

# Uczenie Nienadzorowane - projekt

Piotr Łuczak PhD Eng.  
piotr.luczak.1@p.lodz.pl

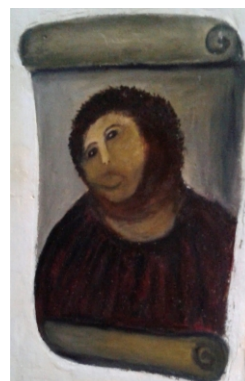
Zima 2025

## 1 Opis

Celem projektu jest zbudowanie modelu uzupełniającego (inpainting) uszkodzenia dzieł sztuki pochodzących z platformy WikiArt. W oparciu o metody uczenia nienadzorowanego (unsupervised) i samonadzorowanego (self-supervised) należy zbudować system pozwalający na grupowanie obrazów na podstawie ich podobieństwa oraz uzupełnianie ich symulowanych uszkodzeń, tak aby uniknąć sytuacji takich jak renowacja Ecce homo<sup>1</sup> przedstawiona na rysunkach 1 i 2.



Rysunek 1: Pierwotny źródło, dostęp 2024-09-02



Rysunek 2: "Odnowiony" źródło, dostęp 2024-09-02

Budowę systemu należy rozpocząć od budowy generatora uszkodzeń obrazów (białych plam / masek). Podstawowa wersja uszkodzania obrazów polega na losowym wycinaniu / maskowaniu kwadratowych fragmentów obrazu, zajmujących nie więcej niż  $\frac{1}{16}$  obrazu. Rozszerzona wersja powinna pozwalać na generowanie uszkodzeń o nieregularnych kształtach.

Kolejnym krokiem powinno być zbudowanie zredukowanej reprezentacji obrazów ze zbioru treningowego a następnie jej klasteryzacji w celu znalezienia naturalnego grupowania stylów. Grupowanie należy następnie porównać z dostępnymi w zbiorze etykietami stylu i autora.

Następnie, dla każdej z grup należy nauczyć osobny model uzupełniający uszkodzenia. Alternatywnie, możliwe jest zbudowanie jednego modelu który na wejściu przyjmowałby uszkodzony obraz oraz identyfikator grupy.

Końcowym etapem będzie zbudowanie modelu zwiększającego rozdzielczość obrazów.

---

<sup>1</sup>Więcej informacji na Wikipedii

## 2 Szczegółowe wymagania

### 2.1 Organizacyjne

Rozwiązania mogą być przygotowywane w grupach maksymalnie 3 osobowych.

Prezentacje śródsesemestralne odbędą się zgodnie z planem zajęć w dniu **2025-11-28 (tyg. 8)** w godzinach **16:15 - 20:00** w sali **E106**, chyba że dojdzie do modyfikacji terminarza uczelni.

Prezentacje końcowe odbędą się zgodnie z planem zajęć w dniu **2026-01-30 (tyg. 15)** w godzinach **16:15 - 20:00** w sali **E106**, chyba że dojdzie do modyfikacji terminarza uczelni.

Możliwe jest również zaproponowanie własnego tematu, pod warunkiem że proponowane rozwiązanie będzie wykorzystywało metody wymienione w sekcji 2.3.

### 2.2 Zbiór danych

Podstawowy zbiór obrazów o wymiarach 256x256 *Artificio/WikiArt\_Full*<sup>2</sup> można pobrać z platformy Hugging Face.

Do uczenia modelu zwiększającego rozdzielczość, jako źródło obrazów o wyższej rozdzielczości można wykorzystać zbiór 81 444 obrazów *huggan/wikiart*<sup>3</sup> lub zbiór 195 394 obrazów z platformy Internet Archive<sup>4</sup>.

Wszystkie wykorzystywane zbiory danych należy podzielić na podzbiory uczące, walidacyjne i testowe.

### 2.3 Wykorzystywane metody

Projekt powinien wykorzystywać przedstawione na zajęciach:

- klasyczne metody klasteryzacji,
- neuronowe, samonadzorowane metody budowy reprezentacji (np. autoenkodery),
- reprezentacji i wizualizacji przestrzeni wysokowymiarowych (np. UMAP),
- neuronowe, samonadzorowane metody uzupełniania obrazów (inpainting),
- neuronowe, samonadzorowane metody zwiększania rozdzielczości obrazów (superresolution).

---

<sup>2</sup>[https://huggingface.co/datasets/Artificio/WikiArt\\_Full](https://huggingface.co/datasets/Artificio/WikiArt_Full)

<sup>3</sup><https://huggingface.co/datasets/huggan/wikiart>

<sup>4</sup>[https://archive.org/details/WikiArt\\_dataset](https://archive.org/details/WikiArt_dataset)

## 2.4 Dozwolone narzędzia

Rozwiązanie musi być napisane w języku Python 3.11 lub nowszym i może wykorzystywać:

- PyTorch lub TensorFlow 2
- bibliotekę Hugging Face datasets i/lub CLI,
- NumPy i/lub CuPy
- Gotowe implementacje UMAP <sup>5</sup>, TSNE, DBSCAN, GaussianMixture, SpectralClustering i innych metod klasteryzacji i redukcji wymiarowości dostępnych w bibliotece Scikit-Learn.
- scikit-image,
- imageio,
- platformę do śledzenia eksperymentów uczenia maszynowego Comet ML,
- Dowolną bibliotekę GUI. Szczególnie warte rozważenia są: Dash i Streamlit.

## 3 Kryteria oceniania

Projekt oceniany będzie na podstawie 20 minutowych prezentacji końcowych. Prezentacja końcowa powinna zawierać demonstracje działania modelu na żywo na losowo wybranym obrazie ze zbioru testowego oraz losowym jego uszkodzeniu. Demonstracja powinna być oparta o autorskie GUI, umożliwiające wybór rodzaju operacji, a także tam gdzie to możliwe jej implementacji.

Warunkiem koniecznym do otrzymania oceny jest dostarczenie Jupyter Notebooka prezentującego kompletny proces generacji (symulowanego uszkodzenia obrazu) i przetwarzania obrazu (przypisanie do stylu, uzupełnienie uszkodzenia, zwiększenie rozdzielczości) wraz z kompletnym kodem źródłowym zaimplementowanych rozwiązań. Kody źródłowe muszą zostać dostarczone w **formie jednego pliku .zip** (wysłane na adres email podany w tej instrukcji) **najpóźniej pierwszego dnia sesji tj. 2026-02-04**.

### 3.1 Szczegółowe kryteria ocen

Zbudowanie reprezentacji przy pomocy autoenkodera oraz jej klasteryzacja + uzupełnianie prostych uszkodzeń obrazów → 3

Rozszerzenie systemu o moduł neuronowego zwiększania rozdzielczości → 4

Rozszerzenie systemu o uzupełnianie nieregularnych uszkodzeń obrazów → 5

---

<sup>5</sup>Prezentacja algorytmu w wykonaniu jego autora: <https://www.youtube.com/watch?v=nq6iPZVUxZU>