Team ID : LTVIP2026TMIDS87105

Team Size : 4

Team Leader : Kagitala Saikumar

Team member : Doddaka Venkata Chandu

Team member : Ravi Lakshmi Sai Varun

Team member : Kambala Venkata Sai Prem Chand

## Prior Knowledge

You must have prior knowledge of the following topics to complete this project.

- DL Concepts
  - Neural Networks:: https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/
  - Deep Learning Frameworks:: https://www.knowledgehut.com/blog/data-science/pytorch-vs-tensorflow
  - Transfer Learning: https://towardsdatascience.com/a-demonstration-of-transfer-learning-of-vgg-convolutional-neural-network-pre-trained-model-with-c9f5b8b1ab0a
  - VGG16: https://www.geeksforgeeks.org/vgg-16-cnn-model/
  - Convolutional Neural Networks (CNNs): https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ s://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
  - Overfitting and Regularization: https://www.analyticsvidhya.com/blog/2021/07/prevent-overfitting-using-regularization-techniques/
  - Optimizers: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/
- Flask Basics: https://www.youtube.com/watch?v=lj4I_CvBnt0

## Project Objectives

By the end of this project, you will:

- Know fundamental concepts and techniques used for Deep Learning.
- Gain a broad understanding of data.
- Have knowledge of pre-processing the data/transformation techniques on outliers and some visualization concepts.

## Project Flow

- The user interacts with the UI (User Interface) to choose the image.
- The chosen image is analyzed by the model which is integrated with the flask application.
- Once the model analyses the input the prediction is showcased on the UI

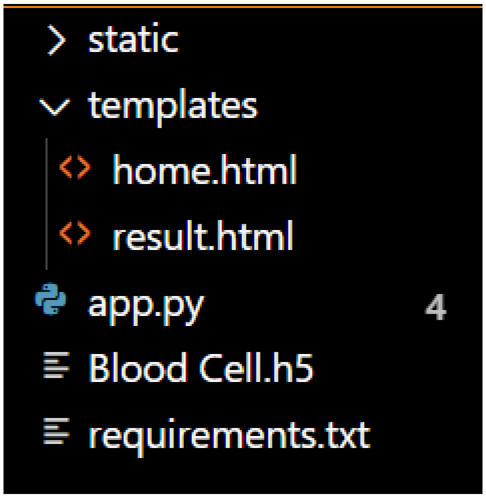
To accomplish this, we have to complete all the activities listed below,

- Data Collection: Collect or download the dataset that you want to train.
- Data pre-processing
  - Data Augmentation
  - Splitting data into train and test

- Model building
    - Import the model-building libraries
    - Initializing the model
    - Training and testing the model
    - Evaluating the performance of the model
    - Save the model
- Application Building
    - Create an HTML file
    - Build python code

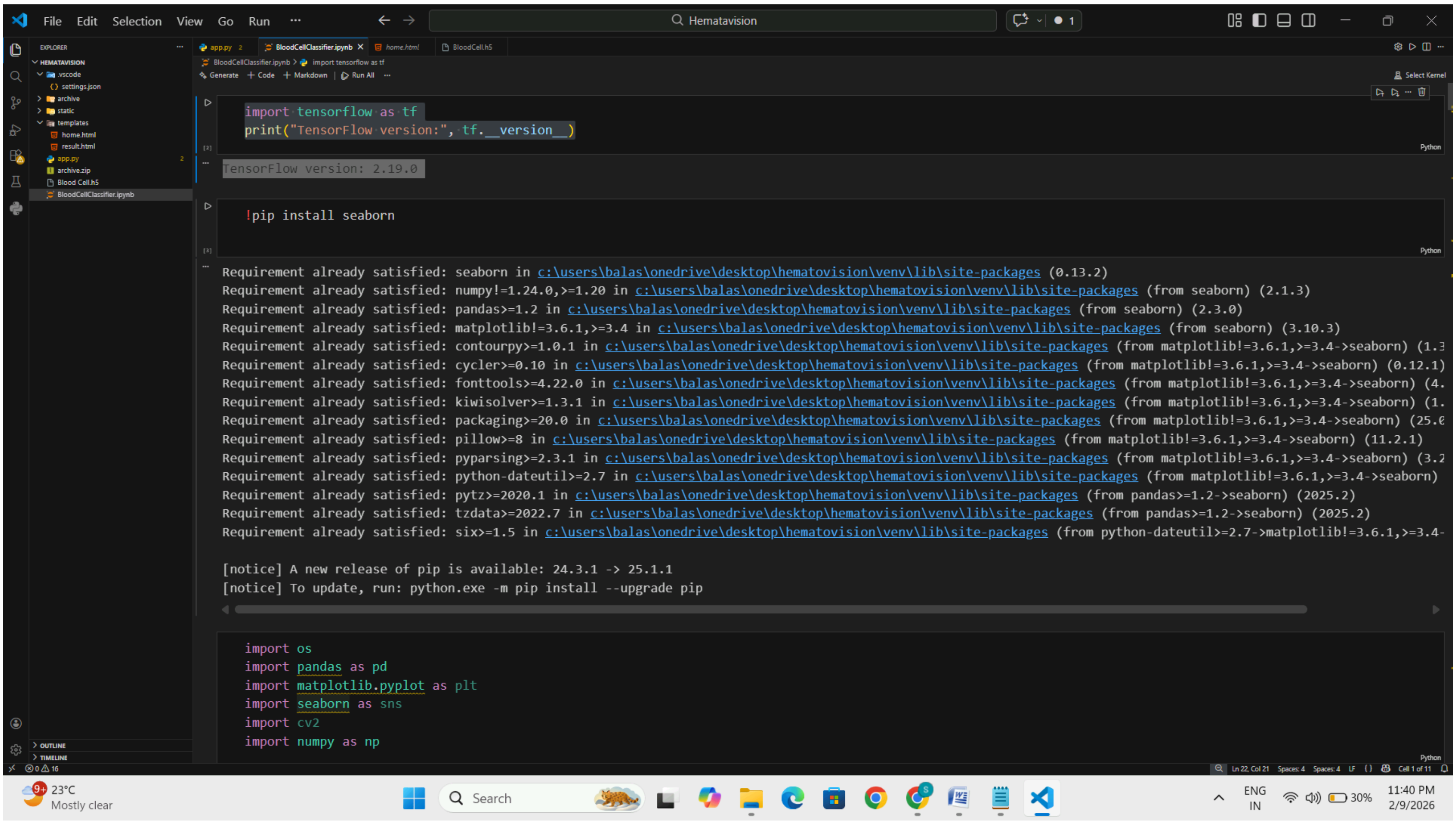
## Project Structure

Create the Project folder which contains files as shown below



```
> static
v templates
    home.html
    result.html
  app.py                4
  Blood Cell.h5
  requirements.txt
```

- We are building a Flask application with HTML pages stored in the templates folder and a Python script app.py for scripting.
- Blood Cell.h5 is our saved model. Further, we will use this model for flask integration.



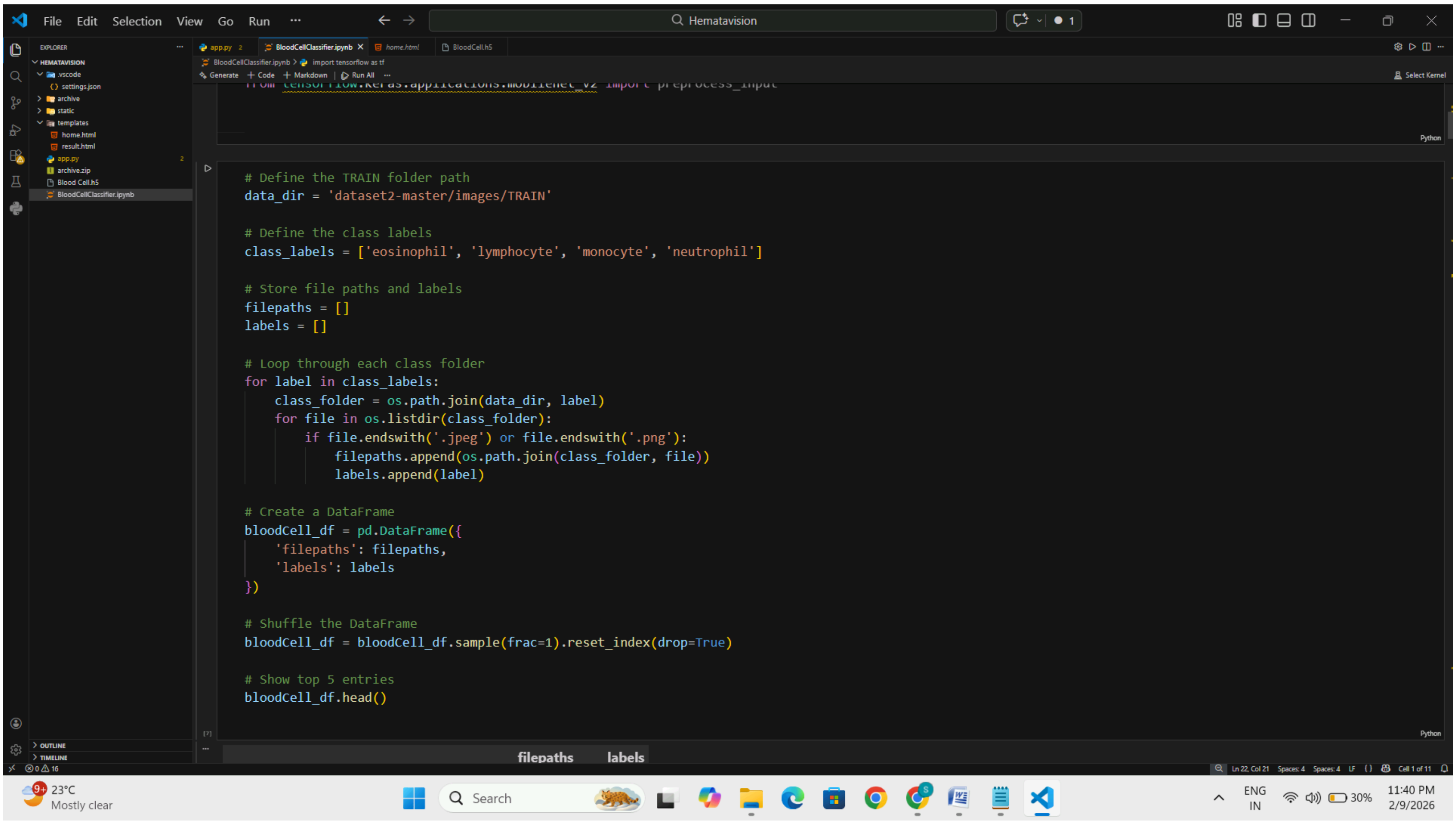
```
import tensorflow as tf
print("TensorFlow version:", tf.__version__)
```

```
TensorFlow version: 2.19.0
```

```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from seaborn) (2.1.3)
Requirement already satisfied: pandas>=1.2 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from seaborn) (2.3.0)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from seaborn) (3.10.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3
Requirement already satisfied: cycler>=0.10 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.
Requirement already satisfied: packaging>=20.0 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (25.0
Requirement already satisfied: pillow>=8 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2
Requirement already satisfied: python-dateutil>=2.7 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
Requirement already satisfied: pytz>=2020.1 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\balas\onedrive\desktop\hematovision\venv\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4-

[notice] A new release of pip is available: 24.3.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

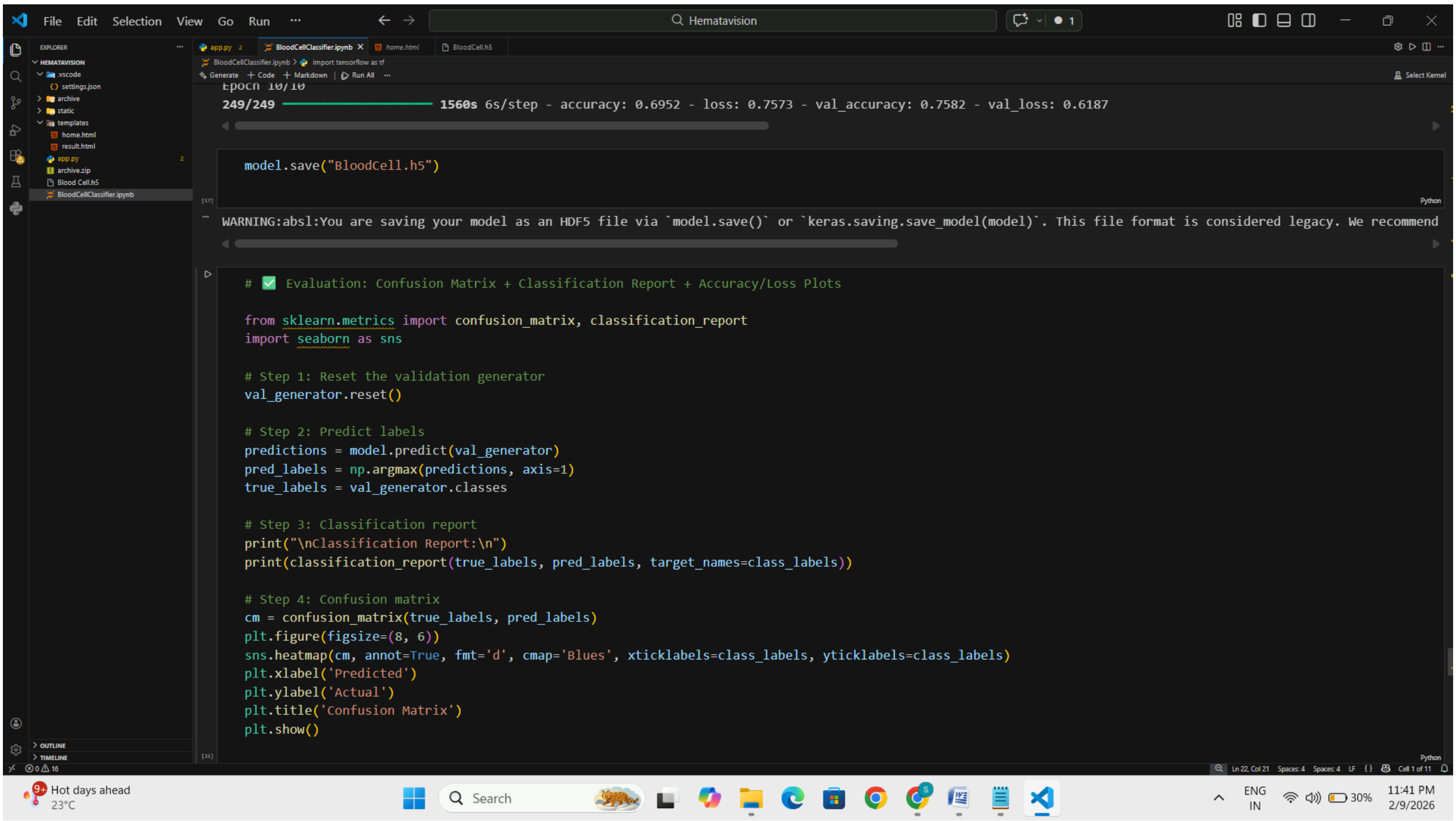
```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
import numpy as np
```



```python
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
```

```python
# Define the TRAIN folder path
data_dir = 'dataset2-master/images/TRAIN'

# Define the class labels
class_labels = ['eosinophil', 'lymphocyte', 'monocyte', 'neutrophil']

# Store file paths and labels
filepaths = []
labels = []

# Loop through each class folder
for label in class_labels:
    class_folder = os.path.join(data_dir, label)
    for file in os.listdir(class_folder):
        if file.endswith('.jpeg') or file.endswith('.png'):
            filepaths.append(os.path.join(class_folder, file))
            labels.append(label)

# Create a DataFrame
bloodCell_df = pd.DataFrame({
    'filepaths': filepaths,
    'labels': labels
})

# Shuffle the DataFrame
bloodCell_df = bloodCell_df.sample(frac=1).reset_index(drop=True)

# Show top 5 entries
bloodCell_df.head()
```

```
filepaths     labels
```

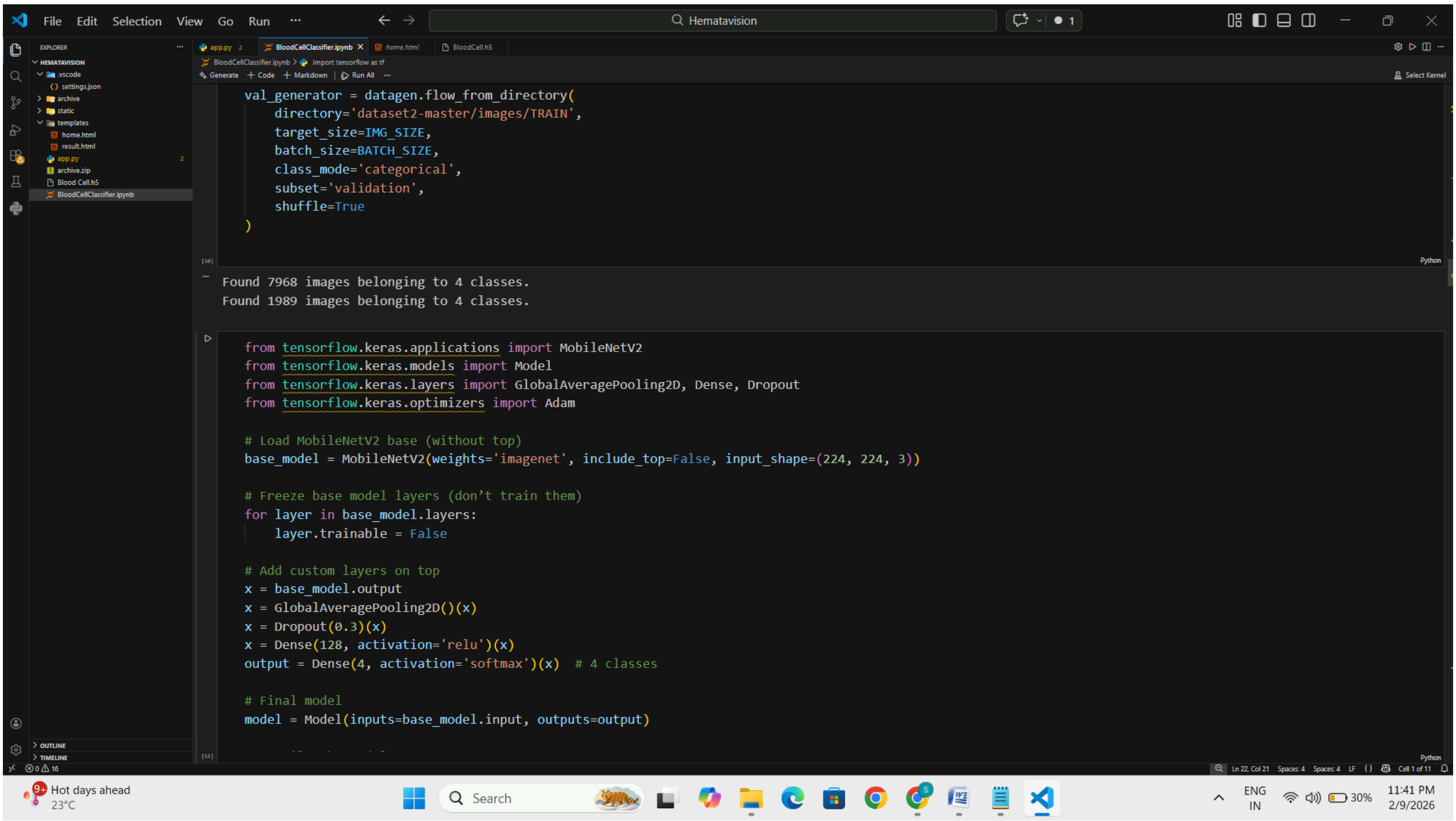


```
Epoch 10/10
249/249 ━━━━━━━━━━━━━━━━━━━━ 1560s 6s/step - accuracy: 0.6952 - loss: 0.7573 - val_accuracy: 0.7582 - val_loss: 0.6187
```

```python
model.save("BloodCell.h5")
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend
```

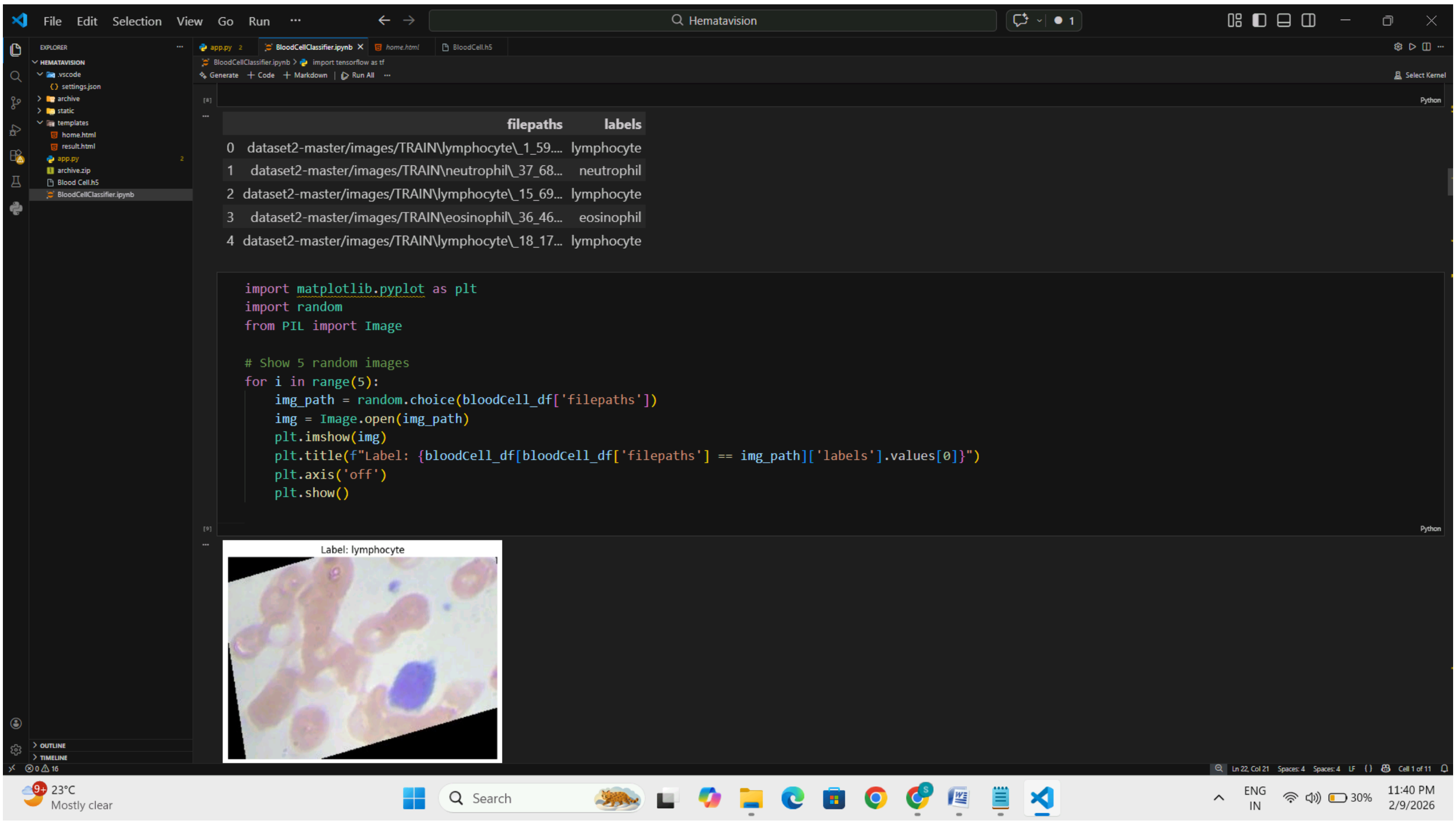
```python
# ✅ Evaluation: Confusion Matrix + Classification Report + Accuracy/Loss Plots

from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns

# Step 1: Reset the validation generator
val_generator.reset()

# Step 2: Predict labels
predictions = model.predict(val_generator)
pred_labels = np.argmax(predictions, axis=1)
true_labels = val_generator.classes

# Step 3: Classification report
print("\nClassification Report:\n")
print(classification_report(true_labels, pred_labels, target_names=class_labels))

# Step 4: Confusion matrix
cm = confusion_matrix(true_labels, pred_labels)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=class_labels, yticklabels=class_labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



```
    val_generator = datagen.flow_from_directory(
        directory='dataset2-master/images/TRAIN',
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode='categorical',
        subset='validation',
        shuffle=True
    )
```

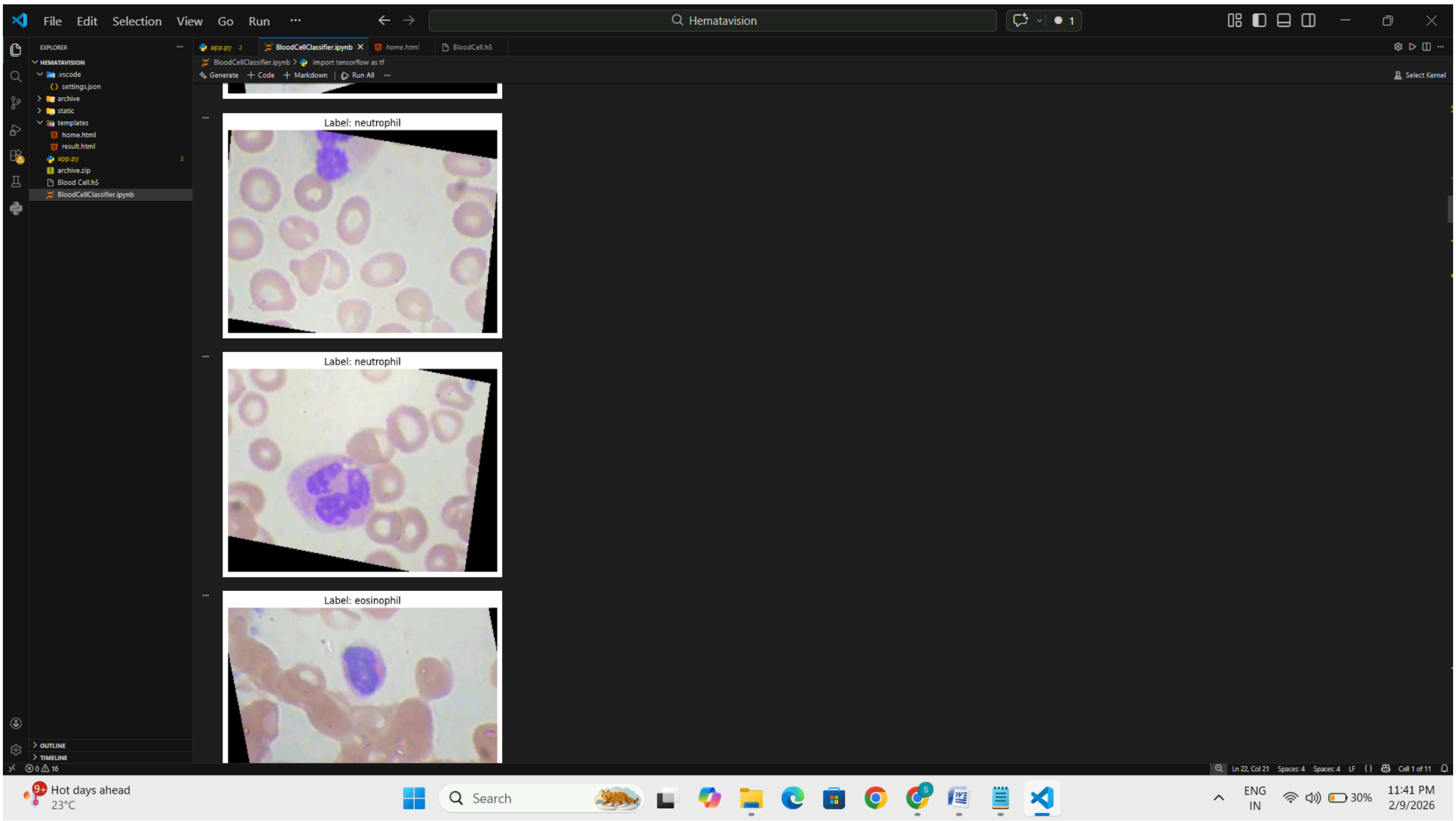
```
Found 7968 images belonging to 4 classes.
Found 1989 images belonging to 4 classes.
```

```
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.models import Model
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Dropout
from tensorflow.keras.optimizers import Adam

# Load MobileNetV2 base (without top)
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Freeze base model layers (don't train them)
for layer in base_model.layers:
    layer.trainable = False

# Add custom layers on top
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.3)(x)
x = Dense(128, activation='relu')(x)
output = Dense(4, activation='softmax')(x)  # 4 classes

# Final model
model = Model(inputs=base_model.input, outputs=output)
```



|   | filepaths | labels |
|---|---|---|
| 0 | dataset2-master/images/TRAIN\lymphocyte_1_59.... | lymphocyte |
| 1 | dataset2-master/images/TRAIN\neutrophil_37_68... | neutrophil |
| 2 | dataset2-master/images/TRAIN\lymphocyte_15_69... | lymphocyte |
| 3 | dataset2-master/images/TRAIN\eosinophil_36_46... | eosinophil |
| 4 | dataset2-master/images/TRAIN\lymphocyte_18_17... | lymphocyte |

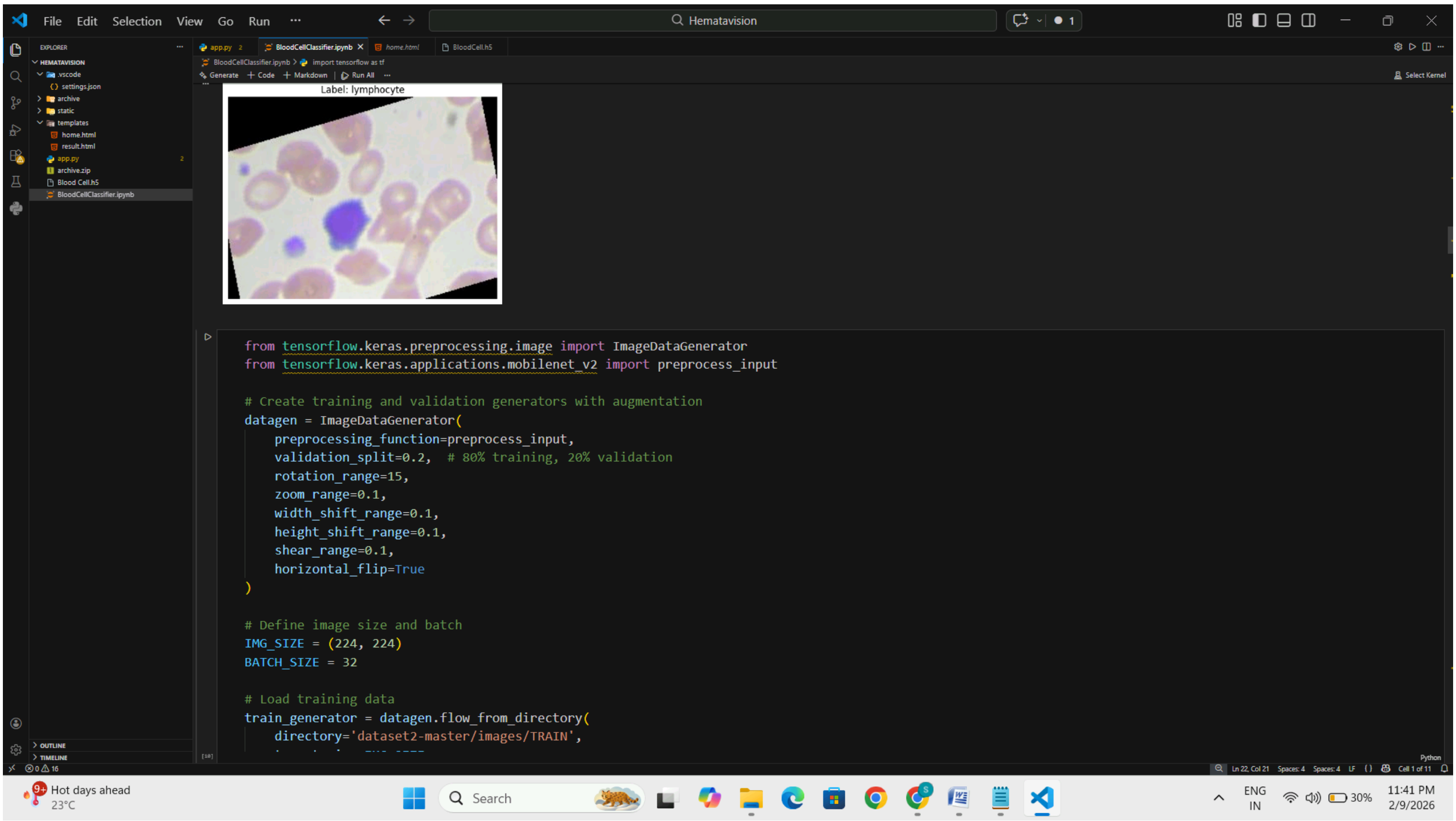
```
import matplotlib.pyplot as plt
import random
from PIL import Image

# Show 5 random images
for i in range(5):
    img_path = random.choice(bloodCell_df['filepaths'])
    img = Image.open(img_path)
    plt.imshow(img)
    plt.title(f"Label: {bloodCell_df[bloodCell_df['filepaths'] == img_path]['labels'].values[0]}")
    plt.axis('off')
    plt.show()
```



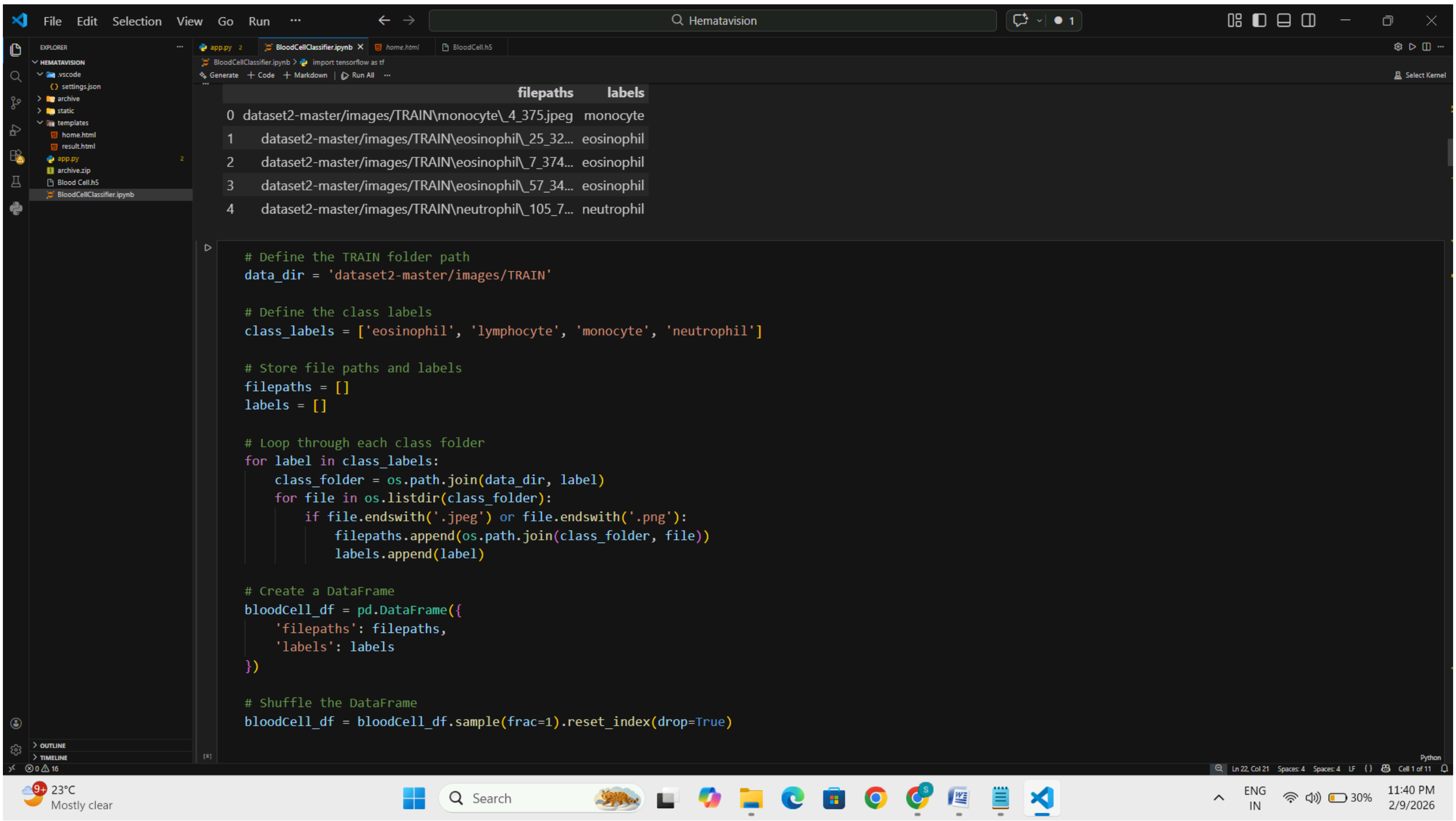
Label: neutrophil

Label: neutrophil

Label: eosinophil

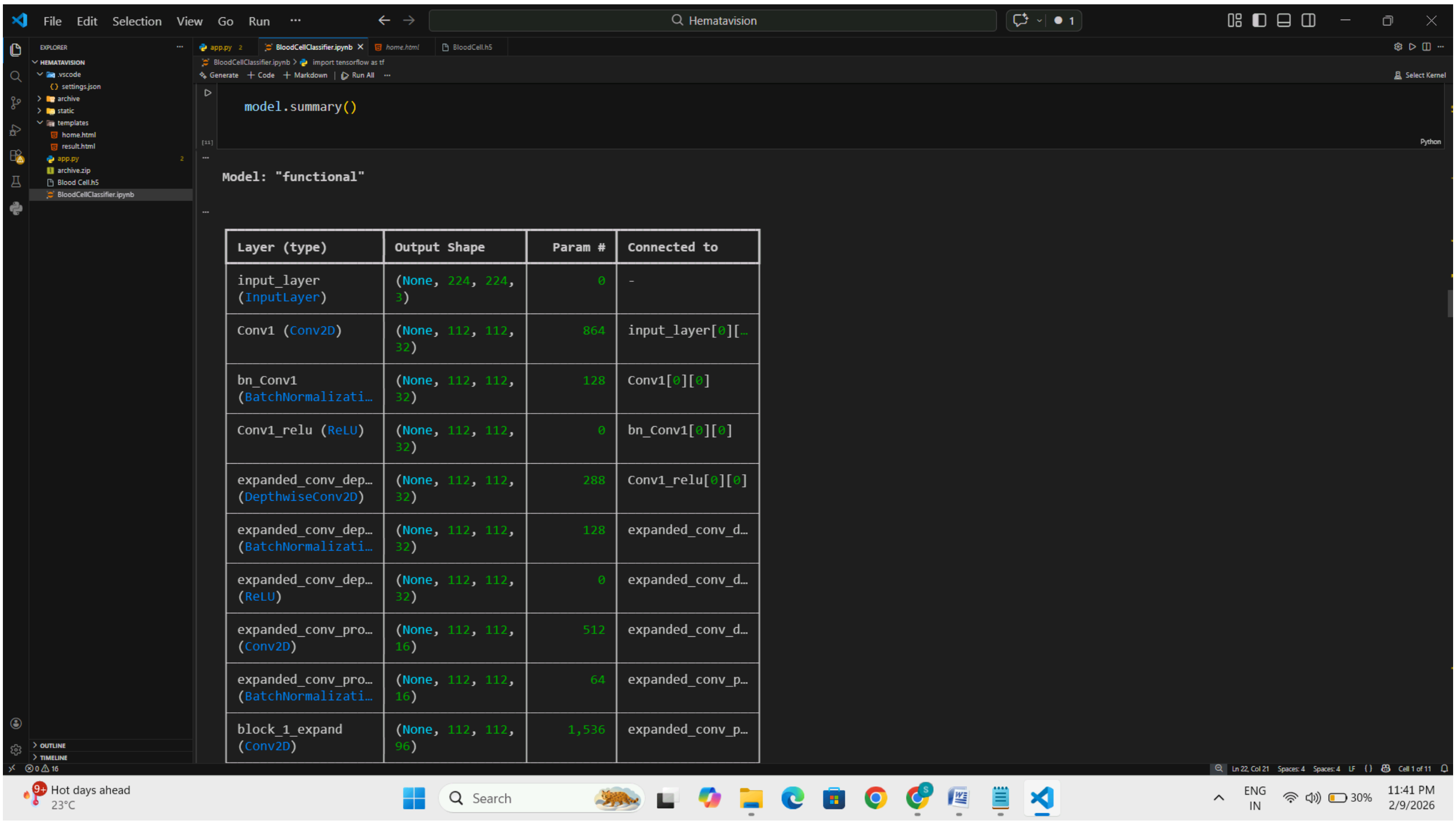


```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input

# Create training and validation generators with augmentation
datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    validation_split=0.2,  # 80% training, 20% validation
    rotation_range=15,
    zoom_range=0.1,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    horizontal_flip=True
)

# Define image size and batch
IMG_SIZE = (224, 224)
BATCH_SIZE = 32

# Load training data
train_generator = datagen.flow_from_directory(
    directory='dataset2-master/images/TRAIN',
```



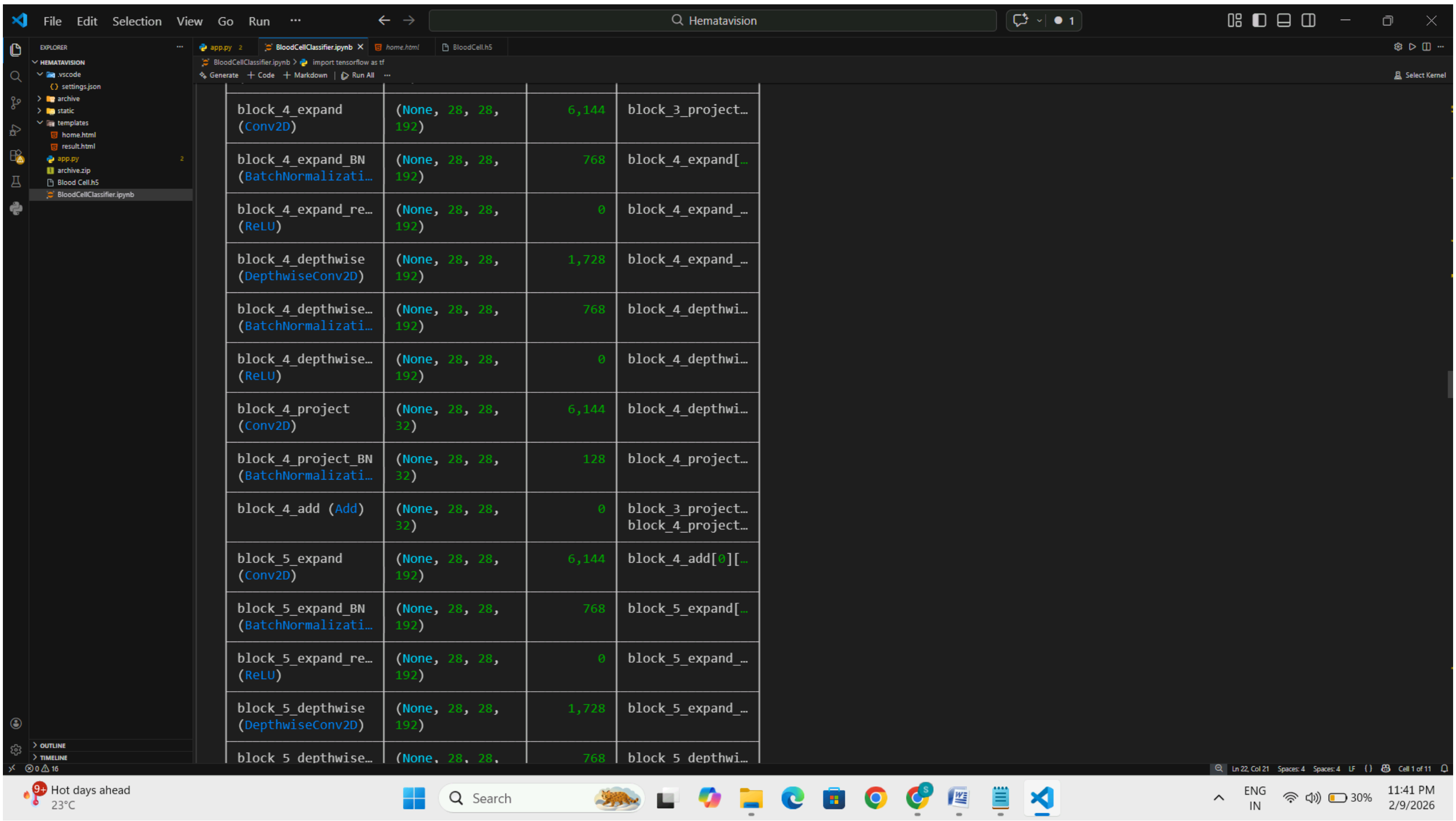
| | filepaths | labels |
|---|---|---|
| 0 | dataset2-master/images/TRAIN\monocyte_4_375.jpeg | monocyte |
| 1 | dataset2-master/images/TRAIN\eosinophil_25_32... | eosinophil |
| 2 | dataset2-master/images/TRAIN\eosinophil_7_374... | eosinophil |
| 3 | dataset2-master/images/TRAIN\eosinophil_57_34... | eosinophil |
| 4 | dataset2-master/images/TRAIN\neutrophil_105_7... | neutrophil |

```python
# Define the TRAIN folder path
data_dir = 'dataset2-master/images/TRAIN'

# Define the class labels
class_labels = ['eosinophil', 'lymphocyte', 'monocyte', 'neutrophil']

# Store file paths and labels
filepaths = []
labels = []

# Loop through each class folder
for label in class_labels:
    class_folder = os.path.join(data_dir, label)
    for file in os.listdir(class_folder):
        if file.endswith('.jpeg') or file.endswith('.png'):
            filepaths.append(os.path.join(class_folder, file))
            labels.append(label)

# Create a DataFrame
bloodCell_df = pd.DataFrame({
    'filepaths': filepaths,
    'labels': labels
})

# Shuffle the DataFrame
bloodCell_df = bloodCell_df.sample(frac=1).reset_index(drop=True)
```

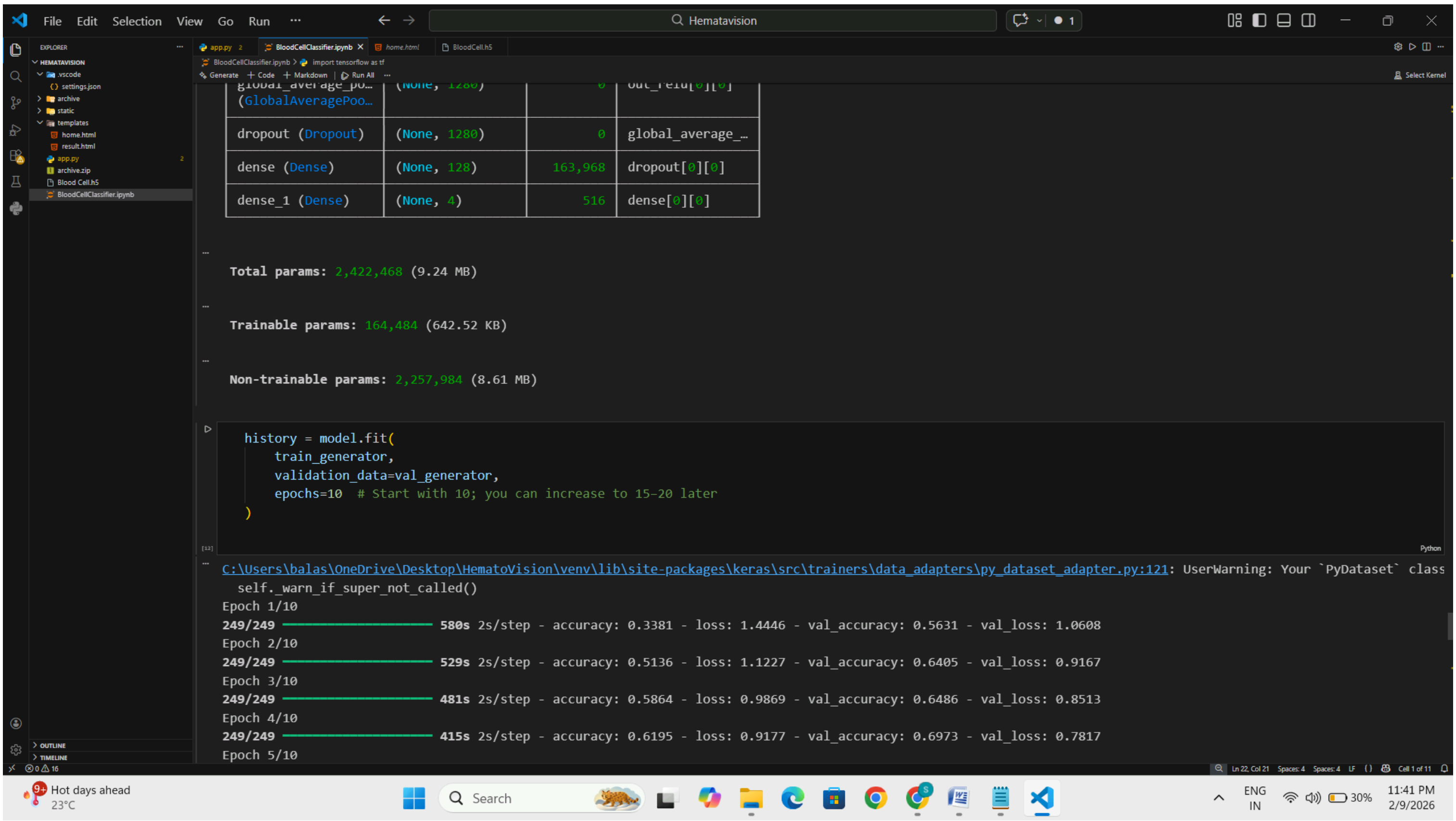


```
model.summary()
```

```
Model: "functional"

┏━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━┓
┃ Layer (type)        ┃ Output Shape      ┃    Param # ┃ Connected to      ┃
┡━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━┩
│ input_layer         │ (None, 224, 224,  │          0 │ -                 │
│ (InputLayer)        │ 3)                │            │                   │
├─────────────────────┼───────────────────┼────────────┼───────────────────┤
│ Conv1 (Conv2D)      │ (None, 112, 112,  │        864 │ input_layer[0][… │
│                     │ 32)               │            │                   │
├─────────────────────┼───────────────────┼────────────┼───────────────────┤
│ bn_Conv1            │ (None, 112, 112,  │        128 │ Conv1[0][0]       │
│ (BatchNormalizati…  │ 32)               │            │                   │
├─────────────────────┼───────────────────┼────────────┼───────────────────┤
│ Conv1_relu (ReLU)   │ (None, 112, 112,  │          0 │ bn_Conv1[0][0]    │
│                     │ 32)               │            │                   │
├─────────────────────┼───────────────────┼────────────┼───────────────────┤
│ expanded_conv_dep…  │ (None, 112, 112,  │        288 │ Conv1_relu[0][0]  │
│ (DepthwiseConv2D)   │ 32)               │            │                   │
├─────────────────────┼───────────────────┼────────────┼───────────────────┤
│ expanded_conv_dep…  │ (None, 112, 112,  │        128 │ expanded_conv_d…  │
│ (BatchNormalizati…  │ 32)               │            │                   │
├─────────────────────┼───────────────────┼────────────┼───────────────────┤
│ expanded_conv_dep…  │ (None, 112, 112,  │          0 │ expanded_conv_d…  │
│ (ReLU)              │ 32)               │            │                   │
├─────────────────────┼───────────────────┼────────────┼───────────────────┤
│ expanded_conv_pro…  │ (None, 112, 112,  │        512 │ expanded_conv_d…  │
│ (Conv2D)            │ 16)               │            │                   │
├─────────────────────┼───────────────────┼────────────┼───────────────────┤
│ expanded_conv_pro…  │ (None, 112, 112,  │         64 │ expanded_conv_p…  │
│ (BatchNormalizati…  │ 16)               │            │                   │
├─────────────────────┼───────────────────┼────────────┼───────────────────┤
│ block_1_expand      │ (None, 112, 112,  │      1,536 │ expanded_conv_p…  │
│ (Conv2D)            │ 96)               │            │                   │
└─────────────────────┴───────────────────┴────────────┴───────────────────┘
```



```
│ block_4_expand      │ (None, 28, 28,   │      6,144 │ block_3_project… │
│ (Conv2D)            │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_4_expand_BN   │ (None, 28, 28,   │        768 │ block_4_expand[… │
│ (BatchNormalizati…  │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_4_expand_re…  │ (None, 28, 28,   │          0 │ block_4_expand_… │
│ (ReLU)              │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_4_depthwise   │ (None, 28, 28,   │      1,728 │ block_4_expand_… │
│ (DepthwiseConv2D)   │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_4_depthwise…  │ (None, 28, 28,   │        768 │ block_4_depthwi… │
│ (BatchNormalizati…  │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_4_depthwise…  │ (None, 28, 28,   │          0 │ block_4_depthwi… │
│ (ReLU)              │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_4_project     │ (None, 28, 28,   │      6,144 │ block_4_depthwi… │
│ (Conv2D)            │ 32)              │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_4_project_BN  │ (None, 28, 28,   │        128 │ block_4_project… │
│ (BatchNormalizati…  │ 32)              │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_4_add (Add)   │ (None, 28, 28,   │          0 │ block_3_project… │
│                     │ 32)              │            │ block_4_project… │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_5_expand      │ (None, 28, 28,   │      6,144 │ block_4_add[0][… │
│ (Conv2D)            │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_5_expand_BN   │ (None, 28, 28,   │        768 │ block_5_expand[… │
│ (BatchNormalizati…  │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_5_expand_re…  │ (None, 28, 28,   │          0 │ block_5_expand_… │
│ (ReLU)              │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_5_depthwise   │ (None, 28, 28,   │      1,728 │ block_5_expand_… │
│ (DepthwiseConv2D)   │ 192)             │            │                  │
├─────────────────────┼──────────────────┼────────────┼──────────────────┤
│ block_5_depthwise…  │ (None, 28, 28,   │        768 │ block_5_depthwi… │
```

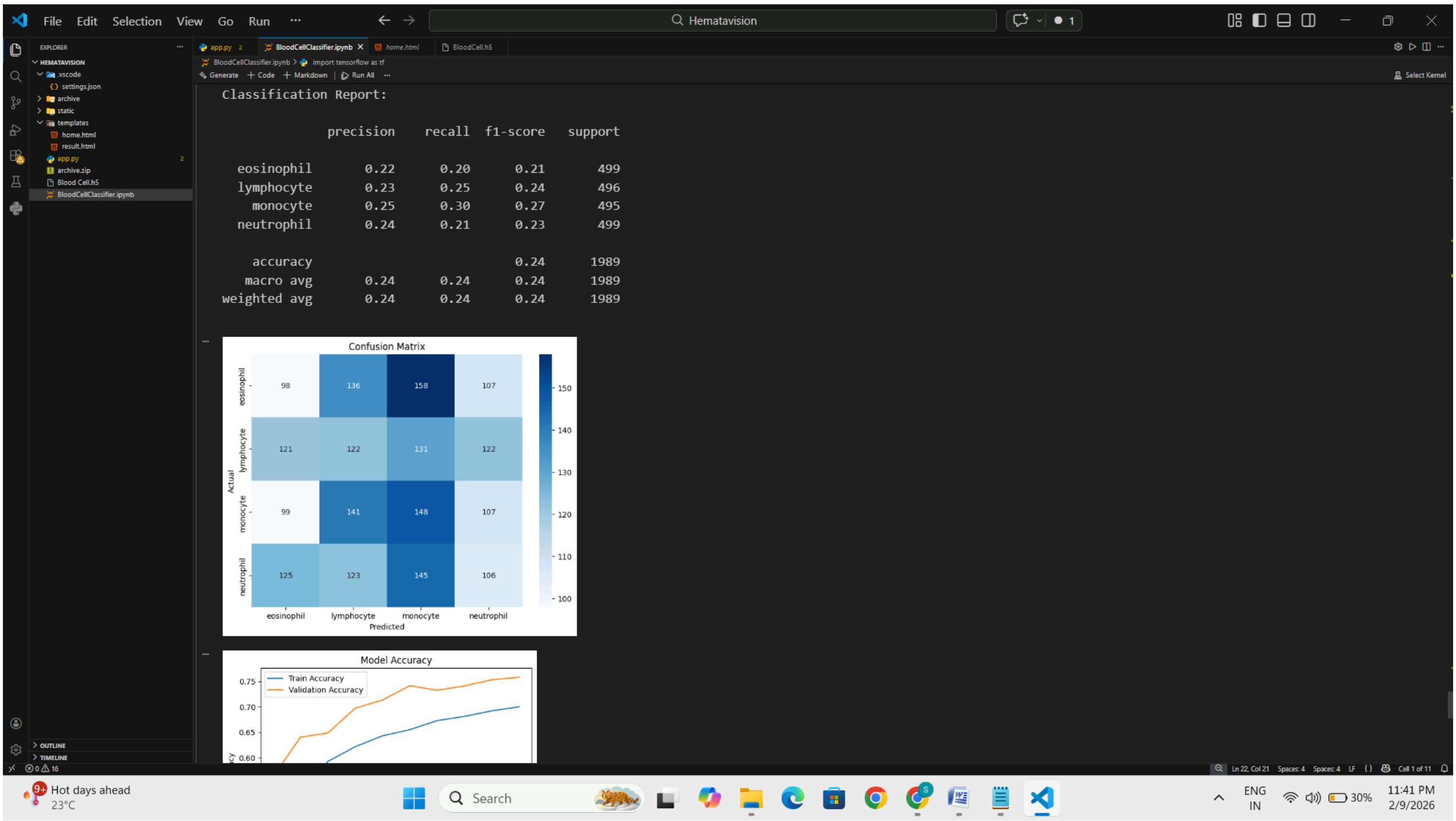


| | | | |
|---|---|---|---|
| global_average_po…<br>(GlobalAveragePoo… | (None, 1280) | 0 | out_relu[0][0] |
| dropout (Dropout) | (None, 1280) | 0 | global_average_… |
| dense (Dense) | (None, 128) | 163,968 | dropout[0][0] |
| dense_1 (Dense) | (None, 4) | 516 | dense[0][0] |

```
Total params: 2,422,468 (9.24 MB)

Trainable params: 164,484 (642.52 KB)

Non-trainable params: 2,257,984 (8.61 MB)
```

```python
history = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=10  # Start with 10; you can increase to 15-20 later
)
```

```
C:\Users\balas\OneDrive\Desktop\HematoVision\venv\lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class
  self._warn_if_super_not_called()
Epoch 1/10
249/249 ━━━━━━━━━━━━━━━━━━━━ 580s 2s/step - accuracy: 0.3381 - loss: 1.4446 - val_accuracy: 0.5631 - val_loss: 1.0608
Epoch 2/10
249/249 ━━━━━━━━━━━━━━━━━━━━ 529s 2s/step - accuracy: 0.5136 - loss: 1.1227 - val_accuracy: 0.6405 - val_loss: 0.9167
Epoch 3/10
249/249 ━━━━━━━━━━━━━━━━━━━━ 481s 2s/step - accuracy: 0.5864 - loss: 0.9869 - val_accuracy: 0.6486 - val_loss: 0.8513
Epoch 4/10
249/249 ━━━━━━━━━━━━━━━━━━━━ 415s 2s/step - accuracy: 0.6195 - loss: 0.9177 - val_accuracy: 0.6973 - val_loss: 0.7817
Epoch 5/10
```

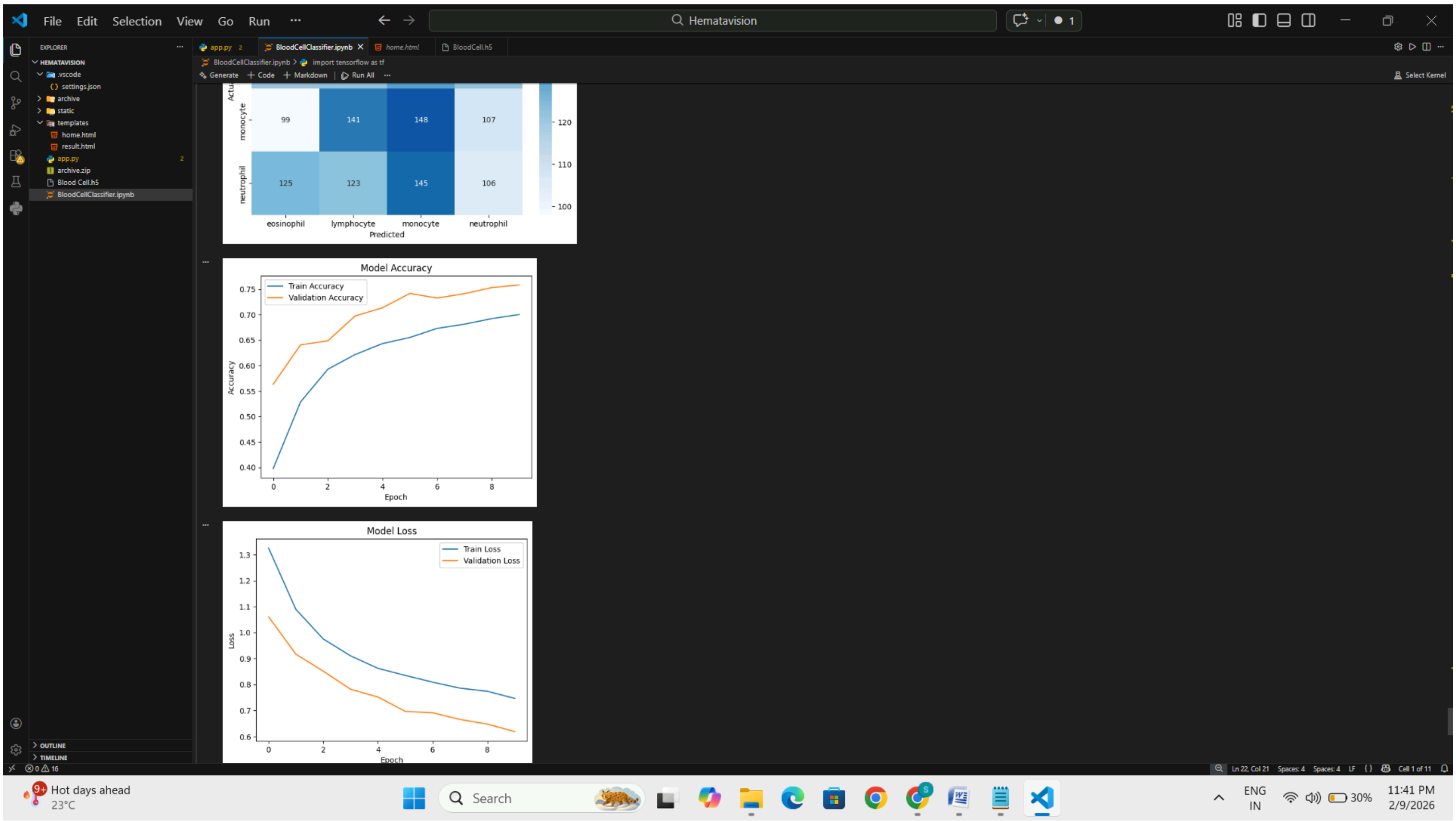


```
Classification Report:

              precision    recall  f1-score   support

  eosinophil       0.22      0.20      0.21       499
  lymphocyte       0.23      0.25      0.24       496
    monocyte       0.25      0.30      0.27       495
  neutrophil       0.24      0.21      0.23       499

    accuracy                           0.24      1989
   macro avg       0.24      0.24      0.24      1989
weighted avg       0.24      0.24      0.24      1989
```

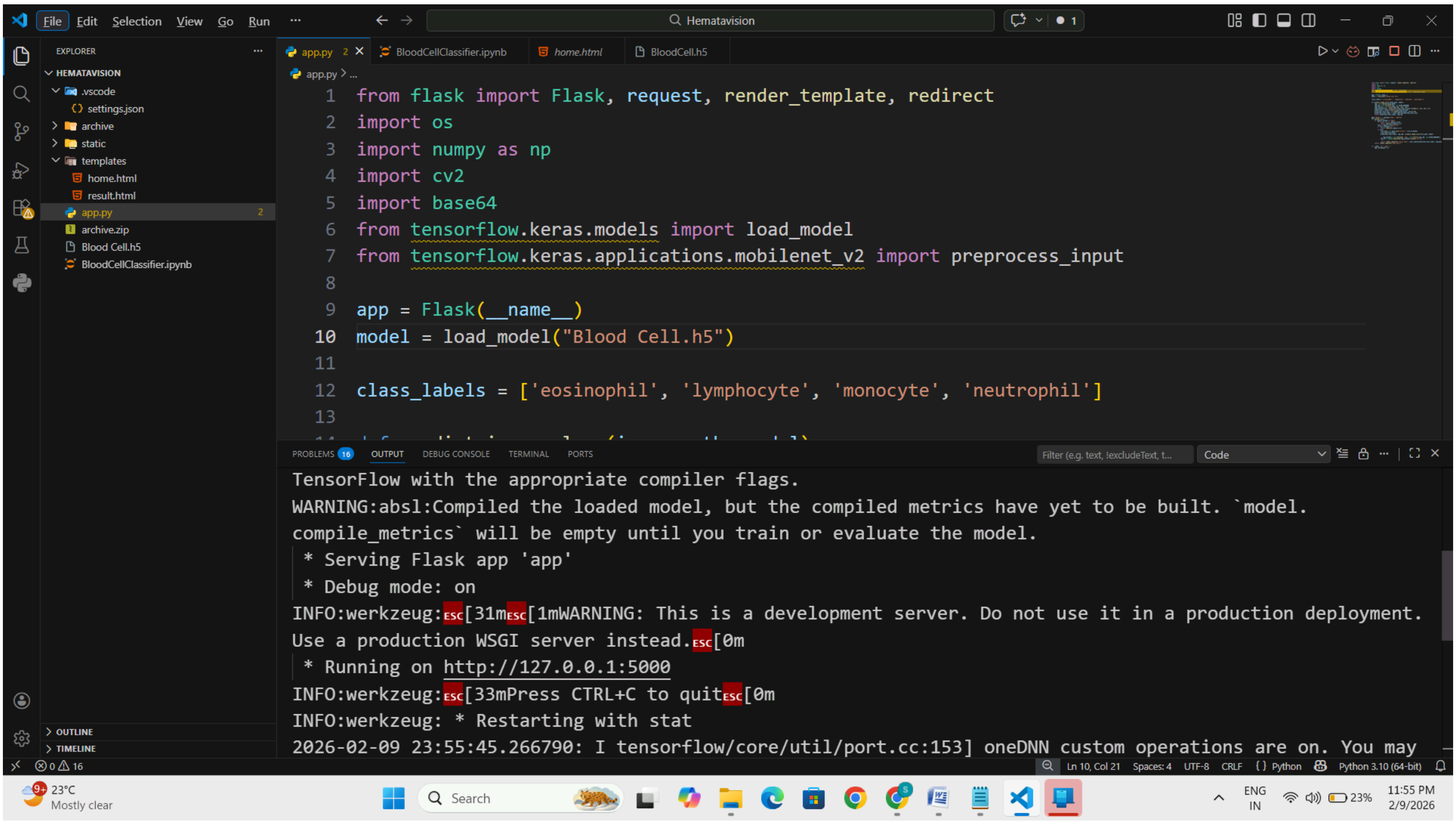
Confusion Matrix

| Actual \ Predicted | eosinophil | lymphocyte | monocyte | neutrophil |
|---|---|---|---|---|
| eosinophil | 98 | 136 | 158 | 107 |
| lymphocyte | 121 | 122 | 131 | 122 |
| monocyte | 99 | 141 | 148 | 107 |
| neutrophil | 125 | 123 | 145 | 106 |

Model Accuracy

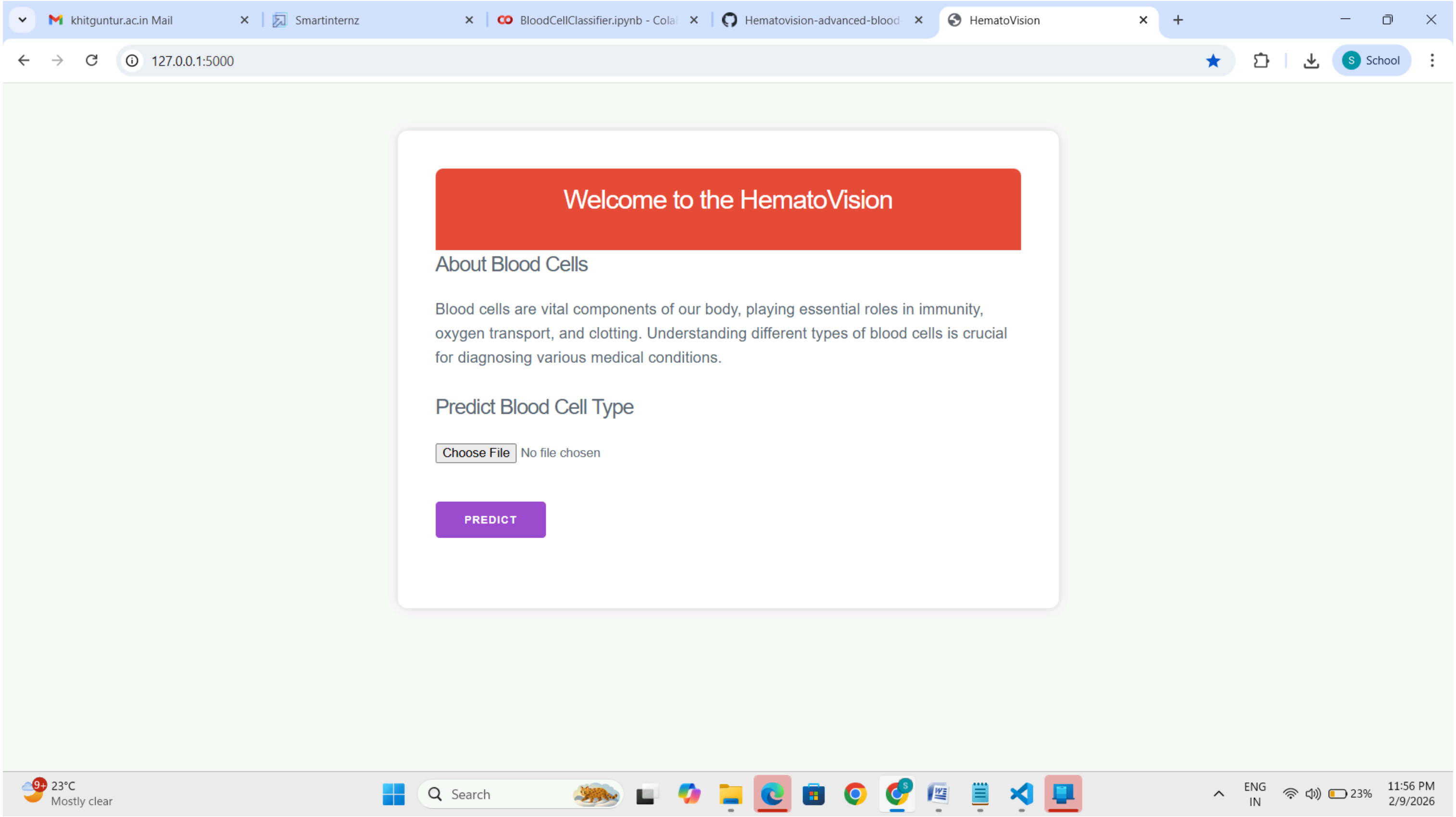


|  | eosinophil | lymphocyte | monocyte | neutrophil |
|---|---|---|---|---|
| monocyte | 99 | 141 | 148 | 107 |
| neutrophil | 125 | 123 | 145 | 106 |



```
from flask import Flask, request, render_template, redirect
import os
import numpy as np
import cv2
import base64
from tensorflow.keras.models import load_model
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input

app = Flask(__name__)
model = load_model("Blood Cell.h5")

class_labels = ['eosinophil', 'lymphocyte', 'monocyte', 'neutrophil']

```

```
TensorFlow with the appropriate compiler flags.
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.
 * Serving Flask app 'app'
 * Debug mode: on
INFO:werkzeug:ESC[31mESC[1mWARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.ESC[0m
 * Running on http://127.0.0.1:5000
INFO:werkzeug:ESC[33mPress CTRL+C to quitESC[0m
INFO:werkzeug: * Restarting with stat
2026-02-09 23:55:45.266790: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may
```



Welcome to the HematoVision

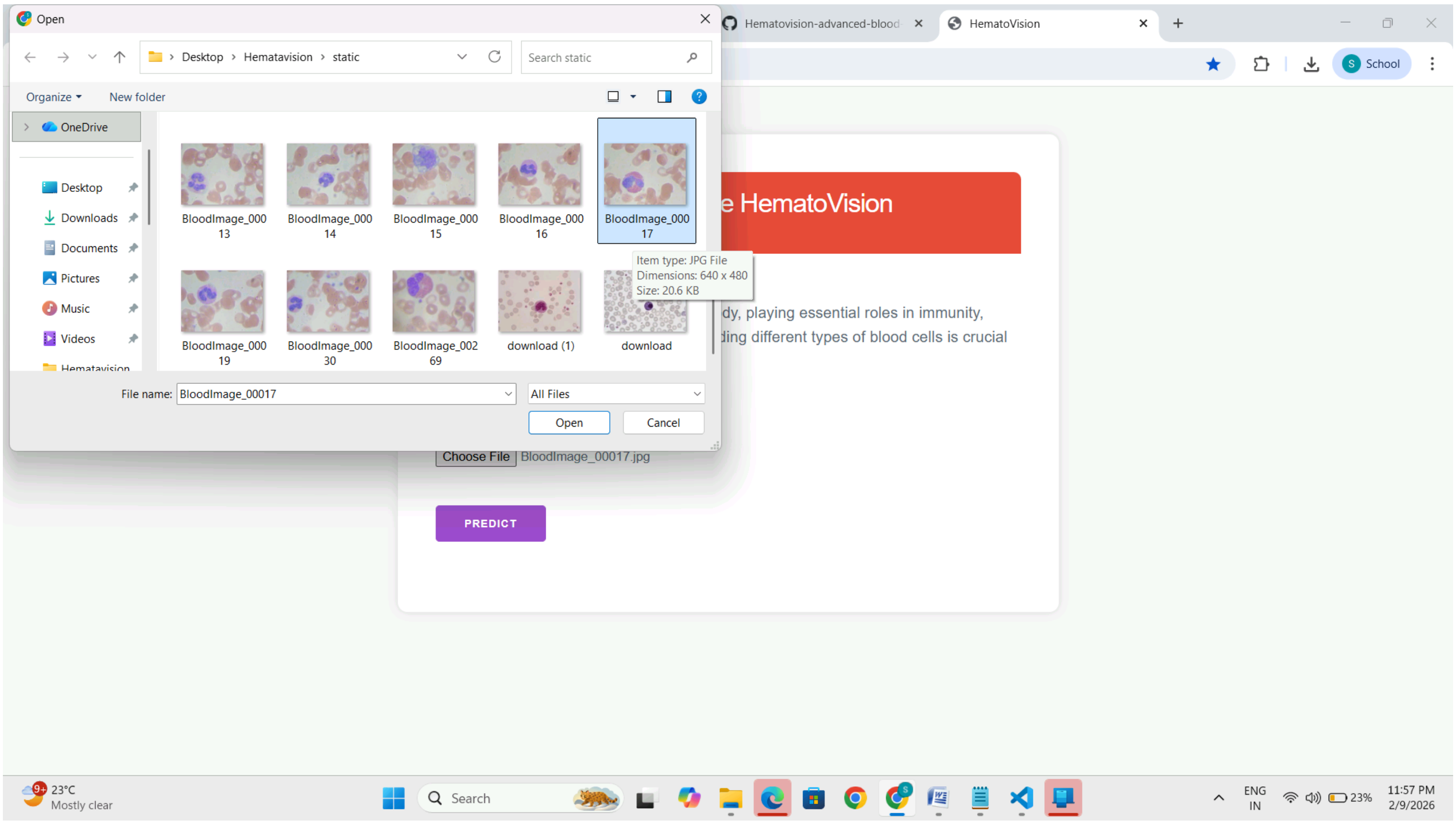
About Blood Cells

Blood cells are vital components of our body, playing essential roles in immunity, oxygen transport, and clotting. Understanding different types of blood cells is crucial for diagnosing various medical conditions.

Predict Blood Cell Type

Choose File No file chosen

PREDICT



Item type: JPG File
Dimensions: 640 x 480
Size: 20.6 KB

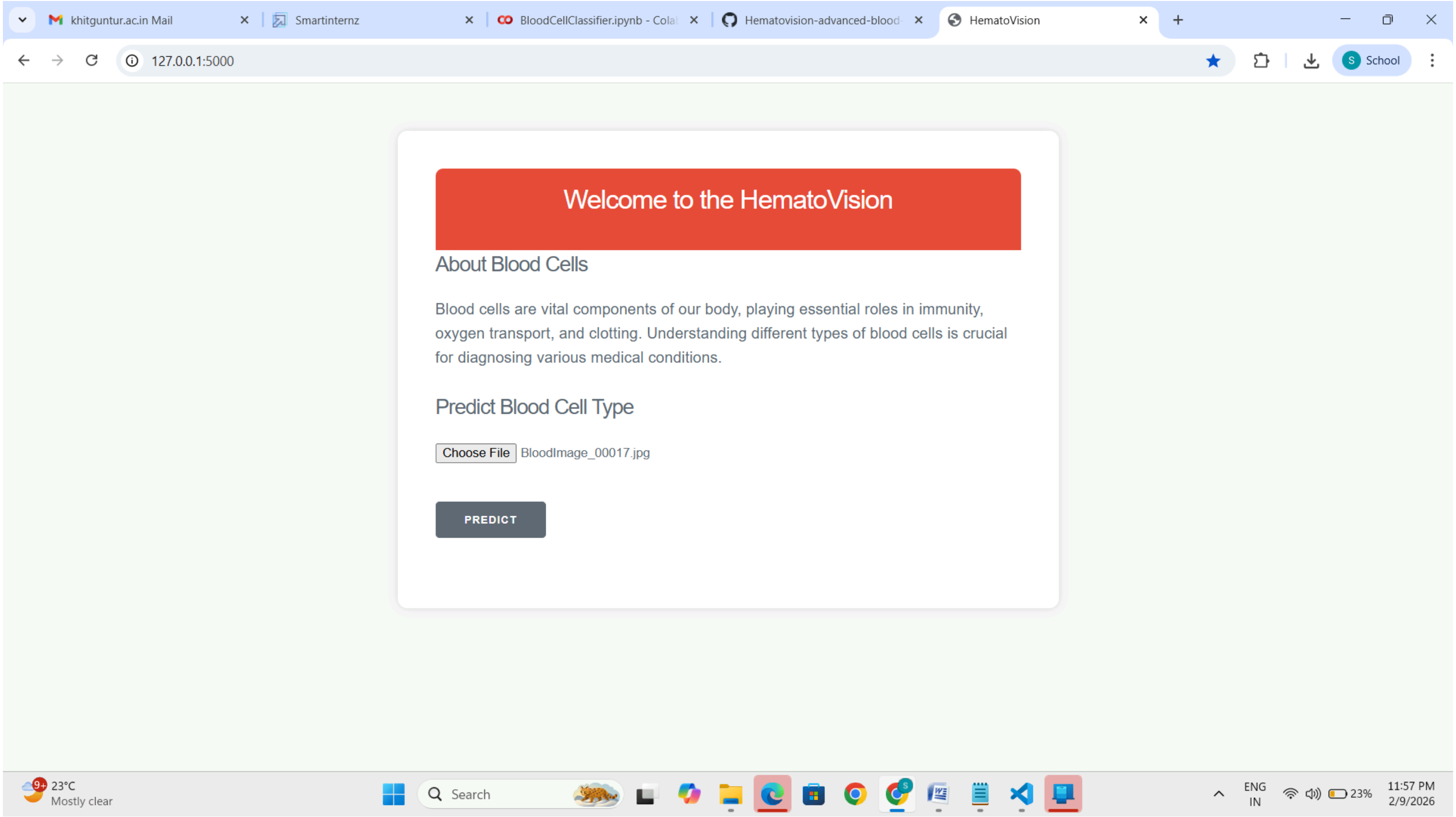
File name: BloodImage_00017

All Files

e HematoVision

dy, playing essential roles in immunity,
ding different types of blood cells is crucial

Choose File BloodImage_00017.jpg



# Welcome to the HematoVision

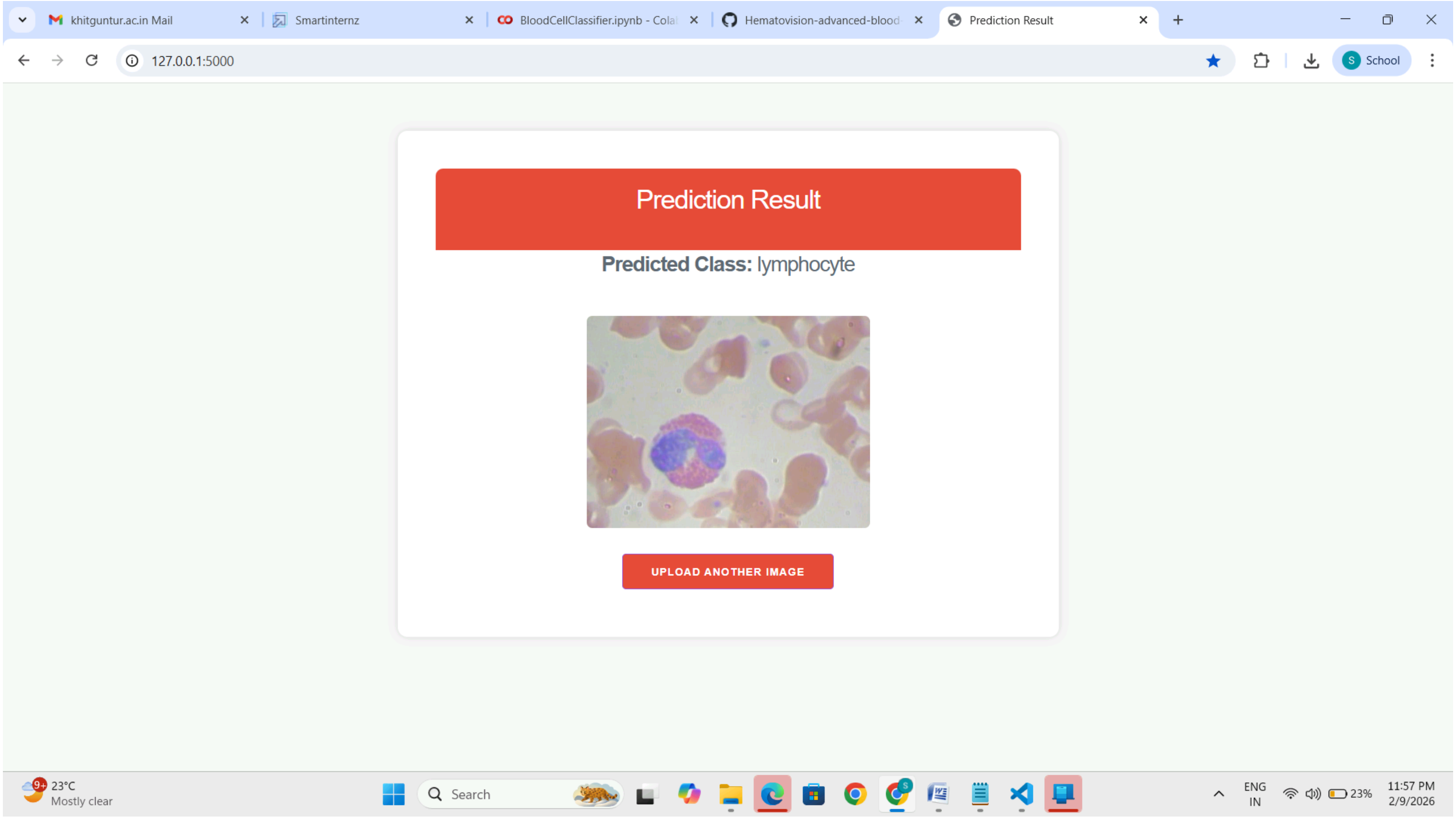
## About Blood Cells

Blood cells are vital components of our body, playing essential roles in immunity, oxygen transport, and clotting. Understanding different types of blood cells is crucial for diagnosing various medical conditions.

## Predict Blood Cell Type

Choose File BloodImage_00017.jpg

PREDICT



## Prediction Result

**Predicted Class:** lymphocyte