

AI & COMPUTER VISION

OPENCV, TRANSFER LEARNING, GAN

ZADANIE 1: OPENCV I LICZNIK PTAKÓW

Do folderu z materiałami dołączono folder z miniaturami obrazów ptaków.



Zadaniem jest policzenie ptaków na obrazie i zwrócenie liczby. Na przykład dla ostatniego obrazu powyżej powinno być 2.

Jak rozwiązać ten problem?

Możesz spróbować następujących kroków:

- Wczytaj obrazy. Możesz spróbować znaleźć funkcję, która iteruje przez wszystkie obrazy w danym folderze.
- Konwertuj je na skale szarości (są to obrazy kolorowe, chociaż wyglądają na szare!).
- Pobaw się filtrami, aby spróbować znaleźć optymalne progi dla wykrywania obiektów. Idealnie byłoby, gdyby tło było białe, a ptaki czarnymi plamami pikseli. Być może znajdziesz lepsze filtry adaptujące się do jasności obrazu.
- Napisz funkcję, która liczy liczbę czarnych plam na obrazie.
- Użyj funkcji do liczenia ptaków na wszystkich obrazach. Wydrukuj listę: nazwa obrazu + liczba ptaków na tym obrazie.

Jeśli masz problemy z tym zadaniem, możesz użyć ChatGPT lub Google Gemini, lub innych narzędzi do pomocy.

ZADANIE 2: TRANSFER LEARNING DLA PSÓW I KOTÓW

Poprzednio klasyfikowaliśmy psy i koty za pomocą sieci konwolucyjnej.

Pytanie jest następujące: czy można wykorzystać jakiś pre-trained model, dotrenować go, by osiągnąć lepsze rozpoznawanie psów i kotów z obrazków?

Jaki jest plan działania:

- Wybierz jakiś model (np. dostępnych w obrębie keras: <https://keras.io/api/applications/>)
- Załaduj model i odpowiednio go zmodyfikuj, doklejając do niego warstwy.
- Przetrenuj tak stworzony model na bazie danych z psami i kotami.
- Porównaj wyniki z wynikami z poprzednich laboratoriów.



ZADANIE 3: GENEROWANIE OBRAZKÓW

Na wykładzie były prezentowane programy gan01.py, ..., gan07.py. Zostały stworzone na podstawie samouczka, do którego warto zajrzeć

<https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-an-mnist-handwritten-digits-from-scratch-in-keras/>

Wygenerujemy obrazy pisanych ręcznie cyfr (MNIST).

- a) Ściągnij pliki pythonowe z wykładu i przejrzyj je. Możesz je uruchomić.
- b) Szczególnie ważny jest plik `gan07.py`. Uruchom go ustawiając liczbę epok na 3 (trenowanie powinno zająć około 10 minut).
- c) Sprawdź jak wyglądają wygenerowane obrazki po każdej epoce (w `generated_plot_eXXX.png`).
- d) Czy model GAN da się zapisywać (wagi sieci w pliku)? Czy taki zapisany model da się potem załadować i dotrenować na kolejnych próbkach danych? Spróbuj znaleźć sposób na to.
- e) Wytrenuj model na większej liczbie epok (np. kilka godzin trenowania). Jak będą wyglądały wygenerowane obrazki? Pokaż je.

Uwaga! Jeśli sieć po paru epokach nie przynosi wyraźnej poprawy, spróbuj zamiast korzystania z załączonych plików napisać program na nowo, korzystając też ze źródeł internetowych.