

Object Oriented Programming - Day 15

Ngày 25 tháng 6 năm 2024

Ngày thực hiện:	25/06/2024
Người thực hiện:	Đinh Thị Tâm
Nguồn:	AIO2024 - Day 15
Nguồn dữ liệu (nếu có):	Link Sources code
Từ khóa:	Object Oriented Programming
Người tóm tắt:	Đinh Thị Tâm

1. Mô tả

Streamlit là một thư viện Python mã nguồn mở mạnh mẽ giúp bạn dễ dàng tạo ứng dụng web tương tác từ mã Python. Nó cung cấp các API đơn giản và trực quan để hiển thị dữ liệu, tạo biểu đồ, thu thập đầu vào của người dùng và xây dựng giao diện người dùng đẹp mắt, tất cả mà không cần kiến thức chuyên sâu về HTML, CSS hoặc JavaScript. Các ứng dụng của Streamlit bao gồm:

- **Bảng điều khiển dữ liệu:** Tạo bảng điều khiển để trực quan hóa dữ liệu thời gian thực, theo dõi hiệu suất và khám phá xu hướng.
- **Ứng dụng học máy:** Triển khai các mô hình học máy và cho phép người dùng tương tác với chúng để đưa ra dự đoán hoặc thực hiện các tác vụ khác.
- **Công cụ giáo dục:** Tạo các ứng dụng giáo dục tương tác để giúp học sinh học các khái niệm mới và thực hành kỹ năng.
- **Ứng dụng nguyên mẫu:** Tạo nguyên mẫu nhanh chóng cho các ý tưởng ứng dụng web mới mà không cần đầu tư nhiều thời gian và nguồn lực.
- **Công cụ phân tích:** Xây dựng các công cụ phân tích để giúp người dùng hiểu dữ liệu của họ và đưa ra quyết định sáng suốt hơn

2. **Bài tập:** Bạn hãy tạo 1 trang Streamlit sử dụng model CNN đã được train với bộ dữ liệu y tế sau: Model CNN ảnh sinh thiết. Hãy dựa vào đoạn code mẫu bên dưới load model ResNet 152 với Pre-train = True và load model đã được train sẵn và triển khai lên Streamlit

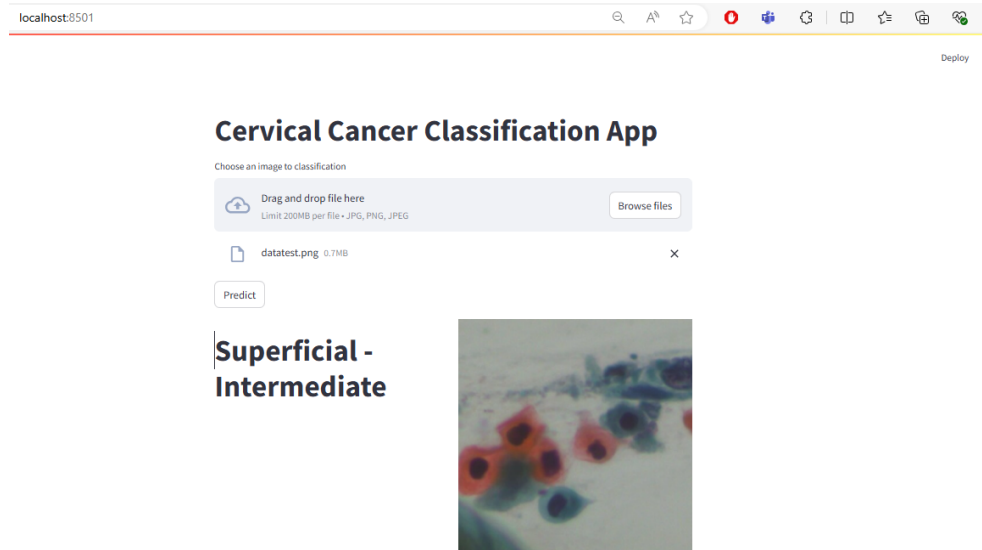
(a) Code

```
1 import torch
2 import torchvision.models as models
3 from torchvision import transforms
4 from PIL import Image
5 import streamlit as st
6 import torch.nn as nn
7 import os
8 CLASS_NAMES = ['Dyskeratotic',
```

```
9         'Koilocytotic',
10         'Metaplastic',
11         'Parabasal',
12         'Superficial - Intermediate']
13 MODEL_NAME = 'model_weights.pth'
14
15
16 def predict(model_name, img_path):
17     # load the model
18     model = models.resnet152(pretrained=True)
19     model.fc = nn.Linear(in_features=2048, out_features=5, bias=True)
20     weights = torch.load(model_name, map_location='cpu')
21     model.load_state_dict(weights)
22
23     # preprocess the image
24     prep_img_mean = [0.485, 0.456, 0.406]
25     prep_img_std = [0.229, 0.224, 0.225]
26     transform = transforms.Compose(
27         [
28             transforms.Resize(224),
29             transforms.ToTensor(),
30             transforms.Normalize(mean=prep_img_mean, std=prep_img_std
31                                 ),
32         ]
33     )
34     image = Image.open(img_path)
35     preprocessed_image = transform(image).unsqueeze(0)
36
37     # predict the class
38     model.eval()
39     output = model(preprocessed_image)
40     pred_idx = torch.argmax(output, dim=1)
41     predicted_class = CLASS_NAMES[pred_idx]
42     return predicted_class
43
44
45 def getpath_of_image(filename):
46     current_file_path = os.path.abspath(__file__)
47     current_directory = os.path.dirname(current_file_path)
48     path_file = current_directory + filename
49     return path_file
50
51
52 def mainprocess():
53     myfile = 'datatest.png'
54     myfile = getpath_of_image(myfile)
55     st.title('Cervical Cancer Classification App')
56     file = st.file_uploader('Choose an image to classification', type=[
57         'jpg', 'png', 'jpeg'])
58     if st.button("Predict"):
59         if file is not None:
60
61             try:
62                 model_name = getpath_of_image("/models/"+MODEL_NAME)
63                 predict_result = predict(model_name, file)
64                 col1, col2 = st.columns(2)
65                 col1.title(predict_result)
66                 col2.image(file)
67             except:
68                 st.write('Not detected object')
```

```
69  
70  
71 mainprocess()
```

(b) Kết quả thực thi chương trình



Hình 1: Result of prediction