filepath = '/content/Shoe vs Sandal vs Boot Dataset'

test\_size = 0.2 kích thước tập test

dev\_size = 0.2 kích thước tập validation

n\_H = 102 2 dòng khai báo kích thước ảnh

n\_W = 136

epochs = 30 số lần lặp

learning\_rate=0.0007 tỷ lệ học (learning rate) cho bộ tối ưu hóa Adam

random.seed(6278)

gán nhãn

X = []

Y = []

for shoe\_folder in os.scandir(filepath): # lặp qua các thư mục con

    shoe\_type = os.path.basename(shoe\_folder) # lưu tên thư mục làm nhãn

    pbar = tqdm(os.scandir(shoe\_folder.path), total=5000) # Sử dụng tqdm iter để hiển thị tiến độ đọc trong hình ảnh

    for file in pbar: # lặp qua các tệp

        pbar.set\_description(f"Processing {shoe\_type} Images") Dòng này cập nhật mô tả của thanh tiến trình hiển thị tiến trình đọc hình ảnh cho từng loại giày (shoe\_type).

        img = cv2.imread(file.path)Dòng này sử dụng thư viện OpenCV để đọc hình ảnh từ đường dẫn của tệp.

        img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB) #chuyển đổi đọc defualt trong cv2 BGR sang RGB

        img\_array = np.asarray(img) Dòng này chuyển đổi hình ảnh thành một mảng numpy

        if img\_array.shape != (102,136,3): # Do conditional resizing to save computing power

            img\_array = resize(img\_array, [n\_H,n\_W], anti\_aliasing=True) # resize to ensure all images are 136 x 102

        X.append(img\_array)

        Y.append(shoe\_type)

Trộn dl

X\_Y = list(zip(X,Y)) # Zip hình ảnh và nhãn liên quan

random.shuffle(X\_Y) # Xáo trộn

X, Y = zip(\*X\_Y) # Unzip back into X and Y

Chia tập, dl + sử lý nhãn đầu vào

num\_dev = int(len(X) \* dev\_size) #tính số mẫu trong tập validation

num\_test = int(len(X) \* test\_size) #tính số mẫu trong tập test

X\_arr = np.asarray(X) / 255 #chẩn hóa gt trong mảng về 0,1

Y\_arr = np.asarray(Y).reshape(-1,1) #chuyển đổi danh sách nhãn y thành mảng numpy và xeepslaij thành 1 cột

del X, Y #xóa X,Y gải phóng bộ nhớ

#chuyển các nhãn thành dạng onehot

ohe = OneHotEncoder(categories='auto', sparse=False)

Y\_ohe = ohe.fit\_transform(Y\_arr)

#chia tập dl

X\_train = X\_arr[:-(num\_test + num\_dev)] lấy các mẫu từ đầu đến vị trí -(num\_test + num\_dev).

X\_dev = X\_arr[-(num\_test + num\_dev):-num\_test] mẫu từ vị trí -(num\_test + num\_dev) đến vị trí -num\_test

X\_test = X\_arr[-num\_test:] lấy các mẫu từ vị trí -num\_test đến cuối danh sách

Y\_train = Y\_ohe[:-(num\_test + num\_dev)]

Y\_dev = Y\_ohe[-(num\_test + num\_dev):-num\_test]

Y\_test = Y\_ohe[-num\_test:]

del X\_arr, Y\_arr, Y\_ohe #giải phóng

print("Training X:", X\_train.shape)

print("Training Y:", Y\_train.shape)

print("Development X:", X\_dev.shape)

print("Development Y:", Y\_dev.shape)

print("Test X:", X\_test.shape)

print("Test Y:", Y\_test.shape)