Лабораторная работа №1 - Встраивание моделей компьютерного зрения в веб-приложение (Django)

Цель:

Разработать простое веб-приложение на Django, в которое будет встроена модель для классификации изображений. Веб-приложение должно позволять пользователю загружать изображение, и отображать предсказание классификатора.

Шаги разработки

1. Подготовка окружения

- Создайте проект Django

```
django-admin startproject image_classifier
cd image_classifier
```

- Создайте приложение для работы с изображениями:

```
python manage.py startapp classifier
```

Убедитесь, что в `settings.py` ваше новое приложение добавлено в список `INSTALLED_APPS`

```
INSTALLED_APPS = [
    ...
    'classifier',
]
```

Также допишите путь в `settings.py` для `TEMPLATES`

- Замените:

```
'DIRS': []
```

- на:

```
'DIRS': [BASE_DIR / 'templates']
```

2. Модель классификации изображений

Перед использованием модели нужно загрузить ее и определить логику для предсказаний.

- В файле `models.py` в приложении `classifier` добавьте код для загрузки предобученной модели (в примере PyTorch)

```
import torch
from torchvision import models, transforms
from PIL import Image

model_path = os.path.join(settings.BASE_DIR, 'classifier', 'путь к вашей модели.pth')
model = torch.load(model_path, map_location=torch.device('cpu'))
model.eval() # установка в режим оценки
```

- Здесь же нужно добавить код для преобразования изображения для модели (можно использовать свой или воспользоваться предоставленным)

```
transform = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
```

- Далее напишем функцию для работы модели

```
def classify_image(image_path):
    # Открытие изображения и его предобработка
    image = Image.open(image_path)
    image = transform(image).unsqueeze(0)  # Добавление batch dimension

    # Прогон через модель
    with torch.no_grad():
        output = model(image)

    # Преобразование результатов
    probabilities = torch.nn.functional.softmax(output[0], dim=0)
    return probabilities
```

- Полный код `models.py` :

```
import os
import torch
from torchvision import transforms
from PIL import Image
from django.db import models
from image_classifier import settings

# Загрузка предобученной модели
model_path = os.path.join(settings.BASE_DIR, 'classifier', 'resnet18.pth')
model = torch.load(model_path, map_location=torch.device('cpu'))
model.eval()

# Преобразование изображения для модели
transform = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])


def classify_image(image_path):
    # Открытие изображения и его предобработка
    image = Image.open(image_path)
    image = transform(image).unsqueeze(0)  # Добавление batch dimension

    # Прогон через модель
```

```
    with torch.no_grad():
        output = model(image)

    # Преобразование результатов
    probabilities = torch.nn.functional.softmax(output[0], dim=0)
    return probabilities

class ImageUpload(models.Model):
    image = models.ImageField(upload_to='uploads/')
    uploaded_at = models.DateTimeField(auto_now_add=True)
```

3. Модель Django для хранения изображений

В `classifier/models.py` создайте модель для загрузки изображений:

```
from django.db import models

class ImageUpload(models.Model):
    image = models.ImageField(upload_to='uploads/')
    uploaded_at = models.DateTimeField(auto_now_add=True)
```

После этого нужно выполнить миграции

```
python manage.py makemigrations
python manage.py migrate
```

4. Загрузка изображения

Создайте форму для загрузки изображения в `classifier/forms.py`

```
from django import forms
from .models import ImageUpload

class ImageUploadForm(forms.ModelForm):
    class Meta:
        model = ImageUpload
        fields = ['image']
```

5. Представление (view) Django для обработки изображений

Создайте вьюшку в `classifier/views.py` для загрузки изображения и классификации

```
from django.shortcuts import render
from .forms import ImageUploadForm
from .models import ImageUpload
from .models import classify_image

def upload_image(request):
    if request.method == 'POST':
        form = ImageUploadForm(request.POST, request.FILES)
        if form.is_valid():
            image_instance = form.save()
            image_path = image_instance.image.path  # Путь к загруженному изображению
            predictions = classify_image(image_path)  # Получаем предсказания

            # Вытаскиваем классы и вероятности для отображения
            class_probabilities = [(i, float(p)) for i, p in enumerate(predictions)]
            return render(request, 'classifier/result.html', {'class_probabilities': c
    else:
        form = ImageUploadForm()

    return render(request, 'classifier/upload.html', {'form': form})
```

6. Шаблоны для загрузки и отображения результатов

Создайте папку `templates` в корне проекта

Создайте шаблон для загрузки `templates/classifier/upload.html`

```
<!DOCTYPE html>
<html>
<head>
    <title>Загрузка изображения</title>
</head>
<body>
    <h2>Загрузите изображение для классификации</h2>
    <form method="post" enctype="multipart/form-data">
```

```
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Загрузить</button>
    </form>
</body>
</html>
```

И шаблон для отображения результатов классификации `templates/classifier/result.html`

```
<!DOCTYPE html>
<html>
<head>
    <title>Результаты классификации</title>
</head>
<body>
    <h2>Распределение по классам</h2>
    <ul>
        {% for class, prob in class_probabilities %}
        <li>Класс {{ class }}: {{ prob|floatformat:2 }}</li>
        {% endfor %}
    </ul>
    <a href="{% url 'upload_image' %}">Загрузить другое изображение</a>
</body>
</html>
```

7. Настройка маршрутов

В файле `classifier/urls.py` создайте маршруты:

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.upload_image, name='upload_image'),
]
```

Далее добавьте этот URL в основной `image_classification/urls.py`

```
from django.contrib import admin
```

```
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('classifier/', include('classifier.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

8. Настройка для загрузки медиафайлов

В `settings.py` добавьте пути для хранения изображений

```
MEDIA_URL = '/media/'
MEDIA_ROOT = BASE_DIR / 'media'
```

9. Запуск проекта

```
python manage.py runserver
```

Задания

1. Загрузка нескольких изображений*

Сейчас возможна загрузка только одного изображения, сделайте возможность загружать несколько изображений

Подсказка: нужно изменить формы, вьюшку, темплейты

2. Сделать визуал для отражения результатов*

На данный момент результаты отображаются не очень информативно. Можете придумать, как это исправить