

COURSE NAME

Artificial intelligence and machine learning

Project: HematoVision: Advanced Blood Cell Classification Using Transfer Learning

Model: VGG16 Transfer Learning

Framework: TensorFlow / Keras

Purpose: Classify blood cell images into 4 categories

HematoVision: Blood Cell Classification Using Transfer Learning (VGG16)

📦 Import Libraries

```
import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.applications import VGG16

from tensorflow.keras import layers, models

import matplotlib.pyplot as plt
```

🔧 Load Pre-trained VGG16 Model (without top classifier)

```
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

base_model.trainable = False # Freeze feature extractor layers
```

🧠 Build Custom Classifier on Top of VGG16

```
model = models.Sequential([  
    base_model,  
    layers.Flatten(),  
    layers.Dense(256, activation='relu'),  
    layers.Dropout(0.5),  
    layers.Dense(4, activation='softmax') # Assuming 4 blood cell types  
)
```

⚙️ Compile the Model

```
model.compile(  
    optimizer='adam',  
    loss='categorical_crossentropy',  
    metrics=['accuracy']  
)
```

📁 Setup Image Data Generator for Preprocessing and Augmentation

```
datagen = ImageDataGenerator(  
    rescale=1./255,  
    validation_split=0.2  
)
```

```
train_data = datagen.flow_from_directory(
```

```
    target_size=(224, 224),
```

```
    batch_size=32,
```

```
    class_mode='categorical',
```

```
    subset='training'
```

```
)
```

```
val_data = datagen.flow_from_directory(
```

```
    'dataset_path' target_size=(224, 224),
```

```
    batch_size=32,
```

```
    class_mode='categorical',
```

```
    subset='validation'
```

```
)
```

🏋️ Train the Model

```
history = model.fit(
```

```
    train_data,
```

```
    validation_data=val_data,
```

```
    epochs=10
```

)

Plot Accuracy and Loss Graphs

```
plt.plot(history.history['accuracy'], label='Train Accuracy')  
plt.plot(history.history['val_accuracy'], label='Val Accuracy')  
plt.xlabel('Epoch')  
plt.ylabel('Accuracy')  
plt.legend()  
plt.title('Training vs Validation Accuracy')  
plt.show()
```

```
plt.plot(history.history['loss'], label='Train Loss')  
plt.plot(history.history['val_loss'], label='Val Loss')  
plt.xlabel('Epoch')  
plt.ylabel('Loss')  
plt.legend()  
plt.title('Training vs Validation Loss')  
plt.show()
```