

```

import os
import numpy as np
import cv2
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score
from sklearn.preprocessing import StandardScaler
from skimage.feature import hog
from tqdm import tqdm

# Path to your dataset
train_dir = '/content/drive/MyDrive/My_ML_Data/train'

# Function to load and preprocess images
def load_images_and_labels(directory):
    images = []
    labels = []
    classes = os.listdir(directory)

    for label, class_name in enumerate(classes):
        class_path = os.path.join(directory, class_name)
        for img_name in tqdm(os.listdir(class_path), desc=f'Loading {class_name}'):
            img_path = os.path.join(class_path, img_name)
            # Load and preprocess image
            img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
            img = cv2.resize(img, (128, 128)) # Resize to a fixed size
            images.append(img)
            labels.append(label)

    return np.array(images), np.array(labels), classes

# Load data
images, labels, class_names = load_images_and_labels(train_dir)
print(f"Loaded {len(images)} images from {len(class_names)} classes.")

➡ Loading 0-: 100%|██████████| 498/498 [00:09<00:00, 52.82it/s]
Loading 0+: 100%|██████████| 596/596 [00:10<00:00, 54.54it/s]
Loading A-: 100%|██████████| 610/610 [00:10<00:00, 57.77it/s]
Loading A+: 100%|██████████| 530/530 [00:09<00:00, 53.37it/s]
Loading AB+: 100%|██████████| 505/505 [00:08<00:00, 57.25it/s]
Loading AB-: 100%|██████████| 551/551 [00:10<00:00, 50.63it/s]
Loading B-: 100%|██████████| 517/517 [00:09<00:00, 51.93it/s]
Loading B+: 100%|██████████| 498/498 [00:08<00:00, 57.42it/s]
Loaded 4305 images from 8 classes.

# Function to extract HOG features
def extract_hog_features(images):

```

```

features = []
for img in tqdm(images, desc="Extracting HOG features"):
    hog_features = hog(img, orientations=9, pixels_per_cell=(8, 8),
                       cells_per_block=(2, 2), block_norm='L2-Hys', feature_vector=True)
    features.append(hog_features)
return np.array(features)

```

```

# Extract HOG features
features = extract_hog_features(images)
print(f"Extracted features shape: {features.shape}")

```

➡ Extracting HOG features: 100% |██████████| 4305/4305 [00:29<00:00, 144.85it/s]  
 Extracted features shape: (4305, 8100)

```

X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_
print(f"Training samples: {len(X_train)}, Testing samples: {len(X_test)}")

```

➡ Training samples: 3444, Testing samples: 861

```

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

```

```

# Initialize and train SVM
svm_model = SVC(kernel='rbf', C=1, gamma='scale', probability=True)
svm_model.fit(X_train_scaled, y_train)

```

```

# Predict on test data
y_pred = svm_model.predict(X_test_scaled)

```

```

# Evaluate model
print("Classification Report:")
print(classification_report(y_test, y_pred, target_names=class_names))
print("Accuracy:", accuracy_score(y_test, y_pred))

```

➡ Classification Report:

	precision	recall	f1-score	support
O-	0.84	0.87	0.85	113
O+	0.90	0.88	0.89	124
A-	0.92	0.92	0.92	136
A+	0.94	0.95	0.94	98
AB+	0.94	0.94	0.94	114
AB-	0.87	0.83	0.85	93
B-	0.91	0.96	0.94	78
B+	0.95	0.93	0.94	105

accuracy			0.91	861
macro avg	0.91	0.91	0.91	861
weighted avg	0.91	0.91	0.91	861

Accuracy: 0.908246225319396

```
import joblib
```

```
# Save model
```

```
joblib.dump(svm_model, 'svm_blood_group_model.pkl')
```

```
# Load model
```

```
svm_model = joblib.load('svm_blood_group_model.pkl')
```