```
!unzip dataset_12.zip
```

```
Archive:  dataset_12.zip
  inflating: dataset_1/keyboard/20210101_19_10_17_000_MHvmWMWftvMzzFmedX6ol5Q5I5t2_T_4000_3000.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210101_19_10_17_000_MHvmWMWftvMzzFmedX6ol5Q5I5t2_T_4000_3000.jpg.jpg
  inflating: dataset_1/keyboard/20210112_10_05_58_000_bmXQbhr8pbh11y4b2Ny5MyRTcck2_F_3264_2448.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210112_10_05_58_000_bmXQbhr8pbh11y4b2Ny5MyRTcck2_F_3264_2448.jpg.jpg
  inflating: dataset_1/keyboard/20210101_12_03_47_000_ZTr8uoDUDPNPmJQIpuiUM01PlU43_T_3120_4160.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210101_12_03_47_000_ZTr8uoDUDPNPmJQIpuiUM01PlU43_T_3120_4160.jpg.jpg
  inflating: dataset_1/keyboard/photo-1484807352052-23338990c6c6.avif
  inflating: __MACOSX/dataset_1/keyboard/._photo-1484807352052-23338990c6c6.avif
  inflating: dataset_1/keyboard/20210102_11_42_48_000_ZTr8uoDUDPNPmJQIpuiUM01PlU43_T_3120_4160.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210102_11_42_48_000_ZTr8uoDUDPNPmJQIpuiUM01PlU43_T_3120_4160.jpg.jpg
  inflating: dataset_1/keyboard/20210101_13_39_33_000_MHvmWMWftvMzzFmedX6ol5Q5I5t2_T_3000_4000.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210101_13_39_33_000_MHvmWMWftvMzzFmedX6ol5Q5I5t2_T_3000_4000.jpg.jpg
  inflating: dataset_1/keyboard/images (27).jpeg
  inflating: __MACOSX/dataset_1/keyboard/._images (27).jpeg
  inflating: dataset_1/keyboard/71-FQvvZiUL._AC_SL1500_.jpg
  inflating: __MACOSX/dataset_1/keyboard/._71-FQvvZiUL._AC_SL1500_.jpg
  inflating: dataset_1/keyboard/20210101_12_07_33_000_PRE8DAtpJ1Q5yPNHZdoqaOWxHNA2_T_4160_3120.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210101_12_07_33_000_PRE8DAtpJ1Q5yPNHZdoqaOWxHNA2_T_4160_3120.jpg.jpg
  inflating: dataset_1/keyboard/premium_photo-1675842663249-a8b70103dbaa.avif
  inflating: __MACOSX/dataset_1/keyboard/._premium_photo-1675842663249-a8b70103dbaa.avif
  inflating: dataset_1/keyboard/ctrlv2.webp
  inflating: __MACOSX/dataset_1/keyboard/._ctrlv2.webp
  inflating: dataset_1/keyboard/202112364_01_43_05_000_aM018pmdRTQQkeL1PokkOIJjDfH3_T_2988_5312.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._202112364_01_43_05_000_aM018pmdRTQQkeL1PokkOIJjDfH3_T_2988_5312.jpg.jpg
  inflating: dataset_1/keyboard/202112365_19_11_55_000_ED8APwkrunTluVxK3VsSuEPMXAr2_F_4160_3120.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._202112365_19_11_55_000_ED8APwkrunTluVxK3VsSuEPMXAr2_F_4160_3120.jpg.jpg
  inflating: dataset_1/keyboard/images.png
  inflating: __MACOSX/dataset_1/keyboard/._images.png
  inflating: dataset_1/keyboard/250px-KBC_Poker_II_--_backlighting_detail.jpg
  inflating: __MACOSX/dataset_1/keyboard/._250px-KBC_Poker_II_--_backlighting_detail.jpg
  inflating: dataset_1/keyboard/20210108_20_49_09_000_lLntf88xwZV7pgnKz1uJVyqXTx73_F_3000_4000.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210108_20_49_09_000_lLntf88xwZV7pgnKz1uJVyqXTx73_F_3000_4000.jpg.jpg
  inflating: dataset_1/keyboard/images (18).jpeg
  inflating: __MACOSX/dataset_1/keyboard/._images (18).jpeg
  inflating: dataset_1/keyboard/71rorY62PbL._AC_UF894,1000_QL80_.jpg
  inflating: __MACOSX/dataset_1/keyboard/._71rorY62PbL._AC_UF894,1000_QL80_.jpg
  inflating: dataset_1/keyboard/images (25).jpeg
  inflating: __MACOSX/dataset_1/keyboard/._images (25).jpeg
  inflating: dataset_1/keyboard/20201231_14_58_03_000_E6PxFNY1Vsb8ebbISmYzvCKKshS2_F_3264_2448.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20201231_14_58_03_000_E6PxFNY1Vsb8ebbISmYzvCKKshS2_F_3264_2448.jpg.jpg
  inflating: dataset_1/keyboard/premium_photo-1664699099313-77683fc43355.avif
  inflating: __MACOSX/dataset_1/keyboard/._premium_photo-1664699099313-77683fc43355.avif
  inflating: dataset_1/keyboard/20210102_11_54_47_000_PRE8DAtpJ1Q5yPNHZdoqaOWxHNA2_F_3264_2448.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210102_11_54_47_000_PRE8DAtpJ1Q5yPNHZdoqaOWxHNA2_F_3264_2448.jpg.jpg
  inflating: dataset_1/keyboard/20210104_14_27_04_000_PRE8DAtpJ1Q5yPNHZdoqaOWxHNA2_T_4160_3120.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210104_14_27_04_000_PRE8DAtpJ1Q5yPNHZdoqaOWxHNA2_T_4160_3120.jpg.jpg
  inflating: dataset_1/keyboard/a0dea343622ba68c6c39fba3cf30c3d1.jpg
  inflating: __MACOSX/dataset_1/keyboard/._a0dea343622ba68c6c39fba3cf30c3d1.jpg
  inflating: dataset_1/keyboard/20210102_12_12_19_000_kZtbG1otWiPSOj3R1bPYdq2J3Bz1_T_1968_4144.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20210102_12_12_19_000_kZtbG1otWiPSOj3R1bPYdq2J3Bz1_T_1968_4144.jpg.jpg
  inflating: dataset_1/keyboard/1_qYGBCM3HT4Kjxsm4UGTN1A.png
  inflating: __MACOSX/dataset_1/keyboard/._1_qYGBCM3HT4Kjxsm4UGTN1A.png
  inflating: dataset_1/keyboard/photo-1561908818-526e64312efd.avif
  inflating: __MACOSX/dataset_1/keyboard/._photo-1561908818-526e64312efd.avif
  inflating: dataset_1/keyboard/premium_photo-1677870728087-2257ce93bfe9.avif
  inflating: __MACOSX/dataset_1/keyboard/._premium_photo-1677870728087-2257ce93bfe9.avif
  inflating: dataset_1/keyboard/20201230_15_22_20_000_M9gVSRUsnJYxozuIsk4ReQjamuJ2_T_4160_3120.jpg.jpg
  inflating: __MACOSX/dataset_1/keyboard/._20201230_15_22_20_000_M9gVSRUsnJYxozuIsk4ReQjamuJ2_T_4160_3120.jpg.jpg
```

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
import matplotlib.pyplot as plt
```

```
dataset_path = 'dataset_1/'
for class_name in os.listdir(dataset_path):
    class_path = os.path.join(dataset_path, class_name)
    if os.path.isdir(class_path):
        num_images = len(os.listdir(class_path))
        print(f"Clase '{class_name}' tiene {num_images} imágenes.")
```

```
Clase 'monitor' tiene 215 imágenes.
Clase 'keyboard' tiene 216 imágenes.
Clase 'mouse' tiene 216 imágenes.
```

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest',
    validation_split=0.2  #20% para validacion
)
```

```
train_generator = train_datagen.flow_from_directory(
    dataset_path,
    target_size=(224, 224),
    batch_size=64,
    class_mode='categorical',
    subset='training'
)

validation_generator = train_datagen.flow_from_directory(
    dataset_path,
    target_size=(224, 224),
    batch_size=64,
    class_mode='categorical',
    subset='validation'
)
```

```
Found 341 images belonging to 3 classes.
Found 83 images belonging to 3 classes.
```

```
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
```

```
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.01))(x)
x = Dropout(0.5)(x)
predictions = Dense(3, activation='softmax')(x)
```

```
model = Model(inputs=base_model.input, outputs=predictions)
```

```
for layer in base_model.layers:
    layer.trainable = False
```

```
model.compile(optimizer=Adam(learning_rate=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
lr_scheduler = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=3)
checkpoint = ModelCheckpoint('best_model.keras', monitor='val_accuracy', save_best_only=True, mode='max')
```

```
history = model.fit(
    train_generator,
    epochs=20,
    validation_data=validation_generator,
    callbacks=[early_stopping, lr_scheduler, checkpoint]
)
```

```
Epoch 1/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 107s 16s/step - accuracy: 0.3274 - loss: 14.3437 - val_accuracy: 0.3855 - val_loss: 13.7073 - lea
Epoch 2/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 95s 14s/step - accuracy: 0.3819 - loss: 13.7693 - val_accuracy: 0.3614 - val_loss: 13.2291 - lea
Epoch 3/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 150s 17s/step - accuracy: 0.3814 - loss: 13.2266 - val_accuracy: 0.6265 - val_loss: 12.7076 - lea
Epoch 4/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 96s 14s/step - accuracy: 0.4239 - loss: 12.7543 - val_accuracy: 0.4699 - val_loss: 12.2700 - lea
Epoch 5/20
```

```
6/6 ━━━━━━━━━━━━━━━━━━━━ 95s 14s/step - accuracy: 0.4606 - loss: 12.2475 - val_accuracy: 0.5181 - val_loss: 11.8081 - lea
Epoch 6/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 142s 14s/step - accuracy: 0.4387 - loss: 11.8581 - val_accuracy: 0.5663 - val_loss: 11.3882 - le
Epoch 7/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 116s 18s/step - accuracy: 0.5087 - loss: 11.4488 - val_accuracy: 0.6506 - val_loss: 10.9674 - le
Epoch 8/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 95s 14s/step - accuracy: 0.4720 - loss: 10.9787 - val_accuracy: 0.5904 - val_loss: 10.5806 - lea
Epoch 9/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 142s 14s/step - accuracy: 0.4104 - loss: 10.6901 - val_accuracy: 0.5783 - val_loss: 10.1987 - le
Epoch 10/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 95s 14s/step - accuracy: 0.4398 - loss: 10.2728 - val_accuracy: 0.5783 - val_loss: 9.8131 - learn
Epoch 11/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 142s 16s/step - accuracy: 0.4548 - loss: 9.8619 - val_accuracy: 0.5060 - val_loss: 9.5021 - learn
Epoch 12/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 98s 15s/step - accuracy: 0.4300 - loss: 9.5629 - val_accuracy: 0.6627 - val_loss: 9.1288 - learn
Epoch 13/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 95s 16s/step - accuracy: 0.4614 - loss: 9.2274 - val_accuracy: 0.6024 - val_loss: 8.8175 - learn
Epoch 14/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 96s 14s/step - accuracy: 0.4689 - loss: 8.9202 - val_accuracy: 0.6747 - val_loss: 8.5207 - learn
Epoch 15/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 116s 18s/step - accuracy: 0.4801 - loss: 8.5672 - val_accuracy: 0.6627 - val_loss: 8.2092 - learn
Epoch 16/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 122s 14s/step - accuracy: 0.5601 - loss: 8.2700 - val_accuracy: 0.6386 - val_loss: 7.9292 - learn
Epoch 17/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 116s 18s/step - accuracy: 0.5013 - loss: 7.9935 - val_accuracy: 0.5904 - val_loss: 7.6569 - learn
Epoch 18/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 122s 14s/step - accuracy: 0.5731 - loss: 7.6849 - val_accuracy: 0.6747 - val_loss: 7.4017 - learn
Epoch 19/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 105s 16s/step - accuracy: 0.5371 - loss: 7.4451 - val_accuracy: 0.5663 - val_loss: 7.1385 - learn
Epoch 20/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 117s 18s/step - accuracy: 0.5725 - loss: 7.2139 - val_accuracy: 0.5783 - val_loss: 6.9312 - learn
```

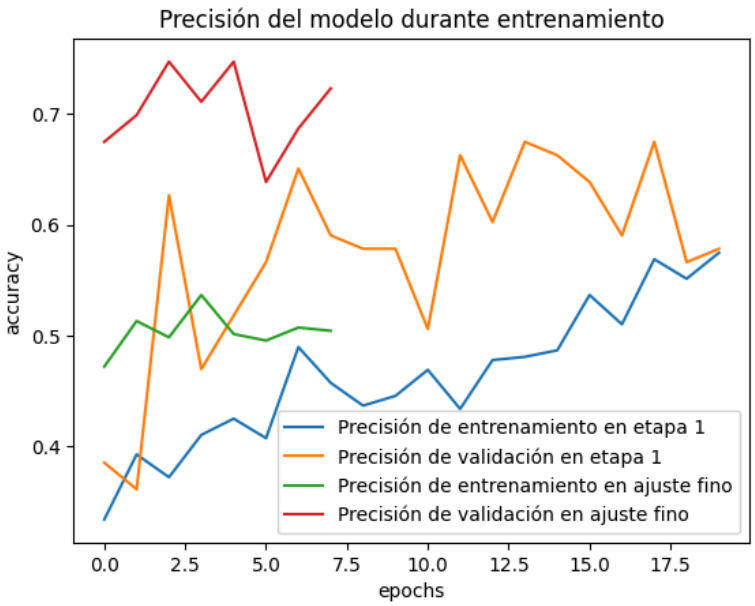
```
for layer in base_model.layers[-20:]:
    layer.trainable = True
```

```
model.compile(optimizer=Adam(learning_rate=1e-6), loss='categorical_crossentropy', metrics=['accuracy'])
```

```
history_finetune = model.fit(
    train_generator,
    epochs=10,
    validation_data=validation_generator,
    callbacks=[early_stopping, lr_scheduler, checkpoint]
)
```

```
Epoch 1/10
6/6 ━━━━━━━━━━━━━━━━━━━━ 145s 19s/step - accuracy: 0.4568 - loss: 7.1084 - val_accuracy: 0.6747 - val_loss: 6.8802 - learn
Epoch 2/10
6/6 ━━━━━━━━━━━━━━━━━━━━ 118s 20s/step - accuracy: 0.5445 - loss: 7.0346 - val_accuracy: 0.6988 - val_loss: 6.8919 - learn
Epoch 3/10
6/6 ━━━━━━━━━━━━━━━━━━━━ 141s 18s/step - accuracy: 0.4451 - loss: 7.1383 - val_accuracy: 0.7470 - val_loss: 6.8578 - learn
Epoch 4/10
6/6 ━━━━━━━━━━━━━━━━━━━━ 152s 19s/step - accuracy: 0.5321 - loss: 7.0125 - val_accuracy: 0.7108 - val_loss: 6.8853 - learn
Epoch 5/10
6/6 ━━━━━━━━━━━━━━━━━━━━ 130s 18s/step - accuracy: 0.5005 - loss: 7.0529 - val_accuracy: 0.7470 - val_loss: 6.8684 - learn
Epoch 6/10
6/6 ━━━━━━━━━━━━━━━━━━━━ 149s 18s/step - accuracy: 0.4718 - loss: 7.0112 - val_accuracy: 0.6386 - val_loss: 6.9258 - learn
Epoch 7/10
6/6 ━━━━━━━━━━━━━━━━━━━━ 135s 17s/step - accuracy: 0.5040 - loss: 7.0047 - val_accuracy: 0.6867 - val_loss: 6.9437 - learn
Epoch 8/10
6/6 ━━━━━━━━━━━━━━━━━━━━ 147s 18s/step - accuracy: 0.5016 - loss: 7.0129 - val_accuracy: 0.7229 - val_loss: 6.8848 - learn
```

```
plt.plot(history.history['accuracy'], label='Precisión de entrenamiento en etapa 1')
plt.plot(history.history['val_accuracy'], label='Precisión de validación en etapa 1')
plt.plot(history_finetune.history['accuracy'], label='Precisión de entrenamiento en ajuste fino')
plt.plot(history_finetune.history['val_accuracy'], label='Precisión de validación en ajuste fino')
plt.title('Precisión del modelo durante entrenamiento')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.legend()
plt.show()
```




```
model.load_weights('best_model.keras')
```

```
test_loss, test_accuracy = model.evaluate(validation_generator)
print(f"Precisión final en validación: {test_accuracy}")
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 22s 4s/step - accuracy: 0.7584 - loss: 6.8507
Precisión final en validación: 0.7469879388809204
```

```
img_path = '/content/dataset_1/mouse/images (1).jpeg'
img = tf.keras.preprocessing.image.load_img(img_path, target_size=(224, 224))
img_array = np.expand_dims(tf.keras.preprocessing.image.img_to_array(img) / 255.0, axis=0)
```

```
prediction = model.predict(img_array)
predicted_class = np.argmax(prediction)
class_labels = list(train_generator.class_indices.keys())
print(f"Clase predicha: {class_labels[predicted_class]}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
Clase predicha: mouse
```