

PHÂN LOẠI HOA

Nhóm Bar

Trần Hằng Ny - 190501030

Huỳnh Phạm Ngọc Lành - 190501006

Trần Kiều Trinh -190501016

Abstract

Với sự phát triển của Thị giác máy tính, không khó để bắt gặp những áp dụng của Thị giác máy tính vào đời sống phổ biến như phân loại, nhận diện đối tượng. Tùy vào công việc và mục đích của người sử dụng mà các model được quyết định sử dụng. Từ điều kiện sử dụng, mà nhóm đã quyết định sử dụng mô hình MobileNets để ứng dụng nhận diện hoa trên điện thoại di động. Với dữ liệu được nhóm chuẩn bị hơn 3000 ảnh và sau 10 lần epochs, kết quả đạt được là 83,76%. Tuy kết quả đạt được chỉ với mức khá cao, song, mô hình này đã phù hợp cho việc ứng dụng nhận diện và phân loại hoa trên điện thoại di động là hết sức phù hợp.

I. Introduction

MobileNet được phát triển bởi đội ngũ Google, mô hình sử dụng cách tính chập tích mang tên DSC (Depthwise Separable Convolution) nhằm giảm kích thước mô hình và giảm độ phức tạp tính toán. Do đó, MobileNet thường được sử dụng cho các ứng dụng Computer Vision trên các thiết bị nhỏ gọn như điện thoại thông minh hay thiết bị nhúng.

II. Approach

1. Thông tin về bộ dữ liệu

Bộ dữ liệu được chuẩn bị với 3670 ảnh, với 5 loại hoa khác nhau [daisy, dandelion, roses, sunflowers, tulips].

Các ảnh được chuẩn bị ban đầu không có kiến thức cố định và nó gần như là rất lớn, vì vậy, khi xử lý ảnh đầu vào, ảnh được tinh chỉnh với kích thước là 224x224.

Label của ảnh sẽ trực tiếp lấy tên của Folder trực tiếp chứa ảnh sẽ phân loại.

Tỉ lệ phân loại các ảnh là 20-80 với 80% dữ liệu được sử dụng để đào tạo và 20% được sử dụng để xác thực.

2. Thực hiện

2.1. Import các thư viện cần thiết cho dự án

```

1 import cv2
2 from keras.applications.mobilenet import MobileNet
3 from keras.layers import GlobalAveragePooling2D, Dense, Dropout
4 from keras.models import Model
5
6 from keras.preprocessing.image import ImageDataGenerator
7 import keras
8 from keras.callbacks import ModelCheckpoint
9

```

2.2. Tạo model

```

10 # 1. Định nghĩa tham số
11 n_class = 5
12
13 # 2. Build model
14 def get_model():
15     # Tạo base model
16     base_model = MobileNet(include_top=False, weights="imagenet", input_shape=(224,224,3))
17     # Tạo model chính
18     x = base_model.output
19     # Add some new Fully connected layers to
20     x = GlobalAveragePooling2D()(x)
21     x = Dense(1024, activation='relu')(x)
22     x = Dropout(0.25)(x)
23     x = Dense(1024, activation='relu')(x)
24     x = Dropout(0.25)(x)
25     x = Dense(512, activation='relu')(x)
26     outs = Dense(n_class, activation='softmax')(x)
27
28     # Đóng băng các layer của base_model
29     for layer in base_model.layers:
30         layer.trainable = False
31
32     model = Model(inputs=base_model.inputs, outputs= outs)
33     return model
34
35 model = get_model()
36 model.summary()

```

2.3. Tinh chỉnh bộ dữ liệu

```

38 # 3. Make data
39 data_folder = "data"
40
41 train_datagen = ImageDataGenerator(preprocessing_function= keras.applications.mobilenet.preprocess_input,
42                                     rotation_range=0.2,width_shift_range=0.2,height_shift_range=0.2,
43                                     shear_range=0.3,zoom_range=0.5,
44                                     horizontal_flip=True, vertical_flip=True,
45                                     validation_split=0.2)
46
47 train_generator = train_datagen.flow_from_directory(data_folder,
48                                                     target_size=(224, 224),
49                                                     batch_size=64,
50                                                     class_mode='categorical',
51                                                     subset='training')
52
53 validation_generator = train_datagen.flow_from_directory(
54     data_folder, # same directory as training data
55     target_size=(224, 224),
56     batch_size=64,
57     class_mode='categorical',
58     subset='validation') # set as validation data
59
60 classes = train_generator.class_indices
61 print(classes)
62 classes = list(classes.keys())

```

2.4. Thực hiện train và lưu model đã train

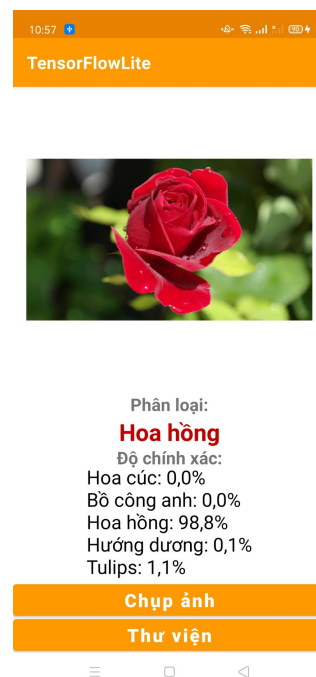
```

64 # 4. Train model
65 n_epochs = 10
66 batch_size = 64
67 model.compile(optimizer='sgd', Loss='categorical_crossentropy', metrics=['accuracy'])
68
69 checkpoint = ModelCheckpoint('models/best.hdf5', monitor='val_loss', save_best_only = True, mode='auto')
70 callback_list = [checkpoint]
71
72 step_train = train_generator.n//batch_size
73 step_val = validation_generator.n//batch_size
74
75 model.fit_generator(generator=train_generator, steps_per_epoch=step_train,
76                    validation_data=validation_generator,
77                    validation_steps=step_val,
78                    callbacks=callback_list,
79                    epochs=n_epochs)
80
81
82 # 5. Lưu model
83 model.save('models/model.h5')
84

```

III. Results

1. Kết quả đạt được



Kết quả đạt được với tỉ lệ khá cao, hơn dự kiến ban đầu.

2. So sánh

- InceptionV3
loss: 0.2158 - accuracy: 0.9289
- MobileNet
loss: 0.4468 - accuracy: 0.8376
- Sequential

loss: 0.1953 - accuracy: 0.9358

IV. Discussion and Future Work

Điều kiện đầu tiên là sự phù hợp của MobileNet với việc xây dựng model cho các ứng dụng nhận diện cho Android, đánh đổi đi là độ chính xác của model đem lại bị giảm đi một phần. Tuy nhiên, kết quả đem lại với độ chính xác cao hơn 80% qua dự án “Nhận diện hoa” đem lại không thể phủ nhận sự tiện lợi của model MobileNet. Trong tương lai, nhóm sẽ thử nghiệm với bộ dữ liệu nhiều hơn, giảm thiểu những phần dư thừa trong hiện tại và tìm đến một vài model khác để hoàn thiện hơn về độ chính xác trong nhận diện hoa.

Reference

[1] Nhận diện hoa với model InceptionV3 :

https://github.com/oleksandr-g-rock/flower_classification

[2] Nhận diện hoa với model Sequential:

https://github.com/NajiAboo/CNN_Architecutre_Comparison