

Architecture Classifier

1. Lista osób:
 - a. Karolina Koza
2. Temat: Splotowa sieć neuronowa rozpoznająca podstawowe style architektoniczne. Oddana do użytku w postaci Single Page Application.
3. Korzystając z dostępnych bibliotek języka Python zamierzam zbudować splotową sieć neuronową rozpoznającą 9 stylów architektonicznych (art deco, barok, konstruktywizm, gotyk, minimalizm, modernizm, neoklasycyzm, postmodernizm i renesans). Następnie mam zamiar napisać prostą aplikację webową która pobiera zdjęcie od użytkownika i wykorzystując stworzony wcześniej model przewiduje styl architektoniczny najbliższy temu przedstawionemu na zdjęciu.
4. Lista technologii:
 - a. Python
 - b. Flask
 - c. tensorflow
 - d. keras
 - e. numpy

Istniejące rozwiązania

1. Google Lens

Opis:

Google Lens to narzędzie do analizy obrazów wykorzystujące sztuczną inteligencję do identyfikacji obiektów na zdjęciach. Może rozpoznawać budynki i podawać ich nazwę oraz informacje historyczne.

Funkcjonalności:

- Rozpoznawanie budynków i obiektów architektonicznych.
- Podawanie informacji o obiektach na podstawie bazy danych Google.
- Integracja z Google Search i Google Maps.

Technologie:

- AI i uczenie maszynowe (ML).
- Rozpoznawanie obrazu (computer vision).

Dostępność:

- Aplikacje mobilne (Android, iOS).
- Wbudowane w Google Photos i Asystenta Google.

Cena:

- Darmowe.

Zalety:

- ✓ Duża baza danych architektonicznych.
- ✓ Łatwa dostępność i integracja z ekosystemem Google.
- ✓ Szybka analiza zdjęć.

Wady:

- ✗ Nie zawsze rozpoznaje styl architektoniczny, raczej podaje nazwę budynku.
- ✗ Brak szczegółowej klasyfikacji stylów architektonicznych.
- ✗ Ograniczona dokładność dla mniej znanych budynków.

Wnioski:

Google Lens może być inspiracją do wykorzystania AI w rozpoznawaniu architektury, jednak jego główną wadą jest brak dokładnej klasyfikacji stylów.

2. ArchiMaps

Opis:

ArchiMaps to aplikacja mobilna pomagająca użytkownikom odkrywać budynki i ich style architektoniczne na podstawie geolokalizacji i zdjęć.

Funkcjonalności:

- Katalog budynków podzielony według stylów architektonicznych.
- Geolokalizacja i rekomendacje miejsc do odwiedzenia.
- Informacje historyczne o budynkach.

Technologie:

- Baza danych architektonicznych.
- Geolokalizacja i interaktywna mapa.

Dostępność:

- Aplikacje mobilne (Android, iOS).
- Strona internetowa.

Cena:

- Częściowo darmowe, wersja premium z dodatkowymi funkcjami.

Zalety:

- ✓ Rozbudowana baza danych architektonicznych.
- ✓ Możliwość wyszukiwania budynków według stylów.
- ✓ Intuicyjna nawigacja po mapie.

Wady:

- ✗ Brak automatycznego rozpoznawania stylu ze zdjęcia.
- ✗ Ograniczona liczba budynków w bazie.
- ✗ Część funkcji dostępna tylko w wersji premium.

Wnioski:

ArchiMaps to dobra baza wiedzy o stylach architektonicznych, ale brak automatycznej analizy zdjęć ogranicza jego zastosowanie do projektu.

3. StyleSnap (Amazon)

Opis:

StyleSnap to narzędzie AI Amazona do rozpoznawania stylów ubrań na podstawie zdjęć. Choć dotyczy mody, wykorzystuje technologie, które można by zaadaptować do architektury.

Funkcjonalności:

- Analiza zdjęcia i porównywanie z bazą danych.
- Sugestie podobnych stylów.
- Integracja z platformą e-commerce.

Technologie:

- AI i deep learning.
- Computer vision.

Dostępność:

- Amazon (aplikacja mobilna i strona internetowa).

Cena:

- Darmowe.

Zalety:

- ✓ Skuteczny algorytm rozpoznawania stylu.
- ✓ Szeroka baza danych.
- ✓ Intuicyjna obsługa.

Wady:

- ✗ Dotyczy mody, nie architektury.
- ✗ Skupia się na porównywaniu produktów, nie na klasyfikacji.

Wnioski:

StyleSnap pokazuje, że AI może skutecznie identyfikować style na podstawie obrazu. Można wykorzystać podobne podejście do klasyfikacji stylów architektonicznych.

Podsumowanie

Analiza powyższych rozwiązań pokazuje, że istnieją narzędzia rozpoznające budynki, katalogujące style oraz wykorzystujące AI do klasyfikacji wizualnej. Jednak żadne z nich nie łączy wszystkich tych funkcji w jedno spójne narzędzie do automatycznego rozpoznawania stylu architektonicznego na podstawie zdjęcia.

Najważniejsze wnioski dla projektu:

- ✓ Warto zastosować AI do analizy stylu architektonicznego, podobnie jak StyleSnap dla mody.
- ✓ Należy stworzyć własną bazę stylów, ponieważ obecne rozwiązania skupiają się na konkretnych budynkach.
- ✓ Integracja z interaktywną mapą (jak ArchiMaps) mogłaby wzbogacić funkcjonalność.
- ✓ Google Lens pokazuje, że rozpoznawanie obrazów jest możliwe, ale wymaga bardziej zaawansowanej klasyfikacji stylów.

Dzięki tej analizie możemy opracować unikalne rozwiązanie, które będzie precyzyjnie rozpoznawało style architektoniczne na podstawie zdjęć, łącząc najlepsze cechy istniejących narzędzi i eliminując ich wady.

Wymagania funkcjonalne i нефunkcjonalne

1. Przetwarzanie obrazów użytkownika

- Użytkownik może przesłać zdjęcie budynku.
- Aplikacja wysyła zdjęcie do serwera, gdzie model CNN analizuje styl architektoniczny.
- Wynik klasyfikacji zwracany jest użytkownikowi w czytelnej formie.
Możliwość przeglądania przykładowych zdjęć dla każdego stylu architektonicznego.

2. Interfejs użytkownika (SPA)

Prosty i intuicyjny interfejs umożliwiający:

- Przesyłanie zdjęcia
- Otrzymywanie wyników klasyfikacji
- Opcjonalne filtrowanie i przeglądanie stylów
Obsługa komunikatów błędów (np. „Nieprawidłowy format pliku” lub „Nie udało się rozpoznać stylu”).

3. Integracja modelu CNN

- Backend przetwarza obraz i zwraca wynik do aplikacji frontendowej.
- Możliwość rozwoju modelu - np. dodawanie nowych stylów.

4. Wymagania niefunkcjonalne

- Intuicyjny interfejs użytkownika.
- Krótki czas przetwarzania zdjęcia.

Lista technologii

- **Python** - Język programowania wykorzystywany do budowy modelu AI i serwera aplikacji. Wybrany ze względu na bogaty ekosystem bibliotek do uczenia maszynowego oraz łatwość implementacji.
- **Flask** - Lekki framework webowy używany do tworzenia backendu aplikacji. Pozwala na łatwą integrację modelu AI z frontendem aplikacji.
- **TensorFlow** - Biblioteka do uczenia maszynowego, umożliwiająca budowę i trenowanie modelu CNN. Zapewnia wysoką wydajność i szerokie wsparcie dla modeli deep learning.
- **Keras** - Wysokopoziomowe API do TensorFlow, pozwalające na szybkie tworzenie i trenowanie modeli neuronowych. Upraszcza proces budowy modelu CNN.
- **NumPy** - Biblioteka do operacji na tablicach i macierzach numerycznych. Wspiera przetwarzanie danych wejściowych oraz operacje matematyczne potrzebne do działania modelu.