LABORATORIUM 5 – Wdrożenie i monitorowanie

W ostatnim kamieniu milowym stworzyliśmy skrypt, który wykorzystując zapisany model *model.pth* analizuje przekazane mu w folderze *image\_folder* zdjęcia spoza dotychczasowych zbiorów.

# run\_model.py

import torch

from torchvision import transforms

from PIL import Image

import matplotlib.pyplot as plt

import os

from model import ASLClassifier

from evaluate import get\_class\_names

# === Konfiguracja ===

device = torch.device("cuda" if torch.cuda.is\_available() else "cpu")

model\_path = "model.pth"          # zapisany model

image\_folder = "image\_folder"     # folder z nowymi zdjęciami

# === Przygotowanie modelu ===

num\_classes = 36

model = ASLClassifier(num\_classes=num\_classes)

model.load\_state\_dict(torch.load(model\_path, map\_location=device))

model.to(device)

model.eval()

# === Transformacja ===

transform = transforms.Compose([

    transforms.Resize((224, 224)),

    transforms.ToTensor(),

    transforms.Normalize(mean=[0.5], std=[0.5])

])

# === Pobranie klas ===

class\_names = get\_class\_names()

# === Wczytanie i analiza wielu zdjęć ===

image\_files = [f for f in os.listdir(image\_folder) if f.endswith(".jpg") or f.endswith(".png") or f.endswith(".jpeg")]

if not image\_files:

    print(f"Brak plików .jpg/.png w folderze: {image\_folder}")

else:

    for filename in image\_files:

        image\_path = os.path.join(image\_folder, filename)

        img = Image.open(image\_path).convert("RGB")

        input\_tensor = transform(img).unsqueeze(0).to(device)

        with torch.no\_grad():

            outputs = model(input\_tensor)

            \_, predicted = torch.max(outputs, 1)

        predicted\_label = class\_names[predicted.item()]

        # === Wyświetlenie ===

        plt.imshow(img)

        plt.title(f"{filename}\nPredykcja: {predicted\_label}")

        plt.axis("off")

        plt.show()

Poniższe sprawdzenia obejmują zarówno zdjęcia nieobrobione (z tłem w postaci ciemnej maty/ekranu), jak i przerobione, żeby zbadać czy w idealnych warunkach model nie gubi dokładności w stosunku do symboli na „surowych” zdjęciach.

Poprawnie zidentyfikowane symbole:

A hand with fingers raised

AI-generated content may be incorrect.A hand with a finger raised

AI-generated content may be incorrect.

A hand making a circle

AI-generated content may be incorrect. A hand with two fingers up

AI-generated content may be incorrect.

A hand with fingers in front of a computer

AI-generated content may be incorrect. A hand with a computer screen

AI-generated content may be incorrect.

A hand with a thumb up in front of a computer screen

AI-generated content may be incorrect.

A hand with a thumb up

AI-generated content may be incorrect.

W niektórych przypadkach, np. gdy znaki mogą przypominać inne wymazane zostało tło, a wtedy faktycznie model zadziałał poprawnie

NIE OK, ALE WIADOMO CZEMU:

A hand with an object gesture

AI-generated content may be incorrect. A hand making a hand gesture

AI-generated content may be incorrect.

A hand with a number on it

AI-generated content may be incorrect. A hand making a gesture with fingers

AI-generated content may be incorrect.

- „i” potraktowane jako „z”, ponieważ model twierdzi, że mały palec jest wskazującym,

- „h” zamiast „g”, ponieważ widzi dwa palce jako jeden,

- „w” zamiast „v”, ponieważ dwa palce blisko siebie, a jako „6”, bo są to bardzo podobne znaki.

NIE OK – HALUCYNACJE:

A hand making a peace sign

AI-generated content may be incorrect.A hand with a finger raised

AI-generated content may be incorrect.

Przez to, że model jest wrażliwy na oświetlenie „j” bardzo często występowało jako błędna detekcja.

„z” posiada mniej więcej podobny kształt, ale jest duża rozbieżność między pozycją i ustawieniem palców.