



Michel Voss

Detekcja kolektorów i paneli słonecznych na podstawie zdjęć lotniczych miasta Poznania z wykorzystaniem głębokich sieci neuronowych

Detection of solar panels based on aerial images of the city of Poznań using deep neural networks

Praca licencjacka

Promotor: dr Maciej Beręsewicz
Pracę przyjęto dnia:

Podpis Promotora

Kierunek: Informatyka i ekonometria
Specjalność: Elektroniczny biznes

Poznań 2019

Spis treści

Wstęp	2
1 Pomiar odnawialnych źródeł energii	3
1.1 Energia konwencjonalna, a odnawialne źródła energii	3
1.1.1 Ekonomiczny aspekt wykorzystania energii ze źródeł odnawialnych	3
1.1.2 Ogniwa fotowoltaiczne a kolektory słoneczne	5
1.1.3 Prawne aspekty wykorzystania odnawialnych źródeł energii w Polsce	7
1.2 Źródła danych o odnawialnych źródłach energii w Polsce	9
1.2.1 Energia odnawialna w polskiej statystyce publicznej	9
1.2.2 Urząd Regulacji Energetyki	11
1.2.3 Zdjęcia lotnicze – ortofotomapy	12
1.3 Wykorzystanie energii Słońca na świecie i w Polsce	13
1.3.1 Świat	13
1.3.2 Polska	14
1.4 Podsumowanie	15
2 Teoretyczne podstawy głębokich sieci neuronowych	16
2.1 Sztuczna inteligencja, uczenie maszynowe i uczenie głębokie	16
2.1.1 Wprowadzenie	16
2.1.2 Dotychczasowe osiągnięcia uczenia głębokiego	18
2.2 Anatomia sieci neuronowej	19
2.2.1 Reprezentacja danych sieci neuronowych – skalary, wektory, macierze, tensorzy	19
2.2.2 Sztuczne sieci neuronowe	20
2.2.3 Funkcja aktywacji	23

2.2.4	Wsteczna propagacja błędu	24
2.3	Konwolucyjne sieci neuronowe	25
2.3.1	Turi Create i TinyYolo	26
2.3.2	Zalety TinyYolo	34
2.3.3	Wykorzystane funkcje Turi Create	35
2.4	Podsumowanie	36
3	Zastosowanie sieci neuronowych w rozpoznawaniu paneli fotowoltaicznych i kolektorów słonecznych	37
3.1	Pozyskanie i przetwarzanie danych przestrzennych	37
3.1.1	Środowisko do obliczeń	37
3.1.2	Zdjęcia lotnicze miasta Poznania i ich przetwarzanie	38
3.1.3	Przygotowanie danych na potrzeby uczenia głębokiego	40
3.2	Wyniki detekcji kolektorów i paneli słonecznych	42
3.2.1	Ewaluacja zastosowanego modelu	42
3.2.2	Lokalizacja paneli i kolektorów słonecznych w Poznaniu w 2016 roku . .	48
Podsumowanie		52
Bibliografia		53
Spis tabel		55
Spis rysunków		57
Spis skryptów oprogramowania		58

Wstęp

Wykorzystanie odnawialnych źródeł energii jako alternatywy dla źródeł kopalnych jest jednym z priorytetów w kontekście zachodzących zmian klimatu. W ramach zobowiązań środowiskowych Unia Europejska wyznaczyła na rok 2020 tzw. „3x20%” czyli zbiór ustaw mających na celu zmniejszenie emisji gazów cieplarnianych o 20% w stosunku do 1990 roku, zwiększenie udziału zużycia energii pochodzącej z odnawialnych źródeł energii do 20%, zwiększenie o 20% efektywności energetycznej w stosunku do przewidywań. Ze względu na przyjęcie ustaw proponowanych przez Unię Europejską, przygotowano projekt „Polityki energetycznej Polski do 2030 roku” mający na celu znaczne zmniejszenie energochłonności.

Jednakże, aby ten cel osiągnąć istotne jest monitorowanie oraz posiadanie aktualnych źródeł danych w zakresie wykorzystania odnawialnych źródeł energii. W tym celu można wykorzystać informacje przygotowywane przez Główny Urząd Statystyczny, dane pochodzące od operatorów energii elektrycznej czy ze źródeł big data (m.in. zdjęć satelitarnych czy lotniczych).

Celem niniejszej pracy jest wykorzystanie głębokich sieci neuronowych do rozpoznawania systemów fotowoltaicznych (paneli oraz kolektorów słonecznych) na przykładzie zdjęć lotniczych miasta Poznania w 2016 roku. W pracy skupiono się wyłącznie na budynkach mieszkalnych (jednorodzinnych, dwurodzinnych i większych).

Metoda ta została wybrana ze względu na możliwość przetwarzania dużych zbiorów danych w krótkim czasie oraz łatwość wykorzystania stworzonego modelu. Natomiast, główną motywacją jest uzyskanie bardziej szczegółowych informacji o wykorzystaniu energii słońca w Poznaniu na podstawie źródeł big data, które do tej pory nie były wykorzystywane w tym celu.

W pierwszym rozdziale zostanie przedstawione pojęcie instalacji fotowoltaicznych, a także kierunek rozwoju energii odnawialnych. Omówiona zostanie również sytuacja prawa w Polsce przy instalacji systemów fotowoltaicznych oraz kwestie ekonomiczne w kraju jak i Europie.

Zrozumienie wykorzystanej technologii wymaga zagłębiania się w pojęcia głębokiego uczenia, konwolucyjnych sieci neuronowych oraz pakietu Turi Create w języku Python, które

zostaną opisane w drugim rozdziale. Pakiet ten reimplementuje bibliotekę Darknet oraz algorytm *You Only Look Once* (YOLO), który jest wykorzystywany do m.in. detekcji ponad 1000 obiektów w czasie rzeczywistym.

Trzeci rozdział zawiera informacje dotyczące wykorzystania środowiska chmurowego, przygotowania danych oraz wyników. Analiza została przeprowadzona na podstawie zdjęć lotniczych (tzw. ortofotomap) pozyskanych bezpłatnie z Zarządu Geodezji i Katastru Miejskiego GEOPOL. W pracy wykorzystano język R do wycięcia budynków mieszkalnych z ortofotomap (danych przestrzennych zapisanych w formacie tif). Uzyskany zbiór zdjęć powiększono ośmokrotnie poprzez obracanie i dodawanie efektu lustrzanego odbicia. Na takim zbiorze danych ręcznie oznaczano instalacje fotowoltaiczne i kolektory słoneczne przez ponad siedem godzin. Następnie, dokonano klasyfikacji paneli i kolektorów słonecznych z wykorzystaniem języka Python oraz pakietu Turi Create.

Uzyskany model charakteryzuje się bardzo wysoką skutecznością detekcji paneli oraz kolektorów słonecznych, a wyniki dotyczące ich lokalizacji są, zgodnie z wiedzą autora oraz promotora, pierwszym tego typu opracowaniem w Polsce.

Rozdział 1

Pomiar odnawialnych źródeł energii

1.1 Energia konwencjonalna, a odnawialne źródła energii

1.1.1 Ekonomiczny aspekt wykorzystania energii ze źródeł odnawialnych

Rozwój cywilizacji wiąże się ze wzrostem zużycia energii. Według Bogdanienko (1989) zarówno tempo spożycia energii, jak i tempo wzrastającego zanieczyszczenia środowiska wykazuje charakter wykładniczy. Pomimo wprowadzenia przez wiele państw programów oszczędnościowych już teraz można wyznaczyć przewidywany rok wyczerpania zasobów paliw kopalnianych. Dotychczas nasza cywilizacja zużyła równoważność 500 mld ton paliwa umownego (Jastrzębska, 2017). Oszacowano, że XXI wieku nastąpi wzrost zużycia energii elektrycznej o około 250-300% (SEV/VSE, 1996).

W pracach Bogdanienko (1989), Ciechanowicz (1997) podano przybliżone lata wyczerpania zasobów paliw: ropy naftowej po 2050 roku, gazu po 2060 roku i węgla po 2200 roku. Ze względu na powyższe szacunki prowadzone są intensywne prace na wykorzystaniu odnawialnych źródeł energii.

Odnawialne źródła energii (inaczej pierwotne) są wynikiem działania Słońca, energii wnętrza Ziemi i ruchu grawitacyjnego Księżyca. Ich działanie dostarcza również inne formy energii: biomasy, promieni słonecznych, cyrkulacji wody czy wiatru. Zasoby tych źródeł energii odnawiają się relatywnie szybko, stąd nie ulegną wyczerpaniu jak zasoby paliw naturalnych. Oprócz energii elektrycznej mogą dostarczyć również ciepło. Według *Renewable Global Status Report* w 2017 roku udział OZE w globalnej produkcji energii elektrycznej osiągnął 26,5% (woda 16,4%, wiatr 5,6 %, biomasa 2,2 %, panele fotowoltaiczne 1,9% i inne 0,4%.

Potencjał ujęty we wszystkich odnawialnych źródłach energii zapewnia rocznie kilkukrotnie większe zasoby niż obecne zużycie. Największy potencjał ze źródeł odnawialnych ma energia Słońca. Wartość światowej konsumpcji energii w 2009 roku według Perez i Perez (2015) wyniosła 18,5 TWrok¹, zaś w prognozach na rok 2050 założono 27 TWrok. W tabeli 1.1 przedstawiono możliwości pozyskiwania energii dla wybranych źródeł, w skali roku według Perez i Perez (2015).

Tabela 1.1. Możliwości pozyskiwania energii z wybranych źródeł w skali roku w TWrok, według Perez i Perez (2015)

Źródło	Słońce	Wiatr	Biomasa	Woda	Geotermia	Pływy	OTEC
Dostępna energia	23 000	75-130	2-6	3-4	0,2-3	0,2-2	3-11

Źródło: Perez i Perez (2015), s. 2. Objasnenia: OTEC – Ocean Thermal Energy Conversion.

Główym zadaniem w dzisiejszych czasach będzie nie tylko zastępowanie tradycyjnych metod pozyskiwania energii, ale również oszczędzanie. Według Jastrzębskiej (2017), w Europie można już dziś zaoszczędzić energię elektryczną w gospodarstwach domowych od 15% do 35%, a w handlu i komunikacji nawet do 50% w porównaniu do aktualnego zużycia. Według szacunków Parlamentu i Komisji Europejskiej rok 2050 będzie przełomowy nie tylko ze względu na prawie całkowite pokrycie zapotrzebowania na energię elektryczną, również na znaczne zmniejszenie zużycia.

Odpowiednie wykorzystanie energii ze źródeł odnawialnych stanowi istotny element stabilnego rozwoju państwa. Sektor energetyczny w Polsce aktualnie zmaga się z poważnymi wyzwaniami. Uzależnienie od zewnętrznych dostaw gazu ziemnego, ropy naftowej, zobowiązań wobec środowiska, coraz większe zapotrzebowanie na energię czy dekapitalizacja infrastruktury elektroenergetycznej wymagają zdecydowanych działań powstrzymujących pogorszeniu się sytuacji odbiorców energii.

W ramach zobowiązań środowiskowych Unia Europejska wyznaczyła na rok 2020 tzw. „3x20%” czyli zbiór ustaw mających na celu zmniejszenie emisji gazów cieplarnianych o 20% w stosunku do 1990 roku, zwiększenie udziału zużycia energii pochodzącej z odnawialnych źródeł energii do 20%, zwiększenie o 20% efektywności energetycznej w stosunku do przewidywań. Ze względu na przyjęcie ustaw proponowanych przez Unię Europejską, przygotowano projekt „Polityki energetycznej Polski do 2030 roku” mający na celu znaczne zmniejszenie energochłonności. Zarówno w ustawie Komisji Europejskiej, jak i projekcie rządowym, jednym z głównych założeń polityki energetycznej Polski do 2020 roku, to osiągnięcie 15% udziału energii ze źródeł odnawialnych w strukturze energii finalnej.

¹1TWrok to 8766 terawat godzin energii.

Osiągnięcie powyższego celu wymaga zwiększenia liczby inwestycji w sektorze energetyki odnawialnej. Energia słoneczna ma zdecydowanie największy potencjał z odnawialnych źródeł energii. Według Matuszczyk, Popławski i Flasza (2015) całkowita moc promieniowania elektromagnetycznego Słońca docierającego do biosfery przekracza 10 000 razy obecne zapotrzebowanie całej ludzkości na energię.

1.1.2 Ogniwa fotowoltaiczne a kolektory słoneczne

1.1.2.1 Fotowoltaika

Nazwa „fotowoltaika” to połączenie dwóch słów: foto – czyli światło oraz wolt – czyli napięcie prądu. Zamiana promieniowania słonecznego na energię elektryczną odbywa się za pomocą ogniw fotowoltaicznego. Rysunek 1.1 przedstawia przykładową instalację fotowoltaiczną na budynku jednorodzinnym.



Rysunek 1.1. Przykładowa instalacja fotowoltaiczna na dachu

Źródło: Zdjęcie lotnicze jednego z budynków w Poznaniu.

Każde z ogniw składa się z dwóch warstw: pozytywnej i negatywnej. W momencie padania promieni słonecznych na ognivo wytwarza się napięcie. Pojedyncze ogniwa tworzą małe napięcie, dlatego są łączone w panele, aby generowały większą moc. Po podłączeniu, do ogniwa, urządzenia pobierające energię, otrzymujemy prąd stały. Dzięki przetwornikowi prąd stały zo-

staje przetworzony na prąd zmienny i może zostać podłączony do sieci budynku.

Po raz pierwszy zjawisko fotowoltaiczne zostało zaobserwowane przez A. C Becquerela w 1839 roku. Rozwój fotowoltaiki miał miejsce w latach sześćdziesiątych XX wieku zapoczątkowany przez badania kosmiczne. Przełomowy był rok 2004, w którym nastąpił znaczny wzrost produkcji (67% w stosunku do poprzedniego roku) ze względu na dużą podaż krzemu, który stanowi główny materiał w produkcji ogniw. Obecnie dostępnych na rynku jest kilka generacji ogniw. Każde z ogniw można scharakteryzować kilkoma parametrami. Produkcja w ostatnich latach ciągle wzrasta, nastąpił rozwój nowych technologii. Według Fraunhofer Institute for Solar Energy Systems ([2018](#)), w globalnej produkcji modułów fotowoltaicznych w 2017 roku udział ogniw krzemowych multikryształycznych wyniósł 60,8 GW^p², monokryształycznych 32,2 GW^p, zaś technologii cienkowarstwowych 4,5 GW^p.

1.1.2.2 Kolektory słoneczne

Kolektor słoneczny to urządzenie do zamiany energii promieniowania słonecznego na ciepło. W przeciwieństwie do paneli fotowoltaicznych, kolektory wykorzystywane są do generowania ciepła, a nie prądu. Rysunek 1.2 przedstawia przykładową instalację kolektorów słonecznych na budynku jednorodzinnym.



Rysunek 1.2. Kolektor słoneczny

Źródło: Zdjęcie pochodzi z portalu www.czysteogrzewanie.pl.

Promienie słoneczne są kumulowane, a następnie podgrzewany jest nośnik ciepła, który przemieszcza się do zbiornika. Tak jak panele, kolektory instalowane są na dachach, gdzie naj-

²GW^p – gigawat mocy wyjściowej modułu fotowoltaicznego, uzyskanego w wyniku konwersji fotowoltaicznej w warunkach optymalnego dopasowania obciążenia

większe jest skupisko promieni słonecznych. Kolektor słoneczny został opatentowany w 1986 roku w Anglii. Kolektory można podzielić na:

- płaskie,
 - cieczowe,
 - gazowe,
 - dwufazowe,
- płaskie próżniowe,
- próżniowo-rurowe,
- skupiające,
- specjalne.

Według EurObserv'ER (2018b) roczna powierzchnia kolektorów słonecznych w 2017 roku w Unii Europejskiej spadła poniżej 2 mln m², tj. o 24,2% w porównaniu do powierzchni za-instalowanej w poprzednim roku (przy czym w Danii, dotychczas państwa z drugą największą powierzchnią kolektorów, aż o 34%). Europejski rynek kolektorów słonecznych ma trudności ze stabilizacją (Hiszpania, Austria, Polska) lub jest uwiezione w spirali spadkowej (Niemcy, Francja, Włochy i Belgia). Technologia walczy o przetrwanie na rynku ogrzewania i produkcji ciepłej wody pomimo niepodważalnej efektywności energetycznej i korzyści wynikających z bilansu CO₂. Kolektory słoneczne muszą przeciwstawić się ostrej konkurencji ze strony innych technologii produkcji energii cieplnej ze źródeł odnawialnych, takich jak pompki ciepła czy kotły spalające biomasę. Ponadto rynek kolektorów został dotknięty cięciami dofinansowań w niektórych krajach z powodu ograniczeń budżetowych lub programów wsparcia dostosowanych do konkurencyjnych technologii. Spodziewane wzrosty w sektorze ciepłowniczym, ciepłownictwie słonecznym i ciepłownictwie przemysłowym, pojawiają się stopniowo, ale są one zbyt małe, aby przeciwdziałać spadkowi wielkości instalacji w sektorze domów jednorodzinnych.

1.1.3 Prawne aspekty wykorzystania odnawialnych źródeł energii w Polsce

Przyjęcie zbioru ustaw „3x20%” poskutkowało nowelizacją ustawy Prawo energetyczne. Ustawa uchwalona przez Sejm 23 lipca 2013r. nosi nazwę *mały trójpak*. Wdrożyła w życie prze-

pisy promujące wykorzystanie energii ze źródeł odnawialnych. Aktualna ustanowiona ustawą Prawo energetyczne pochodzi z dnia 7 czerwca 2018 roku.

Tabela 1.2. Porównanie wybranych aspektów dotyczących instalacji fotowoltaicznych i kolektorów słonecznych

	Instalacja fotowoltaiczna	Kolektory słoneczne
Cel instalacji	Zamiana energii słonecznej na energię elektryczną	Zamiana energii słonecznej na ciepło
Wymagane pozwolenia	Montaż nie wymaga pozwolenia na budowę, jeśli instalacja ma moc do 50kW i cała instalacja nie przekraczała 3 m wysokości. W przypadku ingerencji w obiekt potrzebne jest pozwolenie na budowę.	Montaż nie wymaga pozwolenia na budowę, jeśli kolektory są wolnostojące lub - w przypadku montażu na budynku - cała instalacja nie przekraczała 3 m wysokości. W przypadku ingerencji w obiekt potrzebne jest pozwolenie na budowę.
Konieczność zgłoszenia	Wymagana koncesja URE w ramach prowadzenia działalności gospodarczej. Istnieje możliwość przyłączenia się do sieci w celu odsprzedaży nadwyżki wytworzonej energii. Przed podłączeniem do sieci należy zawiadomić operatora.	Obowiązkowe zgłoszenie w przypadku przekroczenia 3 metrów wysokości instalacji

Źródło: Opracowanie własne na podstawie ustawy: Dz. U. z 2018 r. poz. 1269

W ustawie o odnawialnych źródłach energii (OZE) zdefiniowane są następujące typy instalacji odnawialnych źródeł energii:

- mikroinstalacja – o Łącznej mocy zainstalowanej elektrycznej nie większej niż 50kW. To wielkość instalacji, gdzie nie jest wymagane pozwolenie na budowę czy prowadzenie działalności gospodarczej,
- mała instalacja – od 50kW do 500kW. Wymagane jest prowadzenie działalności i pozwolenie na budowę,
- duża instalacja – powyżej 500kW. Oprócz pozwolenia na budowę i prowadzenia działalności, wymagana jest również koncesja.

Ogniwa fotowoltaiczne mogą być stosowane w trzech segmentach rynku:

- mikroinstalacje PV do 10 kW na budynkach mieszkalnych,
- małe i średnie instalacje PV (10-100kW) na budynkach przemysłowych,

- duże systemy naziemne powyżej 100kW.

Ministerstwo Energetyki zaplanowało zakontraktowanie 750 MW dla elektrowni fotowoltaicznych o mocach do 1 MW.

Za sprawą ustawy zmieniają się również zasady rozliczeń podatkowych w systemie prosumentkim. Czynności związane ze wprowadzeniem i poborem energii będą opodatkowane VAT. Będą podlegały pod odpłatne dostawy towarów. Zwolnione z płacenia podatku VAT są jednostki, których wartość sprzedaży z poprzedniego roku nie przekracza 200 000 zł netto. Ustawa wprowadziła również obowiązek konsultacji projektów budowlanych z ochroną przeciwpożarową. Zmiana ta może znaczco zwiększyć koszty oraz zwolnić przyrost mikroinstalacji systemów PV.

1.2 Źródła danych o odnawialnych źródłach energii w Polsce

1.2.1 Energia odnawialna w polskiej statystyce publicznej

Raport GUS ([2016](#)) definiuje podstawowe pojęcia z zakresu odnawialnych źródeł energii.

Energia ze źródeł odnawialnych określona jest jako *energia uzyskiwana z naturalnych, stale powtarzających się procesów*. Pojęciem „energia ze źródeł odnawialnych” – dla celów statystycznych – obejmowane są niekopalne, odnawialne źródła energii, a w szczególności:

- energia wiatru,
- promieniowania słonecznego,
- geotermalna,
- fal, prądów i pływów morskich,
- spadku rzek oraz
- pozyskiwana z biomasy, biogazu wysypiskowego także biogazu powstałego w procesach odprowadzania lub oczyszczania ścieków albo rozkładu odpadów roślinnych i zwierzęcych,
- a także energia otoczenia (środowiska naturalnego) wykorzystywana przez pompy ciepła.

Według GUSu energia słoneczna przetworzona na energię elektryczną w ogniwach fotowoltaicznych to energia promieniowania słonecznego przetwarzana bezpośrednio na energię elektryczną przy użyciu ogniw fotowoltaicznych, zwykle zbudowanych z materiałów półprzewodnikowych, w których pod wpływem światła dochodzi do wytwarzania energii elektrycznej.

Energia słoneczna przetworzona na ciepło w kolektorach słonecznych definiowana jest jako energia promieniowania słonecznego przetwarzana, z zastosowaniem kolektorów słonecznych, na ciepło użytkowe wykorzystywane do: (1) ogrzewania wody użytkowej, (2) wody w basenach kąpielowych, (3) ogrzewania pomieszczeń oraz (4) w procesach suszarniczych i chemicznych, a także w cieplnych elektrowniach słonecznych.

Raport GUS ([2016](#)) stanowi opis jednolitych zasad metodycznych zbierania i prezentowania danych o OZE. Podstawowym źródłem danych w badaniach OZE są sprawozdania i kwestionariusze przekazywane w ramach badań statystycznych – zawarte w rocznych programach badań statystyki publicznej, ustanowionych przez Radę Ministrów w drodze rozporządzenia.

Prezes Głównego Urzędu Statystycznego, Minister do spraw energii oraz Prezes Urzędu Regulacji Energetyki prowadzi badania z obszaru „Rynek materiałowy i paliwowo-energetyczny”. Informacje na temat odnawialnych źródeł energii gromadzone są przy użyciu następujących formularzy sprawozdawczych i kwestionariuszy³:

- Formularz G-02o – Sprawozdanie o ciepłe ze źródeł odnawialnych,
- Formularz G-02b – Sprawozdanie bilansowe nośników energii i infrastruktury ciepłowniczej,
- Formularz G-03 – Sprawozdanie o zużyciu paliw i energii,
- Ankieta E-GD o zużyciu paliw i energii w gospodarstwach domowych.

Zebrane dane statystyczne dotyczą przede wszystkim wielkości podaży, zużycia poszczególnych nośników energii, stanu ich zapasów oraz infrastruktury technicznej do produkcji i magazynowania tych nośników energii. Statystyka publiczna gromadzi również informacje na temat konwersji energii i informacje niezbędne do oceny efektywności wykorzystania energii w procesach gospodarczych. Pozyskane dane służą do obliczania udziału energii ze źródeł odnawialnych w zużyciu energii ze wszystkich źródeł. Do obliczenia poszczególnych kategorii/wskaźni-

³ Na podstawie dokumentu z konferencji w Ministerstwie Gospodarki: Ciepło z odnawialnych źródeł energii w ujęciu statystycznym – sposób zbierania informacji oraz najnowsze dane

ków wykorzystywane jest, ujednolicony na poziomie międzynarodowym, narzędzie o nazwie SHort Assessment of Renewable Energy Sources (SHARES).

1.2.2 Urząd Regulacji Energetyki

Urząd Regulacji Energetyki inaczej URE jest centralnym organem administracji rządowej regulującą polski rynek energii. Powołany został na mocy ustawy z 10 kwietnia 1997 r. – Prawo energetyczne. Do głównych zadań urzędu należy:

- tworzenie warunków do rozwoju konkurencji i przeciwdziałania tworzeniu się monopolii,
- zapewnienie bezpieczeństwa energetycznego,
- działanie mające na celu zmniejszenie strat energii,
- wydawanie koncesji.

Rodzaj instalacji OZE	Sumaryczna moc zainstalowana [MW]	Liczba instalacji
Instalacje wykorzystujące biogaz	232,07	303
Instalacje wykorzystujące biomasę	1081,57	20
Instalacje wykorzystujące energię promieniowania słonecznego	146,892	671
Instalacje wykorzystujące energię wiatru	5843,07	1198
Instalacje wykorzystujące hydroenergię	252,75	749
Łącznie	7556,35	2941

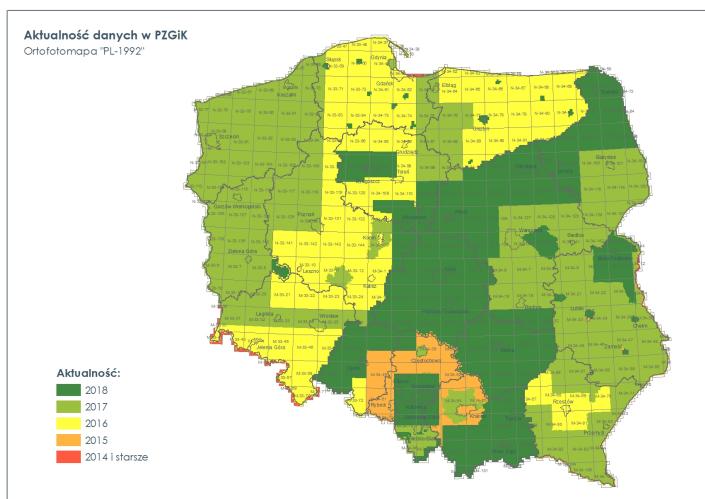
Źródło: Opracowanie własne na podstawie: <https://www.ure.gov.pl/pl/oze/potencjal-krajowy-oze/8108,Instalacje-odnawialnych-zrodel-energii-wg-stanu-na-dzien-31-grudnia-2018.html>

URE jest organem prowadzącym rejestr wytwórców energii z małych instalacji tzn. instalacji większej niż 50kW i nie większej niż 500kW. Kwestie związane ze składaniem wniosków, wpisem do Rejestru oraz wykreślenie z Rejestru zostały ujęte w art. 7-16 i 18 ustawy OZE. Rejestr podmiotów wytwarzających energię jest jawny i dostępny w Biuletynie Informacji Publicznej URE. Poniższa tabela przedstawia instalacje OZE wpisane do rejestru wytwórców energii w małej instalacji w roku 2017.

W 2017 roku wydano łącznie 90 zaświadczeń odnośnie wpisu, zmiany wpisu lub wykreślenia z Rejestru.

1.2.3 Zdjęcia lotnicze – ortofotomapy

Historia rozwoju ortofotomap obejmuje niewiele ponad sześćdziesiąt lat. Z początku reprodukowane były z wykorzystaniem technik fotograficznych i drukarskich. W ostatnich latach dostęp poprzez płyty CD został zastąpiony dostępem zdalnym. Dzisiaj dostęp do ortofotomap na portalach geodezyjnych jest tak powszechny jak do map topograficznych. Cyfrowe ortofotomapy udostępniane są w postaci w formacie geotiff.



Rysunek 1.3. Aktualność ortofotomapy w Państwowym Zasobie Geodezyjnym i Kartograficznym

Źródło: <http://www.codgik.gov.pl/index.php/zasob/ortofotomapa.html>

W ramach pracy dyplomowej zostały pozyskane ortofotomapy z GEOPOZu na podstawie "Wniosku o udostępnienie materiałów powiatowego zasobu geodezyjnego i kartograficznego" w celu prowadzenia badań naukowych/prac rozwojowych w myśl ustawy Prawo geodezyjne i kartograficzne (art. 40a ust. 2 ustawy PGiK). Zarząd Geodezji i Katastru Miejskiego GEOPOZ został utworzony z dniem 01 lipca 1991 roku na mocy Uchwały Rady Miejskiej Poznania z dnia 25 czerwca 1991 roku. Do głównych zadań GEOPOZu należy:

- organizacja i prowadzenie katastru miejskiego,
- realizowanie zadań służby geodezyjnej miasta,
- prowadzenie gospodarki gruntami i nieruchomościami na terenie Poznania.

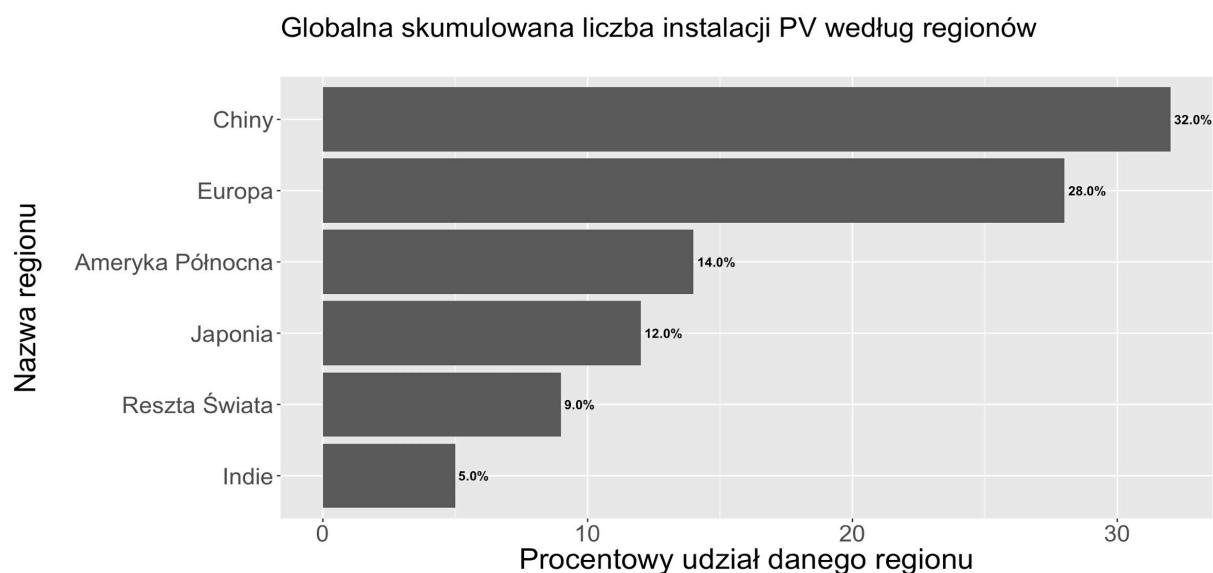
Najstarsza ortofotomapa dostępna na portalu sip.geopoz.pl pochodzi z 1997 roku. Dopiero w 2012 roku rozpoczęto coroczne tworzenie ortofotomap. Zaś 2013 roku utworzona została

ortofotomapa termalna, mająca na celu szybkie rejestrowanie stanu cieplnego obszarze całej aglomeracji.

1.3 Wykorzystanie energii Słońca na świecie i w Polsce

1.3.1 Świat

Globalne zapotrzebowanie na energię słoneczną rośnie z roku na rok. Jak wynika z raportu EurObserv'ER (2018a) w 2017 roku globalny popyt na energię słoneczną wzrósł o prawie 30%, co odpowiada 99 GW nowych systemów fotowoltaicznych. Jeśli chodzi o globalną moc instalacji fotowoltaicznych, pod koniec 2017 r. przekroczyła ona 405 GW.



Rysunek 1.4. Rozkład zainstalowanych systemów fotowoltaicznych według regionów

Źródło: Opracowanie własne na podstawie Fraunhofer ISE

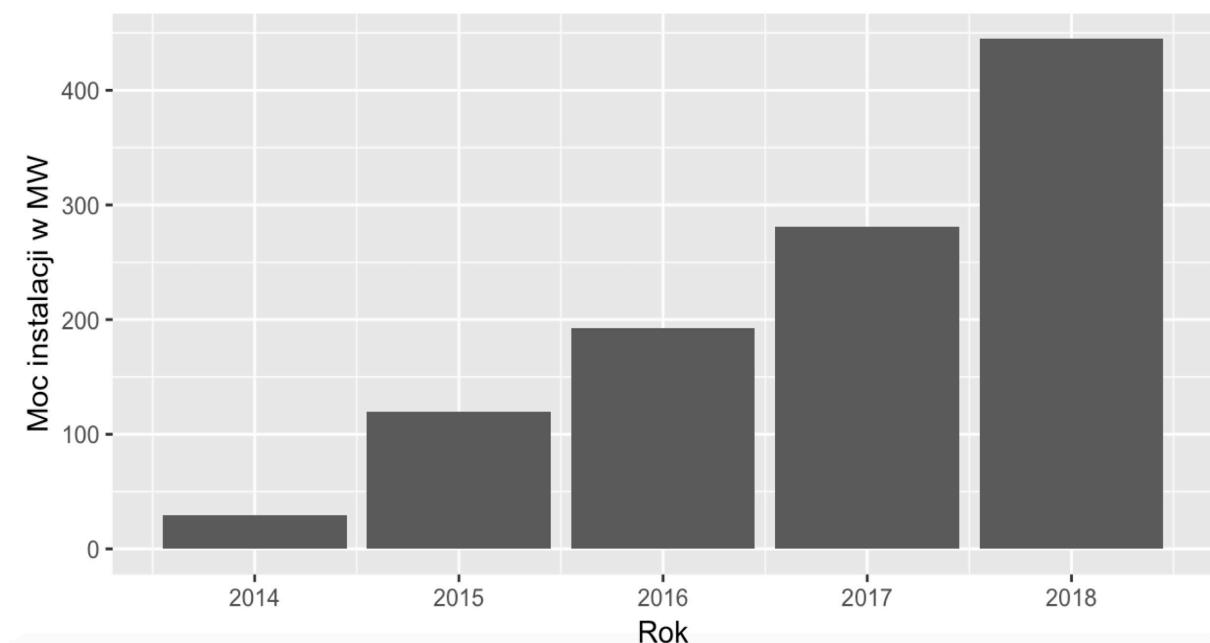
Europa wykazała jednak odmienny trend. Kraje europejskie zainstalowały w 2017 roku około 5562 MW nowych systemów fotowoltaicznych. To o 10,8% mniej niż w 2016 roku. Europejski rynek energetyki słonecznej pod koniec 2017 roku liczył już około 106,6 GW instalacji. Największym producentem energii słonecznej były w 2017 roku Chiny. Krajowa Administracja Energetyczna (NEA) podała na początku 2018 r. dane liczbowe, z których wynika, że w 2017 r. Chiny przyłączyły 52,9 GW nowych mocy, w porównaniu z 34,5 GW w 2016 r. (wzrost o 53%). Moc wszystkich instalacji fotowoltaicznych przyłączonych do sieci wynosiła w Chinach 130,2 GW. W 2017 r. Chiny zainstalowały o 10 GW więcej mocy niż skumulowana moc instalacji w Niemczech. Rynek amerykański skurczył się w 2017 r. zgodnie z oczekiwaniami. Dane opu-

blikowane przez *Solar Energy Industries Association* (SEIA), opracowane we współpracy z *GTM Research*, pokazują, że w 2017 r. podłączono w Stanach 10 608 MW mocy w porównaniu z 15 128 MW w 2016 r., co oznacza spadek o około 30%. W 2016 r. nastąpił szczyt spowodowany masowym napływem projektów ubiegających się o 30% federalną ulgę podatkową przed jej wygaśnięciem pod koniec roku. Nowa polityka federalna rządu USA nie sprzyja energii słonecznej. Na początek 2018 roku prezydent Trump ogłosił wprowadzenie taryfy ad valorem (podatek wyrażony jako procent towarów wwozonych do kraju), nakładanej w wysokości 30% na importowane ogniwa i moduły z krystalicznego krzemu. Ze względu na podwyższenie cen ogniwa GTM Research obniżyło swoje prognozy instalacyjne na lata 2018-2022 o około 13%.

1.3.2 Polska

Rok 2017 nie był korzystny z punktu widzenia energii pozyskiwanej z PV. Z danych URE (Urząd Regulacji Energetyki) wynika, że w porównaniu do 2016 roku przyrost mocy z promieni słonecznych był niewielki i wyniósł 8,65 MW. Dane nie uwzględniają mikroinstalacji.

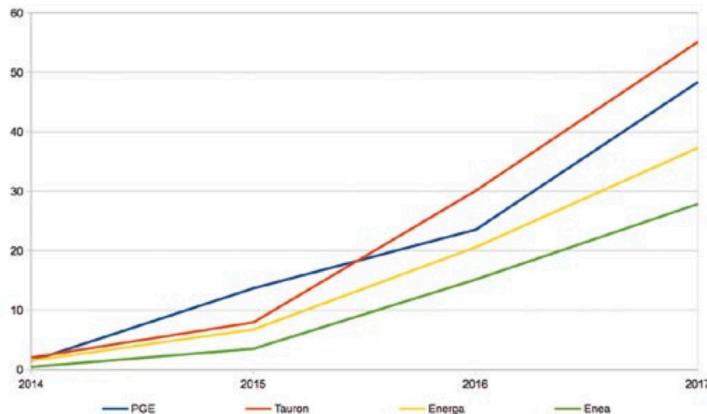
Według raportu Fotowoltaiki (2018) do końca 2017 roku w Polsce zainstalowano 29 197 systemów fotowoltaicznych o łącznej mocy 281,4 MW. Wzrost w 2017 roku w porównaniu do poprzedniego roku istotnie zmalał co mogło być spowodowane chęcią uzyskania przez inwestorów zielonych certyfikatów, których wydawanie zakończyło się w połowie 2016 roku.



Rysunek 1.5. Całkowita moc instalacji fotowoltaicznych w Polsce

Źródło: Opracowanie własne na podstawie raportu "Rynek fotowoltaiki w Polsce – 2017 r."

Przeciwne kształtuje się trend mikroinstalacji fotowoltaicznych przyłączonych do sieci przez Operatorów Systemów Dystrybucyjnych (OSD). Według danych czterech największych OSD⁴ moc przyłączonych instalacji do sieci prosumenckich wyniosła 81 MW czyli wzrost o 22MW w porównaniu do 2016 roku.



Rysunek 1.6. Skumulowana moc zainstalowana dla czterech największych OSD w MW
Źródło: Raport "Rynek fotowoltaiki w Polsce – 2017 r." Polskiego Towarzystwa Fotowoltaiki

Dokładniejsza analiza nie jest możliwa bez szczegółowych danych od OSD, których zwolniono z przekazywania sprawozdań. Przekroczenie mocy 280 MW z instalacji z koncesją URE i mikroinstalacji stanowi z jednej strony znaczący postęp w rozwoju rynku ogniw fotowoltaicznych, zwłaszcza w porównaniu z kilkoma megawatami na koniec 2013 roku. Z drugiej strony, to zdecydowanie za mało jak na państwo o takiej powierzchni i potencjale nasłonecznienia.

1.4 Podsumowanie

Aktualne statystyki dotyczące energii słonecznej opierają się na danych pochodzących z badań częściowych oraz sprawozdań prowadzonych przez GUS. Obecna metodologia charakteryzuje się szacunkową niepewnością i dostarcza jedynie dane krajowe w ujęciu rocznym, podczas gdy proces przechodzenia na energię odnawialną stwarza zapotrzebowanie na informacje na poziomie regionalnym o krótszym przedziale czasowym. W związku z tym ważne jest zapewnienie pełnego i szczegółowego zestawienia aktualnej ilości instalacji paneli słonecznych. Dzięki technologii głębokiego uczenia oraz dostępu do coraz dokładniejszych zdjęć satelitarnych, możliwe jest automatyczne wykrywanie paneli słonecznych.

⁴Mowa o PGE, Tauron, Energa i Enea.

Rozdział 2

Teoretyczne podstawy głębokich sieci neuronowych

2.1 Sztuczna inteligencja, uczenie maszynowe i uczenie głębokie

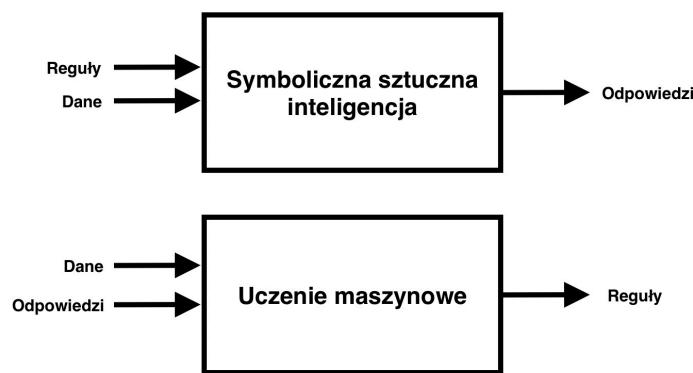
2.1.1 Wprowadzenie

Idea sztucznej inteligencji (SI) narodziła się w latach 50-tych XX wieku, kiedy to garstka pionierów z rodzącej się dziedziny informatyki zaczęła zadawać sobie pytanie, czy komputery można zmusić do "myślenia". Chollet (2018) podaje następującą definicję sztucznej inteligencji: żabieg mający na celu automatyzację procesu myślowego standardowo wykonywanego przez ludzi".

Tak zdefiniowana sztuczna inteligencja jest dziedziną, na którą nie tylko składają się uczenie maszynowe i uczenie głębokie, ale również techniki, które nie zawierają uczenia. Po opracowaniu koncepcji SI uważano, że jedynym sposobem na osiągnięcie sztucznej inteligencji na poziomie ludzkiego umysłu jest utworzenie wystarczająco dużej liczby reguł przetwarzania informacji. Powyższe podejście nazywane jest mianem symbolicznej sztucznej inteligencji i dominowało od lat 50tych do końca lat 80tych ubiegłego wieku. Pod względem zastosowania, podejście symbolicznej sztucznej inteligencji najlepiej sprawdza się w przypadku statycznych czy dobrze zdefiniowanych problemów. Przykładem wykorzystania powyższej koncepcji jest IBM Deep Blue, który w 1997 roku pokonał mistrza szachów Kasparowa.

W paradygmacie symbolicznej sztucznej inteligencji człowiek wprowadza konkretne reguły oraz dane, które mają zostać przetworzone zgodnie z tymi regułami, po czym oczekuje na uzy-

skanie danych wyjściowych. Przy uczeniu maszynowym człowiek wprowadza dane i oczekiwane odpowiedzi, a maszyna ma stworzyć takie reguły, aby z danych uzyskano odpowiedzi.



Rysunek 2.1. Uczenie maszynowe: nowy model programowania

Źródło: Opracowanie własne na podstawie Deep Learning with R, François Chollet, J. J. Allaire, 2018, s. 9

Model uczenia maszynowego jest trenowany poprzez dostarczenie wielu przykładów wyników. Celem modelu jest ustalenie reguły, która umożliwiłaby automatyzację procesu. Uczenie maszynowe zaczęło się rozwijać w latach 90tych XX w., szybko stało się najpopularniejszą dziedziną sztucznej inteligencji. Rozwój napędzany był coraz większym dostępem do szybkiego sprzętu komputerowego oraz dużych zbiorów danych.



Rysunek 2.2. Sztuczna inteligencja, uczenie maszynowe i uczenie głębokie

Źródło: Opracowanie własne na podstawie Deep Learning with R, François Chollet, J. J. Allaire, 2018, s. 8

Głębokie uczenie (ang. *deep learning*) jest podzioborem uczenia maszynowego, który skupia się na uczeniu kolejnych warstw reprezentacji danych. Dane w modelu głębokiego uczenia muszą przejść przez kilka warstw przetwarzania, dlatego zdecydowano się używać określenia „głębokie” uczenie. W uczeniu głębokim wielowarstwowe reprezentacje są uczone przy wykorzy-

staniu modeli zwanych sieciami neuronowymi. Termin sieci neuronowej jest inspirowany biologicznymi sieciami neuronowymi, które tworzą mózg człowieka. Ze względu na wysokość złożoności pracy mózgu, odzwierciedlana jest jedynie organizacja działania w sztucznych sieciach neuronowych. Pomimo uproszczonych modeli względem mózgu, wyniki sztucznych sieci neuronowych bywają zaskakujące. Podstawowe elementy sieci – neurony – oddziałując na siebie w sposób nieliniowy są w stanie uogólniać wiedzę oraz wykrywać zależność między zmiennymi. Jak podaje Tadeusiewicz (1998) proces doszkalania w sztucznych sieciach neuronowych przypomina biologiczne procesy nabywania wiedzy. Biologiczne powiązania mają również konwolucyjne sieci neuronowe (Convolutional Neural Networks, CNN), które są klasą głębokich sieci neuronowych, najczęściej stosowanych do analizy obrazów wizualnych. Prace Hubela i Wiesela (Hubel & Wiesel, 1959) z lat 50-tych i 60-tych pokazały, że kora wzrokowa kotów i małp zawiera neurony, które indywidualnie reagują na małe obszary pola widzenia. Kunihiro Fukushima za-inspirowany wyżej wspomianą pracą Hubela i Wiesla zaimplementował pierwszą konwolucyjną sieć neuronową, którą nazwał „Neokognitron” (Fukushima, 1980).

2.1.2 Dotychczasowe osiągnięcia uczenia głębokiego

Chociaż głębokie uczenie jest całkiem starą dziedziną uczenia maszynowego, dopiero na początku 2010 r. zyskało ono na popularności. W 2011 roku pracownik naukowy Università della Svizzera italiana Dan Ciresan dzięki modelowi głębokiego uczenia, wytrenowanemu na układzie graficznym GPU, zaczął wygrywać akademickie konkursy klasyfikacji obrazów. Za przełomowe wydarzenie uczenia głębokiego uznaje się wejście grupy w składzie Alex Krizhevsky, Ilya Sutskever i Geoffrey Hinton do konkursu ImageNet¹ w 2012 roku. Celem była klasyfikacja wysokiej rozdzielczości obrazów na 1000 kategorii po wcześniejszym wytrenowaniu modelu na zbiorze 1,4 miliona zdjęć. W 2011 roku najwyższa skuteczność wynosiła 74,3%. W 2012 roku zespół dowodzony przez Alexa Krizhevsky'ego uzyskał dokładność na poziomie 83,6%. W kolejnych latach konkurs zdobiły konwolucyjne sieci neuronowe, dzięki którym w 2015 roku uzyskano skuteczność na poziomie 96,4%. Wysoka skuteczność oraz łatwość trenowania na dużych zbiorach danych przekonały Europejską Organizację Badań Jądrowych (CERN), aby zastąpić algorytmy wektorów nośnych i drzew decyzyjnych, głębokimi sieciami neuronowymi opartymi na pakietie Keras².

¹Konkurs, który odbywa się co roku, gdzie założeniem konkursu jest oszacowanie zawartości zdjęć do celów wyszukiwania i automatycznej adnotacji z wykorzystaniem dużych zbiorów danych

²Deep Learning with R, François Chollet, J. J. Allaire, 2018, s. 24

Uczenie głębokie przyczyniło się do rozwoju w następujących obszarach:

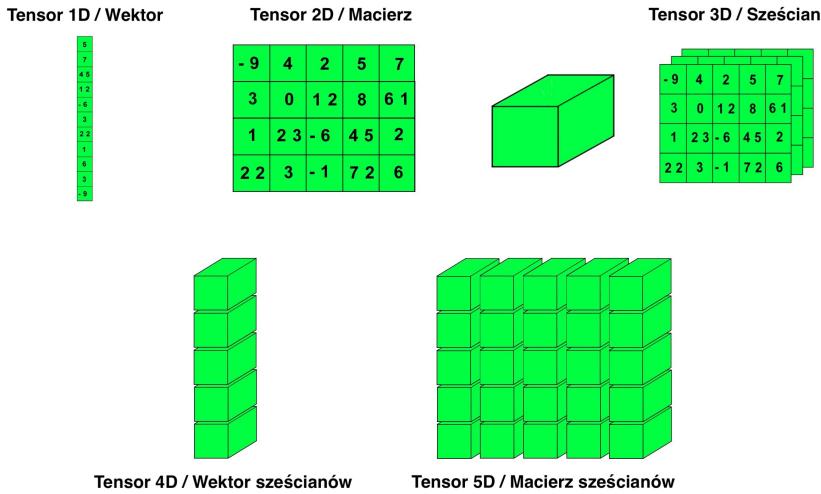
- rozpoznawania mowy,
- poprawy tłumaczenia maszynowego,
- osobistych asystentów głosowych takich jak: Google Now czy Amazon Alexa,
- czytania z ruchu warg,
- pojazdów autonomiczne.

2.2 Anatomia sieci neuronowej

2.2.1 Reprezentacja danych sieci neuronowych – skalary, wektory, macierze, tensorы

Dane wykorzystywane w sieciach neuronowych są przechowywane w wielowymiarowych tablicach nazywanych tensorami. Wszystkie współczesne systemy sztucznej inteligencji wykorzystują tensorы jako podstawową strukturę danych, która służy do przechowywania liczb. Znaczenie tensorów jest tak ważne, że firma Google postanowiła nadać nazwę Tensorflow swojej bibliotece uczenia maszynowego. Tensor stanowi główną strukturę danych do przechowywania liczb. Macierz jest przykładem dwuwymiarowych tensorów. Zaś skalarem jest tensor zero-wymiarowy. Tensor posiadający jedną oś jest również wektorem. W przypadku umieszczenia macierzy w tablicy otrzymamy tensor trójwymiarowy. W wyniku wstawienia kilku trójwymiarowych tensorów powstanie tensor czterowymiarowy. W zagadnieniu uczenia głębokiego korzysta się z tensorów mających od 0 do 4 wymiarów. Do głównych atrybutów tensora należą:

- liczba osi - tensor czterowymiarowy ma cztery osie,
- kształt - wektor wartości całkowitoliczbowych, które opisują, ile wymiarów ma tensor wzdłuż każdej osi,
- typ danych - rodzaj danych zawartych w tensorze.



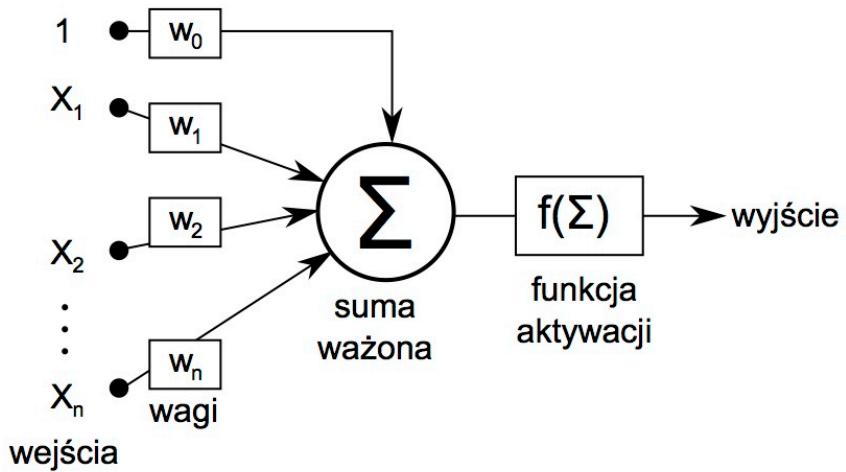
Rysunek 2.3. Typy tensorów

Źródło: Opracowanie własne na podstawie <https://hackernoon.com/learning-ai-if-you-suck-at-math-p4-tensors-illustrated-with-cats-27f0002c9b32>

2.2.2 Sztuczne sieci neuronowe

W 1943 roku, w badaniach nad sztucznymi sieciami neuronowymi, McCulloch i Pitts opracowali pierwszy formalny model sztucznego neuronu (McCulloch & Pitts, 1943). Koncepcja przetrwała do dzisiaj i nadal stanowi podstawowe ogniwo w modelach głębokiego uczenia.

Model przetwarza informacje wejściowe, a wynikiem są informacje wyjściowe. Sztuczna sieć neuronowa zbudowana jest z licznych neuronów, które są ze sobą połączone. Każde wejście ma przypisaną wagę, której wielkość zależy od tego jak ważna jest informacja w stosunku do innych wejść. Rysunek 2.4 przedstawia budowę sztucznego neuronu oraz koncepcję działania. Sygnały wejścia (x_1, x_2, \dots, x_n) mnożone są przez konkretne wagi (w_1, w_2, \dots, w_n). Wyniki sumowane są w tzw. sumatorze, a następnie przekazywane do funkcji aktywacji, gdzie ostatecznie podlegają obliczeniom. Końcową informacją jest tzw. sygnał wyjściowy.



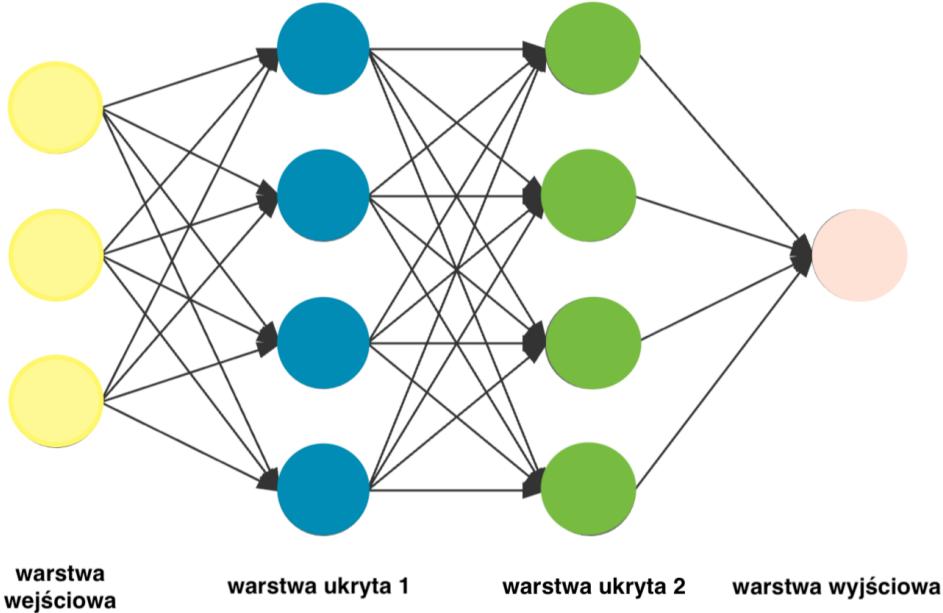
Rysunek 2.4. Budowa neuronu

Źródło: https://pl.wikipedia.org/wiki/Neuron_McCullocha-Pittsa

Podstawowym budulcem sieci neuronowych jest warstwa – moduł przetwarzania danych, którą można traktować jako filtr danych. Warstwa jest "kontenerem", który zazwyczaj otrzymuje ważone dane wejściowe, przetwarza je za pomocą zestawu głównie nielinowych funkcji, a następnie przekazuje te wartości jako dane wyjściowe do następnej warstwy. Głębokie uczenie polega zatem na łączeniu razem prostych warstw, które sekwencyjnie filtrują dane. Istnieją trzy podstawowe rodzaje warstw:

- warstwę wejściową – dostarcza informacji ze świata zewnętrznego do sieci. W warstwach wejściowych nie są wykonywane żadne obliczenia – po prostu przekazują one informacje do ukrytych węzłów,
- warstwy ukryte – wykonują one obliczenia i przesyłają informacje z węzłów wejściowych do węzłów wyjściowych,
- warstwy wyjściowe - odpowiedzialne za obliczenia i przesyłanie informacji z sieci do świata zewnętrznego.

Sieć neuronowa musi składać się z co najmniej jednej warstwy wejściowej i warstwy wyjściowej. Warstw ukrytych może być kilka. Każdy węzeł w jednej warstwie jest połączony z każdym innym węzłem w następnej warstwie. Pogłębiamy sieć, zwiększając liczbę ukrytych warstw.



Rysunek 2.5. Sieć neuronowa czterowarstwowa o strukturze 3-4-4-1

Źródło: Opracowanie własne.

Aby sieć była gotowa do przeszkolenia, musimy wybrać jeszcze trzy rzeczy, jako część etapu komplikacji:

- Funkcja straty (funkcja celu) – określa sposób pomiaru dokładności sieci w procesie przetwarzania treningowego zbioru danych, a tym samym pozwala na dostrajanie parametrów sieci,
- Optymalizator – mechanizm, za pomocą którego sieć będzie się aktualizować na podstawie danych, które widzi i funkcji straty,
- Metryki jakości – funkcje, które służą do oceny jakości modelu podczas trenowania i testowania.

Funkcja straty to wartość, która jest minimalizowana w trakcie trenowania. Przykładowo, w regresji liniowej minimalizujemy funkcję straty określonej przez sumę kwadratów reszt, a w metodzie największej wiarygodności zwykle minimalizujemy określoną funkcję wiarygodności. Sieć neuronowa może posiadać więcej niż jedną funkcję straty, kiedy generuje kilka wartości wyjściowych. Jednak sam proces spadku gradientu musi być oparty na jednej wartości, dlatego skalary z poszczególnych funkcji straty są łączone, a następnie liczona jest średnia. Dobranie odpowiedniej funkcji straty jest kluczowe. W przypadku ustalenia funkcji, która nie bę-

dzie skorelowana z zamierzonym celem, to sieć będzie liczyła niechciane współczynniki. Sam proces spadku gradientu oparty jest na jednej wartości celu.

Optymalizacja to proces poszukiwania parametrów minimalizujących lub maksymalizujących nasze funkcje. W przypadku zagadnień związanych z uczeniem głębokim optymalizowana jest funkcja straty. Wraz z minimalizowaniem jej, ulepszane są parametry modelu. Wybór optymalizatora jest zazwyczaj związany z konkretnym problemem, który planujemy rozwiązać. Znalezienie minimalnej wartości straty nie jest proste i dlatego stosowane są optymalizatory. Największą przeszkodą dla znalezienia globalnego minimum są lokalne minima. Pozbycie się tzw. punktów siodłowych jest często uważane za jeszcze większe wyzwanie. Są to płaskowiąże, gdzie wartość funkcji straty jest prawie stała. W tych punktach gradient jest prawie zerowany we wszystkich kierunkach, co uniemożliwia ucieczkę.

2.2.3 Funkcja aktywacji

Funkcje aktywacji są niezwykle ważnym elementem głębokich sieci neuronowych. Funkcje te definiują wyjście neuronu, biorąc pod uwagę dane wejściowe. W zasadzie to one decydują, czy neuron powinien być aktywowany, czy też nie. Poniżej przedstawiono dwie z nich w kolejności chronologicznej, aby zauważać korzyści wynikające z wyboru aktualnych rozwiązań:

- **Funkcja liniowa**

$$f(x) = ax, \quad (2.1)$$

gdzie x to dane wejściowe i parametry modelu określone są przez a . Sieć neuronowa z funkcją liniowej aktywacji jest po prostu modelem regresji liniowej. Ma ograniczoną wydajność i zdolność do obsługi różnych parametrów danych wejściowych o różnym stopniu złożoności.

- **Funkcja sigmoid**

Funkcja sigmoid jest powszechnie w przypadku zmiennych o rozkładzie dwumianowym, ponieważ jej zakres wynosi $(0, 1)$, pozwala również na interpretacje w kontekście prawdopodobieństwa (np. prawdopodobieństwa wystąpienia określonego zdarzenia).

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (2.2)$$

Dzięki nieliniowości, funkcja potrafi uchwycić skomplikowane relacje. Duża liczba ujemna przechodząca przez funkcję sigmoid staje się 0, a duża liczba dodatnia 1. Ze względu na tę właściwość, funkcja sigmoid często ma łatwą interpretację związaną z nią jako wskaźnik wykluczania neuronów.

2.2.4 Wsteczna propagacja błędu

Propagacja wsteczna jest podstawowym algorytmem stojącym za tym, w jaki sposób sieci neuronowe uczą się. Metoda ta została opisana przez Werbosa (Werbos, 1974), a następnie dopracowana przez Rummelharta, Hintona i Williamsa (Rumelhart, Hinton & Williams, 1985). Algorytm rozpoczyna działanie w momencie obliczenia błędu średniokwadratowego między rzeczywistą, a przewidzianą wartością zgodnie ze wzorem

$$L(w) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2, \quad (2.3)$$

gdzie w oznacza zbiór wszystkich wag w sieci, n jest całkowitą liczbą danych wejściowych, a jest wektorem wyjść z sieci, gdy x jest wejściem, a suma jest nad wszystkimi wejściami treningowymi, x . Oczywiście, wyjście a zależy od x i w . Główną cechą wstecznej propagacji jest obliczenie pochodnej cząstkowej ($\frac{\partial L}{\partial w}$) funkcji straty L względem każdej wagi w w sieci. Wyrażenie to mówi nam, jak szybko zmienia się funkcja straty, gdy zmieniamy wagę.

Następnie, algorytm przechodzi wstecznie przez sieć i sprawdza, które wagę najbardziej przyczyniły się do wartości straty i próbuje znaleźć sposób, aby dostosować ją tak, aby strata zmniejszyła się. Ostatnim krokiem jest pobieranie wszystkich wag, a następnie aktualizacja tak, aby zmieniały się w przeciwnym kierunku do gradientu.

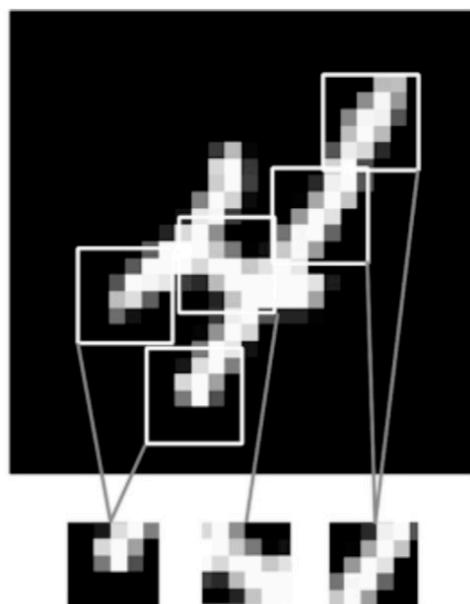
Do głównych zalet algorytmu wstecznej propagacji zalicza się:

- prostota zrozumienia,
- łatwa implementacja,
- pewne znalezienie rozwiązania.

Wsteczna propagacja błędu nie jest doskonałym algorytmem. W trakcie procesu uczenia algorytm często dąży do lokalnego minimum. Przy złym wyborze początkowych wag wsteczna propagacja może wejść w minimum lokalne. Najlepszym rozwiązaniem jest taka modyfikacja algorytmu, aby przyspieszyć proces trenowania oraz uniknięcie lokalnych minimów.

2.3 Konwolucyjne sieci neuronowe

Architektura konwolucyjnych sieci neuronowych jest zbliżona do struktury połączeń neuronów w ludzkim mózgu i została zainspirowana działaniem kory wzrokowej. Fundamentalna różnica między warstwą gęstych powiązań, a warstwą konwolucyjną polega na tym, że warstwy gęste uczą się globalnych wzorców w przestrzeni cech wejściowych, podczas gdy warstwy konwolucyjne uczą się lokalnych wzorców (patrz 2.6): w przypadku obrazów, wzorce występują w dwuwymiarowych oknach.



Rysunek 2.6. Przykład lokalnych wzorców uczenia konwolucyjnej sieci neuronowej

Źródło: Deep Learning with R, François Chollet, J. J. Allaire, 2018, s. 139

Dzięki tej ceście konwolucyjne sieci neuronowe mają dwie ważne właściwości:

- Rozpoznawanie danego wzorca jest niezależne od jego położenia. Po nauce pewnego wzorca w lewym górnym rogu obrazu, sieć może rozpoznać go wszędzie: na przykład w prawym dolnym rogu. Gęsto połączona sieć musiałaby na nowo nauczyć się wzoru, gdyby pojawił się w nowym miejscu.
- Sieć konwolucyjna może uczyć się przestrzennej struktury wzorów. Pierwsza warstwa uczy się drobnych wzorów, takich jak krawędzie, druga warstwa konwolucyjna uczy się większych wzorów wykonanych z cech pierwszych warstw i tak dalej. Dzięki temu CNN

będą mogły efektywnie uczyć się coraz bardziej skomplikowanych i abstrakcyjnych koncepcji wizualnych.

2.3.0.1 Działanie konwolucyjnych sieci neuronowych

Komputer widzi obraz jako tablicę wartości pikseli. Każda z tych liczb ma wartość od 0 do 255, która opisuje intensywność pikseli w tym punkcie. W zależności od rozdzielczości i wielkości obrazu, zobaczy przykładowo tablicę liczb $32 \times 32 \times 3$ (3 odnosi się do RGB).

Pierwsza warstwa w CNN jest zawsze warstwą konwolucyjną. Teraz najlepszym sposobem wytlumaczenia warstwy konwolucyjnej jest wyobrażenie sobie latarki, która świeci nad lewym górnym rogiem obrazu. Powiedzmy, że światło, które świeci w tej latarce, obejmuje obszar 5×5 . Teraz wyobraźmy sobie, że ta latarka przesuwa się po wszystkich obszarach obrazu wejściowego. W kontekście uczenia się maszynowego latarka ta nazywana jest filtrem (lub jądrem), a obszar, nad którym świeci, nazywany jest polem odbiorczym. Filtr ten jest również tablicą liczb (liczby są nazywane wagami lub parametrami). Filtr przesuwa się wzduż obrazu wejściowego, pomnażając wartości w filtrze i oryginalne wartości pikseli obrazu. Wszystkie te iloczyny są zsumowane, a wynikiem jest pojedyncza wartość. Teraz powtarzamy ten proces dla każdej lokalizacji na danych wejściowych. Wynikiem przesunięcia filtrów po wszystkich lokalizacjach jest mapa cech (ang. *feature map*). Zalecane jest używanie kilku filtrów, każdy o innej budowie, aby wykrywać inne krawędzie czy krzywe. W miarę jak dane przechodzą przez sieć i coraz więcej warstw konwolucyjnych, wynikiem są mapy aktywacji, które reprezentują coraz bardziej złożone funkcje. Po zakończeniu trenowania w sieci mogą pojawić się filtry, które aktywują się, gdy na obrazie znajduje się np. pismo odręczne. Na koniec sieci, gdy filtry potrafią wykrywać zaawansowane cechy, dodawana jest warstwa gęsta (ang. *Fully Connected Layer*, FC). Warstwa ta pobiera dane wyjściowe z ostatniej warstwy konwolucyjnej i wyświetla wektor N wymiarowy, gdzie N jest liczbą klas, z których algorytm musi wybierać.

Co do zasady, warstwa FC sprawdza, jakie zaawansowane cechy są najsilniej skorelowane z daną klasą i ma określone wagę, dzięki czemu podczas obliczania iloczynu skalarnego pomiędzy wagami i poprzednią warstwą, uzyskuje się prawidłowe prawdopodobieństwa dla różnych klas.

2.3.1 Turi Create i TinyYolo

W ramach pracy dyplomowej został wykorzystany pakiet Turi Create. Turi Create to narzędzie, które upraszcza tworzenie własnych modeli uczenia maszynowego. W 2016 roku zostało zaku-

pione przez Apple, a pod koniec 2017 roku udostępniono pakiet na Githubie jako open-source. Do korzystania z Turi Create potrzebny jest język Python. Narzędzie zakupione przez Apple to re-implementacja algorytmu TinyYOLO (dokładnie YOLOv2 z siecią bazową Darknet (Redmon, 2016)), który domyślnie rozpoznaje 1000 klas obiektów.

Turi Create udostępnia następujące pakiety³:

Tabela 2.1. Funkcjonalność pakietu Turi Creat

Nazwa pakietu	Opis
Recommender	Personalizacja wyboru dla użytkowników
Image Classification	Oznaczanie obrazów
Drawing Classification	Rozpoznawanie rysunku ołówkiem/rysunków dotykowych i gestów
Sound Classification	Klasyfikacja dźwięków
Object Detection	Rozpoznawanie obiektów w obrazach
Style Transfer	Stylizacja obrazów
Activity Classification	Detekcja aktywności za pomocą czujników
Image Similarity	Wyszukiwanie podobnych obrazów
Classifiers	Przewidywanie etykiet
Regression	Przewidywanie wartości liczbowych
Clustering	Grupowanie podobnych obserwacji
Text Classifier	Analiza sentymentu wiadomości

Źródło: Opracowanie własne na podstawie dokumentacji Turi Create, <https://github.com/apple/turicreate>

2.3.1.1 TinyYolo

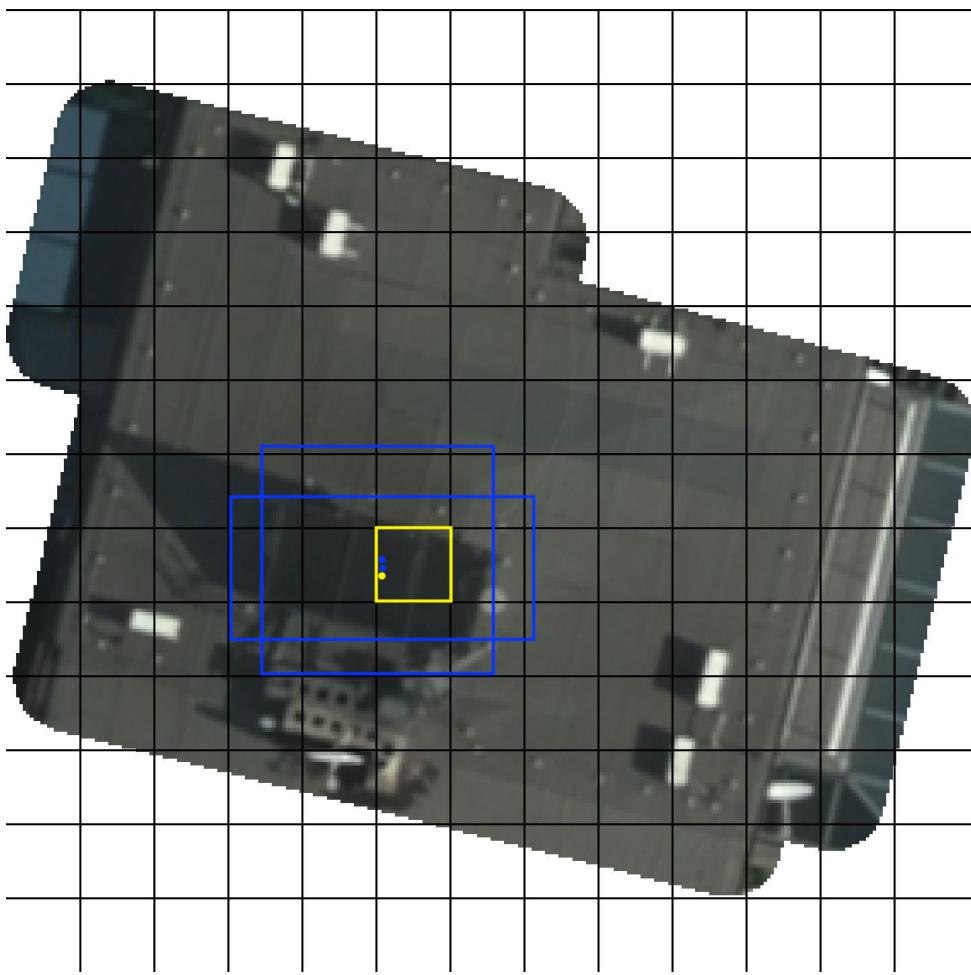
TinyYOLO (YOLO z ang. *You only look once*) jest systemem wykrywania obiektów. Architektura Yolo przypomina konwolucyjna sieć neuronową i przekazuje obraz o rozmiarze $m \times m$ raz przez CNN, a wyjściem jest $(m \times m)$ prawdopodobieństwo przynależności obiektu do określonej klasy (w przypadku problemu klasyfikacyjnego).

YOLO (taka nazwa będzie wykorzystywana w trakcie tego i kolejnego rozdziału) dzieli obraz wejściowy na siatkę o wymiarach $S \times S$. Każda komórka siatki przewiduje tylko jeden obiekt oraz stałą liczbę pól ograniczających (w przypadku Turi Create jest to 15 pól na komórkę). Na rysunku 2.7, żółta komórka siatki tworzy dwie predykcje ramki ograniczające (niebieskie ramki) w celu zlokalizowania miejsca, w którym znajduje się panel.

Dla każdej komórki siatki:

- przewidywane jest pole ograniczające B i każde pole ma jeden wynik *confidence* (po polsku pewność; używana będzie jednak fraza angielska ze względu na lepsze dopasowanie

³Link do repozytorium: <https://github.com/apple/turicreate>



Rysunek 2.7. Przykład działania algorytmu YOLO

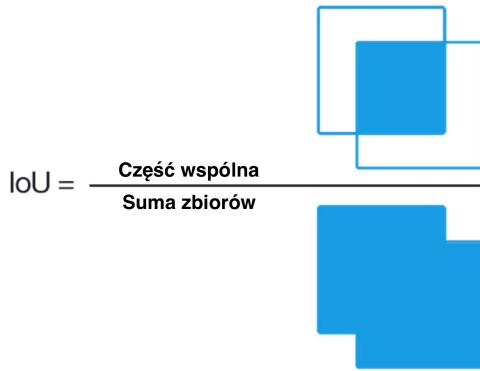
Źródło: Opracowanie własne

znaczeniowe, o którym poniżej),

- wykrywany tylko jeden obiekt niezależnie od liczby pól B ,
- przewidywane jest warunkowe prawdopodobieństwa klas C (po jednym na klasę dla prawdopodobieństwa klasy obiektu).

Każde pole ograniczające składa się z 5 elementów: x (wsp. środkowa), y (wsp. środkowa), w (szerokość pola ograniczającego), h (wysokość pola ograniczającego), i wskaźnika confidence dla prostokąta. Wskaźnik confidence odzwierciedla prawdopodobieństwo, że pole zawiera obiekt i dokładność pola ograniczającego. Formalnie definiujemy confidence jako $Pr_i(\text{obiekt} \in i) \times IOU_i$ (intersection over union), gdzie $Pr_i(\text{obiekt} \in i)$ jest prawdopodobieństwem, że i -ta komórka zawiera obiekt. Wzór na IOU prezentuje się następująco:

W liczniku obliczamy obszar nakładania się przewidywanego prostokąta ograniczającego i prostokąta, który został ręcznie nałożony na obiekt. Jako mianownik przyjmujemy obszar po-



Rysunek 2.8. Część wspólna przez sumę zbiorów – decyzja o wyborze określonej obwiedni (bounding box)

Źródło: Opracowanie własne.

łączenia przewidywanego pola ograniczającego, jak i polem ograniczającym z ręcznie nałożonym polem ograniczającym. Normalizowana jest szerokość w i wysokość h pola ograniczającego względem szerokości i wysokości obrazu.

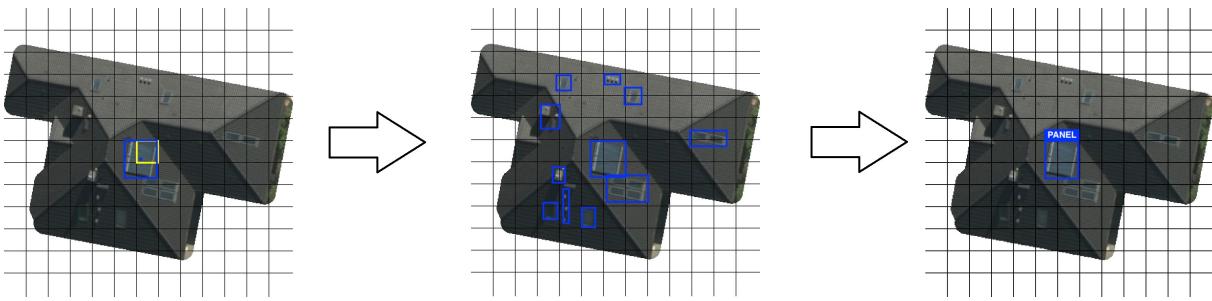
Główną koncepcją TinyYolo jest zbudowanie sieci CNN, aby przewidzieć tensor o rozmiarze $(13, 13, 90)$. Dwa pierwsze wymiary to liczba komórek jaką jest nakładana na zdjęcie. Ostatni wymiar tensora wyjściowego to 90, czyli liczba kanałów dla każdej komórki siatki. Te 90 numerów zawiera dane dla pól ograniczających i prawdopodobieństwa klasowe. Każda komórka siatki przewiduje 15 pól ograniczających, pole ograniczające opisane jest przez 5 elementów i jedną klasę $(15 \times (5 + 1))$.

YOLO wykonuje regresję liniową, wykorzystując dwie w pełni połączone ze sobą warstwy w celu uzyskania przewidywań pola ograniczającego $7 \times 7 \times 15$ (środkowy obrazek poniżej). Dla każdej komórki siatki przewidywany jest wektor \mathbf{Y} , który ma 90 elementów (15×6) .

$$\sum_{i=0}^{S^2} \mathbf{y} = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \end{bmatrix}. \quad (2.4)$$

Aby dokonać ostatecznego przewidywania, zachowujemy te z wysokimi confidence score prostokąta (większymi niż 0,5) jako nasze ostateczne przewidywania (zdjęcie po prawej).

Pojedyncza sieć konwolucyjna przewiduje jednocześnie wiele pól ograniczających i prawdopodobieństwa klasowe dla tych pól. Yolo dzieli obraz wejściowy na siatkę $S \times S$. Jeśli środek



Rysunek 2.9. Przykład działania algorytmu YOLO uwzględniający wiele możliwych lokalizacji paneli

Źródło: Opracowanie własne

obiektu wpada do komórki siatki, komórka ta jest odpowiedzialna za wykrycie tego obiektu. Każda komórka siatki przewiduje B pól ograniczających B i confidence score dla tych pól. Uzyskane confidence scores odzwierciedlają to, jak pewny jest model, że ramka zawiera przedmiot, a także to, jak dokładne jest ta ramka, którą przewiduje.

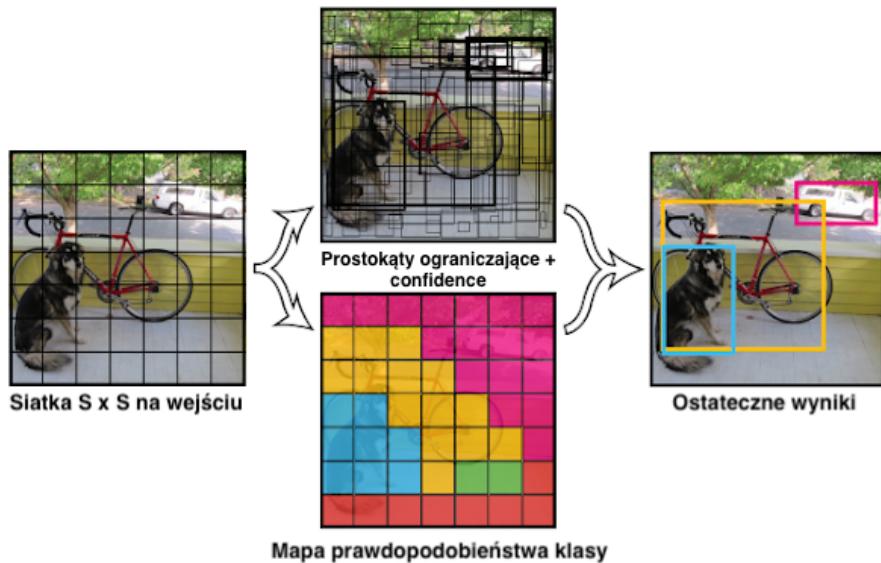
Prawdopodobieństwa te są zależne od komórek siatki zawierającej obiekt. Przewidujemy tylko jeden zestaw prawdopodobieństw klasy na komórkę siatki, niezależnie od liczby pól B . W czasie trwania testu mnożymy warunkowe prawdopodobieństwa klas i indywidualne przewidywania dotyczące confidence,

$$Pr(\text{Klasa}_i|\text{Obiekt}) \times Pr(\text{Obiekt}) \times \text{IOU} = Pr(\text{Klasa}_i) \times \text{IOU}, \quad (2.5)$$

gdzie $Pr(\text{Klasa}_i|\text{Obiekt})$ jest prawdopodobieństwem, że obiekt należy do klasy_i, jeśli obiekt jest obecny, $Pr(\text{Obiekt}) \times \text{IOU}$ jest prawdopodobieństwem zaznaczenia obiektu, zaś $Pr(\text{Klasa}_i) \times \text{IOU}$ jest prawdopodobieństwem znalezienia klasy. Wyniki te określają zarówno prawdopodobieństwo pojawienia się tej klasy w polu, jak i to, jak dobrze przewidywane pole pasuje do obiektu.

Jeżeli chodzi o architekturę sieci YOLO to posiada ona 24 warstwy konwolucyjne, po których następują 2 w pełni połączone warstwy.

W sieci optymalizowana jest wartość błędu średniokwadratowego na wyjściu naszego modelu. Używany jest błąd średniokwadratowy, ponieważ jest łatwy do zoptymalizowania, jednak nie jest idealnie zgodny z naszym celem maksymalizacji średniej precyzji. Waży on błąd lokalizacji w równym stopniu z błędem klasyfikacyjnym, który może nie być idealny. Również w każdym obrazie wiele komórek siatki nie zawiera żadnego obiektu. Powoduje to, że wyniki confidence tych komórek zbliżają się do zera, często przytłaczając gradient komórek, które zawierają obiekty. Może to prowadzić do niestabilności modelu, powodując wcześnie odchylenia



Rysunek 2.10. Przykład działania algorytmu YOLO uwzględniający rozpoznawanie wielu obiektów

Źródło: Opracowanie własne na podstawie
<https://towardsdatascience.com/yolo-you-only-look-once-17f9280a47b0>

treningu.

Aby temu zaradzić, zwiększa się strata wynikająca z przewidywań dotyczących współrzędnych pól ograniczających oraz zmniejsza się strata wynikająca z przewidywań dotyczących confidence w przypadku pól, które nie zawierają obiektów. Wykorzystujemy dwa parametry, λ_{coord} i $\lambda_{noob,j}$, aby to osiągnąć. λ są stałymi. λ_{coord} jest najwyższa, aby skupić się bardziej na wykrywaniu. Zaś $\lambda_{noob,j}$ obciąża wartość straty przy wykrywaniu tła.

TinyYolo przewiduje wiele pól ograniczających na komórkę siatki. Do obliczenia straty za prawdziwie pozytywne, potrzebujemy tylko jednego z nich, który będzie odpowiedzialny za obiekt.

W tym celu wybieramy pole ograniczające z najwyższym IoU. Strategia ta prowadzi do wyspecjalizowania się w przewidywaniach pola ograniczającego. Każda prognoza staje się coraz lepsza w przewidywaniu pewnych rozmiarów i proporcji kształtu.

TinyYolo wykorzystuje błąd średniokwadratowy między predykcjami, a ręcznie nałożonymi prostokątami do obliczenia straty. Funkcja straty składa się z sumy następujących elementów:

- straty klasyfikacyjnej (ang. *classification loss*, ClassL),
- straty lokalizacyjnej (ang. *location loss*, błędy między przewidywanym polem ograniczającym, a ręcznie nałożonym; LocL),

- straty pewności (ang. *confidence loss*; czy istnieje obiekt w polu ograniczającym; ConfL).

2.3.1.2 Strata klasyfikacyjna

Jeśli obiekt zostanie wykryty, strata klasyfikacji w każdej komórce jest kwadratowym błędem klasy prawdopodobieństw warunkowych dla każdej klasy:

$$\text{ClassL} = \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{klasy}} (p_i(c) - \hat{p}_i(c))^2, \quad (2.6)$$

gdzie $\sum_{i=0}^{S^2}$ oznacza sumę po wszystkich komórkach czyli 13×13 $1_i^{\text{obj}} = 1$ jeżeli obiekt występuje w komórce i , w innym wypadku 0. $\hat{p}_i(c)$ oznacza prognozowane warunkowe prawdopodobieństwo klasy dla klasy c w komórce i .

2.3.1.3 Strata lokalizacyjna

Strata wynikająca z lokalizacji mierzy błędy w przewidywanych lokalizacjach i rozmiarach pól ograniczających. Zliczamy jedynie pole odpowiedzialne za wykrycie obiektu.

$$\begin{aligned} \text{LocL} = & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right], \end{aligned} \quad (2.7)$$

gdzie $\sum_{i=0}^{S^2}$ oznacza sumę po wszystkich komórkach siatki, $\sum_{j=0}^B$ oznacza sumę po wszystkich polach ograniczających w każdej komórce, $1_{ij}^{\text{obj}} = 1$ jeżeli j -ty prostokąt ograniczający w komórce i jest odpowiedzialny za wykrywanie obiektu, w przeciwnym razie 0. λ_{coord} zwiększa wagę dla straty współrzędnych pola ograniczającego, zaś $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$ oznaczają po kolei prognozowane środki, szerokość i wysokość.

Błędy bezwzględne nie powinny być ważone w taki sam sposób w dużych i małych polach, tzn. błąd 2-pikseli w dużym polu jest taki sam dla małego pola. W tym celu TinyYolo przewiduje pierwiastek kwadratowy szerokości i wysokości pola ograniczającego. Dodatkowo, aby położyć większy nacisk na dokładność prostokąta, mnożymy stratę przez λ_{coord} (domyślnie: 5).

2.3.1.4 Strata pewności

Jeśli obiekt zostanie wykryty w prostokącie, to strata pewności (mierzenie obiektywości prostokąta, czy znaleziono jakiś obiekt) jest liczona następującym wzorem:

$$\text{ConfL} = \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2, \quad (2.8)$$

gdzie \hat{C}_i to prognozowane confidence score pola j w celi i , $1_{ij}^{\text{obj}} = 1$ jeżeli j w polu ograniczającym w komórce i odpowiada za wykrywanie obiektu, w przeciwnym razie 0.

Jeśli obiekt nie zostanie wykryty w polu, strata pewności jest przedstawiona następującym wzorem:

$$\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2, \quad (2.9)$$

gdzie $1_{ij}^{\text{no obj}} = 1$ jeżeli obiekt nie występuje w komórce i , w innym wypadku 0. \hat{C}_i to prognozowane confidence score pola j w celi i , λ_{noobj} obciąża stratę przy wykrywaniu tła.

Większość pól nie zawiera żadnych obiektów. Powoduje to problem nierównowagi klasowej, tzn. trenujemy model tak, aby wykrywał tło częściej niż wykrywał obiekty. By zaradzić temu problemowi, obniżamy wagę tej straty o współczynnik λ_{noobj} (domyślnie: 0.5).

2.3.1.5 Dokładność modelu

W celu obliczenia czułości (recall) i precyzji (precision) zaznaczania potrzebne jest wyliczenie macierzy błędów. Do uzyskania prawdziwych pozytywów i fałszywych pozytywów wykorzystywane jest IoU. Używając IoU, należy zidentyfikować, czy wykrywanie (pozytywne) jest poprawne (prawda) czy nie (fałszywe). Najczęściej stosowanym progiem jest 0,5 - tzn. jeżeli IoU jest $> 0,5$, to detekcja uważana jest za prawdziwie pozytywną, w przeciwnym razie jest uważana za fałszywie pozytywną. Do obliczenia czułości, należy policzyć negatywne predykcje. Ponieważ każda część obrazu, w której nie przewidziano obiektu jest uważany za negatyw, mierzenie prawdziwych negatywów jest nieco bezcelowe. Mierzymy więc tylko „fałszywe” negatywy, czyli obiekty, które model przeoczył. Innym czynnikiem, który jest brany pod uwagę, jest pewność, że model zgłasza każde wykrycie. Poprzez zmianę naszego progu ufności możemy zmienić to, czy przewidywane pole jest pozytywne czy negatywne. Zasadniczo, wszystkie przewidywania (prostokąt+klasa) powyżej progu są uważane za pozytywne i wszystkie poniżej są negatywne. Każde zdjęcie posiada nałożone ręcznie pola ograniczające, które mówią nam o liczbie rzeczy-

wistych obiektów danej klasy w tym obrazie. Następnie obliczane jest IoU z ręcznie nałożonymi ramkami dla każdego pozytywnego pola detekcji, które zauważył model. Korzystając z tej wartości i naszego progu IoU (0,5), obliczana jest liczba poprawnych detekcji(A) dla każdej klasy w obrazie. Służy to do obliczania precyzji dla każdej klasy.

$$\text{Precyzja} = TP / (TP + FP) \quad (2.10)$$

Dzięki otrzymanej już liczbie poprawnych predykcji (A)(prawdziwe pozytywy) i brakującym wykrycia (fałszywe negatywy), można teraz obliczyć czułość (A/B) modelu dla tej klasy przy użyciu następującej formuły:

$$\text{Czułość} = TP / (TP + FN) \quad (2.11)$$

Zarówno precyzja jak i czułość wyliczane są dla każdej epoki. Popularna obecnie metryka pokazująca dokładność wykrywania obiektów mAP (mean average Precision) została po raz pierwszy sformalizowana w ramach wyzwania PASCAL Visual Objects Classes (VOC) w 2007 roku, które obejmowało różne zadania przetwarzania obrazu. W artykule Everingham, Van Gool, Williams, Winn i Zisserman (2010) autorzy zalecają obliczenie miary zwanej AP (Average Precision), czyli średniej precyzji. AP sumuje kształt krzywej precyzji/czułości i jest określana jako średnia precyzja przy zestawie jedenastu jednako rozłożonych poziomów czułości [0,0.1,...,1]. maP (mean average Precision) jest więc średnią wszystkich średnich wartości precyzji we wszystkich klasach.

Miara maP obliczana jest następującym wzorem:

$$\frac{1}{|\text{klasy}|} \sum_{c \in \text{klasy}} \frac{\#TP(c)}{\#TP(c) + \#FP(c)} \quad (2.12)$$

2.3.2 Zalety TinyYolo

- Przewidywania (lokalizacje i klasy obiektów) są tworzone z jednej sieci. Model może być przeszkolony od początku do końca w celu zwiększenia dokładności.
- TinyYolo ma dostęp do całego obrazu w przewidywaniu granic. W dodatkowym kontekście TinyYolo wykazuje mniej wyników fałszywie pozytywnych w obszarach tła.
- TinyYolo wykrywa jeden obiekt na komórkę siatki. Wymusza to zróżnicowanie prze-

strzenne w tworzeniu prognoz.

- Szybkie tworzenie modeli. Dobre do przetwarzania w czasie rzeczywistym.

2.3.3 Wykorzystane funkcje Turi Create

W ramach pracy dyplomowej wykorzystano narzędzie do wykrywania obiektów (Object Detection). Poniżej znajduje się tabela wykorzystanych funkcji z pakietu wraz z opisem działania.

Nazwa funkcji	Opis
config.set_num_gpus	O określa liczbę jednostek GPU do wykorzystania
image_analysis.load_images	Ładowanie zdjęć z katalogu. Obsługiwane są obrazy JPEG i PNG.
SFrame	Tabelaryczny, czytelny w kolumnach obiekt ramki danych, który może się skalować. Dane w SFrame są przechowywane w kolumnach i są przechowywane na trwałym nośniku (np. na dysku), aby uniknąć ograniczeń związanych z wielkością pamięci.
join	Połączenie dwóch ramek SFrame. Łączy bieżącą (lewy) SFrame z podaną (prawą) SFrame używając operacji join w stylu SQL.
random_split	Przypadkowy podział wierszy SArray na dwa SArray. Pierwszy SArray zawiera wiersze M, próbkiowane równomiernie (bez zamiany) z oryginalnego SArray.
save	Zapisuje SFrame w systemie plików do późniejszego wykorzystania.
object_detector.create	Tworzy model do wykrywania obiektów
evaluate	Ocenia model, dokonując przewidywań i porównując je z ręcznie nałożonymi ramkami ograniczającymi
predict	Przewiduje obiekty na podstawie zdjęć z SFrame
object_detector.util.draw_bounding_boxes	Wizualizuje pola ograniczające (ręcznie nałożone lub z prognozy), na niesione na obrazek

Tabela 2.2. Opracowanie własne na podstawie dokumentacji Turi Create

```

# Import najważniejszych bibliotek
1
import turicreate as tc
2
import sys
3
import os
4
import pandas as pd
5
import json
6
import mxnet
7
    # Wykorzystanie wszystkich dostępnych kart graficznych
8
tc.config.set_num_gpus(-1)
9
    # Załadowanie 6240 zdjęć oraz pliku, w którym znajdują się adnotacje
10
images = tc.image_analysis.load_images('/root/ALL')
11
annotations = tc.SFrame('/root/annotations-serwer-aws.csv')
12
    # Połączenie ramki danych ze zdjęciami oraz adnotacjami. Wydzielenie 80%
13     # jako danych do treningu oraz 20% do testowania modelu
14
data = images.join(annotations)
15
train, val = data.random_split(0.8)
16
    # Uruchomienie treningu na zbiorze train na długość 10000 iteracji
17
model = tc.object_detector.create(train, max_iterations = 10000, feature='image',
    annotations='annotations')
```

Program 2.1. Przykładowe wywołanie programu Turi Create program w Python

Powyższy Program 2.1 po wczytaniu pakietów wykorzystuje wszystkie dostępne karty graficzne. Następnie wczytywane są dane do formatu SFrame, by później wczytać adnotacje. Zdjęcia jak i adnotacje są łączone w jeden obiekt SFrame, aby zostać wprowadzonym jako argument do funkcji detekcji obiektu.

2.4 Podsumowanie

W temacie głębokich sieci neuronowe na przestrzeni ostatnich lat nastąpił ogromy przełom – możliwe stało się rozwiązywanie problemów przy pomocy maszyn, które uchodziły za bardzo trudne. Popularność uczenia głębokiego skutkowała rozwojem coraz większej ilości bibliotek do rozpoznawania obrazów. Jednym z takich pakietów jest Turi Create, którego przewagą nad zwykłym algorytmem klasyfikującym jest informacja zwrotna, pokazująca co zostało uznane za poszukiwany obiekt. Wiedza ta może ludziom pomóc lepiej zrozumieć działanie algorytmów, a następnie pomóc w jej optymalizacji. Wyniki wykorzystania pakietu będą przedstawione w następnym rozdziale.

Rozdział 3

Zastosowanie sieci neuronowych w rozpoznawaniu paneli fotowoltaicznych i kolektorów słonecznych

3.1 Pozyskanie i przetwarzanie danych przestrzennych

3.1.1 Środowisko do obliczeń

W pracy wykorzystano dwa środowiska do obliczeń – infrastrukturę InnoUEP oraz Amazon Web Services. Pierwsze środowisko powstało w ramach Programu Operacyjnego Innowacyjna Gospodarka w ramach, którego utworzono Laboratorium Interdyscyplinarnych Badań Naukowych UEP (InnoUEP), które składa się z kilku specjalistycznych, tematycznych pracowni badawczych ze wspólną bazą obliczeniową i przechowywania danych. Na potrzeby badań wykorzystano maszynę z 30 procesorami, 120 GB RAM oraz 1 TB przestrzeni dyskowej. Infrastruktura InnoUEP została wykorzystana do przygotowania danych na potrzeby uczenia maszynowego.

Na potrzeby głębokiego uczenia wykorzystano środowisko chmurowe oferowane przez Amazon. Amazon Web Services to największa chmura na świecie, która oferuje ponad 140 usług. Najbardziej znane z nich to Amazon EC2 (Amazon Elastic Compute Cloud) – to usługa dostarczająca moc obliczeniową w chmurze. Do obliczeń została wykorzystana maszyna o nazwie p2.xlarge kosztującą około 3.5 zł za godzinę obliczeń. Powyższa maszyna wykorzystuje jedną kartę graficzną NVIDIA Kesla K-80, która przyspieszy kilkukrotnie obliczenia ze względu na większą liczbę procesorów. Dodatkowo wybrana została maszyna wirtualna wraz z oprogramo-

wanie Compute Unified Device Architecture (CUDA) w wersji 8. CUDA jest platformą obliczeń równoległych i modelem programowania opracowanym przez firmę NVIDIA, która znaczco przyspiesza działanie aplikacji obliczeniowych dzięki wykorzystaniu kart graficznych.

Proces instalacji Turi Create wyglądał następująco:

- Instalacja dystrybucji Anaconda – narzędzia m.in. do tworzenia wyizolowanych środowisk Pythona.
- Pakiet libstdc++ – zbiór podstawowych funkcji napisanych w języku C++.
- Utworzenie wyizolowanego środowiska.
- Instalacja Turi Create wewnętrz stworzonego środowiska.

3.1.2 Zdjęcia lotnicze miasta Poznania i ich przetwarzanie

Na potrzeby badania pozyskano zdjęcia lotnicze (ortofotomapy) miasta Poznania z Zarządu Geodezji i Katastru Miejskiego GEOPOZ. Dane zostały pozyskane bezpłatnie na podstawie "Wniosku o udostępnienie materiałów powiatowego zasobu geodezyjnego i kartograficznego" w celu prowadzenia badań naukowych/prac rozwojowych w myśl ustawy Prawo geodezyjne i kartograficzne (art. 40a ust. 2 ustawy PGiK). Nalot w celu wykonania zdjęć dokonano między 2016-04-30, a 2016-05-05. Zdjęcia charakteryzowały się bardzo dobrą rozdzielczością, mianowicie jeden piksel oznaczał 5 cm. Dla porównania, zdjęcia satelitarne charakteryzują się dokładnością na poziomie co najwyżej 20 cm. Zbiór danych posiadał 3 808 zdjęć w formacie TIF i ważył około 100 GB.

Drugim zbiorem pozyskanym na podstawie tej samej ustawy i wniosku była warstwa budynków oraz działek zapisana w formacie ShapeFile (SHP). Aktualizacja pliku zawierającego informacje o budynkach dokonywana jest w sposób ciągły. Dlatego warstwa budynków była aktualna zgodnie z datą wniosku tj. na dzień 2019.03.06. Plik SHP zawierał 100 973 budynków z czego 92 344 stanowiły wybudowane, a 49 398 budynki o charakterze mieszkalnym. Należy zaznaczyć, że informacja o wybudowaniu była aktualna zgodnie ze stanem pliku SHP, więc niektóre budynki mogły nie być wybudowane w okresie przeprowadzenia nalotu ze zdjęciami lotniczymi.

Do wczytania i przetwarzania danych wykorzystano język R (R Core Team, 2018) oraz następujące pakiety sf (Pebesma, 2018), raster (Hijmans, 2019) oraz parallel. Jako infrastrukturę do obliczeń wykorzystano wspomniane wcześniej InnoUEP.

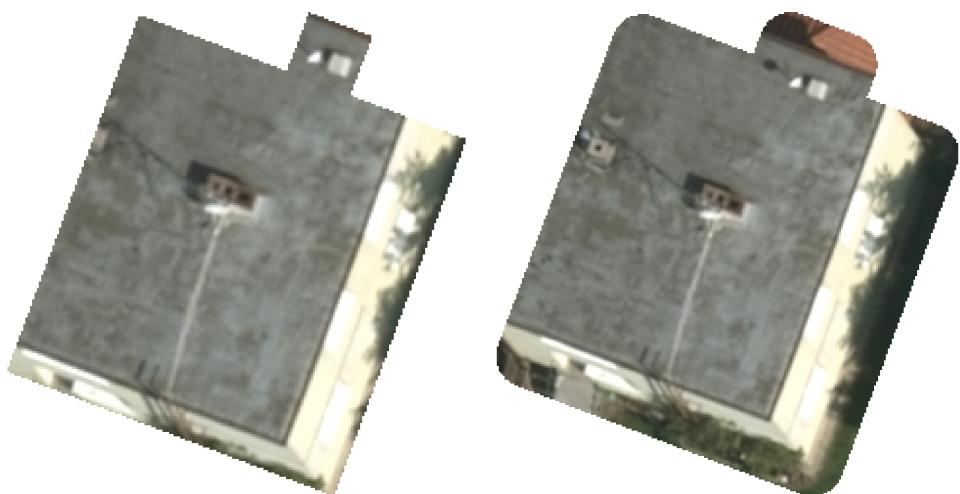
Po wczytaniu powyższych danych należało dokonać ich odpowiedniej redukcji. W pierwszym kroku ograniczono się wyłącznie do budynków mieszkalnych, które miały określony status „wybudowane”. W ten sposób uzyskano 46 985 budynków. Następnie, sprawdzono ile budynków znajduje się wyłącznie w jednym pliku TIF, a ile na większej ich liczbie. Ten rozkład przedstawia tabela 3.1. Przykładowo, wartość 1 i odpowiadająca im liczba 42 147 oznaczała budynki, które były wyłącznie na jednym zdjęciu, podczas gdy wartość 4 i liczba 83 wskazywała na budynki, które były jednocześnie ujęte na 4 zdjęciach lotniczych. Na potrzeby niniejszej pracy ograniczono się wyłącznie do budynków, które znajdowały się na jednym pliku TIF.

Tabela 3.1. Rozkład liczby budynków według występowania na zdjęciach lotniczych

Liczba zdjęć lotniczych	Liczba budynków
1	42 147
2	4685
3	70
4	83

Źródło: Opracowanie własne.

W kolejnym kroku należało wyciąć ze zdjęć lotniczych budynki. Należało jednak w tym przypadku wziąć pod uwagę, że zdjęcia mogły być robione pod kątem. Rysunek 3.1 przedstawiają odpowiednio budynek wycięty zgodnie z warstwą budynków oraz z uwzględnieniem bufora o szerokości 0.8 metra.



Rysunek 3.1. Przykład budynku wyciętego ze zdjęć lotniczych z wykorzystaniem warstwy budynków bez wykorzystania (lewy) oraz z wykorzystaniem bufora (prawy).

Źródło: Opracowanie własne.

Zastosowanie rozszerzenia zdjęcia o określony bufor pomogło zniwelować wpływ kąta wy-

konywanych zdjęć lotniczych. Tak utworzone zdjęcia zostały wykorzystane w kolejnym etapie czyli przygotowania pod algorytm uczenia maszynowego.

3.1.3 Przygotowanie danych na potrzeby uczenia głębokiego

W przypadku użycia narzędzia detekcji obiektu pakiet Turi Create wymaga załadowania nie tylko zdjęć, ale również adnotacji ręcznych. Plik powinien być przygotowany w formacie csv i mieć następującą strukturę przedstawioną w tabeli 3.2.

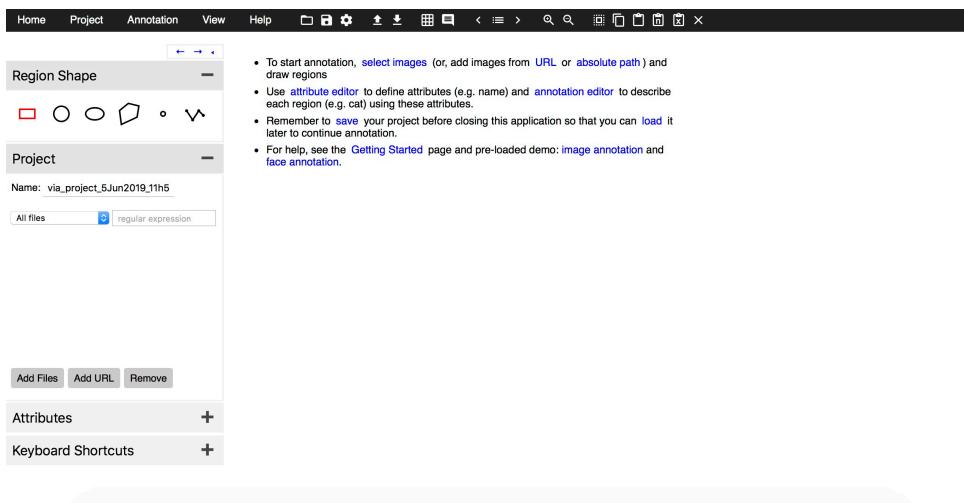
Tabela 3.2. Przykład zbioru określającego położenie obiektów na zdjęciach zgodnie ze strukturą używaną przez Turi Create

path	label	annotations
ściezka do plików	etykieta określająca oznaczane pole	[{"type": "rectangle", "coordinates": {"x": 1207, "y": 813, "width": 219, "height": 197}, "label": "panel"}]

Źródło: Opracowanie własne

Pole ograniczające jest definiowane przez cztery wartości (x , y , szerokość, wysokość), gdzie $(0,0)$ jest lewym górnym rogiem. Środek prostokąta znajduje się przy (x,y) . Przykładowe oznaczenie prezentuje się następująco:

Do oznaczania został wykorzystany darmowy program o nazwie VGG Image Annotator (VIA)¹. Proces zaznaczania pól ograniczających na zdjęciach trwał około 7 godzin. VIA działa w przeglądarce internetowej i nie wymaga żadnej instalacji ani konfiguracji. Kompletne oprogramowanie VIA to jedna strona HTML o rozmiarze mniejszym niż 400 kilobajtów, która działa jako aplikacja offline w większości nowoczesnych przeglądarek internetowych. Rysunek 3.3 przedstawia wygląd VIA.



Rysunek 3.3. Zrzut ekranu z programu VGG Image Annotator

¹Link do strony narzędzia: <http://www.robots.ox.ac.uk/vgg/software/via/>



Rysunek 3.2. Przykład ręcznego oznaczenia paneli lub kolektorów słonecznych w programie VGG Image Annotator

Zdjęcia, które będą oznaczane muszą znajdować się na dysku stacji roboczej bądź na dysku zewnętrznym. Poprzez „Add files” można wgrać wszystkie zdjęcie, które zostaną oznaczone. Lewa górną sekcja „Region shape” ustala kształt zaznaczania, Turi Create wymaga, aby adnotacje miały kształt prostokąta. W sekcji „Attribute name” ustala się nazwy etykiet jakie będą stosowane. Po ukończeniu oznaczania zdjęć należy wybrać z menu „Annotation”, a następnie „Export Annotations (as) csv”. Wygeneruje to plik o formacie csv. W otrzymanym pliku każdy wiersz odpowiada jednej adnotacji. W przypadku programu VGG Image Annotator wartości x i y odpowiadają współrzędnym lewego górnego rogu prostokąta ograniczającego. Wartości te musiały zostać przekształcone następującą formułą, aby reprezentować środek pola ograniczającego:

$$x_{\text{nowe}} = (x_{\text{stare}} + \text{width}/2). \quad (3.1)$$

Ostatnie przekształcenie jako miało miejsce na danych to połączenie kilku adnotacji z jednego obrazka, które zostały zapisane jako osobne wiersze przez VIA, w jeden wiersz.

Model w przypadku zbyt małej ilości danych nie jest w stanie stworzyć uogólnień, które sprawdzą się przy nowych danych. Augmentacja danych to technika polegająca na generowaniu większej ilości danych treningowych z istniejących plików, poprzez szereg losowych przekształceń, które dają wiarygodnie wyglądające obrazy. Ważną zasadą jest to, aby model nigdy nie zobaczył dwukrotnie tego samego zdjęcia. Dzięki większej próbce model może lepiej uogólniać oraz zauważać więcej szczegółów. Początkowa liczba zdjęć zawierających panele to 390. Dzięki obróceniu każdego zdjęcia o 90, 180 i 270 stopni oraz zaaplikowaniu lustrzanego odbicia uzyskano 3120 zdjęć zawierających panele.

3.2 Wyniki detekcji kolektorów i paneli słonecznych

3.2.1 Ewaluacja zastosowanego modelu

Trenowanie modelu w chmurze obliczeniowej AWS trwało ponad 10 godzin. Liczba iteracji jaką wykonał model w trakcie trenowania to 10 000. Zbiór zdjęć złożony był z 6 240 zdjęć, z czego połowa to zdjęcia zawierające panele. Treningowy zbiór zawierał 4 992 obrazów czyli 80% całego zbioru zaś zbiór testowy był złożony z 20% obrazów. Do losowania próbek wykorzystano losowanie proste. Końcowa strata modelu wyniosła 0.5172. Jak już wspomniano w 2 rozdziale, powszechnym sposobem ustalenia, czy pole ograniczające z predykcji jest słuszne, jest Intersection over Union (IoU). W przypadku stworzonego modelu $\text{IoU} > 0,5$ oznacza, że był to sukces, w przeciwnym razie był to błąd. Dla każdej klasy, można obliczyć:

- prawdziwe pozytywna (True Positives, TP) - model przewidział klasę c i taka klasa istniała,
- błąd pierwszego rodzaju (False Positives, FP) - model przewidział klasę c, ale taka klasa nie istniała.

Tabela 3.3. Wynik klasyfikacji na zbiorze uczącym

Prawda / Prognoza	Panel	No Panel	Razem
Panel	2497	0	2497
No Panel	148	2322	2470

Źródło: Opracowanie własne na podstawie wyników modelu. Określenie *Panel* informuje czy na danym zdjęciu widniała instalacja fotowoltaiczna lub kolektory słoneczne, a *No Panel* oznacza brak takich instalacji.

Jak wskazują wyniki w tabeli 3.3 dodatnia wartość predykcyjna inaczej precyza wynosi 0.944. Wartość precyzji wyliczany jest w następujący sposób

$$\text{precyzja} = TP / (TP + FP). \quad (3.2)$$

Precyzja mierzy, jak dokładne są przewidywania, tzn. jaki procent przewidły jest prawidłowy. *Recall* czyli czułość, która na próbie treningowej wynosi 1, wyliczana jest z następującego wzoru:

$$text{Recall} = TP / P = TP / (TP + FN), \quad (3.3)$$

która interpretuje się jako zdolność testu do prawidłowego rozpoznania panelu tam, gdzie on występuje. Czułość 100% oznacza, że wszystkie zdjęcia zawierające panele zostaną rozpoznane.

Tabela 3.4. Wynik klasyfikacji na zbiorze testowym

Prawda / Prognoza	Panel	No Panel	Razem
Panel	621	2	623
No Panel	50	600	650

Źródło: Opracowanie własne na podstawie wyników modelu. Określenie *Panel* informuje czy na danym zdjęciu widniała instalacja fotowoltaiczna lub kolektory słoneczne, a *No Panel* oznacza brak takich instalacji.

Otrzymany współczynnik maP na próbie testowej wynosi 0.9326. Jak wskazują wyniki w tabeli 3.4 precyza wynosi 0.925. Czułość zaś wynosi 0.9997. Wypisanie nazwy modelu skutkuje wypisaniem podsumowania modelu, który prezentuje się następująco:

```

Class : ObjectDetector

Schema
-----
Model : darknet-yolo
Number of classes : 1
Non-maximum suppression threshold : 0.45
Input image shape : (3, 416, 416)

Training summary
-----
Training time : 10h 45s
Training epochs : 64
Training iterations : 10000
Number of examples (images) : 4967
Number of bounding boxes (instances) : 2972
Final loss (specific to model) : 0.5172

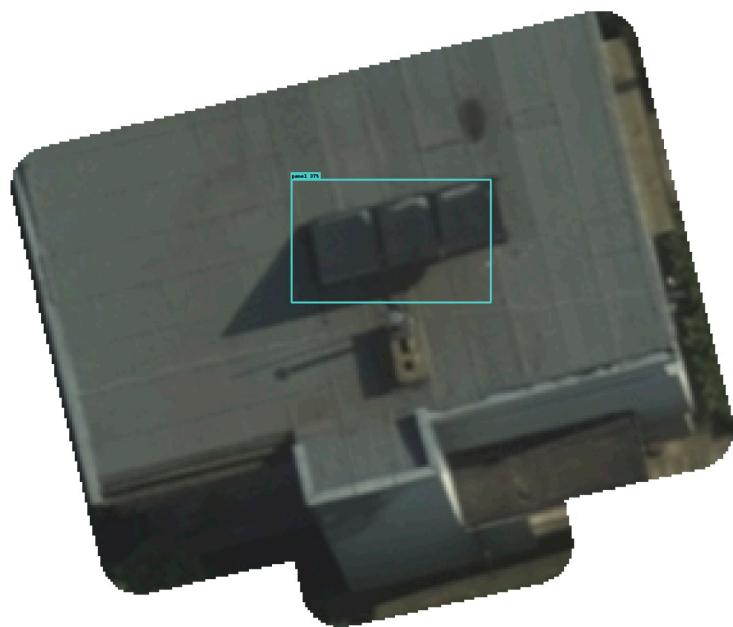
```

Rysunek 3.4. Podsumowanie wytrenowanego modelu

Źródło: Opracowanie własne.

Jak ukazuje pierwszy wiersz rysunku 3.4, wykorzystano klasę *ObjectDetector*, która odpowiada za rozpoznawanie objektów. Następnie została wyświetlona nazwa wykorzystywanej sieci konwolucyjnej (darknet-yolo). *Number of classes (1)* to liczba klas jakie wykorzystano w modelu. *Non-maximum suppression threshold* jest wartością IoU, od której zachowywane są propozycje predykcji ramek ograniczających. Wszystkie ramki posiadające IoU mniejsze niż 0.45 były odrzucane. *Input image shape* to wymiary wprowadzanych zdjęć (416, 416, 3).

Kolejna sekcja to podsumowanie trenowania modelu. *Training time* wynoszący 10 godzin i 45 sekund ukazuje czas przez jaki się model trenował. Następnie ukazana jest liczba epok, która wyniosła 64. Zaś liczba iteracji to 10 000. Różnica między epoką, a iteracją jest taka, że jedna epoka oznacza wszystkie dane przetworzone jeden raz przez sieć, a iteracji oznacza jedną przetworzoną partię danych (domyślnie parta danych w Turicreate oznacza próbkę 512 zdjęć). *Number of examples* to liczba zdjęć w próbce treningowej, która liczy 4 967 zdjęć. *Number of bounding boxes* wynosi 2972 i oznacza liczbę pól ograniczających w próbie treningowej. Ostatnią wartością w podsumowaniu jest końcowa strata (ang. *final loss*), która wskazuje na 0.52. Dzięki wykorzystaniu funkcji *turicreate.object_detector.util.draw_bounding_boxes* można zwizualizować predykcję pól ograniczających na niesionych na obrazek.



Rysunek 3.5. Przykład poprawnej klasyfikacji kolektora lub panela słonecznego (1)

Źródło: Opracowanie własne.

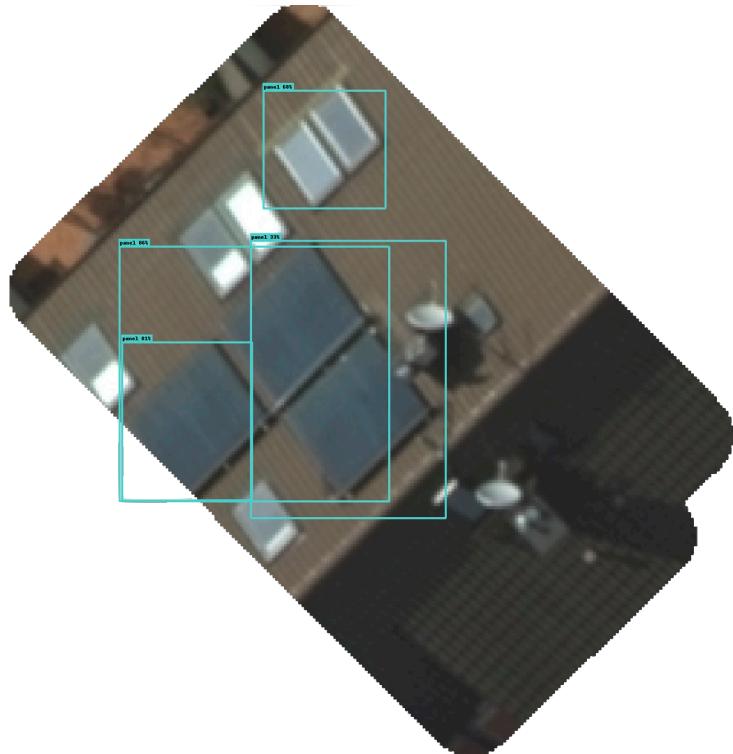
Rysunek 3.5 przedstawia prawidłowo sklasyfikowane zdjęcie. Instalacja fotowoltaiczna została rozpoznana przez model i zaznaczona polem ograniczającym. Jednak w tym przypadku pewność zaznaczenia wynosi jedynie 37%. Przyczyną niskiej pewności może być cień jaki rzuca instalacja, który zaburza pewność predykcji.



Rysunek 3.6. Przykład poprawnej klasyfikacji kolektora lub panela słonecznego (2)

Źródło: Opracowanie własne.

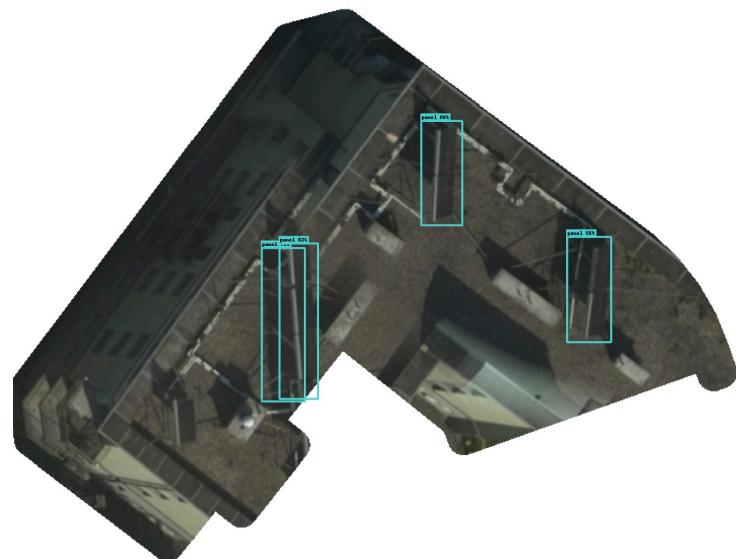
Rysunek 3.6 to prawidłowa predykcja modelu. Instalacja fotowoltaiczna została oznaczona pomimo występowania okien, które wyglądem przypominają panele. Ramka wskazuje dokładność na poziomie 87%.



Rysunek 3.7. Przykład częściowo poprawnej klasyfikacji kolektora lub panela słonecznego

Źródło: Opracowanie własne.

Rysunek 3.7 prezentuje częściowo poprawną predykcję. Oprócz instalacji fotowoltaicznych zostały również zaznaczone okna. Przyczyną błędnej predykcji jest kształt oraz wielkość okien, które w znacznym stopniu przypominają panele.



Rysunek 3.8. Przykład niepełnej klasyfikacji paneli lub kolektorów słonecznych

Źródło: Opracowanie własne.

Rysunek 3.8 przedstawia po części prawidłowo sklasyfikowane zdjęcie. Trzy z czterech paneli fotowoltaicznych zostało rozpoznanych. Przyczyną nie rozpoznania panelu w lewym dolnym rogu zdjęcia może być różniąca się wielkość wobec pozostałych instalacji fotowoltaicznych.



Rysunek 3.9. Przykład niewykrycia panelu lub kolektora słonecznego

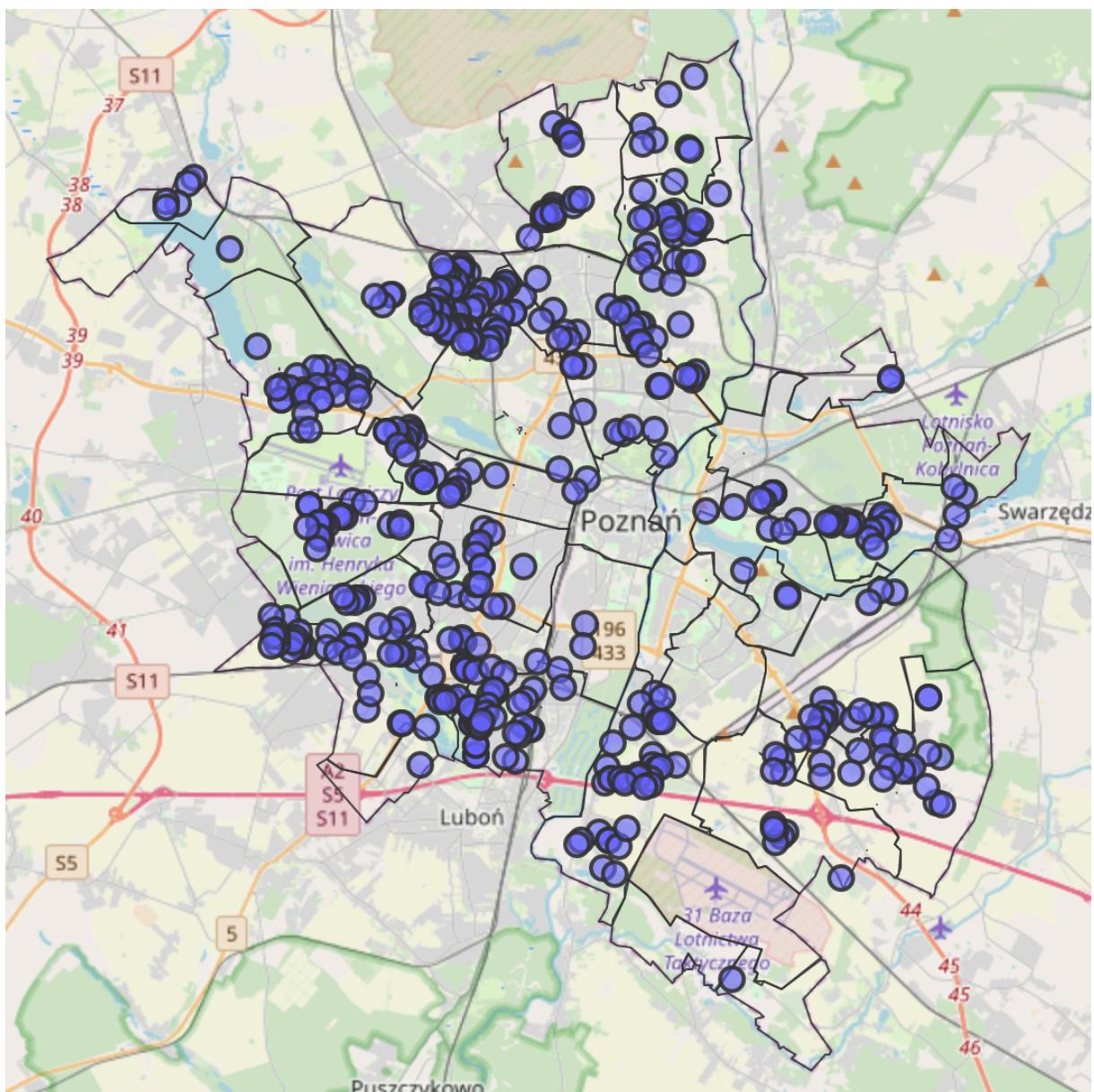
Źródło: Opracowanie własne.

Rysunek 3.7 to nieprawidłowo sklasyfikowane zdjęcie. Model nie zaznaczył panelu, który znajduje się w lewym dolnym rogu obrazka. Możliwym powodem nie wykrycia panelu jest fakt, że instalacja znajduje się przed oknem zatem algorytm postrzega okno jak i panele fotowoltaiczne jako jeden obiekt.

3.2.2 Lokalizacja paneli i kolektorów słonecznych w Poznaniu w 2016 roku

Na wytrenowanym modelu można wywołać funkcję predict z pakietu Turi Create wraz z nazwą zbioru, na którym mają zostać rozpoznane panele słoneczne lub kolektory. Przeanalizowany został cały zbiór zdjęć budynków mieszkalnych w Poznaniu, czyli 42 000 zdjęć. Dzięki modelowi udało się zidentyfikować 476 instalacji fotowoltaicznych i kolektorów słonecznych. Zaprognozowana liczba została sprawdzona poprzez ręczne przeglądanie zdjęć. Wynikiem wywołania funkcji jest plik JSON, z którego można wydobyć identyfikatory budynków identyczne do tych z danych GEOPOL. Listę budynków posiadających panel lub kolektor można nałożyć na obręby Poznania.

Rysunek 3.5 przedstawia rozkład instalacji fotowoltaicznych i kolektorów w poszczególnych obrębach Poznania, których jest 40. W 2016 roku w 33 obrębach Poznania znajdowała się co najmniej jedna instalacja fotowoltaiczna lub kolektor słoneczny. Odsetek budynków mieszkalnych zawierających instalację fotowoltaiczną i kolektor słoneczny wynosi 1.13%. Największa liczba paneli i kolektorów (38) znajduje się na Dębcu czyli południowej części miasta. Najmniej zaś wykryto między innymi na Śródce, Psarskim, Piotrowie, Kotowie, gdzie model rozpoznał jedną instalację. Największy odsetek paneli i kolektorów względem liczby budynków znajduje się w obrębie Morasko, gdzie 5.41% budynków mieszkalnych posiada instalację fotowoltaiczną i kolektor słoneczny.



Rysunek 3.10. Obręby Poznania – lokalizacja OZE w 2016r.

Źródło: Opracowanie własne.

Tabela 3.5 przedstawia rozkład rozpoznanych instalacji fotowoltaicznych w poszczególnych obrębach Poznania.

Tabela 3.5. Rozkład instalacji fotowoltaicznych w poszczególnych obrębach Poznania w 2016 roku

Obręb	Liczba budynków mieszkalnych	Liczba z OZE	Odsetek
DĘBIEC	3844	38	0.99
STRZESZYN	1710	34	1.99
STAROŁĘKA	2052	33	1.61
KRZYŻOWNIKI	2449	28	1.14
JUNIKOWO	2299	27	1.17
GOŁĘCIN	2563	25	0.98
NARAMOWICE	1824	23	1.26
GÓRCZYN	2125	22	1.04
JEŻYCE	1931	22	1.14
GŁOWIENIEC	1314	20	1.52
MORASKO	370	20	5.41
SPŁAWIE	1154	20	1.73
UMULTOWO	987	19	1.93
KOBYLEPOLE	1377	18	1.31
ŁAWICA	2158	18	0.83
ŁAZARZ	2748	17	0.62
PLEWISKA	1391	16	1.15
PODOLANY	831	13	1.56
KRZESINY	779	11	1.41
WINIARY	1480	11	0.74
GŁÓWNA	1456	7	0.48
PIĄTKOWO	1044	7	0.67
RADOJEWKO	420	7	1.67
KIEKRZ	578	5	0.87
KOMANDORIA	415	3	0.72
CHARTOWO	139	2	1.44
KAROLIN	38	2	5.26
RATAJE	260	2	0.77
WILDA	662	2	0.30
KOTOWO	213	1	0.47
PIOTROWO	53	1	1.89
PSARSKIE	168	1	0.60
ŚRÓDKA	196	1	0.51
DASZEWICE	23	0	0.00
GŁUSZYNA	109	0	0.00
GŁUSZYNA II	6	0	0.00
ŁAWICA II	4	0	0.00
POZNAŃ	781	0	0.00
WIELKIE	4	0	0.00
ZĘGRZE	192	0	0.00

Źródło: Opracowanie własne.

Podsumowanie

Celem niniejszej pracy było opracowanie algorytmu uczenia maszynowego do rozpoznawania ze zdjęć lotniczych miasta Poznania paneli i kolektorów słonecznych. W tym celu wykorzystano do analizy dane pozyskane z Zarząd Geodezji i Katastru Miejskiego GEOPOZ na podstawie „Wniosku o udostępnienie materiałów powiatowego zasobu geodezyjnego i kartograficznego”.

Przygotowanie danych stanowiło kluczowy element wpływający na wyniki. Proces ten składał się z kilku kroków. Po pierwsze ze zdjęć lotniczych należało wyciąć budynki mieszkalne. W tym celu wykorzystano warstwę budynków zapisaną w formacie SHP, która również została udostępniona przez GEOPOZ. W drugim kroku aby przygotować zbiór uczący i testowy dokonano manualnego przeglądania ponad 10 tysięcy zdjęć celem wskazania tych zawierających panele i kolektory słoneczne. W trzecim kroku dokonano augmentacji danych, czyli zwiększenia liczby obserwacji zawierających instalacje fotowoltaiczne i kolektory poprzez programistyczne obracanie zdjęć. Każde obrócone zdjęcie stanowiło dla algorytmu nowy przypadek. W kroku czwartym dokonano ręcznego oznaczania instalacji z wykorzystaniem darmowego programu VGG Image Annotator, które trwało około 7 godzin. Finalnym krokiem było zastosowanie algorytmu głębokich sieci neuronowych do detekcji paneli i kolektorów słonecznych. W tym celu zastosowano język Python oraz bibliotekę Turi Create, która zawiera reimplementację algorytmu You Only Look Once (ang. YOLO) z biblioteki darknet.

Obrazy wraz z adnotacjami zostały podzielone na zbiór treningowy (80%) oraz testowy (20%). Sieć w ciągu 10 godzin procesu uczenia wykonała 10 000 iteracji uzyskując bardzo dobrą skuteczność na poziomie 93%. Wartość funkcji straty po optymalizacji wyniosła 0.52. Czułość modelu wynosi 99.97% co oznacza prawie idealną skuteczność w rozpoznawaniu paneli i kolektorów. Natomiast, zaproponowany model w około 93% dobrze klasyfikował zdjęcia bez paneli i kolektorów. Oznacza to, że część zdjęć, która w rzeczywistości nie zawierała badanych instalacji została zaklasyfikowana do grupy budynków z panelami i kolektorami. Finalnie, zlokalizowano 476 instalacji na ponad 42 000 budynkach mieszkalnych w Poznaniu, czyli odsetek

wyniósł 1.13%.

Należy podkreślić, że w pracy wykorzystano dwie infrastruktury: InnoUEP, które jest środowiskiem przeznaczonym do prac badawczych na Uniwersytecie Ekonomicznym w Poznaniu oraz środowisko Amazon Web Services (AWS) – Cloud Computing Services umożliwiające wykorzystanie m.in. kart graficznych do przyspieszenia obliczeń.

Zgodnie z wiedzą autora oraz promotora jest to pierwsze w Polsce zastosowanie głębowich sieci neuronowych do detekcji paneli oraz kolektorów słonecznych. Aktualnie analogiczne prace realizowane są przez m.in. Statistics Netherlands w ramach projektu DeepSolaris czy badaniu DeepSolar opracowywanego przez naukowców z Uniwersytetu Stanfordzkiego. Badanie wpisuje się w trend wykorzystania zdjęć lotniczych i satelitarnych dla potrzeb statystyki, wykorzystania zaawansowanych algorytmów uczenia maszynowego oraz oprogramowania open source.

Podsumowując, głębokie sieci neuronowe są bardzo dobrą metodą do detekcji obiektów. Ze względu na swoje potężne zdolności uogólniania i zalety w radzeniu sobie z dużymi zbiorami danych i zmieniającym się tłem, wykrywanie obiektów oparte na głębokim uczeniu się było w ostatnich latach punktem zwrotnym w badaniach naukowych.

Bibliografia

- Bogdanienko, J. (1989). *Odnawialne źródła energii*. Państwowe Wydawn. Naukowe.
- Chollet, J., F. i Allaire. (2018). *Deep Learning with R*. Manning Publications Company.
- Ciechanowicz, W. (1997). *Energia, środowisko i ekonomia*. Instytut Badań Systemowych PAN.
- EurObserv'ER. (2018a). PHOTOVOLTAIC BAROMETER. *Systemes Solaires*.
- EurObserv'ER. (2018b). Solar thermal and concentrated solar power barometer. *Systemes Solaires*.
- Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303–338.
- Fotowoltaiki, P. T. (2018). Rynek fotowoltaiki w Polsce – 2017 r.
- Fraunhofer Institute for Solar Energy Systems, P. C. C. G. (2018). PHOTOVOLTAICS REPORT.
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4), 193–202.
- GUS. (2016). ZASADY METODYCZNE BADAŃ STATYSTYCZNYCH Z ZAKRESU ENERGII ZE ŹRÓDEŁ ODNAWIALNYCH. *GUS*.
- Hijmans, R. J. (2019). *raster: Geographic Data Analysis and Modeling*. R package version 2.9-5.
- Hubel, D. H. & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3), 574–591.
- Jastrzębska, G. (2017). Energia ze źródeł odnawialnych i jej wykorzystanie. *Wydawnictwa Komunikacji i Łączności*, Warszawa.
- Matuszczyk, P., Popławski, T. & Flasza, J. (2015). Potencjał i możliwości energii promieniowania elektromagnetycznego Słońca. *Przegląd Elektrotechniczny*, 91(1), 183–187.
- McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133.

- Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, 10(1), 439–446. doi:10.32614/RJ-2018-009
- Perez, M. & Perez, R. (2015). Update 2015—A Fundamental Look at Supply Side Energy Reserves for the Planet. *Natural Gas*, 2(9), 215.
- R Core Team. (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria.
- Redmon, J. (2016). Darknet: Open Source Neural Networks in C. <http://pjreddie.com/darknet/>.
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.
- SEV/VSE, B. N. (1996). Menschen und Energie, Realität und Utopien, 13–16.
- Tadeusiewicz, R. (1998). *Elementarne wprowadzenie do techniki sieci neuronowych z przykładowymi programami*. Problemy Współczesnej Nauki: Informatyka. Akademicka Oficyna Wydawnicza.
- Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. *Ph. D. dissertation, Harvard University*.

Spis tabel

1.1	Możliwości pozyskiwania energii z wybranych źródeł w skali roku w TWrok, według Pereza i Perez (2015)	4
1.2	Porównanie wybranych aspektów dotyczących instalacji fotowoltaicznych i kolektorów słonecznych	8
2.1	Funkcjonalność pakietu Turi Creat	27
2.2	Opracowanie własne na podstawie dokumentacji Turi Create	35
3.1	Rozkład liczby budynków według występowania na zdjęciach lotniczych	39
3.2	Przykład zbioru określającego położenie obiektów na zdjęciach zgodnie ze strukturą używaną przez Turi Create	40
3.3	Wynik klasyfikacji na zbiorze uczącym	43
3.4	Wynik klasyfikacji na zbiorze testowym	43
3.5	Rozkład instalacji fotowoltaicznych w poszczególnych obrębach Poznania w 2016 roku	50

Spis rysunków

1.1	Przykładowa instalacja fotowoltaiczna na dachu	5
1.2	Kolektor słoneczny	6
1.3	Aktualność ortofotomapy w Państwowym Zasobie Geodezyjnym i Kartograficznym	12
1.4	Rozkład zainstalowanych systemów fotowoltaicznych według regionów	13
1.5	Całkowita moc instalacji fotowoltaicznych w Polsce	14
1.6	Skumulowana moc zainstalowana dla czterech największych OSD w MW	15
2.1	Uczenie maszynowe: nowy model programowania	17
2.2	Sztuczna inteligencja, uczenie maszynowe i uczenie głębokie	17
2.3	Typy tensorów	20
2.4	Budowa neuronu	21
2.5	Sieć neuronowa czterowarstwowa o strukturze 3-4-4-1	22
2.6	Przykład lokalnych wzorców uczenia konwolucyjnej sieci neuronowej	25
2.7	Przykład działania algorytmu YOLO	28
2.8	Część wspólna przez sumę zbiorów – decyzja o wyborze określonej obwiedni (bounding box)	29
2.9	Przykład działania algorytmu YOLO uwzględniający wiele możliwych lokalizacji paneli	30
2.10	Przykład działania algorytmu YOLO uwzględniający rozpoznawanie wielu obiektów	31
3.1	Przykład budynku wyciętego ze zdjęć lotniczych z wykorzystaniem warstwy budynków bez wykorzystania (lewy) oraz z wykorzystaniem bufora (prawy).	39
3.3	Zrzut ekranu z programu VGG Image Annotator	40

3.2 Przykład ręcznego oznaczenia paneli lub kolektorów słonecznych w programie VGG Image Annotator	41
3.4 Podsumowanie wytrenowanego modelu	44
3.5 Przykład poprawnej klasyfikacji kolektora lub panela słonecznego (1)	45
3.6 Przykład poprawnej klasyfikacji kolektora lub panela słonecznego (2)	45
3.7 Przykład częściowo poprawnej klasyfikacji kolektora lub panela słonecznego . .	46
3.8 Przykład niepełnej klasyfikacji paneli lub kolektorów słonecznych	47
3.9 Przykład niewykrycia panelu lub kolektora słonecznego	47
3.10 Obręby Poznania – lokalizacja OZE w 2016r.	49

Spis Programów

2.1 Przykładowe wywołanie programu Turi Create program w Python 35