

Лабораторная работа №3 - Встраивание моделей компьютерного зрения в веб-приложение (Streamlit)

Цель:

Создать веб-приложение на Streamlit, которое позволит загружать изображение, использовать модель классификации изображений для предсказаний и отображать результаты.

Шаги разработки

1. Подготовка окружения

Установите необходимые библиотеки:

```
pip install streamlit torch torchvision pillow
```

2. Структура проекта

Здесь отображена структура будущего проекта:

```
streamlit_image_classifier/  
├─ app.py  
├─ models.py  
└─ classifier/  
    └─ resnet18.pth
```

3. Модель классификации изображений

Создайте файл `models.py`, в котором реализована логика загрузки предобученной модели (используем ResNet18 из PyTorch) и обработки изображения для предсказания.

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

# Загрузка модели ResNet18
model = models.resnet18(pretrained=True)
model.eval()

# Преобразования 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):
    image = transform(image).unsqueeze(0)
    with torch.no_grad():
        output = model(image)
    probabilities = torch.nn.functional.softmax(output[0], dim=0)
    return probabilities
```

4. Streamlit приложение

Создайте основной файл приложения `app.py`.

```
import streamlit as st
from PIL import Image
import torch
from models import classify_image

# Заголовок приложения
st.title("Классификация изображений")

# Инструкция для пользователя
st.write("Загрузите изображение, чтобы получить предсказание модели")

# Загрузка изображения
uploaded_file = st.file_uploader("Выберите изображение...", type=["jpg", "jpeg", "png"])

if uploaded_file is not None:
    # Отображение загруженного изображения
    image = Image.open(uploaded_file)
    st.image(image, caption="Загруженное изображение", use_column_width=True)

    # Кнопка для запуска классификации
    if st.button("Классифицировать"):
        # Предсказание
        predictions = classify_image(image)

        # Отображение результатов
        st.write("Распределение по классам:")
        for i, prob in enumerate(predictions):
            st.write(f"Класс {i}: {prob:.2f}")
```

5. Запуск приложения

Запустите Streamlit-приложение из командной строки:

```
streamlit run app.py
```

Приложение будет доступно по адресу <http://localhost:8501>.

Задания

1. Загрузка нескольких изображений:

Добавьте возможность загрузки и классификации нескольких изображений. Подсказка: Используйте `st.file_uploader(..., accept_multiple_files=True)` для загрузки нескольких изображений и настройте обработку каждого из них.

2. Визуализация вероятностей:

Улучшите представление результатов с помощью гистограммы или столбчатой диаграммы ненулевых вероятностей классов. Подсказка: Можно использовать `st.bar_chart` для визуализации результатов предсказания.