

Politechnika Warszawska

W Y D Z I A Ł E L E K T R Y C Z N Y



Instytut Sterowania i Elektroniki Przemysłowej

Praca dyplomowa inżynierska

na kierunku Informatyka Stosowana

w specjalności Informatyka Stosowana

Analiza porównawcza neuronowych wizyjnych algorytmów percepcji głębi

Patryk Piotrowski

numer albumu 315564

promotor

dr inż. Witold Czajewski

Warszawa 2024

Analiza porównawcza neuronowych wizyjnych algorytmów percepji głębi

Streszczenie

Streszczenie pracy powinno w zwięzły sposób opisywać to czego dotyczy praca, co jest jej celem, jakie przyjęto założenia, co w ramach pracy zrobiono, co przebadano, jakie rozwiązania zaproponowano i jakie wyniki osiągnięto. Jeśli coś sprawiło jakiś problem, można to także ująć w streszczeniu i skomentować czy udało się ów problem rozwiązać i jak, czy też nie (nic w tym złego, jeśli czegoś się nie udało do końca zrobić).

Streszczenie nie powinno być przeładowane, powinno mieć około 200 słów i zawierać najważniejsze informacje na temat pracy i najważniejsze osiągnięcia, nie należy więc podawać szczegółowych wyników czy też opisywać struktury pracy - na to miejsce znajduje się w samej pracy.

Streszczenie zwykle pisze się na samym końcu pracy - z oczywistych względów.

Przykładowe streszczenie:

W niniejszej pracy inżynierskiej zaprezentowane zostały próby rozwiązania problemu klasyfikacji poziomu zabrudzenia chodników z wykorzystaniem głębokich sieci neuronowych. Wykorzystując dostępny zbiór zdjęć sklasyfikowanych do sześciu klas poziomu zanieczyszczeń, wygenerowano zrównoważone zbiory: sześcioklasowy oraz trzyklasowy. Następnie wybrano trzy sieci różnego rodzaju, aby sprawdzić ich efektywność w rozróżnianiu poszczególnych, często bardzo zbliżonych do siebie zdjęć. Sieć TResNet została wybrana spośród wiodących sieci w rankingach klasyfikacyjnych. Sieć ShuffleNet V2 została wybrana spośród dostępnych implementacji sieci w ramach popularnej biblioteki OpenMMLab. Sieć PMG została wybrana spośród sieci przygotowanych do rozwiązywania specyficznych problemów klasyfikacyjnych dla obrazów o niewielkich różnicach. Wymienione sieci wytrenowano na wygenerowanych zbiorach i osiągnięto dokładność rzędu 80% i 90%, odpowiednio dla problemu sześcioc- i trzyklasowego. Najlepszą siecią, w każdym przypadku, okazała się sieć PMG, a wyniki pozostałych sieci były zbliżone.

Słowa kluczowe: istotne, słowa, kluczowe

Comparative analysis of neural vision algorithms for depth perception

Abstract

Należy zadbać oto, by jakość tłumaczenia streszczenia była wyższa niż ta oferowana przez automaty takie jak powszechnie używany <https://translate.google.pl/> czy mniej znany, ale czasem lepszy: <https://www.deepl.com/translator> czy może też <https://www.translate.com/>. Wszystkie te strony są pomocne i często zgrabnie tłumaczą, ale potrafią też zrobić oczywiste błędy, zwłaszcza w przypadku zdań wielokrotnie złożonych. Warto też skorzystać z serwisu <https://www.grammarly.com>.

W miarę poprawny przykład:

This engineering thesis presents an attempt to solve the problem of sidewalk dirt level classification using deep neural networks. Using an available set of images classified into six classes of dirt levels, balanced sets of six classes and three classes were generated. Three networks of different types were then selected to test their effectiveness in discriminating between individual, often very similar, images. TResNet was selected among the leading networks in the classification rankings. The ShuffleNet V2 network was selected from available network implementations within the popular OpenMMLab library. The PMG network was selected from networks prepared to solve specific classification problems for images with small differences. The aforementioned networks were trained on the generated sets and an accuracy of 80% and 90% was achieved for the six-class and three-class problem, respectively. The PMG network proved to be the best network, in each case, and the results of the other networks were similar.

Keywords: keywords, that, are, indicative

Spis treści

| | |
|--|-----------|
| 1 Wstęp | 1 |
| 1.1 Cel i układ pracy | 3 |
| 2 Wprowadzenie do algorytmów percepacji głębi | 5 |
| 2.1 Paradygmaty uczenia | 5 |
| 2.1.1 Uczenie nadzorowane | 5 |
| 2.1.2 Uczenie nienadzorowane | 6 |
| 2.1.3 Uczenie częściowo nadzorowane | 7 |
| 2.2 Modele sieci neuronowych w algorytmach percepji głębi | 7 |
| 2.2.1 Konwolucyjne sieci neuronowe | 8 |
| 2.2.2 Transformatory | 9 |
| 3 Przegląd istniejących rozwiązań | 11 |
| 3.1 Algorytmy percepji głębi | 11 |
| 3.1.1 AdelaiDepth | 11 |
| 3.1.2 MetaPrompt-SD | 12 |
| 3.1.3 EVP | 13 |
| 3.1.4 ZoeDepth | 14 |
| 3.1.5 Depth Anything | 16 |
| 3.1.6 UniDepth | 17 |
| 3.1.7 Podsumowanie | 18 |
| 3.2 Zbiory danych | 20 |
| 3.2.1 KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) | 20 |
| 3.2.2 NYUv2 (NYU-Depth V2) | 21 |
| 3.2.3 DIODE (Dense Indoor and Outdoor Depth) | 21 |
| 3.2.4 SUN RGB-D | 21 |
| 3.2.5 Matterport3D | 21 |
| 3.2.6 DDAD (Dense Depth for Autonomous Driving) | 22 |
| 3.2.7 Podsumowanie | 22 |
| 4 Przedstawienie zastosowanych narzędzi | 23 |
| 4.1 Język programowania | 23 |

| | | |
|----------|--|-----------|
| 4.2 | Platforma obliczeniowa | 23 |
| 4.3 | Zestawy danych | 24 |
| 4.4 | Narzędzia do analizy i wizualizacji | 24 |
| 5 | Zbiór danych do analizy porównawczej | 25 |
| 6 | Metodologia - opis kryteriów oceny algorytmów | 27 |
| 7 | Analiza i wyniki | 29 |
| 8 | Podsumowanie i wnioski | 31 |
| | Bibliografia | 33 |
| | Wykaz skrótów i symboli | 37 |
| | Spis rysunków | 39 |
| | Spis tabel | 41 |
| | Spis załączników | 43 |

Rozdział 1

Wstęp

Sieci neuronowe jako narzędzie przetwarzania informacji są szeroko eksploatowane w celu rozwiązywania problemów niezliczonych sektorów już od wielu dekad¹. Ich obecność w tworzonym dziś oprogramowaniu stanowi pomoc dla pracowników branży m.in. medycznej, motoryzacyjnej, ekonomicznej, coraz częściej również rozrywkowej. Rozwój technologii powiązanych z zagadnieniem sztucznej inteligencji następuje niezmiennie w wysokim tempie. Wraz z rozwojem rzeczonej technologii zaczęto implementować neuronowe algorytmy wizji komputerowej umożliwiające przetwarzanie obrazów zarejestrowanych w postaci cyfrowej [22]. Do tej grupy należą neuronowe wizyjne algorytmy percepcji głębi. Znajdują one zastosowanie między innymi w zakresie autonomicznej mobilności, systemach rozszerzonej rzeczywistości czy robotyce. Lepsze zrozumienie głębi sceny widzianej jednym obiektywem pełni w tych obszarach ważną rolę. Umożliwia reagowanie na przeszkody w czasie rzeczywistym, analizę pod kątem dostępności powierzchni jak również wykonywanie przybliżonych pomiarów. W połączeniu z innymi technikami, takimi jak segmentacja obrazu [14] polegająca na podziale na charakterystyczne części związane z obiekttami widocznymi na obrazie pozwala budować zaawansowane systemy wizyjne.

Algorytmy percepcji głębi stanowią znaczne uproszczenie w dziedzinie pozyskiwania informacji o głębi dwuwymiarowego obrazu, głównie przez wzgląd na charakterystykę pozostałych znanych dotychczas metod, które zakładają posiadanie kosztownej elektroniki² oraz konieczność jej użycia w trakcie wykonywania fotografii podczas gdy zastosowanie algorytmów może mieć miejsce w dowolnym odstępie czasu następującym po utrwaleniu obrazu. Otworzyło to zatem możliwość rozpoznania głębi obrazów nie tylko wykonanych przy pomocy pojedynczego obiektywu ale również zarejestrowanych historycznie.

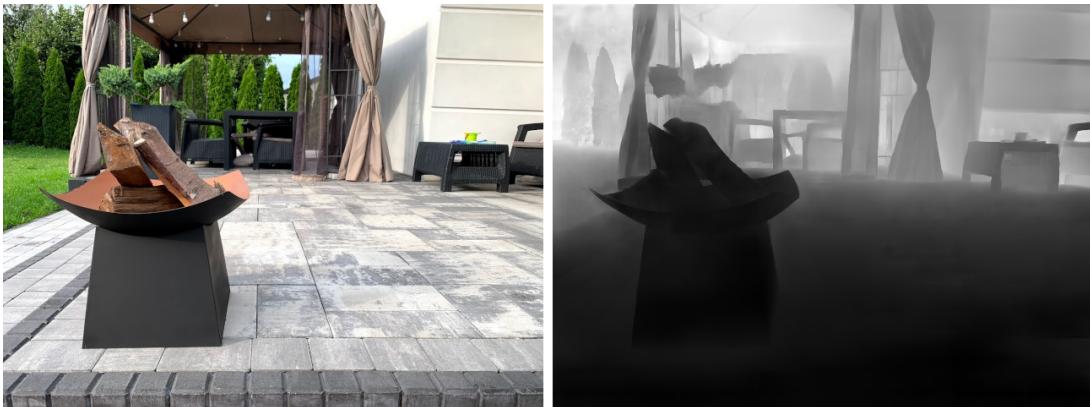
Zadanie wizyjnych algorytmów percepcji głębi opartych o sieci neuronowe polega na estymacji odległości każdego pojedynczego zarejestrowanego piksela względem urządzenia rejestrującego na podstawie pojedynczej fotografii wykonanej jednym obiektywem. W zależności od algorytmu wynikowe odległości mogą mieć charakter względny lub metryczny. Realizacja tego zadania polega

¹Początki sieci neuronowych sięgają lat czterdziestych XX wieku [27].

²Na przykład kamery 3D skorelowanej z systemem LIDAR. [7]

Rozdział 1. Wstęp

na przetworzeniu obrazu wejściowego przez warstwy sieci neuronowej odpowiedniej dla architektury danego algorytmu. Ostatecznym wynikiem realizacji tego zadania jest macierz zawierająca wartości odległości dla pojedynczych pikseli. Wizualną reprezentację takiej macierzy stanowi mapa głębi. Przykładową mapę głębi przedstawia rys. 1.



Rysunek 1. Fotografia i odpowiadająca jej mapa głębi. Źródło: własne

W celu sprawnego funkcjonowania sieci neuronowej należy pierwotnie wykonać jej uczenie. Uczenie to polega na wyznaczeniu wag i parametrów danej sieci poprzez wykonanie algorytmu na zbiorze danych składającym się ze zbioru uczącego i zbioru testowego. Wyniki działania algorytmu na elementach zbioru uczącego porównywane są z odpowiadającymi tym elementom danymi o głębi zmierzonymi odpowiednią aparaturą podczas przygotowywania zbioru danych. Na podstawie tych porównań w kolejnych iteracjach wykonania algorytmu wagi oraz parametry są dopasowywane w taki sposób, aby wyniki następnych wykonień były jak najdokładniejsze. Najczęściej stosowanymi zbiorami danych w domenie głębi obrazu są NYU-Depth V2 [3] zawierający 407024 obrazów uczących przedstawiających sceny wewnętrz budynków zarejestrowanych przy pomocy urządzenia Microsoft Kinect oraz KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) [10] zawierający 93 tysiące obrazów uczących przedstawiających sceny zewnętrzne zarejestrowanych przy pomocy urządzenia z systemem LIDAR.

Pierwszą implementację omawianego algorytmu w 2014 r. zaproponowali pracownicy naukowi Instytutu Nauk Matematycznych Uniwersytetu w Nowym Jorku - David Eigen, Christian Puhrsich oraz Rob Fergus [4]. Zaprojektowana wówczas przez wymienionych autorów architektura rozwiązania oparta została na dwóch współpracujących konwolucyjnych sieciach neuronowych [15]. W dniu dzisiejszym osiągające najlepsze wyniki algorytmy również stosują w swojej architekturze sieci konwolucyjne, chociaż w kwestii częstości implementacji nie ustępują im na tym polu także transformatory [25], które stanowią obecnie około połowę najczęściej używanych rozwiązań.

1.1 Cel i układ pracy

Aktualnie w otwartych źródłach istnieje wiele gotowych realizacji algorytmów percepji głębi zdywersyfikowanych pod kątem architektury, funkcjonalności, osiąganych wyników i przeznaczenia. Wobec powyższego, celem badawczym niniejszej pracy inżynierskiej jest ich analiza porównawcza.

Do realizacji nadrzednego celu pracy przyjęto następujące zadania badawcze:

- przegląd i ogólna charakterystyka dostępnych rozwiązań,
- wybór wiodących rozwiązań,
- weryfikacja metod na zbiorach, na których były uczone oraz na innych zbiorach,
- weryfikacja na własnych scenach,
- porównanie rozwiązań i rekomendacja przypadków użycia.

Rozdział 2

Wprowadzenie do algorytmów percepcji głębi

2.1 Paradygmaty uczenia

Podstawową metodą uczenia algorytmów percepcji głębi jest uczenie nadzorowane. W tejże metodzie do nauki estymacji mapy głębi wykorzystywana jest struktura scen z obrazów określanych jako Ground truth¹. Pozyskanie tych obrazów często bywa kosztowne i problematyczne, skąd wyniknęła potrzeba uczenia algorytmów przy użyciu zmniejszonej ilości danych rzeczywistych tudzież ich całkowitym braku². W ramach usystematyzowania wiedzy, następujący podrozdział skupi się zatem na zebraniu i sklasyfikowaniu paradygmatów uczenia algorytmów percepcji głębi.

2.1.1 Uczenie nadzorowane

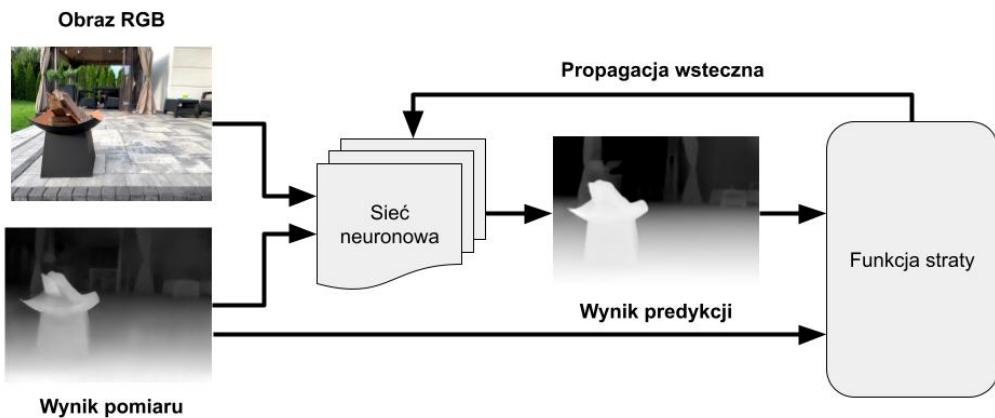
Ta najczęściej obecnie stosowana metoda uczenia sieci neuronowych zakłada posiadanie odpowiednio przygotowanych danych wejściowych oraz odpowiadających im danych wyjściowych. Wówczas celem nauki jest zminimalizowanie wartości odpowiednio sporzązonej funkcji straty, której argumentami są wartości zmierzane i estymowane. Wybór wspomnianej funkcji zależy od charakterystyki rozwiązywanego problemu. W przypadku percepcji głębi najczęściej stosowaną funkcją straty jest błąd średniokwadratowy (MSE od ang. mean square error):

$$MSE = \frac{1}{n} \sum_{t=1}^n (d_i^* - d_i)^2 \quad (1)$$

gdzie d_i^* to wartość predykcji a d_i to wartość zmierzona.

¹Dane rzeczywiste uzyskane za pomocą technologii rejestracji obrazów 3D.

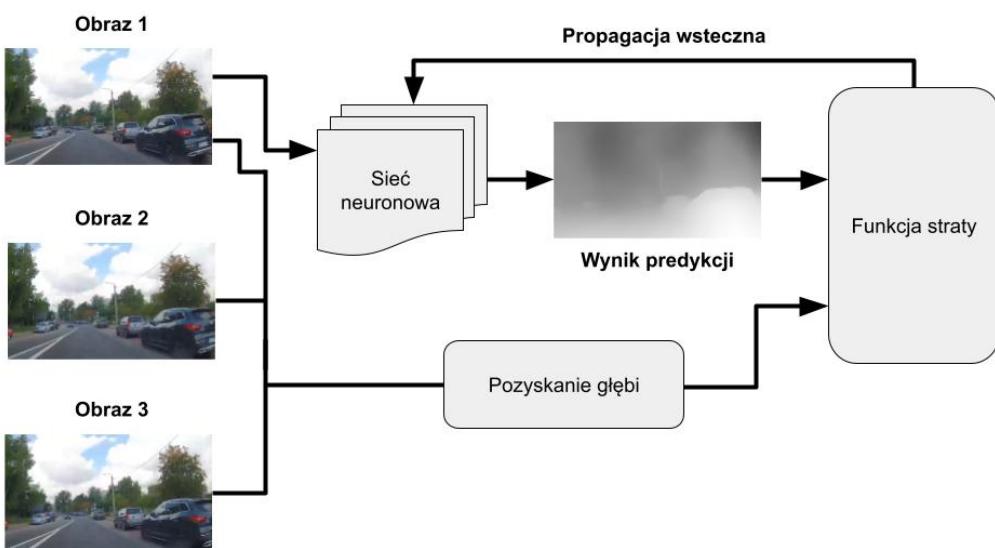
²Wówczas mówimy o rekonstrukcji mapy głębi.



Rysunek 2. Poglądowy model uczenia nadzorowanego. Wejścia stanowią obraz RGB oraz pomiary głębi a wynikiem jest predykcja mapy głębi.

2.1.2 Uczenie nienadzorowane

Z powodu potrzeby uniknięcia kosztownego procesu przygotowania danych na potrzeby uczenia nadzorowanego, rozwijana jest metoda uczenia nienadzorowanego. Algorytmy wykorzystujące tę metodę nauczane są zwykle przy pomocy zdecydowanie prostszych danych - par fotografii RGB lub nagrań wideo, czyli w uproszczeniu sekwencji fotografii RGB. Dane te przetwarzane są za pomocą funkcji, których zadaniem jest określenie głębi sceny przedstawionej na zdjęciu na podstawie zmian w perspektywie pomiędzy poszczególnymi kadrami. W ten sposób przygotowany zestaw wykorzystywany jest do nauki algorytmu podobnie jak w przypadku uczenia nadzorowanego.



Rysunek 3. Schemat przykładowego uczenia nienadzorowanego. Wejścia stanowią trzy kadry z nagrania RGB a wynikiem jest predykcja mapy głębi.

2.1.3 Uczenie częściowo nadzorowane

Sposobem łączącym dwa poprzednio przedstawione jest uczenie częściowo nadzorowane. Metoda ta wykorzystuje w procesie uczenia zarówno dane etykietowane jak i nieoznaczone. Głównymi zaletami stosowania tego sposobu są

- poprawa wydajności algorytmu - ze względu na wykorzystanie większej ilości danych,
- zmniejszenie kosztów pozyskania danych etykietowanych przy jednoczesnym zachowaniu zadowalających rezultatów,
- zwiększenie elastyczności modelu ze względu na brak uzależnienia od wyłącznie danych oznaczonych.

Poniższy schemat przedstawia ogólne podsumowanie paradygmatów uczenia algorytmów.



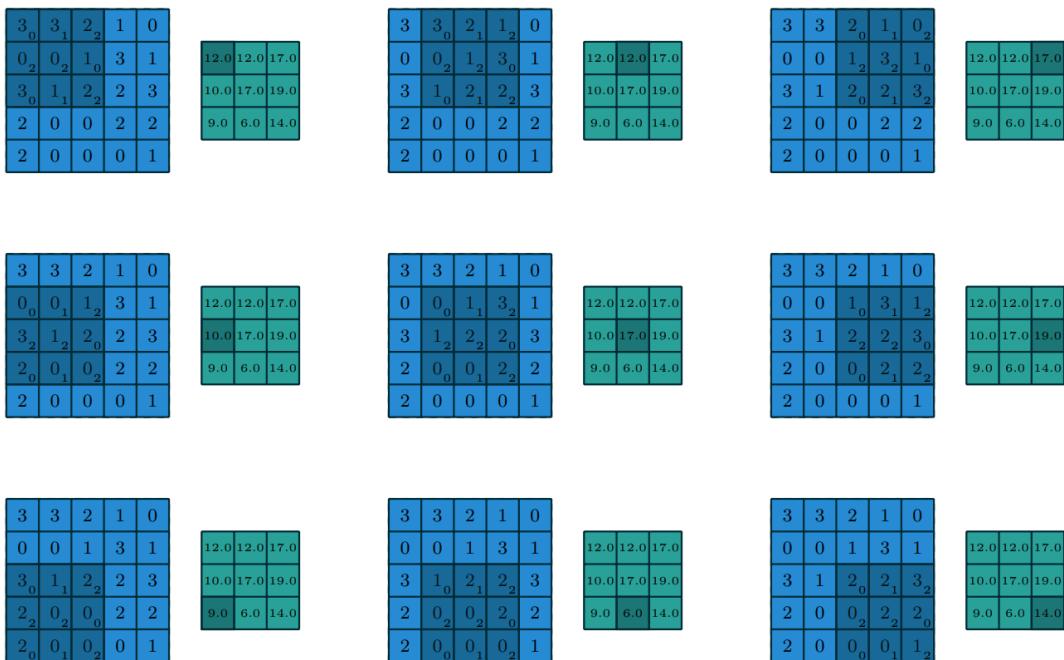
Rysunek 4. Schemat podsumowujący paradygmaty uczenia algorytmów.

2.2 Modele sieci neuronowych w algorytmach percepcji głębi

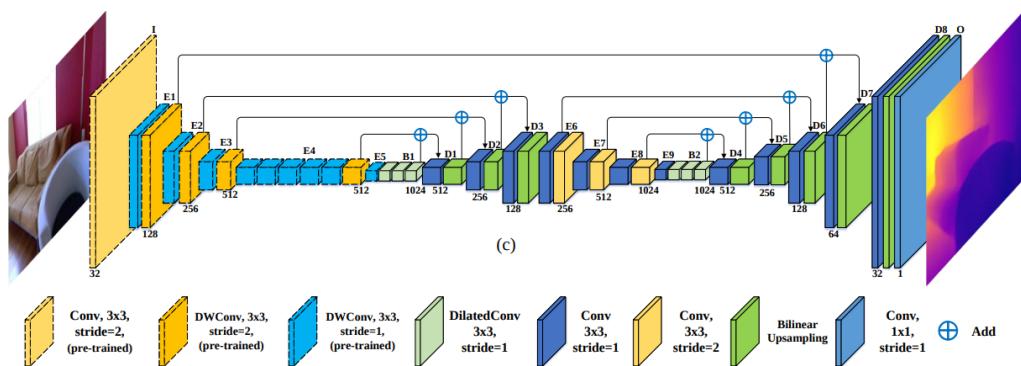
Ważnym etapem implementacji algorytmu percepcji głębi jest konstrukcja architektury sieci neuronowej modelu. Ma ona szczególny wpływ na wydajność i skuteczność wynikowego algorytmu. Ten podrozdział zawiera opis dwóch przeważnie wykorzystywanych w dziedzinie percepcji głębi modeli sieci neuronowych.

2.2.1 Konwolucyjne sieci neuronowe

Zaproponowane przez Kunihiko Fukushima w 1980 r. [9] konwolucyjne sieci neuronowe są niewątpliwym kamieniem milowym w komputerowym przetwarzaniu obrazów. Charakteryzuje je zdolność upraszczania obrazu do postaci znacznie łatwiejszej do przetworzenia przez komputer, bez poświęcenia jakości wnioskowania. Podstawowym elementem takich sieci jest warstwa splotowa, w której dochodzi do mnożenia matryc stanowiących dane wejściowe i jądro. Wynikiem mnożenia jest mapa wyodrębnionych cech wejściowego obrazu. Poprawnie przedstawia to poniższa grafika. W przypadku takiego modelu nauczanie sieci polega między innymi na ustanowieniu odpowiednich wag jądra.



Rysunek 5. Przykład działania warstwy konwolucyjnej z jądem o rozmiarze 3x3. Źródło: [8]



Rysunek 6. Przykład architektury sieci konwolucyjnej użytej w celu rozpoznania głębi obrazu. Źródło: [5]

2.2.2 Transformatory

Zaprezentowane w 2017 r. [25] transformatory wykorzystywane były pierwotnie w przetwarzaniu języka naturalnego. Dzięki asynchronicznej charakterystyce przetwarzania sekwencji wejściowej okazały się znacznie szybsze niż dotychczas znane rozwiązania³.

W kontekście wizyjnych algorytmów wykorzystywane są transformatory wizyjne zaproponowane w 2020 r. przez zespół Google Research w [6]. Jego schemat poglądowy przedstawia poniższa grafika.



Rysunek 7. Schemat modelu transformatora wizyjnego. Źródło: [6]

Wizyjny model transformatora nie generalizuje danych tak dobrze jak robi to sieć konwolucyjna, dlatego przy niewielkiej liczbie obrazów uczących nie jest najlepszym wyborem. Jednak przy wykorzystaniu znacznego rozmiaru zestawu obrazów uczących osiągana dokładność najczęściej przewyższa sieci konwolucyjne.

³Do ówczesnej chwili częściej używane były sieci rekurencyjne.

Rozdział 3

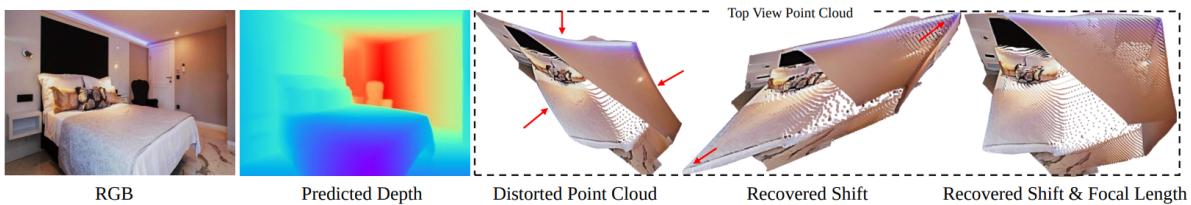
Przegląd istniejących rozwiązań

W tym rozdziale przybliżone zostanie spektrum dostępnych algorytmów percepcji głębi oraz zbiorów danych używanych do ich trenowania. Zestawienie to ogranicza się do rozwiązań z największą liczbą cytowań w opracowaniach i artykułach naukowych. Osiągają one jednocześnie rekordowe na dzień przygotowywania zestawienia rezultaty.

3.1 Algorytmy percepcji głębi

3.1.1 AdelaiDepth

Zaprojektowany w 2020 r. w [30] model przygotowany został w celu rekonstrukcji scen trójwymiarowych. Autorzy podzielili wówczas rozwiązanie na dwa etapy - predykcję głębi obrazu oraz predykcję jej przesunięcia i ogniskowej.



Rysunek 8. Przykładowy wynik działania algorytmu AdelaiDepth. Źródło: [30]

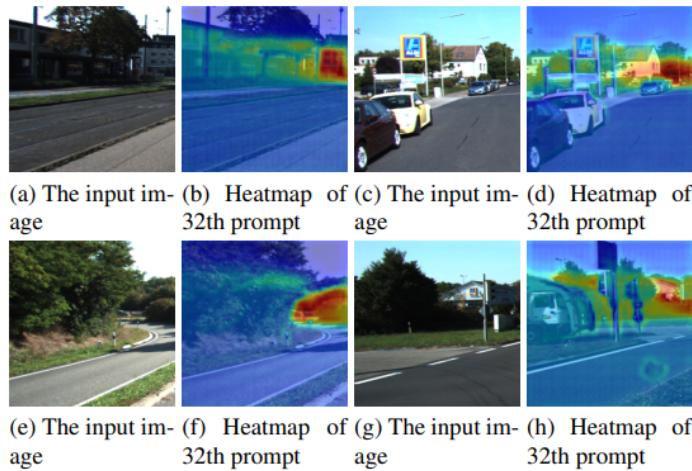
Architektura modelu predykcji głębi została zainspirowana rozwiązaniem przedstawionym w [28]. Jest to konwolucyjna sieć neuronowa ResNet [12] z dekoderem. W celu nauczenia sieci wykorzystane zostało sumarycznie 354 tysiące obrazów RGBD pochodzących z różnych dostępnych zestawów danych, zarejestrowanych jak i wytworzonych syntetycznie z obrazów RGB za pomocą oprogramowania. Całkowity zestaw danych treningowych zawiera w sobie zatem wysokiej jakości obrazy z systemu LIDAR ale też niskiej jakości nagrania. Poniższa tabela obrazuje wyniki osiągane przez ten algorytm na tle wybranych przez autorów podobnych rozwiązań.

| Method | Backbone | OASIS YT3D WHDR \downarrow | | NYU | | KITTI | | DIODE | | ScanNet | | ETH3D | | Sintel | | Rank |
|-------------------|------------|---------------------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|------|
| | | AbsRel \downarrow | $\delta_1 \uparrow$ | AbsRel \downarrow | $\delta_1 \uparrow$ | AbsRel \downarrow | $\delta_1 \uparrow$ | AbsRel \downarrow | $\delta_1 \uparrow$ | AbsRel \downarrow | $\delta_1 \uparrow$ | AbsRel \downarrow | $\delta_1 \uparrow$ | AbsRel \downarrow | $\delta_1 \uparrow$ | |
| OASIS [8] | ResNet50 | 32.7 | 27.0 | 21.9 | 66.8 | 31.7 | 43.7 | 48.4 | 53.4 | 19.8 | 69.7 | 29.2 | 59.5 | 60.2 | 42.9 | 6.7 |
| MegaDepth [26] | Hourglass | 33.5 | 26.7 | 19.4 | 71.4 | 20.1 | 66.3 | 39.1 | 61.5 | 19.0 | 71.2 | 26.0 | 64.3 | 39.8 | 52.7 | 6.7 |
| Xian [47] | ResNet50 | 31.6 | 23.0 | 16.6 | 77.2 | 27.0 | 52.9 | 42.5 | 61.8 | 17.4 | 75.9 | 27.3 | 63.0 | 52.6 | 50.9 | 6.7 |
| WSVD [40] | ResNet50 | 34.8 | 24.8 | 22.6 | 65.0 | 24.4 | 60.2 | 35.8 | 63.8 | 18.9 | 71.4 | 26.1 | 61.9 | 35.9 | 54.5 | 6.6 |
| Chen [7] | ResNet50 | 33.6 | 20.9 | 16.6 | 77.3 | 32.7 | 51.2 | 37.9 | 66.0 | 16.5 | 76.7 | 23.7 | 67.2 | 38.4 | 57.4 | 5.6 |
| DiverseDepth [51] | ResNeXt50 | 30.9 | 21.2 | 11.7 | 87.5 | 19.0 | 70.4 | 37.6 | 63.1 | 10.8 | 88.2 | 22.8 | 69.4 | 38.6 | 58.7 | 4.4 |
| MiDaS [32] | ResNeXt101 | 29.5 | 19.9 | 11.1 | 88.5 | 23.6 | 63.0 | 33.2 | 71.5 | 11.1 | 88.6 | 18.4 | 75.2 | 40.5 | 60.6 | 3.5 |
| Ours | ResNet50 | 30.2 | 19.5 | 9.1 | 91.4 | 14.3 | 80.0 | 28.7 | 75.1 | 9.6 | 90.8 | 18.4 | 75.8 | 34.4 | 62.4 | 1.9 |
| Ours | ResNeXt101 | 28.3 | 19.2 | 9.0 | 91.6 | 14.9 | 78.4 | 27.1 | 76.6 | 9.5 | 91.2 | 17.1 | 77.7 | 31.9 | 65.9 | 1.1 |

Rysunek 9. Porównanie osiąganych wyników przeprowadzone na ośmiu zestawach danych nieuczestniczących w procesie uczenia. Źródło: [30]

3.1.2 MetaPrompt-SD

Głównym założeniem autorów artykułu [26] było wykorzystanie modeli dyfuzji w zadaniach dotyczących percepcji wizyjnej. Wynikowy algorytm służy do estymacji głębi, segmentacji semantycznej i estymacji pozy.



Rysunek 10. Przykładowe wyniki działania modułu estymacji głębi MetaPrompt-SD. Źródło: [26]

Podstawą architektury jest koder VQVAE [16] kodujący obraz wejściowy do przestrzeni ukrytej¹ oraz sieć UNet [21], która wielokrotnie wykorzystana ma za zadanie usunięcie szumów i poprawę jakości cech obrazu. Sieć UNet wspomaga komponent Meta prompts zawierający dodatkowe informacje pomagające w procesie poprawy cech. Po ukończeniu przetwarzania cech obrazu są one wysyłane do dekodera, który przetwarzając je podaje obraz wyjściowy.

¹Jest to przestrzeń przechowująca kluczowe cechy obrazu przy jednoczesnej redukcji jego rozdzielczości.



Rysunek 11. Schemat architektury algorytmu MetaPrompt-SD. Źródło: [26]

Dla percepkcji gębi jako zestawy uczące wykorzystane zostały obrazy scen wewnętrznych i zewnętrznych pochodzące ze zbiorów NYU depth V2 oraz KITTI. W sumie stanowi to prawie 95 tysięcy par map gębi z odpowiadającymi im obrazami RGB pochodzącymi z urządzenia LIDAR firmy Velodyne oraz Microsoft Kinect.

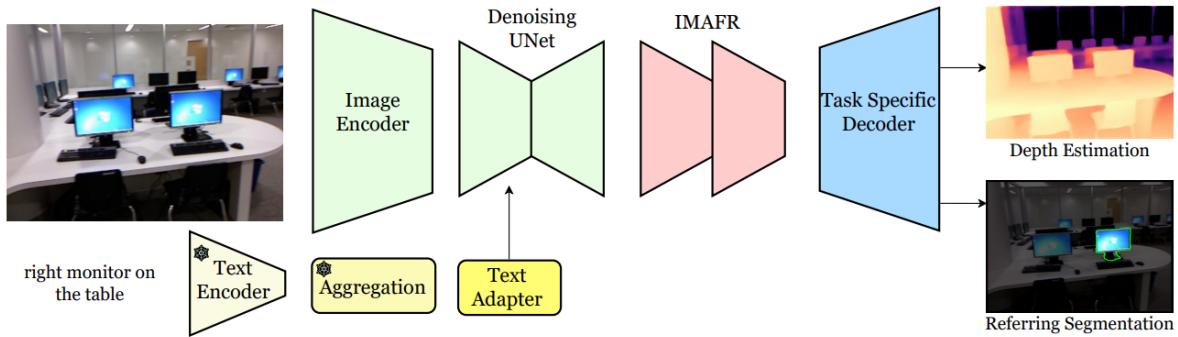
| Method | NYU depth V2 | | | | | KITTI Eigen split | | | | |
|----------------------------|--------------|--------------|---------------------|---------------------|---------------------|-------------------|--------------|---------------------|---------------------|---------------------|
| | RMSE↓ | REL↓ | $\delta_1 \uparrow$ | $\delta_2 \uparrow$ | $\delta_3 \uparrow$ | RMSE↓ | REL↓ | $\delta_1 \uparrow$ | $\delta_2 \uparrow$ | $\delta_3 \uparrow$ |
| <i>non-diffusion-based</i> | | | | | | | | | | |
| GEDepth [46] | - | - | - | - | - | 2.044 | 0.048 | 0.976 | 0.997 | 0.999 |
| MAMo [47] | - | - | - | - | - | 1.984 | 0.049 | 0.977 | 0.998 | 1.000 |
| DepthFormer [22] | 0.339 | 0.096 | 0.921 | 0.989 | 0.998 | 2.143 | 0.052 | 0.975 | 0.997 | 0.999 |
| PixelFormer [1] | 0.322 | 0.090 | 0.929 | 0.991 | 0.998 | 2.081 | 0.051 | 0.976 | 0.997 | 0.999 |
| SwinV2-MIM [44] | 0.287 | 0.083 | 0.949 | 0.994 | 0.999 | 1.966 | 0.050 | 0.977 | 0.998 | 1.000 |
| ZoeDepth [2] | 0.270 | 0.075 | 0.955 | 0.995 | 0.999 | - | 0.057 | - | - | - |
| MeSa [18] | 0.238 | 0.066 | 0.964 | 0.995 | 0.999 | - | - | - | - | - |
| <i>diffusion-based</i> | | | | | | | | | | |
| DDP [17] | 0.329 | 0.094 | 0.921 | 0.990 | 0.998 | 2.072 | 0.050 | 0.975 | 0.997 | 0.999 |
| DepthGen [33] | 0.314 | 0.074 | 0.946 | 0.987 | 0.996 | 2.985 | 0.064 | 0.953 | 0.991 | 0.998 |
| VPD [50] | 0.254 | 0.069 | 0.964 | 0.995 | 0.999 | - | - | - | - | - |
| TADP [19] | 0.225 | 0.062 | 0.976 | 0.997 | 0.999 | - | - | - | - | - |
| Ours | 0.223 | 0.061 | 0.976 | 0.997 | 0.999 | 1.929 | 0.047 | 0.982 | 0.998 | 1.000 |

Rysunek 12. Porównanie osiąganych wyników przeprowadzone na dwóch zestawach danych. Źródło: [26]

3.1.3 EVP

Metoda o nazwie EVP [13] (Enhanced Visual Perception) jest rozbudowaniem metody VPD [31] (Visual Perception with a pre-trained Diffusion model), zadaniem której było podobnie do MetaPrompt-SD wykorzystanie modeli dyfuzji w percepkcji wizyjnej. W stosunku do pierwotnego w modelu EVP dodano moduł o nazwie *IMAFR* (Inverse MultiAttentive Feature Refinement) wspomagający zdolności wyodrębniania cech. Między innymi dzięki tej zmianie autorom udało się uzyskać lepszy wynik w porównaniu do metody VPD na zestawie NYU Depth v2².

²Model EVP uzyskał w tym porównaniu o 11,8% mniejszą wartość błędu średniokwadratowego.



Rysunek 13. Schemat architektury algorytmu EVP. Źródło: [13]

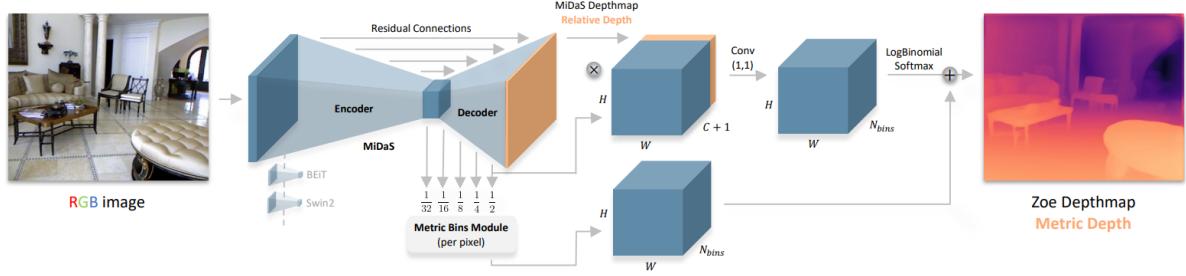
Głównymi elementami architektury rozwiązania są koder obrazu wejściowego, sieć wyodrębniająca cechy UNet oraz wyspecjalizowany w kierunku odpowiedniego zadania dekoder. Metoda jest bowiem w stanie wykonać predykcję głębi jak również dokonać segmentacji semantycznej. Rozwiązanie to zostało wytrenowane przy użyciu zestawów NYU Depth v2 - konkretnie na podzbiorze 50 tysięcy obrazów oraz KITTI - na podzbiorze 26 tysięcy obrazów. Autorzy dokonali porównania rezultatów osiąganych na zbiorze testowym zestawu KITTI - przedstawia je poniższa tabela.

| Method | REL \downarrow | SqREL \downarrow | RMSE \downarrow | RMSE log \downarrow | $\delta_1 \uparrow$ | $\delta_2 \uparrow$ | $\delta_3 \uparrow$ |
|----------------------|------------------|--------------------|-------------------|-----------------------|---------------------|---------------------|---------------------|
| BTS [25] | 0.061 | 0.261 | 2.834 | 0.099 | 0.954 | 0.992 | 0.998 |
| AdaBins [3] | 0.058 | 0.190 | 2.360 | 0.088 | 0.964 | 0.995 | <u>0.999</u> |
| ZoeDepth [5] | 0.057 | 0.194 | 2.290 | 0.091 | 0.967 | 0.995 | <u>0.999</u> |
| NeWCRFs [61] | 0.052 | 0.155 | 2.129 | 0.079 | 0.974 | <u>0.997</u> | <u>0.999</u> |
| iDisc [40] | <u>0.050</u> | 0.148 | 2.072 | 0.076 | 0.975 | <u>0.997</u> | <u>0.999</u> |
| NDDepth [49] | <u>0.050</u> | 0.141 | 2.025 | <u>0.075</u> | <u>0.978</u> | 0.998 | <u>0.999</u> |
| SwinV2-L 1K-MIM [55] | <u>0.050</u> | <u>0.139</u> | 1.966 | <u>0.075</u> | 0.977 | 0.998 | 1.000 |
| GEDepth [57] | 0.048 | 0.142 | 2.044 | 0.076 | 0.976 | <u>0.997</u> | 0.999 |
| EVP | 0.048 | 0.136 | <u>2.015</u> | 0.073 | 0.980 | 0.998 | 1.000 |

Rysunek 14. Porównanie osiąganych wyników przeprowadzone na zbiorze KITTI. Źródło: [13]

3.1.4 ZoeDepth

Opracowane rozwiązanie ZoeDepth [1] skupia się na zachowaniu wydajności przy jednoczesnym użyciu metrycznej skali w wyrażaniu wynikowych predykcji głębi. Proponowany model wykorzystuje 12 różnych zbiorów danych treningowych zawierających głębię relatywną i dwóch zestawów zawierających głębię metryczną, co pozwala osiągnąć założony cel.



Rysunek 15. Schemat architektury algorytmu ZoeDepth. Źródło: [1]

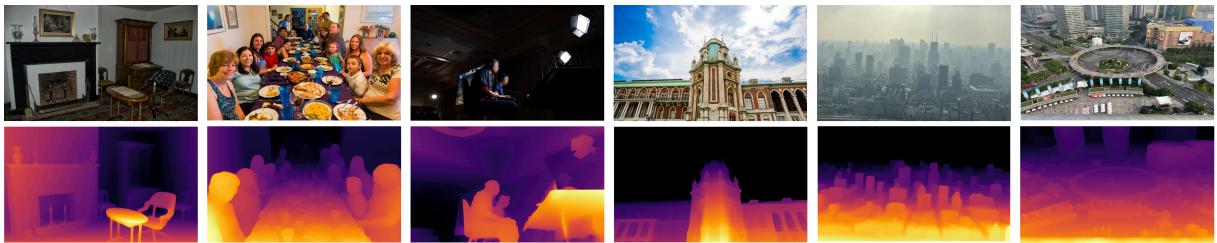
Architektura algorytmu ZoeDepth bazuje na rozwiązaniu o nazwie MiDaS [20]. Obraz wejściowy jest w pierwszej kolejności przetworzony przez ten właśnie algorytm. Wynik - głębia relatywna - jest dostarczany do modułu metrycznego, którego wynik stanowi z kolei wartość głębi metrycznej dla każdego pojedynczego piksela. Oba te wyniki kierowane są do sieci konwolucyjnej i w ten sposób uzyskiwany jest wynik ostateczny - głębia metryczna. Istnieje pięć gotowych przetrenowanych modeli, nazwanych według szablonu ZoeD-[zestaw pierwszy]-[zestaw drugi], gdzie zestawem pierwszym jest zestaw treningowy obrazów z głębią relatywną a zestaw drugi zawiera obrazy z głębią metryczną. Litera "X" z miejscu zestawu pierwszego oznacza, że do celu uczenia modelu wykorzystany został jedynie zestaw drugi.

| Method | $\delta_1 \uparrow$ | $\delta_2 \uparrow$ | $\delta_3 \uparrow$ | $REL \downarrow$ | $RMSE \downarrow$ | $\log_{10} \downarrow$ |
|--------------------------|---------------------|---------------------|---------------------|------------------|-------------------|------------------------|
| Eigen <i>et al.</i> [9] | 0.769 | 0.950 | 0.988 | 0.158 | 0.641 | – |
| Laina <i>et al.</i> [19] | 0.811 | 0.953 | 0.988 | 0.127 | 0.573 | 0.055 |
| Hao <i>et al.</i> [13] | 0.841 | 0.966 | 0.991 | 0.127 | 0.555 | 0.053 |
| DORN [11] | 0.828 | 0.965 | 0.992 | 0.115 | 0.509 | 0.051 |
| SharpNet [31] | 0.836 | 0.966 | 0.993 | 0.139 | 0.502 | 0.047 |
| Hu <i>et al.</i> [14] | 0.866 | 0.975 | 0.993 | 0.115 | 0.530 | 0.050 |
| Lee <i>et al.</i> [22] | 0.837 | 0.971 | 0.994 | 0.131 | 0.538 | – |
| Chen <i>et al.</i> [8] | 0.878 | 0.977 | 0.994 | 0.111 | 0.514 | 0.048 |
| BTS [20] | 0.885 | 0.978 | 0.994 | 0.110 | 0.392 | 0.047 |
| Yin <i>et al.</i> [48] | 0.875 | 0.976 | 0.994 | 0.108 | 0.416 | 0.048 |
| AdaBins [5] | 0.903 | 0.984 | 0.997 | 0.103 | 0.364 | 0.044 |
| LocalBins [6] | 0.907 | 0.987 | 0.998 | 0.099 | 0.357 | 0.042 |
| Jun <i>et al.</i> [16] | 0.913 | 0.987 | 0.998 | 0.098 | 0.355 | 0.042 |
| NeWCRFs [50] | 0.922 | 0.992 | 0.998 | 0.095 | 0.334 | 0.041 |
| ZoeD-X-N | 0.946 | 0.994 | 0.999 | 0.082 | 0.294 | 0.035 |
| ZoeD-M12-N | 0.955 | 0.995 | 0.999 | 0.075 | 0.270 | 0.032 |
| ZoeD-M12-NK | <u>0.953</u> | 0.995 | 0.999 | <u>0.077</u> | <u>0.277</u> | <u>0.033</u> |

Rysunek 16. Porównanie osiąganych wyników przeprowadzone na zbiorze NYU-Depth v2. Źródło: [1]

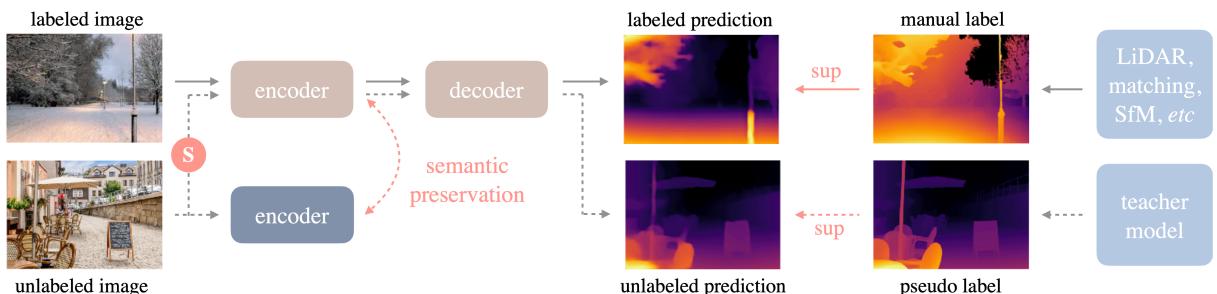
3.1.5 Depth Anything

Niezwykłe interesującą strategię rozwiązania przyjęli autorzy Depth Anything [29]. Położyli oni bowiem nacisk na niezwykle duży zestaw danych uczących składający się nie tylko z danych oznaczonych (1,5 miliona obrazów) ale również z danych nieoznaczonych (aż 62 miliony obrazów). W ten sposób otrzymano model charakteryzujący się bardzo dużą zdolnością generalizacji.



Rysunek 17. Przykładowe wyniki działania modelu Depth Anything. Źródło: [29]

W architekturze tego rozwiązania zastosowano koder wyodrębniający cechy obrazu wejściowego przygotowany na podstawie DINOv2 [17] oraz dekoder DPT do regresji głębi. W pierwszej kolejności model "nauczyciela" uczony jest na zestawie danych oznaczonych. Następnie model ten wykorzystywany jest w celu oznaczenia zbioru danych nieoznaczonych, który razem z zestawem danych oznaczonych weźmie udział w procesie uczenia modelu "ucznia".



Rysunek 18. Schemat architektury algorytmu Depth Anything. Źródło: [29]

| Dataset | Indoor | Outdoor | Label | # Images |
|-----------------------|--------|---------|--------|----------|
| Labeled Datasets | | | | |
| BlendedMVS [76] | ✓ | ✓ | Stereo | 115K |
| DIML [13] | ✓ | ✓ | Stereo | 927K |
| HRWSI [67] | ✓ | ✓ | Stereo | 20K |
| IRS [61] | ✓ | | Stereo | 103K |
| MegaDepth [33] | | ✓ | SfM | 128K |
| TartanAir [62] | ✓ | ✓ | Stereo | 306K |
| Unlabeled Datasets | | | | |
| BDD100K [81] | | ✓ | None | 8.2M |
| Google Landmarks [64] | | ✓ | None | 4.1M |
| ImageNet-21K [49] | ✓ | ✓ | None | 13.1M |
| LSUN [80] | ✓ | | None | 9.8M |
| Objects365 [52] | ✓ | ✓ | None | 1.7M |
| Open Images V7 [30] | ✓ | ✓ | None | 7.8M |
| Places365 [87] | ✓ | ✓ | None | 6.5M |
| SA-1B [27] | ✓ | ✓ | None | 11.1M |

Rysunek 19. Zbiór zestawów danych uczących Depth Anything. Źródło: [29]

Osiągane wyniki w porównaniach przeprowadzonych na zbiorach danych testowych w sposób zdecydowany udowadniają tezę autorów dotyczącą zasadności skalowania zestawów uczących przy pomocy danych nieoznaczonych.

| Method | Higher is better ↑ | | | Lower is better ↓ | | | Method | Higher is better ↑ | | | Lower is better ↓ | | |
|---------------|--------------------|--------------|--------------|-------------------|--------------|--------------|---------------|--------------------|--------------|--------------|-------------------|--------------|--------------|
| | δ_1 | δ_2 | δ_3 | AbsRel | RMSE | log10 | | δ_1 | δ_2 | δ_3 | AbsRel | RMSE | RMSE log |
| AdaBins [3] | 0.903 | 0.984 | 0.997 | 0.103 | 0.364 | 0.044 | AdaBins [3] | 0.964 | 0.995 | 0.999 | 0.058 | 2.360 | 0.088 |
| DPT [46] | 0.904 | 0.988 | 0.998 | 0.110 | 0.357 | 0.045 | DPT [46] | 0.959 | 0.995 | 0.999 | 0.062 | 2.573 | 0.092 |
| P3Depth [43] | 0.898 | 0.981 | 0.996 | 0.104 | 0.356 | 0.043 | P3Depth [43] | 0.953 | 0.993 | 0.998 | 0.071 | 2.842 | 0.103 |
| SwinV2-L [39] | 0.949 | 0.994 | 0.999 | 0.083 | 0.287 | 0.035 | NWCRFs [82] | 0.974 | 0.997 | 0.999 | 0.052 | 2.129 | 0.079 |
| AiT [41] | 0.954 | 0.994 | 0.999 | 0.076 | 0.275 | 0.033 | SwinV2-L [39] | 0.977 | 0.998 | 1.000 | 0.050 | 1.966 | 0.075 |
| VPD [86] | 0.964 | 0.995 | 0.999 | 0.069 | 0.254 | 0.030 | NDDepth [53] | 0.978 | 0.998 | 0.999 | 0.050 | 2.025 | 0.075 |
| ZoeDepth* [4] | 0.951 | 0.994 | 0.999 | 0.077 | 0.282 | 0.033 | GEDepth [75] | 0.976 | 0.997 | 0.999 | 0.048 | 2.044 | 0.076 |
| Ours | 0.984 | 0.998 | 1.000 | 0.056 | 0.206 | 0.024 | ZoeDepth* [4] | 0.971 | 0.996 | 0.999 | 0.054 | 2.281 | 0.082 |
| | | | | | | | Ours | 0.982 | 0.998 | 1.000 | 0.046 | 1.896 | 0.069 |

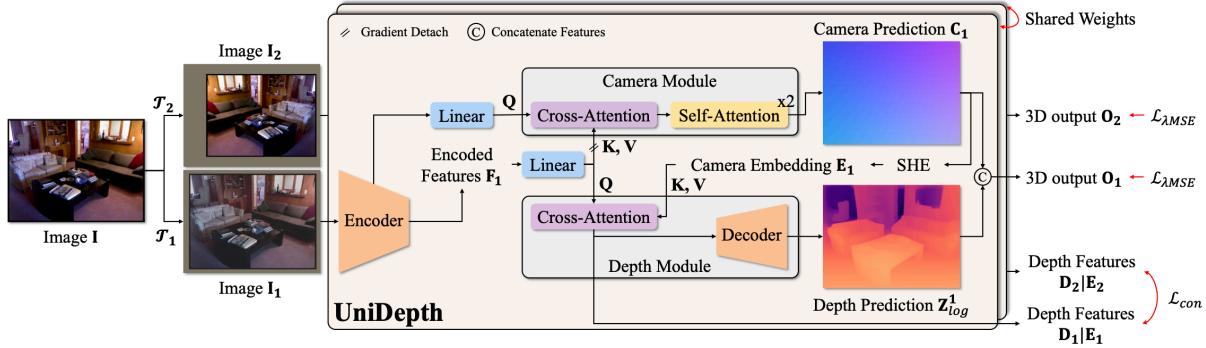
Rysunek 20. Porównanie rezultatów Depth Anything dokonane na podstawie zbioru NYUv2 (po lewej) i KITTI (po prawej). Źródło: [29]

3.1.6 UniDepth

Model o nazwie UniDepth [19] został zaproponowany przez jego autorów naprzeciw ich tezie o niskim stopniu generalizacji konkurencyjnych modeli. W swojej publikacji twierdzą, że ówcześnie najlepsze pod względem osiąganych wyników modele do predykcji głębi weryfikowane są na zbiorach podobnych do uczących oraz często na tyle niewielkich, że wyniki osiągane na pojedynczych obrazach odbiegających od domeny zbiorów uczących są mniej zadowalające. Architektura sieci tego modelu składa się z trzech głównych modułów - kodera, kamery i głębi. Koder przetwarza obraz wejściowy do postaci cech, które następnie przesyłane są do kolejnych dwóch modułów. Może być zarówno oparty o sieć konwolucyjną jak i o transformator wizyjny ViT co ma pozwolić na elastyczne dopasowanie do

Rozdział 3. Przegląd istniejących rozwiązań

potrzeb użytkownika. Moduł kamery odpowiada za generowanie reprezentacji, która jest następnie wykorzystywana do warunkowania cech gębi. Moduł gębi przyjmuje cechy z kodera i warunkuje je na podstawie informacji z modułu kamery wykorzystując przy tym warstwę uwagi krzyżowej. Połączenie wyniku modułu kamery i modułu gębi to wynik działania całego algorytmu.



Rysunek 21. Schemat architektury algorytmu UniDepth. Źródło: [19]

Do nauczenia modelu UniDepth wykorzystano 9 zestawów uczących składających się łącznie na 3 miliony obrazów, co umożliwiło nauczenie modelu różnorodnych scen z różnych punktów widzenia i różnymi warunkami oświetleniowymi. Do weryfikacji rezultatów wykorzystano natomiast 10 zestawów danych niepokrywających się z zestawami uczącymi. W porównaniu rezultatów wykorzystano modele zawierające koder z transformatorem i z siecią konwolucyjną, odpowiednio UniDepth-V i UniDepth-C.

| Method | NuScenes | | | | DDAD | | | | ETH3D | | | | Diode (Indoor) | | | | SUN-RGBD | | | | VOID | | | | IBims-I | | | | HAMMER | | | | | | | | | | | | | | | | | | | | |
|-------------------------|------------|-------------------|--------------|-------------|-------------------|--------------|-------------|-------------------|--------------|------------|-------------------|--------------|----------------|-------------------|--------------|------------|-------------------|--------------|-------------|-------------------|--------------|-------------|-------------------|--------------|----------------------------|-------------------|--------------|-------------|-------------------|--------------|-------------|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------|-------------|------|-------------|-------------|-------------|
| | δ_1 | SLog \downarrow | F \uparrow | δ_1 | SLog \downarrow | F \uparrow | δ_1 | SLog \downarrow | F \uparrow | δ_1 | SLog \downarrow | F \uparrow | δ_1 | SLog \downarrow | F \uparrow | δ_1 | SLog \downarrow | F \uparrow | δ_1 | SLog \downarrow | F \uparrow | δ_1 | SLog \downarrow | F \uparrow | δ_1 | SLog \downarrow | F \uparrow | δ_1 | SLog \downarrow | F \uparrow | | | | | | | | | | | | | | | | | | | |
| BTS [28] | 33.7 | 68.0 | 37.5 | 43.0 | 40.8 | 40.5 | 26.8 | 29.9 | 27.4 | 19.2 | 22.8 | 31.6 | 76.1 | 14.6 | 64.8 | 47.4 | 25.8 | 64.5 | 53.1 | 17.5 | 57.2 | 3.89 | 20.9 | 22.8 | AdaBins [3] | 33.3 | 61.4 | 35.2 | 37.7 | 44.4 | 35.6 | 24.3 | 28.3 | 25.2 | 17.4 | 21.6 | 28.7 | 77.7 | 13.9 | 65.4 | 50.5 | 23.8 | 65.0 | 55.0 | 15.6 | 57.8 | 7.21 | 21.5 | 27.7 |
| NeWCRF [61] | 44.2 | 49.4 | 42.2 | 45.6 | 34.9 | 41.6 | 35.7 | 26.1 | 32.3 | 20.1 | 18.5 | 35.3 | 75.3 | 11.9 | 61.6 | 53.1 | 22.3 | 67.9 | 53.6 | 14.7 | 59.2 | 1.43 | 14.9 | 20.8 | iDisc [44] | 39.4 | 37.1 | 34.5 | 28.4 | 32.2 | 25.8 | 35.6 | 27.5 | 31.4 | 23.8 | 15.8 | 33.4 | 83.7 | 12.4 | 71.0 | 55.3 | 20.3 | 68.6 | 48.9 | 13.2 | 55.4 | 2.58 | 14.0 | 32.6 |
| ZouDep [4] | 28.3 | 31.5 | 26.0 | 27.2 | 31.7 | 21.1 | 35.0 | 17.6 | 26.4 | 36.9 | 12.8 | 40.5 | 86.7 | 9.58 | 75.6 | 63.4 | 15.9 | 72.4 | 58.0 | 10.9 | 59.6 | 0.72 | 9.78 | 21.0 | Metric3D ³ [59] | 72.3 | 29.0 | 53.9 | — | — | 45.6 | 18.9 | 35.9 | 39.2 | 11.1 | 42.1 | 15.4 | 13.4 | 14.4 | 65.9 | 16.2 | 70.4 | 7.9 | 10.1 | 68.5 | 3.40 | 12.1 | 29.0 | |
| UniDepth-C | 83.3 | 22.9 | 62.3 | S3.2 | 21.4 | 59.3 | 49.8 | 13.2 | 33.7 | 60.2 | 9.03 | 50.0 | 94.8 | 8.10 | 81.4 | 86.6 | 12.8 | 85.1 | 79.7 | 8.92 | 66.7 | 20.2 | 8.78 | 57.1 | UniDepth-V | 86.2 | 21.7 | 67.4 | 86.4 | 20.3 | 61.8 | 32.6 | 11.6 | 24.3 | 77.1 | 6.38 | 59.4 | 96.6 | 7.05 | 81.9 | 89.4 | 10.9 | 85.7 | 23.9 | 7.22 | 37.1 | 13.3 | 7.41 | 55.9 |
| UniDepth-C ³ | 83.3 | 22.9 | 60.9 | 83.1 | 21.4 | 57.3 | 22.9 | 13.1 | 25.4 | 60.4 | 9.01 | 49.9 | 92.3 | 8.27 | 75.2 | 86.5 | 12.8 | 85.0 | 79.4 | 8.88 | 64.2 | 12.7 | 9.30 | 54.8 | UniDepth-V ³ | 86.2 | 21.7 | 63.0 | 86.4 | 20.3 | 60.4 | 17.4 | 11.4 | 21.4 | 77.4 | 6.36 | 58.6 | 94.8 | 7.17 | 75.9 | 90.2 | 10.9 | 86.2 | 17.5 | 7.20 | 36.5 | 2.56 | 83.5 | 53.8 |

Rysunek 22. Porównanie rezultatów UniDepth dokonane na zbiorach danych niewidzianych podczas uczenia.
Źródło: [19]

3.1.7 Podsumowanie

Przedstawione algorytmy można skategoryzować ze względu na rodzaj architektury, charakterystykę zestawów uczących i charakterystykę wyników. Poniższa tabela zawiera krótkie podsumowanie.

3.1. Algorytmy percepacji głębi

Tabela 1. Podsumowanie przedstawionych modeli percepji głębi.

| Nazwa | Architektura | Sposób uczenia | Zestawy uczące | Zestawy do oceny |
|---------------|---|-------------------------------------|--|--|
| AdelaiDepth | konwolucyjna sieć neuronowa | nadzorowane | <ul style="list-style-type: none"> • Taskonomy, • 3D Ken Burns, • DIML, • Holopix50K, • HRWSI. | <ul style="list-style-type: none"> • NYU depth V2, • KITTI, • ScanNet, • DIODE, • ETH3D, • Sintel, • OASIS, • YouTube3D, • RedWeb, • iBims-1. |
| MetaPrompt-SD | konwolucyjna sieć neuronowa | nadzorowane | <ul style="list-style-type: none"> • NYU depth V2, • KITTI. | Zestawy tożsame z uczącymi. |
| EVP | konwolucyjna sieć neuronowa | nadzorowane | <ul style="list-style-type: none"> • NYU depth V2, • KITTI. | Zestawy tożsame z uczącymi. |
| ZoeDepth | konwolucyjna sieć neuronowa | nadzorowane i częściowo nadzorowane | <ul style="list-style-type: none"> • NYU depth V2, • KITTI, • HRWSI, • BlendedMVS, • ReDWeb, • DIML-Indoor, • 3D Movies, • MegaDepth, • WSVD, • TartanAir, • ApolloScape, • IRS. | <ul style="list-style-type: none"> • SUN RGB-D, • iBims, • DIODE, • HyperSim, • DDAD, • DIML, • Virtual KITTI 2. |
| UniDepth | W zależności od konfiguracji konwolucyjna sieć neuronowa lub transformator. | nadzorowane | <ul style="list-style-type: none"> • Argoverse2, • Waymo, • DrivingStereo, • Cityscapes, • BDD100K, • MapillaryPSD, • A2D2, • ScanNet, • Taskonomy. | <ul style="list-style-type: none"> • SUN-RGBD, • Diode Indoor, • IBims-1, • VOID, • HAMMER, • ETH-3D, • nuScenes, • DDAD, • NYU-Depth V2, • KITTI. |

| | | | | |
|----------------|---------------|-----------------------|---|--|
| Depth Anything | transformator | częściowo nadzorowane | <ul style="list-style-type: none"> • zbiory oznaczone <ul style="list-style-type: none"> – BlendedMVS, – DIML, – HRWSI, – IRS, – MegaDepth, – TartanAir. • zbiory nieoznaczone <ul style="list-style-type: none"> – BDD100K, – Google Landmarks, – ImageNet-21K, – LSUN, – Objects365, – Open Images V7, – Places365, – OSA-1B. | <ul style="list-style-type: none"> • zbiory do oceny predykcji głębi relatywnej <ul style="list-style-type: none"> – NYU depth V2, – KITTI, – Sintel, – DDAD, – ETH3D, – DIODE, • zbiory do oceny predykcji głębi metrycznej <ul style="list-style-type: none"> – SUN RGB-D, – iBims-1, – HyperSim, – Virtual KITTI 2, – DIODE Outdoor. |
|----------------|---------------|-----------------------|---|--|

3.2 Zbiory danych

3.2.1 KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute)

Wiodącym zestawem danych używanym do trenowania i oceny algorytmów percepacji głębi jest opracowany przez niemiecki Instytut Technologii Karlsruhe'a oraz amerykański Instytut Technologii Toyota zbiór KITTI³ [10]. Zawiera on 93 tysiące obrazów RGB-D zarejestrowanych przy pomocy autorskiej platformy jezdnej Annieway składającej się z lasera firmy Velodyne [33], kamer kolorowych i monochromatycznych oraz systemu GPS zamontowanych na samochodzie osobowym. Obrazy należące do tego zbioru podzielone zostały na pięć kategorii: drogi, miasta, osiedla, kampus i osoby. Prezentują one zatem sceny zewnętrzne.



Rysunek 23. Rejestrująca platforma jezdna użyta w przygotowaniu zbioru KITTI. Źródło: [10]

³Nazwa KITTI jest skrótem nazw instytutów przez które zbiór został opracowany.

3.2.2 NYUv2 (NYU-Depth V2)

Drugim najczęściej wykorzystywanym zestawem obrazów jest NYUv2 przedstawiony w 2012 r. w [3]. Zestaw ten składa się z 407024 obrazów RGB z odpowiadającymi im mapami głębi przygotowanymi przy użyciu urządzenia Microsoft Kinect. Autorzy skategoryzowali obrazy w zbiorze na następujące kategorie: piwnice, łazienki, sypialnie, księgarnia, kawiarnia, salony, jadalnie, sklepy meblowe, biura, kuchnie, biblioteki, bawialnie i inne. Obrazy przygotowane przez autorów NYUv2 przedstawiają wyłącznie sceny wewnętrz budynków. Niniejszy zestaw jest również wykorzystywany w dziedzinie segmentacji obrazu ze względu na przygotowane oznaczenie obrazów.

3.2.3 DIODE (Dense Indoor and Outdoor Depth)

Zbiór DIODE [24] jest wyjątkowy na tle konkurencji przez wzgląd na różnorodność scen. Jest bowiem pierwszym publicznie dostępnym zestawem obrazów prezentujących sceny zewnętrzne i wewnętrzne. Większa różnorodność scen pozwala na uzyskanie lepszych wyników na płaszczyźnie generalizacji modeli percepcji głębi. Na ten zbiór składa się 8574 obrazów scen wewnętrznych oraz 16884 obrazów scen zewnętrznych zarejestrowanych za pomocą tego samego urządzenia - skanera FARO Focus S350. Przygotowane przez twórców zbioru porównanie z podobnymi zestawami wskazuje na wysoką dokładność i zasięg zastosowanej aparatury.

| | DIODE | NYUv2 | KITTI | MAKE3d |
|----------------------------|-------------|--------|---------------|---------|
| Return Density (Empirical) | 99.6%/66.9% | 68% | 16% | 0.38% |
| # Images Indoor/Outdoor | 8574/16884 | 1449/0 | 0/94000 | 0/534 |
| Sensor Depth Precision | ±1 mm | ±1 cm | ±2 cm | ±3.5 cm |
| Sensor Angular Resolution | 0.009° | 0.09° | 0.08°H, 0.4°V | 0.25° |
| Sensor Max Range | 350 m | 5 m | 120 m | 80 m |
| Sensor Min Range | 0.6 m | 0.5 m | 0.9 m | 1 m |

Rysunek 24. Porównanie statystyk zbioru DIODE z innymi popularnymi zbiorami danych. Źródło: [24]

3.2.4 SUN RGB-D

Główym założeniem zestawu SUN RGB-D [23] jest dostarczenie danych dla modeli interpretujących trójwymiarowe sceny. Składa się on z 10335 obrazów pomieszczeń wewnętrznych z mapami głębi pochodzącyimi z sensorów Intel Realsense, Asus Xtion i obu wersji Microsoft Kinect. Na rzeczone obrazy zostały naniesione trójwymiarowe oznaczenia widniejących przedmiotów. Z powodu odmiennego przeznaczenia, zbiór ten jest często wykorzystywany w celu oceny działania modeli percepcji głębi.

3.2.5 Matterport3D

W 2017 r. firma Matterport zaprezentowała zestaw danych Matterport3D [2] przygotowany przy pomocy autorskiego urządzenia rejestrującego. Zestaw składa się z 10800 zdjęć panoramicznych

Rozdział 3. Przegląd istniejących rozwiązań

złożonych z 194400 obrazów z odpowiadającą im mapą głębi. Zdjęcia w zbiorze przedstawiają 90 scen przedstawiających wnętrza budynków.

3.2.6 DDAD (Dense Depth for Autonomous Driving)

Zestaw DDAD [11] został stworzony przez Toyota Research Institute. Zawiera 17050 obrazów treningowych i 4150 obrazów do oceny modelu, w tym zróżnicowane próbki scen miejskich i autostradowych z całego świata nagrane przez flotę samochodów autonomicznych wyposażonych w kamery i lasery LIDAR Luminar-H2. Wykorzystywany jest głównie do ewaluacji i rozwijania metod estymacji głębi w prowadzeniu pojazdów. Zestaw DDAD został wykorzystany do przetrenowania modelu PackNet tego samego autorstwa, nie osiąga on jednak wyników porównywalnych do wybranych w niniejszej pracy modeli.

3.2.7 Podsumowanie

Wykazane w niniejszym rozdziale zestawy danych przyczyniły się w znacznej mierze do rozwoju metod predykcji głębi. Rozbieżność ich cech z kolei przyczynia się pozytywnie do generalizacji modeli. Poniższa tabela stanowi wykaz przedstawionych zestawów z podziałem na charakterystykę scen, liczebność i urządzenie rejestrujące.

Tabela 2. Podsumowanie przedstawionych zbiorów danych używanych przez algorytmy percepji głębi.

| Nazwa | Charakterystyka obrazów | Liczebność zbioru | Urządzenie rejestrujące |
|--------------|-------------------------------|-------------------|---|
| KITTI | sceny zewnętrzne | 93000 | Skaner laserowy Velodyne [33] i system lokalizacji GPS. |
| NYUv2 | sceny wewnętrzne | 407024 | Microsoft Kinect |
| DIODE | sceny zewnętrzne i wewnętrzne | 25458 | Skaner FARO Focus S350 |
| SUN RGB-D | sceny wewnętrzne | 10335 | Intel RealSense 3D, Asus Xtion LIVE PRO i Microsoft Kinect. |
| Matterport3D | sceny wewnętrzne | 194400 | Autorska konstrukcja Matterport. |
| DDAD | sceny zewnętrzne | 21200 | Luminar-H2 |

Rozdział 4

Przedstawienie zastosowanych narzędzi

4.1 Język programowania

Obecnie, Python, język programowania wysokiego poziomu ogólnego przeznaczenia, wprowadzony w 1991 roku przez Guido van Rossum'a [32], dominuje w dziedzinie sztucznej inteligencji. Charakteryzuje się on prostą składnią, która ułatwia naukę i stosowanie w praktyce, co czyni go szczególnie popularnym wśród inżynierów danych. Python jest ceniony za wsparcie licznych bibliotek specjalizujących się w przetwarzaniu danych i uczeniu maszynowym, jak również za otwartoźródłowy charakter, co umożliwia szeroką współpracę w społeczności naukowej. W pracy przedstawione zostały modele wykorzystujące język Python w korelacji z biblioteką PyTorch [18], która jest rozwijana na bazie Torch i umożliwia efektywne budowanie oraz trenowanie modeli głębokiego uczenia z użyciem procesora graficznego.

4.2 Platforma obliczeniowa

W celu uruchomienia analizowanych metod neuronowych wizyjnych algorytmów percepcji głębi, wykorzystano platformę obliczeniową Google Colab. Jest to platforma sieciowa, która umożliwia uruchamianie skryptów w języku Python bezpośrednio w przeglądarce, co jest szczególnie korzystne w kontekście prac badawczych i edukacyjnych¹. Platforma ta opiera się na technologii Jupyter Notebook, co umożliwia interaktywne programowanie i wizualizację danych².

Google Colab oferuje dostęp do mocnych zasobów obliczeniowych, w tym do procesorów graficznych (GPU) i jednostek przetwarzania tensorów (TPU), które są kluczowe przy trenowaniu skomplikowanych modeli głębokiego uczenia. Użytkownicy mogą łatwo skalować użycie zasobów w zależności od potrzeb danego algorytmu, co znaczaco redukuje czas i koszt przetwarzania. Dostępność tych zasobów przez platformę internetową umożliwia również łatwe udostępnianie wyników i współpracę w ramach zespołów rozproszonych geograficznie.

¹<https://colab.google/>

²<https://jupyter.org/>

W kontekście przeprowadzonej analizy, Google Colab okazał się być nieocenionym narzędziem, które pozwoliło na efektywne wykonanie i analizę algorytmów percepji głębi. Możliwość wyświetlania wyników bezpośrednio w przeglądarce internetowej znaczco ułatwiała proces badawczy i pozwoliła na dynamiczne dopasowanie parametrów modelu w odpowiedzi na obserwowane wyniki.

4.3 Zestawy danych

Informacje o zastosowaniu do analizy porównawczej dostępnego publicznie zestawu danych i zestawu autorskiego (+ narzędzia wykorzystane do stworzenia autorskiego zestawu)

4.4 Narzędzia do analizy i wizualizacji

Tutaj będzie m.in. informacja o użyciu biblioteki Matplotlib i OpenCV.

Rozdział 5

Zbiór danych do analizy porównawczej

Rozdział 6

Metodologia - opis kryteriów oceny algorytmów

Rozdział 7

Analiza i wyniki

Rozdział 8

Podsumowanie i wnioski

Bibliografia

- [1] Bhat, S. F., Birkl, R., Wofk, D., Wonka, P. i Müller, M., „ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth”, *arXiv*, 2023. DOI: [10.48550/arXiv.2302.12288](https://doi.org/10.48550/arXiv.2302.12288).
- [2] Chang, A., Dai, A., Funkhouser, T., Halber, M., Nießner, M., Savva, M., Song, S., Zeng, A. i Zhang, Y., „Matterport3D: Learning from RGB-D Data in Indoor Environments”, *arXiv*, 2017. DOI: [10.48550/arXiv.1709.06158](https://doi.org/10.48550/arXiv.1709.06158).
- [3] Couprie, C., Farabet, C., Najman, L. i LeCun, Y., „Indoor Semantic Segmentation using depth information”, *arXiv*, 2013. DOI: [10.48550/arXiv.1301.3572](https://doi.org/10.48550/arXiv.1301.3572).
- [4] David Eigen Christian Puhrsch, R. F., „Depth Map Prediction from a Single Image using a Multi-Scale Deep Network”, *arXiv*, 2014. DOI: [10.48550/arXiv.1406.2283](https://doi.org/10.48550/arXiv.1406.2283).
- [5] Dong, X., Garratt, M. A., Anavatti, S. G. i Abbass, H. A., „MobileXNet: An Efficient Convolutional Neural Network for Monocular Depth Estimation”, *IEEE Transactions on Intelligent Transportation Systems*, 2022. DOI: [10.1109/TITS.2022.3179365](https://doi.org/10.1109/TITS.2022.3179365).
- [6] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. i Houlsby, N., „An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, *arXiv*, 2020. DOI: [10.48550/arXiv.2010.11929](https://doi.org/10.48550/arXiv.2010.11929).
- [7] Dubik, A., „1000 słów o laserach i promieniowaniu laserowym”, w. Wydawnictwo Ministerstwa Obrony Narodowej, 1989, s. 154–155, ISBN: 83-11-07495-X.
- [8] Dumoulin, V. i Visin, F., „A guide to convolution arithmetic for deep learning”, *arXiv*, 2018. DOI: [10.48550/arXiv.1603.07285](https://doi.org/10.48550/arXiv.1603.07285).
- [9] Fukushima, K., „Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.”, *Biol. Cybernetics*, 1980. DOI: [10.1007/BF00344251](https://doi.org/10.1007/BF00344251).
- [10] Geiger, A., Lenz, P. i Urtasun, R., „Are we ready for autonomous driving? The KITTI vision benchmark suite”, *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012. DOI: [10.1109/CVPR.2012.6248074](https://doi.org/10.1109/CVPR.2012.6248074).
- [11] Guizilini, V., Ambrus, R., Pillai, S., Raventos, A. i Gaidon, A., „3D Packing for Self-Supervised Monocular Depth Estimation”, *arXiv*, 2020. DOI: [10.48550/arXiv.1905.02693](https://doi.org/10.48550/arXiv.1905.02693).
- [12] He, K., Zhang, X., Ren, S. i Sun, J., „Deep Residual Learning for Image Recognition”, *arXiv*, 2015. DOI: [10.48550/arXiv.1512.03385](https://doi.org/10.48550/arXiv.1512.03385).

Bibliografia

- [13] Lavreniuk, M., Bhat, S. F., Müller, M. i Wonka, P., „EVP: Enhanced Visual Perception using Inverse Multi-Attentive Feature Refinement and Regularized Image-Text Alignment”, *arXiv*, 2023. DOI: [10.48550/arXiv.2312.08548](https://doi.org/10.48550/arXiv.2312.08548).
- [14] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N. i Terzopoulos, D., „An Introduction to Convolutional Neural Networks”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, t. 44, s. 3523–3542, 2021. DOI: [10.1109/TPAMI.2021.3059968](https://doi.org/10.1109/TPAMI.2021.3059968).
- [15] O'Shea, K. i Nash, R., „An Introduction to Convolutional Neural Networks”, *arXiv*, 2015. DOI: [10.48550/arXiv.1511.08458](https://doi.org/10.48550/arXiv.1511.08458).
- [16] Oord, A. van den, Vinyals, O. i Kavukcuoglu, K., „Neural Discrete Representation Learning”, *arXiv*, 2018. DOI: [10.48550/arXiv.1711.00937](https://doi.org/10.48550/arXiv.1711.00937).
- [17] Oquab, M., Darabet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A. i Bojanowski, P., „DINOv2: Learning Robust Visual Features without Supervision”, *arXiv*, 2024. DOI: [10.48550/arXiv.2304.07193](https://doi.org/10.48550/arXiv.2304.07193).
- [18] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. i Chintala, S., „PyTorch: An Imperative Style, High-Performance Deep Learning Library”, *arXiv*, 2019. DOI: [10.48550/arXiv.1912.01703](https://doi.org/10.48550/arXiv.1912.01703).
- [19] Piccinelli, L., Yang, Y.-H., Sakaridis, C., Segu, M., Li, S., Van Gool, L. i Yu, F., „UniDepth: Universal Monocular Metric Depth Estimation”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. DOI: [10.48550/arXiv.2403.18913](https://doi.org/10.48550/arXiv.2403.18913).
- [20] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K. i Koltun, V., „Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer”, *arXiv*, 2020. DOI: [10.48550/arXiv.1907.01341](https://doi.org/10.48550/arXiv.1907.01341).
- [21] Ronneberger, O., Fischer, P. i Brox, T., „U-Net: Convolutional Networks for Biomedical Image Segmentation”, *arXiv*, 2015. DOI: [10.48550/arXiv.1505.04597](https://doi.org/10.48550/arXiv.1505.04597).
- [22] Ryszard Tadeusiewicz, M. F., *Rozpoznawanie obrazów*. Państwowe Wydawnictwo Naukowe, 1991, ISBN: 83-01-10558-5.
- [23] Song, S., Lichtenberg, S. P. i Xiao, J., „SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [24] Vasiljevic, I., Kolkin, N., Zhang, S., Luo, R., Wang, H., Dai, F. Z., Daniele, A. F., Mostajabi, M., Basart, S., Walter, M. R. i Shakhnarovich, G., „DIODE: A Dense Indoor and Outdoor DEpth Dataset”, *arXiv*, 2019. DOI: [10.48550/arXiv.1908.00463](https://doi.org/10.48550/arXiv.1908.00463).
- [25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. i Polosukhin, I., „Attention Is All You Need”, *arXiv*, 2023. DOI: [10.48550/arXiv.1706.03762](https://doi.org/10.48550/arXiv.1706.03762).

- [26] Wan, Q., Huang, Z., Kang, B., Feng, J. i Zhang, L., „Harnessing Diffusion Models for Visual Perception with Meta Prompts”, *arXiv*, 2023. DOI: [10.48550/arXiv.2312.14733](https://doi.org/10.48550/arXiv.2312.14733).
- [27] Warren S. McCulloch, W. P., „A logical calculus of the ideas immanent in nervous activity”, *The bulletin of mathematical biophysics*, t. 5, 1943. DOI: [10.1007/BF02478259](https://doi.org/10.1007/BF02478259).
- [28] Xian, K., Zhang, J., Wang, O., Mai, L., Lin, Z. i Cao, Z., „Structure-Guided Ranking Loss for Single Image Depth Prediction”, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [29] Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J. i Zhao, H., „Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data”, *arXiv*, 2024. DOI: [10.48550/arXiv.2401.10891](https://doi.org/10.48550/arXiv.2401.10891).
- [30] Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S. i Shen, C., „An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, *arXiv*, 2020. DOI: [10.48550/arXiv.2012.09365](https://doi.org/10.48550/arXiv.2012.09365).
- [31] Zhao, W., Rao, Y., Liu, Z., Liu, B., Zhou, J. i Lu, J., „Unleashing Text-to-Image Diffusion Models for Visual Perception”, *arXiv*, 2023. DOI: [10.48550/arXiv.2303.02153](https://doi.org/10.48550/arXiv.2303.02153).
- [32] Python Software Foundation, *Python: informacje o języku*, <https://www.python.org/about/>, data dostępu: 24.04.2024.
- [33] Velodyne Lidar, *Velodyne Lidar: informacje o firmie*, <https://velodynelidar.com/about/>, data dostępu: 21.04.2024.

Wykaz skrótów i symboli

| | |
|-------|---|
| ANN | sztuczna sieć neuronowa (ang. Artificial Neural Network) |
| CNN | splotowa sieć neuronowa (ang. Convolutional Neural Network) |
| FLOPS | liczba operacji zmiennoprzecinkowych na sekundę |
| GPU | procesor graficzny (ang. Graphical Processing Unit) |
| TPU | tensorowa jednostka przetwarzania (ang. Tensor Processing Unit) |

Spis rysunków

| | | |
|----|--|----|
| 1 | Fotografia i odpowiadająca jej mapa głębi. Źródło: własne | 2 |
| 2 | Poglądowy model uczenia nadzorowanego. Wejścia stanowią obraz RGB oraz pomiary głębi a wynikiem jest predykcja mapy głębi. | 6 |
| 3 | Schemat przykładowego uczenia nienadzorowanego. Wejścia stanowią trzy kadry z nagrania RGB a wynikiem jest predykcja mapy głębi. | 6 |
| 4 | Schemat podsumowujący paradygmaty uczenia algorytmów. | 7 |
| 5 | Przykład działania warstwy konwolucyjnej z jądrem o rozmiarze 3x3. Źródło: [8] . . . | 8 |
| 6 | Przykład architektury sieci konwolucyjnej użytej w celu rozpoznania głębi obrazu. Źródło: [5] | 8 |
| 7 | Schemat modelu transformatora wizyjnego. Źródło: [6] | 9 |
| 8 | Przykładowy wynik działania algorytmu AdelaiDepth. Źródło: [30] | 11 |
| 9 | Porównanie osiąganych wyników przeprowadzone na ośmiu zestawach danych nieuczestniczących w procesie uczenia. Źródło: [30] | 12 |
| 10 | Przykładowe wyniki działania modułu estymacji głębi MetaPrompt-SD. Źródło: [26] | 12 |
| 11 | Schemat architektury algorytmu MetaPrompt-SD. Źródło: [26] | 13 |
| 12 | Porównanie osiąganych wyników przeprowadzone na dwóch zestawach danych. Źródło: [26] | 13 |
| 13 | Schemat architektury algorytmu EVP. Źródło: [13] | 14 |
| 14 | Porównanie osiąganych wyników przeprowadzone na zbiorze KITTI. Źródło: [13] . . . | 14 |
| 15 | Schemat architektury algorytmu ZoeDepth. Źródło: [1] | 15 |
| 16 | Porównanie osiąganych wyników przeprowadzone na zbiorze NYU-Depth v2. Źródło: [1] | 15 |
| 17 | Przykładowe wyniki działania modelu Depth Anything. Źródło: [29] | 16 |
| 18 | Schemat architektury algorytmu Depth Anything. Źródło: [29] | 16 |
| 19 | Zbiór zestawów danych uczących Depth Anything. Źródło: [29] | 17 |
| 20 | Porównanie rezultatów Depth Anything dokonane na podstawie zbioru NYUv2 (po lewej) i KITTI (po prawej). Źródło: [29] | 17 |
| 21 | Schemat architektury algorytmu UniDepth. Źródło: [19] | 18 |
| 22 | Porównanie rezultatów UniDepth dokonane na zbiorach danych niewidzianych podczas uczenia. Źródło: [19] | 18 |

| | | |
|----|---|----|
| 23 | Rejestrująca platforma jezdna użyta w przygotowaniu zbioru KITTI. Źródło: [10] . . . | 20 |
| 24 | Porównanie statystyk zbioru DIODE z innymi popularnymi zbiorami danych. Źródło: [24] | 21 |

Spis tabel

| | | |
|---|--|----|
| 1 | Podsumowanie przedstawionych modeli percepcji głębi. | 19 |
| 2 | Podsumowanie przedstawionych zbiorów danych używanych przez algorytmy percepcji głębi. | 22 |

Spis załączników

| | | |
|---|--|----|
| 1 | Specyfikacja komputera użytego w pracy | 45 |
| 2 | Dodatkowe zdjęcia badanego obiektu | 47 |

Załącznik 1

Specyfikacja komputera użytego w pracy

| Laptop | |
|-----------------------|--|
| Lenovo YOGA 720-15IKB | |
| OS | Ubuntu 18.04.4 LTS |
| OS typ | 64-bit |
| CPU | Intel Core i7-7700HQ CPU @ 2.80GHz x 8 |
| GPU | GeForce GTX 1050/PCIe/SSE2 |
| RAM | 16GiB |

Załącznik 2

Dodatkowe zdjęcia badanego obiektu

Tu można wstawić dodatkowe zdjęcia, które zajęłyby zbyt dużo miejsce w samej pracy, ale dla lepszej ilustracji tego co zostało zrobione, można je umieścić w załączniku.