We are having a problem billing your account. Please update your payment method or call your payment provider for details on why the transaction failed.
You can contact support with any questions.

Image-Processing / **Image_classification_via_fine_tuning_with_EfficientNet.ipynb**

VuHa123 Kaggle Notebook | Image_Clasification | Version 9 b897cfb · 5 minutes ago

1 lines (1 loc) · 1.1 MB

Open in Kaggle

Setup and data loading

In [1]:
```
import numpy as np
import tensorflow_datasets as tfds
import tensorflow as tf
import matplotlib.pyplot as plt
import keras
from keras import layers
from keras.applications import EfficientNetB0

IMG_SIZE = 224
BATCH_SIZE = 64
```

In [2]:
```
# Tập dataset Stanford dog chưa hình ảnh của 120 giống chó, trong
dataset_name = "stanford_dogs"
(ds_train, ds_test), ds_info = tfds.load(#ds_info: chưa thông tin về
    dataset_name, split=["train", "test"], with_info=True, as_supervi
)
NUM_CLASSES = ds_info.features["label"].num_classes
```

```
Downloading and preparing dataset 778.12 MiB (download: 778.12 MiB, ge
nerated: Unknown size, total: 778.12 MiB) to /root/tensorflow_dataset
s/stanford_dogs/0.2.0...
Dl Completed...: 0 url [00:00, ? url/s]
Dl Size...: 0 MiB [00:00, ? MiB/s]
Dl Completed...: 0 url [00:00, ? url/s]
Dl Size...: 0 MiB [00:00, ? MiB/s]
Extraction completed...: 0 file [00:00, ? file/s]
Generating splits...:   0%|          | 0/2 [00:00<?, ? splits/s]
Generating train examples...:   0%|          | 0/12000 [00:00<?, ? exa
mples/s]
Shuffling /root/tensorflow_datasets/stanford_dogs/incomplete.D9TKFB_0.
2.0/stanford_dogs-train.tfrecord*...:   …
Generating test examples...:   0%|          | 0/8580 [00:00<?, ? examp
les/s]
Shuffling /root/tensorflow_datasets/stanford_dogs/incomplete.D9TKFB_0.
2.0/stanford_dogs-test.tfrecord*...:   0…
Dataset stanford_dogs downloaded and prepared to /root/tensorflow_data
sets/stanford_dogs/0.2.0. Subsequent calls will reuse this data.
```

In [3]:
```
size = (IMG_SIZE, IMG_SIZE)
ds_train = ds_train.map(lambda image, label: (tf.image.resize(image,
ds_test = ds_test.map(lambda image, label: (tf.image.resize(image, si
```

Visualizing the data

In [4]:
```
def format_label(label):
    string_label = label_info.int2str(label)
    return string_label.split("-")[1]


label_info = ds_info.features["label"]
for i, (image, label) in enumerate(ds_train.take(9)):
```

```
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(image.numpy().astype("uint8"))
    plt.title("{}".format(format_label(label)))
    plt.axis("off")
```

Data augmentation

In [5]:

```
# Lớp tăng cường dữ liệu
img_augmentation_layers = [
    layers.RandomRotation(factor=0.15),
    layers.RandomTranslation(height_factor=0.1, width_factor=0.1),
    layers.RandomFlip(),
    layers.RandomContrast(factor=0.1),
]

#Hàm áp dụng tăng cường dữ liệu cho các ảnh
def img_augmentation(images):
    for layer in img_augmentation_layers:
        images = layer(images)
    return images
```

In [6]:

```
for image, label in ds_train.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        aug_img = img_augmentation(np.expand_dims(image.numpy(), axis
        aug_img = np.array(aug_img)
        plt.imshow(aug_img[0].astype("uint8"))
        plt.title("{}".format(format_label(label)))
        plt.axis("off")
```

In [7]:

```
# One-hot / categorical encoding
def input_preprocess_train(image, label):
    image = img_augmentation(image)
    label = tf.one_hot(label, NUM_CLASSES)
    return image, label


def input_preprocess_test(image, label):
    label = tf.one_hot(label, NUM_CLASSES)
    return image, label


ds_train = ds_train.map(input_preprocess_train, num_parallel_calls=tf
ds_train = ds_train.batch(batch_size=BATCH_SIZE, drop_remainder=True)
ds_train = ds_train.prefetch(tf.data.AUTOTUNE)

ds_test = ds_test.map(input_preprocess_test, num_parallel_calls=tf.da
ds_test = ds_test.batch(batch_size=BATCH_SIZE, drop_remainder=True)
```

Training a model from scratch

In [8]:

```
model = EfficientNetB0(
    include_top=True,
    weights=None,
    classes=NUM_CLASSES,
    input_shape=(IMG_SIZE, IMG_SIZE, 3),
)
model.compile(optimizer="adam", loss="categorical_crossentropy", metr

model.summary()

epochs = 15  # @param {type: "slider", min:10, max:100}
hist = model.fit(ds_train, epochs=epochs, validation_data=ds_test)
```

Model: "efficientnetb0"

Layer (type)	Output Shape	Pa

input_layer (InputLayer)	(None, 224, 224, 3)	
rescaling (Rescaling)	(None, 224, 224, 3)	
normalization (Normalization)	(None, 224, 224, 3)	
stem_conv_pad (ZeroPadding2D)	(None, 225, 225, 3)	
stem_conv (Conv2D)	(None, 112, 112, 32)	
stem_bn (BatchNormalization)	(None, 112, 112, 32)	
stem_activation (Activation)	(None, 112, 112, 32)	
block1a_dwconv (DepthwiseConv2D)	(None, 112, 112, 32)	
block1a_bn (BatchNormalization)	(None, 112, 112, 32)	
block1a_activation (Activation)	(None, 112, 112, 32)	
block1a_se_squeeze (GlobalAveragePooling2D)	(None, 32)	
block1a_se_reshape (Reshape)	(None, 1, 1, 32)	
block1a_se_reduce (Conv2D)	(None, 1, 1, 8)	
block1a_se_expand (Conv2D)	(None, 1, 1, 32)	
block1a_se_excite (Multiply)	(None, 112, 112, 32)	
block1a_project_conv (Conv2D)	(None, 112, 112, 16)	
block1a_project_bn (BatchNormalization)	(None, 112, 112, 16)	
block2a_expand_conv (Conv2D)	(None, 112, 112, 96)	

(Activation)		
block2a_dwconv_pad (ZeroPadding2D)	(None, 113, 113, 96)	
block2a_dwconv (DepthwiseConv2D)	(None, 56, 56, 96)	
block2a_bn (BatchNormalization)	(None, 56, 56, 96)	

block2a_activation (Activation)	(None, 56, 56, 96)	
block2a_se_squeeze (GlobalAveragePooling2D)	(None, 96)	
block2a_se_reshape (Reshape)	(None, 1, 1, 96)	
block2a_se_reduce (Conv2D)	(None, 1, 1, 4)	
block2a_se_expand (Conv2D)	(None, 1, 1, 96)	
block2a_se_excite (Multiply)	(None, 56, 56, 96)	
block2a_project_conv (Conv2D)	(None, 56, 56, 24)	
block2a_project_bn (BatchNormalization)	(None, 56, 56, 24)	
block2b_expand_conv (Conv2D)	(None, 56, 56, 144)	
block2b_expand_bn (BatchNormalization)	(None, 56, 56, 144)	
block2b_expand_activation (Activation)	(None, 56, 56, 144)	
block2b_dwconv (DepthwiseConv2D)	(None, 56, 56, 144)	
block2b_bn (BatchNormalization)	(None, 56, 56, 144)	
block2b_activation (Activation)	(None, 56, 56, 144)	
block2b_se_squeeze (GlobalAveragePooling2D)	(None, 144)	
block2b_se_reshape (Reshape)	(None, 1, 1, 144)	
block2b_se_reduce (Conv2D)	(None, 1, 1, 6)	
block2b_se_expand (Conv2D)	(None, 1, 1, 144)	
block2b_se_excite (Multiply)	(None, 56, 56, 144)	
block2b_project_conv (Conv2D)	(None, 56, 56, 24)	
block2b_project_bn (BatchNormalization)	(None, 56, 56, 24)	
block2b_drop (Dropout)	(None, 56, 56, 24)	

block2b_add (Add)	(None, 56, 56, 24)	
block3a_expand_conv (Conv2D)	(None, 56, 56, 144)	
block3a_expand_bn (BatchNormalization)	(None, 56, 56, 144)	
block3a_expand_activation (Activation)	(None, 56, 56, 144)	
block3a_dwconv_pad (ZeroPadding2D)	(None, 59, 59, 144)	
block3a_dwconv (DepthwiseConv2D)	(None, 28, 28, 144)	
block3a_bn (BatchNormalization)	(None, 28, 28, 144)	
block3a_activation (Activation)	(None, 28, 28, 144)	
block3a_se_squeeze (GlobalAveragePooling2D)	(None, 144)	
block3a_se_reshape (Reshape)	(None, 1, 1, 144)	
block3a_se_reduce (Conv2D)	(None, 1, 1, 6)	
block3a_se_expand (Conv2D)	(None, 1, 1, 144)	
block3a_se_excite (Multiply)	(None, 28, 28, 144)	
block3a_project_conv (Conv2D)	(None, 28, 28, 40)	
block3a_project_bn (BatchNormalization)	(None, 28, 28, 40)	
block3b_expand_conv (Conv2D)	(None, 28, 28, 240)	
block3b_expand_bn (BatchNormalization)	(None, 28, 28, 240)	
block3b_expand_activation (Activation)	(None, 28, 28, 240)	
block3b_dwconv (DepthwiseConv2D)	(None, 28, 28, 240)	
block3b_bn (BatchNormalization)	(None, 28, 28, 240)	
block3b_activation (Activation)	(None, 28, 28, 240)	
block3b_se_squeeze (GlobalAveragePooling2D)	(None, 240)	

block3b_se_reshape (Reshape)	(None, 1, 1, 240)	
block3b_se_reduce (Conv2D)	(None, 1, 1, 10)	
block3b_se_expand (Conv2D)	(None, 1, 1, 240)	
block3b_se_excite (Multiply)	(None, 28, 28, 240)	
block3b_project_conv (Conv2D)	(None, 28, 28, 40)	
block3b_project_bn (BatchNormalization)	(None, 28, 28, 40)	
block3b_drop (Dropout)	(None, 28, 28, 40)	
block3b_add (Add)	(None, 28, 28, 40)	
block4a_expand_conv (Conv2D)	(None, 28, 28, 240)	
block4a_expand_bn (BatchNormalization)	(None, 28, 28, 240)	
block4a_expand_activation (Activation)	(None, 28, 28, 240)	
block4a_dwconv_pad (ZeroPadding2D)	(None, 29, 29, 240)	
block4a_dwconv (DepthwiseConv2D)	(None, 14, 14, 240)	
block4a_bn (BatchNormalization)	(None, 14, 14, 240)	
block4a_activation (Activation)	(None, 14, 14, 240)	
block4a_se_squeeze (GlobalAveragePooling2D)	(None, 240)	
block4a_se_reshape (Reshape)	(None, 1, 1, 240)	
block4a_se_reduce (Conv2D)	(None, 1, 1, 10)	
block4a_se_expand (Conv2D)	(None, 1, 1, 240)	
block4a_se_excite (Multiply)	(None, 14, 14, 240)	
block4a_project_conv (Conv2D)	(None, 14, 14, 80)	1
block4a_project_bn (BatchNormalization)	(None, 14, 14, 80)	

block4b_expand_conv (Conv2D)	(None, 14, 14, 480)	3
block4b_expand_bn (BatchNormalization)	(None, 14, 14, 480)	
block4b_expand_activation (Activation)	(None, 14, 14, 480)	
block4b_dwconv (DepthwiseConv2D)	(None, 14, 14, 480)	
block4b_bn (BatchNormalization)	(None, 14, 14, 480)	
block4b_activation (Activation)	(None, 14, 14, 480)	
block4b_se_squeeze (GlobalAveragePooling2D)	(None, 480)	
block4b_se_reshape (Reshape)	(None, 1, 1, 480)	
block4b_se_reduce (Conv2D)	(None, 1, 1, 20)	
block4b_se_expand (Conv2D)	(None, 1, 1, 480)	1
block4b_se_excite (Multiply)	(None, 14, 14, 480)	
block4b_project_conv (Conv2D)	(None, 14, 14, 80)	3
block4b_project_bn (BatchNormalization)	(None, 14, 14, 80)	
block4b_drop (Dropout)	(None, 14, 14, 80)	
block4b_add (Add)	(None, 14, 14, 80)	
block4c_expand_conv (Conv2D)	(None, 14, 14, 480)	3
block4c_expand_bn (BatchNormalization)	(None, 14, 14, 480)	
block4c_expand_activation (Activation)	(None, 14, 14, 480)	
block4c_dwconv (DepthwiseConv2D)	(None, 14, 14, 480)	
block4c_bn (BatchNormalization)	(None, 14, 14, 480)	
block4c_activation (Activation)	(None, 14, 14, 480)	
block4c_se_squeeze (GlobalAveragePooling2D)	(None, 480)	

block4c_se_reshape (Reshape)	(None, 1, 1, 480)	
block4c_se_reduce (Conv2D)	(None, 1, 1, 20)	
block4c_se_expand (Conv2D)	(None, 1, 1, 480)	1
block4c_se_excite (Multiply)	(None, 14, 14, 480)	
block4c_project_conv (Conv2D)	(None, 14, 14, 80)	3
block4c_project_bn (BatchNormalization)	(None, 14, 14, 80)	
block4c_drop (Dropout)	(None, 14, 14, 80)	
block4c_add (Add)	(None, 14, 14, 80)	
block5a_expand_conv (Conv2D)	(None, 14, 14, 480)	3
block5a_expand_bn (BatchNormalization)	(None, 14, 14, 480)	
block5a_expand_activation (Activation)	(None, 14, 14, 480)	
block5a_dwconv (DepthwiseConv2D)	(None, 14, 14, 480)	1
block5a_bn (BatchNormalization)	(None, 14, 14, 480)	
block5a_activation (Activation)	(None, 14, 14, 480)	
block5a_se_squeeze (GlobalAveragePooling2D)	(None, 480)	
block5a_se_reshape (Reshape)	(None, 1, 1, 480)	
block5a_se_reduce (Conv2D)	(None, 1, 1, 20)	
block5a_se_expand (Conv2D)	(None, 1, 1, 480)	1
block5a_se_excite (Multiply)	(None, 14, 14, 480)	
block5a_project_conv (Conv2D)	(None, 14, 14, 112)	5
block5a_project_bn (BatchNormalization)	(None, 14, 14, 112)	
block5b_expand_conv (Conv2D)	(None, 14, 14, 672)	7

block5b_expand_bn (BatchNormalization)	(None, 14, 14, 672)	
block5b_expand_activation (Activation)	(None, 14, 14, 672)	
block5b_dwconv (DepthwiseConv2D)	(None, 14, 14, 672)	1
block5b_bn (BatchNormalization)	(None, 14, 14, 672)	
block5b_activation (Activation)	(None, 14, 14, 672)	
block5b_se_squeeze (GlobalAveragePooling2D)	(None, 672)	
block5b_se_reshape (Reshape)	(None, 1, 1, 672)	
block5b_se_reduce (Conv2D)	(None, 1, 1, 28)	1
block5b_se_expand (Conv2D)	(None, 1, 1, 672)	1
block5b_se_excite (Multiply)	(None, 14, 14, 672)	
block5b_project_conv (Conv2D)	(None, 14, 14, 112)	7
block5b_project_bn (BatchNormalization)	(None, 14, 14, 112)	
block5b_drop (Dropout)	(None, 14, 14, 112)	
block5b_add (Add)	(None, 14, 14, 112)	
block5c_expand_conv (Conv2D)	(None, 14, 14, 672)	7
block5c_expand_bn (BatchNormalization)	(None, 14, 14, 672)	
block5c_expand_activation (Activation)	(None, 14, 14, 672)	
block5c_dwconv (DepthwiseConv2D)	(None, 14, 14, 672)	1
block5c_bn (BatchNormalization)	(None, 14, 14, 672)	
block5c_activation (Activation)	(None, 14, 14, 672)	
block5c_se_squeeze (GlobalAveragePooling2D)	(None, 672)	
block5c_se_reshape (Reshape)	(None, 1, 1, 672)	
block5c_se_reduce	(None, 1, 1, 28)	1

(Conv2D)		
block5c_se_expand (Conv2D)	(None, 1, 1, 672)	1
block5c_se_excite (Multiply)	(None, 14, 14, 672)	
block5c_project_conv (Conv2D)	(None, 14, 14, 112)	7
block5c_project_bn (BatchNormalization)	(None, 14, 14, 112)	
block5c_drop (Dropout)	(None, 14, 14, 112)	
block5c_add (Add)	(None, 14, 14, 112)	
block6a_expand_conv (Conv2D)	(None, 14, 14, 672)	7
block6a_expand_bn (BatchNormalization)	(None, 14, 14, 672)	
block6a_expand_activation (Activation)	(None, 14, 14, 672)	
block6a_dwconv_pad (ZeroPadding2D)	(None, 17, 17, 672)	
block6a_dwconv (DepthwiseConv2D)	(None, 7, 7, 672)	1
block6a_bn (BatchNormalization)	(None, 7, 7, 672)	
block6a_activation (Activation)	(None, 7, 7, 672)	
block6a_se_squeeze (GlobalAveragePooling2D)	(None, 672)	
block6a_se_reshape (Reshape)	(None, 1, 1, 672)	
block6a_se_reduce (Conv2D)	(None, 1, 1, 28)	1
block6a_se_expand (Conv2D)	(None, 1, 1, 672)	1
block6a_se_excite (Multiply)	(None, 7, 7, 672)	
block6a_project_conv (Conv2D)	(None, 7, 7, 192)	12
block6a_project_bn (BatchNormalization)	(None, 7, 7, 192)	
block6b_expand_conv (Conv2D)	(None, 7, 7, 1152)	22
block6b_expand_bn	(None, 7, 7, 1152)	

(BatchNormalization)		
block6b_expand_activation (Activation)	(None, 7, 7, 1152)	
block6b_dwconv (DepthwiseConv2D)	(None, 7, 7, 1152)	2
block6b_bn (BatchNormalization)	(None, 7, 7, 1152)	
block6b_activation (Activation)	(None, 7, 7, 1152)	
block6b_se_squeeze (GlobalAveragePooling2D)	(None, 1152)	
block6b_se_reshape (Reshape)	(None, 1, 1, 1152)	
block6b_se_reduce (Conv2D)	(None, 1, 1, 48)	5
block6b_se_expand (Conv2D)	(None, 1, 1, 1152)	5
block6b_se_excite (Multiply)	(None, 7, 7, 1152)	
block6b_project_conv (Conv2D)	(None, 7, 7, 192)	22
block6b_project_bn (BatchNormalization)	(None, 7, 7, 192)	
block6b_drop (Dropout)	(None, 7, 7, 192)	
block6b_add (Add)	(None, 7, 7, 192)	
block6c_expand_conv (Conv2D)	(None, 7, 7, 1152)	22
block6c_expand_bn (BatchNormalization)	(None, 7, 7, 1152)	
block6c_expand_activation (Activation)	(None, 7, 7, 1152)	
block6c_dwconv (DepthwiseConv2D)	(None, 7, 7, 1152)	2
block6c_bn (BatchNormalization)	(None, 7, 7, 1152)	
block6c_activation (Activation)	(None, 7, 7, 1152)	
block6c_se_squeeze (GlobalAveragePooling2D)	(None, 1152)	
block6c_se_reshape (Reshape)	(None, 1, 1, 1152)	
block6c_se_reduce (Conv2D)	(None, 1, 1, 48)	5

block6c_se_expand (Conv2D)	(None, 1, 1, 1152)	5
block6c_se_excite (Multiply)	(None, 7, 7, 1152)	
block6c_project_conv (Conv2D)	(None, 7, 7, 192)	22
block6c_project_bn (BatchNormalization)	(None, 7, 7, 192)	
block6c_drop (Dropout)	(None, 7, 7, 192)	
block6c_add (Add)	(None, 7, 7, 192)	
block6d_expand_conv (Conv2D)	(None, 7, 7, 1152)	22
block6d_expand_bn (BatchNormalization)	(None, 7, 7, 1152)	
block6d_expand_activation (Activation)	(None, 7, 7, 1152)	
block6d_dwconv (DepthwiseConv2D)	(None, 7, 7, 1152)	2
block6d_bn (BatchNormalization)	(None, 7, 7, 1152)	
block6d_activation (Activation)	(None, 7, 7, 1152)	
block6d_se_squeeze (GlobalAveragePooling2D)	(None, 1152)	
block6d_se_reshape (Reshape)	(None, 1, 1, 1152)	
block6d_se_reduce (Conv2D)	(None, 1, 1, 48)	5
block6d_se_expand (Conv2D)	(None, 1, 1, 1152)	5
block6d_se_excite (Multiply)	(None, 7, 7, 1152)	
block6d_project_conv (Conv2D)	(None, 7, 7, 192)	22
block6d_project_bn (BatchNormalization)	(None, 7, 7, 192)	
block6d_drop (Dropout)	(None, 7, 7, 192)	
block6d_add (Add)	(None, 7, 7, 192)	
block7a_expand_conv (Conv2D)	(None, 7, 7, 1152)	22

block7a_expand_bn (BatchNormalization)	(None, 7, 7, 1152)	
block7a_expand_activation (Activation)	(None, 7, 7, 1152)	
block7a_dwconv (DepthwiseConv2D)	(None, 7, 7, 1152)	1
block7a_bn (BatchNormalization)	(None, 7, 7, 1152)	
block7a_activation (Activation)	(None, 7, 7, 1152)	
block7a_se_squeeze (GlobalAveragePooling2D)	(None, 1152)	
block7a_se_reshape (Reshape)	(None, 1, 1, 1152)	
block7a_se_reduce (Conv2D)	(None, 1, 1, 48)	5
block7a_se_expand (Conv2D)	(None, 1, 1, 1152)	5
block7a_se_excite (Multiply)	(None, 7, 7, 1152)	
block7a_project_conv (Conv2D)	(None, 7, 7, 320)	36
block7a_project_bn (BatchNormalization)	(None, 7, 7, 320)	
top_conv (Conv2D)	(None, 7, 7, 1280)	40
top_bn (BatchNormalization)	(None, 7, 7, 1280)	
top_activation (Activation)	(None, 7, 7, 1280)	
avg_pool (GlobalAveragePooling2D)	(None, 1280)	
top_dropout (Dropout)	(None, 1280)	
predictions (Dense)	(None, 120)	15

Total params: 4,203,291 (16.03 MB)

Trainable params: 4,161,268 (15.87 MB)

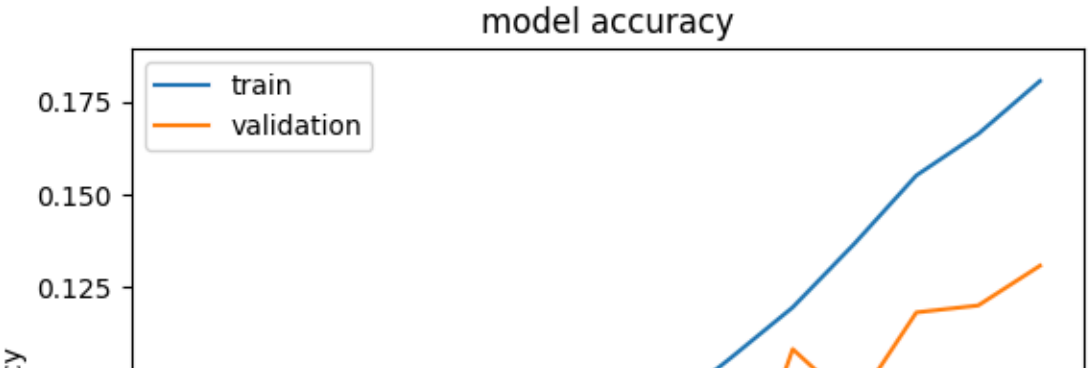
Non-trainable params: 42,023 (164.16 KB)

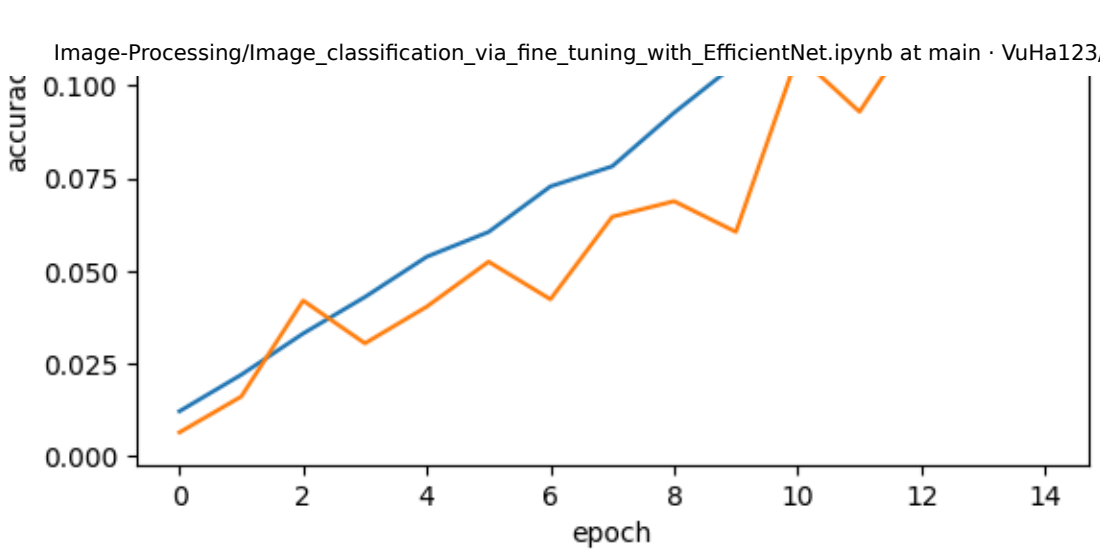
```
Epoch 1/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 172s 569ms/step - accuracy: 0.0100 - los
s: 5.0865 - val_accuracy: 0.0064 - val_loss: 4.8804
Epoch 2/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 102s 543ms/step - accuracy: 0.0198 - los
s: 4.6292 - val_accuracy: 0.0161 - val_loss: 4.9402
Epoch 3/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 102s 543ms/step - accuracy: 0.0325 - los
s: 4.4517 - val accuracy: 0.0420 - val loss: 4.4230
```

```
Epoch 4/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 142s 542ms/step - accuracy: 0.0419 - los
s: 4.3171 - val_accuracy: 0.0304 - val_loss: 5.0991
Epoch 5/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 102s 544ms/step - accuracy: 0.0508 - los
s: 4.2145 - val_accuracy: 0.0403 - val_loss: 4.8225
Epoch 6/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 102s 545ms/step - accuracy: 0.0587 - los
s: 4.1271 - val_accuracy: 0.0525 - val_loss: 4.4168
Epoch 7/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 101s 542ms/step - accuracy: 0.0718 - los
s: 4.0392 - val_accuracy: 0.0423 - val_loss: 7.4204
Epoch 8/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 101s 541ms/step - accuracy: 0.0775 - los
s: 3.9842 - val_accuracy: 0.0646 - val_loss: 4.2874
Epoch 9/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 101s 540ms/step - accuracy: 0.0927 - los
s: 3.9022 - val_accuracy: 0.0688 - val_loss: 4.4391
Epoch 10/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 102s 544ms/step - accuracy: 0.1025 - los
s: 3.7931 - val_accuracy: 0.0605 - val_loss: 4.3240
Epoch 11/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 102s 545ms/step - accuracy: 0.1160 - los
s: 3.7005 - val_accuracy: 0.1082 - val_loss: 3.9038
Epoch 12/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 102s 544ms/step - accuracy: 0.1348 - los
s: 3.6048 - val_accuracy: 0.0929 - val_loss: 4.0729
Epoch 13/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 102s 545ms/step - accuracy: 0.1531 - los
s: 3.4991 - val_accuracy: 0.1181 - val_loss: 3.7514
Epoch 14/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 101s 542ms/step - accuracy: 0.1679 - los
s: 3.4164 - val_accuracy: 0.1200 - val_loss: 3.9157
Epoch 15/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 101s 542ms/step - accuracy: 0.1778 - los
s: 3.3308 - val_accuracy: 0.1307 - val_loss: 3.6917
```

In [9]:

```python
import matplotlib.pyplot as plt


def plot_hist(hist):
    plt.plot(hist.history["accuracy"])
    plt.plot(hist.history["val_accuracy"])
    plt.title("model accuracy")
    plt.ylabel("accuracy")
    plt.xlabel("epoch")
    plt.legend(["train", "validation"], loc="upper left")
    plt.show()


plot_hist(hist)
```





Nhận xét

- Ban đầu, accuracy trên trainset và valset gần bằng nhau, đây là giai đoạn model được học các feature từ tập dữ liệu.
- Sau một thời gian, biểu đồ của accuracy cho thấy accuracy tăng nhanh và khá smooth. Cho thấy trong quá trình training model đang học khá tốt từ tập dữ liệu. Tuy nhiên, có khả năng model đang bị overfiting. Trong khi đó, tập valset lại cho thấy sự tăng chậm của accuracy và gấp khúc, điều đó cho thấy trong tập valset có nhiều dữ liệu mới, và model đang cố gắng tổng quát hóa nó.

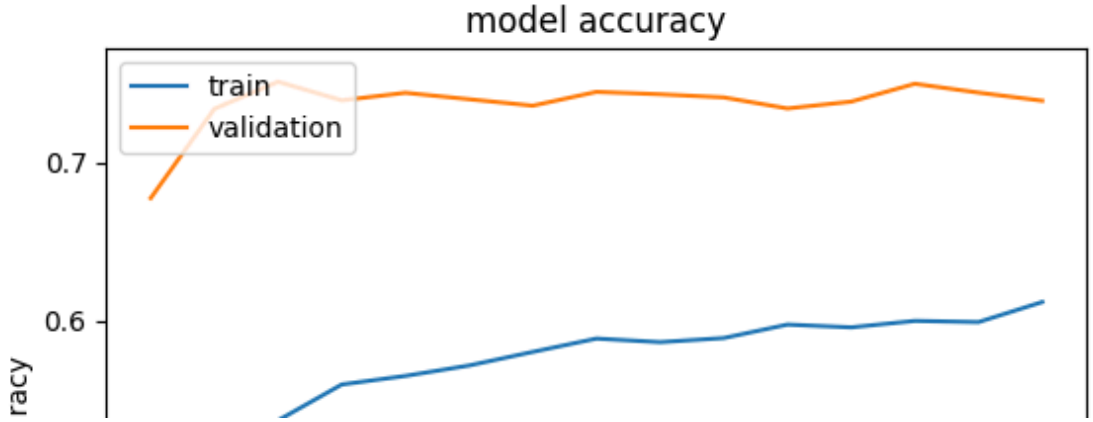
Transfer learning from pre-trained weights

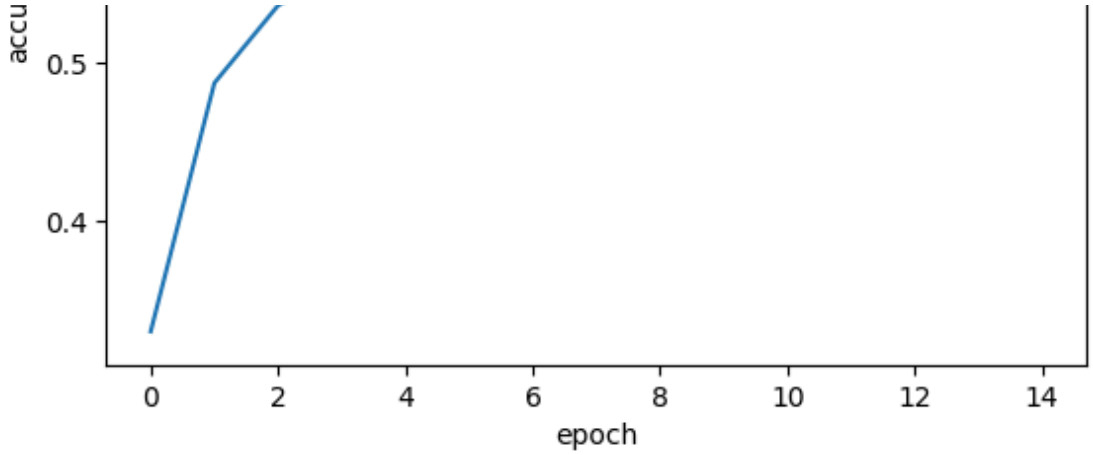
In [13]:
```
def build_model(num_classes):
  #Định nghĩa đầu vào
    inputs = layers.Input(shape=(IMG_SIZE, IMG_SIZE, 3))
  #Tải các lớp Efficient B0 mà không bao gồm các lớp đầu ra cuối
    model = EfficientNetB0(include_top=False, input_tensor=inputs, we

    # Đóng băng các trọng số được huấn luyện trước đó.
    model.trainable = False

    # Xây dựng lại phần đầu ra
    x = layers.GlobalAveragePooling2D(name="avg_pool")(model.output)
    x = layers.BatchNormalization()(x)

    top_dropout_rate = 0.2
    x = layers.Dropout(top_dropout_rate, name="top_dropout")(x)
    outputs = layers.Dense(num_classes, activation="softmax", name="p

    # Compile
    model = keras.Model(inputs, outputs, name="EfficientNet")
    optimizer = keras.optimizers.Adam(learning_rate=1e-2)
    model.compile(
        optimizer=optimizer, loss="categorical_crossentropy", metrics
    )
    return model
```

In [14]:
```
model = build_model(num_classes=NUM_CLASSES)

epochs = 15  # @param {type: "slider", min:8, max:80}
```

```
hist = model.fit(ds_train, epochs=epochs, validation_data=ds_test)
plot_hist(hist)
```

```
Downloading data from https://storage.googleapis.com/keras-application
s/efficientnetb0_notop.h5
16705208/16705208 ━━━━━━━━━━━━━━━━━━━━ 1s 0us/step
Epoch 1/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 98s 434ms/step - accuracy: 0.2619 - loss:
4.3843 - val_accuracy: 0.6775 - val_loss: 1.1235
Epoch 2/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 75s 398ms/step - accuracy: 0.4910 - loss:
2.0585 - val_accuracy: 0.7340 - val_loss: 0.8764
Epoch 3/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 74s 394ms/step - accuracy: 0.5394 - loss:
1.6848 - val_accuracy: 0.7512 - val_loss: 0.8244
Epoch 4/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 74s 395ms/step - accuracy: 0.5621 - loss:
1.5977 - val_accuracy: 0.7394 - val_loss: 0.8523
Epoch 5/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 75s 402ms/step - accuracy: 0.5649 - loss:
1.5429 - val_accuracy: 0.7442 - val_loss: 0.8776
Epoch 6/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 75s 400ms/step - accuracy: 0.5726 - loss:
1.5567 - val_accuracy: 0.7401 - val_loss: 0.8751
Epoch 7/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 75s 400ms/step - accuracy: 0.5902 - loss:
1.4652 - val_accuracy: 0.7360 - val_loss: 0.8924
Epoch 8/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 77s 409ms/step - accuracy: 0.5944 - loss:
1.4804 - val_accuracy: 0.7448 - val_loss: 0.8777
Epoch 9/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 75s 399ms/step - accuracy: 0.5938 - loss:
1.4483 - val_accuracy: 0.7432 - val_loss: 0.8939
Epoch 10/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 82s 400ms/step - accuracy: 0.5825 - loss:
1.4864 - val_accuracy: 0.7413 - val_loss: 0.8953
Epoch 11/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 75s 403ms/step - accuracy: 0.5991 - loss:
1.4360 - val_accuracy: 0.7343 - val_loss: 0.9281
Epoch 12/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 76s 406ms/step - accuracy: 0.5967 - loss:
1.4291 - val_accuracy: 0.7386 - val_loss: 0.9042
Epoch 13/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 77s 412ms/step - accuracy: 0.5984 - loss:
1.4416 - val_accuracy: 0.7499 - val_loss: 0.8880
Epoch 14/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 77s 411ms/step - accuracy: 0.6014 - loss:
1.4492 - val_accuracy: 0.7443 - val_loss: 0.8972
Epoch 15/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 77s 409ms/step - accuracy: 0.6113 - loss:
1.3971 - val_accuracy: 0.7392 - val_loss: 0.9298
```

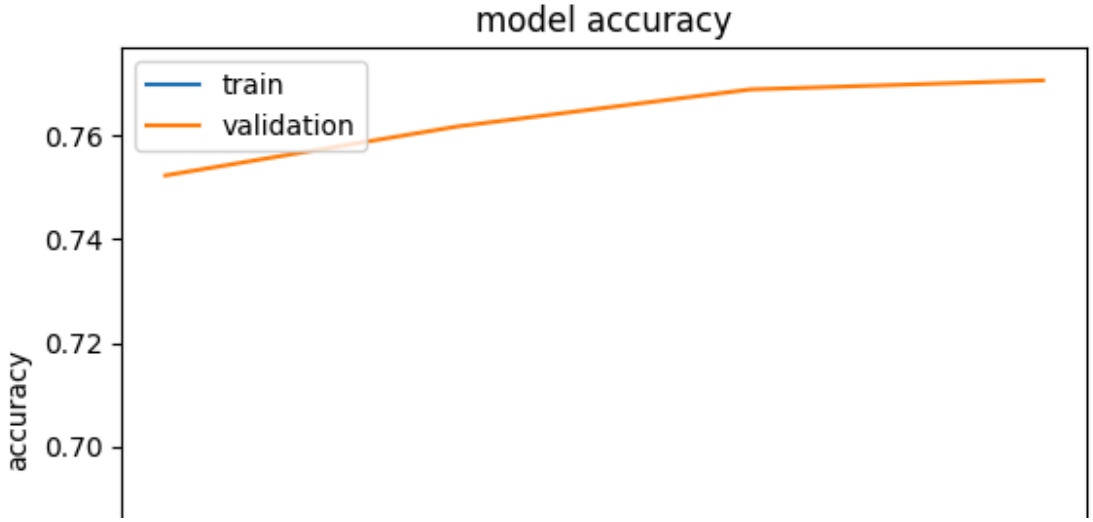




In [15]:

```
#Mở khóa các lớp của model để cho phép người dùng được trained, ng
def unfreeze_model(model):
    # Chúng tôi giải phóng 20 lớp trên cùng trong khi vẫn giữ nguyê
    for layer in model.layers[-20:]:
        if not isinstance(layer, layers.BatchNormalization):#Kiểm tr
            layer.trainable = True

    optimizer = keras.optimizers.Adam(learning_rate=1e-5)
    model.compile(
        optimizer=optimizer, loss="categorical_crossentropy", metrics
    )


unfreeze_model(model)

epochs = 4  # @param {type: "slider", min:4, max:10}
hist = model.fit(ds_train, epochs=epochs, validation_data=ds_test)
plot_hist(hist)
```

```
Epoch 1/4
187/187 ━━━━━━━━━━━━━━━━━━━━ 100s 429ms/step - accuracy: 0.6153 - los
s: 1.3522 - val_accuracy: 0.7522 - val_loss: 0.8735
Epoch 2/4
187/187 ━━━━━━━━━━━━━━━━━━━━ 78s 415ms/step - accuracy: 0.6331 - loss:
1.2929 - val_accuracy: 0.7617 - val_loss: 0.8417
Epoch 3/4
187/187 ━━━━━━━━━━━━━━━━━━━━ 82s 413ms/step - accuracy: 0.6506 - loss:
1.2387 - val_accuracy: 0.7688 - val_loss: 0.8209
Epoch 4/4
187/187 ━━━━━━━━━━━━━━━━━━━━ 77s 413ms/step - accuracy: 0.6580 - loss:
1.2080 - val_accuracy: 0.7705 - val_loss: 0.8043
```




0.68