

KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN KHOA HỌC MÁY TÍNH

BÁO CÁO THỊ GIÁC MÁY TÍNH

ĐỀ TÀI

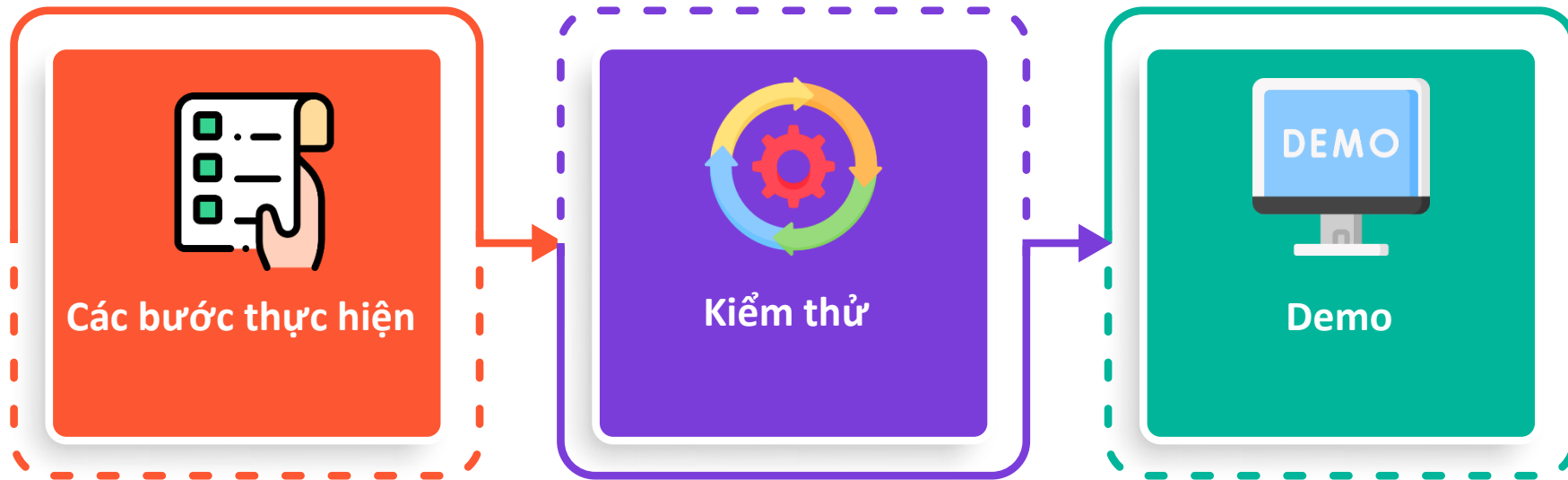
NHẬN DẠNG CÁC LOẠI MÈO

Giáo viên hướng dẫn:
Phạm Nguyên Hoàng

Sinh viên thực hiện:

- Lê Phát Thời B1812307
- Trần Vạn Đại B1812334
- Nguyễn Thị Kiều Trang B1812314
- Phạm Thị Thanh Giang B1812337

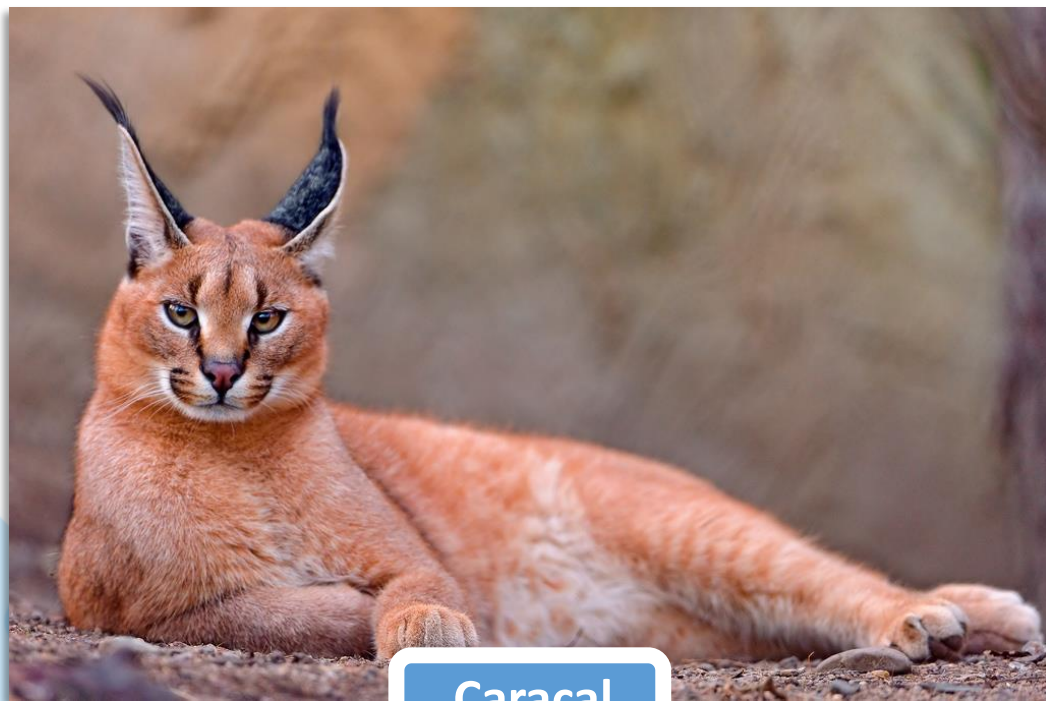
NỘI DUNG BÁO CÁO



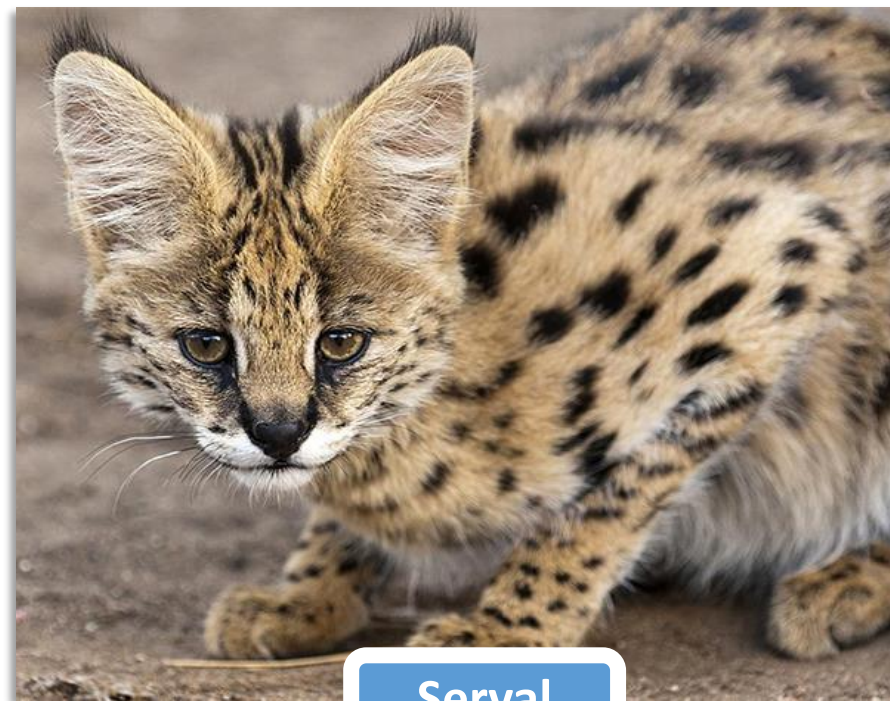
1. CÁC BƯỚC THỰC HIỆN

Bước 1: Thu thập dữ liệu

Thu thập hình ảnh của 8 loài mèo dưới đây:



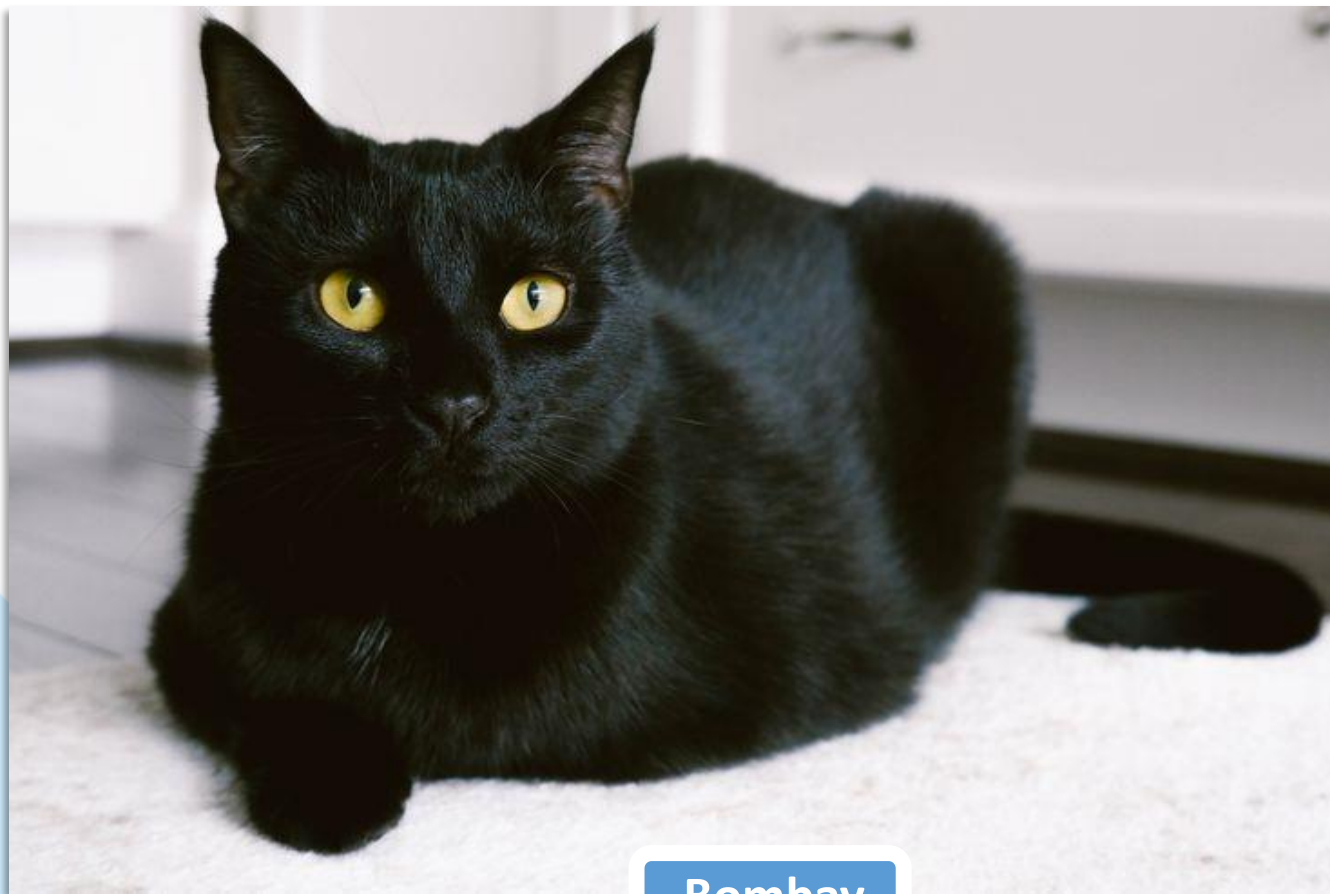
Caracal



Serval

1. CÁC BƯỚC THỰC HIỆN

Bước 1: Thu thập dữ liệu



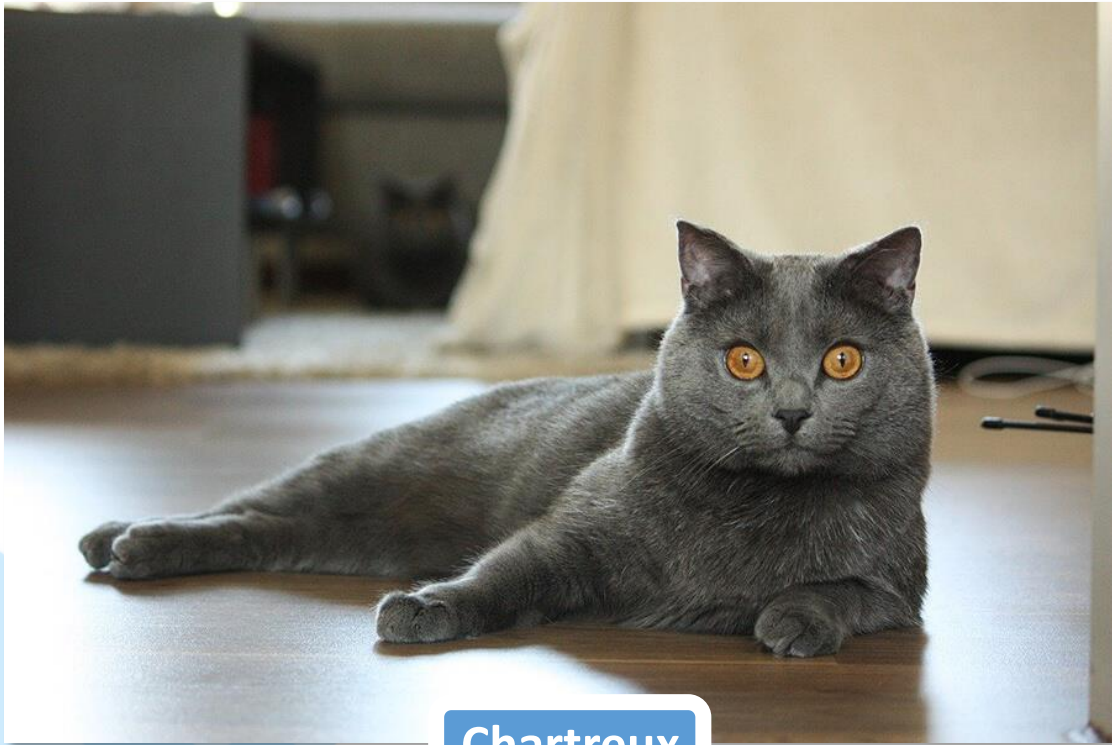
Bombay



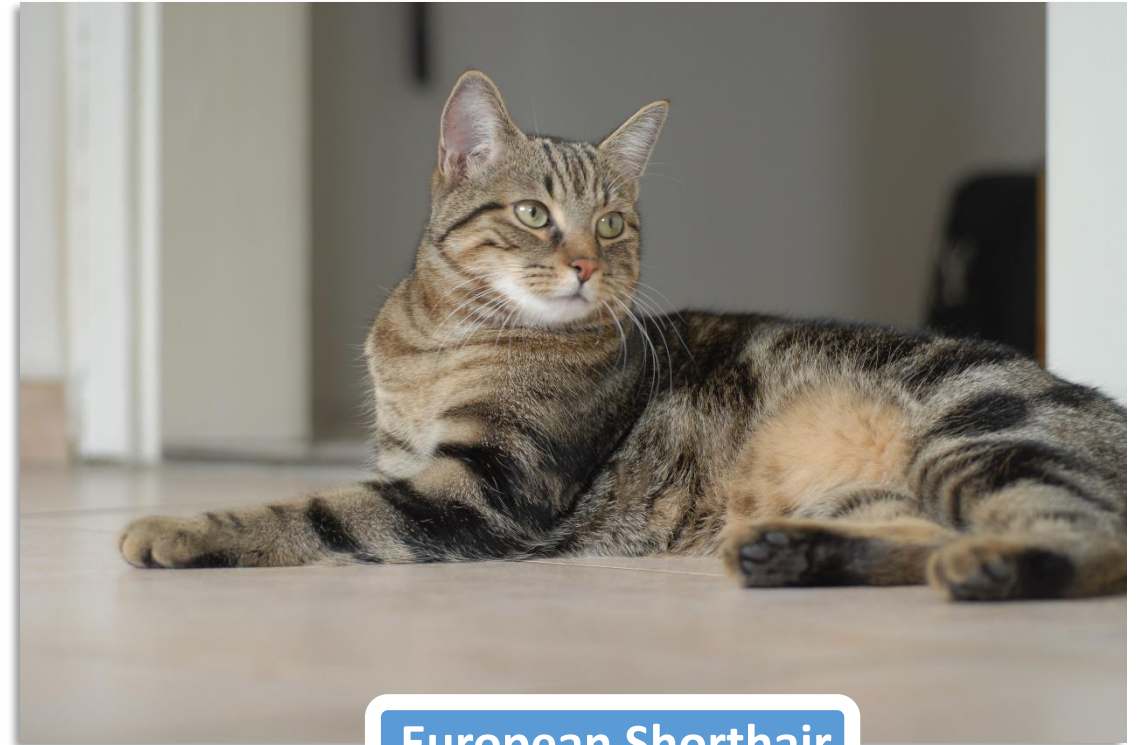
Angora Turc

1. CÁC BƯỚC THỰC HIỆN

Bước 1: Thu thập dữ liệu



Chartreux



European Shorthair

1. CÁC BƯỚC THỰC HIỆN

Bước 1: Thu thập dữ liệu



Sacre de Birmanie



Abyssin

1. CÁC BƯỚC THỰC HIỆN

Bước 2: Kiểm tra định dạng file ảnh

- ❑ Sử dụng code Python để thực hiện kiểm tra toàn bộ ảnh có đúng định dạng hay không.
- ❑ Tensorflow chấp nhận các định dạng file sau: **bmp, gif, jpeg, png**

```
C:\Users\admin\anaconda3\envs\tensorflow\python.exe D:/Tensorflow/workspace/training_demo/images/train/findFile.py
D:\Tensorflow\workspace\training_demo\images\train\anhthesx.png is not an image
D:\Tensorflow\workspace\training_demo\images\train\eps_202.jpg is not an image

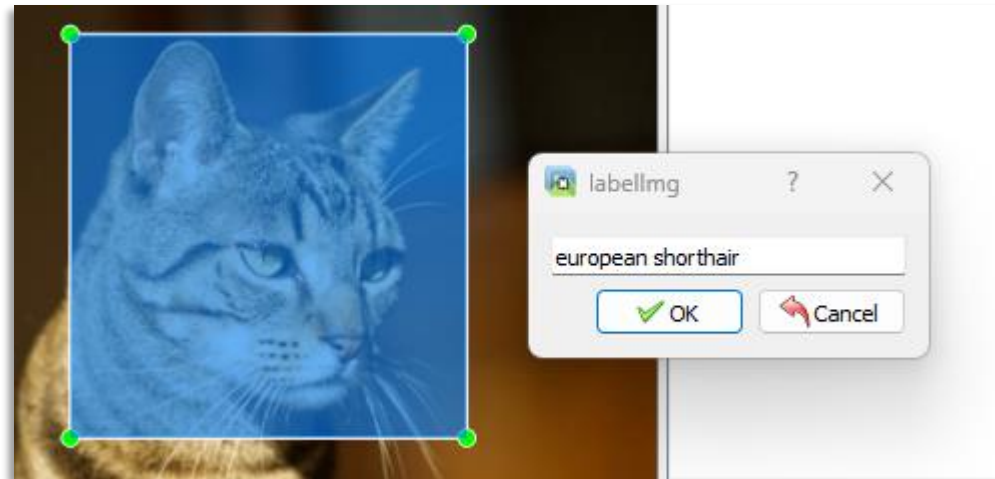
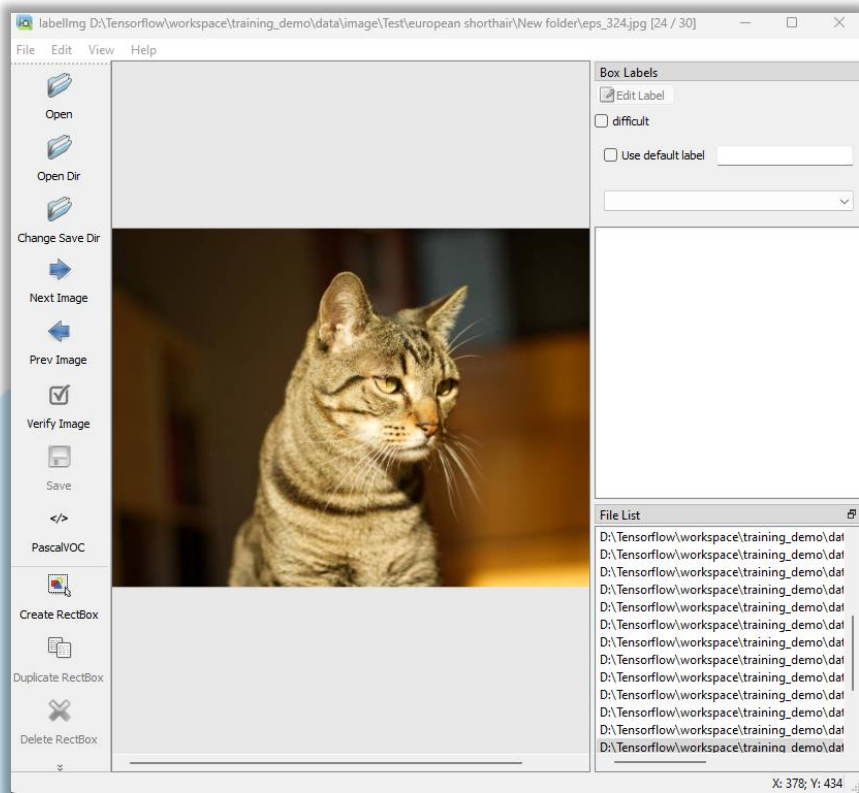
Process finished with exit code 0
```

Thử thay đổi định dạng file bằng
cách thủ công
VD: eps_202.jpeg -> eps_202.jpg

1. CÁC BƯỚC THỰC HIỆN

Bước 3: Gán nhãn

- ☐ Sử dụng **LabelImg** để gán nhãn
- ☐ Vùng gán nhãn: Phần mặt của mèo
- ☐ Dùng công cụ **Create RectBox** để tạo tọa độ vùng (Xmax, Ymax, Xmin, Ymin)



- ☐ Khởi tạo tọa độ
- ☐ Nhập nhãn cần gán

1. CÁC BƯỚC THỰC HIỆN

Bước 3: Gán nhãn

Tập **.XML** được sinh ra tương ứng lưu trữ các thông tin quan trọng như:

- `<filename>`: Tên ảnh gốc
- `<size>`: Kích thước của ảnh gốc
- `<name>`: Tên nhãn đã gán trước đó
- `<bndbox>`: Tọa độ vùng gán nhãn

```
<annotation>
  <folder>New folder</folder>
  <filename>eps_324.jpg</filename>
  <path>D:\Tensorflow\workspace\training_demo\data\image
\Test\europaean shorthair\New folder\eps_324.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>612</width>
    <height>409</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>europaean shorthair</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>199</xmin>
      <ymin>45</ymin>
      <xmax>432</xmax>
      <ymax>282</ymax>
    </bndbox>
  </object>
</annotation>
```

1. CÁC BƯỚC THỰC HIỆN

Bước 4: Kiểm tra nhãn

Sử dụng code Python thực hiện song song 2 chức năng:

- ☐ Đếm số lượng ảnh mèo trong từng lớp
- ☐ Kiểm tra các nhãn bị lỗi trong quá trình gán (sai tên ...)

```
C:\Users\admin\anaconda3\envs\tensorflow\python.exe D:/Tensorflow/workspace/training_demo/images/train/readXML_final.py
{'chartreux': 282, 'european shorthair': 300, 'abyssin': 300, 'angora turc': 300, 'bombay': 300, 'caracal': 300, 'sacre de birmanie': 300, 'serval': 300}

Process finished with exit code 0
```

Trường hợp gán nhãn không lỗi

```
C:\Users\admin\anaconda3\envs\tensorflow\python.exe D:/Tensorflow/workspace/training_demo/images/train/readXML_final.py
Error: 1
File gán nhãn sai là:
['E:\\CODE\\Thi Giác May Tính\\XML ALL\\docobject\\file_1 - Copy.xml']
{'chartreux': 282, 'european shorthair': 300, 'abyssin': 300, 'angora turc': 300, 'bombay': 300, 'caracal': 300, 'sacre de birmanie': 300, 'serval': 300}

Process finished with exit code 0
```

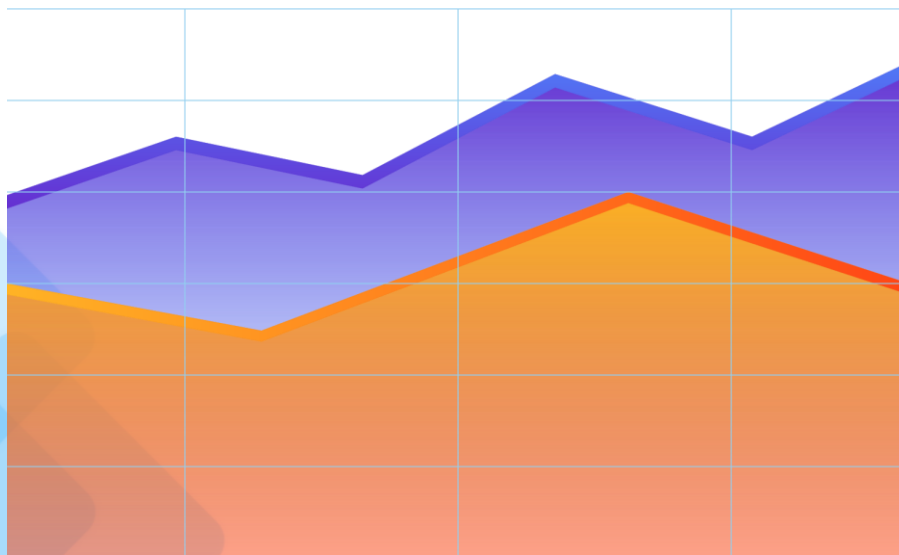
Trường hợp gán nhãn có lỗi -> Chỉ ra tên của file và đường dẫn kèm theo

1. CÁC BƯỚC THỰC HIỆN

Bước 5: Phân vùng tập dữ liệu

- ❑ Sử dụng **80%** ảnh mèo kèm file .XML tương ứng cho thư mục train
- ❑ Sử dụng **20%** ảnh còn lại kèm file

Name	Date modified	Type	Size
test	3/5/2022 12:20 AM	File folder	
train	3/31/2022 7:38 PM	File folder	



TRAIN



TEST

1. CÁC BƯỚC THỰC HIỆN

Bước 6: Tạo file labelmap

- ❑ Tạo các item tương ứng với số lớp, tại mỗi item sẽ chứa id và tên của một lớp mèo.
- ❑ Ánh xạ các nhãn thành giá trị số nguyên để sử dụng cho quá trình huấn luyện và nhận dạng

```
item {
  id: 1
  name: 'chartreux'
}

item {
  id: 2
  name: 'european shorthair'
}

item {
  id: 3
  name: 'caracal'
}

item {
  id: 4
  name: 'bombay'
}

item {
  id: 5
  name: 'angora ture'
}

item {
  id: 6
  name: 'abyssin'
}

item {
  id: 7
  name: 'sacre de birmanie'
}

item {
  id: 8
  name: 'serval'
}
```

1. CÁC BƯỚC THỰC HIỆN



Bước 7: Chuyển *.xml sang TFRecord

- ❑ Sử dụng toàn bộ file *.xml ở 2 thư mục train và test để thực hiện quá trình chuyển đổi

```
python generate_tfrecord.py -x [PATH_TO_IMAGES_FOLDER]/* -l [PATH_TO_ANNOTATIONS_FOLDER]/label_map.pbtxt  
-o [PATH_TO_ANNOTATIONS_FOLDER]/*.record
```

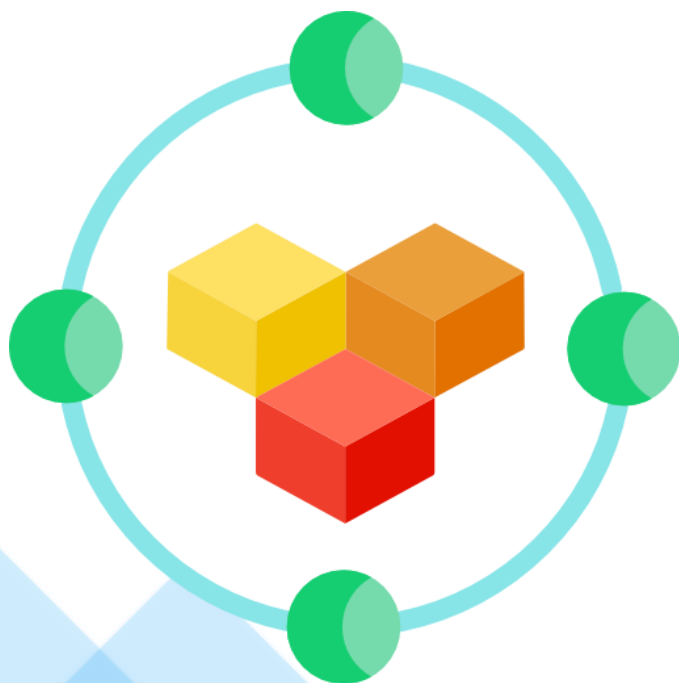
train.record và test.record được tạo ra

```
PS D:\Tensorflow\scripts\preprocessing> python generate_tfrecord.py -x D:/Tensorflow/workspace/training_demo/images/test -l D:/Tensorflow/workspace/training_demo/annotations/label_map.pbtxt -o D:/Tensorflow/workspace/training_demo/annotations/test.record  
Successfully created the TFRecord file: D:/Tensorflow/workspace/training_demo/annotations/test.record  
PS D:\Tensorflow\scripts\preprocessing> python generate_tfrecord.py -x D:/Tensorflow/workspace/training_demo/images/train -l D:/Tensorflow/workspace/training_demo/annotations/label_map.pbtxt -o D:/Tensorflow/workspace/training_demo/annotations/train.record  
Successfully created the TFRecord file: D:/Tensorflow/workspace/training_demo/annotations/train.record  
PS D:\Tensorflow\scripts\preprocessing>
```

 test.record	3/5/2022 1:05 AM	RECORD File	31,079 KB
 train.record	3/5/2022 1:05 AM	RECORD File	127,544 KB

1. CÁC BƯỚC THỰC HIỆN

Bước 8: Cấu hình Model



- Model sử dụng để huấn luyện: SSD ResNet50 V1 FPN 640x640
- Cấu hình lại model trong file ***pipeline.config***, chỉnh sửa lại một số chỗ như
 - ☐ Số lượng lớp (num_classes)
 - ☐ Batch_size
 - ☐ Các đường dẫn
 - ☐ fine_tune_checkpoint_type: "detection"

1. CÁC BƯỚC THỰC HIỆN

Bước 9: Train Model

- ❑ Chuyển sang thư mục training_demo.
- ❑ Thực hiện câu lệnh sau trong cmd

```
python model_main_tf2.py --model_dir=models/my_ssd_resnet50_v1_fpn  
--pipeline_configpath=models/my_ssd_resnet50_v1_fpn/pipeline.config
```

```
INFO:tensorflow:Step 24900 per-step time 0.669s  
I0305 05:53:36.231349 17204 model_lib_v2.py:705] Step 24900 per-step time 0.669s  
INFO:tensorflow: {'Loss/classification_loss': 0.14740227,  
'Loss/localization_loss': 0.044395402,  
'Loss/regularization_loss': 0.15771493,  
'Loss/total_loss': 0.34951258,  
'learning_rate': 1.8656253e-06}  
I0305 05:53:36.231349 17204 model_lib_v2.py:708] {'Loss/classification_loss': 0.14740227,  
'Loss/localization_loss': 0.044395402,  
'Loss/regularization_loss': 0.15771493,  
'Loss/total_loss': 0.34951258,  
'learning_rate': 1.8656253e-06}  
INFO:tensorflow:Step 25000 per-step time 0.669s  
I0305 05:54:43.052117 17204 model_lib_v2.py:705] Step 25000 per-step time 0.669s  
INFO:tensorflow: {'Loss/classification_loss': 0.086163044,  
'Loss/localization_loss': 0.02740234,  
'Loss/regularization_loss': 0.15771484,  
'Loss/total_loss': 0.27128023,  
'learning_rate': 0.0}  
I0305 05:54:43.052117 17204 model_lib_v2.py:708] {'Loss/classification_loss': 0.086163044,  
'Loss/localization_loss': 0.02740234,  
'Loss/regularization_loss': 0.15771484,  
'Loss/total_loss': 0.27128023,  
'learning_rate': 0.0}  
PS D:\Tensorflow\workspace\training_demo>
```

❑ Tổng số step: **25000**

1. CÁC BƯỚC THỰC HIỆN

Bước 10: Xuất Model

- ❑ Copy script `exporter_main_v2.py` và chuyển nó vào thư mục `training_demo`.
- ❑ Sau đó thực hiện lệnh sau vào cmd tại thư mục `training_demo`:

```
python .\exporter_main_v2.py --input_type image_tensor
--pipeline_config_path .\models\my_ssd_resnet50_v1_fpn\pipeline.config
--trained_checkpoint_dir .\models\my_ssd_resnet50_v1_fpn\
--output_directory .\exported-models\my_model
```

```
Use:
results = tf.nest.map_structure(tf.stop_gradient, tf.map_fn(fn, elems))
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.meta_architectures.ssd_meta_arch.SSDMetaArch object at 0x000001DC8DA1FDF0>
W0305 05:59:02.870842 2456 save_impl.py:71] Skipping full serialization of Keras layer <object_detection.meta_architectures.ssd_meta_arch.SSDMetaArch object at 0x000001DC8DA1FDF0>
2022-03-05 05:59:09.209323: W tensorflow/python/util/util.cc:368] Sets are not currently considered sequences, but this may change in the future, so consider using DataStructures.SequenceWrapper in some cases.
W0305 05:59:20.819850 2456 save.py:263] Found untraced functions such as WeightSharedConvolutionalBoxPredictor_layer_call_fn, WeightSharedConvolutionalBoxPredictor_layer_call_and_return_conditional_losses, WeightSharedConvolutionalBoxHead_layer_call_fn, WeightSharedConvolutionalBoxHead_layer_call_and_return_conditional_losses, WeightSharedConvolutionalBoxPredictor_layer_call_and_return_conditional_losses, WeightSharedConvolutionalBoxHead_layer_call_and_return_conditional_losses. These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: .\exported-models\my_model\saved_model\assets
I0305 05:59:26.388021 2456 builder_impl.py:783] Assets written to: .\exported-models\my_model\saved_model\assets
INFO:tensorflow:Writing pipeline config file to .\exported-models\my_model\pipeline.config
I0305 05:59:27.106415 2456 config_util.py:253] Writing pipeline config file to .\exported-models\my_model\pipeline.config
PS D:\Tensorflow\workspace\training_demo>
```

1. CÁC BƯỚC THỰC HIỆN

Bước 11: Xuất Model

Name	Date modified	Type	Size
checkpoint	3/5/2022 5:59 AM	File folder	
saved_model	3/5/2022 5:59 AM	File folder	
pipeline.config	3/5/2022 5:59 AM	XML Configuratio...	5 KB

Cấu trúc thư mục **my_model** sau khi xuất model thành công

checkpoint	3/5/2022 5:59 AM	File	1 KB
ckpt-0.data-00000-of-00001	3/5/2022 5:59 AM	DATA-00000-OF-0...	124,251 KB
ckpt-0.index	3/5/2022 5:59 AM	INDEX File	31 KB

Cấu trúc bên trong thư mục **checkpoint**

Name	Date modified	Type	Size
assets	3/5/2022 5:59 AM	File folder	
variables	3/5/2022 5:59 AM	File folder	
saved_model.pb	3/5/2022 5:59 AM	PB File	7,959 KB

Cấu trúc bên trong thư mục **saved_model**

2. KIỂM THỬ

Lập bảng Confusion matrix

	Abyssin	Angora turc	Caracal	Chartreux	European Shorthair	Serval	Bombay	Sacre de Birnamie
Abyssin	27	0	1	0	2	0	0	0
Angora turc	1	29	0	0	0	0	0	0
Caracal	1	0	28	0	1	0	0	0
Chartreux	0	0	0	28	1	0	1	0
European Shorthair	0	0	0	1	29	0	0	0
Serval	0	0	0	0	1	29	0	0
Bombay	0	1	0	0	0	0	26	3
Sacre de Birnamie	0	1	0	0	1	0	0	28

2. KIỂM THỬ

Tính F1_score (Liên hệ giữa Precision và Recall)

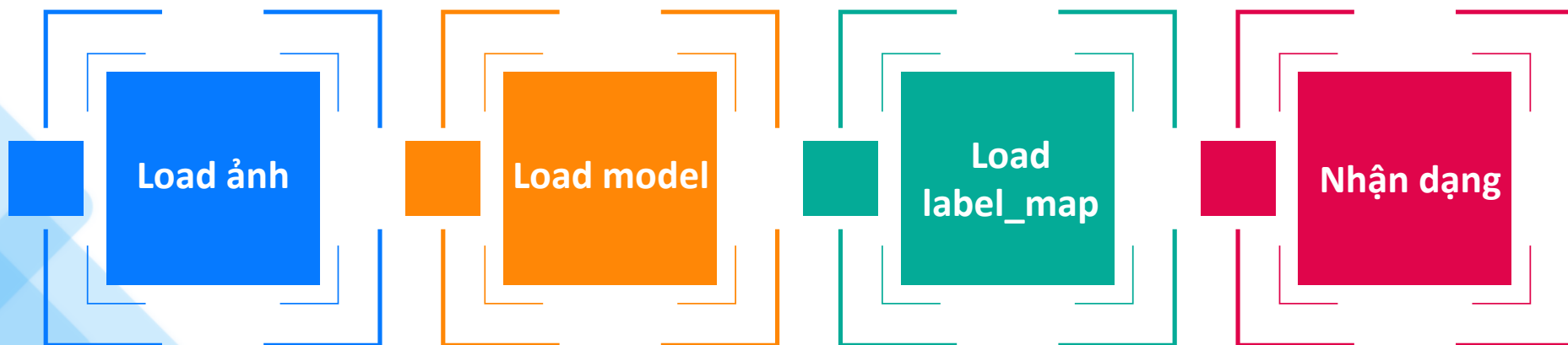
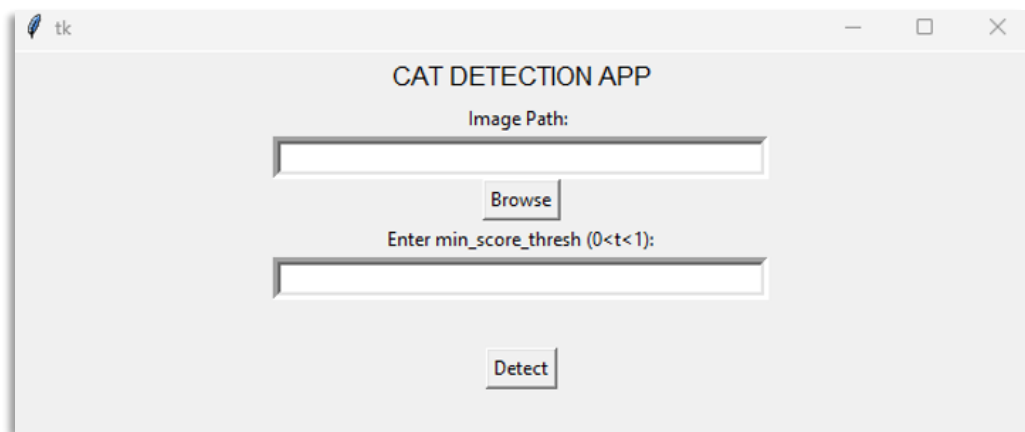
	Precision	Recall
Abyssin	0.90	0.93
Angora turc	0.97	0.94
Caracal	0.93	0.97
Chartreux	0.93	0.97
European Shorthair	0.97	0.83
Serval	0.97	1.00
Bombay	0.87	0.96
Sacre de Birnamie	0.93	0.90
Average	0.93	0.94

$$F1_score = \frac{2 * recall * precision}{recall + precision} = 0.93$$

3. DEMO CHƯƠNG TRÌNH

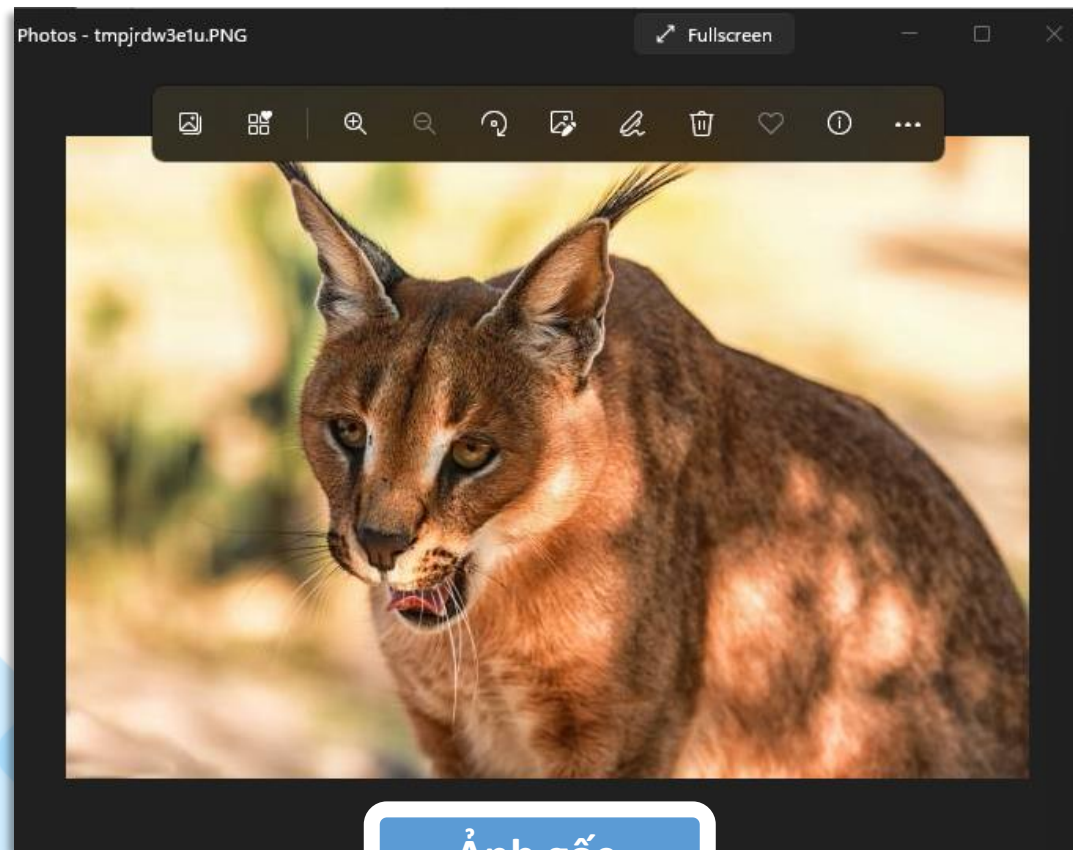
Giao diện

- Image Path: Chọn hình ảnh cần nhận dạng
- Min_score_thresh: Ngưỡng nhận dạng
- Detect: Sử dụng tf2 saved model

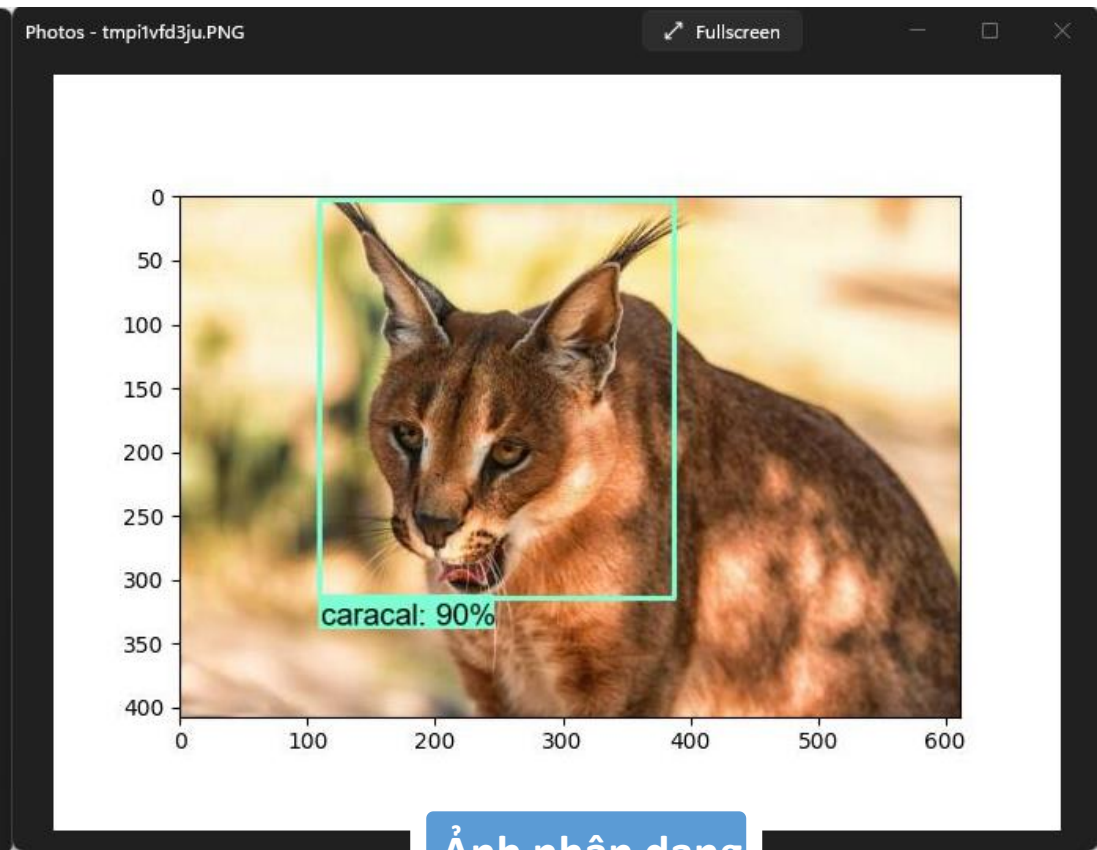


3. DEMO CHƯƠNG TRÌNH

KẾT QUẢ



Ảnh gốc



Ảnh nhận dạng



Thank You

CẢM ƠN THẦY VÀ CÁC BẠN ĐÃ LẮNG NGHE !