

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Systemy informatyczne w automatyce (ASI)

**PRACA DYPLOMOWA
MAGISTERSKA**

Wieloparametrowa identyfikacja obrazów
malarskich z wykorzystaniem głębokich sieci
neuronowych

Multiparameter painting recognition with deep
neural nets

AUTOR:
Kacper Stenka

PROWADZĄCY PRACĘ:
dr inż. Piotr Ciskowski

OCENA PRACY: 5.0

Spis treści

1	Wstęp	3
1.1	Metody opisu dzieł malarskich	3
1.2	Sieci neuronowe	4
1.2.1	Podobieństwa między biologicznymi a sztucznymi sieciami neuronowymi	4
1.2.2	Różnice między biologicznymi a sztucznymi sieciami neuronowymi	5
1.2.3	Perceptron	6
1.2.4	Głębokie sieci neuronowe	7
2	Głębokie sieci neuronowe w zadaniu klasyfikacji obrazów	9
2.1	Historia	9
2.2	Cyfrowy zapis obrazów	10
2.3	Sieci konwolucyjne - opis ogólny	12
2.4	Sieci konwolucyjne - budowa	14
2.4.1	Warstwa wejściowa	14
2.4.2	Warstwa łącząca	14
2.4.3	Warstwa gęsta	15
2.4.4	Warstwa typu dropout	15
2.4.5	Warstwa wyjściowa	15
3	Architektura systemu	17
3.1	Sieci poziomu niższego	18
3.2	Sieci poziomu wyższego	19
3.2.1	Perceptron wielowarstwowy	20
3.2.2	Regresja logistyczna	20
3.2.3	Maszyna wektorów nośnych	21
3.2.4	Las losowy	22
4	Baza danych	23
4.1	Wariancja wewnątrzklasowa i międzyklasowa	24
4.2	Niezbalansowanie bazy danych	27
5	Szkolenie sieci	31
5.1	Sieci niższego rzędu	31
5.1.1	Sieć do identyfikacji gatunku	33
5.1.2	Sieć do identyfikacji artysty	41
5.1.3	Sieć do identyfikacji stylu	44
5.1.4	Sieć do identyfikacji wieku	49
5.1.5	Porównanie wyników sieci niższego rzędu	52

5.2	Sieci nadrzędne	54
6	Przykłady stosowania	61
6.1	Losowanie obrazów	61
6.2	Obrazy spoza bazy danych	66
7	Podsumowanie i wnioski	71
	Literatura	73

Rozdział 1

Wstęp

W rozdziale przedstawiono cel i zakres pracy dyplomowej, przybliżono idee przyświecające identyfikacji sztuki oraz wprowadzono pojęcia najbardziej fundamentalnego narzędzia, które zostało w pracy dyplomowej wykorzystane: sztucznych sieci neuronowych.

1.1 Metody opisu dzieł malarskich

Zasoby sztuki przechowywane w postaci cyfrowej w wielkich zbiorach ciągle się powiększają, a dostęp do nich staje się coraz powszechniejszy. Twierdzenie to sprawdza się dla różnych dziedzin sztuki - serwisy streamingowe (np. Netflix, Spotify) stają się najpopularniejszymi dostawcami filmu i muzyki, a obrazy malarskie udostępniane są poprzez wirtualne encyklopedie - takie jak wykorzystany w niniejszej pracy serwis WikiArt.org. Aby sprawnie poruszać się po tych wielkich zbiorach sztuki, warto wykorzystywać takie pojęcia jak jej:

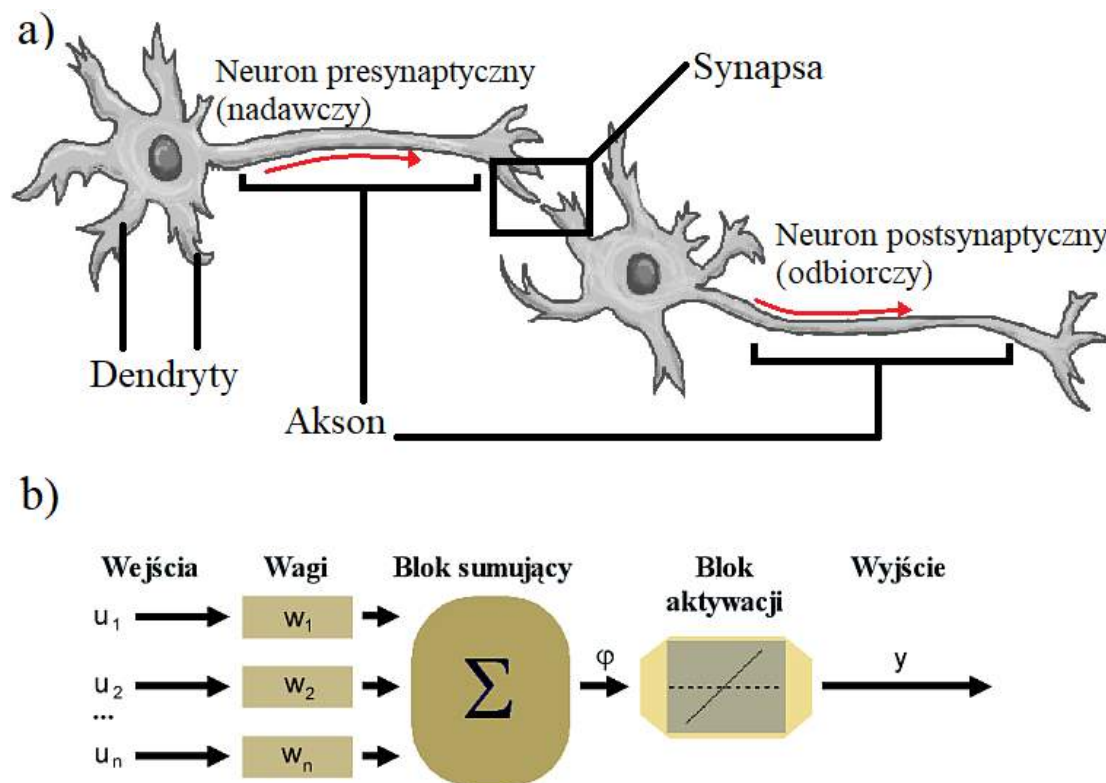
- styl,
- gatunek,
- twórca,
- czas powstania.

W przypadku obrazów malarskich styl odnosi się do sposobu przedstawienia (imprejonizm, kubizm itp.), natomiast gatunek - do tego, co jest na obrazie przedstawione (portret, krajobraz itp.). I choć istnieją dzieła, dla których próby zamknięcia w obrębie jednej klasy są krzywdzące i nie oddają do końca wiadomości, która się za nimi kryje, to takie katalogowanie jest potrzebne by ułatwiać komunikację między miłośnikami sztuki, pozwalać osobom zainteresowanym lepiej wgłębiać się w dany nurt lub uprościć poszukiwanie nowych ulubionych artystów.

Cztery wymienione powyżej kategorie są ze sobą silnie powiązane. W niniejszej pracy zaprojektowano system klasyfikujący wykorzystujący tę zależność. Składa się on z czterech osobnych konwolucyjnych sieci neuronowych, których zadaniami są identyfikacja artysty, stylu, gatunku albo stulecia oraz czterech sieci wyższego rzędu, które podejmują decyzję na bazie zebranych przewidywań wszystkich sieci konwolucyjnych.

1.2 Sieci neuronowe

Ludzkość w swoim rozwoju technologicznym od zawsze poszukiwała inspiracji w świecie natury. Podobnie jak budowa ptaków zainspirowała konstrukcję samolotu, tak w próbach stworzenia maszyny myślącej człowiek zwrócił swoją uwagę na ten obiekt występujący w przyrodzie, który najbardziej przypominał jego cel - czyli swój własny mózg. Mózg człowieka zbudowany jest z neuronów połączonych ze sobą za pomocą synaps. Neurony pobudzone zewnętrznym bodźcem przekazują impuls elektryczny o wielkości ładunku zależnej od siły połączeń synaptycznych. Taką sieć nazywa się *biologiczną* siecią neuronową, w odróżnieniu od *sztucznych* sieci neuronowych [1], które są tematem niniejszej pracy. Sztuczne sieci neuronowe zostały zainspirowane sieciami biologicznymi i są ich uproszczeniem poprzez wyrażenie matematyczne.



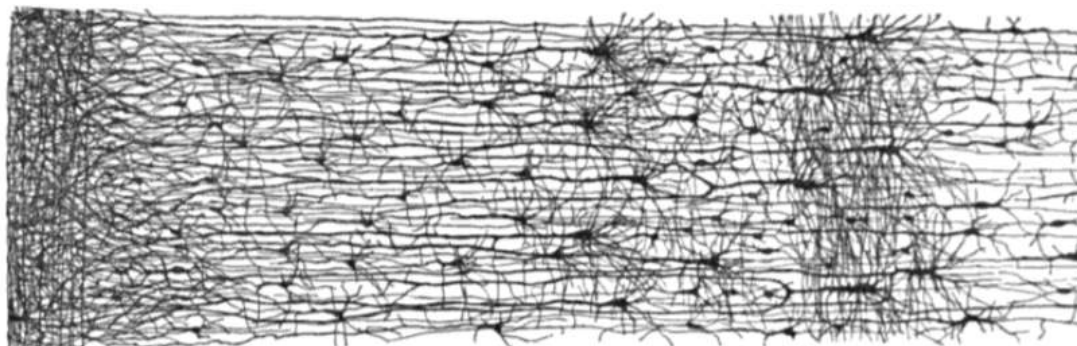
Rysunek 1.1 a) Połączenie dwóch neuronów biologicznych [Rysunek autorski],
b) Neuron sztuczny [8].

1.2.1 Podobieństwa między biologicznymi a sztucznymi sieciami neuronowymi

W przypadku sztucznych sieci neuronowych, neurony połączone są za pomocą wag, które pełnią tę samą rolę, co połączenia synaptyczne w biologicznych sieciach neuronowych. Warstwa wejściowa sztucznej sieci neuronowej jest odpowiednikiem receptorów biologicznej sieci neuronowej, a warstwa wyjściowa sztucznej sieci neuronowej jest odpowiednikiem efektorów biologicznej sieci neuronowej. Warstwa wejściowa i wyjściowa lub jej biologiczne odpowiedniki są niezbędne w przypisanych im typach sieci. Dendryty kumulują wszystkie przychodzące do neuronu impulsy, co było inspiracją do stworzenia funkcji

sumującej w neuronach sieci sztucznych. Aksony są odpowiednikami połączeń między warstwami sztucznych sieci neuronowych [9].

Uczenie sztucznej sieci neuronowej odbywa się poprzez zmianę wag pomiędzy neuronami. Podobnie jak biologiczne sieci neuronowe do nauki nowych rzeczy potrzebują impulsu zewnętrznego, tak do nauki sztucznej sieci neuronowej potrzebny jest zbiór danych treningowych zawierający przykładowe wejścia oraz wyjścia sieci [1].

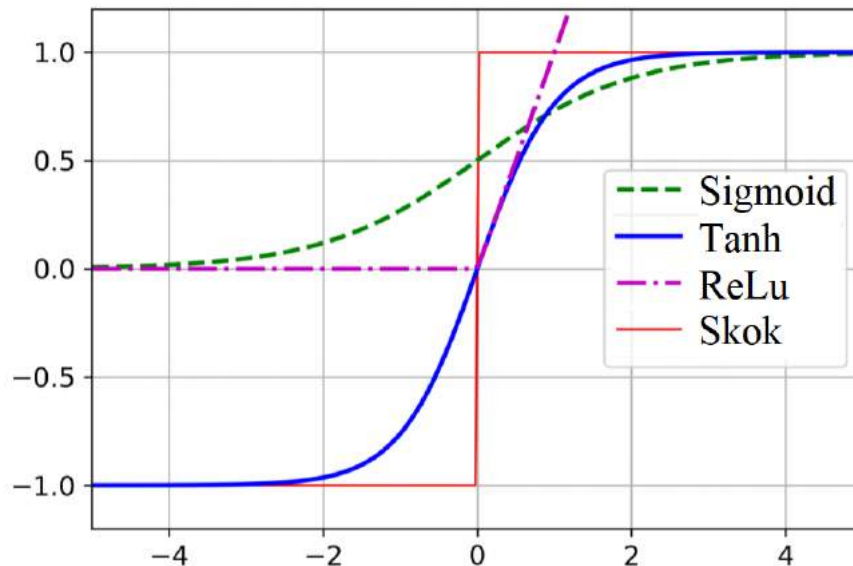


Rysunek 1.2 Wielowarstwowa biologiczna sieć neuronowa - kora mózgowa [6].

1.2.2 Różnice między biologicznymi a sztucznymi sieciami neuronowymi

W tym miejscu warto odwołać się do przywołanej wcześniej analogii z ptakiem i samolotem. Podobnie jak samolot, choć z kształtu wciąż przypominający ptaka, w celu latania nie macha skrzydłami, tak i sztuczne sieci neuronowe w wielu aspektach zaczęły na tyle odbiegać od budowy ludzkiego mózgu, że pojawiają się dyskusje, czy nazwa wciąż jest adekwatna i czy antropomorfizacja sztucznych sieci neuronowych nie ogranicza wyobraźni człowieka lub nie kieruje jej na niewłaściwe tory [6].

Jedną z kluczowych różnic jest stosowanie funkcji aktywacyjnych. Biologiczny neuron albo jest aktywny albo nie - czyli operuje zero-jedynkowo. Natomiast do aktywacji sztucznych neuronów używa się funkcji ciągłych. W przeszłości powszechnie stosowane były takie funkcje jak sigmoidalna lub tangens hiperboliczny, ale wprowadzały one problem zanikającego gradientu. Oznaczało to, że im więcej warstw sieci neuronowej, tym mniejszy wpływ na jej wyjście miały pierwsze warstwy sieci, ponieważ stosowane funkcje aktywacyjne działały na bardzo niewielkim zakresie wartości wyjściowych [17]. Problem ten został rozwiązany przez zastosowanie w roli funkcji aktywacyjnych takich funkcji jak ReLu, która nie posiada wartości maksymalnej oraz jest łatwiejsza do użycia w obliczeniach [6]. Spowodowało to uproszczenie modelu. Porównanie funkcji aktywacyjnych zamieszczono na Rysunku 1.3.



Rysunek 1.3 Porównanie funkcji aktywacyjnych [6].

Zastosowanie funkcji ciągłych umożliwiło szkolenie wielopoziomowych sztucznych sieci neuronowych metodą propagacji wstecznej, co okazało się być jednym z kamieni milowych w rozwoju tej dziedziny.

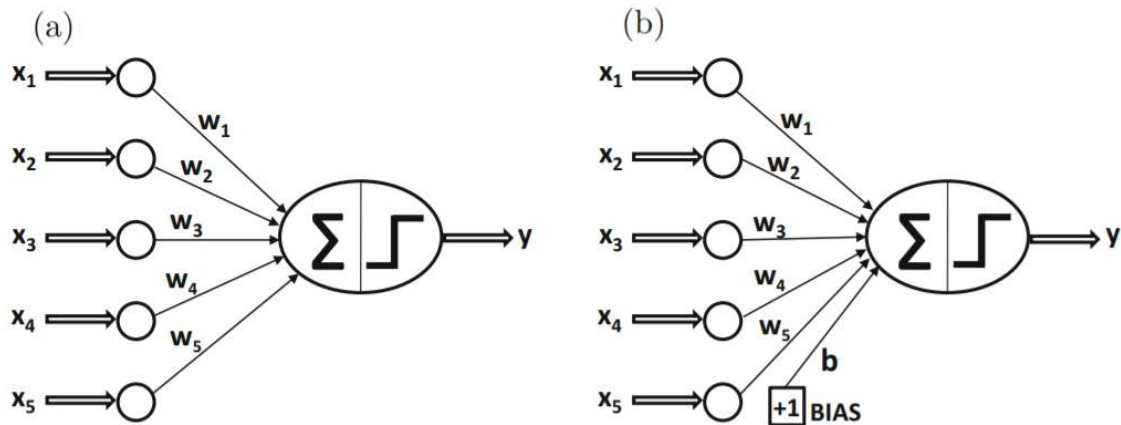
Do innych różnic między biologicznymi i sztucznymi sieciami neuronowymi zalicza się: rozmiar (w przypadku ludzkiego mózgu liczbę neuronów szacuje się na okolice 86 miliardów połączonych nawet tysiącem kwantylionów synaps, kiedy liczba neuronów sztucznej sieci neuronowej do typowych zadań zawiera się zazwyczaj w przedziale od 10 do 1000), topologię sieci, szybkość działania, odporność na błędy, zużycie energii, sposób uczenia [17].

Mimo tych różnic, podczas projektowania i uczenia sztucznych sieci neuronowych wciąż wykorzystuje się analogie z działaniem ludzkiego mózgu oraz przebiegiem procesu uczenia u dzieci.

Ze względu na przedmiot badań pracy, od tej pory używane w niej pojęcie „sieci neuronowe” będzie odnosiło się domyślnie do „sztucznych sieci neuronowych”.

1.2.3 Perceptron

Pierwszą siecią neuronową, która dała nadzieje na posiadanie przez ludzkość zaawansowanej sztucznej inteligencji, był zaprezentowany w 1957 roku perceptron [21]. Perceptron jest siecią opartą na jednym neuronie o skokowej funkcji aktywacyjnej, który sumuje wszystkie wejścia przemnożone przez wagi i na podstawie uzyskanej wartości przypisuje wyjściu wartość +1 lub -1. Suma podawana na perceptron może być opcjonalnie wzmocniona o bias, zgodnie z Rysunkiem 1.4.



Rysunek 1.4 a) Perceptron bez użycia biasu, b) Perceptron z użyciem biasu [17].

1.2.4 Głębokie sieci neuronowe

Sieć neuronową nazywa się *głęboką* (zamiennie z: *wielowarstwową*) wówczas, gdy między warstwę wejściową a wyjściową posiada ona więcej niż jedną warstwę neuronów. Warstwy neuronów między warstwę wejściową a warstwę wyjściową nazywa się *warstwami ukrytymi*, ponieważ obliczenia w nich wykonywane są niewidoczne dla użytkownika - poznaje on tylko dane wejściowe i wyjściowe, czyli znajdujące się na skrajnych warstwach [1].

Rozdział 2

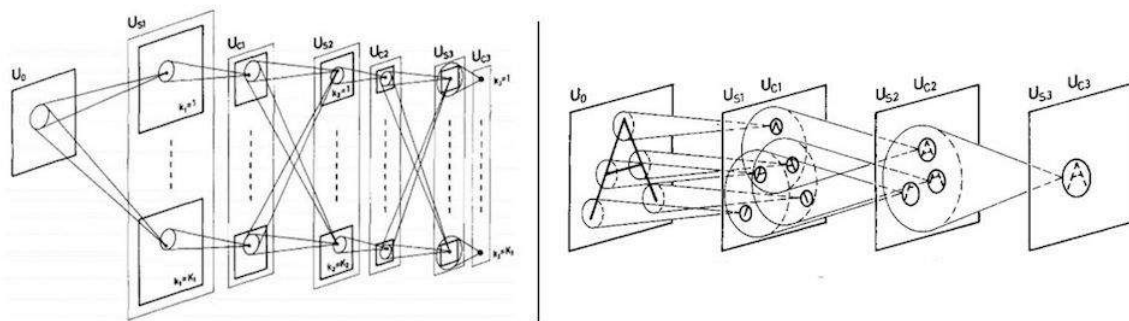
Głębokie sieci neuronowe w zadaniu klasyfikacji obrazów

Idea wykorzystania głębokich sieci neuronowych w operacjach na obrazach (poprzez wykonywanie takich zadań jak np. opracowywana w tej pracy klasyfikacja) zdobywa w ostatnich latach coraz większą popularność. Głębokie sieci neuronowe pozwalają uzyskać wysoką dokładność i umożliwiają zastosowanie nauczania maszynowego, czyli nie wymagają ręcznego wskazywania cech obrazów, a przy użyciu takich metod jak np. propagacja wsteczna są w stanie same uczyć się odpowiednich wzorców. Mający miejsce od początku XXI wieku szybki rozwój technologii komputerowej oraz coraz łatwiejszy dostęp do wielkich zbiorów danych umożliwiły szkolenie sieci nie tylko przez naukowców, ale i hobbystów; zaczęły się pojawiać różne konkursy na najlepsze sieci do konkretnych zastosowań, co skutkuje obecnie dalszym napędzaniem ich rozwoju.

W tym rozdziale omówiono historię badań nad głębokimi sieciami neuronowymi i nauczaniem maszynowym oraz przedstawiono sposób ich działania.

2.1 Historia

Wykorzystanie sieci neuronowych do przetwarzania obrazów zostało po raz pierwszy zaproponowane w latach 1960-tych, kiedy to Hubel i Wiesel dowiedli, że w mózgach kotów i małp istnieją neurony odpowiedzialne za osobne obszary pola widzenia [12]. Zainspirowało to Fukushima do wprowadzenia w 1980 pojęcia Neocognitronu, czyli pierwszej sztucznej sieci neuronowej próbującej naśladować działanie mózgów ssaków opisane uprzednio od strony biologicznej przez Hubela i Wiesela. Działała ona na bazie przesuwających się filtrów, wyszukujących pewnych wzorców na całej przestrzeni obrazu [5] - czyli filtrów konwolucyjnych, będących podstawą dla obecnie stosowanych w zadaniach przetwarzania obrazów sieci konwolucyjnych.



Rysunek 2.1 Neocognitron [24].

Neocognitron osiągnął dobre rezultaty w rozpoznawaniu wzorców, jednak ówczesny brak odpowiednich technik szkolenia sieci, takich jak wszechobecna w dzisiejszych czasach propagacja wsteczna, przełożył się na ograniczone możliwości udoskonalenia (i w efekcie - stosowania) tego typu sieci.

W 1986 roku Rumelhart i Hinton [22] po raz pierwszy skutecznie wykorzystali propagację wsteczną do szkolenia sieci neuronowej, a w 1990 roku Yamaguchi zaproponował użycie warstw tzw. max-poolingu [28], które stały się kolejnym z kluczowych elementów obecnie stosowanych sieci konwolucyjnych.

Powyższe badania znalazły zastosowanie, kiedy to w 1998 zespół Yanna LeCuna, byłego podopiecznego i współpracownika wspomnianego powyżej Hintona, wyszkolił sieć o nazwie LeNet, której celem była klasyfikacja ręcznie zapisanych cyfr. Osiągnęła w tym zadaniu skuteczność 99,3% [15] i była stosowana do automatycznego czytania 10-20% czeków wydrukowanych na terenie Stanów Zjednoczonych. Była to pierwsza sieć konwolucyjna wykorzystana na skalę przemysłową. Zbudowana była z siedmiu warstw: warstwy wejściowej, dwukrotnego połączenia warstwy konwolucyjnej z warstwą poolingu oraz dwóch warstw gęstych na wyjściu.

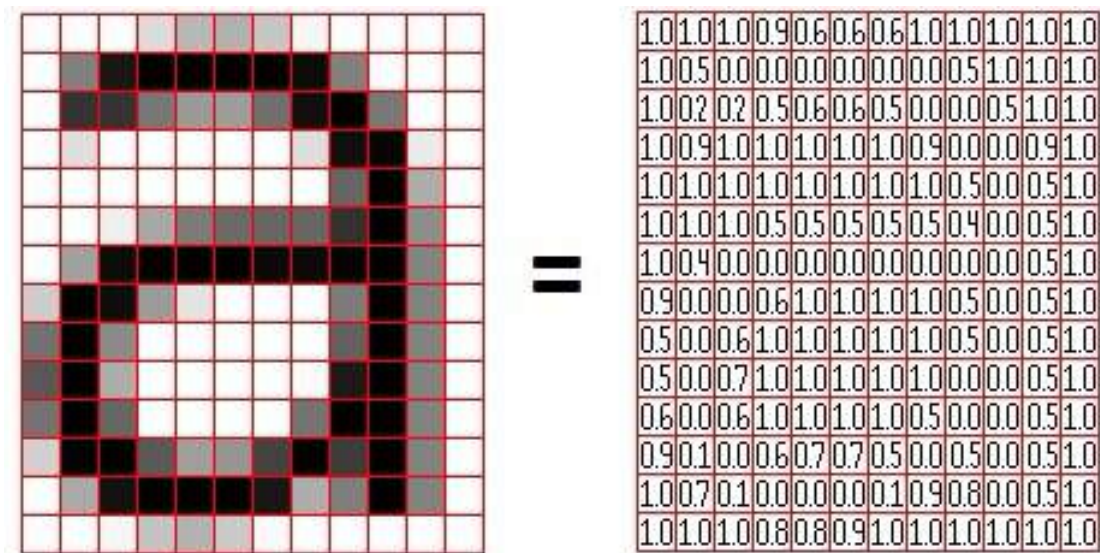
Prawdziwy rozwój sieci konwolucyjnych nastąpił w latach 2000-nych, kiedy do szkolenia sieci neuronowych zaczęto stosować karty graficzne, co znacznie przyspieszyło obliczenia i umożliwiło tworzenie bardziej skomplikowanych sieci. Oh i Jung w 2004 roku udowodnili, że dzięki przeniesieniu obliczeń z procesora na kartę graficzną można przyspieszyć ich wykonywanie 20-krotnie [18].

2.2 Cyfrowy zapis obrazów

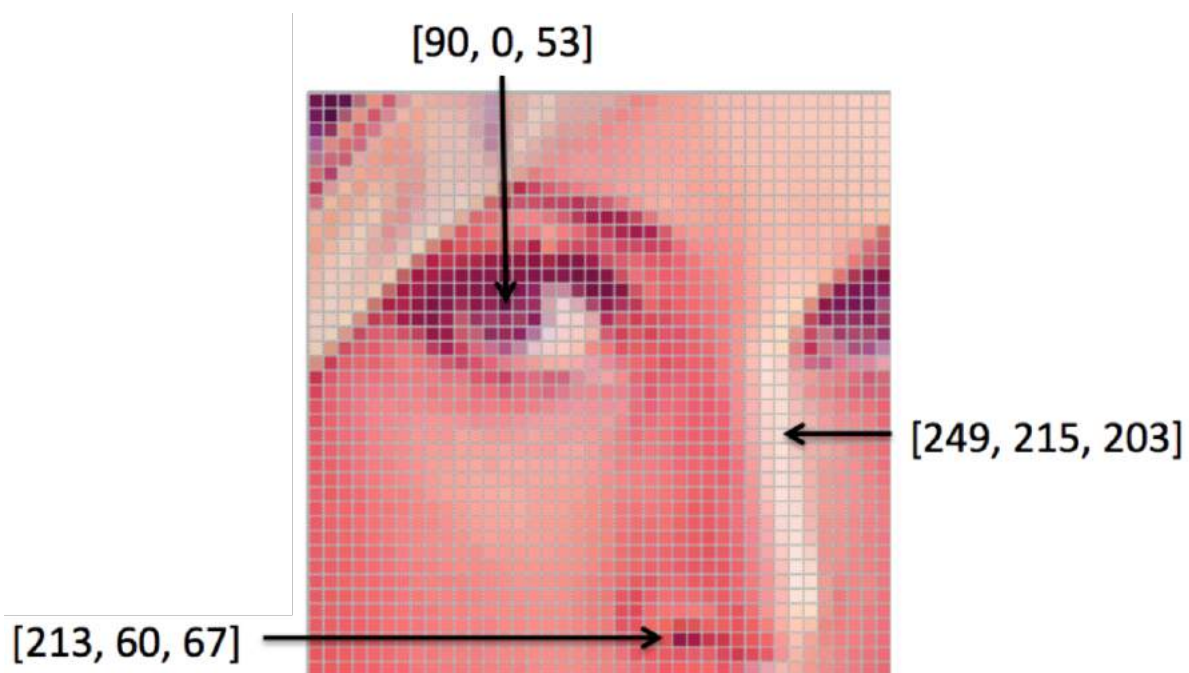
Każdy istniejący obraz można zapisać jako połączenie maksymalnie trzech kolorów: czerwonego, zielonego i niebieskiego (RGB). Dlatego, aby zapisać obraz w formacie cyfrowym, wystarczy zastosować trzy macierze - po jednej dla każdego koloru. Każda komórka takiej macierzy odpowiada wówczas jednemu pikselowi - swoim położeniem w macierzy określając pozycję danego piksela na obrazie, a wartość zapisana w komórce oznacza natężenie danego koloru w odpowiadającym jej pikselu. Dzięki zastosowaniu tej metody, po nałożeniu na siebie wszystkich trzech macierzy otrzymuje się kompletny obraz.

Najprostszym przypadkiem jest obraz czarno-biały. Wówczas można go zapisać za pomocą tylko jednej macierzy, utworzonej w sposób analogiczny do opisanego powyżej. Przykład przeniesienia obrazu czarno-białego do postaci macierzowej zaprezentowano na Rysunku 2.2, natomiast na Rysunku 2.3 przedstawiono przykładowy obraz kolorowy wraz z zestawieniem wartości konkretnych komórek trzech macierzy umożliwiających uzyskanie

określonego koloru kilku przykładowych podpisanych pikseli.



Rysunek 2.2 Czarno-biały obraz wraz z odpowiadającą mu macierzą zawierającą informacje o wypełnieniu jego pikseli [16]. W przyjętej skali oznaczenie 1 odpowiada kolorowi białemu, 0 czarnemu, a wartości pomiędzy są różnymi odcieniami szarości.



Rysunek 2.3 Przykładowe wartości pojedynczych komórek macierzy RGB i odpowiadające im piksele [29].

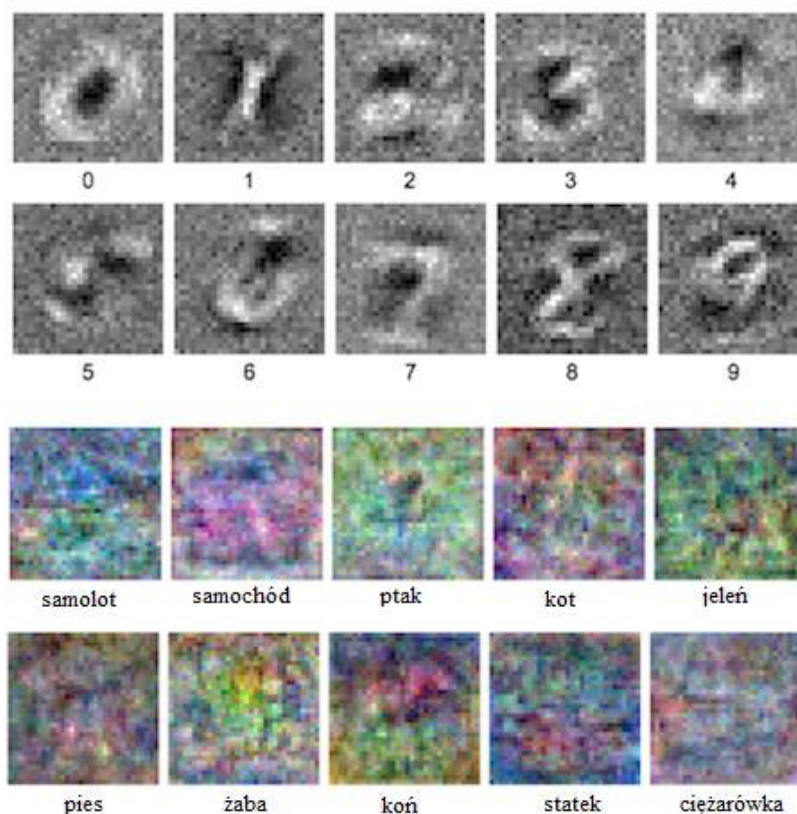
Wykorzystany w niniejszej pracy do szkolenia sieci i opisany powyżej format obrazów zbudowany z prostokątnej siatki pikseli nazywa się rastrowym. Możliwy jest również zapis obrazu w formacie wektorowym, który znajduje zastosowanie np. w opisie czcionek (w tym czcionki użytej w niniejszym tekście - dzięki grafice wektorowej, przy przybliżaniu tekstu nie następuje rozmazanie).

2.3 Sieci konwolucyjne - opis ogólny

Sieci konwolucyjne są obecnie uważane za najefektywniejszy rodzaj sieci neuronowych w kwestii analizy obrazu. Do ich najpopularniejszych zastosowań zalicza się:

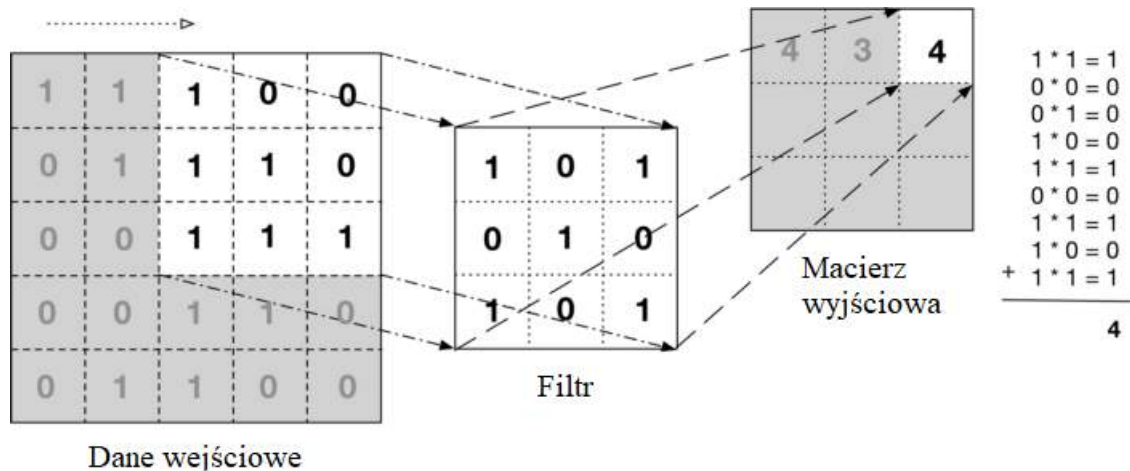
- wykrywanie i klasyfikacje obiektów na obrazach,
- nawigacja i sterowanie pojazdami autonomicznymi [4],
- generowanie obrazów, dźwięku oraz tekstu,
- zamiana mowy w tekst.

Przewagę w wykonywaniu wyżej wymienionych zadań zapewniło im wprowadzenie nowego typu warstwy: warstwy konwolucyjnej. Warstwa konwolucyjna pod względem matematycznym jest podobna do warstwy gęstej (czyli każdy jej neuron posiada komplet połączeń z każdym neuronem warstwy poprzedzającej), natomiast pod względem architektonicznym jest niejako jej rozwinięciem. W zadaniu klasyfikacji obrazów, zbudowanie sieci neuronowej złożonej tylko z jednej warstwy gęstej skutkuje uzyskaniem wag układających się w uśrednione wartości wszystkich obrazów danej kategorii, co pozwala uzyskać przyzwoite rezultaty dla najprostszych zadań, ale przy zadaniach bardziej złożonych efekty są już znacznie słabsze. Można to zaobserwować na Rysunku 2.4, gdzie zestawiono ze sobą wagi jednowarstwowej sieci dla dwóch bardzo popularnych baz danych: MNIST, zawierającej ręcznie pisane monochromatyczne cyfry, oraz CIFAR-10, zawierającej kolorowe zwierzęta i obiekty.



Rysunek 2.4 Zestawienie wag sieci jednowarstwowej opartej na warstwie gęstej do w identyfikacji obiektów bazy MNIST (rzędy górne) i CIFAR-10 (rzędy dolne) [14].

Zadaniem konwolucji jest wykrywanie konkretnych cech analizowanych obrazów. Dokonuje tego za pomocą nakładania odpowiednich, z reguły niewielkich obszarowo filtrów (z ang. kernel). Działają one na zasadzie sumowania sąsiednich wartości macierzy wejściowej przemnożonych przez odpowiednie wagi, tak jak na przykładzie na Rysunku 2.5.



Rysunek 2.5 Przykład działania konwolucji [30].

Oznacza to, że dla przykładowego zadania rozpoznania samochodu [1], sieć nie będzie próbowała nauczyć się szczegółowego rozpoznawania całego pojazdu, a jego cech charakterystycznych (np. kół, drzwi, szyb) i na ich podstawie sklasyfikować zadany obraz jako zawierający samochód lub samochodu niezawierający.

W przypadku sieci głębokiej często stosuje się więcej niż jedną warstwę konwolucyjną. Wówczas, kiedy pierwsza warstwa konwolucyjna znajduje cechy charakterystyczne na obrazie, kolejna wyszukuje wzorców pośród cech wyciągniętych przez warstwę ją poprzedzającą. Dzięki temu, kiedy pierwsza warstwa spełnia rolę wyszukiwania prostych wzorów, takich jak krawędzie, kolejna warstwa może nauczyć się wyszukiwać już kombinacje tych wzorów. Prowadzi to do sytuacji, w której dzięki dodawaniu kolejnych warstw konwolucyjnych możliwym jest nauczanie sieci wykrywania skomplikowanych i złożonych cech.

Na proces konwolucji wpływają 3 hiperparametry:

- krok (stride) - liczba komórek, o które przesuwa się okno filtru,
- padding - określa zachowanie na komórkach skrajnych,
- głębokość (depth) - liczba filtrów zastosowanych w warstwie.

W dziedzinie uczenia maszynowego przyjęto, że hiperparametry odpowiadają za określenie przebiegu procesu uczenia, a parametry są powstałe w wyniku procesu uczenia i opisują wytworzoną sieć neuronową.

2.4 Sieci konwolucyjne - budowa

Obecnie stosowane sieci konwolucyjne, w tym wykorzystane w poniższej pracy, składają się głównie z następujących typów warstw [25]:

- wejściowa,
- konwolucyjna (z aktywacją),
- łącząca (poolingu),
- gęsta (fully-connected lub dense),
- typu dropout,
- wyjściowa.

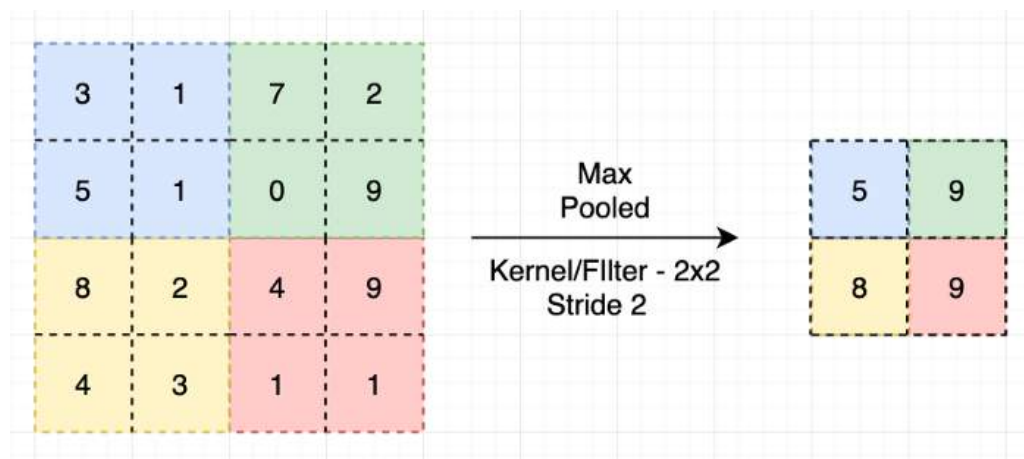
Wyróżniającą ten typ sieci warstwa konwolucyjna została już opisana osobno w Rozdziale 2.3 Sieci konwolucyjne - opis ogólny. Pozostałe warstwy mają następujące cechy i zadania:

2.4.1 Warstwa wejściowa

Warstwa wejściowa zawiera informacje o obrazie w formie macierzy pikseli zadanego obrazu.

2.4.2 Warstwa łącząca

Wykonuje operację poolingu. Operacja poolingu pozwala na zmniejszanie wymiaru macierzy poprzez wybieranie konkretnych cech obrazu i zachowanie ich. Najczęściej stosowany jest max-pooling, czyli wybranie największej liczby z danego obszaru i przeniesienie jej do nowej macierzy. Przykład takiej operacji zaprezentowano na Rysunku 2.6.



Rysunek 2.6 Przykład max-poolingu [19]

Proces poolingu określa się za pomocą 2 hiperparametrów:

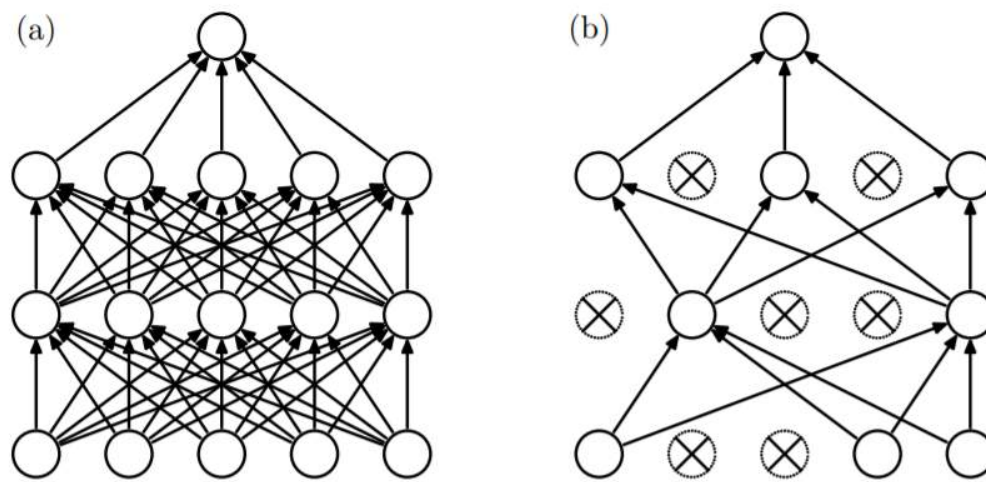
- krok (stride) - podobnie jak w procesie konwolucji, opisuje on liczbę komórek, o które przesuwa się okno filtru,
- filtr (kernel) - określa rozmiar okna filtru.

2.4.3 Warstwa gęsta

Zbudowana jest z neuronów, wag i biasów. Każdy jej neuron posiada komplet połączeń z każdym neuronem warstwy poprzedzającej. Umieszczona z reguły pod koniec sieci, umożliwia rozpoczęcie klasyfikacji na bazie danych dostarczonych przez poprzednie warstwy. Wykorzystuje w tym celu takie funkcje jak softmax (w przypadku klasyfikacji wieloklasowej, czyli między więcej niż dwoma klasami) lub regresję logistyczną (dla klasyfikacji binarnej, czyli między dwoma klasami).

2.4.4 Warstwa typu dropout

Warstwa wykorzystana do generalizacji i zwalczania przetrenowania, działająca poprzez usuwanie niektórych połączeń [27], (Rys. 2.7).



Rysunek 2.7 a) Sieć niewykorzystująca dropoutu, b) Sieć wykorzystująca dropout [27].

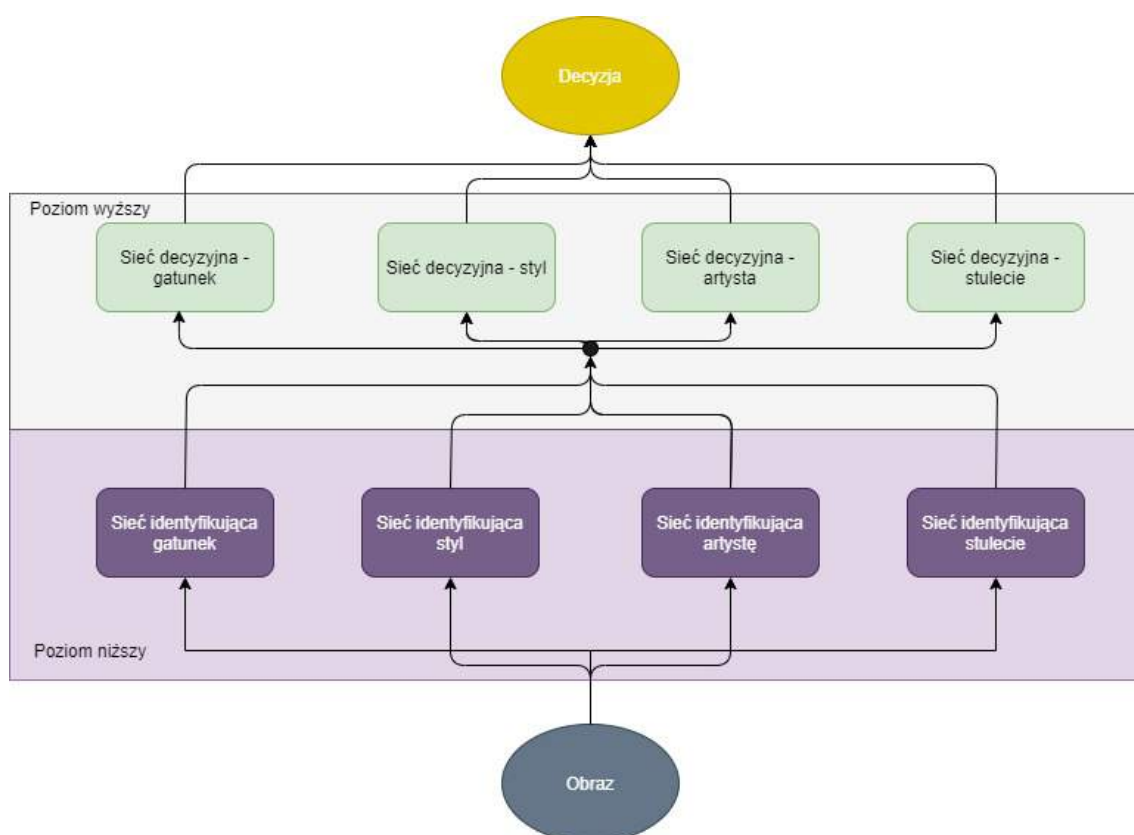
2.4.5 Warstwa wyjściowa

Ostatnia warstwa sieci konwolucyjnej, zawiera nazwy klas spośród których następowała klasyfikacja.

Rozdział 3

Architektura systemu

Zaprojektowano system, którego zadaniem była identyfikacja: stylu, gatunku, artysty oraz wieku powstania obrazu malarskiego. Warto przypomnieć, że w przypadku obrazów malarskich styl odnosi się do sposobu przedstawienia (impresjonizm, kubizm itp.), natomiast gatunek - do tego, co jest na obrazie przedstawione (portret, krajobraz itp.). Zaprojektowany system był złożony z 8 sieci rozłożonych na dwa poziomy. Na poziomie niższym składały się cztery sieci konwolucyjne mające za zadanie rozpoznanie odpowiednio: gatunku, stylu, artysty oraz stulecia, w którym powstało dzieło poprzez bezpośrednią analizę obrazu. Zadaniem poziomu wyższego było podjęcie decyzji na bazie danych wyjściowych każdej sieci poziomu niższego, czyli przewidywanych przez nie prawdopodobieństw wystąpienia każdej klasy z osobna. Sieci poziomu wyższego nie miały żadnej bezpośredniej styczności z obrazem, a jedynie z przewidywaniami klas poziomu niższego, zgodnie z Rysunkiem 3.1.



Rysunek 3.1 Architektura systemu.

Decyzja o podejściu dwupoziomowym motywowana była pomysłem, żeby podczas podejmowania ostatecznej decyzji np. dotyczącej twórcy danego dzieła uwzględnić również kontekst, czyli informacje o gatunku, stylu i wieku powstania zakładając, że usprawni to proces decyzyjny.

3.1 Sieci poziomu niższego

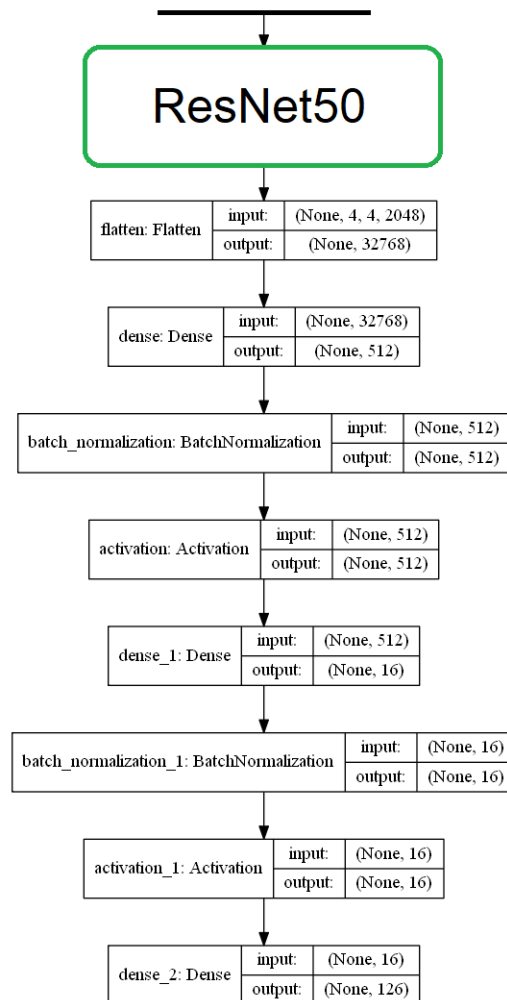
Sieci poziomu niższego oparto na zaprezentowanej w 2015 roku architekturze ResNet50 [11]. Jest ona jedną z najpowszechniej stosowanych sieci konwolucyjnych w zadaniach klasyfikacji obrazów i w wielu z nich osiąga wyniki na poziomie człowieka [1]. Nazwa pochodzi od angielskiego Residual Network, co można przetłumaczyć jako sieć szczątkową. ResNet50 jest jedną z możliwych sieci szczątkowych, końcowy człon swojej nazwy zawdzięczająca liczbie 50 warstw, z których się składa. Oprócz ResNet50 popularnymi odmianami są również m. in. ResNet18 i Resnet152. Idea sieci szczątkowej polega na wprowadzeniu połączeń skrótowych, umożliwiających pominięcie kilku warstw. Dzięki temu zabiegowi unika się problemu zanikających gradientów, który przed wprowadzeniem sieci szczątkowych był kluczową przeszkodą w stosowaniu zbyt głębokich sieci neuronowych [11].

Biblioteki Pythona, takie jak Keras, umożliwiają bezpośrednie wczytanie modelu ResNet50 uprzednio przeszkolonego (ang. pre-trained) na bazie danych z serwisu ImageNet, która zawiera ponad 14 milionów zdjęć rozłożonych na ponad 20 tysięcy klas i jest powszechnie stosowana do oceny komputerowych systemów wizyjnych [6].

Wykorzystaną architekturę ResNet50 wzbogacono o dodatkowe warstwy wyjściowe, mające za zadanie dokonanie poprawnej klasyfikacji na bazie cech wyciągniętych z obrazu za pomocą warstw ResNet50. Zastosowano w tym celu warstwy:

- Spłaszczającą (z ang. Flatten - konwertującą macierze do wektorów),
- Normalizację wsadową (z ang. Batch Normalization - skalująca i centrująca),
- Gęstą,
- Aktywacyjną.

Ustawione zgodnie z Rysunkiem 3.2.



Rysunek 3.2 Końcowe warstwy sieci do identyfikacji artysty.

Ponieważ modele wszystkich sieci różniły się tylko liczbą neuronów warstwy wyjściowej (równej liczbie klas każdej kategorii), dla przykładu w Załączniku 1 został przedstawiony kompletny graf architektury sieci do identyfikacji gatunku.

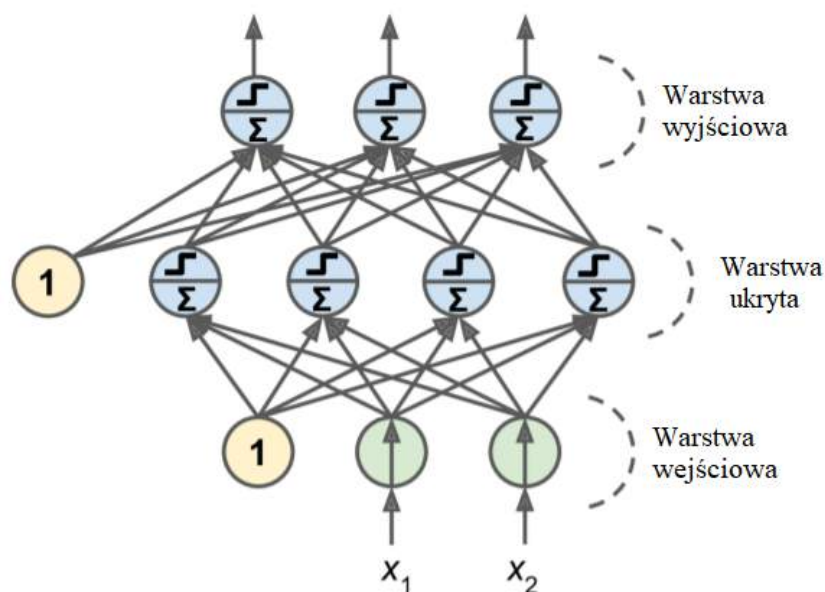
3.2 Sieci poziomu wyższego

Do podjęcia decyzji na poziomie wyższym wypróbowane zostały cztery typy sieci neuronowych oraz trzy klasyfikatory:

- sieć z jedną warstwą ukrytą,
- sieć z dwiema warstwami ukrytymi,
- sieć z trzema warstwami ukrytymi,
- sieć typu perceptron wielowarstwowy,
- klasyfikator - regresja logistyczna,
- klasyfikator - maszyna wektorów nośnych (SVM),
- klasyfikator - las losowy.

3.2.1 Perceptron wielowarstwowy

Perceptron wielowarstwowy jest rozwinięciem opisanego w podrozdziale 1.2.3 perceptronu. Składa się z warstwy wejściowej, przynajmniej jednej warstwy ukrytej oraz warstwy wyjściowej, zgodnie z Rysunkiem 3.3.

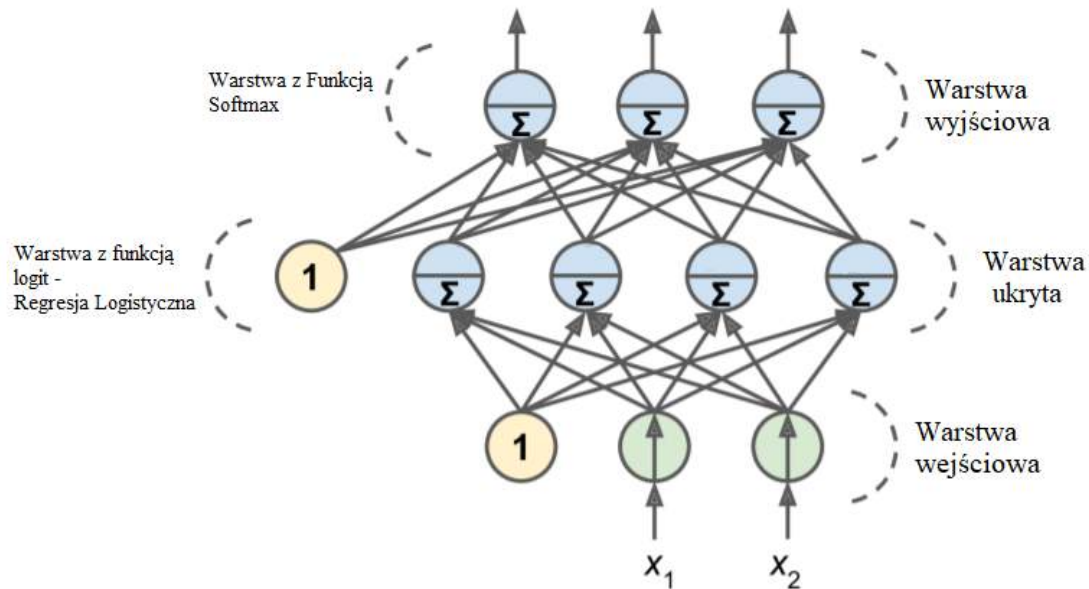


Rysunek 3.3 Perceptron wielowarstwowy [6].

W przypadku zastosowania więcej niż jednej warstwy ukrytej można mówić o jednym z pierwszych i najprostszych typów głębokich sieci neuronowych.

3.2.2 Regresja logistyczna

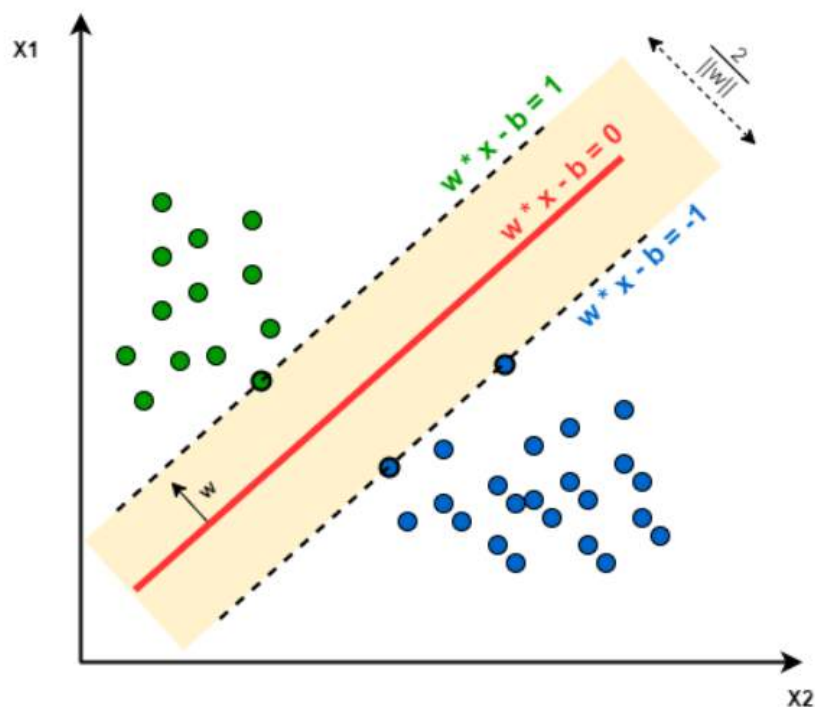
Klasyfikator oparty na regresji logistycznej zbudowany jest analogicznie do perceptronu - może być zapisany w postaci sieci neuronowej o jednej warstwie ukrytej (Rys. 3.4). Jego typowym zadaniem jest klasyfikacja binarna, czyli między dwoma klasami - w celu rozwiązania problemu klasyfikacji wieloklasowej zastosowano regresję logistyczną wielomianową. Opiera się ona na funkcji typu softmax.



Rysunek 3.4 Regresja logistyczna wielomianowa jako sieć neuronowa [6].

3.2.3 Maszyna wektorów nośnych

Maszyna wektorów nośnych (SVM), w swojej podstawowej (liniowej) formie usiłuje znaleźć linię, która maksymalizuje oddzielenie między punktami dwóch klas na dwumiarowej płaszczyźnie (Rys. 3.5).



Rysunek 3.5 Podział punktów przy użyciu liniowej maszyny wektorów nośnych (problem dwuwymiarowy) [2]

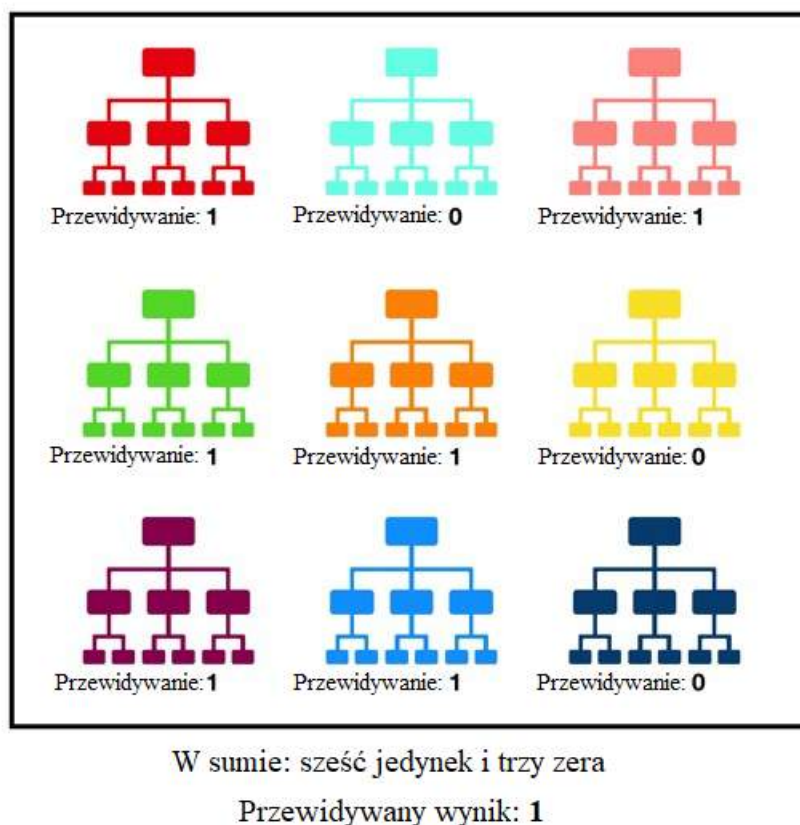
Generalizując, dla przypadków wielowymiarowych zadaniem jest znalezienie takiej hiperpłaszczyzny, która zmaksymalizuje oddzielenie punktów należących do różnych klas

w n -wymiarowej przestrzeni. Punkty najbliższe wyznaczonej hiperpłaszczyzny nazywa się wektorami nośnymi [2].

Dla klasyfikacji wieloklasowej przy użyciu maszyny wektorów nośnych stosuje się podejścia typu jeden vs jeden, gdzie stosuje się jeden binarny klasyfikator na każdą parę klas, lub jeden kontra reszta, gdzie stosuje się binarny klasyfikator dla każdej klasy.

3.2.4 Las losowy

Las losowy polega na użyciu wielu różnych drzew decyzyjnych, których przewidywania są zsumowane i na bazie większości podejmowana jest decyzja (Rys. 3.6).



Rysunek 3.6 Przykładowy las losowy w zadaniu klasyfikacji binarnej [31].

Jedną z głównych zalet lasu losowego jest fakt, że drzewa decyzyjne osłaniają się od pojedynczych, nawet najbardziej kardynalnych błędów. Ponieważ decyzja jest podejmowana na bazie większości, błąd na wyjściu pojedynczego drzewa nie spowoduje błędu na wyjściu całego lasu.

Rozdział 4

Baza danych

Jako bazę danych przyjęto zbiór obrazów z serwisu WikiArt.org, składającą się z 81 444 obrazów o łącznej wadze 31,4 GB. Baza ta została udostępniona w serwisie archive.org. Jest ona podzielona na 27 folderów nazwanych według stylów, z których wywodzące się zawarte w nich obrazy. Dodatkowo, w skład bazy danych wchodził plik wclasses.csv będący tabelą z podpisanymi kolejno: ścieżkami plików, artystą, gatunkiem oraz stylem każdego obrazu w postaci numerycznej (Rys. 4.1). W celu rozkodowania wykorzystano plik classes.php, który zawierał informacje odnośnie tego, która liczba odpowiada któremu artyście, gatunkowi i stylowi.

	file	artist	genre	style
0	Realism/vincent-van-gogh_pine-trees-in-the-fen...	22	133	161
1	Baroque/rembrandt_the-angel-appearing-to-the-s...	20	136	144
2	Post_Impressionism/paul-cezanne_portrait-of-th...	16	135	160
3	Impressionism/pierre-auguste-renoir_young-girl...	17	131	152
4	Romanticism/ivan-aivazovsky_morning-1851.jpg	9	139	163
...
81439	Naive_Art_Primitivism/mary-fedden_butterflies-...	0	139	155
81440	Naive_Art_Primitivism/arman-manookian_watercol...	0	139	155
81441	Naive_Art_Primitivism/andre-bauchant_exotic-fl...	0	139	155
81442	Naive_Art_Primitivism/ivan-generalic_cows-in-a...	0	139	155
81443	Naive_Art_Primitivism/natalia-goncharova_sheep...	0	139	155

81444 rows × 4 columns

Rysunek 4.1 Zawartość pliku wclasses.csv.

Większość nazw plików składała się z: autora, tytułu obrazu oraz roku wykonania. Informację potrzebną do podpisania obrazów dla sieci identyfikującej stulecie wyciągnięto bezpośrednio z nazwy pliku i przekonwertowano z postaci konkretnego roku wykonania na wiek wykonania.

Jak można zaobserwować na wycinku zaprezentowanym na ostatnich wierszach ramki danych z Rysunku 4.1, niektórzy artyści byli wspólnie sklasyfikowani jako 0, co oznaczało Artystę Nieznanego. Po zbadaniu paru przypadków okazało się, że dane artysty zawar-

te w nazwie pliku lepiej oddawały rzeczywistą sytuację, więc w przyszłych badaniach i szkoleniach to właśnie je wykorzystywano.

Nazwy klas przetłumaczono na język polski. Zestawienie przykładowego obrazu z przysługującymi mu klasami wszystkich kategorii zamieszczono na Rysunku 4.2.

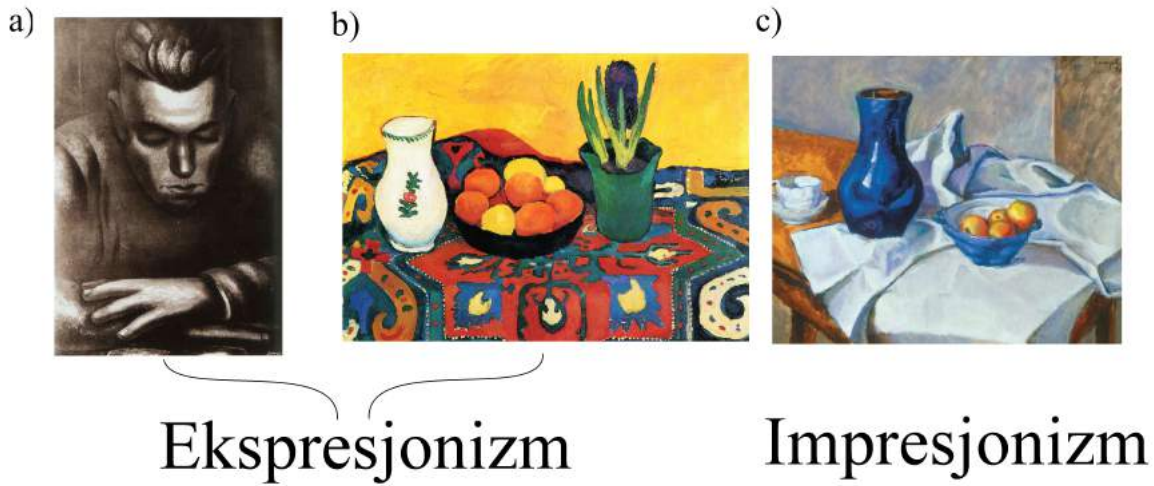


Artysta	Styl	Gatunek	Wiek
Vincent Van Gogh	Post Impresjonizm	Pejzaż Miejski	XIX

Rysunek 4.2 Przykładowy obraz i jego klasy - „Taras kawiarni w nocy”.

4.1 Wariancja wewnątrzklasowa i międzyklasowa

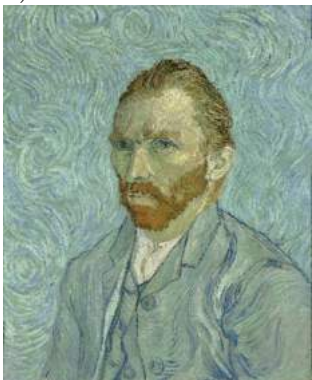








Jedną z trudności w kwalifikacji była wariancja wewnątrzklasowa i międzyklasowa, czyli duża różnica między obrazami tej samej klasy i podobieństwo do obrazów innej klasy. Kategorią, w której ten problem był nad wyraz wyraźny jest styl. Przykład zademonstrowano na Rysunku 4.3, gdzie mimo pozornego większego podobieństwa między obrazem ekspresjonistycznym (b) i impresjonistycznym (c) należą one do różnych stylów, natomiast dwa obrazy ekspresjonistyczne (a i b) nie posiadają wiele wyraźnych wizualnych punktów wspólnych.



Rysunek 4.3 Zestawienie portretu ekspresjonistycznego (a) z ekspresjonistyczną martwą naturą (b) oraz impresjonistyczną martwą naturą (c).

Wynika to między innymi z faktu, że rozróżnianie obrazów malarskich przez człowieka odbywa się na bazie emocji i uczuć, jakie obrazy w nim wzbudzają, a także z informacji pobocznych, takich jak miejsce i czas powstania. Zwłaszcza pierwsza część - emocje i uczucia - są na razie niemożliwe do zaprogramowania.

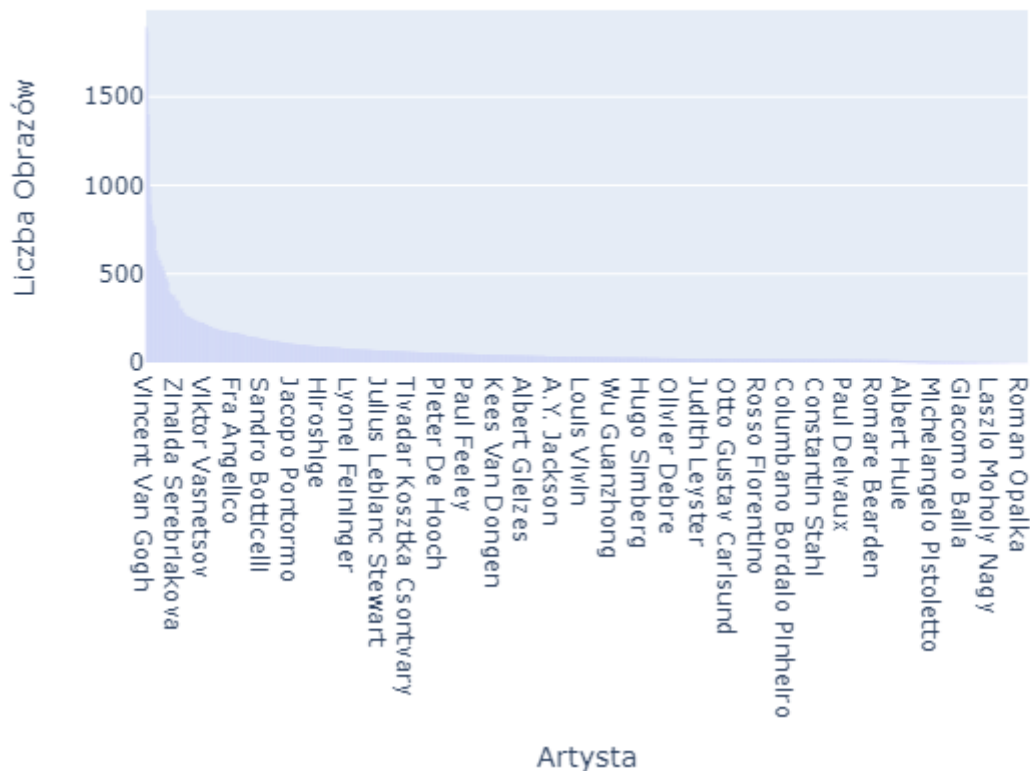
Tabela 4.1 Obrazy najpopularniejszych stylów i gatunków, podpisane według ich autorów

	Portret	Krajobraz	M. Rodzajowe
Impresjonizm	a) 	b) 	c) 
	Vincent van Gogh	Claude Monet	Claude Monet
Realizm	d) 	e) 	f) 
	Vincent van Gogh	Vincent van Gogh	Vincent van Gogh
Romantyzm	g) 	h) 	i) 
	Girodet-Trioson	Ivan Aivazovsky	Ivan Aivazovsky

Na zestawieniu w Tabeli 4.1 można zaobserwować również spore różnice wśród obrazów pojedynczych artystów - ludzkie postacie przedstawione przez van Gogha na obrazach a), d) i f) nie wydają się mieć wiele cech wspólnych w sposobie przedstawienia, a należy pamiętać, że malował on również dzieła ludzi nieprzedstawiające - np. obraz c). Obraz f) jest podpisany jako malarstwo rodzajowe, mimo widocznej na nim postaci ludzkiej i twarzy wyraźniejszej niż na portrecie d). Obrazy h) oraz i) sprawiają wrażenie jednego gatunku, jednak należą do odpowiednio krajobrazu i malarstwa rodzajowego.

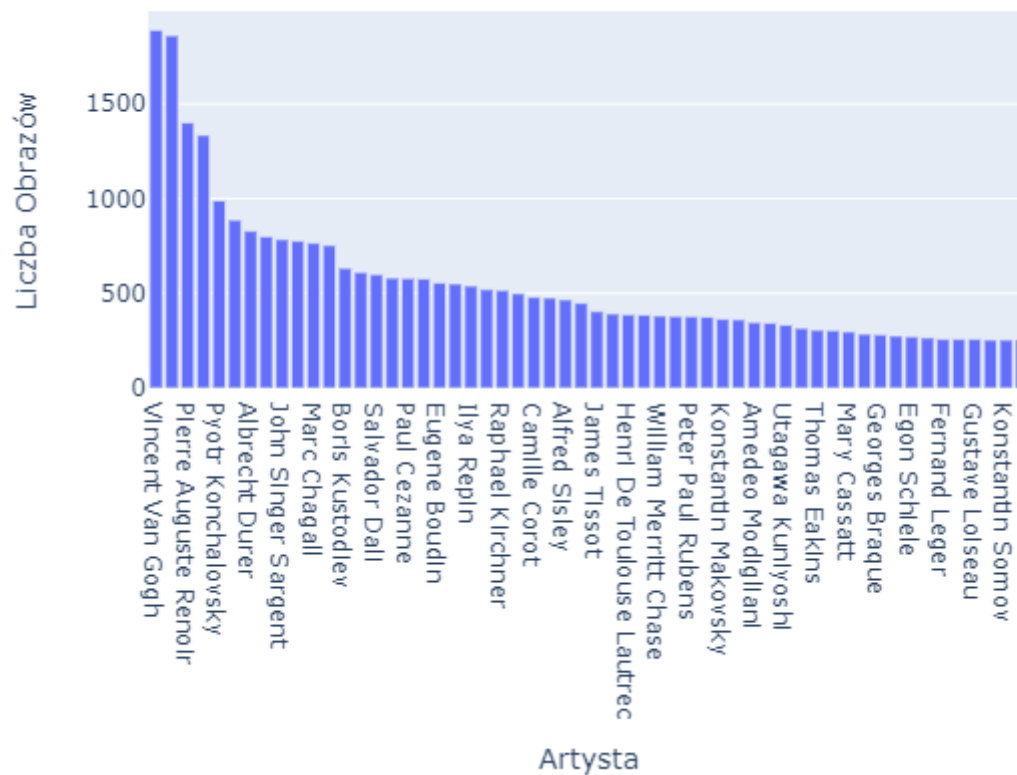
4.2 Niezbalansowanie bazy danych

Innym utrudnieniem było niezbalansowanie bazy danych, czyli wysoka rozbieżność między liczbami obrazów poszczególnych klas dla każdej kategorii. Na Rysunku 4.4 zamieszczono rozkład liczebności obrazów wszystkich artystów.

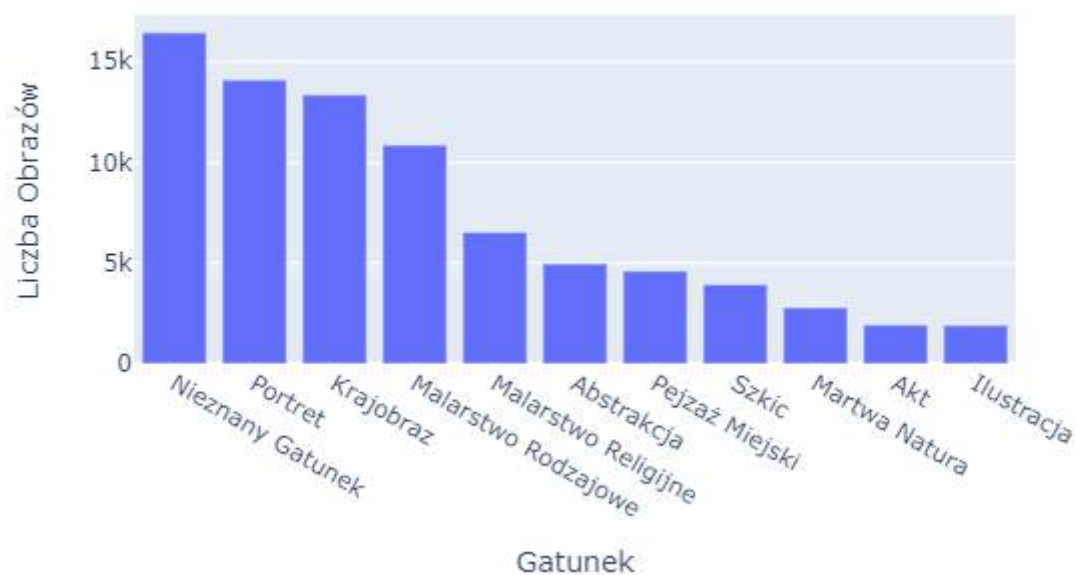


Rysunek 4.4 Rozkład liczebności obrazów różnych artystów z przykładowymi nazwiskami na osi odciętych.

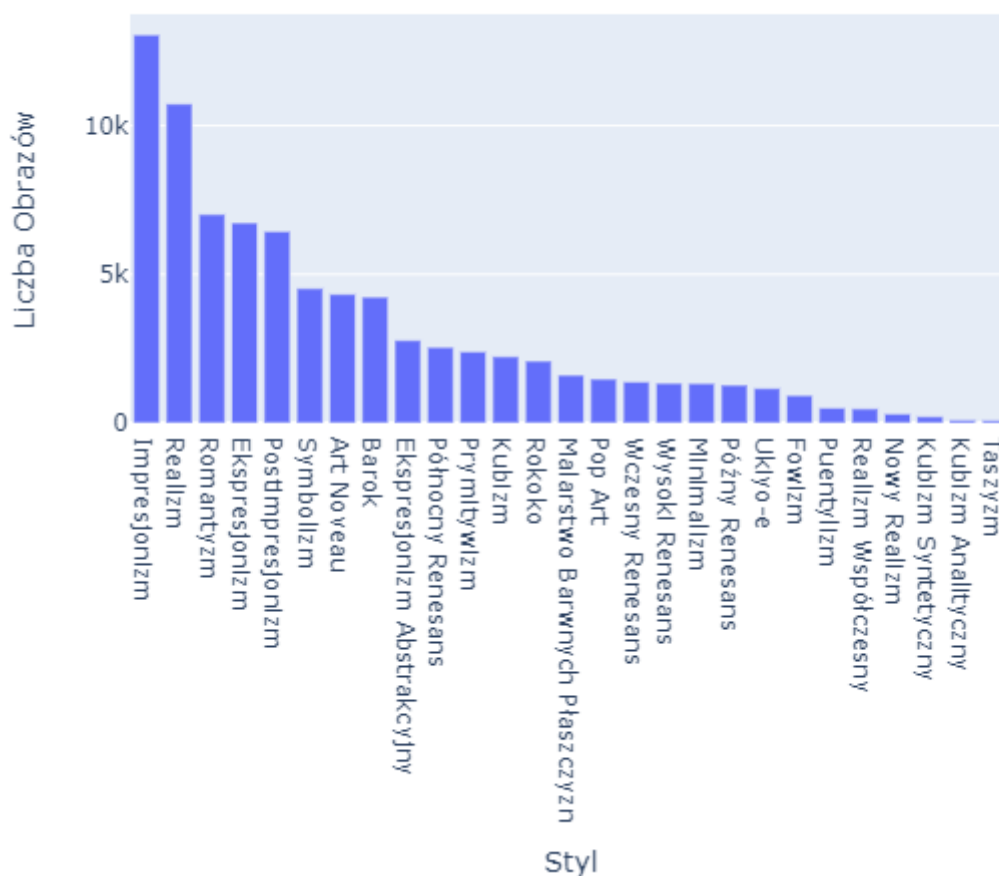
Przytłaczająca większość artystów ma bardzo niewiele obrazów w bazie danych - średnia wynosi 72,78, natomiast mediana wynosi 30. Ostatecznie do szkolenia przyjęto wyłącznie artystów z liczbą obrazów przekraczającą 250 - liczebność klas po dokonaniu tej selekcji pokazuje Rysunek 4.5



Rysunek 4.5 Liczebność obrazów przykładowych artystów - artyści o liczbie obrazów przekraczającej 250.



Rysunek 4.6 Liczebność obrazów poszczególnych gatunków.



Rysunek 4.7 Liczebność obrazów poszczególnych stylów.



Rysunek 4.8 Liczebność obrazów poszczególnych wieków.

Jednym ze sposobów pozwalających osiągać dobre rezultaty na niezbalansowanych bazach danych jest wprowadzenie wag klas. Wagi klas oblicza się dla każdej z klas z osobna za pomocą Wzoru 4.1.

$$W_j = \frac{L_w}{L_k \cdot L_j} \quad (4.1)$$

gdzie

W_j - waga danej klasy j ,

L_w - liczba wszystkich próbek w bazie danych,

L_k - liczba klas,

L_j - liczba próbek danej klasy j .

Większą wagę otrzymują klasy o mniejszej reprezentacji w zbiorze. Wówczas funkcja kosztu przyjmuje dla nich większą wartość, a algorytm nauczania jest bardziej skupiony na poprawie wyników dla niedoreprezentowanych klas zamiast na ignorowaniu ich.

Innym ważnym aspektem jest zastosowanie podczas podziału na podzbiory stratyfikacji (z ang. stratify), czyli zapewnienia takiego samego proporcjonalnego udziału każdej klasy w zbiorach treningowym, walidacyjnym i testowym. Dzięki temu nie dojdzie do sytuacji, w której któraś z klas o małej liczebności jest w zbiorze nieobecna lub obecna w zbyt małej liczebności.

Rozdział 5

Szkolenie sieci

Do szkolenia sieci wykorzystano komputer z kartą graficzną nVidia GeForce GTX 1050. W kwestii oprogramowania zdecydowano się na Python w środowisku Jupyter Notebook z wykorzystaniem bibliotek: TensorFlow, NumPy, Matplotlib, Plotly, pandas scikit-learn. Aby móc użyć karty graficznej do szkolenia sieci skorzystano z pakietów CUDA oraz CuDNN.

5.1 Sieci niższego rzędu

Idea przyświecająca szkoleniu sieci neuronowej jest następująca: obrazy z bazy danych dzieli się na trzy zbiory: **treningowy**, **walidacyjny** oraz **testowy**. Zbiór treningowy składa się z obrazów, które są pokazywane szkolonej sieci do analizy. Zbiór walidacyjny to obrazy niepokazywane bezpośrednio sieci podczas szkolenia, a wykorzystywane do dostrojenia hiperparametrów. Zbiór testowy to również obrazy niepokazywane sieci podczas szkolenia, ale wykorzystywane do oceny jakości sieci już po zakończeniu szkolenia. W niektórych przypadkach, zwłaszcza dla baz danych o niewielkiej liczebności, w roli zbioru testowego wykorzystuje się zbiór walidacyjny, jest to jednak unikane ze względu na możliwość wyszkolenia sieci do dobrego rozpoznawania obrazów ze zbiorów treningowego i walidacyjno-testowego, a słabej generalizacji na obrazach spoza tych zbiorów. W zależności od konkretnego przypadku przyjmuje się różne proporcje podziału na poszczególne zbiory - typowe wartości to 70:15:15 dla podziału na zbiór treningowy, testowy i walidacyjny oraz 70:30 dla podziału na zbiór treningowy i testowo-walidacyjny, jednak zwłaszcza dla baz danych o wysokiej liczbie reprezentantów dla każdej klasy, przyjęcie większego procentu zawartości bazy danych dla zbioru treningowego skutkowało lepszą dokładnością klasyfikacji [23]. Istotną kwestią podziału na podzbiory jest zachowanie tej samej proporcji w liczebności poszczególnych klas, co dla niezbalansowanych baz danych wiąże się z koniecznością zastosowania stratyfikacji.

Obrazy z bazy treningowej podawane są sieci w partiach (z ang. batch) o określonej liczebności (**batch_size**), które sieć analizuje poprzez nakładanie odpowiednich filtrów i odszukuje cechy pozwalające zaklasyfikować obraz do jednej z zadanych klas.

Algorytmem umożliwiającym szkolenie sieci o wielu warstwach jest propagacja wsteczna. Propagacja wsteczna polega na wykorzystaniu spadku gradientowego przy użyciu gradientów obliczanych komputerowo. W dwóch przejściach przez sieć - jednym od warstwy wejściowej do wyjściowej, drugim od warstwy wyjściowej do wejściowej - algorytm propagacji wstecznej jest w stanie obliczyć gradient błędu sieci (zazwyczaj w postaci **funkcji kosztu**) z odniesieniem do każdego parametru. Sprowadza się to do zdolności obliczenia,

jak wartość każdej wagi i biasu powinna zostać zmieniona, aby zmniejszyć błąd [6]. Proces powtarza się przez określoną liczbą iteracji, nazywanych **epokami** szkolenia. Epoka szkolenia polega na pokazaniu sieci całego zbioru uczącego (podzielonego na partie, czyli batche). Liczba epok szkolenia może być:

- liczbą stałą,
- zależną od osiągniętej wartości funkcji kosztu - szkolenie może zatrzymywać się po osiągnięciu wymaganej jakości sieci,
- zależną od zmiany wartości funkcji kosztu - szkolenie może być zatrzymane, gdy kolejne epoki szkolenia nie zapewniają poprawy.

Obrazy mogą być poddane **augmentacji**, czyli procesowi zmiany niektórych cech obrazu za pomocą takich operacji jak m. in. rotacja, przycięcie, odbicie lustrzane, zmiana nasycenia kolorów, przeskalowanie. Proces ten ma na celu wykonanie dwóch zadań:

- zwiększenie liczebności bazy danych, co jest bardzo pożądane w przypadku małych baz danych,
- zwiększenie różnorodności bazy danych, co pozwala na zwalczanie przeuczania (z ang. overfitting).

Aby nie zatracić istotnych informacji dla danego gatunku/stylu/artysty/stulecia, w przypadku tego zadania zdecydowano się wypróbować tylko niewielką augmentację. Porównanie wyników sieci nauczanej na zbiorze poddanym augmentacji oraz z pominięciem augmentacji wykonano na przykładzie opisanej w podrozdziale 5.1.1 Sieci do identyfikacji gatunku.

Podczas nauczania sieci, jej jakość określa się za pomocą czterech parametrów:

- dokładności klasyfikacji dla zbioru treningowego,
- funkcji kosztu dla zbioru treningowego,
- dokładności klasyfikacji dla zbioru walidacyjnego,
- funkcji kosztu dla zbioru walidacyjnego.

Przeuczanie można zaobserwować, gdy mimo poprawy wyników dla zbioru treningowego, wyniki dla zbioru walidacyjnego nie poprawiają się lub wręcz zaczynają się pogarszać [26]. Oznacza to „naukę na pamięć” zbioru treningowego zamiast wynalezienia takich cech jego obrazów, które są charakterystyczne dla danej klasy. Prowadzi to do sytuacji, w której sieć postawiona przed zadaniem klasyfikacji obrazów spoza zbioru treningowego osiąga gorsze rezultaty od sieci, która nie „poznała” tak dobrze obrazów zbioru treningowego.

Jako funkcję kosztu przyjęto kategoriową entropię krzyżową (ang. categorical cross-entropy). Jest ona rozwinięciem binarnej entropii krzyżowej, która jest techniczną nazwą dla funkcji kosztu opartej na regresji logistycznej. Kategoriowa entropia krzyżowa jest jej uogólnieniem dla problemu klasyfikacji wieloklasowej poprzez wprowadzenie funkcji typu softmax [20].

Do optymalizacji parametrów sieci stosuje się m. in. metody adaptacyjne i metody bazujące na momencie (ang. momentum-based). Przykładem łączenia powyższych dwóch metod może być wykorzystany w niniejszej pracy optymalizator ADAM. Jego nazwa pochodzi od angielskiego Adaptive Moment Estimation. Wykorzystuje on zarówno średnią kwadratów przeszłych gradientów (podobnie jak metody adaptacyjne, takie jak

RMSProp), jak i średnią przeszłych gradientów (podobnie jak metody bazujące na momencie) [6]. ADAM jest obecnie jednym z najpowszechniej stosowanych optymalizatorów w zadaniu klasyfikacji obrazów; ma niewielkie wymagania dotyczące pamięci i jest wydajny obliczeniowo.

Za jeden z najważniejszych hiperparametrów uważa się szybkość uczenia sieci (z ang. **learning rate**, stąd przyjęte powszechnie oznaczenie *lr*) [6]. Opisuje on tempo, z jakim system przystosowuje się do nowych danych. Przy zbyt wysokim tempie nauki, szkolona sieć przykłada dużą wagę do ostatnich przekazanych jej danych kosztem danych pokazanych uprzednio, co negatywnie wpływa na uogólnienie jej klasyfikacji. Wpływ działania zmiany szybkości nauczania zaprezentowano na przykładzie sieci z podrozdziału 5.1.2 Sieć do identyfikacji artysty.

W poniższej części rozdziału opisano szczegółowo procesy szkoleń poszczególnych sieci i wykonane podczas nich eksperymenty. Rezultaty dla najlepszych sieci z każdej kategorii zestawiono w Podrozdziale 5.1.5, gdzie przebadano je pod kątem również innych wskaźników (precyzji, czułości, F1) i porównano z wynikami z literatury.

Szkolenie każdej sieci niższego rzędu rozplanowano na dwa etapy:

- przeszkolenie wszystkich warstw, czyli zarówno ResNetu50, odpowiedzialnego za wyszukanie właściwości obrazu, jak i warstw wyjściowych, odpowiedzialnych za dokonanie klasyfikacji,
- zamrożenie warstw ResNetu50 i doszkolenie samych warstw wyjściowych.

5.1.1 Sieć do identyfikacji gatunku

W pierwszym etapie dokonano szkolenia wszystkich warstw. Ze względu na wprowadzanie przez nią znacznego błędu, pominięto klasę „Nieznany gatunek”, w efekcie czego szkolenie odbyło się dla 64 992 obrazów należących do 10 klas. Ponieważ przed przejściem do szkolenia właściwego planowano przetestować możliwości sieci i słuszność wybranych hiperparametrów początkowych, obrazy podzielono w stosunku 8:2 na zbiór testowy oraz walidacyjny, a do oceny jakości sieci skupiono się na dokładności uzyskiwanej na zbiorze treningowym. Ze względu na zróżnicowanie rozmiarów poszczególnych obrazów, do szkolenia ustalono ich rozmiar na 224×224 . Z uwagi na znaczną liczebność bazy danych i obawę przed utratą istotnych informacji, obrazy poddano tylko elementarnej augmentacji: losowemu przycięciu, przybliżeniu na poziomie 0,2 oraz odwróceniu w poziomie.

Jednym z kluczowych hiperparametrów do uzyskania wysokiej zdolności sieci do klasyfikacji jest `batch_size`, czyli liczebność jednej partii obrazów użytych do szkolenia sieci. Użyta karta graficzna okazała się mieć zbyt mało pamięci, by umożliwić ustawienie `batch_size` na więcej niż 8. Przyjęto taką wartość. Wykonano pierwszy etap szkolenia dla 15 epok, co dało skuteczność klasyfikacji na zbiorze treningowym na poziomie 76,74%.

Dokonano przejścia do drugiego etapu szkolenia. Zamrożenie warstw ResNetu50 wpłynęło na zmniejszenie obciążenia karty graficznej, co umożliwiło zwiększenie `batch_size` do 16. Dokonano szkolenia przez 10 epok i otrzymano skuteczność na zbiorze treningowym wynoszącą 84,43%.

Ponieważ spodziewano się, że istnieje możliwość uzyskania lepszego wyniku, zbadano inne rozwiązanie. Na obciążenie karty graficznej podczas procesu szkolenia sieci neuro nowej wpływa wiele czynników, w tym również przyjęty rozmiar obrazów wejściowych. Zmniejszenie go do 128×128 umożliwiło zwiększenie `batch_size` do 16 lub 32. Porównanie obrazów po poddaniu augmentacji i przeskalowaniu do 224×224 oraz 128×128 zamieszczono na Rysunku 5.1.



Rysunek 5.1 Przykładowe obrazy poddane augmentacji. Po lewej - w rozmiarze 224×224 , po prawej - w rozmiarze 128×128 .

Przebadano skuteczność procesu szkolenia przy użyciu tych wartości. Rezultaty zestawiono w Tabeli 5.1.

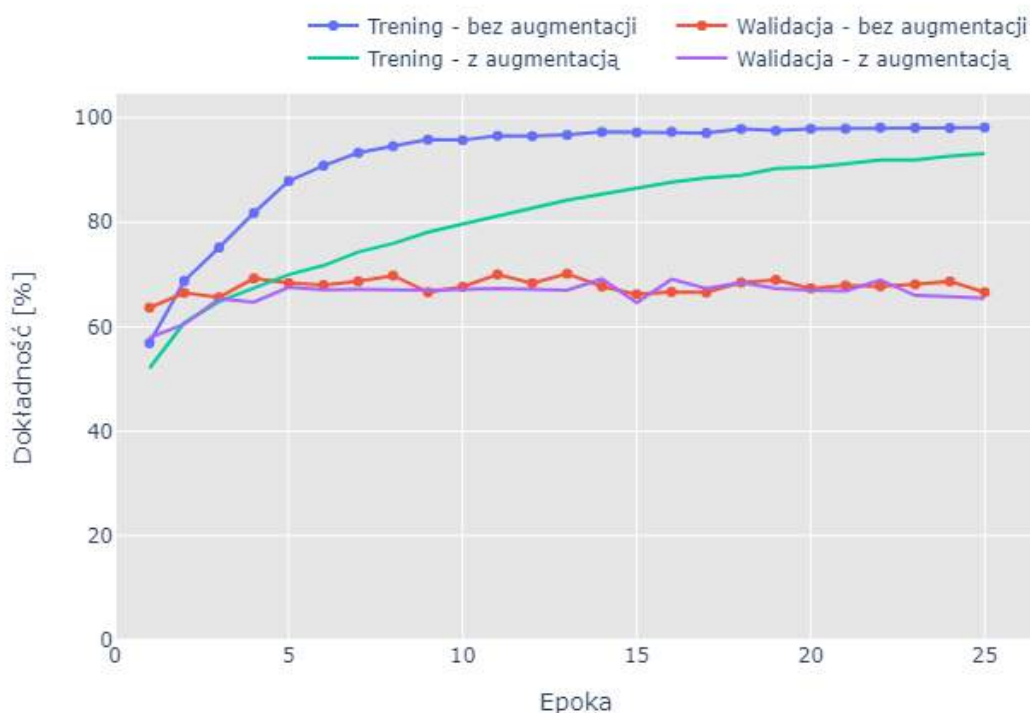
Tabela 5.1 Dokładność na zbiorze treningowym przy różnych wartościach `batch_size` dla obu etapów szkolenia.

batch_size 1. etapu	Dokładność po 1. etapie	batch_size 2. etapu	Dokładność po 2. etapie
8	76,74%	16	84,43%
16	83,05%	32	90,16%
32	89,27%	16	93,54%
		32	94,14%

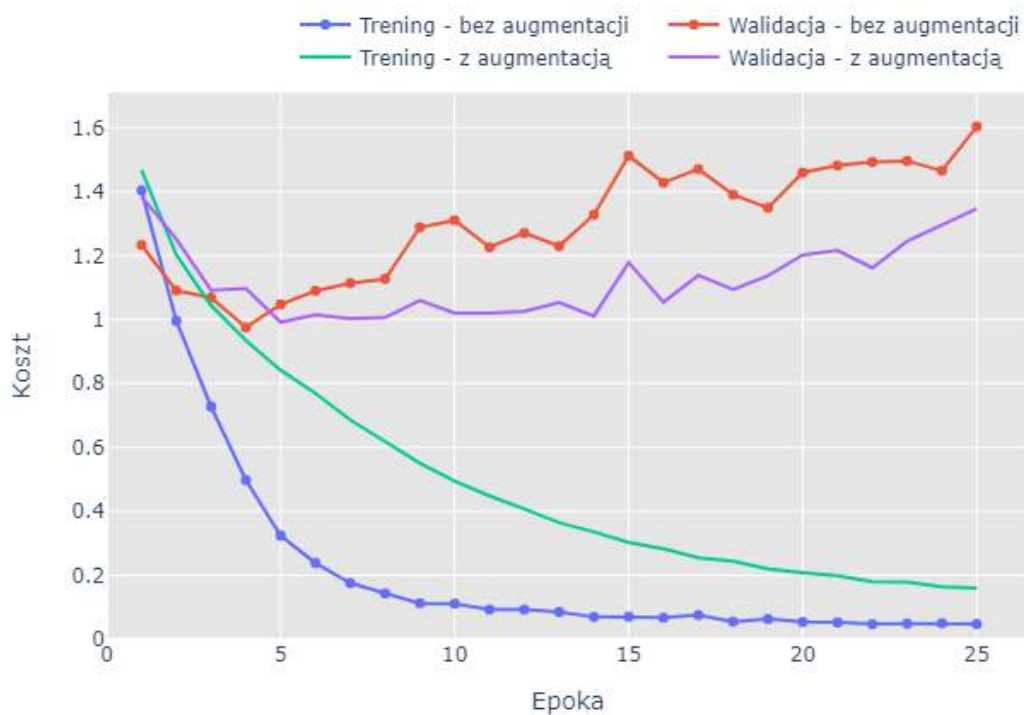
Najskuteczniejsze okazało się szkolenie przy `batch_size` wynoszącym 32 dla zarówno pierwszego jak i drugiego etapu. Dla tych ustawień dokonano dokładniejszej i dogłębniej zbadanej klasyfikacji. Bazę danych podzielono na nowo: najpierw z całości wydzielono 15% na zbiór walidacyjny, następnie z reszty wydzielono kolejne 15% (czyli 11,25% całej bazy danych) na zbiór testowy, a pozostałą część (73,75%) przeznaczono na zbiór treningowy. Tak relatywnie niski procentowy udział zbioru testowego motywowany jest niewielką

liczbą klas (10) i dobrym rozłożeniem przykładów dla każdej z nich (nawet najmniej liczna klasa w zbiorze testowym zawierała 285 obrazów). Następnie dokonano szkolenia na pierwszym etapie przez 25 epok oraz na drugim przez 15 epok. Dokonano ewaluacji na zbiorze treningowym, walidacyjnym i testowym. Dla modelu po pierwszym etapie szkolenia otrzymano 91,59% dokładności na zbiorze treningowym, 65,63 % dokładności na zbiorze walidacyjnym i 65,38% dokładności na zbiorze testowym. Dla najlepszego modelu pod kątem wartości funkcji kosztu na zbiorze walidacyjnym po drugim etapie szkolenia - czyli po 14 epokach szkolenia - rezultaty poprawiły się do 96,70% dokładności na zbiorze treningowym, 68,87% na zbiorze walidacyjnym i 69,93% dokładności na zbiorze testowym.

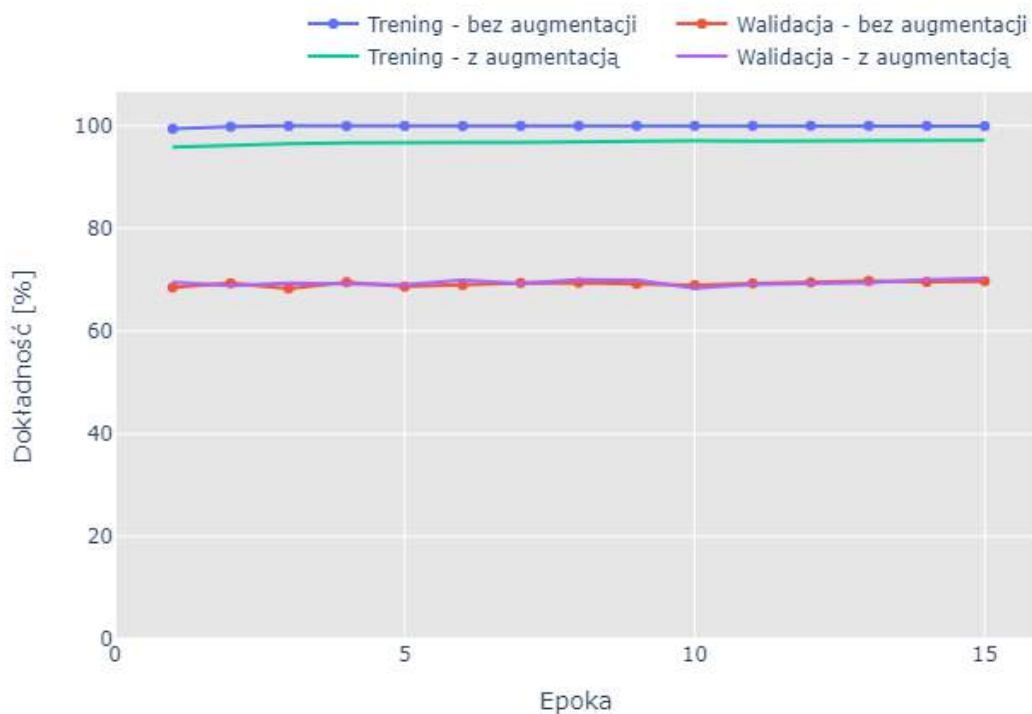
Obliczenia trwały 13 godzin 56 minut dla pierwszego etapu i 8 godzin 13 minut dla drugiego etapu, co dało 22 godziny 9 minut dla całości. Zbadano wpływ zrezygnowania z augmentacji na wynik końcowy oraz czas obliczeń. Motywowano to dużym rozmiarem bazy danych, niewielkim stopniem zaawansowania augmentacji oraz możliwą utratą istotnych informacji. Wykonano szkolenie ponownie, tym razem bez jakiegokolwiek augmentacji, a przebiegi szkoleń porównano ze sobą i zestawiono na Rysunkach 5.2, 5.3, 5.4 i 5.5



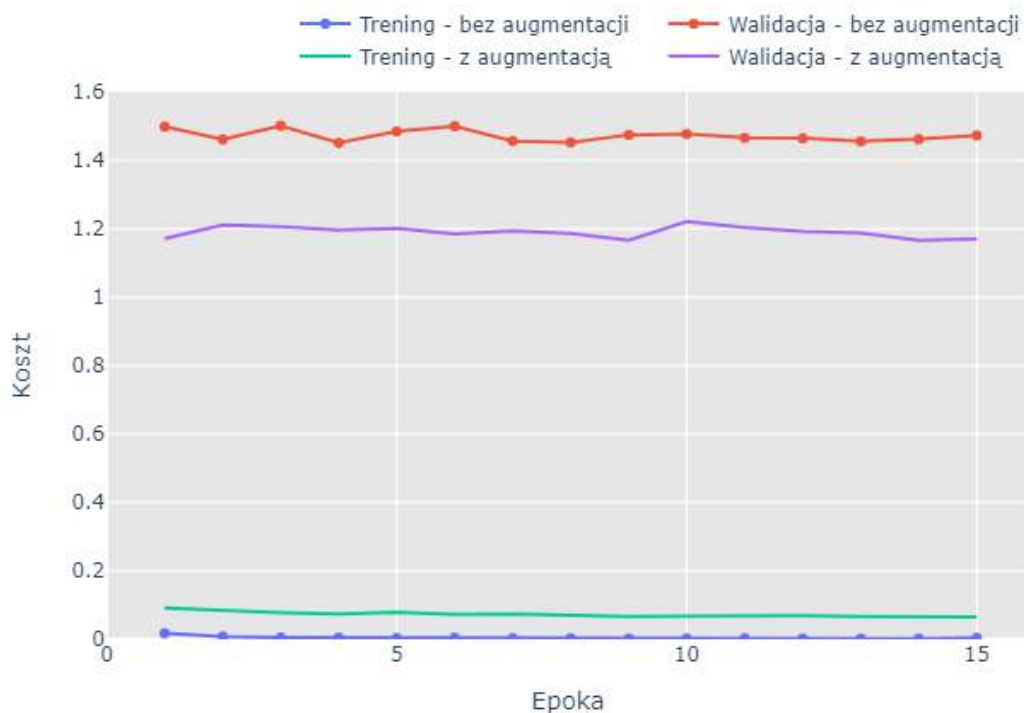
Rysunek 5.2 Dokładność sieci do identyfikacji gatunku w pierwszym etapie szkolenia.



Rysunek 5.3 Funkcja kosztu sieci do identyfikacji gatunku w pierwszym etapie szkolenia.



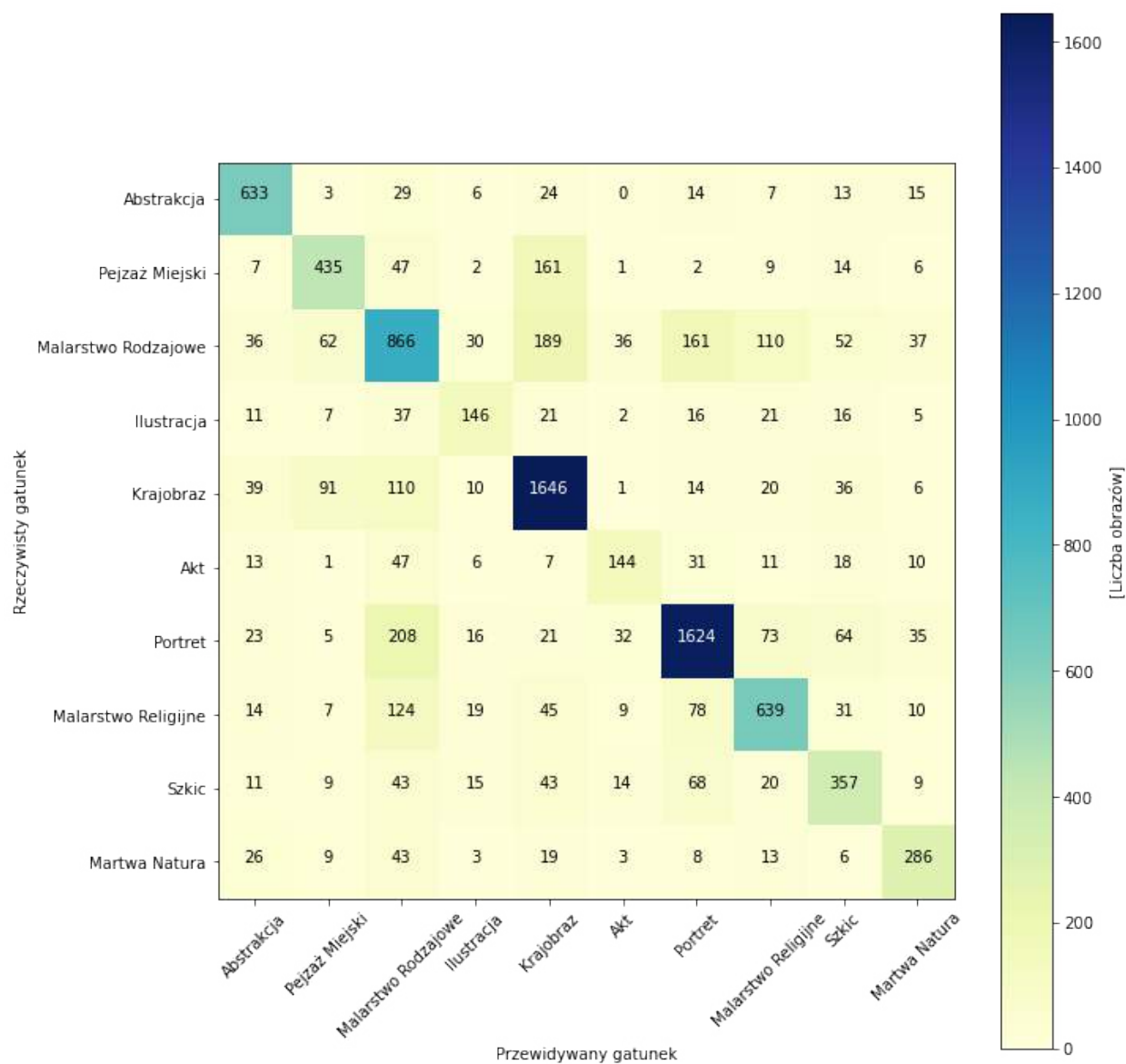
Rysunek 5.4 Dokładność sieci do identyfikacji gatunku w drugim etapie szkolenia.



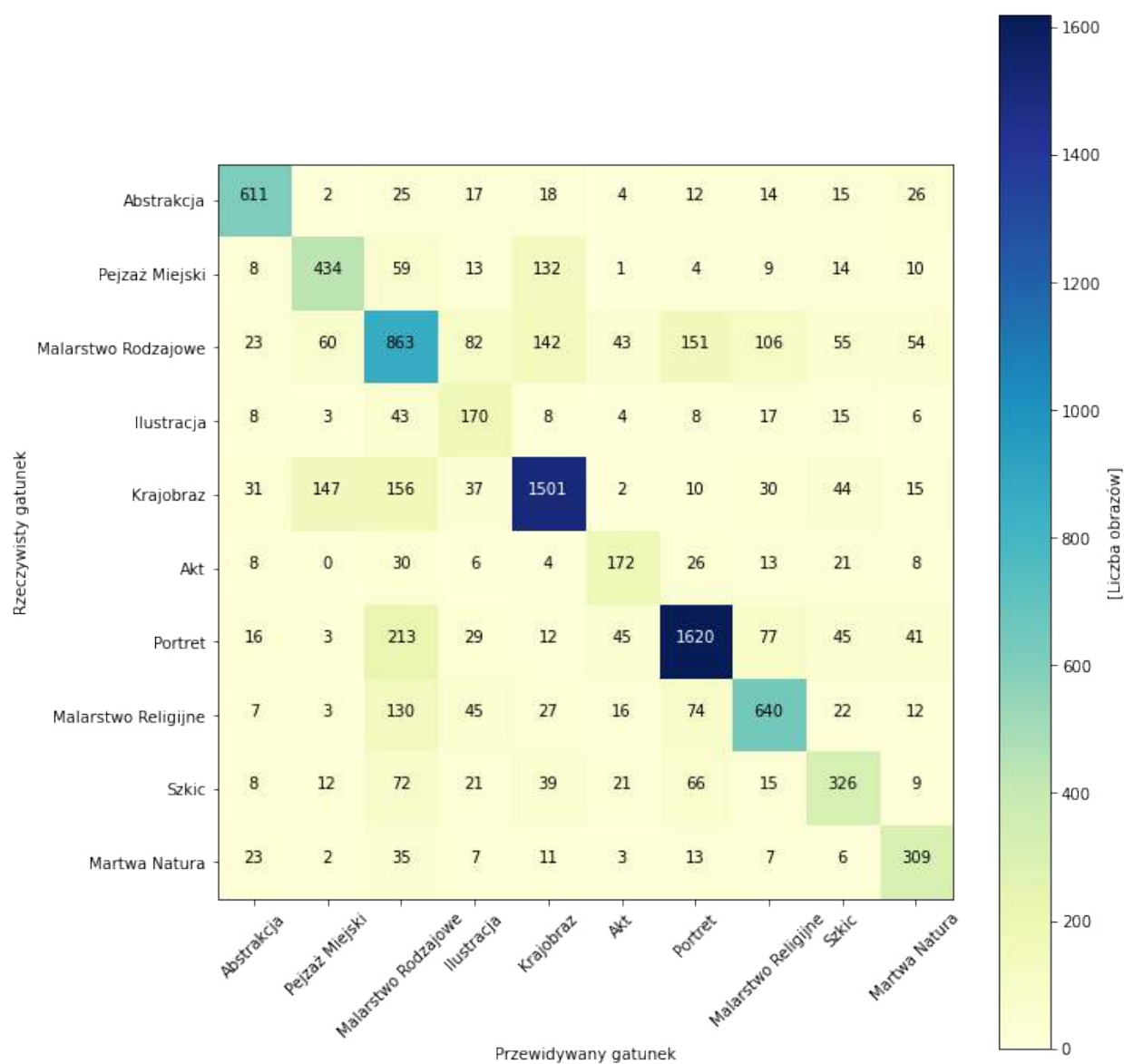
Rysunek 5.5 Funkcja kosztu sieci do identyfikacji gatunku w drugim etapie szkolenia.

Dla podejścia bez augmentacji skuteczność najlepszego modelu pod kątem funkcji kosztu na zbiorze walidacyjnym (po 10 epokach drugiego etapu szkolenia) wyniosła 99,99% dla zbioru treningowego, 69,77% dla zbioru walidacyjnego i 70,37% dla zbioru testowego. Czas obliczeń skrócił się do 21 godzin 28 minut.

Wykreślono macierze pomyłek dla obu prób (Rys. 5.6 i 5.7).



Rysunek 5.6 Macierz pomyłek - szkolenie z augmentacją.



Rysunek 5.7 Macierz pomyłek - szkolenie bez augmentacji.

Można zaobserwować, że na stosowaniu augmentacji głównie skorzystały takie gatunki jak krajobraz, natomiast straciła martwa natura oraz akt. Potwierdzają to również raporty klasyfikacji wykreślone dla zbioru testowego (Rys. 5.9 i Rys. 5.8).

	Precyzja (precision)	Czułość (recall)	F1	Liczba obrazów (support)
Abstrakcja	0.82	0.82	0.82	744
Pejzaż Miejski	0.65	0.63	0.64	684
Malarstwo Rodzajowe	0.53	0.55	0.54	1579
Ilustracja	0.40	0.60	0.48	282
Krajobraz	0.79	0.76	0.78	1973
Akt	0.55	0.60	0.57	288
Portret	0.82	0.77	0.79	2101
Malarstwo Religijne	0.69	0.66	0.67	976
Szkic	0.58	0.55	0.57	589
Martwa Natura	0.63	0.74	0.68	416
Dokładność (accuracy)			0.69	9632
Średnia	0.65	0.67	0.65	9632
Średnia ważona	0.70	0.69	0.69	9632

Rysunek 5.8 Raport klasyfikacji - szkolenie z augmentacją.

	Precyzja (precision)	Czułość (recall)	F1	Liczba obrazów (support)
Abstrakcja	0.78	0.85	0.81	744
Pejzaż Miejski	0.69	0.64	0.66	684
Malarstwo Rodzajowe	0.56	0.55	0.55	1579
Ilustracja	0.58	0.52	0.55	282
Krajobraz	0.76	0.83	0.79	1973
Akt	0.60	0.50	0.54	288
Portret	0.81	0.77	0.79	2101
Malarstwo Religijne	0.69	0.65	0.67	976
Szkic	0.59	0.61	0.60	589
Martwa Natura	0.68	0.69	0.69	416
Dokładność (accuracy)			0.70	9632
Średnia	0.67	0.66	0.67	9632
Średnia ważona	0.70	0.70	0.70	9632

Rysunek 5.9 Raport klasyfikacji - szkolenie bez augmentacji.

Dla tej sieci, w celu zbadania przebiegu nauczania, szkolenie wykonywano przez stałą liczbę epok. W następnych przypadkach, w celu zaoszczędzenia czasu obliczeń, wprowadzono funkcję zatrzymującą obliczenia w przypadku osiągnięcia trzech kolejnych epok bez zmniejszenia funkcji kary dla zbioru walidacyjnego.

5.1.2 Sieć do identyfikacji artysty

Początkowo zbadano skuteczność szkolenia dla wszystkich 1108 artystów znajdujących się w bazie, przy czym przewidywano wystąpienie znacznego błędu ze względu na niewielką liczebność dzieł co niektórych artystów. Po 10 epokach szkolenia otrzymano dokładność na zbiorze treningowym wynoszącą zaledwie 11,53%, ze względu na co zaprzestano szkolenia. Zbadano skuteczność przy dokonywaniu klasyfikacji dla artystów posiadających przynajmniej 250 obrazów w bazie danych. Oznaczało to szkolenie na 30 664 obrazów podzielonych między 56 klas.

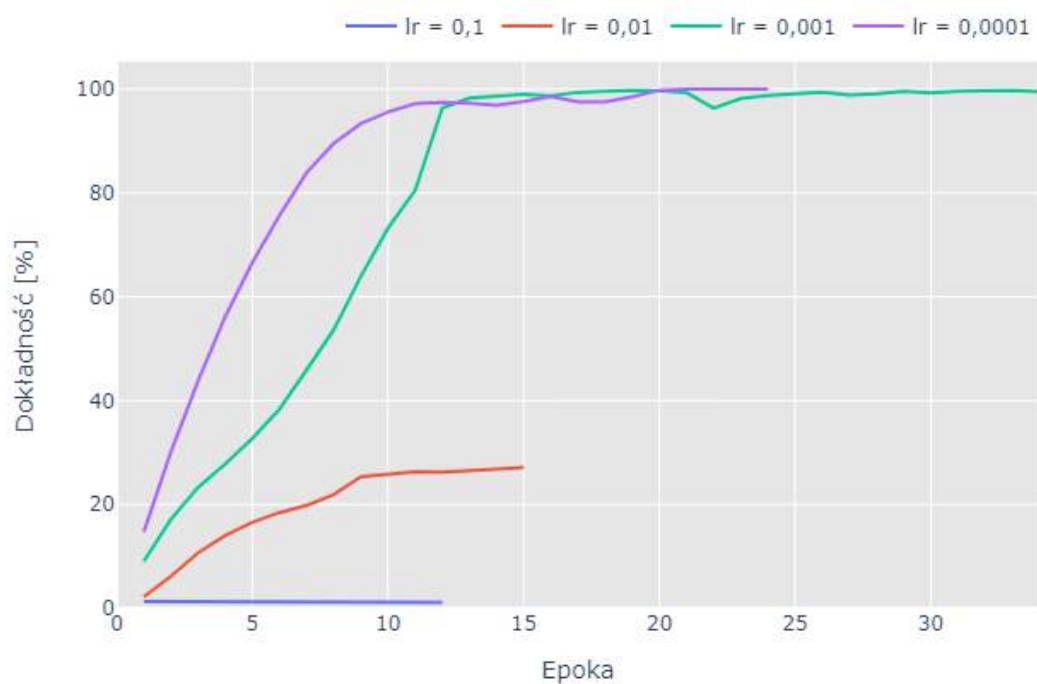
Podczas nauki tej sieci zbadano wpływ innego z hiperparametrów - szybkości nauczania (z ang. learning rate). Sprawdzono 4 wartości: $lr = 0,1$; $lr = 0,01$; $lr = 0,001$; $lr = 0,0001$.

Ponieważ przebieg długiego szkolenia zaprezentowano już w podrozdziale 5.1.1, w celu oszczędzenia czasu obliczeń, podczas badań nad siecią do identyfikacji artysty użyto funkcji EarlyStopping, pozwalającej na zatrzymanie szkolenia w przypadku braku poprawy. Warunek zatrzymania ustawiono jako brak zmniejszenia funkcji kosztu dla zbioru walidacyjnego w ciągu 3 kolejnych epok.

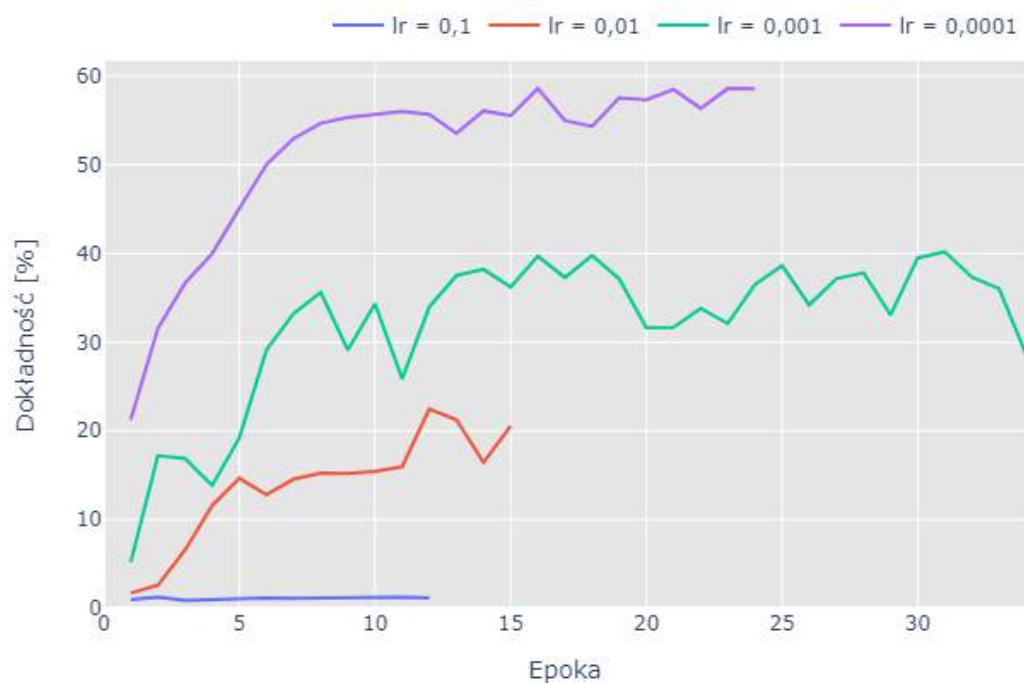
W celu zwiększenia przejrzystości, w wizualizacjach przebiegu szkolenia połączono oba etapy szkolenia.

W przypadku sieci o szybkości uczenia $lr = 0,001$ wystąpiły problemy z pamięcią, co skutkowało kilkukrotnym przerwaniem szkolenia. Kontynuowano wówczas z ostatniego zapisanego punktu, którym była poprzednia epoka - jedynym więc wymiernym skutkiem był brak wcześniejszego przerwania szkolenia dla tej sieci.

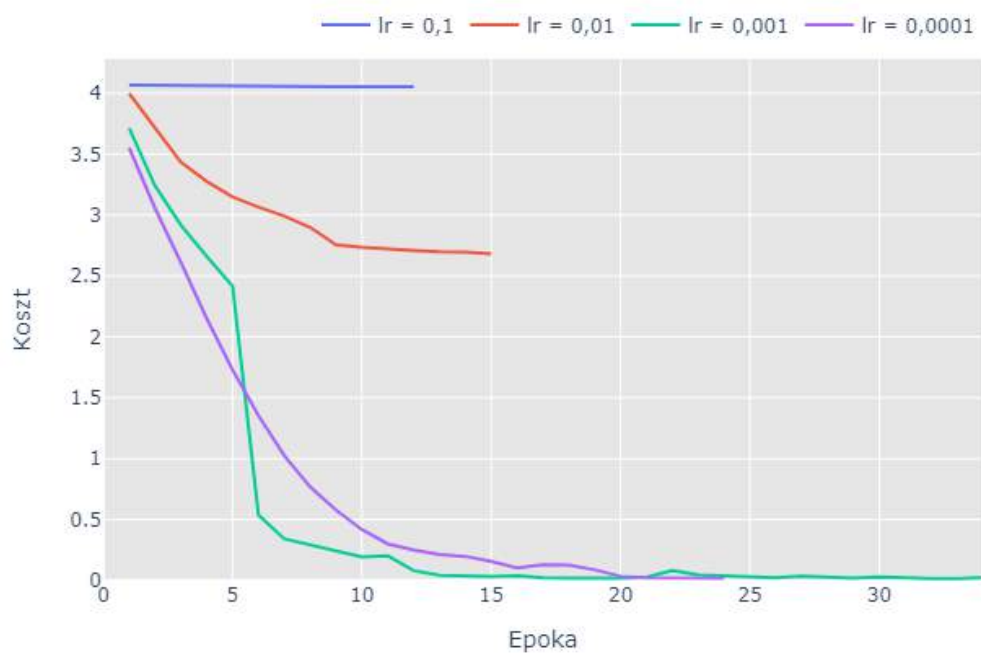
Dla sieci o szybkości uczenia $lr = 0,0001$ problemy z pamięcią stały się jeszcze większe i zapas systemu szkolenia następowała z reguły już po jednej epoce. Aby oszczędzić pamięć i umożliwić dokończenie szkolenia, zrezygnowano z wcześniej stosowanej funkcji wykreślenia bieżących macierzy pomyłek. Macierz pomyłek wykreślono dopiero po zakończeniu szkolenia.



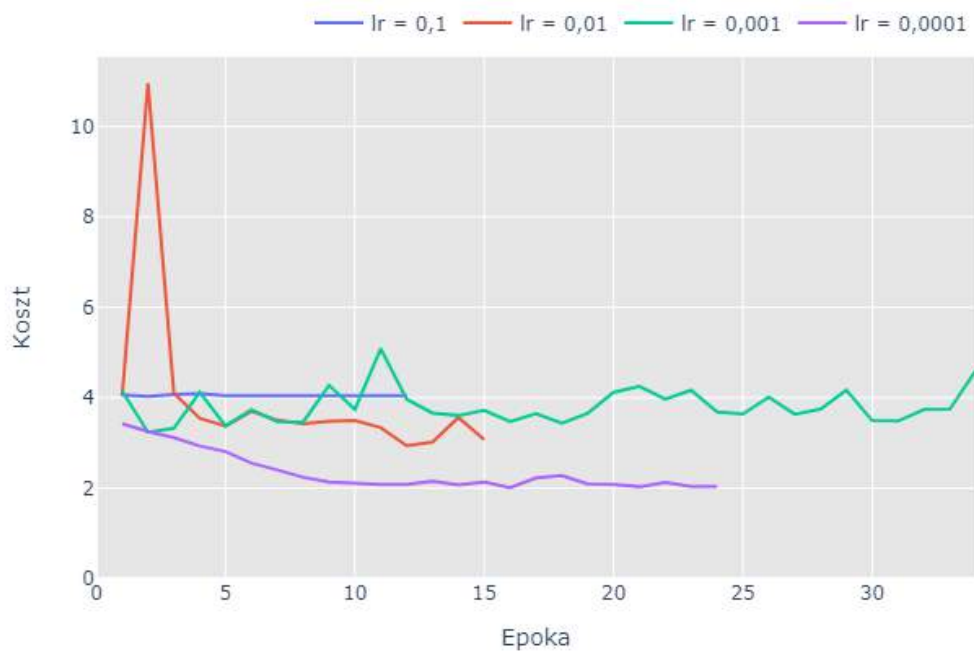
Rysunek 5.10 Dokładność na zbiorze treningowym w kolejnych epokach szkolenia.



Rysunek 5.11 Dokładność na zbiorze walidacyjnym w kolejnych epokach szkolenia.



Rysunek 5.12 Funkcja kosztu na zbiorze treningowym w kolejnych epokach szkolenia.



Rysunek 5.13 Funkcja kosztu na zbiorze walidacyjnym w kolejnych epokach szkolenia.

Można zaobserwować, że najbardziej pożądaną przebieg szkolenia uzyskuje się przy

użyciu szybkości szkolenia wynoszącej $lr = 0,0001$.

Największą anomalię zaobserwowano na Rysunku 5.13, gdzie w drugiej epoce otrzymano bardzo wysoką wartość funkcji kosztu dla szybkości szkolenia $lr = 0,01$.

Dokonano ewaluacji najlepszych modeli na zbiorach treningowym i testowym, zmierzono dokładność i zapisano w Tabeli 5.2.

Tabela 5.2 Dokładność uzyskana po ewaluacji dla różnych wartości szybkości szkolenia.

lr	Dokładność - zbiór treningowy	Dokładność - zbiór testowy
0,1	1,23%	1,28%
0,01	22,04%	20,72%
0,001	99,52%	29,87%
0,0001	99,99%	60,37%

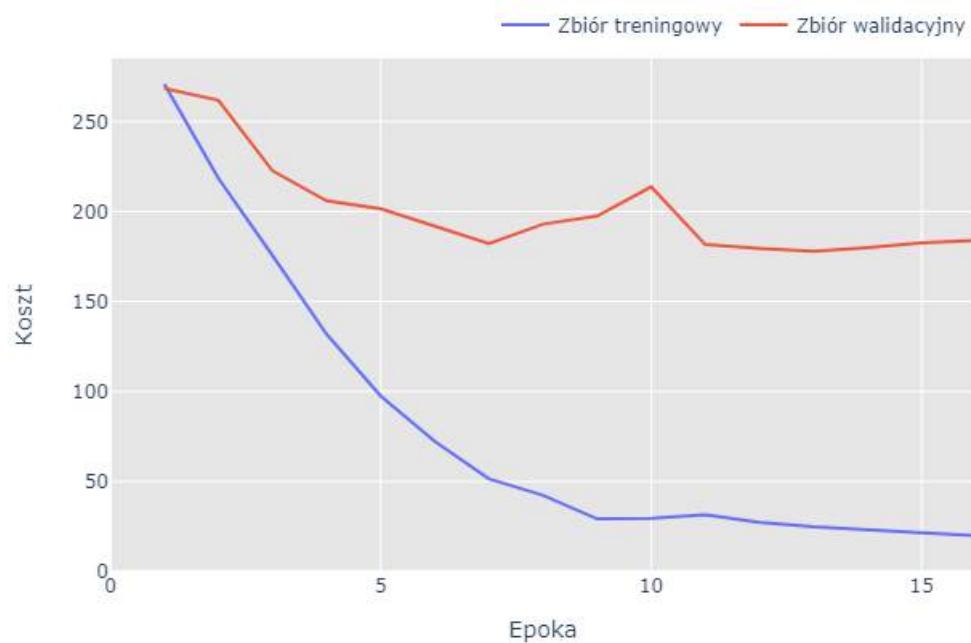
5.1.3 Sieć do identyfikacji stylu

Do określenia ilości epok szkolenia sieci do identyfikacji stylu ponownie wykorzystano funkcję EarlyStopping, która zatrzymywała szkolenie po 3 kolejnych epokach bez zmniejszenia wartości funkcji kosztu.

Przebieg obu etapów szkolenia połączono i zamieszczono na Rysunkach 5.14 i 5.15.



Rysunek 5.14 Dokładność sieci do identyfikacji stylu.



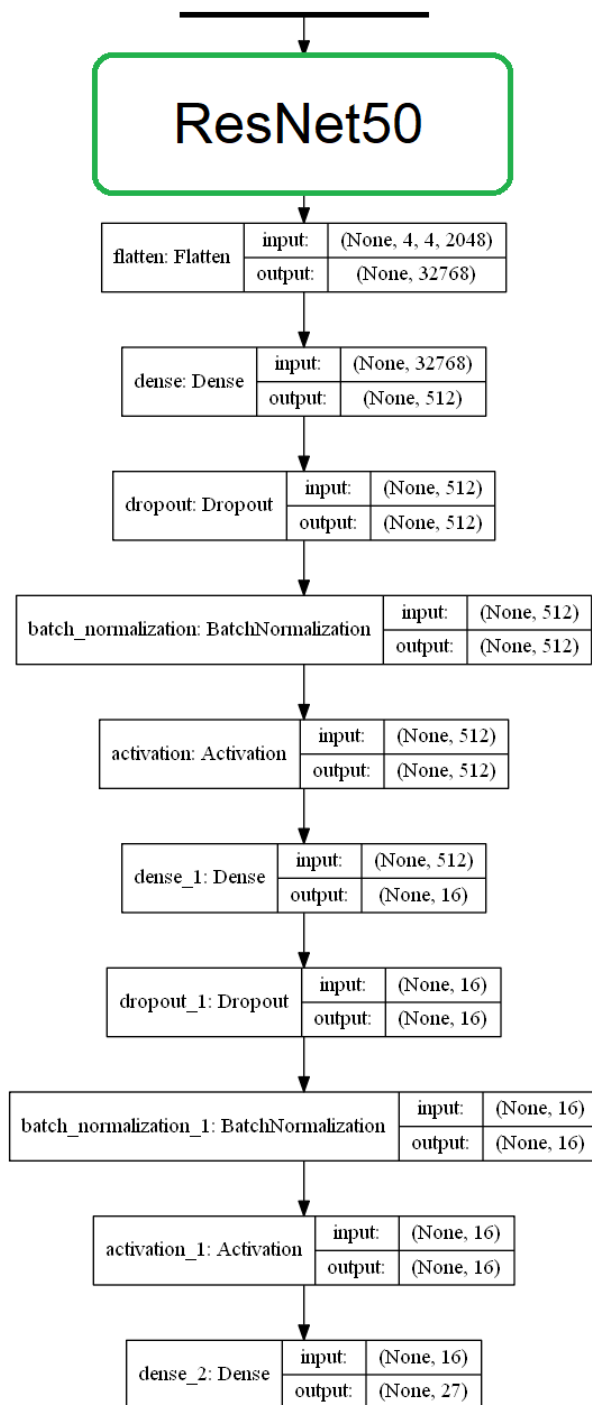
Rysunek 5.15 Funkcja kosztu sieci do identyfikacji stylu.

Dokonano ewaluacji na zbiorze treningowym, walidacyjnym i testowym. Otrzymano dokładności wynoszące odpowiednio 87,04%; 47,42% i 47,12 %. Wykreślono raport klasyfikacji dla zbioru testowego (Rys. 5.16)

	Precyzja (precision)	Czułość (recall)	F1	Liczba obrazów (support)
Ekspresjonizm Abstrakcyjny	0.43	0.52	0.47	417
Taszyzm	0.46	0.40	0.43	15
Kubizm Analityczny	0.67	0.59	0.62	17
Art Nouveau	0.44	0.45	0.44	641
Barok	0.49	0.60	0.54	631
Malarstwo Barwnych Płaszczyzn	0.72	0.75	0.74	241
Realizm Współczesny	0.34	0.19	0.25	72
Kubizm	0.57	0.32	0.41	331
Wczesny Renesans	0.54	0.41	0.47	208
Ekspresjonizm	0.36	0.30	0.33	1009
Fowizm	0.22	0.26	0.24	140
Wysoki Renesans	0.48	0.31	0.38	199
Impresjonizm	0.58	0.61	0.59	1905
Późny Renesans	0.39	0.44	0.41	191
Minimalizm	0.65	0.71	0.68	201
Prymitywizm	0.49	0.38	0.42	356
Nowy Realizm	0.30	0.30	0.30	47
Północny Renesans	0.60	0.56	0.58	383
Puentylizm	0.43	0.55	0.48	74
Pop Art	0.62	0.40	0.49	223
Postimpresjonizm	0.33	0.39	0.35	964
Realizm	0.45	0.44	0.44	1605
Rokoko	0.54	0.48	0.51	313
Romantyzm	0.48	0.44	0.46	1044
Symbolizm	0.34	0.41	0.37	662
Kubizm Syntetyczny	0.62	0.56	0.59	32
Ukiyo-e	0.81	0.82	0.81	175
Dokładność (accuracy)			0.47	12096
Średnia	0.49	0.47	0.47	12096
Średnia ważona	0.48	0.47	0.47	12096

Rysunek 5.16 Raport klasyfikacji - sieć do identyfikacji stylu.

Dysproporcja pomiędzy wynikami sieci dla zbioru treningowego, a walidacyjnego i testowego okazała się być bardzo duża. Oznacza to słabą generalizację. W celu uzyskania lepszego wyniku, postanowiono podjąć próbę wprowadzenia do ostatnich warstw modelu dwóch dodatkowych warstw typu dropout umiejscowionych między warstwami gęstymi a funkcją aktywacji. Otrzymano ostatnie warstwy sieci zgodnie ze schematem z Rysunku 5.17.

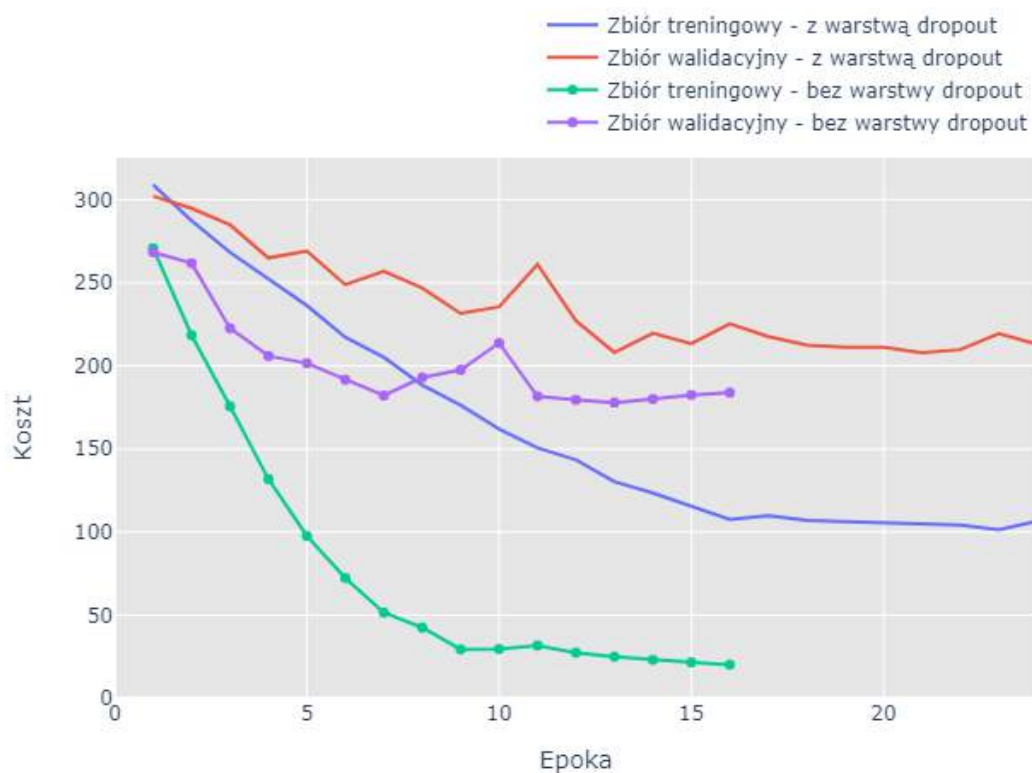


Rysunek 5.17 Warstwy wyjściowe sieci do identyfikacji stylu po dodaniu warstw typu dropout.

Przebieg szkolenia zestawiono z przebiegiem szkolenia sieci bez dodatkowych warstw typu dropout i zamieszczono na Rysunkach 5.18 i 5.19.



Rysunek 5.18 Dokładność sieci do identyfikacji stylu podczas szkolenia.



Rysunek 5.19 Funkcja kosztu sieci do identyfikacji stylu podczas szkolenia.

Koncepcja ta nie sprawdziła się. Otrzymano niższe dokładności - wynoszące 58,61% na zbiorze treningowym, 38,32% na zbiorze walidacyjnym i 37,32% na zbiorze testowym.

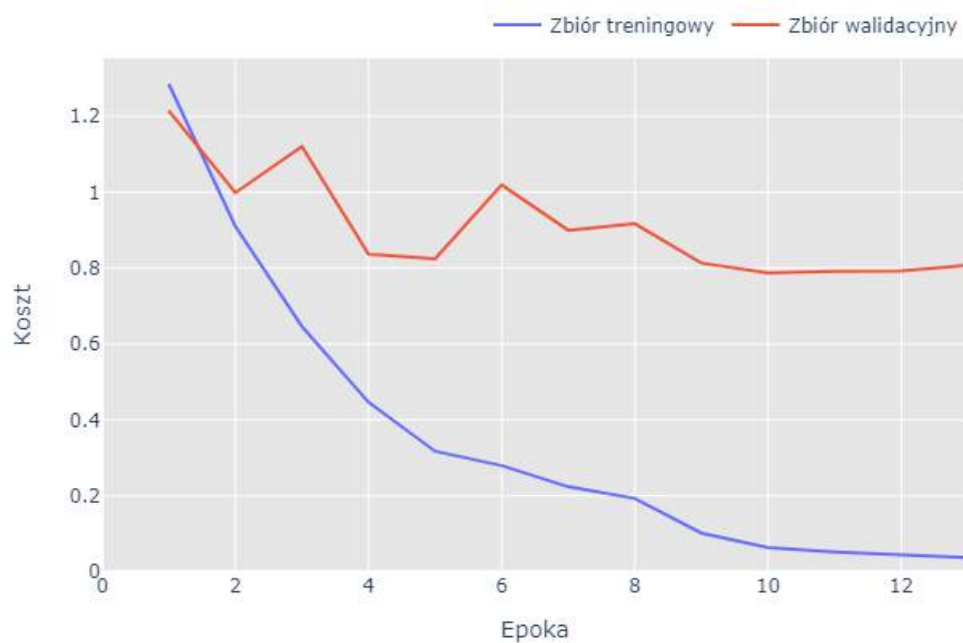
5.1.4 Sieć do identyfikacji wieku

Do określenia ilości epok szkolenia sieci do identyfikacji wieku ponownie wykorzystano funkcję EarlyStopping, która zatrzymywała szkolenie po 3 kolejnych epokach bez zmniejszenia wartości funkcji kosztu.

Przebieg obu etapów szkolenia połączono i zamieszczono na Rysunkach 5.20 i 5.21.

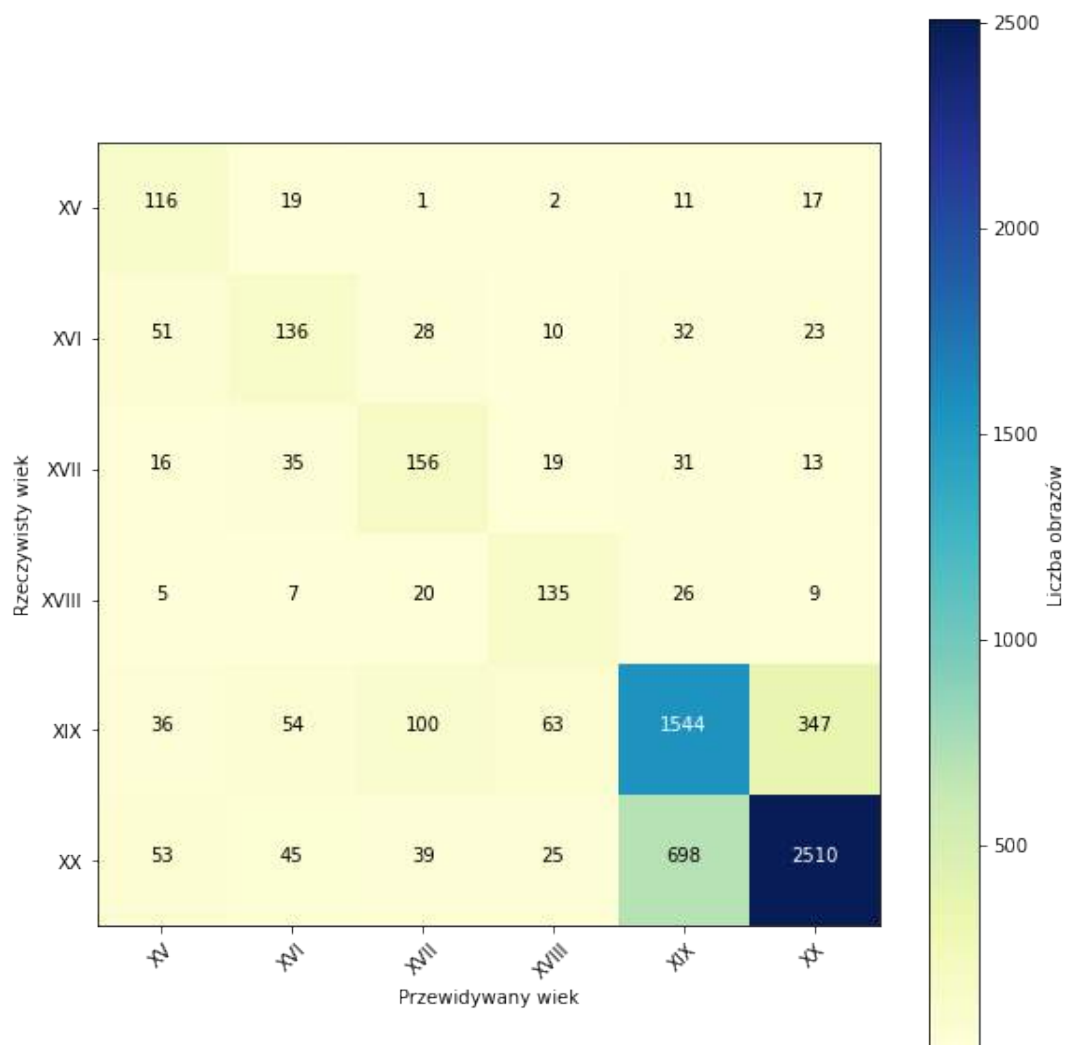


Rysunek 5.20 Dokładność sieci do identyfikacji wieku podczas szkolenia.



Rysunek 5.21 Funkcja kosztu sieci do identyfikacji wieku podczas szkolenia.

Wykreślono macierz pomyłek (Rys 5.22) i raport klasyfikacji dla zbioru testowego (Rys. 5.23).



Rysunek 5.22 Macierz pomyłek - sieć do identyfikacji wieku.

	Precyzja (precision)	Czułość (recall)	F1	Liczba obrazów (support)
XV	0.59	0.54	0.56	166
XVI	0.54	0.50	0.52	280
XVII	0.56	0.50	0.53	270
XVIII	0.62	0.52	0.57	202
XIX	0.67	0.75	0.70	2144
XX	0.84	0.80	0.82	3370
Dokładność (accuracy)			0.74	6432
Średnia	0.64	0.60	0.62	6432
Średnia ważona	0.75	0.74	0.74	6432

Rysunek 5.23 Raport klasyfikacji - sieć do identyfikacji wieku.

Dokonano ewaluacji i sprawdzono dokładność na zbiorze treningowym, walidacyjnym i testowym. Otrzymano odpowiednio 83,44%, 74,13% i 71,46%.

5.1.5 Porównanie wyników sieci niższego rzędu

Do określenia jakości uzyskiwanych sieci stosowano do tej pory dokładność, która jest podstawowym wskaźnikiem. W zależności od planowanego zastosowania sieci wykorzystuje się dodatkowo: precyzję (PRE), czułość (REC) i wskaźnik F1. Nawiązują one do możliwych wyników testu: prawdziwie dodatnich (PD), prawdziwie ujemnych (PU), fałszywie dodatnich (FD) i fałszywie ujemnych (FU), przypisywanych jak w macierzy pomyłek dla problemu binarnego (Rys. 5.24).

		Wynik rzeczywisty	
		Dodatni	Ujemny
Wynik przewidywany	Dodatni	Prawdziwie dodatni (PD)	Fałszywie dodatni (FU)
	Ujemny	Fałszywie ujemny (FU)	Prawdziwie ujemny (PU)

Rysunek 5.24 Macierz pomyłek dla problemu binarnego.

Precyzja opisuje dokładność rozpoznania wewnątrz klasy i określa się ją Wzorem 5.1 [20]. Informuje ona, ile z przykładów zakwalifikowanych jako dodatnie jest w rzeczywistości dodatnich.

$$PRE = \frac{PD}{PD + FD} \quad (5.1)$$

Czułość często jest traktowana jako uzupełnienie do precyzji, ponieważ bierze poprawkę na ilość elementów wewnątrz klasy (Wzór 5.2). Jest to tzw. odsetek prawdziwie pozytywnych (ang. true positive rate, oznaczany również jako *TPR*).

$$REC = \frac{PD}{PD + FU} \quad (5.2)$$

Wskaźnik F1 jest średnią harmoniczną pomiędzy precyzją i czułością i opisuje go Wzór 5.3. Pozwala wziąć pod uwagę oba wskaźniki. W idealnej sytuacji, gdy precyzja i czułość wynoszą 1, F1 również przyjmuje wartość 1.

$$F1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC} \quad (5.3)$$

W Tabeli 5.3 zestawiono wyniki tych wskaźników dla wszystkich sieci niższego rzędu.

Tabela 5.3 Porównanie sieci niższego rzędu.

	Sieć identyfikująca	Gatunek	Artystę	Styl	Stulecie
	Dokładność na zbiorze treningowym	0.99	0.99	0.87	0.82
	Dokładność na zbiorze walidacyjnym	0.70	0.58	0.47	0.74
	Dokładność na zbiorze testowym	0.70	0.60	0.47	0.71
Średnia	Precyzja (precision)	0.67	0.59	0.49	0.56
	Czułość (recall)	0.66	0.58	0.47	0.65
	F1	0.67	0.58	0.47	0.60
Średnia ważona	Precyzja (precision)	0.70	0.61	0.48	0.74
	Czułość (recall)	0.70	0.60	0.47	0.71
	F1	0.70	0.60	0.47	0.72
	Liczba klas	10	56	27	6
	Liczba obrazów w zbiorze testowym	9632	4576	12096	6432

Wyniki te są porównywalne z wynikami podobnych zadań znalezionymi w literaturze:

- Eksperymenty nad sieciami do identyfikacji gatunku przeprowadził Golara w 2019 [7]. Uzyskał 71 % dokładności na zbiorze testowym na klasycznym ResNecie50 i do 62% na zbiorze testowym na ResNecie50 wzbogaconym o Transfer Learning. Użył do tego bazy danych („Painters by numbers” z serwisu kaggle.com) o innym podziale na gatunki (zawierającej 44 klasy) - co było utrudnieniem, ale również znajdowało się w niej więcej obrazów (100 000), a także dzięki braku ograniczeń sprzętowych zastosował większy rozmiar obrazu (224×224) i znacznie większy rozmiar partii danych (batch_size=2048 dla klasycznego ResNetu50 i batch_size=512 dla ResNetu50 z zastosowaniem Transfer Learning), co umożliwiło poprawienie rezultatów.
- W zadaniu rozpoznawania artysty, różne sieci neuronowe zostały zestawione przez Davida w 2016 [3], gdzie osiągał dokładność do 96.52%, należy jednak zauważyć, że jego sieci były nauczone klasyfikacji między zaledwie 3 artystami. Halder w 2019 na sieci opartej na architekturze ResNet50 dla 11 artystów otrzymał dokładność 84,44% [10]. Wydaje się być naturalne, że przy znacznym zwiększeniu liczebności artystów - do 56 - sieć opisywana w niniejszej pracy uzyskała odpowiednio mniejszą dokładność - 60%.
- W 2019 Joshi do porównania swojej sieci do identyfikacji stylu z ResNetem50 wykorzystał Resnet50 (bez augmentacji) o dokładności 50,1% na zbiorze testowym [13], zastosował jednak większe rozmiary wejściowe obrazów (224×224), co w niniejszej pracy było niemożliwe ze względu na ograniczenia mocy wykorzystanego GPU.
- Innej pracy bazującej na sieciach neuronowych i traktującej o zadaniu rozpoznawaniu stulecia, w którym powstał obraz, nie znaleziono.

5.2 Sieci nadrzędne

W celu wyszkolenia sieci poziomu wyższego stworzono nową bazę danych, składającą się tylko z obrazów nieodrzuconych dla żadnej kategorii, czyli dzieł artystów posiadających przynajmniej 250 obrazów w bazie danych, o określonym gatunku i znanym roku stworzenia zawartym w przedziale od XV do XX wieku. Oznaczało to zmniejszenie bazy do 14 498 obrazów. Zebrano przewidywania z każdej sieci w jeden plik .csv, gdzie kolejne kolumny zawierały prawdopodobieństwo wystąpienia konkretnych stylów, gatunków, artystów i wieków, a kolejne wiersze odnosiły się do konkretnych obrazów. Następnie, dla każdej z kategorii, stworzono osobny plik zawierający poza przewidywaniami dodatkową kolumnę z prawidłową klasą - przykładową część takiego pliku dla sieci do identyfikacji artysty zamieszczono na Rysunku 5.25.

	Ekspresjonizm Abstrakcyjny	Taszyzm	Kubizm Analityczny	Barok	...	vincent- van-gogh	william- merritt- chase	...	XVIII	XIX	XX	Artist_name
0	0.000589	0.002188	0.002482	0.012482	...	9.747736e-01	0.000382	...	0.001303	0.944801	0.047749	vincent-van-gogh
1	0.001132	0.001029	0.000615	0.841680	...	9.461818e-04	0.000032	...	0.005275	0.196726	0.011566	rembrandt
2	0.000163	0.000487	0.001985	0.000112	...	2.047761e-04	0.000144	...	0.002706	0.685608	0.300687	pierre-auguste-renoir
3	0.013019	0.003855	0.018558	0.002844	...	9.773053e-05	0.000003	...	0.000116	0.002581	0.996280	boris-kustodiev
4	0.001197	0.001207	0.002134	0.014106	...	1.027618e-06	0.000210	...	0.007869	0.951011	0.010199	ivan-aivazovsky
...
14493	0.935406	0.001151	0.000083	0.000731	...	7.247594e-07	0.000023	...	0.001445	0.000704	0.996538	sam-francis
14494	0.969081	0.000231	0.000050	0.000629	...	1.285128e-07	0.000013	...	0.078744	0.005381	0.896141	sam-francis
14495	0.718757	0.002415	0.000289	0.002462	...	1.149921e-07	0.000009	...	0.015423	0.016029	0.939346	sam-francis
14496	0.888244	0.004572	0.001014	0.000492	...	2.730702e-03	0.001567	...	0.003904	0.024709	0.958595	sam-francis
14497	0.942257	0.000780	0.000214	0.003300	...	9.156649e-07	0.000044	...	0.000018	0.000004	0.999960	sam-francis

14498 rows × 100 columns

Rysunek 5.25 Utworzony plik z przewidywaniami dla każdego obrazu.

W efekcie możliwe było zastosowanie klasyfikatorów, które każdy obraz, na podstawie pierwszych (99) kolumn z przewidywaniami z poszczególnych sieci podrzędnych, przydzielały jednemu z artystów/gatunków/stylów/stuleci.

Podziału na zbiory treningowy, walidacyjny i testowy dokonano na zasadzie przypisania tych samych obrazów konkretnej kategorii do tych samych zbiorów (np. obraz będący w zbiorze testowym przy szkoleniu sieci do identyfikacji artysty na niższym poziomie był również w zbiorze testowym przy szkoleniu sieci do identyfikacji artysty na wyższym poziomie). Przy tak znacznym okrojeniu bazy okazało się to problematyczne - niektóre klasy z kategorii artysty oraz stylu nie miały wówczas reprezentantów w zbiorze testowym lub walidacyjnym. Aby zachować koherentność podziału z sieciami niższego rzędu, połączono zbiory testowy i walidacyjny w jeden, co zapewniło istnienie reprezentantów każdej klasy w każdym zbiorze.

Każdy z klasyfikatorów wymaga podania różnych parametrów, od których zależy sku-

teczność jego działania. Podobnie jak w przypadku sieci neuronowych, nie istnieją żadne pewne metody gwarantujące uzyskanie najlepszych rezultatów, a dostrajanie ich odbywa się metodą prób i błędów.

Wykonano eksperymenty i najlepsze rezultaty otrzymano:

- dla sieci neuronowych z jedną, dwoma lub trzema warstwami ukrytymi o liczbie neuronów warstw ukrytych równą 8 i funkcją aktywacyjną typu ReLu oraz optymalizatorem ADAM i szkoleniu partiami danych o rozmiarach `batch_size=8`,
- dla sieci typu perceptron wielowarstwowy o wymiarach 200×20 wykorzystującej algorytm BFGS,
- dla regresji logistycznej wielomianowej wykorzystującej algorytm BFGS,
- dla maszyny wektorów nośnych (SVM) w trybie jeden vs jeden,
- dla lasu losowego o liczbie drzew wynoszącej 1000 i maksymalnej głębokości jednego drzewa wynoszącej 100.

Tabela 5.4 Porównanie sieci do identyfikacji gatunku.

		Precyzja (precision)	Czułość (recall)	F1
Sieć Niższego Poziomu	Dokładność (accuracy)			0,70
	Średnia	0,67	0,66	0,67
	Średnia ważona	0,70	0,70	0,70
Regresja Logistyczna	Dokładność (accuracy)			0,71
	Średnia	0,65	0,67	0,65
	Średnia ważona	0,71	0,71	0,71
Maszyna Wektorów Nośnych	Dokładność (accuracy)			0,71
	Średnia	0,67	0,65	0,66
	Średnia ważona	0,71	0,71	0,70
Las Losowy	Dokładność (accuracy)			0,71
	Średnia	0,63	0,67	0,64
	Średnia ważona	0,71	0,71	0,70
Perceptron Wielowarstwowy	Dokładność (accuracy)			0,72
	Średnia	0,66	0,68	0,66
	Średnia ważona	0,72	0,72	0,72
Sieć z Jedną Warstwą Ukrytą	Dokładność (accuracy)			0,71
	Średnia	0,63	0,68	0,64
	Średnia ważona	0,71	0,71	0,71
Sieć z Dwoma Warstwami Ukrytymi	Dokładność (accuracy)			0,70
	Średnia	0,63	0,67	0,64
	Średnia ważona	0,71	0,70	0,70
Sieć z Trzema Warstwami Ukrytymi	Dokładność (accuracy)			0,71
	Średnia	0,63	0,66	0,64
	Średnia ważona	0,71	0,71	0,71

Tabela 5.5 Porównanie sieci do identyfikacji stylu.

		Precyzja (precision)	Czułość (recall)	F1
Sieć Niższego Poziomu	Dokładność (accuracy)			0,47
	Średnia	0,49	0,47	0,47
	Średnia ważona	0,48	0,47	0,47
Regresja Logistyczna	Dokładność (accuracy)			0,71
	Średnia	0,61	0,54	0,55
	Średnia ważona	0,71	0,71	0,71
Maszyna Wektorów Nośnych	Dokładność (accuracy)			0,73
	Średnia	0,58	0,52	0,53
	Średnia ważona	0,73	0,73	0,72
Las Losowy	Dokładność (accuracy)			0,58
	Średnia	0,55	0,44	0,45
	Średnia ważona	0,59	0,58	0,58
Perceptron Wielowarstwowy	Dokładność (accuracy)			0,68
	Średnia	0,59	0,51	0,53
	Średnia ważona	0,69	0,68	0,68
Sieć z Jedną Warstwą Ukrytą	Dokładność (accuracy)			0,71
	Średnia	0,55	0,54	0,52
	Średnia ważona	0,71	0,71	0,71
Sieć z Dwoma Warstwami Ukrytymi	Dokładność (accuracy)			0,69
	Średnia	0,54	0,53	0,51
	Średnia ważona	0,69	0,69	0,69
Sieć z Trzema Warstwami Ukrytymi	Dokładność (accuracy)			0,68
	Średnia	0,49	0,51	0,49
	Średnia ważona	0,68	0,68	0,68

Tabela 5.6 Porównanie sieci do identyfikacji artysty.

		Precyzja (precision)	Czułość (recall)	F1
Sieć Niższego Poziomu	Dokładność (accuracy)			0,60
	Średnia	0,59	0,58	0,58
	Średnia ważona	0,61	0,60	0,60
Regresja Logistyczna	Dokładność (accuracy)			0,62
	Średnia	0,58	0,58	0,57
	Średnia ważona	0,63	0,62	0,61
Maszyna Wektorów Nośnych	Dokładność (accuracy)			0,60
	Średnia	0,59	0,53	0,54
	Średnia ważona	0,63	0,60	0,60
Las Losowy	Dokładność (accuracy)			0,54
	Średnia	0,48	0,41	0,41
	Średnia ważona	0,57	0,54	0,54
Perceptron Wielowarstwowy	Dokładność (accuracy)			0,61
	Średnia	0,54	0,59	0,55
	Średnia ważona	0,63	0,61	0,61
Sieć z Jedną Warstwą Ukrytą	Dokładność (accuracy)			0,58
	Średnia	0,51	0,56	0,51
	Średnia ważona	0,61	0,58	0,58
Sieć z Dwoma Warstwami Ukrytymi	Dokładność (accuracy)			0,56
	Średnia	0,50	0,55	0,50
	Średnia ważona	0,61	0,56	0,58
Sieć z Trzema Warstwami Ukrytymi	Dokładność (accuracy)			0,55
	Średnia	0,48	0,54	0,48
	Średnia ważona	0,60	0,55	0,56

Tabela 5.7 Porównanie sieci do identyfikacji wieku.

		Precyzja (precision)	Czułość (recall)	F1
Sieć Niższego Poziomu	Dokładność (accuracy)			0,74
	Średnia	0,64	0,60	0,62
	Średnia ważona	0,75	0,74	0,74
Regresja Logistyczna	Dokładność (accuracy)			0,87
	Średnia	0,82	0,78	0,80
	Średnia ważona	0,87	0,87	0,87
Maszyna Wektorów Nośnych	Dokładność (accuracy)			0,88
	Średnia	0,89	0,75	0,80
	Średnia ważona	0,88	0,88	0,88
Las Losowy	Dokładność (accuracy)			0,84
	Średnia	0,90	0,73	0,79
	Średnia ważona	0,84	0,84	0,84
Perceptron Wielowarstwowy	Dokładność (accuracy)			0,92
	Średnia	0,88	0,88	0,88
	Średnia ważona	0,92	0,92	0,92
Sieć z Jedną Warstwą Ukrytą	Dokładność (accuracy)			0,88
	Średnia	0,86	0,79	0,82
	Średnia ważona	0,88	0,88	0,88
Sieć z Dwoma Warstwami Ukrytymi	Dokładność (accuracy)			0,88
	Średnia	0,84	0,80	0,81
	Średnia ważona	0,88	0,88	0,88
Sieć z Trzema Warstwami Ukrytymi	Dokładność (accuracy)			0,88
	Średnia	0,82	0,81	0,81
	Średnia ważona	0,88	0,88	0,88

W niektórych przypadkach, wyniki otrzymywane różnymi sposobami są zbliżone, a w zależności od wybranego wskaźnika, inna sieć może zostać uznana za „najlepszą”. Do zaprezentowania działania na przykładowych obrazach wybrano po jednym klasyfikatorze i jednej sieci neuronowej. W przypadku klasyfikatora była to regresja logistyczna, a w przypadku sieci - sieć z jedną warstwą ukrytą, za wyjątkiem sieci do identyfikacji wieku, gdzie zdecydowanie najlepsze rezultaty osiągał perceptron wielowarstwowy i to on został wybrany do prezentacji w Rozdziale 6.

Rozdział 6

Przykłady stosowania

Sieci obu poziomów zostały złożone w jednym programie, którego zadaniem jest dokonanie klasyfikacji w każdej z kategorii. Wypisuje on tabelę z rzeczywistymi klasami (w przypadku ich znajomości), a także przewidywaniami sieci niższego poziomu, przewidywaniami wyższego poziomu dokonanymi na podstawie najlepszego klasyfikatora oraz przewidywaniami wyższego poziomu dokonanymi na podstawie najlepszej sieci neuronowej. W nawiasach umieszczono prawdopodobieństwo poprawności klasyfikacji według sieci. Pozwala to zaobserwować postęp uzyskany dzięki dodaniu kolejnego poziomu oraz porównać wyniki.

6.1 Losowanie obrazów

Ze względu na konieczność stosowania stratyfikacji, dla każdej kategorii podział na zbiory treningowy, testowy i walidacyjny był wykonywany od nowa i do zbiorów każdej kategorii trafiały inne obrazy. Aby zademonstrować skuteczność systemu, wybrano losowe obrazy każdej kategorii będące w zbiorach testowych na obu poziomach i zamieszczono na Rysunkach 6.1, 6.2, 6.3 i 6.4 wraz z przewidywaniami.



	Artysta	Styl	Gatunek	Wiek
Rzeczywisty	Alfred Sisley	Impresjonizm	Krajobraz	XIX
Przewidywany -Poziom Niższy	Alfred Sisley(95.51%)	Impresjonizm(79.98%)	Krajobraz(99.98%)	XX(86.62%)
Przewidywany -Poziom Wyższy -Klasyfikator	Alfred Sisley(99.91%)	Impresjonizm(99.14%)	Krajobraz(99.98%)	XIX(77.58%)
Przewidywany -Poziom Wyższy -Sieć Neuronowa	Alfred Sisley(100.00%)	Impresjonizm(99.15%)	Krajobraz(100.00%)	XIX(100.00%)

Rysunek 6.1 Losowy obraz - test gatunku.

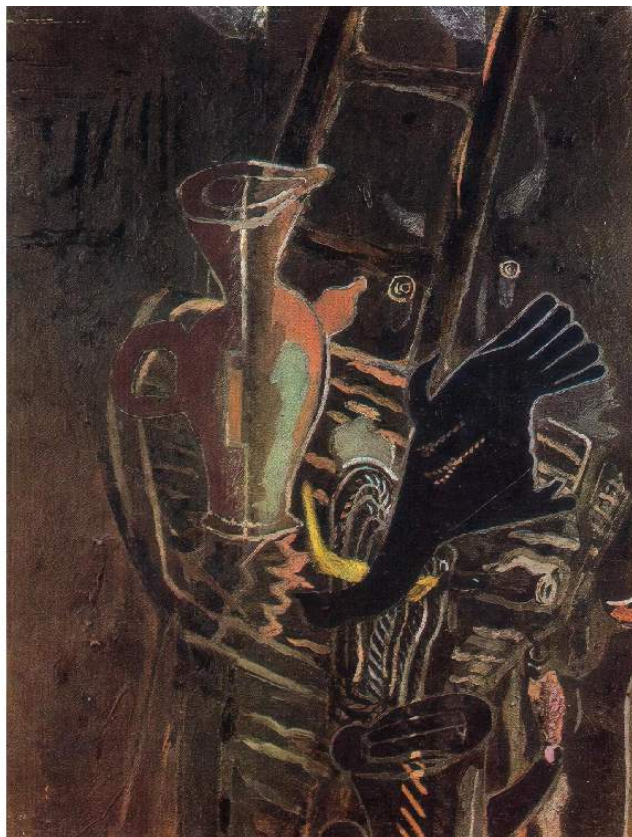
Dzięki zastosowaniu poziomu wyższego można zaobserwować poprawę w pewności przewidywań lub utrzymanie ich poziomu dla wszystkich kategorii, a także zmianę w przewidywanym wieku z błędnego na prawidłowy.



	Artysta	Styl	Gatunek	Wiek
Rzeczywisty	Titian	Późny Renesans	Portret	XVI
Przewidywany -Poziom Niższy	Titian(99.86%)	Art Nouveau(47.97%)	Portret(99.96%)	XVI(96.92%)
Przewidywany -Poziom Wyższy -Klasyfikator	Titian(99.83%)	Realizm(54.80%)	Portret(99.97%)	XVIII(80.39%)
Przewidywany -Poziom Wyższy -Sieć Neuronowa	Titian(100.00%)	Realizm(68.01%)	Portret(100.00%)	XVI(100.00%)

Rysunek 6.2 Losowy obraz - test stylu.

Żadna z sieci nie przewidziała poprawnie stylu; jednak przewidywano style wizualnie podobne, a dodatkowo żadna metoda nie dała zdecydowanego wyniku - pewność osiągnęła maksymalnie 68,01%. Klasyfikator poziomu wyższego wskazał błędny wiek powstania, natomiast sieć neuronowa poziomu wyższego wskazała prawidłowy z 100% pewnością.



	Artysta	Styl	Gatunek	Wiek
Rzeczywisty	Georges Braque	Ekspresjonizm	Martwa Natura	XX
Przewidywany -Poziom Niższy	Georges Braque(87.44%)	Ekspresjonizm(66.25%)	Martwa Natura(99.95%)	XX(79.31%)
Przewidywany -Poziom Wyższy -Klasyfikator	Georges Braque(99.78%)	Ekspresjonizm(95.30%)	Martwa Natura(99.95%)	XX(99.69%)
Przewidywany -Poziom Wyższy -Sieć Neuronowa	Georges Braque(100.00%)	Ekspresjonizm(98.96%)	Martwa Natura(100.00%)	XX(100.00%)

Rysunek 6.3 Losowy obraz - test artysty.

Na poziomie wyższym wzrosły pewności wszystkich przewidywanych klas.



	Artysta	Styl	Gatunek	Wiek
Rzeczywisty	Alfred Sisley	Impresjonizm	Krajobraz	XIX
Przewidywany -Poziom Niższy	Alfred Sisley(42.29%)	Impresjonizm(78.46%)	Krajobraz(100.00%)	XIX(82.54%)
Przewidywany -Poziom Wyższy -Klasyfikator	Alfred Sisley(84.95%)	Impresjonizm(97.93%)	Krajobraz(99.97%)	XIX(97.93%)
Przewidywany -Poziom Wyższy -Sieć Neuronowa	Pierre Auguste Renoir(89.98%)	Impresjonizm(96.78%)	Krajobraz(100.00%)	XIX(100.00%)

Rysunek 6.4 Losowy obraz - test wieku.

Przewidywanie sieci neuronowej wyższego poziomu do identyfikacji artysty zmieniło decyzję na błędną. Pozostałe kategorie zanotowały wzrost pewności wobec podjętej już wcześniej poprawnej decyzji.

6.2 Obrazy spoza bazy danych

Do zaprezentowania działania systemu na obrazach nie znajdujących się w zbiorach danych wykorzystano obrazy polskich malarzy.



	Artysta	Styl	Gatunek	Wiek
Rzeczywisty	Jan Matejko	-	-	-
Przewidywany -Poziom Niższy	Gustave Courbet(48.78%)	Realizm(29.01%)	Malarstwo Rodzajowe(98.49%)	XIX(92.76%)
Przewidywany -Poziom Wyższy -Klasyfikator	Gustave Courbet(84.92%)	Realizm(97.13%)	Malarstwo Rodzajowe(70.00%)	XIX(99.20%)
Przewidywany -Poziom Wyższy -Sieć Neuronowa	Gustave Courbet(99.86%)	Realizm(78.23%)	Malarstwo Rodzajowe(100.00%)	XIX(100.00%)

Rysunek 6.5 Obraz Jana Matejki i przewidywane klasy.

Obraz Jana Matejki został przypisany do klas zbliżonych wizualnie do tego, co obraz przedstawia (Rys. 6.6).



Rysunek 6.6 Obraz należący do klas przewidywanych dla obrazu Matejki - „Pogrzeb w Ornans” Gustave’a Courbeta.



	Artysta	styl	Gatunek	Wiek
Rzeczywisty	Zdzisław Beksiński	-	-	-
Przewidywany -Poziom Niższy	Camille Corot(24.79%)	Ekspresjonizm Abstrakcyjny(25.92%)	Abstrakcja(70.36%)	XIX(69.61%)
Przewidywany -Poziom Wyższy -Klasyfikator	Camille Corot(17.50%)	Realizm(89.47%)	Krajobraz(48.00%)	XIX(77.26%)
Przewidywany -Poziom Wyższy -Sieć Neuronowa	Salvador Dali(87.67%)	Realizm(68.62%)	Abstrakcja(99.92%)	XX(100.00%)

Rysunek 6.7 Obraz Zdzisława Beksińskiego i przewidywane klasy.

Zaprezentowany obraz Zdzisława Beksińskiego trudno jednoznacznie zmieścić w ramy konkretnego stylu i gatunku, a artysta nie znajdował się w bazie danych. Powoduje to dużą różnorodność w przewidywaniach poszczególnych poziomów. Większość z nich jest zgodna z intuicją, choć zestawienie realizmu z abstrakcją sprawia wrażenie irracjonalnego - sieć jednak nie posiada wiedzy o tej irracjonalności.



	Artysta	styl	Gatunek	Wiek
Rzeczywisty	Aleksandra Mochel Modzelewska	-	-	-
Przewidywany -Poziom Niższy	David Burliuk(45.87%)	Pop Art(31.54%)	Abstrakcja(76.53%)	XX(99.56%)
Przewidywany -Poziom Wyższy -Klasyfikator	David Burliuk(64.56%)	Impresjonizm(38.95%)	Abstrakcja(85.35%)	XX(99.92%)
Przewidywany -Poziom Wyższy -Sieć Neuronowa	David Burliuk(99.57%)	Ekspresjonizm(51.26%)	Abstrakcja(99.96%)	XX(100.00%)

Rysunek 6.8 Obraz Aleksandry Mochel Modzelewskiej i przewidywane klasy.

Rysunek 6.8 przedstawia obraz z prywatnej kolekcji. Podobnie jak w przypadku obrazu Beksińskiego, jest on niejednoznaczny do klasyfikacji. W obrazach sugerowanego Davida Burliuka można zaobserwować pewne analogie wobec zamieszczonego powyżej obrazu, choć nie są tak wyraźne jak w przypadku Matejki i klas jemu sugerowanych.



Rysunek 6.9 Przykładowe obrazy Davida Burliuka.

Jak można zaobserwować na powyższych przykładach, system jest gotów do klasyfikacji różnych obrazów z bazy danych oraz spoza bazy danych, a jego wybory znajdują umotywowanie w obserwacjach dokonywanych przez człowieka.

Rozdział 7

Podsumowanie i wnioski

Celem pracy było zbadanie zdolności głębokich konwolucyjnych sieci neuronowych do rozpoznawania artysty, stylu, gatunku oraz wieku powstania obrazu malarskiego. Zaimplementowano dwupoziomowy system złożony z ośmiu sieci:

- czterech sieci konwolucyjnych, służących do identyfikacji każdego wyżej wymienionego parametru z osobna,
- czterech sieci nadrzędnych, które uwzględniają wyniki wszystkich sieci konwolucyjnych i wykorzystują zależności między nimi, aby poprawić dokładność przewidywań systemu.

Wyniki wykonanych badań prowadzą do następujących wniosków i spostrzeżeń:

1. Wykorzystanie zależności między różnymi kategoriami może znacznie poprawić zdolności systemu do dokładnej klasyfikacji w obrębie jednej kategorii. Największą poprawę odnotowano dla sieci do identyfikacji stylu. Dobrym wskaźnikiem w przypadku niebalansowanych baz danych, takich jak zastosowana w tej pracy, jest ważony wskaźnik F1 - dzięki zastosowaniu na wyższym poziomie sieci z jedną warstwą ukrytą jego wartość wzrosła z wynoszącej 0,47 do 0,71, a klasyfikator bazujący na maszynie wektorów nośnych odnotował jeszcze lepszy wynik - 0,72.
2. W niektórych zadaniach proste klasyfikatory dają lepsze efekty od bardziej wyrafinowanych sieci neuronowych.
3. Lepsze rezultaty sieci niższego poziomu byłyby możliwe do uzyskania przy zastosowaniu lepszej karty graficznej - wówczas możliwe byłyby eksperymenty z większym rozmiarem wejściowym obrazów lub większą liczebnością partii danych `batch_size`. Można się spodziewać, że lepsza dokładność sieci niższego poziomu przełożyłaby się na lepsze rezultaty na wyjściu całego systemu.
4. Augmentacja, mimo swojej powszechności stosowania, nie powinna być brana za obowiązkową przy każdym szkoleniu sieci neuronowej. Przykład sieci niższego rzędu do identyfikacji gatunku pokazał, że wprowadzenie jej nie gwarantuje lepszych wyników w każdej kategorii, a przedłuża czas obliczeń i zwiększa złożoność problemu.
5. Odpowiednie dostrojenie hiperparametrów, takich jak szybkość uczenia `learning rate` czy liczebność partii danych `batch_size`, są kluczowe dla wyszkolenia sieci o dobrej dokładności. Przy $lr = 0,1$ sieć do identyfikacji artystów osiągnęła dokładność zaledwie 1,28% na zbiorze testowym.

6. Dodanie warstw typu Dropout nie przyczyniło się do poprawy generalizacji i nie spowodowało poprawy dokładności sieci do identyfikacji stylów.
7. Zastosowanie dodatkowych wskaźników, takich jak czułość, precyzja czy F1, pozwala lepiej ocenić sieci mierzące się z zadaniem klasyfikacji na niezbalansowanym zbiorze danych. Sieć niższego rzędu do identyfikacji stulecia osiągnęła najlepsze efekty ze wszystkich pod kątem dokładności na zbiorze testowym i walidacyjnym, jednak jej średnia precyzja, czułość i wskaźnik F1 nie osiągały już przodujących wartości. Spowodowane jest to wyraźną dominacją dwóch klas - XIX i XX wieku - którą można zaobserwować na Rysunku 4.8.
8. Python jest odpowiednim językiem do badań i tworzenia sieci neuronowych. Powszechność jego użycia w tej dziedzinie ułatwia zdobycie niezbędnej wiedzy, a wsparcie dodatkowych bibliotek, takich jak wykorzystany w tej pracy TensorFlow, ułatwia implementację mniej i bardziej złożonych sieci.
9. Niedoskonałościami wykorzystanej w pracy bazy danych serwisu WikiArt.org, które utrudniły pracę nad sieciami, były:
 - duża rozbieżność między liczebnością poszczególnych klas w każdej kategorii,
 - zbyt mała liczebność niektórych klas, co spowodowało ich odrzucenie,
 - brak znanego roku powstania niektórych obrazów,
 - niewłaściwe zaklasyfikowanie części obrazów jako dzieł Nieznanego Artysty - wiązało się to z koniecznością przyjęcia nazwisk artystów na podstawie nazw plików,
 - błąd wprowadzany przez szkolenie z zachowaniem obrazów sklasyfikowanych jako Nieznany Gatunek również sugeruje, że mogły się w nim znajdować dzieła należące do innych gatunków.

Główną zaletą bazy danych była jej duża liczebność, która pozwoliła na wyszkolenie sieci mimo konieczności odrzucenia części obrazów ze względu na wyżej wymienione problemy.

10. Sieć niższego poziomu najgorzej sprawdziła się w tym zadaniu klasyfikacji, gdzie dobór klas był w dużym stopniu motywowany wywoływanymi przez obraz emocjami - czyli na stylu. Wynik jej został poprawiony dopiero przez uwzględnienie drugiej części definicji stylu, czyli okoliczności powstania - dlatego sieć wyższego rzędu, która uwzględniała przewidywanego artystę, wiek powstania i gatunek, uzyskała znacznie lepsze rezultaty.
11. Systemy dwupoziomowe są warte wypróbowania w klasyfikacji wielokategorialnej, gdy poszczególne kategorie są ze sobą powiązane. W niektórych zadaniach mogą zapewnić znaczny wzrost dokładności, dlatego przy coraz większej dostępności danych podpisanych pod kątem wielu kategorii, metoda ta może znajdować szersze zastosowanie niż dotychczas. Warta odnotowania jest również uniwersalność tej metody - w tej pracy na niższym poziomie zostały zastosowane sieci oparte na architekturze ResNet50, ale nic nie stoi na przeszkodzie, by spróbować wprowadzić sieci wyższego poziomu nad sieciami opartymi na innym rozwiązaniu - jedyne, co jest potrzebne do ich szkolenia, to przewidywane prawdopodobieństwa każdej klasy.

Literatura

- [1] C. C. Aggarwal. *Neural Networks and Deep Learning*. Springer, Cham, 2018.
- [2] E. Baeldung. Multiclass classification using support vector machines, 2020.
- [3] O. E. David, N. S. Netanyahu. DeepPainter: Painter Classification Using Deep Convolutional Autoencoders. *International Conference on Artificial Neural Networks*, 2016.
- [4] H. Fujiyoshi, T. Hirakawa, T. Yamashita. Deep learning-based image recognition for autonomous driving,. *IATSS Research, Volume 43, Issue 4*, 2019.
- [5] K. Fukushima. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position . *Biol. Cybernetic*, 1980.
- [6] A. Geron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., wydanie 1st, 2017.
- [7] S. Golara. Painting genre classification. Raport instytutowy, Stanford University, 2019. CS230-Fall 2019 Final Project Report.
- [8] A. Gołda, Łukasz Sanocki. Wstęp do sieci neuronowych. *Katedra Elektroniki AGH*.
- [9] E. Guresen, G. Kayakutlu. Definition of artificial neural networks with comparison to other networks. *Procedia Computer Science*, 3:426–433, 2011. World Conference on Information Technology.
- [10] S. Haldar. Deepartist : Identify artist from art, 2019. *kaggle.com*.
- [11] K. He, X. Zhang, S. Ren, J. SunK. Deep Residual Learning for Image Recognition. *Microsoft Research*, 2015.
- [12] D. Hubel, T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol*, 1962.
- [13] A. Joshi, A. Agrawal. Art style classification with self-trained ensemble of autoencoding transformations, 12 2020.
- [14] G. Kogan. Machine learning for artists: Convolutional neural networks, 2019. *kaggle.com*.
- [15] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner. Employing neocognitron neural network base ensemble classifiers to enhance efficiency of classification in handwritten digit datasets. *IEEE*, 1998.

- [16] J. A. Moka. Image classification with hsv color model processing. *Data Science Central*.
- [17] R. Nagyfi. The differences between artificial and biological neural networks, 2018.
- [18] K. Oh, K. Jung. GPU implementation of neural networks. *Pattern Recognition*, 2004.
- [19] K. Rana. Pooling layer — short and simple. *AI In Plain English*.
- [20] S. Raschka. *Python Machine Learning*. Packt Publishing, Birmingham, UK, 2016.
- [21] F. Rosenblatt. The perceptron - a perceiving and recognizing automaton, 1957. *Cornell Aeronautical Laboratory, Inc*.
- [22] D. Rumelhart, G. Hinton, R. Williams. Learning representations by back-propagating errors. *Nature volume 323*, 1986.
- [23] A. Rácz, D. Bajusz, K. Héberger. Effect of Dataset Size and Train/Test Split Ratios in QSAR/QSPR Multiclass Classification. *J Big Data 6*, 2016.
- [24] N. Saxena, Q. A. Kazmi, C. Pal, O. Vyas, M. Tech. Employing neocognitron neural network base ensemble classifiers to enhance efficiency of classification in handwritten digit datasets. 2011.
- [25] D. Shahid. Convolutional neural network.
- [26] C. Shorten, T. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *J Big Data 6*, 2016.
- [27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.
- [28] K. Yamaguchi, K. Sakamoto, T. Akabane, Y. Fujimoto. A Neural Network for Speaker-Independent Isolated Word Recognition . *First International Conference on Spoken Language Processing (ICSLP 90)*, 1990.
- [29] S. Yeung. Computer vision: Image filtering. *Stanford AI Lab*.
- [30] L. Yin. A summary of neural network layers. *Machine Learning for Li*.
- [31] T. Yiu. Understanding random forest, 2019. *towardsdatascience.com*.