

## WikiArt Art Movements/Styles

### Temat projektu

Celem projektu jest stworzenie modelu sztucznej inteligencji zdolnego do klasyfikacji obrazów na podstawie ich stylu artystycznego. Model ten będzie wykorzystywał dane z zestawu **WikiArt Art Movements/Styles**, który zawiera obrazy przypisane do różnych stylów sztuki. Zastosowaniem naszego projektu może być pomoc w organizacji dzieł w muzeach sztuki, w pomocy rozróżniania stylów artystycznych itp..

### Zakres projektu

- Przygotowanie i eksploracja zbioru danych
- Ewentualne wstępne przetwarzanie obrazów (normalizacja)
- Wybór odpowiedniej architektury modelu - CNN
- Trenowanie i walidacja modelu
- Testowanie dokładności na zbiorze testowym
- Optymalizacja i wdrożenie

### Wymagania techniczne

1. **Dane wejściowe:** Wylosowane obrazy z różnych stylów artystycznych, 13 kategorii: Academic Art, Art Nouveau, Baroque, Realism, Renaissance (Western), Romanticism, Expressionism, Japanese Art, Neoclassicism, Primitivism, Rococo, Symbolism, Western Medieval.

W niektórych przypadkach style zostały połączone w jedną kategorię (np. sztuka bizantyjska i średniowieczna) z powodu ograniczonej liczby próbek. Autor podkreśla, że nie jest ekspertem, więc zakładamy że w zbiorze uczącym mogą być obrazy niepoprawnie sklasyfikowane. Każda z kategorii ma średnio 2-3 tysiące obrazków (niektóre najmniejsze koło 1000, a największe koło 5000), różnej wielkości, w kolorach (RGB), o rozszerzeniu “.jpg” .

2. **Dane wyjściowe:** Klasa (styl artystyczny) przypisana do obrazu, jeden z wcześniej wymienionych stylów.
3. **Wydajność:** Model powinien osiągać wysoką dokładność ( $>80\%$ ). Dataset ma dużą ilość danych wejściowych (prawie 30 GB danych), co powinno być wystarczające do osiągnięcia wysokiej dokładności.
4. **Skalowalność:** Możliwość uruchamiania na GPU w celu przyspieszenia obliczeń. Wykorzystywana będzie biblioteka PyTorch, z która będzie można wykonać obliczenia na GPU z użyciem CUDA.
5. **Bezpieczeństwo:** Ograniczenie możliwości błędnej klasyfikacji poprzez monitoring modelu, na przykład generowanie wykresu wyników treningu co kilka epok i zapewnienie różnorodności danych (30 GB), aby model był bardziej odporny na nieprzewidziane warianty obrazów (Data Augmentation).

### WikiArt Art Movements/Styles:

<https://www.kaggle.com/datasets/sivarazadi/wikiart-art-movementsstyles>

---

## CZEŚĆ II - Zbiór danych i ich przygotowanie

W ramach przygotowania danych do klasyfikacji stylów artystycznych rozpoczęliśmy od ich wstępного przetworzenia. Obrazy były zapisane w strukturze katalogów odpowiadającej stylom (niektóre z nich zawierały dodatkowe podkatalogi).

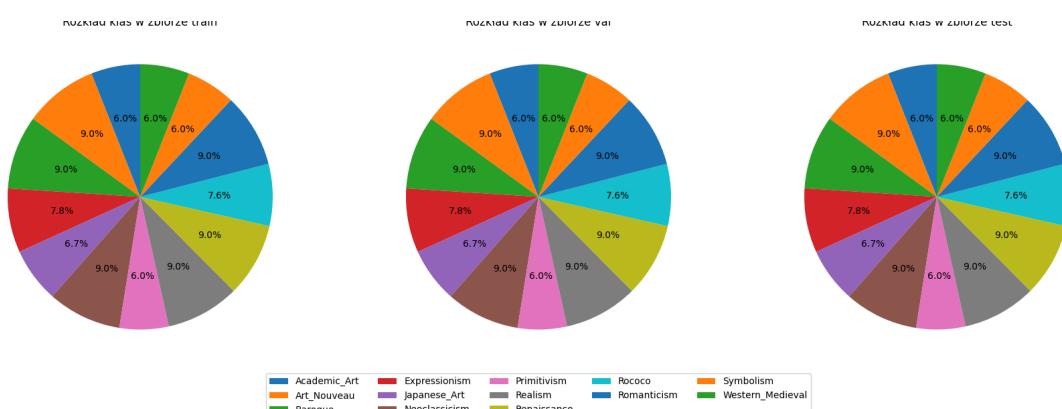
Aby ułatwić dalsze przetwarzanie, zdecydowaliśmy się na ujednolicenie rozmiaru obrazów. Każdy obraz został przeskalowany do wymiaru 512x512 pikseli z zachowaniem proporcji i dodaniem czarnego tła (paddingu), co pozwoliło na zachowanie spójności.

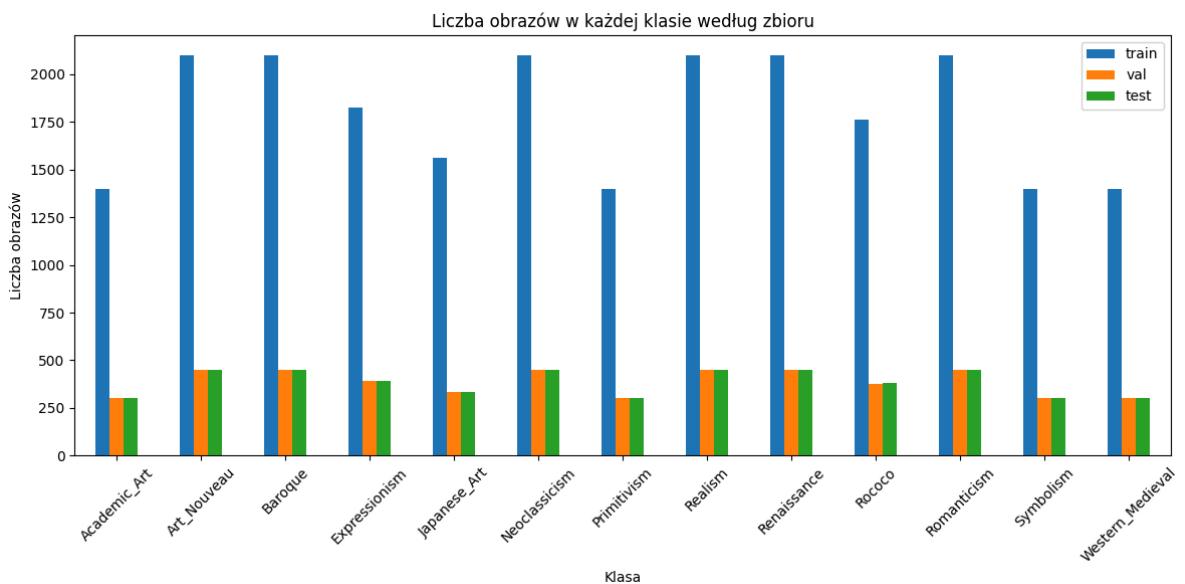
W kolejnym kroku skupiliśmy się na przygotowaniu zbiorów treningowego, walidacyjnego i testowego. Podział ten wynosił odpowiednio 70%, 15% i 15%. Aby unormować dane, postanowiliśmy, że liczba obrazów w każdej klasie (czyli folderze stylu) powinna mieścić się w przedziale od 2000 do 3000. Foldery zawierające zbyt dużą liczbę zdjęć zostały zredukowane – losowo wybraliśmy określona liczbę przykładów. Natomiast w przypadku klas zbyt ubogich w dane, zdecydowaliśmy się na prostą augmentację – poprzez rotację wybranych obrazów pod kilkoma kątami utworzyliśmy nowe przykłady, które zwiększyły liczosć klasy.

W trakcie przetwarzania wprowadzono też dodatkowe przekształcenia. Dla zbioru treningowego zastosowaliśmy klasyczne techniki augmentacji, takie jak obrót czy odbicie lustrzane. Celem było zwiększenie różnorodności danych, a tym samym poprawienie uogólnienia modelu. Dla zbiorów walidacyjnego i testowego ograniczyliśmy się wyłącznie do przeskalowania i centralnego przycięcia, unikając w ten sposób przypadkowego zniekształcenia obrazów.

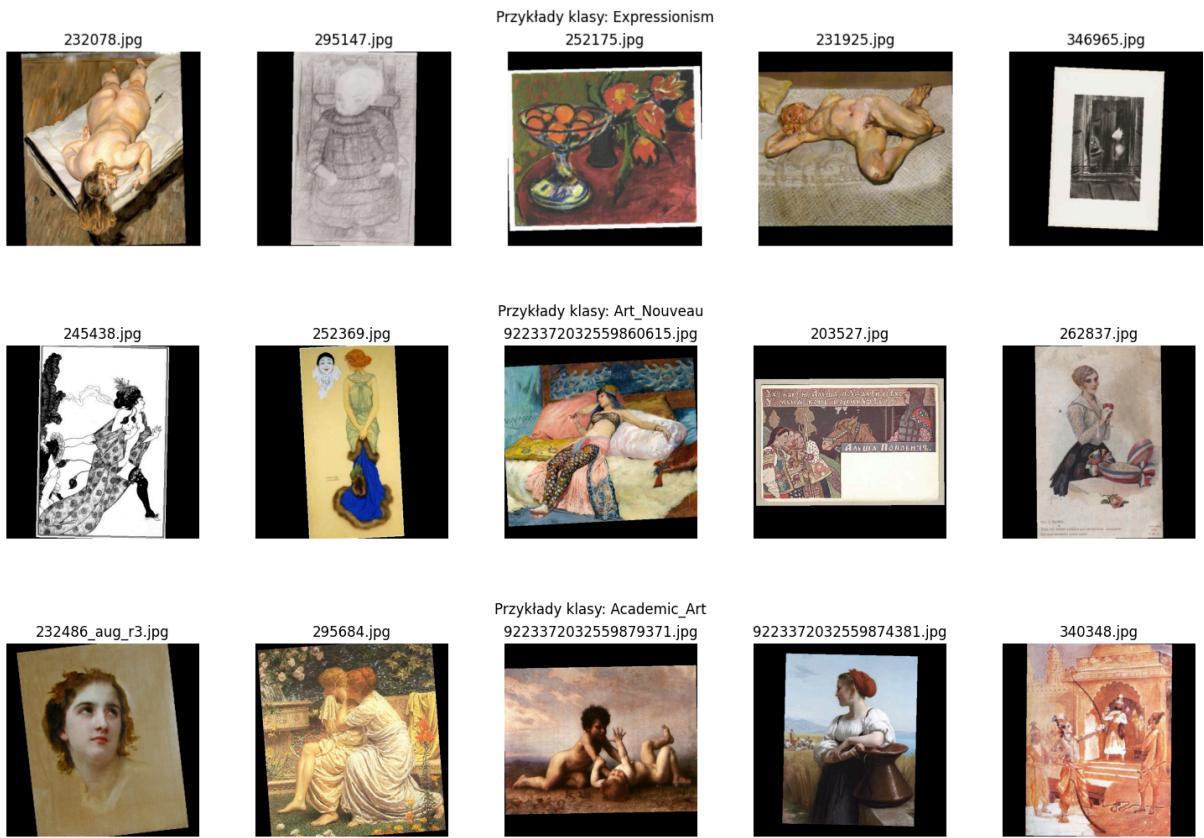
Aby móc efektywnie trenować modele, przygotowaliśmy klasę Dataset, która odpowiada za ładowanie obrazów oraz przypisywanie im etykiet. Do ładowania danych wykorzystaliśmy DataLoader z odpowiednią konfiguracją – zbiór treningowy był mieszany (shuffle=True), podczas gdy walidacyjny i testowy pozostawały w ustalonej kolejności.

W celu oceny jakości podziału i weryfikacji zbalansowania danych, wygenerowaliśmy wykresy kołowe oraz słupkowe przedstawiające liczebność poszczególnych klas w każdym ze zbiorów:





Dodatkowo przygotowano wizualizacje przykładowych obrazów wraz z przypisanymi etykietami:



### **CZĘŚĆ III - Wybór i implementacja modelu AI**

Zdecydowaliśmy się na implementację własnej architektury konwolucyjnej, inspirowanej klasycznymi rozwiązaniami, takimi jak ResNet jednak w pełni zaimplementowanej ręcznie, bez korzystania z gotowych klas modeli dostępnych w bibliotece torchvision.

Stworzona została klasa ArtStyleCNN dziedzicząca po nn.Module, w której zawarte zostały cztery główne bloki konwolucyjne, każdy złożony z następujących elementów:

warstwa konwolucyjna nn.Conv2d z odpowiednim paddingiem,

warstwa normalizacji nn.BatchNorm2d,

funkcja aktywacji nn.ReLU,

warstwa sub-samplująca nn.MaxPool2d.

Po przejściu przez bloki konwolucyjne, dane są spłaszczone za pomocą nn.Flatten(), a następnie przekazywane do w pełni połączonych warstw (nn.Linear), gdzie ostatnia warstwa zawiera liczbę neuronów równą liczbie klas (stylów artystycznych) – w naszym przypadku 13. Dodatkowo zastosowano Dropout w warstwie w pełni połączonej, co ma na celu ograniczenie ryzyka przeuczenia modelu i poprawę jego ogólności.

#### **Funkcja treningowa:**

Do trenowania modelu przygotowaliśmy dedykowaną funkcję, która w każdej epoce wykonuje: przejście przez zbiór treningowy i obliczenie wartości funkcji kosztu (wykorzystano CrossEntropyLoss, odpowiednią dla zadań klasyfikacyjnych), optymalizację wag sieci przy pomocy algorytmu Adam (torch.optim.Adam) z ustalonym współczynnikiem uczenia (learning rate = 0.001), validację wyników na zbiorze walidacyjnym oraz zapis wartości straty i dokładności.

#### **Hiperparametry treningu:**

Do nauki modelu przyjęto następujące parametry:

liczba epok: 10

batch size: 32

funkcja kosztu: CrossEntropyLoss

optymalizator: Adam z parametrem lr=0.001 learning rate

#### **Przebieg treningu:**

Model był trenowany, jednak nie osiągnął satysfakcyjny poziom. Funkcja kosztu malała wraz z liczbą epok, co świadczy o poprawnym procesie uczenia. Dokładność na zbiorze walidacyjnym sukcesywnie wzrastała.]

---

## CZEŚĆ IV - Ocena wyników modelu i optymalizacja

W celu sprawdzenia poprawności naszego modelu, dodaliśmy model ResNet18 w wersji pre-trained i bazowej (nie trenowanej). Z wykorzystaniem odpowiednich metryk możemy porównać wyniki i różnicę pomiędzy modelami oraz sposób ich usprawnienia.

ResNet18\_pretrained (z wagami wstępnie wytrenowanymi),

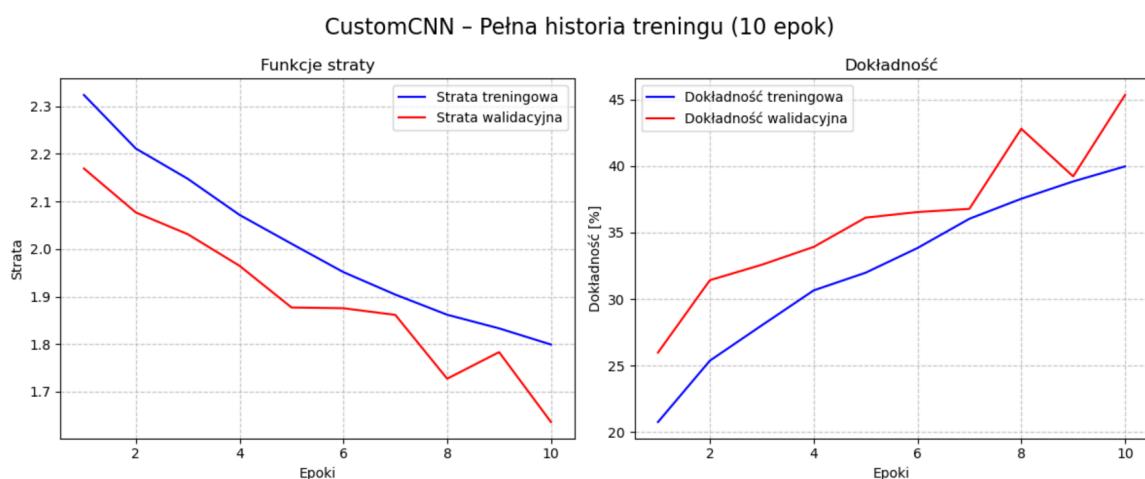
ResNet18\_scratch (trenowany od zera)

ArtStyleCNN (model własny)

Ocena została przeprowadzona na podstawie: wykresów funkcji straty i dokładności dla zbioru treningowego i walidacyjnego, metryk klasyfikacyjnych: Accuracy, Precision, Recall, F1-score, oraz macierzy pomyłek.

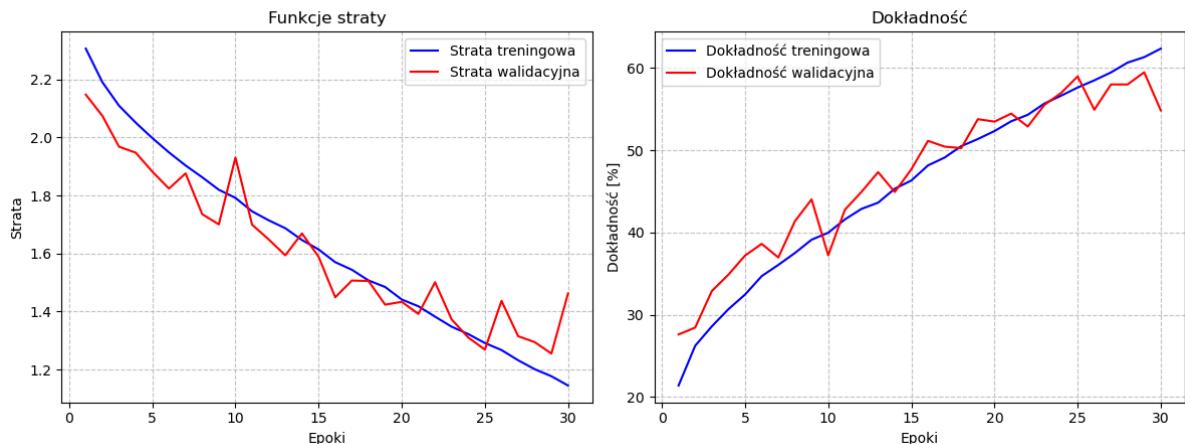
### 1. Analiza wykresów funkcji straty

Wszystkie 3 modele były trenowane (wstępnie) przez 10 epok. Poniżej dla każdego modelu przedstawione zostały: po lewej - wykres funkcji straty, po prawej - wykres dokładności modelu.



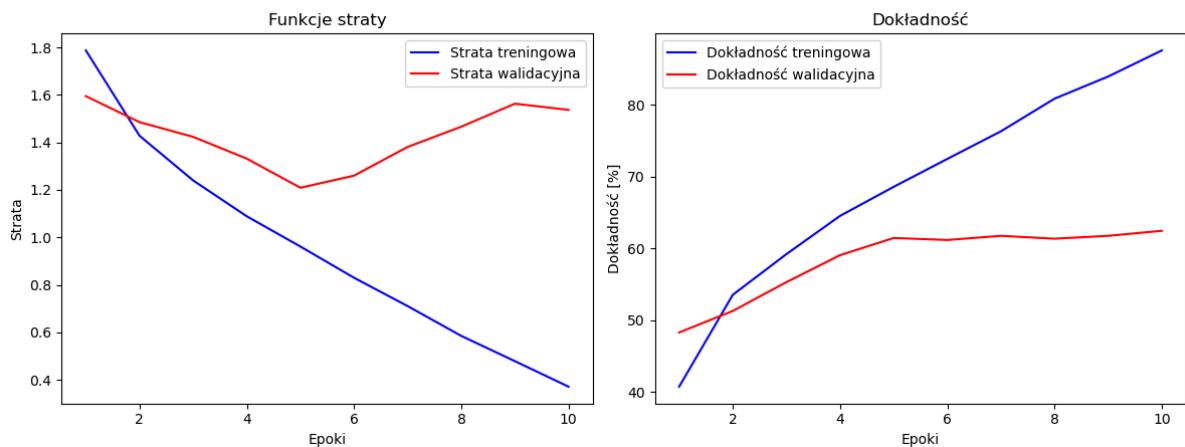
Jak widać z wykresu model dobrze się spisuje, linia czerwona i niebieska (trening i walidacja) na obu wykresach jest blisko siebie. Oznacza to że model się wciąż uczy, nie występuje overfitting i zalecane jest douszczenie modelu. W związku z tym dalej uczyliśmy model do 30 epok.

### CustomCNN – Pełna historia treningu (30 epok)



Model uczy się wciąż, jednak pod koniec około 29 epoki widać że prawdopodobnie model zaczął się przeszukać. Warto byłoby sprawdzić dalej.

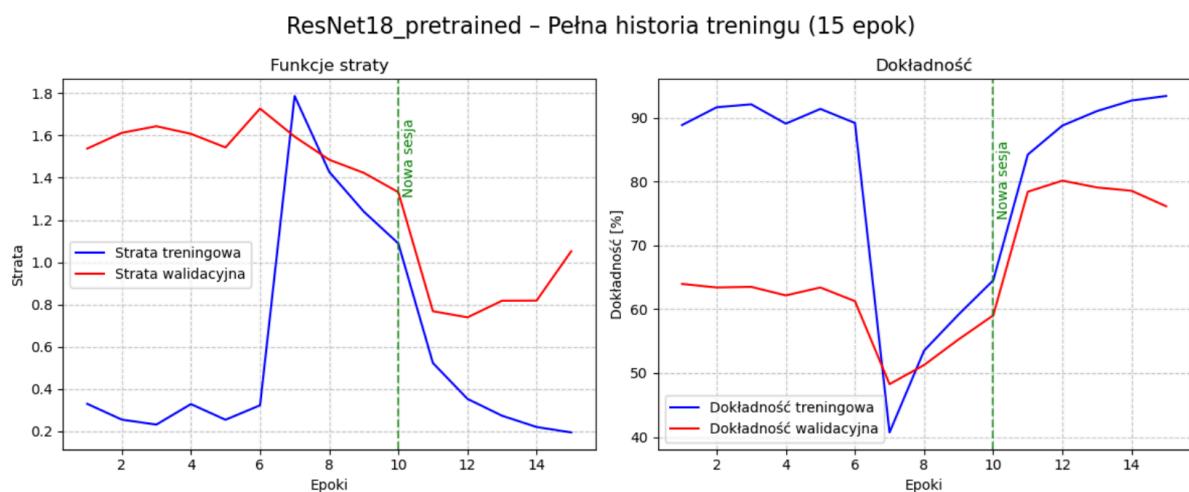
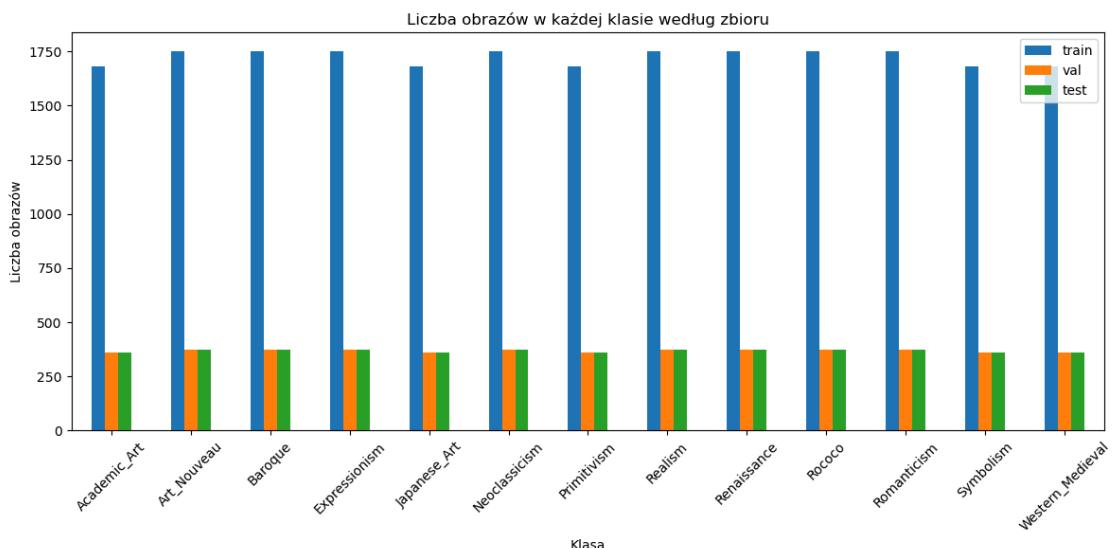
### ResNet18\_pretrained – Historia treningu



Wykres funkcji straty - linia treningowa (niebieska) spada systematycznie, jednak linia walidacyjna (czerwona) spada tylko do 5 epoki, następnie zaczyna rosnąć. Oznacza to nic innego jak overfitting, czyli model zaczyna się przeuczać ->zaczyna uczyć się danych ze zbioru treningowego na pamięć.

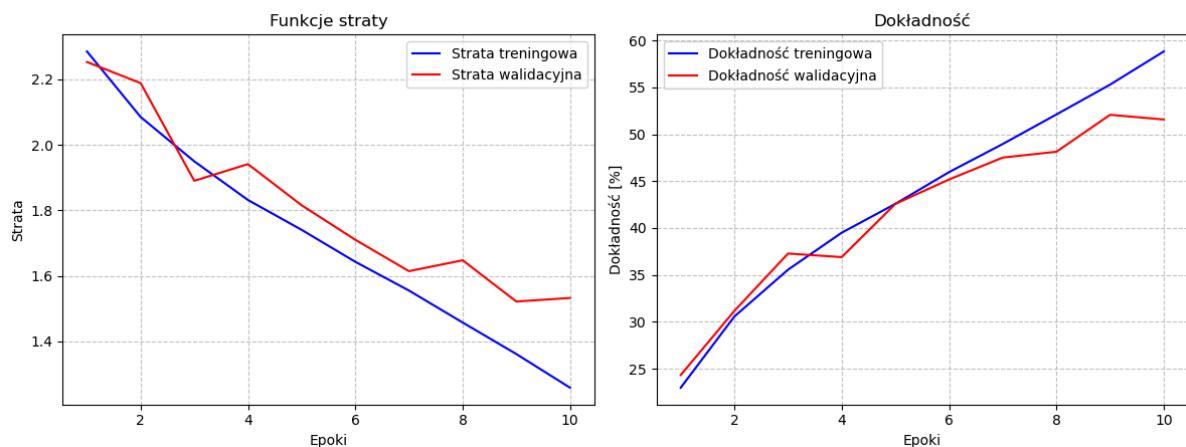
Z wykresu dokładności oraz bazując na wcześniejszym wykresie możemy stwierdzić, że swoje "maximum" zdolności generalizacji (czyli umiejętności rozpoznawania nowych danych) model osiągnął po ok. 5 epoce z wartością około 62%.

Jednak widząc wyniki jakie osiągnął po tych kilku epokach uznaliśmy, że ponownie damy mu szansę wykazać się, ale po ponownej augmentacji naszych danych, aby różnica między największym i najmniejszym zbiorem nie była zbyt duża.



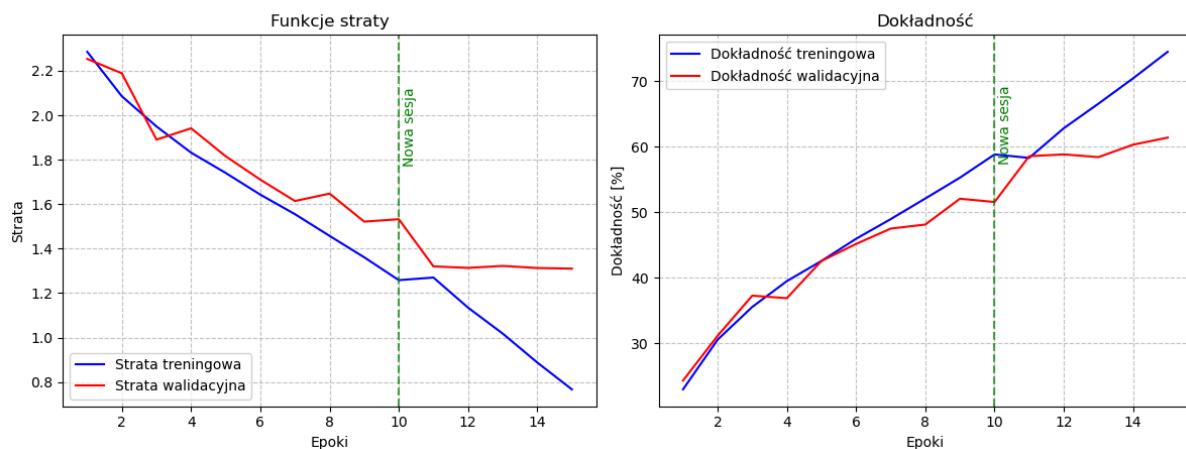
Jak widać na wykresach powyżej powstał błąd w trakcie tworzenia wykresu (ale ignorujemy - patrzmy na dane od 10 - 15 epok)). Model został przeuczony po ok. 12 epoce. Do tej epoki strata jak i dokładność poprawiają się. Niesety po tej epoce jak widać na wykresie funkcji straty, strata walidacyjna zaczyna gwałtownie wzrastać, za to treningowa wciąż spada. Ma to swoje odzwierciedlenie na wykresie dokładności, gdzie widać wyraźnie, iż pomimo dalszego uczenia dokładność modelu na zbiorze walidacyjnym pozostaje w okolicy 80%.

ResNet18\_scratch – Pełna historia treningu (10 epok)



Model ResNet18 trenowany od zera osiąga stopniową poprawę zarówno na zbiorze treningowym, jak i walidacyjnym. Strata treningowa i walidacyjna systematycznie maleją, a dokładność obu zbiorów rośnie. Po obiecujących wynikach i braku “nauki na pamięć” modelu przetrenowaliśmy model na kolejne 5 epok.

ResNet18\_scratch Pełna historia treningu (15 epok)



Niestety jak widać po dalszym treningu po epoce 11 nasz model zaczął się przeuczać.

## 2. Porównanie metryk klasyfikacyjnych dla 10 epok

Metryki	ArtStyleCNN	ResNet18_pretrained	ResNet18_strach
Accuracy	45.33%	62.44%	52.07%
Precision	0.4607	0.6252	0.5331

Metryki	ArtStyleCNN	ResNet18_pretrained	ResNet18_strach
Recall	0.4533	0.6244	0.5207
F1-Score	0.4503	0.6218	0.5198

**Accuracy** - odsetek poprawnie sklasyfikowanych próbek względem wszystkich próbek.

**Precision** - z jakim prawdopodobieństwem model ma rację, gdy przewiduje daną klasę (jak często „tak” oznacza rzeczywiście „tak”)

**Recall** - ile z rzeczywistych przykładów danej klasy model odkrył (jak dobrze model znajduje wszystkie „tak”)

**F1-Score** - to średnia harmoniczna precyzji i recall. Balansuje oba te aspekty w jednej liczbie.  
(Wysoki F1-score = model ma zarówno wysoką precyzję, jak i dobry recall)

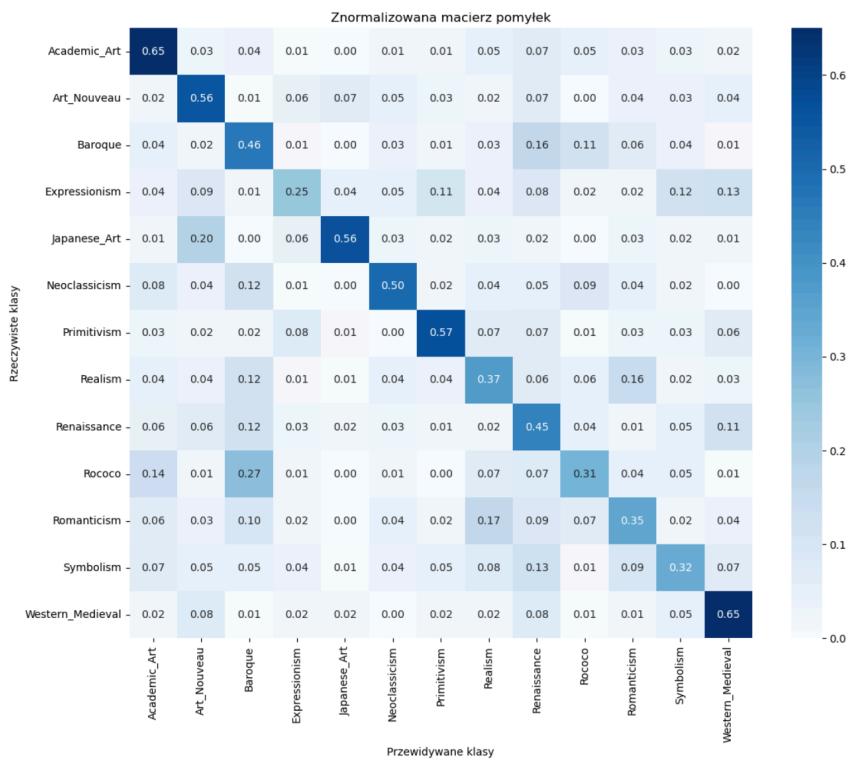
ResNet18\_scratch uzyskał do 10 epoki najlepsze wyniki we wszystkich metrykach (accuracy, precision, recall, F1). Świadczy to o tym, że model nauczył się skutecznie klasyfikować i dobrze dopasował się do danych. Gorsze wyniki uzyskał ResNet18\_pretrained a najgorsze nasz własny model ArtStyleCNN.

### 3. Szczegółowa analiza klasyfikacji do 10 epoki

Wiersze - rzeczywiste klasy ; Kolumny - przewidywane klasy ; Wartości - ile razy dana klasa została przewidziana jako inna (lub poprawnie).

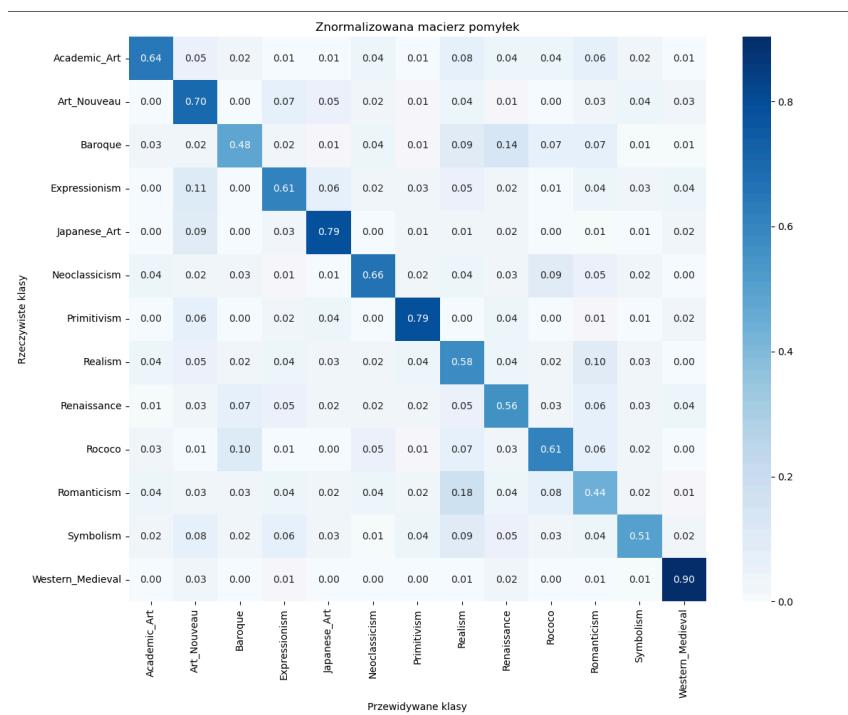
Tutaj macierz jest znormalizowana, czyli wartości są od 0 do 1 i pokazują procent trafień lub pomyłek dla każdej klasy. Idealnie byłoby, gdyby wszystkie wartości były wysokie na przekątnej (model poprawnie definiuje).

**ArtStyleCNN**



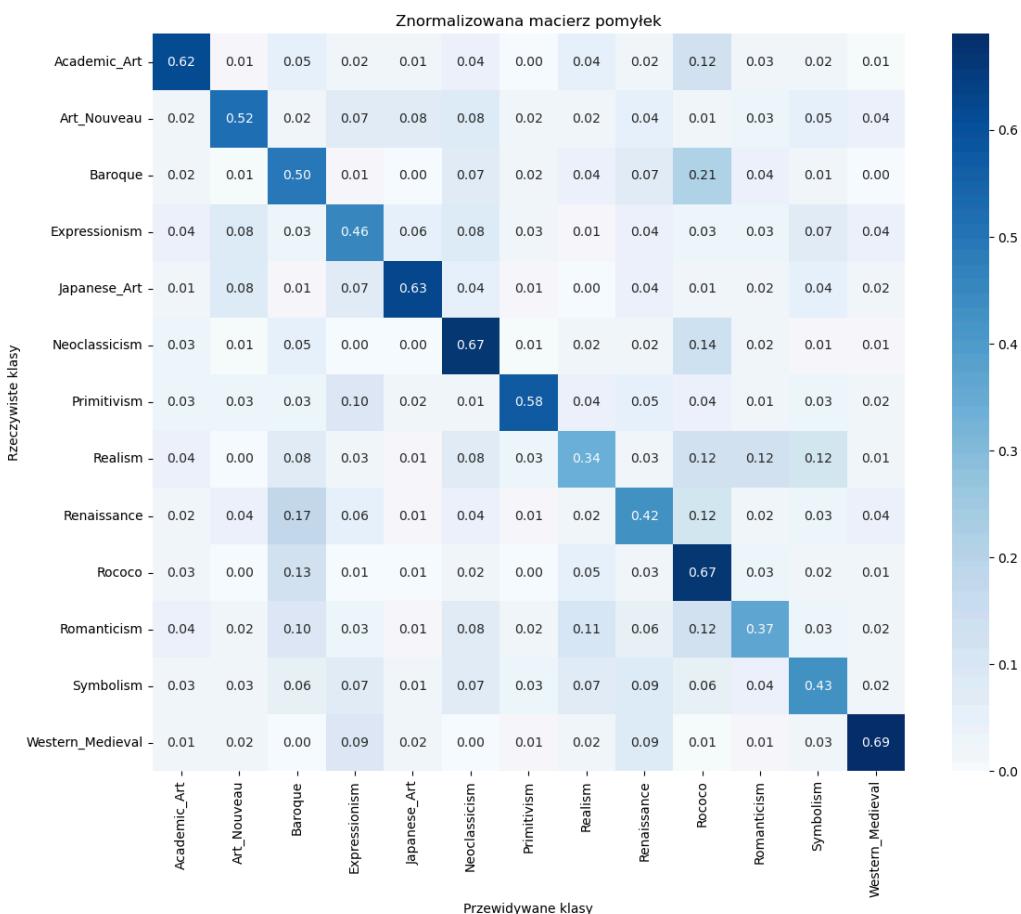
ArtStyleCNN najlepiej radzi sobie z rozpoznawaniem stylów takich jak Academic\_Art, Primitivism, Japanese\_Art i Western\_Medieval, dla których trafność klasyfikacji wynosi powyżej 56–65%. Natomiast style takie jak Expressionism i Rococo są dla modelu bardziej problematyczne, z trafnością odpowiednio 25% i 31%, oraz dużą liczbą pomyłek z klasami pokrewnymi wizualnie, np. Romanticism czy Baroque.

ResNet18\_pretrained:



Model skutecznie rozpoznaje większość stylów artystycznych, szczególnie Western\_Medieval (90%), Japanese\_Art, Primitivism i Art\_Nouveau (ponad 70%). W porównaniu do modelu ArtStyleCNN widać znacznie lepsze rozpoznanie trudniejszych klas, takich jak Expressionism czy Renaissance. Nadal jednak pojawiają się pomyłki między klasami bliskimi wizualnie, np. Romanticism, Baroque i Symbolism.

ResNet18 scratch:



Model ResNet18 scratch radzi sobie dobrze z klasyfikacją kilku stylów, takich jak Neoclassicism, Rococo, Japanese\_Art czy Western\_Medieval, osiągając dokładność powyżej 60%. Jednak niektóre klasy, zwłaszcza Realism (34%) i Romanticism(37%), sprawiają trudności i są często mylone z innymi podobnymi stylami. Model generuje więcej rozproszonych błędów.

#### 4. Wnioski

Najlepsze uzyskane wyniki osiągnęliśmy dla Resnet\_pretrained - najlepsza dokładność walidacyjna 80.12% (epoka 12)

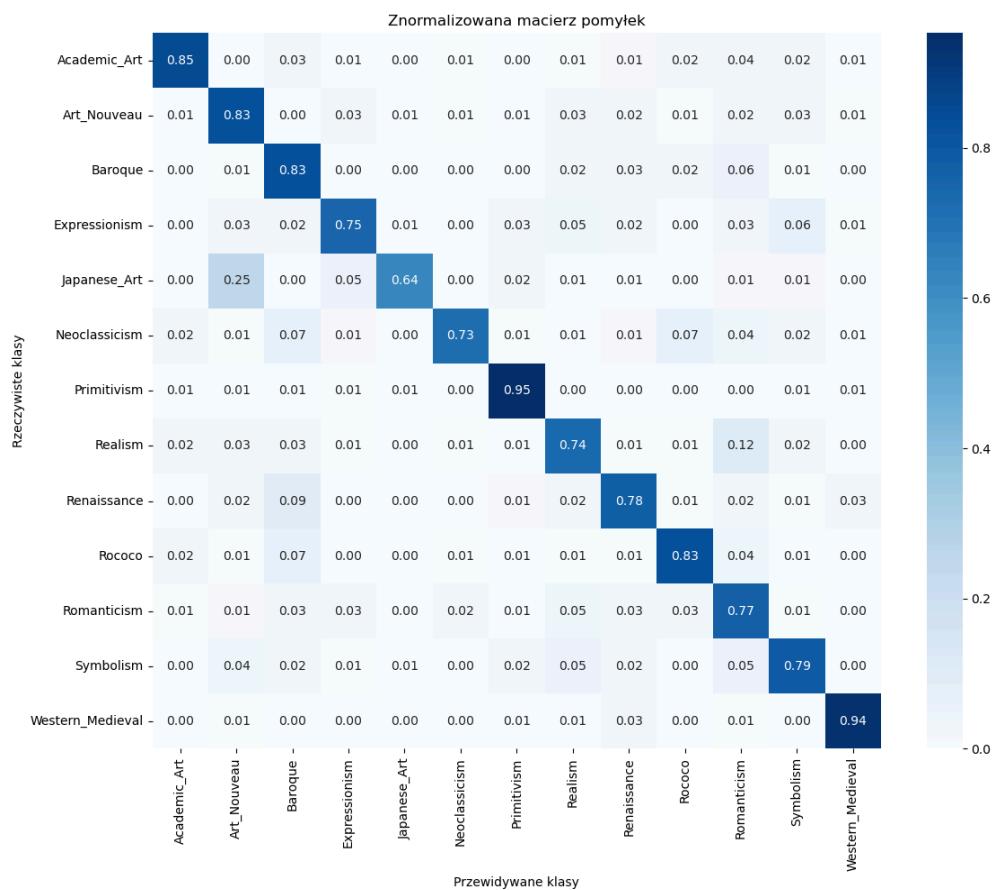
Finalne metryki:

- Accuracy: 80.12%

- Precision: 0.8134

- Recall: 0.8013

- F1-Score: 0.8021



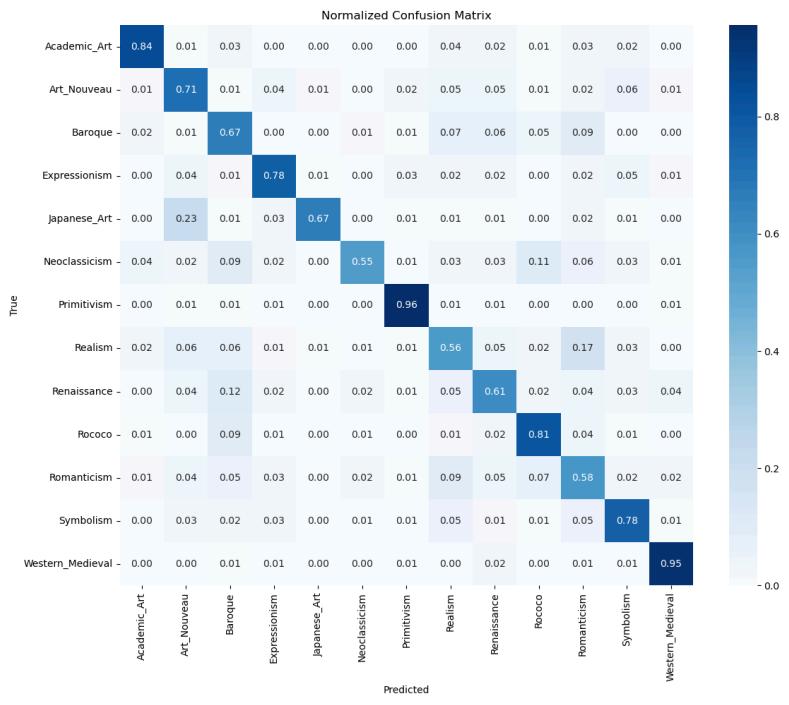
Model osiąga bardzo wysoką trafność klasyfikacji dla niemal wszystkich stylów artystycznych. Większość klas jest rozpoznawana z dokładnością powyżej 75%, a niektóre – jak Primitivism (95%) i Western\_Medieval (94%) – niemal bezbłędnie. Pojedyncze błędy dotyczą głównie klasy Japanese\_Art, która w 25% przypadków była mylona z Art\_Nouveau.

## CZĘŚĆ V - Wdrożenie modelu i monitorowanie

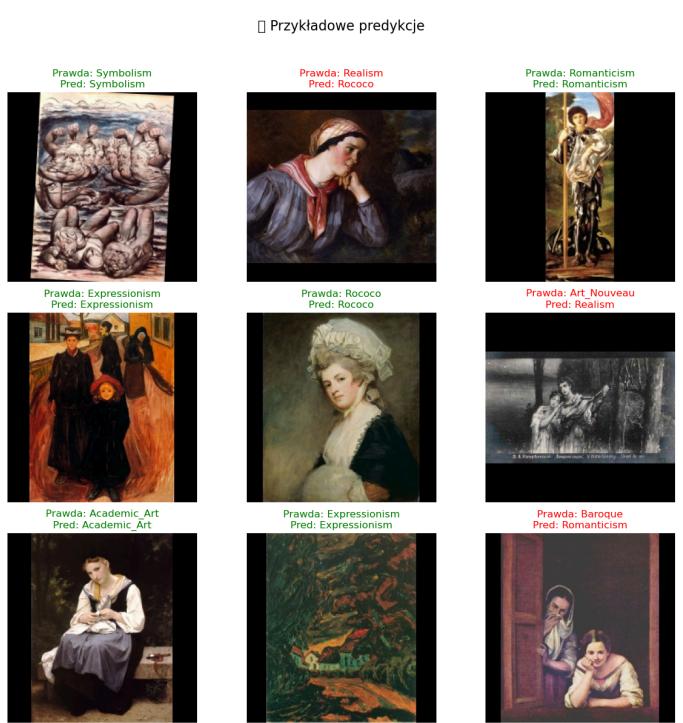
Po zakończeniu treningu podaliśmy nasze modele sprawdzeniu ich na zbiorze danych, który jeszcze nie widzieli/ nie trenowali się na nim. W wyniku zostały wygenerowane losowe zdjęcia i predykcja modelu, klasyfikacji obrazu do wybranego stylu artystycznego:

Dla najlepszego modelu czyli epoka 12 Resnet18\_pretrained:

Macierz po inferencji:



Macierz pomyłek po inferencji pokazuje, że model bardzo dobrze rozpoznaje wiele klas, zwłaszcza Primitivism (96%) i Western\_Medieval (95%). Trafność klasyfikacji dla większości stylów utrzymuje się powyżej 70%. Największe problemy model ma z rozróżnieniem stylów pokrewnych, takich jak Neoclassicism, Realism czy Romanticism, które są często mylone między sobą lub z Baroque i Rococo. Widoczne są też trudności w rozróżnianiu Japanese\_Art i Art\_Nouveau. Mimo to, ogólna jakość klasyfikacji jest bardzo dobra, a model wykazuje solidną zdolność rozpoznawania stylów artystycznych w danych testowych.



□ Przykładowe predykcje

Prawda: Realism  
Pred: Realism



Prawda: Baroque  
Pred: Baroque



Prawda: Rococo  
Pred: Rococo



Prawda: Western\_Medieval  
Pred: Western\_Medieval



Prawda: Academic\_Art  
Pred: Academic\_Art



Prawda: Primitivism  
Pred: Primitivism



Prawda: Expressionism  
Pred: Expressionism



Prawda: Baroque  
Pred: Realism



Prawda: Western\_Medieval  
Pred: Western\_Medieval

