

Warsztaty uczenia maszynowego - projekt

Paweł Drabarek, Mateusz Fabisiewicz, Jakub Gazewski,
Adrian Goworek, Adam Wodzisławski

Kwiecień 2022

1. Temat i opis

Tematem projektu jest stworzenie rozwiązania wykorzystującego uczenie maszynowe służącego do analizy obrazów, na których mogą znajdować się ryby oceaniczne. Zadaniem programu jest zakwalifikowanie obrazu do jednej z 8 kategorii - 6 z nich dotyczy gatunku ryby, która znajduje się na obrazie (mogą to być – tuńczyk biały, tuńczyk wielkooki, koryfena, strojnik, gatunek rekina lub tuńczyk żółtopłetwy), pozostałe dwie odpowiadają sytuacji gdy na obrazie nie znajduje się ryba lub znajduje się ryba innego gatunku niż te rozpoznawane.

Szczegóły dotyczące danych i zadaniu do wykonania znajdują się pod linkiem [1].

2. Założenia technologiczne

Program zostanie wykonany w języku Python. Zostanie wykorzystana biblioteka umożliwiająca korzystanie z sieci neuronowej, prawdopodobnie PyTorch lub TensorFlow, nasze doświadczenie podczas pisania tego etapu dokumentacji nie pozwala nam jeszcze określić która z wymienionych bibliotek najlepiej pasuje do naszego problemu (być może ostatecznie wybrana zostanie zupełnie inna, nie wymieniona).

3. Podział pracy

W projekcie możemy rozróżnić takie etapy pracy jak:

- przygotowanie zbiorów treningowych i walidacyjnych;
- ustalenie odpowiedniego modelu, wyodrębnienie atrybutów i ustalenie hiperparametrów;
- implementacja i trening sieci neuronowej;
- testy modelu;
- sporządzenie dokumentacji.

W projekcie porównamy działanie modeli bez fragmentacji (model A) i z fragmentacją (model B) wejściowego obrazu. Proponujemy następujący podział prac dla poszczególnych członków zespołu:

- Paweł Drabarek: przygotowanie zbiorów treningowych i walidacyjnych, implementacja modelu B, pomoc w sporządzaniu dokumentacji;
- Mateusz Fabiszewicz: ustalenie szczegółów modelu B i jego implementacja, sporządzenie dokumentacji na temat modelu B;
- Jakub Gazewski: ustalenie szczegółów modelu B i jego implementacja, testowanie modelu B, pomoc w sporządzaniu dokumentacji;
- Adrian Goworek: ustalenie szczegółów modelu A i jego implementacja, testowanie modelu A, pomoc w sporządzeniu dokumentacji;
- Adam Wodzisławski: implementacja modelu A, sporządzenie dokumentacji na temat modelu A.

4. Eksploracja danych

Pod linkiem [1] w sekcji 'Data' znajdują się dane treningowe do pobrania. Jest to około 3,8 tys. zdjęć, na których można znaleźć ryby, podzielonych na 8 kategorii:

- ALB: Albacore tuna (Tuńczyk biały),

- BET: Bigeye tuna (Tuńczyk wielkooki),
- DOL: Dolphish/Mahi Mahi (Koryfena),
- LAN: Opah/Moonfish (Lampris),
- SHARK: różne gatunki rekinów,
- YFT: Yellowfin tuna (Tuńczyk żółtopłetwy),
- NoF: brak ryby,
- OTHER: inny gatunek niż wyżej wymienione.

Zdjęcia są zrobione na różnych łodziach, pod różnymi kątami, a ryby znajdują na nich w różnych miejscach, czasami są częściowo zasłonięte. Niektóre zdjęcia mają nienaturalne kolory, np. są zbyt zielone, a inne są niewyraźne (np. z powodu kropli wody na obiektywie kamery). Podejrzewamy, że zabiegi takie jak odszumianie albo balans bieli mogłyby zwiększyć skuteczność działania naszych modeli.

5. Prace implementacyjne (stan na 21.04.2022)

W ramach prac implementacyjnych stworzyliśmy jak dotąd cztery różne modele sieci neuronowych, które są umieszczone w repozytorium projektu [2]. Przygotowaliśmy następujące modele:

- Model korzystający z transfer learningu do użycia wytrenowanej sieci VGG16. Maksymalna skuteczność: około 88%,
- Model używający sieci Resnet50,
- Model używający z niewytrenowanej sieci VGG16,
- Własny model sieci CNN, składający się z dwudziestu warstw.

Korzystając z tego, że niektóre modele zostały utworzone z wykorzystaniem wytrenowanych sieci neuronowych, a inne z niewytrenowanych, będziemy mogli porównać skuteczności, jakie można otrzymać używając każdego z tych podejść. Podejrzewamy, że sprawność sieci można jeszcze zwiększyć lepiej dobierając hiperparametry oraz za pomocą lepszego przystosowania danych wejściowych.

References

- [1] `www.kaggle.com/c/the-nature-conservancy-fisheries-monitoring`
- [2] *`https://github.com/JakubGazewski/WTUM72022`*