: {label} | Confidence: {confidence:.2f}%"

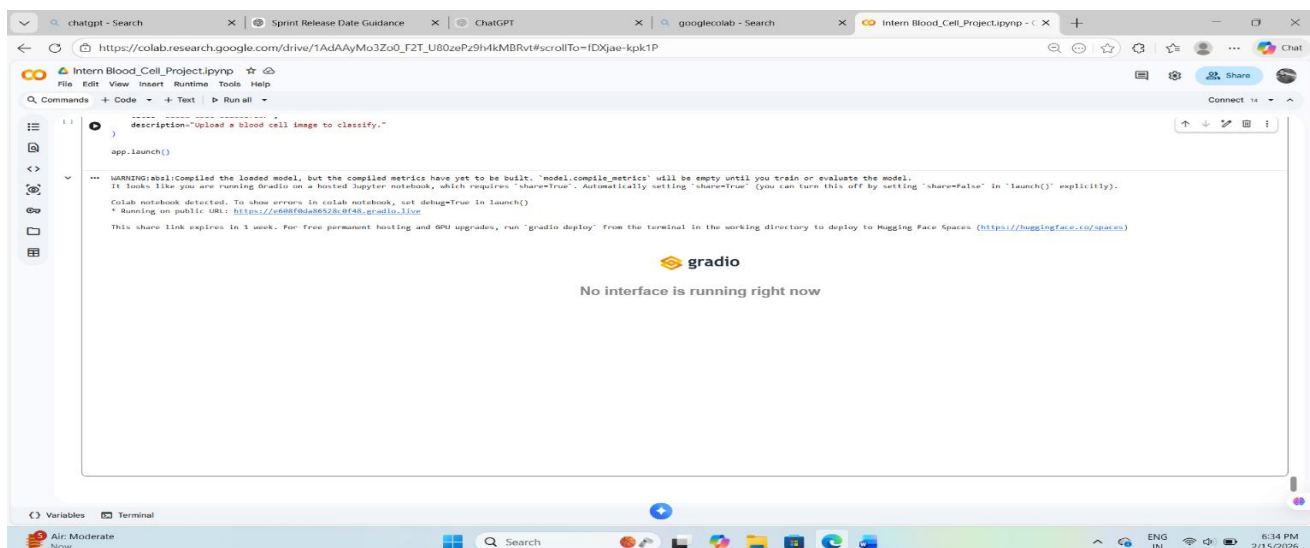
# Create simple web UI
app = gr.Interface(
    fn=predict_image,
    inputs="Image (type='numpy')",
    outputs="Text",
    title="Blood Cell Classifier",
    description="Upload a blood cell image to classify."
)

app.launch()
```

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. 'model.compile\_metrics' will be empty until you train or evaluate the model.  
It looks like you are running Gradio on a hosted Jupyter notebook, which requires 'share=True'. Automatically setting 'share=True'. You can turn this off by setting 'share=False' in 'launch()' explicitly.


## Launching the Web Interface for Model Deployment

This part shows the execution of app.launch(), which starts the Gradio web interface for user interaction. Once launched, the application generates a public shareable link, allowing users to upload blood cell images and receive classification results in real time. The console output confirms successful deployment and provides the temporary URL for accessing the application online.

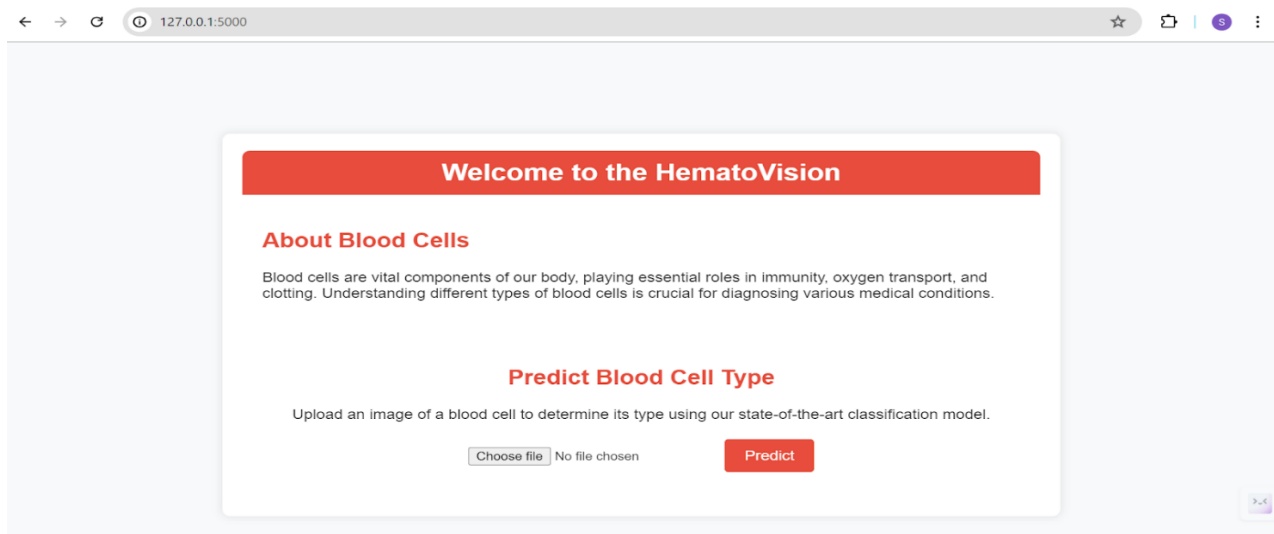


```
app.launch()
```

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. 'model.compile\_metrics' will be empty until you train or evaluate the model.  
It looks like you are running Gradio on a hosted Jupyter notebook, which requires 'share=True'. Automatically setting 'share=True'. You can turn this off by setting 'share=False' in 'launch()' explicitly).  
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()  
\* Running on public URL: <https://c000f6a06923c0f46-gradio-110e>  
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run "gradio deploy" from the terminal in the working directory to deploy to Hugging Face Spaces (<https://huggingface.co/spaces>)

 **gradio**

No interface is running right now



By clicking on choose file it will ask us to upload the image , then by clicking on the predict button it will take us to the result.html.

Activity 1: Test For Class-1 – Neutrophil

## Predict Blood Cell Type

Upload an image of a blood cell to determine its type using our state-of-the-art classification model.



## Activity 2: Test For Class-2 – Monocyte

### Predict Blood Cell Type

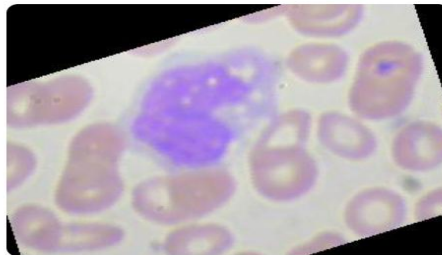
Upload an image of a blood cell to determine its type using our state-of-the-art classification model.

Choose file \_3\_9423.jpeg

Predict

#### Prediction Result

**Predicted Class:** monocyte



Upload Another Image

## Activity 3: Test For Class-3 – Lymphocyte

### Predict Blood Cell Type

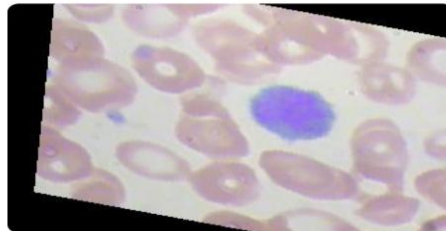
Upload an image of a blood cell to determine its type using our state-of-the-art classification model.

Choose file \_5\_9201.jpeg

Predict

#### Prediction Result

**Predicted Class:** lymphocyte



Upload Another Image

#### Activity 4: Test For Class-4 – Eosinophil

### Predict Blood Cell Type

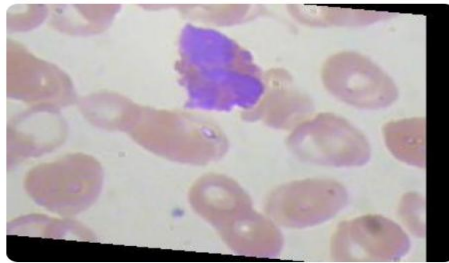
Upload an image of a blood cell to determine its type using our state-of-the-art classification model.

Choose file \_3\_9885.jpeg

Predict

#### Prediction Result

**Predicted Class:** eosinophil



Upload Another Image

Demo Link:

[https://drive.google.com/file/d/1oXSLu4IW-SQOzftMFpitagMg\\_7hxfgKW/view?usp=drivesdk](https://drive.google.com/file/d/1oXSLu4IW-SQOzftMFpitagMg_7hxfgKW/view?usp=drivesdk)

Google Colab Link :

<https://colab.research.google.com/drive/1nnHWn4UKBKx6oxfna4zrmP5ovKtj4S5b>