```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator

IMG_SIZE = 244
BATCH_SIZE = 32

from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define IMG_SIZE and BATCH_SIZE
IMG_SIZE = 224
BATCH_SIZE = 32
train_datagen = ImageDataGenerator(rescale=1./255,
validation_split=0.2)


train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/ML_TEAM6/1SV21CS021/images',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)

val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/ML_TEAM6/1SV21CS021/images',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

```
Found 273 images belonging to 5 classes.
Found 66 images belonging to 5 classes.
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

```
# Define the model
model = keras.Sequential([
    layers.Conv2D(32, (3,3), activation='relu', input_shape=(IMG_SIZE,
IMG_SIZE, 3)),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(64, (3,3), activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(128, (3,3), activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(1,activation='softmax')  # Output layer for multi-
class classification
])


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

model.fit(train_generator, validation_data=val_generator, epochs=5)
```

```
Epoch 1/5
9/9 [==============================] - 42s 4s/step - loss: 1.1406 -
accuracy: 0.2000 - val_loss: 0.5438 - val_accuracy: 0.2000
Epoch 2/5
9/9 [==============================] - 37s 4s/step - loss: 0.5183 -
accuracy: 0.2000 - val_loss: 0.5129 - val_accuracy: 0.2000
Epoch 3/5
9/9 [==============================] - 40s 4s/step - loss: 0.5060 -
accuracy: 0.2000 - val_loss: 0.5064 - val_accuracy: 0.2000
Epoch 4/5
9/9 [==============================] - 37s 4s/step - loss: 0.5028 -
accuracy: 0.2000 - val_loss: 0.5028 - val_accuracy: 0.2000
Epoch 5/5
9/9 [==============================] - 41s 5s/step - loss: 0.5014 -
accuracy: 0.2000 - val_loss: 0.5015 - val_accuracy: 0.2000

<keras.src.callbacks.History at 0x7ea418307bb0>
```

```
model.save("Model.h5","label.txt")

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

model = load_model('Model.h5')
test_image_path =
'/content/drive/MyDrive/ML_TEAM6/1SV21CS021/images/Blue
budgies/Blue_budgie (1).jpeg'
img = image.load_img(test_image_path, target_size=(224, 224))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)

img_array = img_array / 255.0

predictions = model.predict(img_array)
print(predictions)
```

```
1/1 [==============================] - 0s 251ms/step
[[1.]]
```

```python
if predictions < 0.5:
    print('It is a Blue budgies')
else:
    print('It is a Yellow budgies')
```

```
It is a Yellow budgies
```