

Życiorys



Marcin Furtak

Kierunek: Automatyka i Robotyka

Specjalność: Informatyka Przemysłowa

Urodzony: 27 czerwca 1995 r. we Wrocławiu

Numer indeksu: 269821

Urodziłem się 27 czerwca 1995 roku we Wrocławiu. W 2011 roku ukończyłem Gimnazjum im. Powstańców Warszawy w Piasecznie i rozpocząłem naukę w XIV Liceum Ogólnokształcącym im. Stanisława Staszica w Warszawie w klasie o profilu matematyczno-fizycznym. W roku 2014 zdałem maturę i rozpocząłem studia stacjonarne na Wydziale Mechatroniki Politechniki Warszawskiej. Po ukończeniu czwartego semestru studiów wybrałem specjalność Informatyka Przemysłowa na kierunku Automatyka i Robotyka.

.....

Politechnika Warszawska

W Y D Z I A Ł M E C H A T R O N I K I



Praca dyplomowa inżynierska

na kierunku Automatyka i Robotyka
w specjalności Robotyka

Projekt systemu elektronicznego podstawy jezdnej robota mobilnego
Kurier

numer pracy według wydziałowej ewidencji prac: 114B-ISP-WR/259155/1076469

Łukasz Zieliński

numer albumu 259155

promotor
Daniel Koguciuk

konsultacje
—

WARSZAWA 2017



Politechnika Warszawska

załącznik do zarządzenia nr 28/2016 r.
Rektora PW

„załącznik nr 3 do zarządzenia nr 24/2016 Rektora PW

Warszawa, 04.05.2017
miejscowość i data

Lukasz Zielinski

imię i nazwisko studenta

259155

numer albumu

Automatyka i Robotyka

kierunek studiów

OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

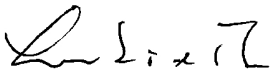
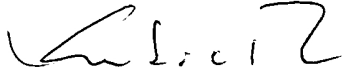
Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płyście kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

Lukasz Zielinski

czytelny podpis studenta

PRACA DYPLOMOWA inżynierska	
<u>Specjalność:</u> Robotyka	
<u>Instytut prowadzący specjalność:</u>	Instytut Automatyki i Robotyki
<u>Instytut prowadzący pracę:</u>	Instytut Automatyki i Robotyki
Temat pracy: Projekt systemu elektronicznego podstawy jezdnej robota mobilnego Kurier	
Temat pracy (w jęz. ang.): Design of electronic system of the base of the Kurier mobile robot	
Zakres pracy: <ol style="list-style-type: none"> 1. Przegląd dostępnych sensorów i aktuatorów podstawy robota mobilnego Kurier 2. Projekt i wykonanie systemu elektronicznego robota 3. Dobór sterownika silników 4. Projekt i wykonanie elektronicznego modułu sterującego podstawą robota 	
Podstawowe wymagania: <ol style="list-style-type: none"> 1. Umiejętność doboru sprzętu 2. Znajomość architektury ARM 3. Umiejętność projektowania obwodów drukowanych 	
Literatura: <ol style="list-style-type: none"> 1. STMicroelectronics, <i>Nota katalogowa mikrokontrolerów z rodziny STM32F4</i> 2. Kandlhofer, Martin and Steinbauer, Gerald, <i>Evaluating the impact of educational robotics on pupils' technical-and social-skills and science related attitudes</i> 3. Atmatzidou, Soumela and Demetriadis, Stavros, <i>Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences</i> 	
Słowa kluczowe: robot, mobilny, sterowanie, elektronika, kurier	
Praca dyplomowa jest realizowana we współpracy z przemysłem?	
Tak/Nie *,	
Imię i nazwisko dyplomanta:	Imię i nazwisko promotora:
Łukasz Zieliński	mgr inż. Daniel Koguciuk
	Imię i nazwisko konsultanta:
	-
Temat wydano dnia:	Termin ukończenia pracy:
01.10.2016	08.05.2017
Miejsce wykonywania praktyki przeddyplomowej: Induprogres, Warszawa	
Zatwierdzenie tematu	
	
Opiekun specjalności	Z-ca Dyrektora Instytutu odpowiedzialny za sprawy dydaktyczne

Streszczenie

Tematem pracy jest przegląd publikacji zawierających opis algorytmów klasyfikacji obiektów 3D wykorzystujących metody uczenia maszynowego. Problem klasyfikacji danych trójwymiarowych jest kluczowy w przypadku takich dziedzin jak automatyka przemysłowa, kontrola jakości, a przede wszystkim robotyka mobilna, w której bez możliwości semantycznej klasyfikacji rejestrowanych obiektów nie jest możliwe rozwiązanie wielu istotnych zagadnień. Dodatkowo, praca zawiera opis eksperymentalnego sprawdzenia efektywności działania wybranego klasyfikatora na rzeczywistych chmurach punktów.

Pierwszy rozdział pracy stanowi wprowadzenie w zagadnienie klasyfikacji 3D. Zaprezentowane w nim są główne związane z nim problemy. Przedstawione są również najczęściej wykorzystywane podejścia do konstrukcji klasyfikatorów. Opisana w nim jest również baza modeli 3D Princeton ModelNet oraz zbiór ModelNet40, służący do treningu i testowania rozwiązań dotyczących klasyfikacji 3D.

W kolejnych trzech rozdziałach zaprezentowane są najpopularniejsze podejścia dotyczące sposobu formatowania danych podawanych na wejście klasyfikatora. W każdej z nich opisane są publikacje, zawierające propozycje rozwiązań bazujących na opisanym we wstępie formacie danych wejściowych. Publikacje wybrane są spośród prac, które zawierają wyniki ewaluacji proponowanych rozwiązań na zbiorze ModelNet40. Na końcu trzeciego z rozdziałów zawarte jest podsumowanie dotyczące wszystkich typów rozwiązań, opisujące ich główne wady i zalety.

Następny rozdział opisuje eksperyment, polegający na próbie klasyfikacji chmur punktów, zebranych z rzeczywistych obiektów przy pomocy klasyfikatora nauczonego na syntetycznych danych ze zbioru ModelNet40. Rozdział zawiera uzasadnienie wyboru danego klasyfikatora do wykorzystania przy rozwiązywaniu powyżej zdefiniowanego zadania, jak również opis przebiegu eksperymentu. Zdefiniowane są w nim założenia, sposób, w jaki skonstruowano tor zbierania oraz przetwarzania danych oraz rezultaty eksperymentu.

W ostatnim rozdziale zawarto podsumowanie oraz wnioski z eksperymentu. Prócz tego zaprezentowane opisane zostaną przewidywane kierunki dalszego rozwoju rozwiązań związanych z klasyfikacją 3D.

*Overview of 3D machine learning classification algorithms with experimental evaluation
of chosen algorithm in task of real object classification*

Abstract

Thesis topic is overview of publications containing description of 3D classification algorithms using machine learning methods. 3D classification problem is essential in areas such as industrial automatics, quality control and, most of all, in mobile robotics, where without semantic categorization of registered objects it is impossible to solve many tasks. Additionally, thesis contains description of experiment consisting of testing chosen, pretrained on synthetic models classifier in task of recognition point clouds obtained from real objects.

Spis treści

Spis treści	13
1 Wstęp	17
1.1 Klasyfikacja 3D	17
1.1.1 Wprowadzenie w problematykę	17
1.2 Princeton ModelNet [?]	18
1.2.1 Opis bazy	18
1.2.2 Opis zbioru ModelNet40	19
1.3 Wybór publikacji	20
2 Klasyfikatory operujące na danych wejściowych w postaci wokseli	22
2.1 Opis	22
2.2 3D ShapeNets: A Deep Representation for Volumetric Shapes	23
2.2.1 Opis	23
2.2.2 Budowa klasyfikatora	23
2.2.3 Rezultaty	23
2.3 VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition [?]	24
2.3.1 Opis	24
2.3.2 Budowa klasyfikatora	25
2.3.3 Rezultaty	25
2.4 Generative and Discriminative Voxel Modeling with Convolutional Neural Networks [?]	26

2.4.1	Opis	26
2.4.2	Budowa klasyfikatora	27
2.4.3	Rezultaty	28
3	Klasyfikatory operujące na danych w postaci widoków 2D	29
3.1	Opis	29
3.2	Multi-view Convolutional Neural Networks for 3d Shape Recognition [?] . .	30
3.2.1	Opis	30
3.2.2	Budowa klasyfikatora	30
3.2.3	Rezultaty	31
3.3	Exploiting the PANORAMA Representation for Convolutional Neural Network Classification and Retrieval [?]	31
3.3.1	Opis	31
3.3.2	Budowa klasyfikatora	33
3.3.3	Rezultaty	33
3.4	Ensemble of PANORAMA-based convolutional neural networks for 3D model classification and retrieval [?]	34
3.4.1	Opis	34
3.4.2	Budowa klasyfikatora	35
3.4.3	Rezultaty	35
4	Klasyfikator operujący na danych wejściowych w postaci chmur punktów	37
4.1	Opis	37
4.2	PointNet:Deep Learning on Point Sets for 3D Classification and Segmentation [?]	38
4.2.1	Opis	38
4.2.2	Budowa klasyfikatora	38
4.2.3	Rezultaty	39
4.3	Podsumowanie	41

5	Eksperymentalne ewaluacja efektywności klasyfikacji na zebranych danych 3D	43
5.1	Opis problemu	43
5.2	Tor przetwarzania informacji	43
5.3	Zebrane dane	44
5.4	Otrzymane wyniki	45
6	Podsumowanie	47
6.1	Dyskusja otrzymanych wyników	47
6.2	Przewidywane kierunki dalszego rozwoju rozwiązań	48

Rozdział 1

Wstęp

1.1 Klasyfikacja 3D

1.1.1 Wprowadzenie w problematykę

Wzrost mocy obliczeniowej komputerów, pozwalający na przetwarzanie dużej ilości danych, oraz większa dostępność sprzętu rejestrującego obraz 2,5D - to jest obraz, który zawiera informację o głębi rejestrowanego piksela - sprawiają, że zagadnienie klasyfikacji obiektów 3D cieszy się obecnie coraz większą popularnością. W efekcie powstaje coraz więcej klasyfikatorów, których głównym celem jest osiągnięcie jak najlepszej efektywności semantycznej kategoryzacji obiektów. Innymi istotnymi wymaganiami jest ich jak najmniejsza złożoność, pozwalająca na szybszy trening, oraz zdolność radzenia sobie z brakującymi danymi czy też szumami.

Klasyfikatory można podzielić ze względu na format wykorzystywanych danych wejściowych na dwie główne kategorie: klasyfikatory operujące na danych dwu- lub trójwymiarowych. W przypadku reprezentacji dwuwymiarowej, danymi wejściowymi są zawsze obrazy o określonej szerokości i wysokości. Jeżeli chodzi o reprezentację trójwymiarową, najpopularniejszy jest opis danych przy pomocy wokseli. Woksele stanowią odpowiednik pikseli w przestrzeni trójwymiarowej. Innym sposobami opisu danych jest opięcie na nich siatki (mesh [?]) czy wykorzystywanie chmury punktów [?].

W przypadku analizowanych publikacji, żaden z proponowanych klasyfikatorów nie operował na danych jedno- czy też cztero- lub więcej wymiarowych. Fakt ten wynika z kilku czynników. Przede wszystkim klasyfikatory dwuwymiarowe wykorzystywane są powszechnie również w zagadnieniach związanych z klasyfikacją 2D, takich jak rozpoznawanie twa-

rzy [?], pojazdów [?], znaków drogowych [?] czy też w zagadnieniach związanych z medycyną [?]. W efekcie próby opisanie obiektów trójwymiarowych w przestrzeni dwuwymiarowej, a następnie klasyfikacja przy pomocy sprawdzonych metod jest często prostszą i efektywniejszą opcją, niż próba konstrukcji wielowymiarowego deskryptora. W przypadku klasyfikatorów trójwymiarowych, operujących bezpośrednio na danych 3D większość z przeanalizowanych prac opiera się na wykorzystaniu metod znanych z klasyfikacji 2D na danych wzbogaconych o jeden wymiar. Ostatecznie, problem reprezentacji można sprowadzić do problemu odpowiedniej konwersji danych trójwymiarowych, która mimo utraty wymiaru nie spowoduje utraty kluczowych informacji o obiekcie (w przypadku reprezentacji dwuwymiarowej) bądź do sposobu przeniesienia rozwiązań wykorzystywanych do klasyfikacji 2D na kanwę trzech wymiarów (w przypadku reprezentacji trójwymiarowej).

Innym, mniej intuicyjnym jest podział ze względu na wykorzystywanie przez klasyfikator metod uczenia maszynowego. W przypadku będących w zdecydowanej mniejszości klasyfikatorów nie korzystających z metod uczenia maszynowego obiekt kategoryzuje się przy pomocy jego parametrów geometrycznych, takich jak pole powierzchni, środek ciężkości czy stosunek pola do obwodu. Jeżeli chodzi o rozwiązania wykorzystujące metody uczenia maszynowego, przede wszystkim wykorzystuje się sztuczne sieci neuronowe. Klasyfikator samodzielnie, na podstawie odpowiednio przygotowanego zbioru danych, jest w stanie dokonać generalizacji parametrów przyjętego modelu matematycznego zjawiska. W przypadku kategoryzacji obiektów parametrami są cechy charakterystyczne dla danego obiektu. Cechy te mogą dla człowieka wydawać się nieintuicyjne, jednak jak pokazują testy zastosowanie odpowiednich klasyfikatorów 2D choćby w takich zadaniach jak rozpoznawanie twarzy daje lepsze rezultaty niż ocena człowieka [?].

1.2 Princeton ModelNet [?]

1.2.1 Opis bazy

Prócz braków mocy obliczeniowej, inną kluczową przeszkodą w rozwijaniu metod klasyfikacji 3D był brak odpowiedniej ilości odpowiednio przygotowanych danych. Podczas gdy łatwo dostępne są bazy z obrazami 2D o konkretnym profilu, takich baz z obiektami trójwymiarowymi brakuje. Ponownie jednak, dzięki rozwojowi technologii, dzięki której łatwiej dostępne są metody tworzenia czy też akwizycji modeli 3D możliwa jest konstrukcja baz rozwiązujących ten problem.

ModelNet to zbiór obiektów, przygotowany przez badaczy z Princeton, zawierający 127 915 modeli CAD, podzielonych na 662 unikatowe kategorie. Modele dostępne są w formacie .off (Object File Format), zawierającym informacje o liczbie punktów oraz ich współrzędnych, liczbie ścian oraz, dla każdej z nich, liczbę punktów, na których jest opięta i ich indeksy oraz (nieobowiązkowo) liczbę krawędzi. Zbiór ten przygotowany został z myślą o pracach badawczych w dziedzinie klasyfikacji i segmentacji obiektów 3D. Pierwszą pracą, która opierała się na danych w nim zawartych była publikacja 3DShapeNets autorstwa grupy naukowców odpowiedzialnych za skonstruowanie zbioru.

Dane znajdujące się w zbiorze zostały wyszukane przy pomocy internetowych wyszukiwarek modeli 3D. Każdy z obiektów poddano klasyfikacji przez człowieka. Ludzie ci zostali zrekrutowani w ramach programu Amazon Mechanical Turk. Obiekty, które nie zostały przez większość z badanych zaklasyfikowane w ten sam sposób zostały odrzucone.

W ramach 662 występujących klas autorzy wyodrębnili dwa podzbiory, ModelNet10 i ModelNet40, zawierające odpowiednio 10 i 40 klas obiektów najczęściej spotykanych w rzeczywistym świecie. Popularność obiektów określano na bazie statystyk znajdujących się w bazie danych SUN. W ramach ModelNet10 i ModelNet40 znajdujące się w nich dane zostały poddane dodatkowej klasyfikacji przez autorów.

1.2.2 Opis zbioru ModelNet40

Zbiór ModelNet40 składa się z 12311 modeli CAD, podzielonych na 40 kategorii. Dodatkowo, w ramach każdej z kategorii wydzielone są obiekty służące do treningu i testów. Dla całego zbioru modeli treningowych jest 9843, zaś testowych 2468.

Modele w ramach zbioru są przystosowane do wykorzystania w zastosowaniach wykorzystujących metody uczenia maszynowego. Są one odszumione, w przeciwieństwie jednak do modeli znajdujących się w zbiorze ModelNet10 nie są zorientowane w jednym kierunku. Powoduje to konieczność konstrukcji klasyfikatora odpornego na rotację danych bądź też samodzielne zorientowanie modeli. Zbiór zorientowany w ramach publikacji ORION [?] jest dostępny, jednak wszystkie poniżej opisane publikacje operują na danych ze zbioru oryginalnego.

W poniższej tabeli znajduje się zestawienie, prezentujące nazwy kategorii zawartych w ramach zbioru ModelNet40, liczbę modeli oraz liczbę modeli przeznaczonych do treningu dla każdej z nich.

1.3 Wybór publikacji

Wszystkie publikacje opisane w pracy proponują rozwiązania oparte o metody uczenia maszynowego. W każdej z nich autorzy dokonali ewaluacji swojego rozwiązania na zbiorze ModelNet40. W kolejnych rozdziałach przedstawione jest krótkie wprowadzenie, dotyczące typu reprezentacji, na których bazują opisane w danym rozdziale klasyfikatory, oraz opis jednej lub więcej prac bazujących na danym typie reprezentacji.

Nazwa kategorii	Liczba modeli	Liczba modeli treningowych	Procent modeli przeznaczonych do treningu
Airplane	726	626	86%
Bathtub	156	106	68%
Bed	615	515	84%
Bench	193	173	90%
Bookshelf	672	572	85%
Bottle	435	335	77%
Bowl	84	64	76%
Car	297	197	66%
Chair	989	889	90%
Cone	187	167	89%
Cup	99	79	80%
Curtain	158	138	87%
Desk	286	200	70%
Door	129	109	84%
Dresser	286	200	70%
Flower Pot	169	149	88%
Glass Box	271	171	63%
Guitar	255	155	61%
Keyboard	165	145	88%
Lamp	144	124	86%
Laptop	169	149	88%
Mantel	384	284	74%
Monitor	565	465	82%
Night Stand	286	200	70%
Person	108	88	81%
Piano	331	231	70%
Plant	340	240	71%
Radio	124	104	84%
Range Hood	215	115	53%
Sink	148	128	86%
Sofa	780	680	87%
Stairs	144	124	86%
Stool	110	90	82%
Table	492	392	80%
Tent	183	163	89%
Toilet	444	344	77%
TV Stand	367	267	73%
Vase	575	475	83%
Wardrobe	107	88	82%
XBOX	123	103	84%
Suma	12311	9844	80%

Rozdział 2

Klasyfikatory operujące na danych wejściowych w postaci wokseli

2.1 Opis

Większość klasyfikatorów 3D operuje na danych reprezentowanych przy pomocy wokseli. Wokselem nazywamy obiekt matematyczny, posiadający 3 wymiary. Analogicznie do piksela, który jest podstawową jednostką cyfrowego obrazu 2D, woksel uznawany jest za podstawową jednostkę obrazu trójwymiarowego.

Duża liczba powstających klasyfikatorów działających w oparciu o reprezentację wokselową wynika z kilku czynników. Najważniejszym jest fakt, że klasyfikator można skonstruować na zasadzie podobnej do tych działających na obrazach 2D - różnicą jest konieczność uwzględnienia wystąpienia trzeciego wymiaru. Dodatkowo, przejście z innego typu reprezentacji na reprezentację wokselową jest często znacznie prostsze niż odwrotna transformacja danych. Z chmury punktów można utworzyć woksele poprzez skonstruowanie odpowiedniej funkcji, uzależniającej wypełnienie wokseli od liczby czy rozmieszczenia znajdujących się w nim punktów. W przypadku transformacji z obrazów 2D sytuacja wygląda analogicznie.

Poniżej opisano trzy z publikacji bazujące na reprezentacji wokselowej. Zaproponowane w każdej z nich rozwiązanie zostało przez autorów przetestowane na zbiorze ModelNet40. Pierwsza z nich przedstawia propozycję klasyfikatora autorstwa twórców bazy ModelNet. Jest to zarazem pierwsza praca wykorzystująca zbiór ModelNet40. Następna praca przedstawia koncepcję konstrukcji funkcji, przypisującej wokselowi zawierającym określoną grupę punktów odpowiednią wartość. Ostatnia z prac zawiera propozycję klasyfikatora o

najwyższej efektywności klasyfikacji ze wszystkich klasyfikatorów opartych na reprezentacji wokselowej.

2.2 3D ShapeNets: A Deep Representation for Volumetric Shapes

2.2.1 Opis

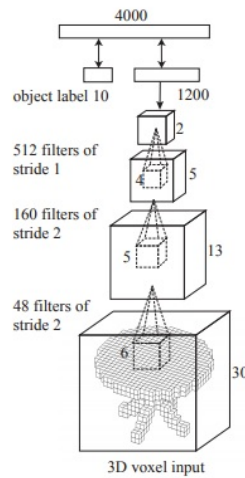
ShapeNets to pierwsza praca, wykorzystująca metody głębokiego uczenia maszynowego w zagadnieniu klasyfikacji 3D. Głównym jej założeniem jest przyjęcie probabilistycznego modelu reprezentacji obiektu. W ramach siatki wokselowej o wymiarach 30x30x30 wokseli przy pomocy wartości 0 oznacza się woksle, które nie są elementami powierzchni obiektu. Woksle stanowiące powierzchnie oznaczane są przy pomocy wartości 1.

2.2.2 Budowa klasyfikatora

Autorzy proponują klasyfikator działający w oparciu o wyznaczanie dystrybucji danych w modelach zdefiniowanych powyżej. W tym celu wykorzystują Convolutional Deep Belief Network, sieć będącą modyfikacją sieci Deep Belief Networks. Sieci DBN służą do wyznaczania modeli dystrybucji w obrazach 2D. Autorzy w celu przystosowania ich do działania na obiektach 3D decydują się na zastosowanie konwolucji w celu redukcji parametrów, od których zależne będą wyniki klasyfikacji. Dodatkowo, co nietypowe dla sieci konwolucyjnych, decydują się oni na brak łączenia warstw ukrytych. Taka konstrukcja sieci sprawia, że jest ona bardziej podatna na wszelkie szумы, ale jednocześnie pozwala na lepszą rekonstrukcję obiektu w przypadku posiadania o nim ograniczonych informacji. Konstrukcja sieci zaprezentowana jest poniżej:

2.2.3 Rezultaty

W ramach zbioru ModelNet40 klasyfikator zaproponowany w publikacji osiąga efektywność na poziomie 77,32%. Jest to, zwłaszcza w obliczu najefektywniejszych obecnie rozwiązań, wartość przeciętna. Istotne jest jednak, aby mieć na uwadze, że praca ShapeNets była pracą pionierską w swojej dziedzinie. Jej autorzy jako pierwsi podjęli się prób klasyfikacji 3D przy pomocy metod głębokiego uczenia maszynowego, czym de facto zapoczątkowali nowy rozdział w historii tej dziedziny.



2.3 VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition [?]

2.3.1 Opis

Problem wykorzystania konwolucyjnych sieci neuronowych na trójwymiarowych danych wejściowych wymaga odpowiedniej ich reprezentacji. Autorzy publikacji proponują zastosowanie siatki zajętości (occupancy grid). Jest to struktura, opisująca przestrzeń trójwymiarową przy pomocy zbioru wokseli. Każdy z nich posiada określoną wartość, zależną od liczby znajdujących się w nim punktów.

Kluczowym zagadnieniem związanym z siatką zajętości jest sposób zdefiniowania funkcji, która zbiorowi punktów zawartemu w wokselsie przypisuje wartość, uznawaną jako wartość danego woksela. Od sposobu konstrukcji tej funkcji zależą parametry obiektu w przestrzeni siatki. Autorzy prezentują 3 koncepcje sformułowania wyżej wymienionej zależności.

Pierwszą z nich jest binarny opis każdego woksela (binary grid). Każdy z wokseli ma 2 stany - może on być zajęty bądź pusty. Równanie określające logarytmiczne prawdopodobieństwo zajętości każdego z wokseli prezentuje się następująco: $l_{ijk}^t = l_{ijk}^{t-1} + z^t * l_{occ} + (1 - z^t)l_{free}$, gdzie i, j, k reprezentują współrzędne konkretnego woksela. Jako z^t rozumiemy punkt numer t , zarejestrowany przy pomocy pomiaru uwzględniającego głębokość obrazu. Wartość z^t może być równa 1 (w przypadku gdy punkt zawiera się w wokselsie) bądź zero (gdy znajduje się on poza nim). Parametry l_{occ} oraz l_{free} określają logarytmiczne prawdopodobieństwo bycia woksela w stanie zajętych bądź pustym. Wartości te, zgodnie z [?] określone są jako stałe i równe kolejno -1,38 oraz 1,38. Początkowa wartość prawdo-

podobieństwa zajętości dla każdego z wokseli wynosi 50% (jest to równoważne wartości $l_{ijk}^0 = 0$). Wartość l_{ijk}^t została ograniczona przez autorów wartościami -4 oraz 4 w celu uniknięcia potencjalnych problemów związanych z błędami numerycznymi. W przypadku zastosowania tego typu opisu siatki wartość l_{ijk}^t jest podawana na wejście sieci neuronowej.

Druga opisana możliwość to reprezentacja gęstościowa (density grid). Każdy woksel ma w niej ciągłą gęstość, przez którą rozumiemy prawdopodobieństwo zablokowania przez niego wiązki sensora odległościowego. Do opisu opisu wykorzystano funkcje wyprowadzone w [?]: $\alpha_{ijk}^t = \alpha_{ijk}^{t-1} + z^t$ oraz $\beta_{ijk}^t = \beta_{ijk}^{t-1} + (1 - z^t)$. Parametry α oraz β są określone w przedziale $[0,1]$ i dobrane są w taki sposób, aby przy ich pomocy można było opisać eulerowską funkcję beta [?] (konkretnie $f(\alpha, \beta)$ jest funkcją beta Eulera). Wartości α_{ijk}^0 oraz β_{ijk}^0 są równe 1 dla dowolnego z wokseli. Wartość woksela wyznacza się według wzoru $\mu_{ijk}^t = \frac{\alpha_{ijk}^t}{\alpha_{ijk}^t + \beta_{ijk}^t}$. Tak otrzymana wartość jest parametrem podawanym na wejście sieci.

Ostatnim z zaprezentowanych sposobów opisu funkcji wypełnienia siatki jest określanie wartości wokseli na bazie trafień (hit grid). Jest to najprostszy sposób opisu, nie uwzględniający w żaden sposób informacji o pustych przestrzeniach pomiędzy zarejestrowanymi punktami. Każdemu wokselowi przypisuje się początkową wartość $h_{ijk}^0=0$, modyfikowaną następnie według zależności: $h_{ijk}^t = \min(h_{ijk}^{t-1}, 1)$.

2.3.2 Budowa klasyfikatora

Autorzy opisują chmurę punktów przy pomocy siatki zajętości o rozmiarze 32x32x32 woksele. Tak otrzymane są następnie podawane na dwie, szeregowo połączone warstwy konwolucyjne, o wymiarach kolejno 32x5x2 i 32x3x1. Następnie, dane przekazywane są do warstwy pooling, zastępującej każdy blok o rozmiarze 2x2x2 zawartą w nim wartością maksymalną. Ostatecznie trafiają one na dwie warstwy fully connected. Każda z nich przekazuje na wyjście wektor zawierający kolejno 128 oraz K elementów, gdzie K jest liczbą klas, zawartych w danym zbiorze.

2.3.3 Rezultaty

Efektywność osiągnięta przez klasyfikator zależała od sposobu implementacji rotacji danych. Wprowadzenie jej jest kluczowe dla efektywnego działania modelu, ponieważ obiekty zbioru ModelNet40 nie są jednakowo zorientowane w przestrzeni. Poniżej zaprezentowana tabela przedstawia efektywność klasyfikacji w zależności od tego, czy rotacja była implementowana w trakcie treningu, czy testów.

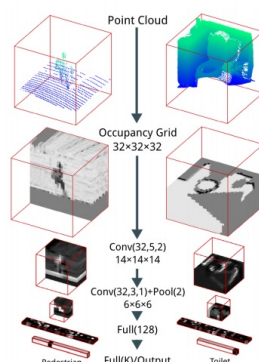


Tabela 2.1: My caption

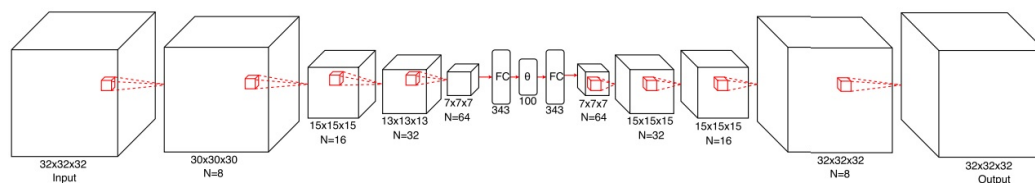
Rotacja podczas nauczania	Rotacja podczas testowania	Efektywność klasyfikacji (w procentach)
Nie	Nie	61
Nie	Tak	69
Tak	Nie	82
Tak	Tak	83

2.4 Generative and Discriminative Voxel Modeling with Convolutional Neural Networks [?]

2.4.1 Opis

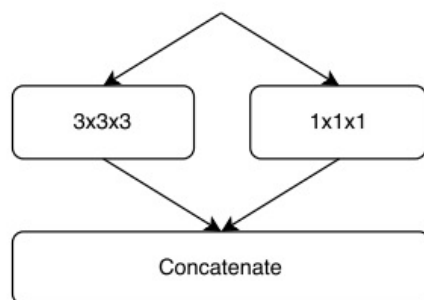
Rozwiązanie zaprezentowane w publikacji operuje na koncepcjach analogicznych do pracy VoxNet. Główna różnica polega na określeniu sposobu definicji funkcji opisującej probabilistyczną wartość wypełnienia woksela. W przypadku stosowania siatki zajętości konieczny jest kompromis pomiędzy dokładnością klasyfikacji a złożonością obliczeniową danego rozwiązania. Zastosowanie gęstszej siatki zdecydowanie poprawia efektywność klasyfikacji, istotnie wydłuża jednak czas zarówno uczenia, jak i ewaluacji klasyfikatora.

W opisywanej pracy autorzy zdecydowali się na obejście problemu definicji funkcji probabilistycznej poprzez zastosowanie biblioteki Variational Autoencoder [?] (w skrócie VAE). Jest to framework wykorzystywany do mapowania określonego rodzaju danych wejściowych na probabilistyczny wektor, określający z jakim prawdopodobieństwem dana cecha zawiera się w analizowanym zbiorze danych. Dodatkowo, pozwala ono na rekonstrukcję obiektu posiadając jedynie wektor cech. Zasada działania obu członów jest analogiczna.

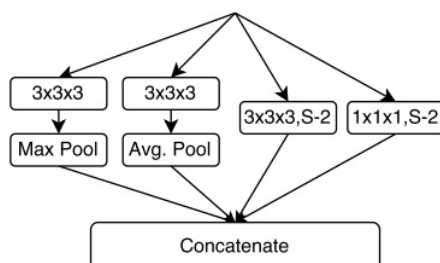


2.4.2 Budowa klasyfikatora

Proponowane rozwiązanie wykorzystuje połączenie struktury ResNet [?] z proponowaną przez autorów siecią Voxception, zwane dalej VRN. Zasada działania bloku Voxception opiera się na konkatenacji danych uzyskanych przy zastosowaniu filtrów o rozmiarze $1 \times 1 \times 1$ oraz $3 \times 3 \times 3$. Tak uzyskane dane podawane są na sieć konwolucyjną, która ma możliwość generalizacji cech globalnych (poprzez wybór danych uzyskanych przez zastosowanie filtra $1 \times 1 \times 1$) czy też lokalnych (poprzez dane przefiltrowane przy użyciu siatki o rozmiarze $3 \times 3 \times 3$).

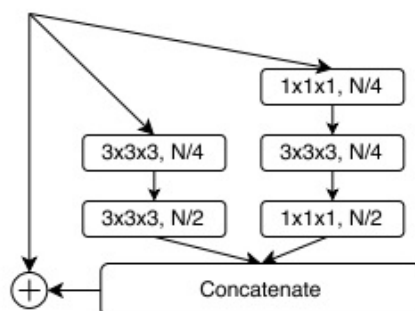


Rozszerzeniem tej architektury jest zastosowanie downsamplingu poprzez zastosowanie filtra max-pooling i average-pooling na danych wejściowych. W efekcie sieć otrzymuje ogólniejsze, łatwiejsze do generalizacji informacje o cechach obiektu.

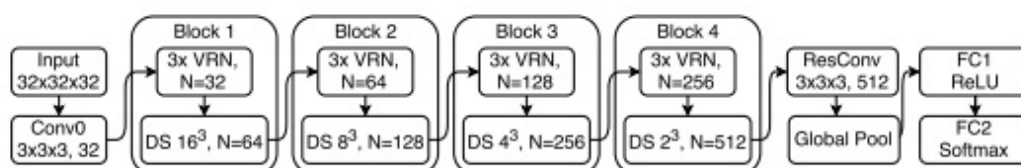


Blok VRN łączy cechy bloku Voxception z architekturą ResNet. Dodatkowo, w celu zwiększenia efektywności tak skonstruowanego bloku pierwsze stosowane warstwy zawierają dwukrotnie mniej filtrów niż warstwy końcowe. Dodatkowo, na przestrzeni kolejnych

warstw prawdopodobieństwo zachowania wartości uzyskanych na wejściu zmniejszana jest z 1 do 0,25.



Ostatecznie, struktura klasyfikatora zaprezentowana przez autorów składa się z 4 bloków. W ramach każdego z nich wchodzi 3 połączone sieci VRN oraz sieć Voxception-Downsample. Dane na wyjściu powyższego toru informacji podawane są na wejście standardowej sieci konwolucyjnej, z której przekazywane są do warstwy global pooling oraz dwóch warstw fully-connected.



2.4.3 Rezultaty

Klasyfikator VRN Ensemble w przypadku zbioru ModelNet40 cechuje się średnią efektywnością klasyfikacji na poziomie 95,54%. W momencie publikacji było to rozwiązanie o najwyższej efektywności klasyfikacji. W momencie pisania pracy jest to nadal klasyfikator o najwyższej skuteczności kategoryzacji wśród rozwiązań operujących na danych w postaci wokseli.

Rozdział 3

Klasyfikatory operujące na danych w postaci widoków 2D

3.1 Opis

W przeciwieństwie do reprezentacji wokselowej, modele działające na bazie widoków de facto operują na obrazie lub obrazach 2D. Każdy z nich jest klasyfikowany niezależnie, następnie zaś wyniki poszczególnych klasyfikacji są analizowane wspólnie. Na bazie otrzymanych wyników cząstkowych określone jest prawdopodobieństwo, z jakim dany obiekt należy do konkretnej klasy.

Rozwiązanie to ma niezaprzeczalne plusy. Problem klasyfikacji 2D jest powszechniej analizowany, dzięki czemu możliwych rozwiązań jest znacznie więcej. Dodatkowo, istnieje więcej potencjalnych danych testowych, dzięki którym możliwe jest douczenie klasyfikatora. Co więcej, pracę wykorzystującą dane dwuwymiarowe są bardziej odporne na szumy i utratę danych względem reprezentacji wokselowej. Głównym problemem tego podejścia jest natomiast konieczność przekształcenia obrazu 3D do jednego lub wielu widoków 2D. Jest to zadanie wymagające czasu oraz sporej mocy obliczeniowej, przede wszystkim jednak wymaga ona odpowiedniego algorytmu, pozwalającego dla określonego typu obiektów uzyskać obraz reprezentujący jego kluczowe cechy. Transformacji można dokonać na wiele sposobów - zaprezentowane poniżej prace opierają się na rzutowaniu obiektu względem określonej płaszczyzny.

Poniżej, podobnie jak w przypadku klasyfikatorów wokselowych, zaprezentowany jest opis 3 prac. Pierwsza z nich jest jednocześnie pierwszą pracą opartą o reprezentację w postaci widoków, której ewaluacji dokonano na zbiorze ModelNet40. Kolejna praca wpro-

wadza nowatorski sposób opisu obiektu 3D przy pomocy reprezentacji dwuwymiarowej, osiągając zarazem wyższą efektywność klasyfikacji. Ostatnia publikacja jest rozwinięciem idei zawartych w drugiej z prac. Jest to praca, która aktualnie cechuje się najwyższą efektywnością klasyfikacji ze wszystkich rozwiązań przetestowanych na zbiorze ModelNet40 niezależnie od sposobu reprezentacji danych wejściowych.

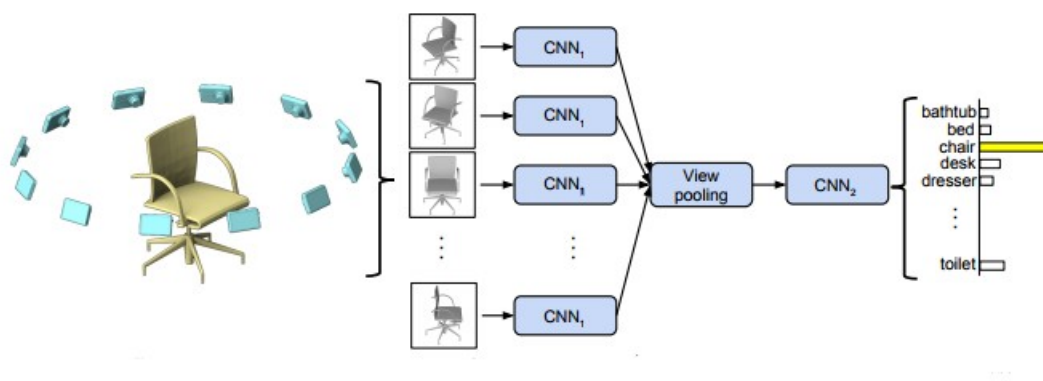
3.2 Multi-view Convolutional Neural Networks for 3d Shape Recognition [?]

3.2.1 Opis

Klasyfikator MVCNN to pierwsze rozwiązanie, operujące na reprezentacji dwuwymiarowej w celu skategoryzowania obiektów ze zbioru ModelNet40. Konceptyjnie, problem analizowany w pracy nie różni się istotnie od problemu klasyfikacji 2D. Kluczowym zagadnieniem jest zatem sposób przejścia z przestrzeni 3D do 2D. Autorzy pracy proponują obejście tego problemu - ich pomysł opiera się na zebraniu dwuwymiarowych danych poprzez rejestrację obiektu z różnych, przesuniętych względem siebie pozycji. Tym samym główną trudnością, z którą muszą się zmierzyć jest opracowanie sposobu generalizacji danych z różnych widoków w celu określenia kategorii obiektu.

3.2.2 Budowa klasyfikatora

Schemat ideowy klasyfikatora zaprezentowany jest poniżej:



Każdy z widoków (ich domyślnie założona liczba to 12) zostaje zaklasyfikowany przy pomocy tej samej konwolucyjnej sieci neuronowej do danej klasy. Następnie informacje

uzyskane na pierwszym etapie są brane pod uwagę w celu dokonania ostatecznej klasyfikacji obiektu do jednej ze znanych kategorii. Nowatorskim rozwiązaniem, znacznie usprawniającym działanie sieci zarówno pod kątem efektywności klasyfikacji, jak i złożoności obliczeniowej, jest zastosowanie drugiej warstwy konwolucyjnej, poprzedzonej warstwą View pooling. W przeciwieństwie do najprostszego algorytmu, polegającego na uśrednieniu informacji ze wszystkich widoków, rozwiązanie zaproponowane w artykule pozwala na swoiste złożenie widoków poprzez nałożenie przez warstwę view pooling wag na każdy z widoków, dzięki czemu sieć w większym stopniu wykorzystuje dane otrzymane z istotniejszych widoków, tym samym zmniejszając ryzyko popełnienia błędu klasyfikacji.

3.2.3 Rezultaty

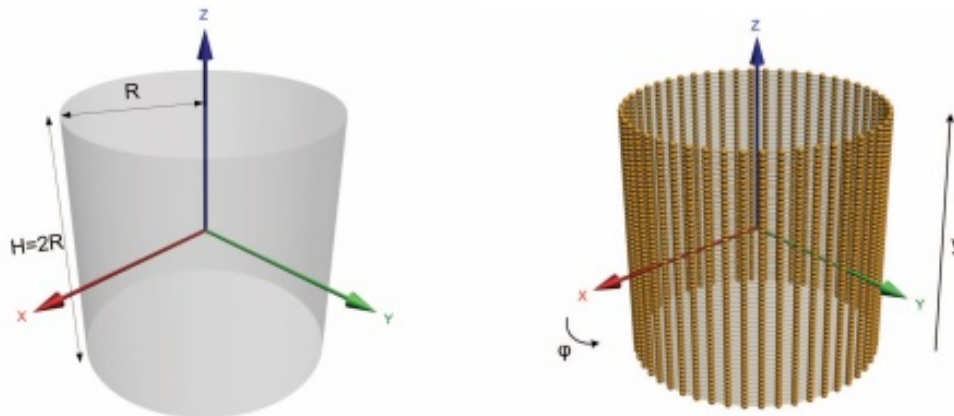
Główną zaletą zaprezentowanego rozwiązania jest prostota otrzymanego modelu – w przeciwieństwie do mocno rozbudowanych rozwiązań opartych o dane w postaci wokseli, sieć składa się jedynie z 3 warstw, połączonych ze sobą szeregowo. Jakość klasyfikacji jest bardzo wysoka – w przypadku zastosowania 12 warstw i pre-treningu na zbiorze danych ImageNet model klasyfikował z dokładnością 89,9%. Ze względu na wspomnianą wcześniej dużą bazę dostępnych obrazów 2D możliwe jest również dodanie większej liczby warstw, co potencjalnie może poprawić jakość klasyfikacji.

3.3 Exploiting the PANORAMA Representation for Convolutional Neural Network Classification and Retrieval [?]

3.3.1 Opis

Publikacja opiera się na opracowanym przez autorów systemie reprezentacji cech trójwymiarowego obiektu PANORAMA [?]. Pozwala on, wykorzystując operacje geometryczne, na wyznaczenie unikalnej dla danego trójwymiarowego obiektu dwuwymiarowej reprezentacji. Obiekt rzutuje się na walec o promieniu R oraz wysokości $H=2R$. Centrum układu współrzędnych znajduje się w środku ciężkości walca, oś z jest zaś wyznaczana przez jego oś symetrii. Wartość R określa się jako podwojoną maksymalną odległość powierzchni modelu od jego geometrycznego środka ciężkości. Boczna powierzchnia walca jest parametryzowana poprzez równanie $s(\phi, y)$, gdzie $\phi \in [0, 2\pi]$, zaś $y \in [0, H]$. ϕ określa kąt w

płaszczyźnie XY , y przekazuje informacje dotyczące wysokości. Parametry próbkowane są (kolejno) z częstotliwością $2B$ oraz B , gdzie $B = 180$. Różnica w częstotliwości próbkowania spowodowana jest próbą uwzględnienia zależności pomiędzy obwodem a wysokością walca. Uwzględniając powyższe efektem parametryzacji jest zbiór punktów $s(\phi_u, y_v)$, gdzie $\phi_u = u * 2\pi / (2B)$, $y_v = v * H / B$, $u \in [0, 2B-1]$, $v \in [0, B-1]$.

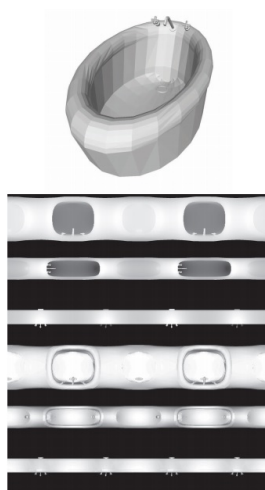


Każdy z punktów $s(\phi_u, y_v)$ jest następnie wyznaczany dla wszystkich określonych powyżej v na dwa sposoby, pozwalające określić dwie istotne cechy powierzchni. Pierwsza z nich określana jest jako Spatial Distribution Map (w skrócie: SDM) i niesie informacje o pozycji powierzchni w przestrzeni trójwymiarowej. Wyznacza się ją przy pomocy równania $s_1(\phi_u, y_v) = pos(\phi_u, y_v)$, gdzie $pos(\phi_u, y_v)$ definiuje się jako odległość od punktu c_v do najdalszego mu punktu przecięcia prostej wyprowadzonej z punktu c_v pod kątem ϕ_u i powierzchnią obiektu. Punkt c_v znajduje się w osi walca i określa środek płaszczyzny przecinającej walec na wysokości y_v . Wartość SDM zawarta jest w przedziale $[0, R]$. Drugą z cech nazwano Normals' Deviation Map (NDM). Jest ona miarą informacji o orientacji obiektu w przestrzeni. Wartość NDM oznacza się przy pomocy równania $s_2(\phi_u, y_v) = |\cos(ang(\phi_u, y_v))|^n$, w którym $n=2$. Kąt $ang(\phi_u, y_v)$ to kąt pomiędzy promieniem wyprowadzonym z punktu c_v pod kątem ϕ_u i normalną najbardziej oddalonego od c_v trójkąta, powstałego na skutek przecięcia powierzchni modelu z wyżej wymienionym promieniem.

Po wyznaczenie SDM oraz NDM, przy pomocy wykorzystującego je algorytmu SYMPAN [?] dokonywana jest normalizacja położenia obiektu względem płaszczyzny symetrii. Założenie jednakowej orientacji obiektów nie jest spełnione dla wielu baz danych obiektów 3D (również dla ModelNetu), dlatego krok ten jest niezwykle istotny.

3.3.2 Budowa klasyfikatora

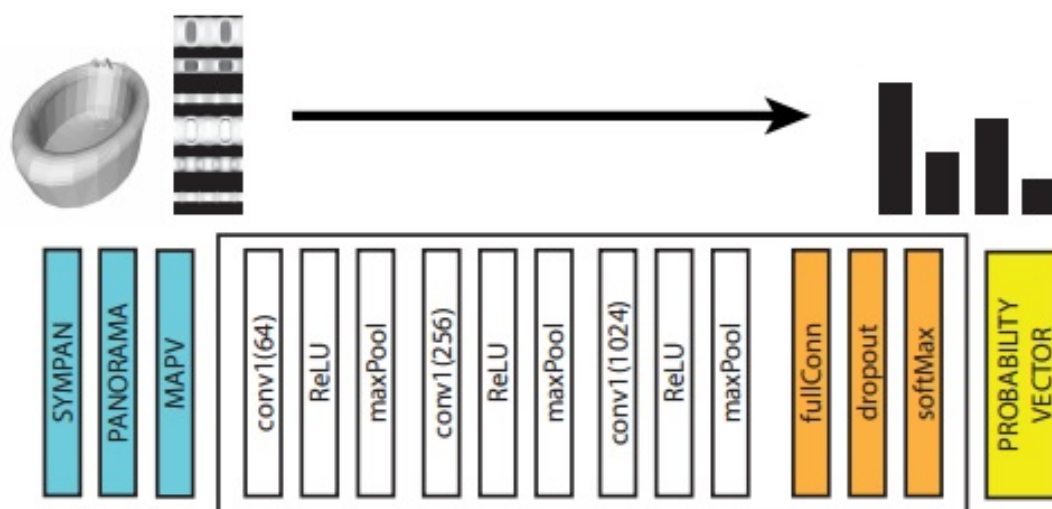
Struktura klasyfikatora opiera się o zastosowanie konwolucyjnej sieci neuronowej. Na jej wejście podawany jest dwuwymiarowy obraz, zawierający jednocześnie 6 składowych - NDM oraz SDM, wyznaczone oddzielnie dla każdej z głównych osi układu współrzędnych. Dodatkowo, dla każdego z tych obrazów jego pierwsza połowa jest kopiowana i dodawana na końcu obrazu. Operacja ta ma na celu wyeliminowanie potencjalnych problemów związanych z zawijaniem się obrazu. Pierwotnie obraz składowy ma wymiar 540x1080 pikseli.



Autorzy na bazie eksperymentów wyznaczyli stopień jego kompresji do 10% oryginalnego rozmiaru, to jest 54x108 pikseli. Kompresja ta nie wpływa w sposób znaczący na otrzymaną efektywność klasyfikacji, natomiast pozytywnie wpływa na szybkość procesu uczenia. Sama sieć składa się z 3 warstw konwolucyjnych, po których dodane są warstwa ReLU [?] i max-pooling [?] o rozmiarze 2x2 piksele. Powiązane z każdą kolejną warstwą parametry to wektor cech o rozmiarze 64, 256 oraz 1024, filtr o rozmiarze 5,5 i 3 oraz warstwa padding o rozmiarze 2 dla każdej z warstw.

3.3.3 Rezultaty

W momencie publikacji PANORAMA-NN cechowała się najwyższą efektywnością klasyfikacji wśród klasyfikatorów działających w oparciu o widoki. Średni poziom efektywności klasyfikacji modeli ze zbioru ModelNet40 wynosił 90,7%. Najważniejszym osiągnięciem publikacji było jednak pokazanie, że reprezentacja PANORAMA doskonale nadaje się do wykorzystania przy problemie klasyfikacji 3D. Pomimo prostej konstrukcji sieci i wykorzystaniu jedynie podstawowych informacji, uzyskanych przy pomocy parametrów SDM

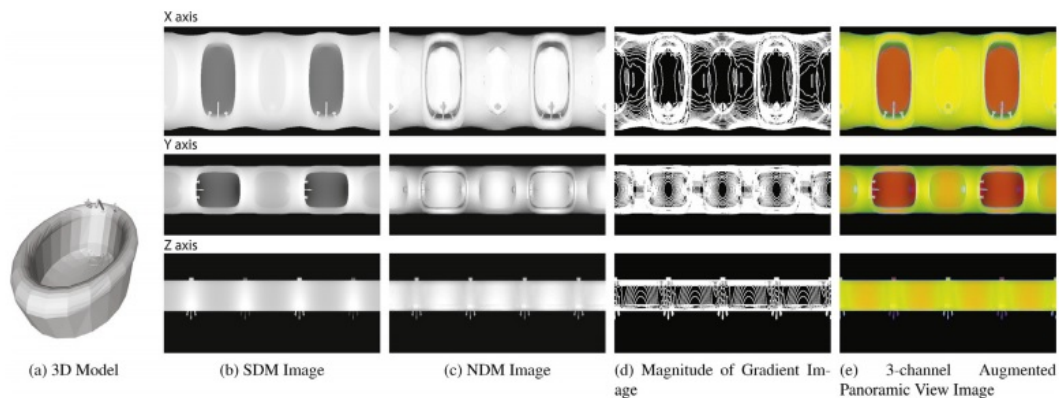


oraz NDM efektywność klasyfikacji przekroczyła 90%. Ogromny potencjał rozwojowy tego rozwiązania najlepiej obrazuje poniżej opisana praca.

3.4 Ensemble of PANORAMA-based convolutional neural networks for 3D model classification and retrieval [?]

3.4.1 Opis

Praca ta stanowi bezpośrednią kontynuację publikacji „Exploiting the PANORAMA Representation for Convolutional Neural Network Classification and Retrieval”. Autorzy w ten sam sposób przygotowują dane wejściowe, również przekazywane na wejście konwoLucyjnej sieci neuronowej. Elementem różniącym nowy klasyfikator jest wzbogacenie danych o informacje dotyczące wartości wielkości gradientu obrazu NDM. Modyfikacji uległa również struktura danych podawana na wejście sieci. W przeciwieństwie do poprzedniego rozwiązania, PANORAMA-ENN wykorzystuje trójkanałowy obraz, będący efektem konkatenacji obrazów SDM, NDM oraz wartości gradientu NDM. Każdy kanał odpowiada jednej z osi układu współrzędnych. Wynikowy obraz przetwarzany jest w sposób analogiczny jak obraz wykorzystywany w klasyfikatorze PANORAMA-NN. Różnicą jest początkowy, a tym samym zredukowany rozmiar obrazu, wynoszące kolejno 1080 na 1080 oraz 108 na 108 pikseli.



3.4.2 Budowa klasyfikatora

W celu optymalnego wykorzystania informacji z każdego z kanałów, związanych z osiami układu współrzędnych, autorzy zdecydowali się rozbudować strukturę sieci wykorzystanej w klasyfikаторze PANORAMA-NN o dodanie dwóch analogicznych struktur. Każda ze struktur ma za zadanie klasyfikację jedynie wykorzystując dane z osi X, Y lub Z. Otrzymane w ten sposób trzy probabilistyczne wektory, określające prawdopodobieństwo przynależności obiektu do danej klasy są następnie uśredniane. Tak uzyskany wynik maksymalny jest uznawany za kategorię, do której przynależy obiekt.



3.4.3 Rezultaty

Efektywność klasyfikacji osiągnięta przez autorów publikacji wyniosła na zbiorze ModelNet40 95,56%. Wynik ten jest o 0,02% niż rezultat osiągnięty przez klasyfikator VRN Ensemble. Tym samym PANORAMA-ENN jest, w momencie pisania pracy i w oparciu o zbiór ModelNet40, najefektywniejszym istniejącym klasyfikatorem 3D. Co ciekawe,

w przypadku ModelNet10, cechującego się znacznie mniejszą liczbą danych, klasyfikator osiągnął nieznacznie gorsze rezultatu od VRN (96,85% w porównaniu do 97,14%). Niemniej, w przypadku zastosowania większych zbiorów danych PANORAMA-ENN jest efektywniejsza. Dodatkowo, w publikacji wykorzystano jedynie jedną dodatkową cechę względem klasyfikatora PANORAMA-NN. Jak zaznaczają autorzy, w przypadku powstania baz uwzględniających choćby kolory obiektów dodanie kolejnych cech nie stanowiłoby żadnego problemu, potencjalnie natomiast stwarzałoby możliwość jeszcze efektywniejszej klasyfikacji.

Rozdział 4

Klasyfikator operujący na danych wejściowych w postaci chmur punktów

4.1 Opis

W przypadku zarówno klasyfikatorów opartych na reprezentacji wokselowej, jak i widokowej, głównym problemem jest konieczność transformacji danych wejściowych. W przypadku zbierania rzeczywistych danych są one przedstawione w postaci chmury punktów. Aby przekształcić ją do postaci używanej przez powyższe klasyfikatory, niezbędne jest wykonanie dodatkowych operacji. Co istotne, operacje te nie są w swojej naturze trywialne. Opracowanie algorytmu, pozwalającego uogólnić sposób przeprowadzenia tej transformacji nie jest zadaniem łatwym. Przykładowo, dobór odpowiedniego rozmiaru woksela wymaga wiedzy dotyczącej analizowanego zagadnienia, a także wiedzy o dystrybucji punktów w analizowanej chmurze.

Odpowiedzią na powyższe problemy jest konstrukcja klasyfikatora działającego bezpośrednio na chmurach punktów. Klasyfikator taki, ze względu na brak konieczności manipulacji danymi wejściowymi, miałby szansę być znacznie szybszy. W przypadku efektywności zbliżonej do innych rozwiązań byłby on opcją preferowaną w przypadku wszelkich zastosowań wymagających optymalizacji czasu działania.

Poniżej opisano pracę jako pierwszą prezentującą rozwiązanie wykorzystujące jako dane wejściowe chmury punktów.

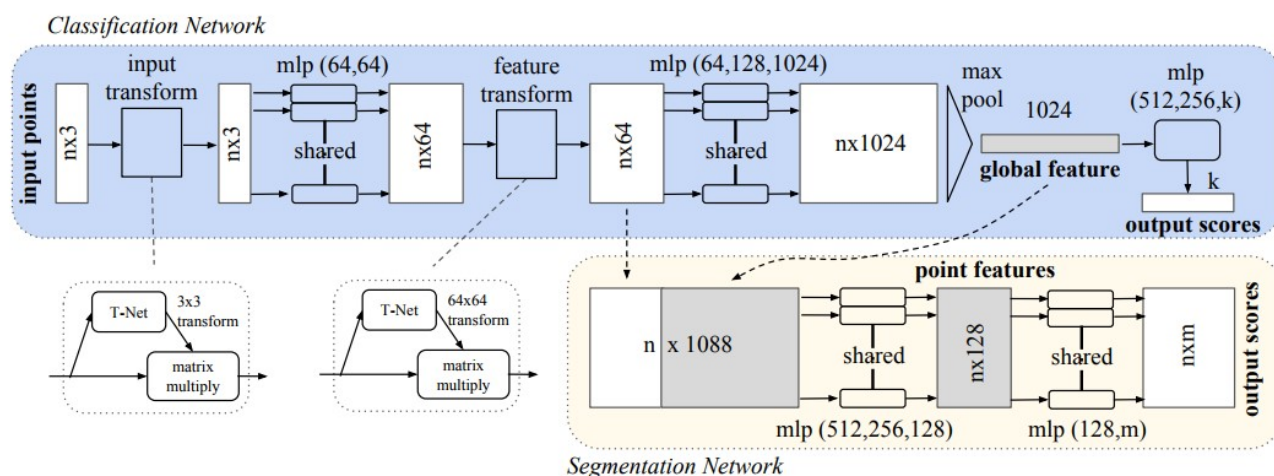
4.2 PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation [?]

4.2.1 Opis

Algorytm PointNet miał u swojej podstawy spełnienie następujących założeń: - działanie na danych wejściowych w postaci nieuporządkowanej chmury punktów; - uwzględnienie semantycznej informacji związanej z położeniem punktów w przestrzeni (punkty najmniej oddalone od siebie są ze sobą powiązane); - niezmiennność (invariance) klasyfikacji przy zastosowaniu operacji geometrycznych, takich jak rotacja czy translacja. W przypadku spełnienia powyższych założeń, uzyskano by klasyfikator zdolny do działania na bezpośrednio zebranych danych. Dodatkowo, klasyfikator ten byłby w stanie potencjalnie radzić sobie z mocno danymi o dużym zaszumieniu. W przypadku szumów bliskich danym kluczowym dla reprezentacji obiektu nie powinny one wpływać na otrzymane wyniki klasyfikacji dzięki powiązaniu ich poprzez ich bliską odległość z tymi punktami. Jeżeli szumy byłyby punktami oddalonymi od obiektu, dzięki uwzględnieniu semantycznej informacji o ich odległości również nie miałyby one istotnego wpływu na jakość klasyfikacji.

4.2.2 Budowa klasyfikatora

Spełnienie poniższych założeń zagwarantowała poniższa konstrukcja toru przetwarzania danych wejściowych: Ideowo, sieć składa się z trzech głównych modułów: a) szeregowo



połączone dwie sieci MLP, otrzymujące na wejście (kolejno) informacje o położeniu kluczowych punktów chmury oraz jej cechy; b) struktury służącej do łączenia informacji

globalnych oraz lokalnych (w przypadku segmentacji); c) warstwę max pooling, pełniącą rolę funkcji symetrycznej. Pierwszy moduł składa się z części przetwarzającej informacje o położeniu punktów (odpowiada za to pierwsza sieć MLP) oraz z części odpowiedzialnej za ekstrakcję cech kluczowych globalnych. W przypadku obu z sieci wprowadzone są dodatkowe transformacje na danych wejściowych. Input transform służy modyfikacji początkowej chmury punktów poprzez transformację każdego z punktów przy pomocy struktury będącej uproszczoną siecią PointNet do postaci macierzy 3×3 , mnożonej następnie przez macierz jednostkową. Analogiczny proces występuje w warstwie Feature transform. Warstwy odpowiedzialne za modyfikację pierwotnych parametrów danych wejściowych zostały dodane przez twórców na podstawie przeprowadzonych eksperymentów – sieć zawierająca input transform oraz feature transform osiągała efektywność klasyfikacji o 2,1 punkta procentowego wyższą od sieci bazowej. Dodatkowo, powyższe moduły zapewniają utrzymanie wysokiej efektywności klasyfikacji w przypadku geometrycznych transformacji danych wejściowych. Klasyfikacja obiektu dokonywana jest przez sieć MLP, otrzymującą na wejściu zestaw cech globalnych danego modelu. Ekstrakcja tych cech odbywa się poprzez zastosowanie funkcji max pooling, która w prosty sposób – poprzez wybór jednego punktu z danego zestawu znajdujących się blisko siebie punktów – wydobywa punkty kluczowe dla danego modelu. Dodatkowo, jest ona funkcją symetryczną – dzięki jej zastosowaniu klasyfikacja nie traci na efektywności przy utracie czy zakłóceniu danych. Sieć PointNet wyróżnia się na tle sieci opartych o dane w postaci wokseli przede wszystkim, jeżeli chodzi o jej zdolność do efektywnej klasyfikacji nawet w przypadku usunięcia dużej ilości danych wejściowych bądź też ich zaszumienia. Efektywność klasyfikacji osiąga wartości poniżej 70% poprawnie zaklasyfikowanych modeli dopiero w przypadku, gdy z bazowej chmury punktów usunięte zostanie 80% danych. W przypadku sieci VoxNet spadek efektywności do takiej wartości wymaga usunięcia 20% danych.

4.2.3 Rezultaty

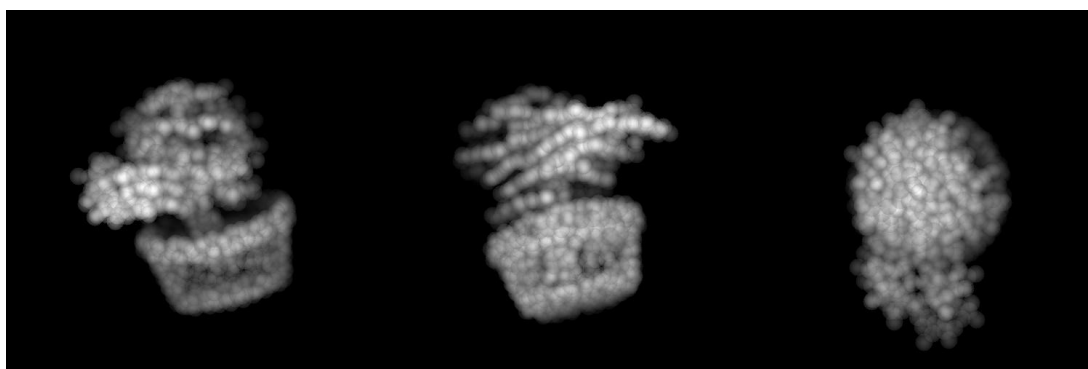
Średnia dokładność ewaluacji zbioru ModelNet40 wyniosła 87,93 procent (średnia wartość ewaluacji dla każdej z klas – 85,19 procent). W przypadku klas airplane, guitar, keyboard, laptop (10% wszystkich analizowanych klas) sieć dokładnie zaklasyfikowała obiekt w 100% przypadków. Najmniejszą dokładność klasyfikacji uzyskano dla klasy flower pot – wyniosła ona zaledwie 25%; jest to jedyna klasa, której dokładność klasyfikacji wyniosła poniżej 60%.

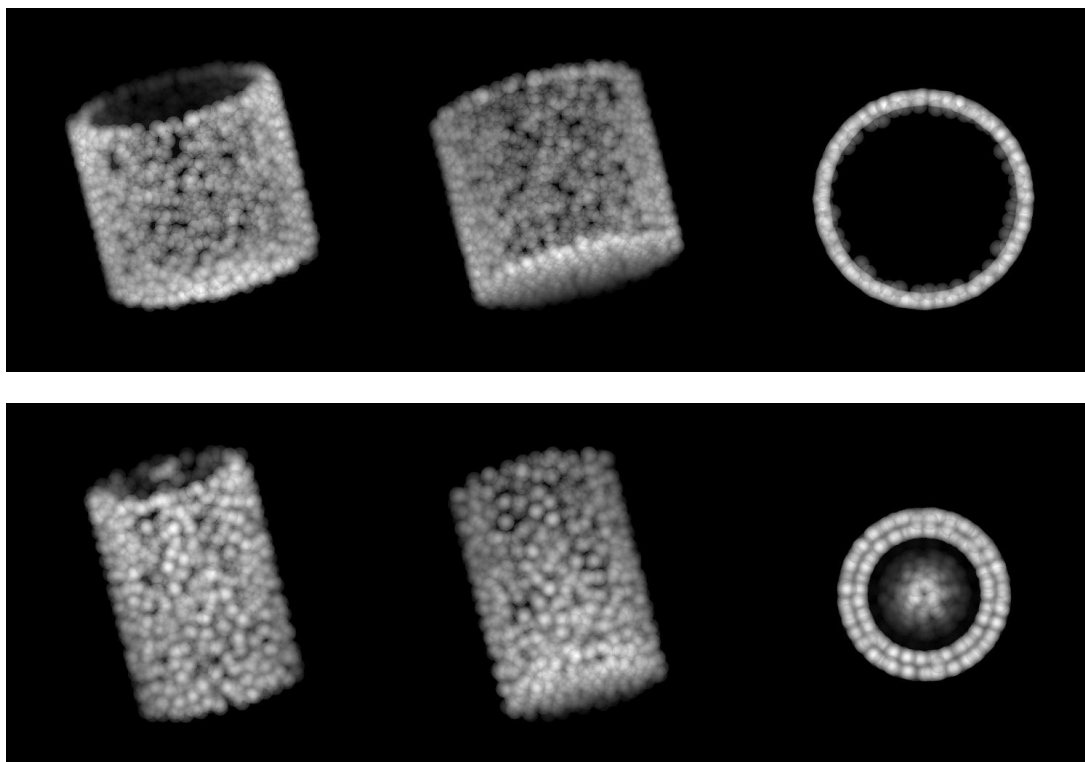
Powodami, dla których rezultaty klasyfikacji dla niektórych klas są wyraźnie gorsze są dwa. Pierwszy, mniej istotny, to różna liczba obiektów, na których uczona jest sieć – dla

Tabela 4.1: Nazwa kategorii oraz procent poprawnie zaklasyfikowanych należących do niej obiektów

Airplane	100	Chair	95	Glass Box	93	Person	85	Stool	85
Bathtub	84	Cone	95	Guitar	100	Piano	88	Table	81
Bed	97	Cup	60	Keyboard	100	Plant	73	Tent	95
Bench	65	Curtain	90	Lamp	95	Radio	75	Toilet	99
Bookshelf	92	Desk	81,4	Laptop	100	Range Hood	91	TV Stand	82
Bottle	94	Door	90	Mantel	95	Sink	70	Vase	76
Bowl	90	Dresser	73,3	Monitor	94	Sofa	97	Wardrobe	70
Car	97	Flower Pot	25	Night Stand	69,8	Stairs	90	XBOX	75

przykładu w przypadku klasy Airplane jest ich znacznie więcej niż dla klasy XBOX. Drugi powód to podobieństwo między klasami. Wszystkie z klas, których obiekty sieć była w stanie zaklasyfikować bezbłędnie, są unikalne – ich obiekty nie przypominają obiektów innych klas z sieci. W przypadku choćby klasy Flower Pot obiekty do niej należące mają wiele cech wspólnych z obiektami kilku innych klas.





4.3 Podsumowanie

Zaprezentowane powyżej publikacje opierają się na różnych typach reprezentacji danych. To, co łączy je to wysoka efektywność klasyfikacji (wszystkie rozwiązania oprócz ShapeNets, będącego pierwszym opracowanym klasyfikatorem osiągają średnią efektywność klasyfikacji na poziomie 85%) oraz wykorzystanie metod uczenia maszynowego.

Dwa najefektywniejsze rozwiązania, pomimo korzystania z różnego typu danych wejściowych, osiągnęły niemal identyczną efektywność klasyfikacji. To, co je łączy to struktura oparta na wielokrotnym wykorzystaniu konkretnego bloku przetwarzania informacji. W przypadku klasyfikatora VRN Ensemble jest to szeregowe połączenie bloków, składających się z sieci VRN oraz Voxception-Downsample. Taki sposób łączenia wyżej opisanych bloków ma na celu precyzyjniejsze wyznaczenie istotnych cech obiektu. PANORAMA-ENN opiera się natomiast na równoległym połączeniu trzech torów przetwarzania informacji względem jednego toru przetwarzania wykorzystanego w klasyfikatorze PANORAMA. Równoległe połączenie tych bloków ma na celu dokonanie wyznaczenia cech dla każdego z kanałów x,y,z. Rozwiązanie to pozytywnie wpływa na jakość klasyfikacji.

W przypadku obu wyżej wymienionych rozwiązań większe pole do rozwoju posiada PANORAMA-ENN. Konstrukcja równoległa pozwala na łatwiejsze rozszerzenie, na przykład (o czym wspominają autorzy) o kanał odpowiadający za kolor obiektu. Przy pojawie-

niu się większej liczby zbiorów posiadających dane zawierające więcej cech istnieje duża szansa na poprawienie efektywności klasyfikacji wyżej wymienionego klasyfikatora. Jeżeli chodzi o rozwiązanie VRN Ensemble, modyfikacja struktury szeregowej nie jest zadaniem trywialnym. Wyznaczenie optymalnej struktury w publikacji zostało dokonane na bazie drogi eksperymentalnej. Tym samym manipulacja wykorzystywanymi danymi wejściowymi spowodowałaby konieczność ponownych testów, ponieważ ustalenie struktury na bazie ogólnej reguły nie jest możliwe.

Na specjalną uwagę zasługuje klasyfikator PointNet. Efektywność klasyfikacji zbioru ModelNet40 z jego wykorzystaniem jest wyraźnie niższa niż przy pomocy najlepszych klasyfikatorów wokselowych i widokowych. Co istotne, jest on od nich jednak znacznie szybszy, zarówno jeżeli chodzi o trening, jak i testy. Związane jest to ze znacznie mniejszą liczbą operacji na danych wejściowych. W porównaniu do sieci MVCNN czy 3D CNN [?], osiągających podobną efektywność klasyfikacji, PointNet wymaga kolejno 8 i 141 razy mniej operacji w celu klasyfikacji danego obiektu (440MFLOP/obiekt w porównaniu do 3633 i 62057 MFLOP/obiekt). Złożoność obliczeniowa PointNetu jest liniowa (zależy jedynie od liczby danych wejściowych), podczas gdy złożoność MVCNN rośnie kwadratowo, zaś 3D CNN kubicznie wraz ze wzrostem liczby danych. Charakterystyki te są typowe dla wszystkich rozwiązań opartych o dany rodzaj danych wejściowych. Dodatkowo, PointNet cechuje się najprostszą strukturą. Jest to zatem rozwiązanie perspektywiczne. Wykorzystanie choćby struktury szeregowej (jak w VRN-E) czy równoległej (jak w PANORAMA-ENN) może poprawić efektywność klasyfikacji, jednocześnie znacząco nie wpływając na złożoność obliczeniową tak skonstruowanego klasyfikatora.

Rozdział 5

Eksperymentalne ewaluacja efektywności klasyfikacji na zebranych danych 3D

5.1 Opis problemu

Wszystkie z zaprezentowanych klasyfikatorów uczone są na sztucznie przygotowanych modelach. Zasadnym jest pytanie, czy tak wytrenowany klasyfikator jest w stanie poradzić sobie z zadaniem klasyfikacji rzeczywistych obiektów. Jeżeli jakość klasyfikacji nie uległaby znaczącemu pogorszeniu oznaczałoby to możliwość znacznego usprawnienia procesu uczenia nowych modeli - niepotrzebne byłoby zbieranie rzeczywistych danych, a jedynie przygotowanie syntetycznego zbioru.

Do eksperymentu, mającego odpowiedzieć na powyższe pytanie wybrano klasyfikator PointNet. Z analizowanych rozwiązań to właśnie on najlepiej radził sobie z mocno zaszumionymi oraz brakującymi danymi. Ponieważ dane zebrane ze środowiska w większości charakteryzują się powyższymi cechami, to właśnie PointNet wydaje się mieć największe szanse powodzenia w efektywnej ich klasyfikacji.

5.2 Tor przetwarzania informacji

Dane testowe pozyskiwano przy pomocy kamery ASUS Xtion Pro Live. Obsługiwana ona była przy pomocy programu kinfu remake [?], będącego nową wersją programu kinfu [?].

Program ten umożliwia zebranie tak zwanego SLAMu, czyli trójwymiarowego modelu złożonego z kilku ujęć kamery 2,5D. SLAM zarejestrować można poprzez obejście obiektu z kamerą zawierającą czujnik odległości. KinFu automatycznie, w czasie rzeczywistym dokonuje łączenia zebranych chmur punktów w reprezentację jednego obiektu. Istotnym jest, aby przejścia pomiędzy kolejnymi ujęciami nie były zbyt dynamiczne - powodują one pojawienie się szumów. Powyższe środowisko wykorzystywano ze względu na fakt, że kluczowym w procesie ewaluacji sieci było posiadanie danych reprezentujących w pełni trójwymiarowy model. W przypadku rejestracji jedynie z jednego ujęcia zostałoby utraconych bardzo dużo informacji, zaś model uczony na danych o jednorodnie rozmieszczonych punktach nie miałby żadnych szans powodzenia w zadaniu klasyfikacji zebranych danych.

Zebrane w pierwszym etapie przetwarzania dane wczytywane były do programu Heuros [?], w którym dokonywana była segmentacja obiektów. W przypadku, gdy segmentacja nie wykonywana była poprawnie dane były zbierane ponownie bądź też, w przypadku możliwości niezależnej segmentacji dwóch fragmentów tego samego obiektu, fragmenty te były łączone przy pomocy programu dokonującego konkatenacji dwóch chmur punktów. Program ten, w oparciu o bibliotekę PCL [?], wczytywał dwa pliki wyjściowe zawierające chmury fragmentów obiektu, na wyjściu zwracając obiekt będący ich złożeniem.

Wykorzystywana implementacja PointNetu [?] na wejściu wymaga chmury znormalizowanej do unitarnej przestrzeni sferycznej, składającej się z 2048 punktów. Z wykorzystaniem biblioteki PCL napisano program, dokonujący najpierw normalizacji chmury punktów, następnie zaś jej downsampling. Downsampling wykonywano przy pomocy aplikacji wykorzystującej zasadę siatki wokselowej. W ramach woksela określonego rozmiaru zbierane są wszystkie zawarte w nim punkty, które zastępowane są jednym punktem, posiadającym współrzędne środka ciężkości wszystkich punktów. Operacje zdecydowano się na wykonywanie w tej kolejności ze względu na potencjalną możliwość utraty informacji w przypadku uprzedniego usunięcia dużej liczby punktów.

5.3 Zebrane dane

Zdecydowano się na zebranie danych z 4 klas: cup, chair, monitor oraz keyboard. W ramach każdej z nich zebrano 6 chmur danych obiektów. W przypadku 2 obiektów klasy chair konieczne było dokonanie konkatenacji zebranych danych w celu uzyskania odpowiedniej chmury. Ze względu na niską jakość zebranych danych klasy cup zdecydowano się na zebranie dodatkowych 5 chmur obiektów klasy people. Dodatkowo, ze względu na

bardzo niską jakość jednej z zebranych chmur zdecydowano się na odrzucenie jednej z chmur reprezentującej obiekty klasy monitor.

Dane zebrane w ramach każdej z klas różniły się istotnie pod względem ich liczby punktów. W efekcie przy downsamplingu konieczne było zastosowanie różnych rozmiarów filtra siatki wokselowej. W poniższej tabeli zamieszczone są informacje o pierwotnej liczbie punktów, liczbie punktów po downsamplingu oraz rozmiarze zastosowanej siatki wokselowej.

Osiągnięcie wartości równej 2048 przy konieczności manipulacji rozmiarem siatki i zmiennej liczbie punktów danych wejściowych było zadaniem niemożliwym. Zdecydowano się na zaakceptowanie wystąpienia po downsamplingu liczby punktów oddalonej od wartości oczekiwanej o 10%, a następnie dodanie lub usunięcie liczby punktów wymaganej do osiągnięcia wartości 2048. Działanie to wprowadza dodatkowe szumy, z którymi powinien poradzić sobie badany model.

Wszystkie obiekty ze zbioru ModelNet40 są zorientowane w ten sam sposób względem osi z. Ręcznie dokonano orientacji względem osi z uprzednio przygotowanych modeli. Ponieważ dane wejściowe obracane są jedynie wokół niej nie było konieczne ustalenie konkretnej orientacji względem osi x oraz osi y.

5.4 Otrzymane wyniki

Niestety, żaden z modeli nie został zaklasyfikowany poprawnie. Poniżej zaprezentowana jest tabela z otrzymanymi wynikami klasyfikacji (z uwzględnieniem operacji orientowania względem osi z) oraz przykładowe zdjęcia modeli przed i po całym torze przetwarzania danych.

Rozdział 6

Podsumowanie

6.1 Dyskusja otrzymanych wyników

Brak sukcesu w zadaniu klasyfikacji przygotowanych modeli wynikać może z wielu czynników. Najistotniejszymi wydają się duże szумы, którymi obarczony był każdy z modeli. Dodatkowo, znaczny, negatywny wpływ na klasyfikację mogła mieć orientacja przestrzenna modeli. Dokonano jej ręcznie, jednak daleka była ona od perfekcji. Obrót danych treningowych podczas nauczania o losowe kąty względem każdej z osi mógłby znacząco poprawić odporność na zaburzenia orientacji.

Ewentualnym rozwiązaniem, które mogłoby spowodować poprawę jakości klasyfikacji byłoby zebranie większej ilości danych. Pozwoliłoby to na pewniejszą ewaluację wyuczonej sieci. Dodatkowo, umożliwiłoby ewentualne douczenie sieci przy pomocy pewnej części zebranego zbioru. Być może klasyfikator nauczony na danych syntetycznych, dodatkowo douczony małą liczbą danych zebranych podczas rejestracji rzeczywistych obiektów cechowałby się efektywnością zbliżoną do klasyfikacji danych tego samego typu co dane treningowe. Ze względu na małą ilość danych i trudności związane z ich akwizycją zrezygnowano z eksperymentu sprawdzającego tą hipotezę, jest to jednak zagadnienie godne uwagi w przyszłych pracach związanych z klasyfikacją 3D.

6.2 Przewidywane kierunki dalszego rozwoju rozwiązań

Rozwiązania przedstawione w zaprezentowanych publikacjach pokazują, że w przypadku syntetycznych zbiorów danych zagadnienie ich klasyfikacji już obecnie wykonywane jest z zadowalającą skutecznością. Istotnym problemem w tym kontekście staje się problem klasyfikacji danych zebranych z rzeczywistych obiektów. Dane te cechują się znacznie mniejszą regularnością względem ręcznie przygotowanych i odszumionych modeli. Obecnie brakuje baz danych, zawierających dużą liczbę skategoryzowanych danych tego typu. Do momentu ich powstania, w celu poradzenia sobie z tym problemem najrozsądniejszym rozwiązaniem wydaje się zastosowaniem transfer learningu [?].

Jeżeli chodzi o samą architekturę nowych rozwiązań, prawdopodobne wydaje się pojawienie się większej liczby klasyfikatorów opartych na zastosowaniu powtórzonych struktur określonego typu. Jak pokazują wyniki osiągnięte przez klasyfikatory VRN Ensemble oraz PANORAMA-NN, zastosowanie tego typu konstrukcji pozwala na osiągnięcie bardzo wysokiej efektywności klasyfikacji niezależnie od sposobu reprezentacji danych wejściowych. Ze względu na swoją łatwą rozszerzalność prawdopodobnie częściej wykorzystywane będą struktury równoległe, pozwalające na maksymalne wykorzystanie informacji dostępnych w danych.

Podsumowując, klasyfikacja danych trójwymiarowych jest kluczowym zagadnieniem ze względu na dużą liczbę potencjalnych zastosowań, wymagających dużej efektywności semantycznej kategoryzacji konkretnych obiektów. Niezależnie od sposobu reprezentacji danych, obecnie występujące klasyfikatory wykorzystujące metody uczenia maszynowego bardzo dobrze radzą sobie z zadaniem klasyfikacji syntetycznych modeli. Wyzwaniami, które będą kluczowe w najbliższej przyszłości dziedziny jest opracowanie metod, pozwalających na wykorzystanie powyższych rozwiązań do klasyfikacji rzeczywistych obiektów.