

Politechnika Warszawska

W Y D Z I A Ł E L E K T R O N I K I
I T E C H N I K I N F O R M A C Y J N Y C H



Instytut Radioelektroniki i Technik Multimedialnych

Praca dyplomowa

na kierunku Studia Podyplomowe
w specjalności Głębokie Sieci Neuronowe - Zastosowania w Mediach Cyfrowych

Metoda detekcji uczestników ruchu drogowego na podstawie obrazów
radarowych

Janko Muzykant

Numer albumu 283673

promotor
dr Jan Sebastian

WARSZAWA 2024

Metoda detekcji uczestników ruchu drogowego na podstawie obrazów radarowych

Streszczenie.

Głównym celem pracy jest Nacisk w pracy został położony na ... W pierwszej części pracy przedstawiono ... Następnie opisano ... Druga część pracy rozpoczyna się od ... Eksperymenty opisane w dalszej części pracy przedstawiają ... Dodatkowym etapem pracy jest ... W zakończeniu pracy przedstawiono podsumowanie wykonanych działań oraz dalsze działania, które mogą przynieść poprawę ...

Słowa kluczowe: NFT, ChatGPT, Buzzword300



.....
miejscowość i data

.....
imię i nazwisko studenta

.....
numer albumu

.....
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 związanego 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 (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....
czytelny podpis studenta

Spis treści

1. Wstęp	9
2. Cel pracy	10
2.1. Zbiór danych	10
2.2. Ocena rozwiązania	10
3. Przegląd literatury	11
4. Opis rozwiązania	12
4.1. Zbiór danych	12
4.2. Augmentacja danych	12
4.3. Trening i ewaluacja	12
4.4. Eksperymenty	12
5. Wyniki ewaluacji eksperymentalnej	13
6. Podsumowanie	14
Spis rysunków	15
Spis tabel	15

1. Wstęp

Od wynalezienia perceptronu, które można uznać za pierwszy etap rozwoju sztucznych sieci neuronowych, ich podstawowym zastosowaniem było rozpoznawanie obrazów. Zapoczątkowało to dziedzinę wizji komputerowej, która z czasem objęła szeroki wachlarz zagadnień i technik. Obecnie jest to jeden z najbardziej rozwiniętych obszarów badań wykorzystujących sieci neuronowe.

Niniejsza praca zajmuje się jednym z klasycznych zagadnień wizji komputerowej, jakim jest detekcja obiektów. Przedstawia ona rozwiązanie zadania konkursowego *SpaceNet 8: Flood Detection Challenge Using Multiclass Segmentation spacenet8*, które polegało na detekcji dróg i budynków na zdjęciach satelitarnych oraz klasyfikacji zniszczeń tych obiektów dokonanych przez powodzie. Tym, co odróżnia to zadanie od typowego zagadnienia segmentacji obrazu jest to, że dla każdej lokalizacji ze zbioru danych są dostępne dwa zdjęcia – przed i po katastrofie. To pozwala zastosować modele o bardziej złożonej architekturze, np. sieć syamską.

Cały kod potrzebny do ukończenia pracy został napisany przez autorów w języku Python. Wykorzystano gotowy zbiór danych dostarczony przez organizatorów konkursu. Zaimplementowano własne procedury do trenowania i ewaluacji modeli oparte na bibliotece Pytorch Lightning i wykorzystujące serwis neptune.ai do zbierania wyników. Do implementacji architektury modeli wykorzystano bibliotekę MMSegmentation. Modele trenowane przy użyciu pojedynczych kart graficznych.

2. Cel pracy

Niniejsza praca przedstawia rozwiązywanie zadania konkursowego *SpaceNet 8: Flood Detection Challenge Using Multiclass Segmentation spacenet8*. Celem konkursu, który zakończył się w październiku 2022 roku, było wyłonienie najlepszych algorytmów wykrywających budynki i drogi, a także ich zniszczenia na skutek katastrof naturalnych, na zdjęciach satelitarnych. Oceniane były rozwiązania dwóch zagadnień:

1. Segmentacja budynków i dróg na pojedynczych zdjęciach oraz klasyfikacja dróg ze względu na maksymalną dozwoloną prędkość
2. Segmentacja zniszczonych budynków i dróg na podstawie par zdjęć wykonanych przed i po katastrofie naturalnej

Ta praca skupia się na rozwiązyaniu drugiego zagadnienia.

2.1. Zbiór danych

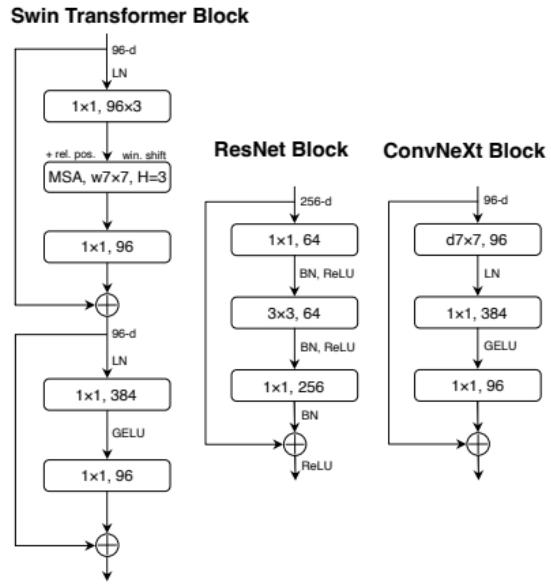
Zbiór danych udostępniony przez organizatorów konkursu składa się ze zdjęć satelitarnych w formacie TIFF, podzielonych na zdjęcia wykonane przed i po katastrofie, adnotacji budynków i dróg w formacie GeoJSON oraz plików CSV zawierających przypisanie adnotacji do zdjęć. Podział danych na zbiór treningowy i testowy jest zadany z góry.

2.2. Ocena rozwiązania

3. Przegląd literatury

Przegląd literatury związanej z rozwiązywanym problemem.

W pracy **he2016deep** opisano W **liu2022convnet** zaproponowano nowatorską architekturę.... Przykładowy obraz 3.1.



Rysunek 3.1. Porównanie bloków rezydualnych wykorzystywanych w sieci Swin Transformer, ResNet i ConvNeXt.

4. Opis rozwiązania

Rozwiązanie zadania konkursowego zostało w całości zaimplementowane w języku Python w wersji 3.8. Najważniejsze biblioteki jakie zostały wykorzystane w rozwiążaniu to:

1. torch do obsługi danych i modeli.
2. albumentations do przetwarzania obrazów.
3. pytorch_lightning do trenowania i ewaluacji modeli.
4. segmentation-models-pytorch do budowy modeli.
5. neptune do monitorowania przebiegu treningu i ewaluacji.

4.1. Zbiór danych

Do wstępnej obróbki zbioru danych wykorzystano skrypty dostarczone przez organizatorów konkursu razem z rozwiązaniem *baseline* w niezmienionej postaci. Przetwarzają one adnotacje w formacie GeoJSON na obrazy w formacie TIFF i rozdzielczości identycznej jak odpowiadające im zdjęcia. Dla jednej pary zdjęć (przed i po katastrofie) generowane są maksymalnie cztery obrazy:

1. Jednokanałowa maska budynków o wartościach 0, 1.
2. Jednokanałowa maska dróg o wartościach 0, 1.
3. Jednokanałowa maska dróg o wartościach 0 – 7, odpowiadających różnym rodzajom dróg. Nieużywana w tej pracy.
4. Czterokanałowa maska budynków i dróg po katastrofie o wartościach 0, 1. Kanały odpowiadają następującym obiektom, w kolejności: niezniszczony budynek, zniszczony budynek, niezniszczona droga, zniszczona droga.

Po przetworzeniu zbiór danych zawierał 801 par zdjęć z kompletem masek dróg i budynków przed i po katastrofie. Według podziału narzuconego w konkursie 679 par znalazło się w zbiorze treningowym i 122 w zbiorze testowym.

Zdjęcia przed katastrofą miały rozdzielcość 1300×1300 pikseli, natomiast zdjęcia po katastrofie miały różne rozdzielcości. Aby efektywnie używać ich jako wsadu do modeli o różnych architekturach, zaimplementowano skalowanie wszystkich obrazów i masek po ich wczytaniu do tej samej rozdzielcości. W eksperymentach ustalono tę rozdzielcość na 1024×1024 piksele.

4.2. Augmentacja danych

Ze względu na mały rozmiar zbioru danych, kluczowe znaczenie dla rozwiązania miała augmentacja danych treningowych.

4.3. Trening i ewaluacja

4.4. Eksperymenty



Rysunek 4.1. Przykładowa para zdjęć ze zbioru danych z adnotacjami. Wszystkie obiekty na zdjęciu po prawej są sklasyfikowane jako zniszczone.

5. Wyniki ewaluacji eksperymentalnej

6. Podsumowanie

Podsumowanie i krytyczna analiza osiągniętych rezultatów. Ocena, czy osiągnięto założony cel pracy. Dyskusja co można było zrobić lepiej i propozycja dalszych prac w celu usprawdzenia opracowanego rozwiązania.

Spis rysunków

3.1 Porównanie bloków rezydualnych wykorzystywanych w sieci Swin Transformer, ResNet i ConvNeXt.	11
4.1 Wizualizacja danych	13

Spis tabel