

ROZPOZNAWANIE STYLU OBRAZÓW

Sebastian Borukał¹, Jakub Buła², Jakub Smakowski³

1. nr albumu: [REDACTED] e-mail: [REDACTED]
2. nr albumu: [REDACTED] e-mail: [REDACTED]
3. nr albumu: [REDACTED] e-mail: [REDACTED]

Streszczenie: Celem projektu było opracowanie i implementacja algorytmu do automatycznego rozpoznawania stylu obrazów na podstawie ich cech wizualnych, bez wykorzystania głębokich sieci neuronowych. W badaniach wykorzystano zbiór danych WikiArt, z którego wyekstrahowano szereg cech opisujących kolor, teksturę oraz kształty (m.in. HOG, GLCM, statystyki krawędzi). Jako klasyfikator zastosowano maszynę wektorów nośnych (SVM). Przeprowadzono eksperymenty mające na celu optymalizację doboru cech metodą RFE oraz analizę wpływu redukcji rzadkich klas. Ostateczny model osiągnął skuteczność Top-1 na poziomie 37,71% oraz Top-5 na poziomie 79,7% dla pełnego zbioru klas.

Słowa kluczowe: klasyfikacja obrazów, ekstrakcja cech, RFE

1. WSTĘP

Problem automatycznej klasyfikacji dzieł sztuki jest złożony; łączy on komputerowe przetwarzanie obrazu z wiedzą z zakresu historii sztuki. W niniejszym projekcie podjęto próbę stworzenia systemu, który na podstawie analizy cyfrowej obrazu przypisze go do odpowiedniego stylu artystycznego, takiego jak na przykład impresjonizm, kubizm czy surrealizm. Problem ten jest istotny z punktu widzenia między innymi cyfryzacji zbiorów muzealnych oraz tworzenia inteligentnych wyszukiwarek multimedialnych.

Motywacją do podjęcia tematu była chęć sprawdzenia, na ile klasyczne metody przetwarzania obrazów i uczenia maszynowego są w stanie uchwycić subiektywne cechy definiujące styl malarski. Celem projektu było opracowanie algorytmu, który automatycznie analizuje i ekstrahuje charakterystyczne cechy stylu, a następnie porównuje je z bazą wiedzy. Kluczowym ograniczeniem technologicznym narzuconym w projekcie był zakaz wykorzystywania sieci neuronowych, co wymusiło skupienie się na inżynierii cech.

W literaturze problem klasyfikacji stylu malarskiego był szeroko analizowany m.in. przez Saleha i Elgammala (2015) [1]. Badania te wykazały, że **pojedyncze deskryptory wizualne** (jak Classemes czy PiCoDes) osiągają na zbiorze ograniczoną skuteczność rzędu **22,6%**. Dopiero **zaawansowana fuzja wielu cech** w połączeniu z uczeniem metryk pozwalała w tamtym okresie uzyskać wyniki na poziomie **39–41%** [1]. Choć współczesne **architektury głębokie**, takie jak ResNet-50, przesunęły ten wynik powyżej **66%** (Zhao i in., 2021 [2]), celem niniejszego projektu jest zbadanie potencjału klasycznej inżynierii cech i sprawdzenie, jak bardzo można zbliżyć się do granicy wyznaczonej przez literaturę bez użycia sieci neuronowych.

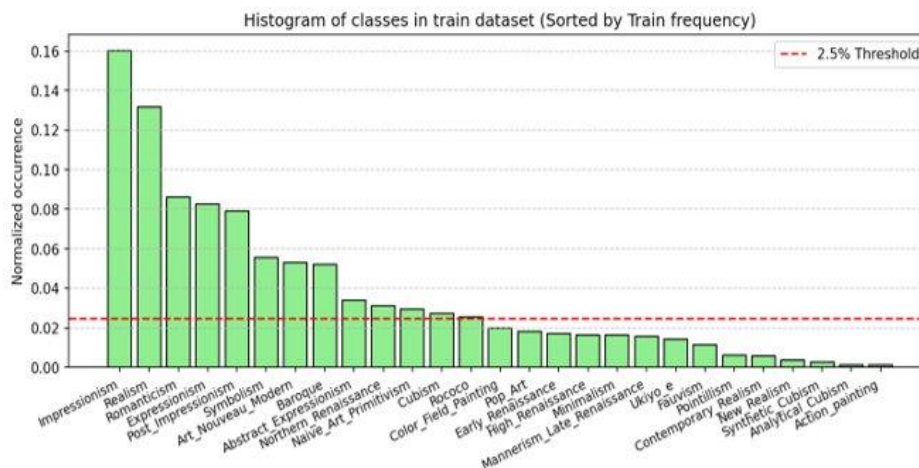
2. MATERIAŁY I METODY

2.1. Dane

W projekcie wykorzystano zbiór danych **steubk/wikiart** (dostępny w serwisie Kaggle) [3], który jest jednym z największych dostępnych zbiorów cyfrowych dzieł sztuki. Zbiór charakteryzuje się następującą specyfiką:

- **Liczba obrazów:** około 80 020 plików.
- **Liczba klas:** 27 unikalnych stylów artystycznych.
- **Format:** pliki JPG o zmiennej rozdzielczości.
- **Rozkład klas:** bardzo nierównomierny, co stanowiło wyzwanie podczas trenowania modelu.

Dominującymi klasami w zbiorze są impresjonizm oraz realizm, podczas gdy niektóre style (np. action painting) reprezentowane są przez niewielką liczbę przykładów (por. Rysunek 1).



Rys. 1. Rozkład licznosci klas w zbiorze treningowym.

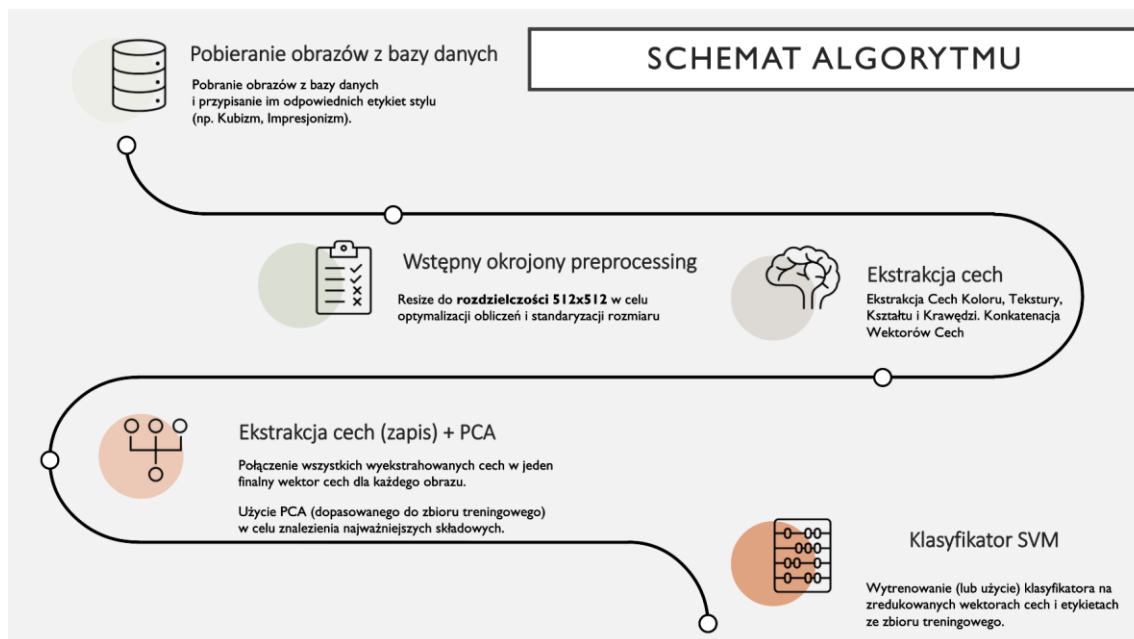
W ramach przygotowania danych (preprocessing) wykonano ujednolicanie rozdzielczości wszystkich obrazów do **512x512 pikseli**. Zabieg ten miał na celu standaryzację danych wejściowych oraz optymalizację czasu obliczeń.

2.2. Metody

Opracowana metoda opiera się na klasycznym schemacie przetwarzania danych w uczeniu maszynowym (por. Rysunek 2). Proces ten składa się z następujących etapów:

1. **Przygotowanie zbioru danych:** Pobranie obrazów oraz przypisanie im etykiet definiujących styl malarski.

2. **Wstępne przetwarzanie:** Skalowanie obrazu.
3. **Ekstrakcja cech:** Obliczenie numerycznych deskryptorów opisujących zawartość wizualną.
4. **Fuzja cech:** Połączenie wektorów cech w jeden długi wektor dla każdego obrazu.
5. **Redukcja wymiarowości:** Zastosowanie analizy głównych składowych (PCA) w celu znalezienia najważniejszych cech i zmniejszenia szumu.
6. **Klasyfikacja:** Wytrenowanie klasyfikatora SVM (Support Vector Machine).



Rys. 2. Ogólny schemat działania opracowanej metody.

Kluczowym elementem metody była ekstrakcja cech. W toku eksperymentów przetestowano szeroki wachlarz filtrów, podzielonych na trzy fazy implementacji:

Faza 1 - Filtry początkowe: [4] [5] [6] [7] [8]

- **LBP (Local Binary Patterns):** Histogram jednolitych wzorców binarnych (analiza mikrotekstur).
- **Symetria strukturalna (SSIM):** Obliczanie symetrii pionowej i poziomej obrazu.
- **Entropia obrazu (Shannon Entropy):** Miara złożoności informacji zawartej w histogramie.
- **Widmo mocy radialnej (Radial Power Spectrum):** Rozkład energii obrazu w zależności od promienia w dziedzinie częstotliwości.
- **Momenty kolorów (Color Moments):** Średnia, wariancja i skośność dla każdego kanału RGB.
- **Barwność i Nasycenie (Colorfulness & Saturation):** Metryka barwności Haslera-Süsstrunka oraz średnie nasycenie z modelu HSV.
- **Harmonia i kontrast kolorów:** Kontrast między kanałami oraz nakładanie się histogramów kanałów (RGB).
- **Statystyki konturów:** Liczba obiektów, średnia długość, pole powierzchni i proporcje obwiedni.
- **Cechy Haralicka (GLCM):** Statystyki macierzy współwystępowania.
- **Filtry Gabora:** Odpowiedzi filtrów teksturowych dla różnych częstotliwości i kątów orientacji.

- **HOG (Histogram of Oriented Gradients):** Histogram zorientowanych gradientów (używany do detekcji kształtów).
- **Podsumowanie widma radialnego (Spectral Summary):** Nachylenie widmowe oraz stosunek energii niskich do wysokich częstotliwości.
- **Wymiar fraktalny (Fractal Dimension):** Estymacja złożoności geometrycznej.

Faza 2 - Filtry wspomagające: [9]

- **Statystyki krawędzi (Edge Statistics):** Wykorzystanie detektora Canny i Sobela do obliczenia gęstości i siły krawędzi.
- **Histogram w przestrzeni LAB (LAB Histogram):** Trójwymiarowy histogram w przestrzeni barw CIELAB.
- **Analiza teksturowa falkami (Wavelet Texture):** Określenie ziarnistości i gładkości obrazu.
- **Głębina ostrości (Depth of Field Proxy):** Heurystyka symulująca rozmycie tła.
- **Reguła trójkądnego podziału (Rule of Thirds Stats):** Porównanie statystyk centrum obrazu z jego brzegami.

Faza 3 - Filtry uzupełniające:

- **Bag of Visual Words (BoVW):** Słownik wizualny częstości występowania wzorców.
- **Autokorelogram kolorów:** Łączenie informacji o kolorze z układem przestrzennym.

W ostatecznej wersji algorytmu zastosowano metodę **RFE (Recursive Feature Elimination)**, aby wybrać zestaw cech dający najwyższą skuteczność przy optymalnej wymiarowości. Algorytm ten iteracyjnie usuwał najmniej istotne cechy, pozostawiając jedynie te, które wniosły największą wartość dyskryminacyjną dla klasyfikatora SVM.

W rezultacie przeprowadzonej selekcji, ostateczny wektor cech wykorzystany w finalnym modelu składał się z 13 grup deskryptorów:

1. **Statystyki krawędzi (Edge Statistics).**
2. **Cechy Haralicka (GLCM).**
3. **HOG (Histogram of Oriented Gradients).**
4. **Analiza teksturowa falkami (Wavelet Texture).**
5. **LBP (Local Binary Patterns).**
6. **Entropia obrazu (Shannon Entropy).**
7. **Barwność i Nasycenie (Colorfulness & Saturation).**
8. **Harmonia i kontrast kolorów.**
9. **Momenty kolorów (Color Moments).**
10. **Podsumowanie widma radialnego (Spectral Summary).**
11. **Filtry Gabora.**
12. **Barwność i nasycenie z modelu HSV.**
13. **Głębina ostrości (Depth of Field).**

Pozostałe testowane filtry zostały odrzucone przez algorytm RFE jako wnoszące zbyt mało informacji w stosunku do kosztu obliczeniowego.

2.3. Technologie i narzędzia

Projekt zrealizowano w języku **Python**. Z wykorzystaniem następujących bibliotek: **cv2** (OpenCV), **skimage** (scikit-image), **PIL** (Pillow) - do wczytywania, skalowania i ekstrakcji cech z obrazów; **numpy**, **pandas**, **scipy** - do operacji macierzowych i przechowywania wektorów cech; **sklearn** (scikit-learn) - implementacja algorytmów SVM, PCA, RFE oraz metryk oceny; **kagglehub** (pobieranie danych); **datasets**, **comet_ml** (śledzenie wyników eksperymentów); **matplotlib**: wizualizacja wyników i wykresów; **joblib**, **pickle**: serializacja i zapisywanie wytrenowanych modeli; **tqdm**: wizualizacja postępu obliczeń; **Concurrent**: równoleglenie procesów ekstrakcji cech oraz **functools**, **time** i **random**.

Kod projektu został zoptymalizowany pod kątem wykorzystania na standardowych komputerach z systemem Windows 11, a procedurę selekcji cech metodą RFE przeprowadzono na podzbiore stanowiącym 10% danych treningowych w celu zredukowania czasu obliczeń.

Link do repozytorium GitHub: https://github.com/Se-Boruk/AoC_project

2.4. Eksperymenty

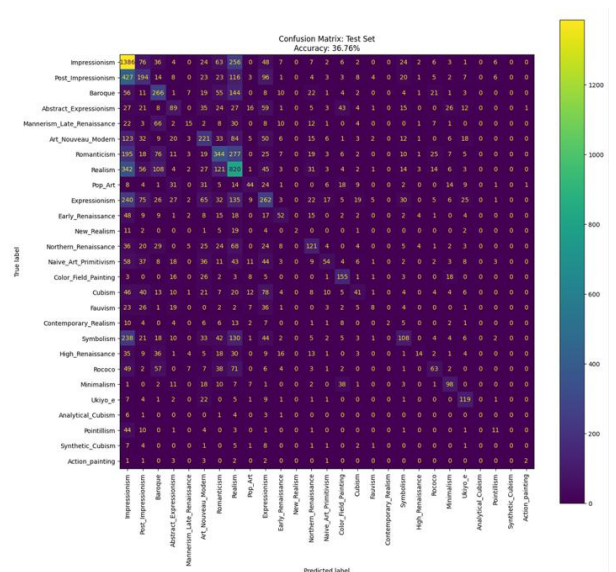
Eksperymenty przeprowadzono w kilku etapach, oceniając wpływ dodawania nowych filtrów oraz redukcji liczby klas. Zbiór danych został podzielony na część treningową i testową. Jako miary jakości przyjęto:

- **Accuracy (Dokładność):** Procent poprawnych klasyfikacji.
- **Top-N Accuracy (Top-1, Top-3, Top-5):** Sprawdzenie, czy poprawna etykieta znajduje się wśród N najbardziej prawdopodobnych wskazań modelu.
- **Macierz pomyłek (Confusion Matrix):** Analiza błędów między konkretnymi stylami.

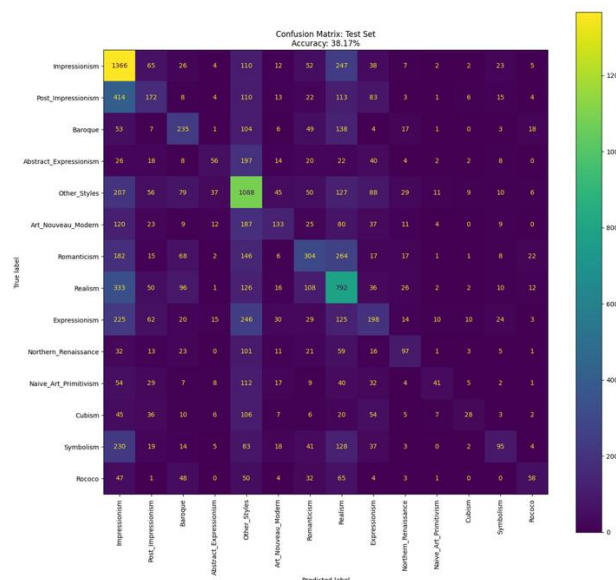
3. REZULTATY

W tej sekcji przedstawiono wyniki w zależności od zastosowanego modelu i zestawu cech.

Etap 1: Model bazowy. Początkowy zestaw filtrów dla wszystkich 27 klas pozwolił osiągnąć dokładność (Accuracy) na poziomie **36,76%**. Był to punkt odniesienia dla dalszych prac. Przeprowadzono również eksperyment polegający na zgrupowaniu rzadkich klas (występujących rzadziej niż 2,5% w zbiorze) do jednej kategorii "Other_Styles". Zabieg ten pozwolił podnieść dokładność (Accuracy) do **38,17%**. Poniżej przedstawiono macierz pomyłek dla modelu na pełnym zbiorze klas (por. Rysunek 3) i zredukowanym (por. Rysunek 4). Macierz ukazuje, które style są najczęściej mylone ze sobą.



Rys. 3. Macierz pomyłek – WSZYSTKIE KLASY.

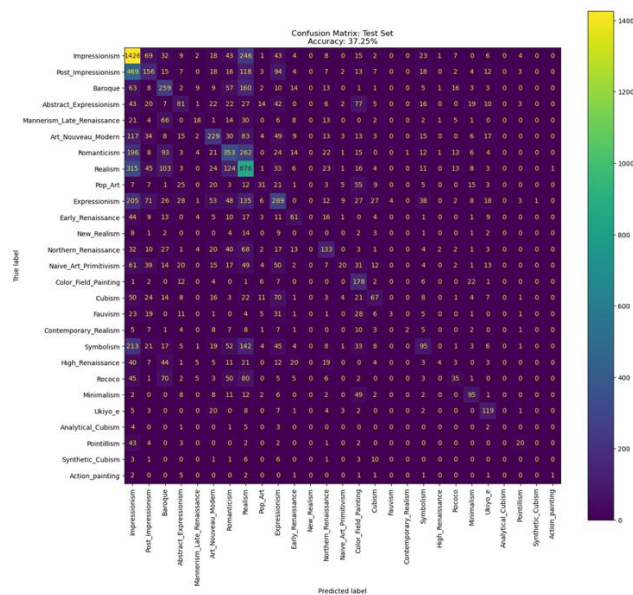


Rys. 4. Macierz pomyłek – REDUKCJA LICZBY KLASY.

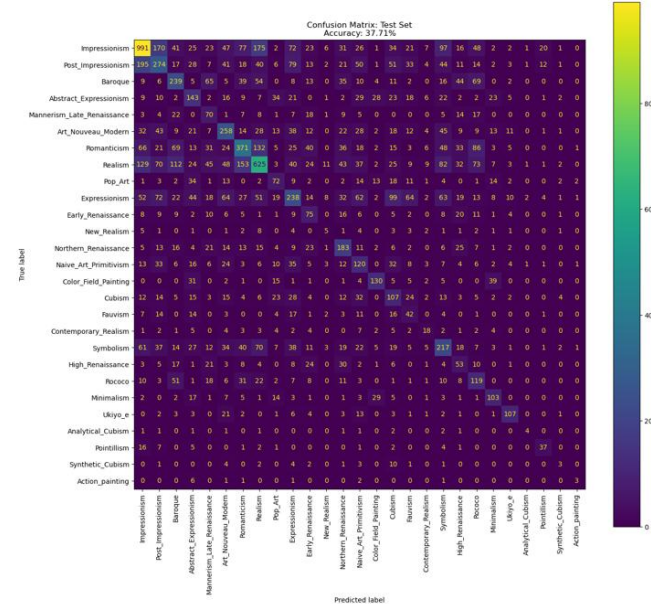
Etap 2: Rozszerzony zestaw cech. Po dodaniu filtrów wspomagających wynik dla pełnego zbioru klas wzrósł do **37,25%**. Poprawa o około 0,5 p.p. wskazała na użyteczność nowych deskryptorów. Dla zredukowanej liczby klas wzrósł do **39,42%**, czyli o 1,25 p.p. Poniżej przedstawiono macierz pomyłek dla modelu na pełnym

zbiorze klas (por. Rysunek 5) i zredukowanym (por. Rysunek 6).

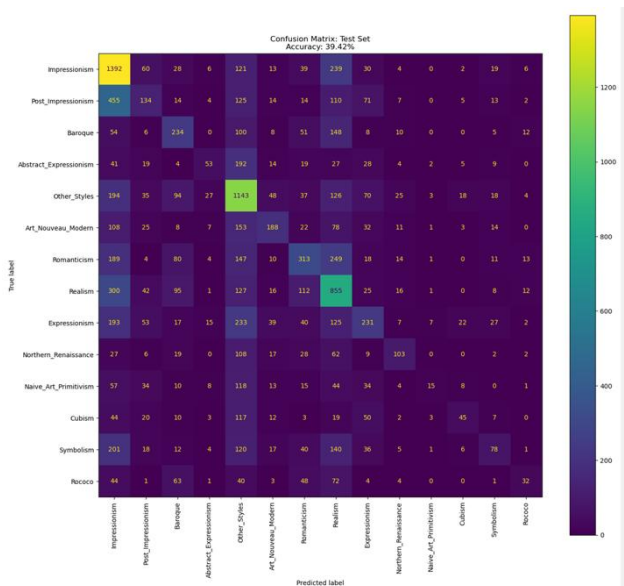
pomyłek dla modelu na pełnym zbiorze klas (por. Rysunek 7) i zredukowanym (por. Rysunek 8).



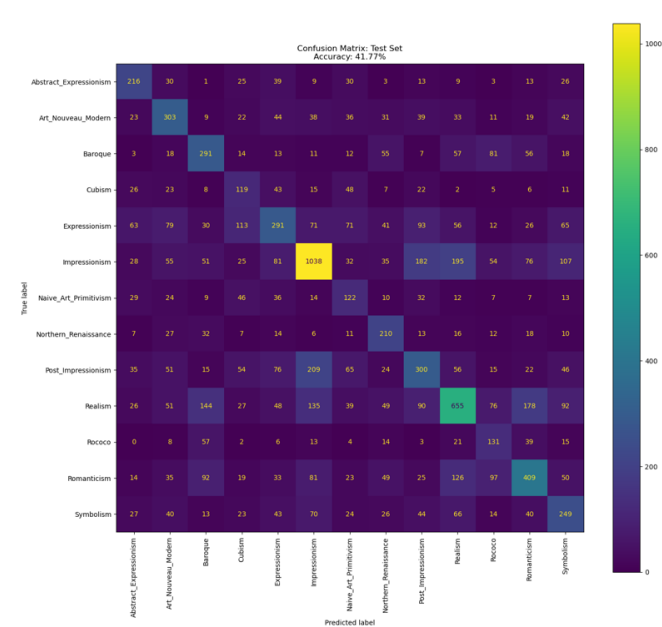
Rys. 5. Macierz pomyłek – WSZYSTKIE KLASY.



Rys. 7. Macierz pomyłek – WSZYSTKIE KLASY.



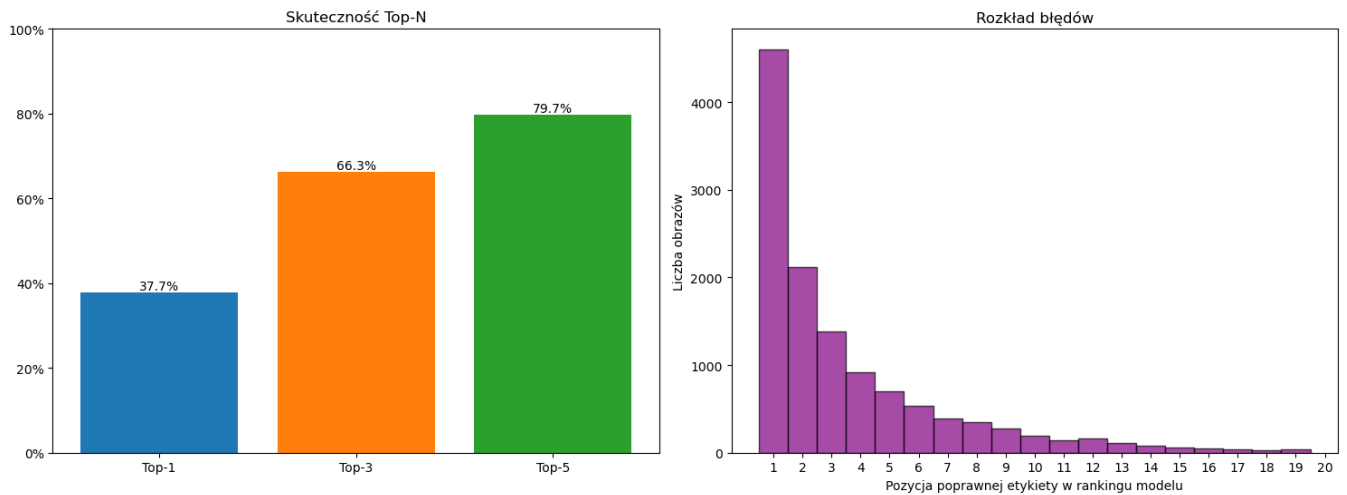
Rys. 6. Macierz pomyłek – REDUKCJA LICZBY KLAS.



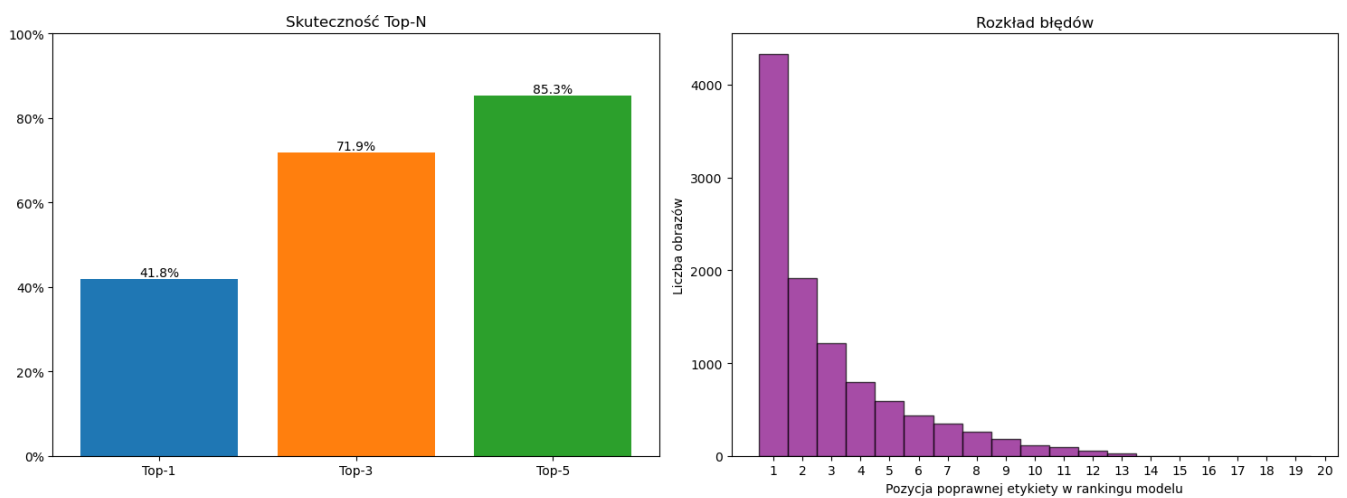
Rys. 8. Macierz pomyłek – REDUKCJA LICZBY KLAS.

Etap 3: Optymalizacja RFE (Wynik końcowy). Po dodaniu filtrów uzupełniających zastosowano rekurencyjną eliminację cech (RFE), która pozwoliła na dalszą poprawę wyników. Ostateczny model dla wszystkich klas osiągnął dokładność (Accuracy) na poziomie **37,71%**, a dla zredukowanej liczby klas **41,77%**, czyli poprawa o około 0,5 i 2 p.p. w porównaniu do poprzedniego modelu. Poniżej przedstawiono macierz

Wykres skuteczności Top-N oraz rozkład błędów (pozycja poprawnej etykiety w rankingu modelu) dla wszystkich klas (por. Rysunek 9) i zredukowanej liczby (por. Rysunek 10) przedstawiono poniżej.



Rys. 9. Skuteczność Top-N oraz rozkład błędów - WSZYSTKIE KLASY.



Rys. 10. Skuteczność Top-N oraz rozkład błędów - REDUKCJA LICZBY KLAS.

4. DYSKUSJA

Analiza wyników pozwoliła na wyciągnięcie istotnych wniosków dotyczących natury stylów malarskich oraz skuteczności poszczególnych metod ekstrakcji cech.

Dzięki zastosowaniu metody RFE ustalono, że najbardziej dyskryminującymi cechami dla rozpoznawania stylu są:

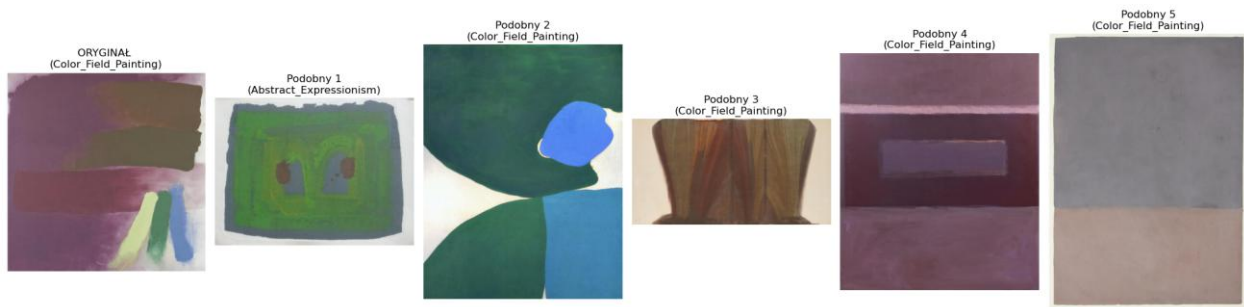
1. **Statystyki krawędzi (Edge Statistics).**
2. **Cechy Haralicka (GLCM).**
3. **HOG.**

Analiza macierzy pomyłek wskazuje na istnienie silnych powiązań historycznych i wizualnych między mylonymi klasami. Przykładowo, *Post-Impressionism* jest bardzo często klasyfikowany jako *Impressionism*. Jest to błąd zrozumiały, wynikający z ewolucyjnego charakteru sztuki, gdzie granice między stylami są płynne.

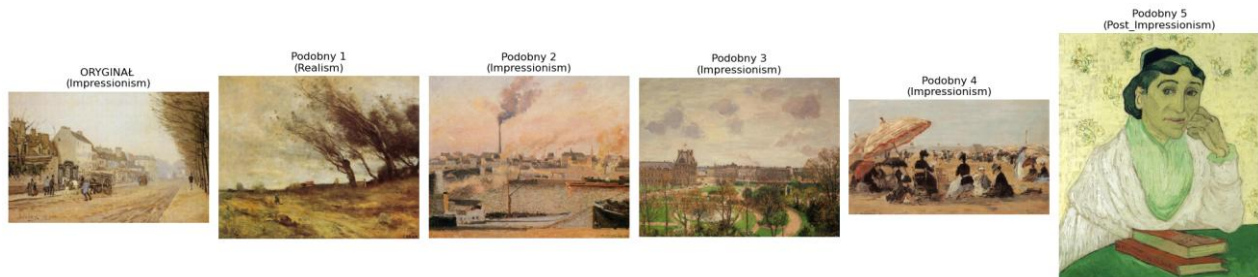
Analiza najbliższych sąsiadów (KNN) w przestrzeni wyekstrahowanych cech pokazała, że algorytm skutecznie grupuje obrazy o podobnych cechach wizualnych (kompozycja, kolorystyka, faktura), nawet jeśli należą one do różnych stylów (por. Rysunek 11, 12, 13, 14, 15).



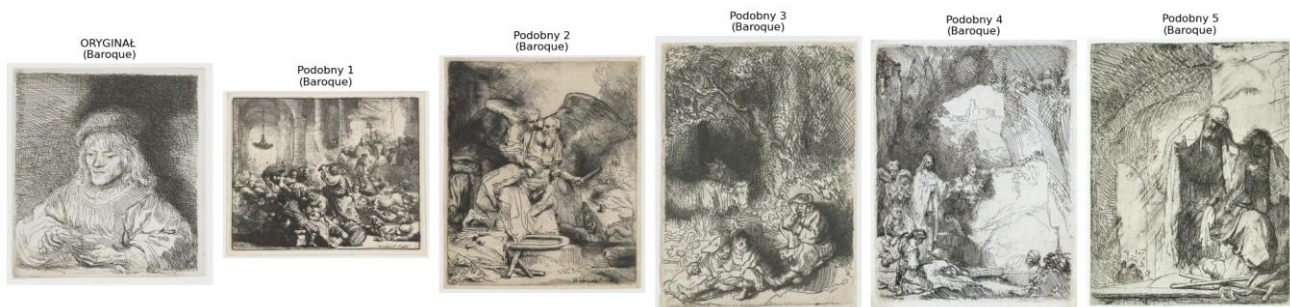
Rys. 11. Przykłady wyszukiwania obrazów podobnych wizualnie przez algorytm.



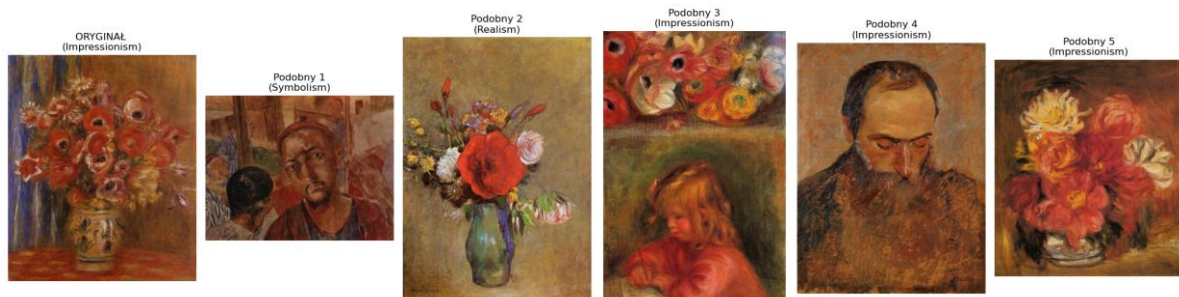
Rys. 12. Przykłady wyszukiwania obrazów podobnych wizualnie przez algorytm.



Rys. 13. Przykłady wyszukiwania obrazów podobnych wizualnie przez algorytm.



Rys. 14. Przykłady wyszukiwania obrazów podobnych wizualnie przez algorytm.



Rys. 15. Przykłady wyszukiwania obrazów podobnych wizualnie przez algorytm.

Otrzymany wynik Top-1 (~38%) jest porównywalny z wynikami literaturowymi dla metod klasycznych, co potwierdza poprawność implementacji. Jest on bezpośrednio porównywalny z osiągnięciami literaturowymi dla podejścia, gdzie złożone algorytmy fuzji cech osiągały ok. 39–41% (Saleh i Elgammal, 2015 [1]). Wynik ten potwierdza, że zaimplementowany zestaw filtrów skutecznie ekstrahuje kluczowe informacje o stylu, zbliżając się do limitu skuteczności dla metod opartych na ręcznej ekstrakcji cech. Jednocześnie pokazuje to ograniczenie metod klasycznych w porównaniu z sieciami głębokimi.

5. PODSUMOWANIE I WNIOSKI

W ramach projektu zrealizowano system klasyfikacji stylów malarskich, osiągając wyniki na poziomie **37,71% (Top-1)** oraz blisko **80% (Top-5)**. Jest to rezultat satysfakcjonujący, biorąc pod uwagę ograniczenie polegające na braku

możliwości użycia sieci neuronowych oraz dużą liczbę klas (27) i ich nierównomierny rozkład. **Główne wnioski:**

1. Klasyczne metody ML mają trudności z rozróżnianiem stylów historycznie powiązanych, ale bardzo dobrze radzą sobie z wyszukiwaniem obrazów podobnych wizualnie.
2. Redukcja rzadkich klas poprawia ogólną celność modelu (do ok. 42%), co sugeruje, że niezbalansowanie zbioru jest istotnym problemem.

W przyszłości projekt można by rozwinąć poprzez zastosowanie hierarchicznej klasyfikacji (najpierw epoka, potem konkretny styl). Praktycznym zastosowaniem metody może być system rekomendacji dzieł sztuki podobnych wizualnie dla użytkowników galerii cyfrowych.

6. DEKLARACJA UDZIAŁU AUTORÓW

Wkład autorów w realizację projektu:

Sebastian Borukalo: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation. **Jakub Bukala:** Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation. **Jakub Smakowski:** Conceptualization, Methodology, Validation, Formal Analysis, Writing - Original Draft, Writing - Review & Editing, Visualization. **Dr inż. Piotr Duch:** Supervision.

7. BIBLIOGRAFIA

[1] Saleh, B., i Elgammal, A. (2015). Large-scale class[1] Saleh, B., & Elgammal, A. (2015). Large-scale classification of fine-art paintings: Learning the right metric on the right feature.
<https://doi.org/10.48550/arXiv.1505.00855>

[2] Zhao, W., Zhou, D., Qiu, X., & Jiang, W. (2021). Compare the performance of the models in art classification. PLoS ONE, 16(3), e0248414.
<https://doi.org/10.1371/journal.pone.0248414>

[3] WikiArt Dataset, Kaggle Repository (user: steubk), <https://www.kaggle.com/datasets/steubk/wikiart> [dostęp: luty 2026].

[4] Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., & Winnemoeller, H. (2014). Recognizing Image Style.
<https://doi.org/10.48550/arXiv.1311.3715>

[5] Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2006). Studying Aesthetics in Photographic Images Using a Computational Approach.
https://doi.org/10.1007/11744078_23

[6] Misumi, M., Orii, H., Sharmin, T., Mishima, K., & Tsuruoka, T. (2016). Image classification for the painting style with SVM.

[7] Hallur, S., & Gavade, A. (2025). Image Feature Extraction Techniques: A Comprehensive Review.
https://www.researchgate.net/publication/395782910_Image_Feature_Extraction_Techniques_A_Comprehensive_Review

[8] Kowaliw, T., McCormack, J., & Dorin, A. (2010). Evolutionary Automated Recognition and Characterization of an Individual's Artistic Style.

[9] Mavridaki, E., & Mezaris, V. (2015). A comprehensive aesthetic quality assessment method for natural images using basic rules of photography.
<https://doi.org/10.1109/ICIP.2015.7350927>