

## PROJEKT Z PRZEDMIOTU UMIR – 2024Z

Projekt ma na celu praktyczne zaznajomienie się z **metodami klasyfikacji** wykorzystującymi **głębokie sieci neuronowe** oraz **technikami wizualizacji** obszarów uwagi i wewnętrznych warstw sieci. Poniżej znajdują się szczegóły zadań klasyfikacji oraz wizualizacji. Każdy zespół otrzyma **indywidualny zestaw bazy danych** oraz **głębokiej sieci neuronowej**, które należy wykorzystać w projekcie.

Obowiązują **dwuosobowe** lub **trzyosobowe** zespoły projektowe.

Każdy zespół rozwiąże 3 zadania:

### 1. Uczenie klasyfikatora

- Zastosować wstępnie wytrenowaną sieć do **uczenia tylko części klasyfikującej** (ostatnie warstwy o połączeniach kompletnych).
- Zanalizować wyniki klasyfikacji.
- Zastąpić część klasyfikującą sieci przez **SVM** dla jądra liniowego, kwadratowego i RBF.
- Zanalizować wyniki klasyfikacji. W szczególności, zbadać efekt dopuszczenia błędnych klasyfikacji, porównać z wynikami 1a.

### 2. Uczenie sieci głębokiej

- Przeprowadzić uczenie **ostatniej warstwy splotowej** wraz z częścią klasyfikującą.
- Wytrenować **całą sieć** dla zadanych danych.
- Uprościć strukturę sieci** wytrenowanej w zadaniu 2c (np. poprzez usunięcie jednej lub więcej końcowych warstw splotowych, usunięcie warstw regularyzujących itp.) i ponowić uczenie.
- Zanalizować wyniki 2 abc.

### 3. Wizualizacja

- Dokonać **wizualnej analizy błędnych klasyfikacji** dla wybranej wersji sieci z zadania 1 lub 2 pod kątem znalezienia potencjalnych przyczyn oraz detekcji błędnych oznaczeń w wykorzystywanej bazie danych.
- Dokonać **wizualizacji obszarów uwagi** sieci wytrenowanych w zadaniu 1 oraz 2 z wykorzystaniem metod Class Activation Map (CAM).
- Dokonać **wizualizacji aktywacji** wewnętrznych warstw sieci z wykorzystaniem techniki DeepDream.
- Przygotować **własny zbiór zdjęć** dla wybranych kategorii (kilkadziesiąt zdjęć) oraz **zanalizować wyniki klasyfikacji** dla wszystkich wytrenowanych sieci z punktów 1 oraz 2. Do przygotowania zbioru można wykorzystać modele generatywne AI (np. Midjourney)

W każdym przypadku należy **podzielić dane** na uczące, walidacyjne i testujące oraz **zwielokrotnić zbiór danych** z wykorzystaniem technik data augmentation. Odpowiednie oprogramowanie sieci neuronowych proszę wybrać samodzielnie.

### Bazy danych:

1. Analiza **kondycji paneli słonecznych** na podstawie zdjęć, ponad 2500 zdjęć, łączny rozmiar bazy - 1.3 GB:  
<https://www.kaggle.com/datasets/hemanthsai7/solar-panel-dust-detection>
  - a. Dodatkowy zbiór danych zawierający kilkaset zdjęć paneli z zanieczyszczeniami oraz uszkodzeniami:  
<https://www.kaggle.com/datasets/pythonafroz/solar-panel-images>
2. Wykrywanie **chorób roślin** na podstawie zdjęć liści (inteligentne rolnictwo), zbiór zawiera 39 klas (przed przystąpieniem do realizacji zadań należy je zagregować, część klas można odrzucić), ponad 61 000 zdjęć, łączny rozmiar bazy – ok. 1 GB: <https://data.mendeley.com/datasets/tywbtsjrjv/1>
3. Wykrywanie chorób skóry na podstawie **zdjęć dermatoskopowych** (informatyka medyczna, onkologia), zbiór zawiera siedem klas, które należy zagregować do trzech klas: A={bkl, df, vasc}, B={nv}, C={akiec, bcc, mel}. Baza zawiera 10015 obrazków i ma rozmiar ok. 3 GB:  
<https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000>

### Głębokie sieci neuronowe:

1. AlexNet
2. SqueezeNet
3. VGG-16
4. VGG
5. MobileNetV1
6. MobileNetV3
7. ResNet-50
8. ResNet-101
9. Inception

### Wskazówki:

1. Przykładowe rozwiązania umożliwiające wykonanie projektu: Tensorflow + Keras, PyTorch, Caffe, Deep Learning Toolbox for MATLAB, MatConvNet, R.

2. Wyniki powinny być przedstawione w postaci błędów klasyfikacji (top-1, top-5), skuteczności klasyfikacji, lub macierzy pomyłek.
3. Istnieje możliwość przeprowadzenia klasyfikacji dla części klas dostępnych w ramach wykorzystywanego zbioru zdjęć oraz połączenia kilku klas w jedną.

#### Rozwiązanie powinno zawierać:

1. Analizę działania każdej z wersji zadania, porównanie wyników uzyskanych dla różnych wariantów, trzy rodzaje wizualizacji oraz wnioski z tych działań wyływające.
2. Ocenę – z punktu widzenia zaawansowanego użytkownika – narzędzi programistycznych zastosowanych przy rozwiązywaniu problemów.

#### Zasady zaliczenia projektu:

W ramach zaliczenia niezbędne jest złożenie raportu oraz prezentacja ustna pod koniec semestru:

1. **Raport końcowy** (30 pkt.) – raport obejmujący opis wszystkich eksperymentów, wyniki i wnioski, proszę nie przekraczać 12 stron A4. **Termin oddania: dzień poprzedzający termin egzaminu z przedmiotu, do którego chcą przystąpić członkowie zespołu. Przesłanie raportu w terminie jest warunkiem udziału w egzaminie z przedmiotu.**

Osoby chętne mogą przygotować raport w formie przypominającej artykuł naukowy z wykorzystaniem pakietu LaTeX do składania dokumentów. **Jest to wygodne narzędzie na przykład do przygotowania pracy dyplomowej i warto spróbować się z nim zapoznać.**

2. **Prezentacja ustna** (10 pkt.) – prezentacja prac, każda z osób w zespole powinna zaprezentować część materiału, dodatkowo prezentacja powinna zawierać opinię o środowisku pracy. Czas na prezentację: 15 minut. **Prezentacje odbędą się poprzez platformę Microsoft Teams. Termin: 14-15 tydzień semestru** lub inny, ustalony wcześniej termin.

*Raporty w postaci plików PDF + kody (na przykład w postaci repozytorium typu Github) należy przesłać w postaci elektronicznej prowadzącym projekt do godziny 23:59 określonego dnia.*

**Prowadzący:**

**Konsultacje z prowadzącym odbywają się z wykorzystaniem poczty elektronicznej, platformy Microsoft Teams lub stacjonarnie (po wcześniejszym kontakcie mailowym)**

<b>Imię i nazwisko</b>	<b>Email</b>
dr inż. Konrad Ciecierski	konrad.ciecierski@pw.edu.pl
dr inż. Ewelina Bartuzi-Trokielewicz	ewelina.trokielewicz@pw.edu.pl
mgr inż. Michał Hałoń	michal.halon@pw.edu.pl